2023 IEEE 23rd International Conference on Nanotechnology (IEEE-NANO 2023)
July 2-5, 2023 Jeju, Korea

Floating-Point Formats and Arithmetic for Highly Accurate Multi-Layer Perceptrons

Farzad Niknia, Ziheng Wang, Shanshan Liu, Pedro Reviriego, Ahmed Louri, and Fabrizio Lombardi

***Abstract*—The data precision can significantly affect the accuracy and overhead metrics of hardware accelerators for different applications such as artificial neural networks (ANNs). This paper evaluates the inference and training of multi-layer perceptrons (MLPs), in which initially IEEE standard floating-point (FP) precisions (half, single and double) are utilized separately and then compared with mixed-precision FP formats. The mixed-precision calculations are investigated for three critical propagation modules (activation functions, weight updates, and accumulation units). Compared with applying a simple low-precision format, the mixed-precision format prevents an accuracy loss and the occurrence of overflow/underflow in the MLPs while potentially incurring in less hardware overhead in terms of area/power. As the multiply-accumulation is the most dominant operation in trending ANNs, a fully pipelined hardware implementation for the fused multiply-add units is proposed for different IEEE FP formats to achieve a very high operating frequency.**

I. Introduction

Multi-layer perceptrons (MLPs) are a type of artificial neural networks (ANNs) consisting of interconnected neurons [1]. They have been extensively used in modeling nonlinear problems due to their simplicity, because often the implementation of an ANN can be complex for applications requiring a large volume of data [2]. To improve the training and then the accuracy of ANNs, the floating-point (FP) format has been widely used in hardware design; however, the utilized precision of an FP implementation is often application dependent as degradation and possible overflows/underflows may occur. The IEEE FP standard defines three formats (namely half, single, and double) that are widely used for hardware design; the larger the bit width of the FP format, the higher the range and better accuracy is expected in the calculations of the ANN. The selection of a data format can vary and is often based on different operational tradeoffs;

Recently, a scheme using different precision formats (i.e., mixed-precision) has been studied [3]. In this scheme, different FP formats are utilized in specific parts of the hardware, and it has been proven to achieve excellent performance in many machine learning (ML) tasks. Hardware implementations have also been proposed for mixed-precision operation [3]; however, such implementations have an operational frequency of at most 500 MHz while also incurring in a considerable power dissipation.

This paper analyzes and evaluates the performance of IEEE standard FP formats (half/single/double precision) in MLP training; since overflow may occur during training, networks with and without batch normalization are considered. Moreover, the different stages of calculation in MLP propagation have unique requirements on the data format; based on their features, mixed-precision FP formats are also discussed. The performance of these schemes is verified by simulation on five widely used ML classification datasets. Finally, a high performance fused multiply-add (FMA) unit is proposed to perform multiply-accumulation that is the most dominant operation in ANN computation.

The rest of the paper is organized as follows. Section II reviews MLPs, IEEE FP formats, and the use of FMA for implementing an MLP. Section III evaluates the performance of MLPs using different FP formats. A fully pipelined hardware design of FMA is presented in Section IV. Finally, the paper ends with the conclusion in Section V.

II. Preliminaries

A. Multi-layer perceptron (MLP)

The MLP is a fully connected feedforward ANN, that has been extensively utilized in modeling highly nonlinear functions; moreover, it also has a simple structure. These features make an MLP suitable for many tasks. In general, an MLP consists of an input layer, one or more hidden layers, and an output layer. The size of the valid sample features determines the number of neurons in the input layer, while the number of valid classes in the dataset determines the number of neurons in the output layer [1].

The interconnected layers can model a nonlinear transfer of an input vector to an output vector due to the use of nonlinear activation functions between the layers. This can solve classification problems of linearly inseparable data after backpropagation training [2]. Also, each neuron computes a non-linear function of the sum of products. Let n_k^{i+1} denote the value of neuron k in layer i+1, $w_{j,k}^{i+1}$ is the weight of neuron j in layer i mapping to neuron k in layer i+1, and m is the number of neurons in layer i. Also, b_k^{i+1} is

Farzad Niknia, Ziheng Wang, and Fabrizio Lombardi are with the Department of Electrical and Computer Engineering, Northeastern University, Boston, MA 02115, USA.

Shanshan Liu is the corresponding author and with the Klipsch School of Electrical and Computer Engineering, New Mexico State University, Las Cruces, NM 88001, USA, ssliu@nmsu.edu.

Pedro Reviriego is with the Departamento de Ingeniería de Sistemas Telemáticos, Universidad Politécnica de Madrid, Madrid 28040, Spain.

Ahmed Louri is with the Department of Electrical and Computer Engineering, the George Washington University, Washington DC 20052, USA.

2023 IEEE 23rd International Conference on Nanotechnology (NANO) | 979-8-3503-3346-6/23/$31.00 ©2023 IEEE | DOI: 10.1109/NANO58406.2023.10231201

the bias value related to n_k^{i+1} and Φ is the activation function (in this paper, ReLU is used as the activation function in the hidden layers and softmax is applied in the output layer). Then n_k^{i+1} is computed as:

$$n_k^{i+1} = \Phi\left(\sum_{j=1}^{m} w_{j,k}^{i+1} \cdot n_j^i + b_k^{i+1}\right) \quad (1)$$

Batch normalization is a technique for training ANNs when a mini-batch is applied [4]; it normalizes the input batch of any layer as:

$$y_i = \gamma \frac{x_i - \mu}{\sqrt{\sigma^2 + \epsilon}} + \beta \quad (2)$$

where μ and σ^2 are the mean and variance of the input batch. ϵ is a small positive value to avoid a zero denominator. γ and β are the so-called scaling and shift factors that can be determined by training. Batch normalization reduces the discrepancy in the distribution between the mini-batch and the entire dataset, i.e., dealing with the internal covariate shift. It also accelerates convergence and rescales the weights to avoid possible overflows in the training process.

B. IEEE 754 floating-point formats

Floating-point (FP) numbers based on the IEEE 754 standard are extensively utilized in modern processors. FP numbers have different precisions which makes them suitable for different applications due to their possible range. An FP number consists of 3 parts: sign (*S*), exponent (*E*), and mantissa (*M*). The value of an FP number is given in (3), where *H* is the hidden bit and is calculated by a logic OR of all bits in the biased exponent. *H* determines that a number is normal (subnormal) when it has a value of 1 (0) [5].

$$FP\ Value = (-1)^S \times 2^{(E-bias)} \times (H.M) \quad (3)$$

The number of bits for the different parts and the *Bias* value are specified for each precision format. This paper initially evaluates the three IEEE standard FP formats: double-precision (DP), single-precision (SP), and half-precision (HP).

C. Mixed-precision FP formats

To reduce computational complexity, low-precision formats (such as HP or SP) are extensively used in ANNs [6], [7]. mixed-precision FP formats have been proposed to retain the advantages of a small bit width, but also to address issues, such as accuracy loss. A mixed-precision format uses a high-precision (such as DP) in some critical arithmetic modules while keeping a low-precision in the remaining parts of the ANN [8]. There are several implementations of this combined FP format that apply high-precision arithmetic to different modules. This paper mainly investigates the following three cases:

- The high-precision arithmetic can be applied to activation functions to avoid possible overflows (if batch normalization is not applied). As per empirical observations, overflow occasionally occurs with a HP format when using activation functions including exponential units (such as sigmoid and softmax). The mixed-precision FP format can therefore prevent accuracy loss or failure in training.
- The high-precision arithmetic can be applied to the addition units to avoid the accumulation of rounding errors caused by precision limitations [9]. A mixed-precision FP format maintains high accuracy in critical arithmetic modules used for MLP training.
- The high-precision arithmetic can be applied in the gradient updates for weights to deal with underflow [10]. A very small learning rate is usually applied as a scaler, so underflows may either cause an accuracy loss, or stop the convergence process. Therefore, a mixed-precision FP format accomplishes more accurate parameter updates.

D. Fused multiply-add (FMA)

The principle of FP fused multiply-add (FMA) units was introduced in the IBM RS/6000 processors to increase performance and accuracy of FP calculations by making indivisible the multiply-accumulation operation [11]. This feature made FMA suitable to be utilized in general-purpose processors to calculate $A + (B \times C)$ (where *A*, *B*, *C* are numerical data). Although the main purpose of this unit is to calculate the accumulation of products, independent FP addition or multiplication can also be performed by setting $B = 1$ or $C = 1$ for the potential FP addition, and by setting $A = 0$ for the potential multiplication [12]. Different from conventional FP multiply-accumulate (MAC) units that use a separate FP multiplier and adder to implement the multiply-accumulation process, FMA units integrate these operations in a single design.

Generally, the advantages of FMA over MAC are as follows: 1) The FMA unit uses a single rounding at the end of the multiply-accumulation process, while the MAC unit uses two rounding units (so separate for the multiplier and the adder). This reduction in the number of rounding units decreases the overall delay and error. 2) The integration of arithmetic units in FMA leads to a reduction in the overall hardware overhead because some circuits are shared among addition and multiplication [11], [12].

III. MLP WITH DIFFERENT FP FORMATS

In this section, the evaluation of an MLP with both the IEEE standard and mixed-precision FP formats is pursued. A five-layer MLP is simulated in Matlab with 512 neurons in the hidden layers and 10 neurons in the output layer. Five commonly used ML datasets (MNIST [13], Fashion-MNIST [14], Cifar-10 [15], SVHN [16], and STL-10 [17]) are utilized in this paper.

A. IEEE Standard FP Formats

The performance of an MLP with different FP formats is assessed for classification accuracy. Two cases are initially evaluated, i.e., with or without batch normalization. The substitution method is applied when overflow occurs to continue the training process; in this method, the overflow value is replaced by the closest number within the applicable range [18]. The results are summarized in Table I (the cases in which overflow occurred, are labeled as "OV"). The results reveal the following features:

TABLE I
INFERENCE ACCURACY WITH DIFFERENT FP FORMATS

Dataset		DP	SP	HP
With norm.	MNIST	98.48%	98.47%	98.43%
	Fashion MNIST	91.22%	91.19%	90.81%
	CIFAR-10	81.34%	81.27%	80.52%
	SVHN	86.38%	86.37%	86.25%
	STL-10	69.74%	69.71%	68.87%
Without norm.	MNIST	98.06%	97.84%	97.01%
	Fashion MNIST	90.76%	90.63%	88.87%(OV)
	CIFAR-10	81.07%	81.01%	78.98%(OV)
	SVHN	86.11%	85.96%	85.33%
	STL-10	69.15%	68.84%	66.39%(OV)

1) For the case in which no overflow occurs, the FP format with a larger number of bits in the representation achieves the highest precision, but it only leads to a rather marginal improvement; for example, the SP format shows almost the same performance compared with the DP format. The HP format has a slightly lower accuracy compared to the SP format.

2) If batch normalization is applied (and usually this is a computationally intensive process executed externally to the ANN), no overflow occurs; if batch normalization is not applied, overflow is observed when using the HP format.

These results show the feasibility of low-precision FP formats in MLP training. The SP format achieves a very close classification accuracy to the DP format, while the HP format sacrifices accuracy. Moreover, overflow is shown as a notable problem of low-precision FP formats, especially when batch normalization is not applied. Even with substitution, the overflowed values update the gradients incorrectly and cause a large accuracy loss; in many applications, it may cause more serious consequences such as divergence or exploding gradients and thus, result in an invalid model or system failure.

B. Mixed-precision FP Formats

The performance of mixed-precision FP formats has been also assessed; the mixed-precision FP formats apply a high-precision format (DP) in several critical computational modules to achieve a high accuracy and then, use low-precision formats (SP/HP) in the remaining parts. Three cases of mixed-precision FP formats introduced in Section II.C are evaluated.

An advantage of mixed-precision FP formats is that they can avoid possible overflow problems. The simulation results of this paper show that no overflow occurs with normalization after the mixed-precision FP format is applied to the activation functions including the exponential operations. Table II compares the previous overflow cases (dataset "Fashion MNIST", "CIFAR-10", and "STL-10" without batch normalization) with the HP format and the results with mixed-precision FP format (evaluated by the inference accuracy and the average number of overflows in each iteration). According to our simulation, the mixed-precision FP format (specified high-precision format on the activation function) can avoid overflows that occur when using the HP format; therefore, when considering the computationally intensive process of batch normalization, the mixed-precision FP format is a feasible alternative to solve potential overflow problems.

TABLE II
COMPARISON OF OVERFLOWED CASES WITH HP FORMAT AND RESULT WITH MIXED-PRECISION FP FORMAT FOR ACTIVATION FUNCTION

Dataset	HP		Mixed-precision (HP + DP for activation function)	
	# overflows	Inference accuracy	# overflows	Inference accuracy
Fashion MNIST	23	88.87%	0	90.14%
CIFAR-10	16	78.98%	0	80.67%
STL-10	41	66.09%	0	68.26%

Next, the other two mixed-precision format schemes that are performed for the weight-update and addition units respectively, are assessed. As discussed previously, the use of a mixed-precision FP format during weight updates can overcome possible underflows when considering small learning rates as scalers. One of the objectives when using a mixed-precision format during additions is to reduce the accumulation of rounding errors. The results for these two scenarios are summarized in Table III. For all datasets, the mixed-precision format during the weight update process only brings a slightly higher accuracy, while the mixed-precision format during accumulations leads to more pronounced improvements. These results suggest that rounding errors of low-precision FP formats are the main reason for an accuracy loss; thus, by applying a high-precision format only in critical accumulation modules yields very good results. Such a scheme achieves a performance very similar to the DP format in terms of accuracy.

IV. FMA HARDWARE DESIGN

In this section, a novel fully pipelined FMA unit design used to perform the multiply-accumulation operations in an MLP is proposed; it supports different IEEE standard FP formats. Fig. 1 shows the proposed design (note that both addition and multiplication have the same precision). Assume the input numbers have a width of p bits (as based

TABLE III
INFERENCE ACCURACY WHEN APPLYING MULTI-PRECISION FORMAT FOR WEIGHT UPDATE AND ACCUMULATION

Dataset	DP	SP	HP	Multi-precision (SP + DP for weight update)	Multi-precision (HP + DP for weight update)	Multi-precision (SP + DP for addition)	Multi-precision (HP + DP for addition)
MNIST	98.48%	98.45%	98.04%	98.46%	98.11%	98.48%	98.42%
Fashion-MNIST	91.22%	91.19%	90.81%	91.19%	90.88%	91.21%	91.14%
CIFAR-10	81.34%	81.27%	80.52%	81.29%	80.64%	81.31%	81.28%
SVHN	86.38%	86.37%	86.25%	86.37%	86.29%	86.38%	86.35%
STL-10	69.74%	69.71%	68.87%	69.72%	68.98%	69.74%	69.72%

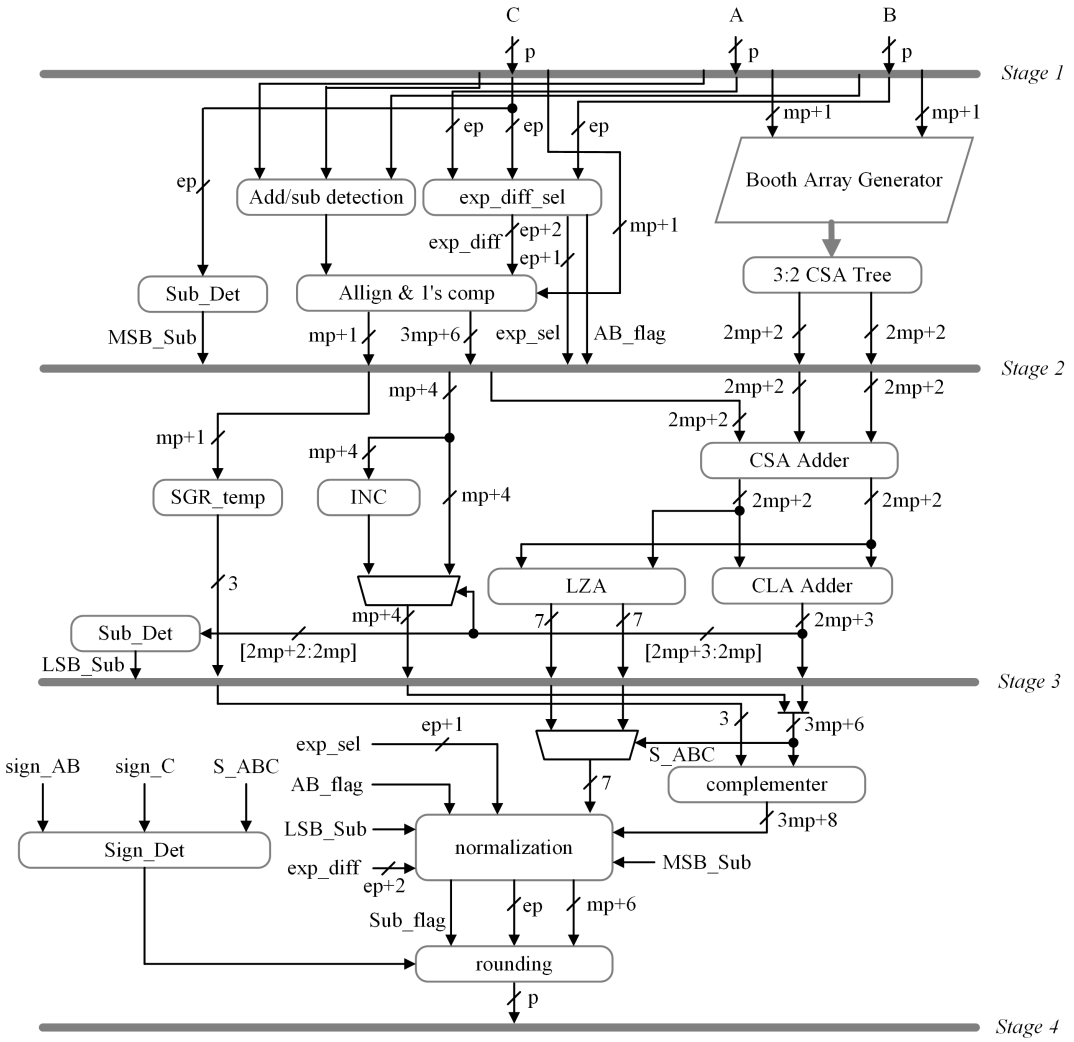

Fig. 1. Proposed FMA unit for IEEE standard precision format.

on the FP precision format), *ep* and *mp* denote the numbers of bits for the exponent and the mantissa of the FP format.

The proposed FMA design consists of 4 stages. The first stage extracts the different sectional parts of the input data including sign, exponent, mantissa and calculates the hidden bit for each number.

The second stage calculates the multiplication of *H.M* (*P*) of the two input numbers *A* and *B* using a radix-4 Booth array generator [19]. The result is in a carry-save format (i.e., in the carry-save adder, CSA, in Fig. 1). At the same time, the add/sub detection unit compares the signs of ($A \times B$) and *C* to check whether the required final operation is addition or subtraction, i.e., it determines whether *C* needs to be complemented during the alignment. For simplicity in the design of the alignment unit, the *H.M* part of *C* is located to the left of the most significant bit (MSB) of the multiplication result. Two additional bits (including guard and round bits) are also considered to account for the case when $H_c.M_c$ does not need to be shifted. Hence, the shift operation results in $d = (((e_A + e_B) - e_C) + mp + 4)$ where $mp + 4$ is the alignment offset due to placing P_c to the left of the multiplication result and added to the 3 bits including guard, round and MSB of the multiplication bits. Based on the number of shifts, the exp_diff_sel unit chooses the exponent (exp_sel) for the next processing step and a flag which determines the selection of the exponent (AB_flag). So, the largest exponent is set as the final exponent if it is 1; then the exponent of the multiplication is chosen, else the exponent of *C* is chosen. Also, the sub_det unit determines if *C* is a subnormal number, because the input numbers are not normalized during the input processing and based on them, the multiplication result can be subnormal too.

The third stage adds the carry-save format of the Booth array generator with the *2mp+2* least significant bits (LSBs) of the aligned P_c in carry-save format. Then, the result is again in a carry-save format . The use of the LSB of the aligned bit reduces the hardware required for the design of the leading zero anticipators (LZA) unit; the LZA unit estimates the number of leading zero bits prior to the conversion of the carry-save format to a conventional binary format (as performed by the carry-lookahead adder (CLA) unit). In this paper, the LZA design of [13] is utilized. As the sum (i.e., the result of the CLA) can be positive or negative, then the LZA reports both the leading zero and leading one. For *k*-bit operands *A* and *B*, the indicator (*f*) is generated to

show the location of first one or zero. To generation of f is given as follows where $T=A\oplus B$, $G=AB$, $Z=\overline{A}\overline{B}$:

$$\begin{aligned} f_{k-1} &= \overline{T}_{k-1}T_{k-2} \\ f_i &= T_{i+1}\left(G_i\overline{Z}_{i-1} + Z_i\overline{G}_{i-1}\right) \\ &\quad +\overline{T}_{i+1}\left(Z_i\overline{Z}_{i-1} + G_i\overline{G}_{i-1}\right),\ i < k\text{-}1 \end{aligned} \tag{4}$$

The MSB of the result of the CLA unit determines if the mp+4 MSBs of the aligned result should be incremented; moreover, the SGR_temp unit calculates the temporary SGR value, because it may be required when d is greater than 3mp+5.

In the fourth stage, the MSB of the result of the CLA determines if a complementing operation is needed. Also, this MSB selects between the number of leading (0 or 1) bits from the LZA unit. The normalization unit normalizes the result of the complementor based on the result of the LZA; it determines whether P_c is shifted more than the aligning offset. If so, the normalization is performed based on the result of the LZA; LSB_Sub determines if the result is subnormal, otherwise MSB_sub determines if the number is subnormal, and the normalization is equal to exp_diff. During normalization, the exponent is updated too. As mentioned previously, normalization using the LZA unit could be 1 bit off; then, the rounding unit is responsible to address this probable error by shifting the result by one more bit to the left. For rounding to the near/even, this is accomplished based on the processing detailed in [5].

The proposed design is implemented using Verilog HDL and then synthesized with Cadence Genus Synthesis tool using a 32nm library (at 25°C and TT corner). As per the synthesized hardware metrics reported in Table IV, the proposed FMA design can achieve a very high operating frequency due to the fully pipelining scheme.

V. Conclusion

This paper has investigated different floating-point (FP) formats (double, single, half, and mixed precisions) applied to the training and inference of multi-layer perceptrons (MLPs). The simulation results have shown that that these FP formats have advantages for different application scenarios: i) The double-precision format is a safe choice with the best classification performance, while it requires a very large hardware. ii) The single-precision format is appropriate in most cases if normalization is applied; it achieves a slightly lower accuracy than the double-precision format but at a significantly less hardware. iii) The half-precision format requires the least hardware, but the accuracy is reduced. iv) The mixed-precision format is a feasible alternative to double-prevision format because it prevents an accuracy loss in most cases (by avoiding the overflow/underflow issues), while it potentially incurs in less hardware overhead. A fully pipelined fused multiply-add (FMA) unit design that supports all these formats has been proposed to perform the multiply-accumulation operations for the network. The hardware design of an FMA utilizing mixed-precision format and the assessment of its viability in ANN implementation are left for future work.

TABLE IV
Synthesis Results for the Proposed FMA Unit with Different FP Precisions

Precision	Area (μm^2)	Power (mw)	Frequency (MHz)
HP	5549.23	4.48	934.58
SP	13894.56	9.63	800.00
DP	43340.96	24.23	666.66

References

[1] Rosenblatt, F. (1961). *Principles of neurodynamics. perceptrons and the theory of brain mechanisms* (No. VG-1196-G-8). Cornell Aeronautical Lab Inc Buffalo NY.

[2] M. W. Gardner, and S. R. Dorling. "Artificial neural networks (the multilayer perceptron)—a review of applications in the atmospheric sciences." *Atmospheric environment* vol. 32, no. 14-15, pp. 2627-2636, Aug. 1998.

[3] H. Zhang, D. Chen and S. -B. Ko, "Efficient Multiple-Precision Floating-Point Fused Multiply-Add with Multi-precision-Precision Support," in *IEEE Transactions on Computers*, vol. 68, no. 7, pp. 1035-1048, July 2019.

[4] Ioffe, S., & Szegedy, C. (2015, June). Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning* (pp. 448-456). PMLR.

[5] IS Committee. (2008). 754–2008 IEEE standard for floating-point arithmetic. *IEEE Computer Society Std*, *2008*.

[6] Courbariaux, M., Bengio, Y., & David, J. P. (2014). Training deep neural networks with low precision multiplications. *arXiv preprint arXiv:1412.7024*.

[7] Venkatesh, G., Nurvitadhi, E., & Marr, D. (2017, March). Accelerating deep convolutional networks using low-precision and sparsity. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 2861-2865). IEEE.

[8] Das, D., Mellempudi, N., Mudigere, D., Kalamkar, D., Avancha, S., Banerjee, K., ... & Pirogov, V. (2018). Mixed precision training of convolutional neural networks using integer operations. *arXiv preprint arXiv:1802.00930*.

[9] Zhang, H., Chen, D., & Ko, S. B. (2019). Efficient multiple-precision floating-point fused multiply-add with mixed-precision support. *IEEE Transactions on Computers*, *68*(7), 1035-1048.

[10] Micikevicius, P., Narang, S., Alben, J., Diamos, G., Elsen, E., Garcia, D., ... & Wu, H. (2018, February). Mixed Precision Training. In *International Conference on Learning Representations*.

[11] R. K. Montoye, E. Hokenek, and S. L. Runyon, "Design of the IBM RISC System/6000 floating-point execution unit," IBM J. Res.Develop., vol. 34, no. 1, pp. 59–70, Jan. 1990.

[12] J. D. Bruguera and T. Lang, "Floating-point fused multiply-add: reduced latency for floating-point addition," *17th IEEE Symposium on Computer Arithmetic (ARITH'05)*, Cape Cod, MA, USA, 2005, pp. 42-51.

[13] LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, *86*(11), 2278-2324.

[14] Xiao, H., Rasul, K., & Vollgraf, R. (2017). Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*.

[15] Krizhevsky, A., & Hinton, G. (2009). Learning multiple layers of features from tiny images.

[16] Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., & Ng, A. Y. (2011). Reading digits in natural images with unsupervised feature learning.

[17] Coates, A., Ng, A., & Lee, H. (2011, June). An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics* (pp. 215-223). JMLR Workshop and Conference Proceedings.

[18] Hauser, J. R. (1996). Handling floating-point exceptions in numeric programs. *ACM Transactions on Programming Languages and Systems (TOPLAS)*, *18*(2), 139-174.

[19] Bewick, G.W., 1994. *Fast multiplication: Algorithms and implementation*. Stanford University.