Research | November 01, 2019

Low Precision Floating-Point Formats: The Wild West of Computer Arithmetic

By Srikara Pranesh

Floating-point arithmetic is fundamental to scientific computing and lies at the heart of almost all numerical computations. The 1985 IEEE standard 754 for floating-point arithmetic marked the end of a turbulent period in scientific computing, during which vendors had their own implementations of floating-point arithmetic. All hardware vendors gradually adopted the IEEE standard of single and double precisions.

Until recently, the landscape of floating-point arithmetic—following employment of the IEEE standard—largely remained the same. However, hardware continually advanced to achieve higher performance, and numerical libraries evolved to efficiently use the hardware. Jack Dongarra and his colleagues [4] demonstrated this progression for singular value decomposition and deduced that *communication is far more expensive than computation*. Therefore, algorithms that minimize communication at the expense of increased computation are the norm; numerical libraries like PLASMA [5] are based on this philosophy. One thing that remained consistent during all of these developments was the floating-point formats, and this was about to change.

The 2008 revision of the IEEE 754 standard introduced half precision (or fp16) as a storage format. This was meant to reduce the cost of data movement, as it is cheaper to move 16 bits of data than 32 or 64 bits. However, once half precision was deemed sufficient for deep learning applications, researchers began using fp16 for computation, with a natural extension of the arithmetic rules. Half precision is now available on the NVIDIA P100 (2016) and V100 (2017) graphics processing units (GPUs), as well as the AMD Radeon Instinct MI25 GPU (2017). Although fp16 offers massive speedups, the maximum value it can represent is approximately 65,500, thus making overflow very likely. To address this issue, Google proposed an alternative half-precision format called the bfloat16. Properties of fp16 and bfloat16 are displayed in Table 1.

	u	$x_{\min}^s$	$x_{\min}$	$x_{\max}$
bfloat16	3.91×10^{-3}	9.18×10^{-41}	1.18×10^{-38}	3.39×10^{38}
fp16	4.88×10^{-4}	5.96×10^{-8}	6.10×10^{-5}	6.55×10^{4}

Table 1. Parameters for bfloat16, fp16 arithmetic, to three significant figures: unit roundoff u, smallest positive (subnormal) number $x_{\min}^s$, smallest normalized positive number $x_{\min}$, and largest finite number $x_{\max}$. Intel's bfloat16

specification does not support subnormal numbers.

The range of bfloat16—the format currently used in Google tensor processing units (TPUs)—is similar to single precision but has a lower precision than fp16. Intel will support bfloat16 in its upcoming Nervana Neural Network Processor and Cooper Lake processors. To further accelerate deep learning applications, an eight-bit floating-point format is also under consideration [12]. Additionally, researchers are contemplating nonstandard rounding modes—such as stochastic rounding—to enhance computational accuracy with these low-precision formats [9]. Another interesting technological innovation is the block-fused multiply-add unit, which can perform

$$C + A \times B, \quad A, B, C \in \mathbb{R}^{n \times n} \tag{1}$$

in a single clock cycle with one rounding error for some specific value of n. This feature is already available in the tensor cores of NVIDIA V100 (where $n = 4$). The Summit machine at Oak Ridge National Laboratory (ORNL), which leads the latest Top500 lists, comprises 27,000 V100s and has achieved an exaop performance using the tensor cores. Furthermore, 133 systems in the June 2019 Top 500 list employ accelerators, over 73 percent of which use GPUs that support fp16. Multiprecision computing units called matrix units (MXU), which operate on 128×128 matrices, are present in Google TPUs as well. However, Google TPUs are not commercially accessible, and details of MXU computation are not publicly available.

With regard to future machines, the Japanese Fugaku exascale machine will be based on the A64FX ARM processor with fp16 support. The Frontier exascale machine—to be installed at ORNL—will use AMD GPUs, which support fp16. In short, GPUs and low-precision formats are here to stay and have transformed the friendly neighbourhood of floating-point arithmetic into the Wild West. Development of algorithms that can exploit these new floating-point formats is therefore of great interest.

In the field of numerical linear algebra, Erin Carson and Nicholas Higham have proposed an algorithm for the solution of a linear system of equations that is given in double precision using fp16 [2]. They perform lower-upper (LU) factorization in fp16 and solve the update equation of iterative refinement via the generalized minimal residual method (GMRES), with the low-precision LU factors as preconditioners. A speedup of up to four over-highly-optimised libraries using the tensor cores of NVIDIA V100 has been demonstrated [6]. The algorithm achieved a performance of 445 petaflops—almost three times that of an optimised double-precision solver—when solving a dense linear system of 10 million equations at scale on the Summit machine.

Several matrices appearing in actual applications have entries that exceed the overflow limit of fp16. For example, many metals' modulus of elasticity is $\mathcal{O}(10^9)$. To address this issue, researchers have proposed a scaling algorithm with application to the solution of a linear system [8]. Even with the enormous computing power already available, it is still impossible to run very high-fidelity simulation models in climate studies, which can predict the extent of the effects of

global warming [10]. Therefore, scientists are contemplating multiprecision ideas to solve climate models of higher fidelity [3]. To enhance the speed of Monte Carlo simulations, researchers are considering representing the samples in low precision, with applications in Ising models [13] and finance [1].

Higham wrote about the challenges and potential benefits of multiprecision algorithms in a previous issue of *SIAM News* [7]. The two years since his article have seen further changes in the landscape of floating-point arithmetic because of architectural advancements like tensor cores. In 2005, Herb Sutter announced the advent of multicore architectures and proclaimed that "the free lunch is over" [11]. The onset of hardware that supports low precision marks the end of yet another free lunch, as new algorithms—rather than software optimisation—are the key to extracting benefits from such hardware.

References

[1] Belletti, F., King, D., Yang, K., Nelet, R., Shafi, Y., Chen, Y., & Anderson, J. (2019). Tensor processing units for financial Monte Carlo. Preprint, *arXiv:1906.02818*.

[2] Carson, E., & Higham, N.J. (2018). Accelerating the solution of linear systems by iterative refinement in three precisions. *SIAM J. Sci. Comput., 40*(2), A817-A847.

[3] Dawson, A., Düben, P.D., MacLeod, D.A., & Palmer, T.N. (2018). Reliable low precision simulations in land surface models. *Clim. Dyn., 51*(7), 2657-2666.

[4] Dongarra, J., Gates, M., Haidar, A., Kurzak, J., Luszczek, P., Tomov, S., & Yamazaki, I. (2018). The singular value decomposition: Anatomy of optimizing an algorithm for extreme scale. *SIAM Rev., 60*(4), 808-865.

[5] Dongarra, J., Gates, M,. Haidar, A., Kurzak, J., Luszczek, P., Wu, P.,..., Relton, S. (2019). PLASMA: Parallel linear algebra software for multicore using OpenMP. *ACM Trans. Math. Soft., 45*(2), 16-35.

[6] Haidar, A., Tomov, S., Dongarra, J., & Higham, N.J. (2018). Harnessing GPU tensor cores for fast FP16 arithmetic to speed up mixed-precision iterative refinement solvers. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage, and Analysis* (pp. 47:1-47:11). Dallas, TX: IEEE Press.

[7] Higham, N.J. (2017). A multiprecision world. *SIAM News, 50*(8), p. 2.

[8] Higham, N.J., Pranesh, S., & Zounon, M. (2018). Squeezing a matrix into half precision, with an application to solving linear systems. *SIAM J. Sci. Comp., 41*(4), A2536-A2551.

[9] Hopkins, M., Mikaitis, M., Lester, D.R., & Furber, S. (2019). Stochastic rounding and reduced-precision fixed-point arithmetic for solving neural ODEs. Preprint, *arXiv:1904.11263*.

[10] Palmer, T. (2014). Climate forecasting: Build high-resolution global climate models. *Nat. News, 515*(7527), 338.

[11] Sutter, H. (2005). The free lunch is over: A fundamental turn toward concurrency in software. *Dr. Dobb's Journ., 30*(3), 202-210.

[12] Wang, N., Choi, J., Brand, D., Chen, C.-Y., & Gopalakrishnan, K. (2018). Training deep neural networks with 8-bit floating point numbers. *Adv. Neur. Info. Process. Syst., 31*, 7686-7695.

[13] Yang, K., Chen, Y.-F., Roumpos, G., Colby, C., & Anderson, J. (2019). High performance Monte Carlo simulation of Ising model on TPU clusters. Preprint, *arXiv:1903.11714*.

Srikara Pranesh is a research associate in the Department of Mathematics at the University of Manchester. His research interests mainly include numerical analysis and numerical linear algebra.