Ch 1: Floating Point Numbers

Wednesday, August 27, 2025 6:35 PM

See lab Ch1_DataTypes.ipynb

Learning objectives:

- Distinguish underflow vs. below machine epsilon
- There's a largest float
- Spacing between floats scales relatively

How can we represent numbers on a computer?

Integers are easy. $6 = 1\cdot 4 + 1\cdot 2 + 0\cdot 1$, ie., 110 (binary)

What about fractions?

$$\underbrace{\frac{18}{19}}_{10\text{ bits}} + \underbrace{\frac{22}{23}}_{10\text{ bits}} = \frac{18\cdot 23 + 22\cdot 19}{19\cdot 23} = \frac{414+418}{437} = \underbrace{\frac{832}{437}}_{19\text{ bits}}$$

5 bits → 18, 5 bits → 19; 832) 10 bits, 437) 9 bits

Problem: memory increases

Could try decimals (fixed pt.)

$\underbrace{131}_{3}.\underbrace{467}_{3}$ or in binary $\ldots 10011 0.100110 \ldots$ ($2^2\ 2^1\ 2^0\ 2^{-1}\ 2^{-2}$)

Done in embedded system

Instead, the standard for numerical computing is floating pt.

Usually use IEEE 754 "double precision" → 64 bits = 8 bytes

(Single precision = 32 bits)

Store numbers

$\mathbb{F}$ = Floating Pt. $\quad (-1)^s\,\overbrace{(1+f)}^{\text{significand}}\cdot 2^e$ ← radix or base $\qquad e = c - 1023$

$\mathbb{R}$ = real numbers

s = sign bit (1 bit)

e = exponent or characteristic, 11 bits $\quad 2^{11} = 2048$

f = mantissa, 52 bits

For exponent, 11 bits, think of 1 sign bit so $\pm 2^{10}$ i.e. ± 1024

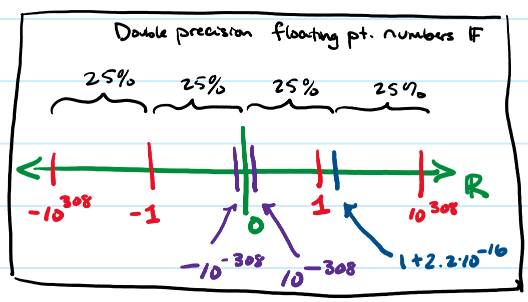
(scientific notation)

$\mathbb{F}$ also includes 0, NaN ($0/0$, $0\cdot\infty$, ∞/∞), $\pm\infty$

Rule of thumb: precision $\approx 2^{52} = 4.5\cdot 10^{15}$

15 digits of precision in double

8 digits ... in single



Implications

① We can't represent very large (or very negative) numbers

Due to limitations in exponent

$x \in \mathbb{F}$, then $|x| \leq 2^{1024} = 10^{308}$

ie., $x = -10^{400}$ is not in $\mathbb{F}$ (it is $-\infty$)

Overflow if not in range

Note: we're being a little loose w/ a few technicalities because we focus on the bigger message

② we can't get too small in magnitude (close to 0)

$x \in \mathbb{F}$, $|x| \geq 2^{-1022} \approx 10^{-308}$

underflow if $|x| < 2^{-1022} \approx 10^{-308}$, might be treated as 0

Ch 1: Floating Point Numbers, p. 2

Wednesday, August 27, 2025 6:40 PM

relative spacing

③ limit to spacing

"Machine epsilon", SILENT ERROR

due to limits in mantissa

$1 \overset{?}{==} 1+\varepsilon$, ie., 1 and $1+\varepsilon$ are indistinguishible

true if $\varepsilon < \varepsilon_{machine} = 2^{-52} \approx 2.2 \cdot 10^{-16}$

NOTE: youtube video is wrong, it is relative

2 vs $2+\varepsilon$, $\varepsilon < 2 \cdot \varepsilon_{machine}$

Def of ε_m: the difference between 1 and the next largest floating pt. number (aka "mainstream def" on wikipedia)

Notation: $x \in \mathbb{R}$, $fl(x)$ is nearest number to x that's in $\mathbb{F}$

$$\frac{|fl(x) - x|}{|x|} \le \tfrac{1}{2}\varepsilon_{machine}, \quad fl(x) = x\cdot(1-u), \text{ some } |u| \le \tfrac{1}{2}\varepsilon_{machine}$$

depends on precise definition

$$\left.\frac{|fl(x)-x|}{|x|}\right\} \text{relative error} = \text{accuracy}$$

$|fl(x)-x|$ = absolute error

digits of accuracy $\approx -\log_{10}(\text{rel. error})$

Precision : #digits, need not be correct!

Accuracy : relative error, limited by precision

More implications: lose associative rule

$$(a+b)+c = a+(b+c)$$

$$\underbrace{\underbrace{\left(1+\varepsilon_{machine}/2\right)}_{1} - 1}_{0} \overset{?}{\neq} 1 + \underbrace{\left(\varepsilon_m/2 - 1\right)}_{\varepsilon_m/2 \,\approx\, 1.11\cdot 10^{-16}}$$

Ex: softmax

$\log_{10}(10^{400} + 10^{400})$

$= \log_{10}(2) + 400$

but computer will overflow w/ naive implementation

Catastrophic Cancellation

Pretend we have 3 digits of precision.

$x = 1.23\overbrace{587325}^{\text{garbage}}\ldots$

$y = 1.22358192 6\ldots$

see demos

$x - y = 0.01\underbrace{\sim\sim\sim\sim\sim\sim}_{\text{garbage}}$

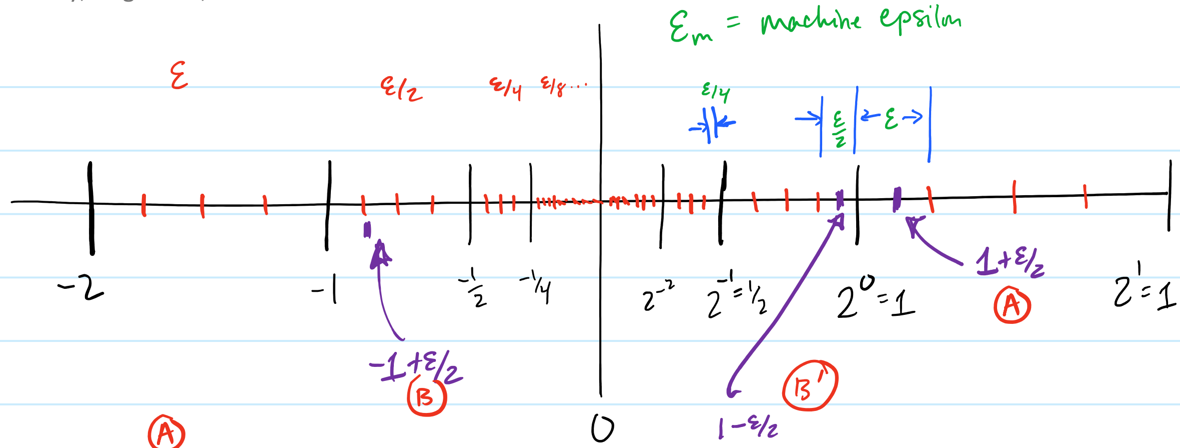
"Silent error"!

$= 1.\boxed{\sim\sim}\sim\sim\sim \cdot 10^{-2}$

you might think these are accurate since within our precision.

Ch 1: Floating Point Numbers, extra diagram

Thursday, August 28, 2025 8:10 AM



A

$$\underbrace{\left(1 + \varepsilon/2\right)}_{=1 \text{ in floating pt.}} - 1 = 1 - 1 = 0$$

B, B'

$$1 + (\varepsilon/2 - 1) = 1 - \underbrace{(1 - \varepsilon/2)}_{\text{this is representable in floating pt.}} = \varepsilon/2$$

Two definitions of ε_{mach}

1. $1 + \varepsilon_{mach}$ is next floating pt. number after 1

   (what I drew in picture) $\varepsilon = 2^{52} \approx$ 2.22e-16

   This is "variant" in wikipedia

2. Smallest number such that $1 + \varepsilon_m \neq 1$

   i.e., "unit roundoff" $\varepsilon = 2^{53} \approx$ 1.11e-16

   if we "round to nearest" then

   this is 1/2 the value from definition 1

In practice, we care that $\varepsilon_{mach} \approx 10^{-16}$, specific numbers not always important

2'. largest number ε such that $1 + \varepsilon$ rounds to 1

Take-home points:

Floating pt. numbers have almost a constant <u>relative</u> accuracy, hence <u>absolute</u> accuracy depends on magnitude

i.e., spacing of floating pt. #'s is <u>not</u> uniform.