UNIVERSITY OF LEEDS

Implementation of Stochastic Rounding

Mantas Mikaitis

School of Computing, University of Leeds, Leeds, UK

10th International Congress on Industrial and Applied Mathematics
MS 00827
Tokyo, Japan, Aug. 22, 2023 (Virtual)

Introduction

- In computer operations **round-to-nearest** (RN) is a default.
- Deterministic, optimal accuracy per operation.
- Closest machine number to real answer—cannot improve.
- Accumulates error of factor n, where n a problem size.

What we get from today's talk

Learn about hardware implementation of **stochastic rounding** (SR) which accumulates error of factor $\sqrt{n}$.

Floating-point (FP) number representation

A floating-point system $F \subset \mathbb{R}$ is described with $\beta, t, e_{min}, e_{max}$ with elements

$$x = \pm m \times \beta^{e-t+1}.$$

Virtually all computers have $\beta = 2$ (binary FP).

Here t is precision, $e_{min} \leq e \leq e_{max}$ an exponent, $m \leq \beta^p - 1$ a significand (m, t, e, $m \in \mathbb{Z}$).

Standard model [Higham, 2002]

Given $x \in \mathbb{R}$ that lies in the range of F it can be shown that

$$\mathrm{fl}(x \text{ op } y) = (x \text{ op } y)(1 + \delta), \quad |\delta| \leq u,$$

where $u = 2^{-t}$, $\mathrm{op} \in \{+, -, \times\}$ and **round-to-nearest** mode.

Rounding error analysis

Rounding errors δ accumulate. For example, consider computing $s = x_1y_1 + x_2y_2 + x_3y_3$.

We compute $\widehat{s}$ with

$$\begin{aligned}\widehat{s} &= \Big(\big(x_1y_1(1+\delta_1) + x_2y_2(1+\delta_2)\big)(1+\delta_3) + x_3y_3(1+\delta_4)\Big)(1+\delta_5)\\ &= x_1y_1(1+\delta_1)(1+\delta_3)(1+\delta_5) + x_2y_2(1+\delta_2)(1+\delta_3)(1+\delta_5)\\ &\quad + x_3y_3(1+\delta_4)(1+\delta_5).\end{aligned}$$

Therefore we deal with a lot of terms of the form $\prod_{i=1}^{n}(1+\delta_i)$.

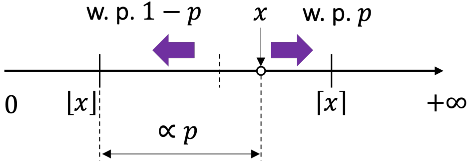
Worst case backward error bound (exact result for perturbed inputs)

$\prod_{i=1}^{n}(1+\delta_i) = 1+\theta_n, \quad |\theta_n| \leq \gamma_n$, with $\gamma_n = \frac{nu}{1-nu}$.

What is stochastic rounding

In **stochastic rounding** (**SR**), we are not rounding a number to the same direction, but to either direction with probability.

Given some x and FP neighbours $\lfloor x \rfloor$, $\lceil x \rceil$, we round to $\lceil x \rceil$ with prob. p and $\lfloor x \rfloor$ with $p - 1$.



Mode 1 SR (nearness): $p = \frac{x - \lfloor x \rfloor}{\lceil x \rceil - \lfloor x \rfloor}$ **Mode 2 SR**: $p = 0.5$

Mode 2

With **Mode 1 SR** we round x depending on its distances to the nearest two FP numbers, **cancelling out errors of different signs**.

Mode 1 SR example

Consider rounding real numbers to integers. Round 0.25 indefinitely and then consider running total error.

With **SR**, probability of rounding up is 0.25 and down is 0.75.

With **RN** the total error from n roundings is $-0.25n$.

With **SR**, we can assume we **round up on every 4th number**. Error growth:

$$\downarrow -0.25 \qquad \downarrow -0.5 \qquad \downarrow -0.75 \qquad \uparrow 0$$

$$\uparrow 0.75 \qquad \downarrow 0.5 \qquad \downarrow 0.25 \qquad \downarrow 0$$

Rounding error analysis with SR

Standard error model for SR

With SR we replace u by $2u$ since it can round to the second nearest neighbour in F.

Rounding error analysis

Worst-case error analysis determines the **upper bounds of errors**, while probabilistic error analysis describes **more realistic bounds**.

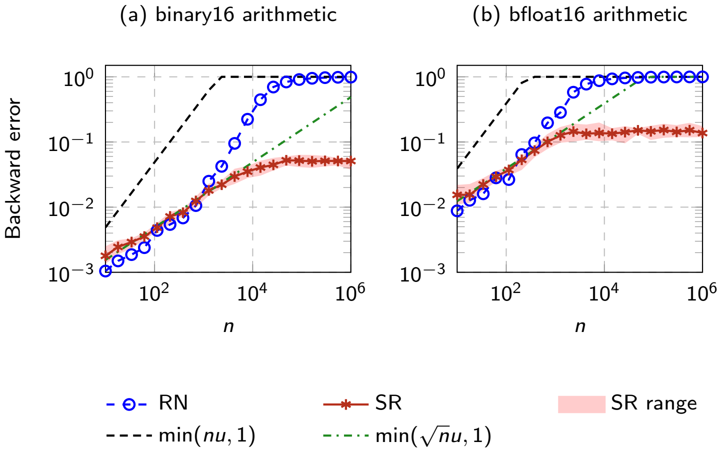
- Worst-case b-err bound with **RN**: $\frac{nu}{1-nu}$.
- Probabilistic bound with **RN**: $\lambda\sqrt{n}u + \mathcal{O}(u^2)$ w. p. $1 - 2e^{-\lambda^2/2}$. Requires an assumption that δ_n are mean independent zero-mean quantities—often satisfied [Connolly, Higham, Mary, 2021].

Wilkinson rule of thumb

$\sqrt{n}u$ error growth is a rule of thumb with **RN**, but always holds with **SR**.

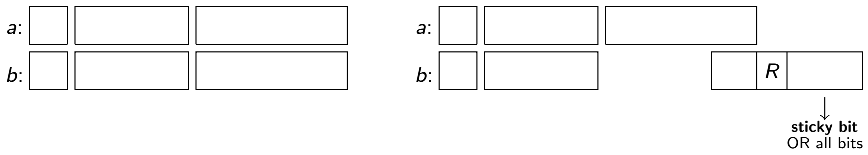
Example error growth with SR in mat-vec prod

Backward error in $y = Ax$ where $A \in \mathbb{R}^{100 \times n}$ with entries from uniform dist over $[0, 10^{-3}]$ and $x \in \mathbb{R}^n$ over $[0, 1]$.



How do we implement this? First, consider standard modes

Consider $a, b \in \mathbb{F}$ with $a, b > 0$ and $a > b$.



round-sticky	RD	RU	RN
00	D	D	D
01	D	U	D
10	D	U	D/U
11	D	U	U

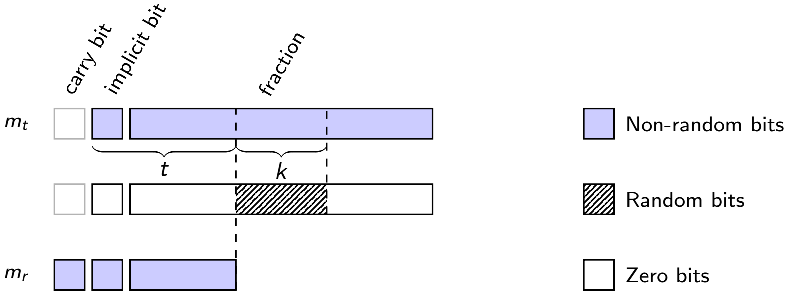
Guard bit

Guard bit is a complication that arises when we consider non-normalized floating-point significands, to compute the R bit correctly.

Implementation of SR

Take m_t to be a high precision unrounded significand from an operation.

Take t to be source precision and k the precision of random numbers.

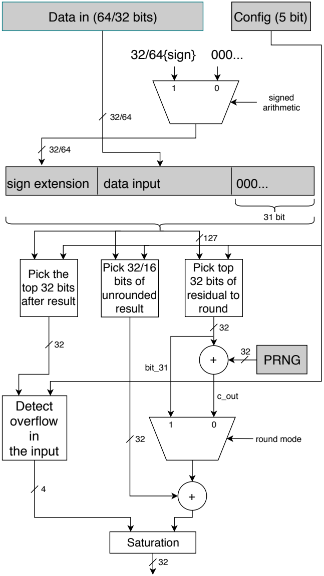


SR in hardware

Commercial hardware that implements SR is 100% for machine learning:

- **Graphcore IPU**
- **Intel Loihi**
- **Tesla Dojo**
- **Amazon Trainium**

Hybrid fixed/floating-point hardware implementation

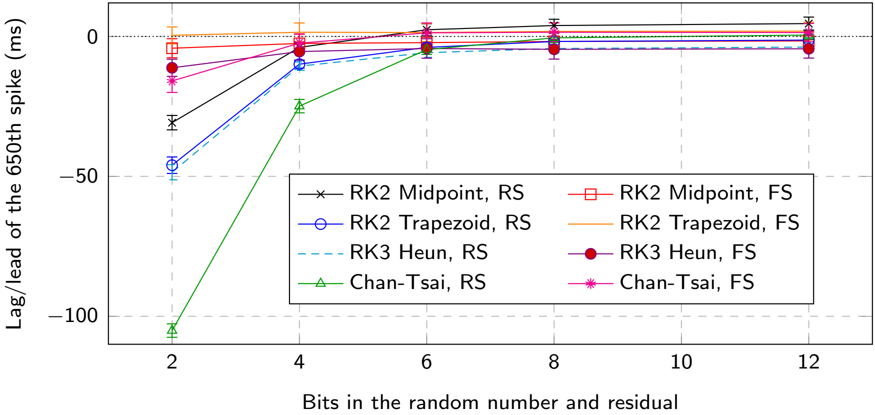


- Design and synthesis study available [Mikaitis, 2021].
- RN and SR in one.
- Programmable destination precision: round 1 to 32 bits.
- binary32 → bfloat16 rounding (16 bits).
- 32-bit uniform PRNG with 4 separate streams (seeds can come from TRNG).
- Accelerator integrated to each core in a 152-core chip.
- Operation: Write to a memory location, read back rounded.

Random number precision experiments

The question of k, precision of random numbers in SR, still open.

We did some experiments with ODE solvers in fixed-point arithmetics (Hopkins et al, 2020).

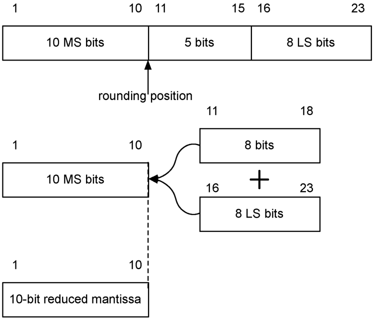


Patents from industry

There are numerous patents for SR from industry giants: NVIDIA, AMD, IBM. See our SR survey [Croci et al, 2021].

Here we focus on NVIDIA's ([NVIDIA, 2019]).

Below binary32 $\rightarrow$ binary16 example.



- Does not use PRNG.
- Take 8 bottom discarded bits and add to the top 8.
- Deterministic and cheaper to implement.
- Effect on numerical results not known.

Proposed IEEE 754 style properties

There is no standard way to implement SR.

We proposed a set of rules ([Croci et al, 2022]):

- If $x \in F$, $\mathrm{SR}(x) = x$.
- If x is in the range of F, round as though x is held in $p + k$ bits and rounded to p bits.
- **Overflows**: numbers between maximum value and $\pm\infty$: round as though exponent is not limited.
- When x is smaller than the smallest representable number, round stochastically to zero or that smallest number.
- If **subnormals** are disabled, round to zero or smallest normalized value.
- $\pm\infty$ and ± 0 should not be changed. NaNs should not be rounded.
- Exceptions signalled as standard.

Summary

Main takeaway

Implementations have been attempted, but key questions on random number generation remain. No official standard.

Open research questions about **SR**:

- Precision of random numbers.
- Implementation of **SR** alongside **RN** in hardware.
- How to switch between **SR** and **RN** at software level.

More details in the stochastic rounding survey paper

M. Croci, M. Fasi, N. J. Higham, T. Mary, and M. Mikaitis. *Stochastic rounding: implementation, error analysis and applications*. **R. Soc. Open Sci.**. Mar. 2022.
`https://bit.ly/3Kzw7mA`.

References I

N. J. Higham
Accuracy and Stability of Numerical Algorithms.
2nd ed. SIAM. 2002.

M. P. Connolly, N. J. Higham, T. Mary
Stochastic rounding and its probabilistic backward error analysis.
SIAM J. Sci. Comput. 43. 2021.

M. Mikaitis
Stochastic Rounding: Algorithms and Hardware Accelerator.
International Joint Conference on Neural Networks (IJCNN). 2021.

M. Hopkins, M. Mikaitis, D. R. Lester, S. Furber
Stochastic rounding and reduced-precision fixed-point arithmetic for solving neural ordinary differential equations.
Phil. Trans. R. Soc. 378. 2020.

References II

J. M. Alben, P. Micikevicius, H. Wu, M. Y. Siu.
Stochastic Rounding of Numerical Values.
2019. Patent Status: Active.