25th IMEKO TC4 International Symposium
23rd International Workshop on ADC and DAC Modelling and Testing
IMEKO TC-4 2020
Brescia, Italy / September 12-14, 2022

Floating-Point Roundoff Error Analysis in Artificial Neural Networks

Hussein Al-Rikabi and Balázs Renczes

Department of Measurement and Information Systems
Budapest University of Technology and Economics
Budapest, Hungary

***Abstract* – In this paper, roundoff errors in Artificial Neural Networks (ANNs) are analyzed on a model for Solid-State Power Amplifiers (SSPAs). Calculations are carried out on 32-bit Floating-Point (FP32) arithmetics, and results are verified using 64-bit floating-point representation as reference. Besides the modeling of quantization noise at every operation, error propagation is also taken into consideration when calculating the cumulative Quantization Noise Power (QNP) after each stage and at the final output. By this means, the predictability of roundoff errors in the ANN is demonstrated. Consequently, it can be determined whether the FP32 arithmetic is sufficient instead of applying the computationally more demanding 64-bit calculations.**

***Keywords* – Artificial Neural Networks, Quantization Noise, Floating-Point Number Representation, Solid-State Power Amplifier.**

I. INTRODUCTION

Over the last decades, the evolution of deep learning algorithms has been noticeable, as they have been used in various applications, starting with video processing, voice enhancement, and generally in digital signal processing [1]. Recently, improving the performance of Artificial Neural Networks (ANNs) has been an essential scope of research. The main parameters to investigate are size, speed, performance, and power consumption of the ANN architectures as they have been used extensively in current applications [2]. Using a finite number of bits to represent the data and the coefficients is called quantization. The quantization of ANNs has been an important scope of study recently as this technique is widely used in large models and systems such as artificial intelligence chips. The quantization process of weights, biases, and operations in ANNs significantly reduces the storage size of the system. Furthermore, the reduction in the number of bits can also substantially accelerate the ANNs [1]. In this paper, a nonlinear model of Solid-State Power Amplifiers (SSPAs) is used as an example for the proposed analysis to predict its behavior using ANNs. SSPAs have been used in a variety of applications recently, especially in mobile communication systems, due to their small size and low phase distortion [3].

There have been similar studies in the literature that are summarized in the following part. Nichols et al. [4] studied the feasibility of using the 32-bit Floating-Point (FP32) format for ANNs on Field Programmable Gate Arrays (FPGAs), and they compared this representation with the 16-bit fixed-point representation. They showed that the FP32 arithmetic makes the system thirteen times bigger in size than the 16-bit fixed-point format; hence it would be area consuming for the hardware implementations. Lian et al. [5] presented an implementation of a Convolutional neural network (CNN) accelerator on an FPGA device with Block Floating-Point (BFP) arithmetic. They mixed the 16-bit Floating-Point (FP16) and the FP32 formats in their architecture. They showed that the BFP can efficiently reduce the size, signal traffic, and hence energy as this method provided all these merits with only 0.12% accuracy loss. Peric et al. [6] have made a comparison between 32-bit fixed-point and FP32 representations in terms of quantization noise and signal to noise ratio. They concluded that the accuracy of the floating-point quantizer is the same as that of the fixed-point quantizer.

In this study, our contribution is to investigate the effect of FP32 quantization on ANNs theoretically. This means that we do not compare experimental results, but the QNP is predicted in advance without launching the network structure. For the investigation, an example of an SSPA model is used in this paper. The analysis includes a comparison between theoretical and simulated results. Simulations were carried out using MATLAB 2021b. The remainder of this paper is organized as follows. Fundamental concepts about the investigated structures are presented in Section II, including ANNs and SSPAs. Section III gives a background on the IEEE floating-point standard and the quantization noise. Simulation results and the discussion are presented in Section IV, while Section V concludes the paper.

II. INVESTIGATED STRUCTURES

This section presents a theoretical overview of the ANNs and the SSPAs.

A. *Artificial Neural Networks*

ANNs have been used for many applications recently. It is important to implement them with a small size, complexity, and with low energy while maintaining acceptable performance. ANNs are effective tools for tackling complex nonlinear problems and identifying universal input-output mappings. An ANN consists of inputs, outputs, and hidden layers. Each link between two neurons, or between an input and a neuron, has its own weight. Furthermore, each neuron has its own bias. Weights are considered through multiplication, while biases are added to the sum of input-weight products [7]. An activation function is applied to the summation result in each neuron. Nonlinear activation functions are essential in artificial neural networks because they allow them to learn complex mappings between their inputs and outputs [8].

The accuracy of the ANN is determined by the weights and biases established throughout the training phase. Fig. 1 depicts a simplified ANN architecture [7].

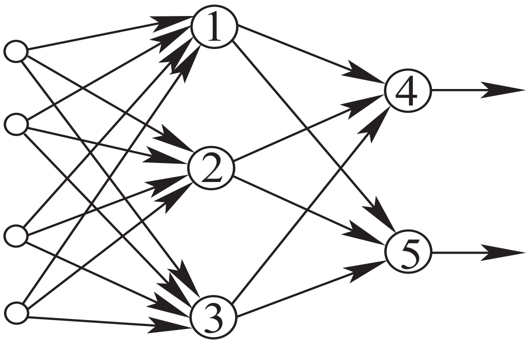

Fig. 1. A basic ANN architecture [7].

B. *Solid-State Power Amplifier Model*

High Power Amplifiers (HPAs) are utilized in mobile communications and base stations to transmit suitable power levels. The performance of Orthogonal Frequency-Division Multiplexing (OFDM) systems is greatly influenced by HPA nonlinearities [9]. Traveling-Wave Tube Amplifiers (TWTAs) and SSPAs are the two most prevalent types of HPAs. In this study, SSPA is preferred since the phase distortion of typical SSPAs is significantly smaller than that of TWTAs. The SSPA model can be described as follows [9]:

$$F_A(x) = \frac{x}{\left[1 + (x/A_o)^{2p}\right]^{-2p}}, \tag{1}$$

where $F_A(x)$ is the output of the SSPA, x is the SSPA input, p is the smoothness from the linear region to the saturation region and A_o is the output maximum saturation level.

III. BACKGROUND OF THE ERROR ANALYSIS

A basic overview of floating-point number representation systems, and modeling of quantization noise in these systems will be presented in this section.

A. *IEEE floating-point Standard*

Computer manufacturers utilized incompatible floating-point representations for a long time. The output of one computer could not be directly translated by another. The IEEE 754 Floating-Point Standard, which defines floating-point numbers, was created in 1985 by the Institute of Electrical and Electronics Engineers (IEEE). This is the floating-point format that is described in this section because it is now nearly universally used [10].

The IEEE 754 Standard defines the FP32 representation [11]. It uses one bit for the sign s, 8 bits for the exponent e, and 23 bits for the mantissa m. The first bit of the mantissa (to the left of the binary point) is always 1 in floating-point and hence does not need to be stored. It's referred to as the implied leading one. Round down, round up, round toward zero, and round to nearest are the four rounding modes. Round to nearest is the default rounding mode [10].

This format is also called full-precision, contrary to the FP16 format, which is referred to as half-precision. The FP32 representation is also known as single precision compared to the double precision, which is the 64-bit Floating-Point (FP64) format. The bitmap of the FP32 representation is illustrated in Fig. 2.

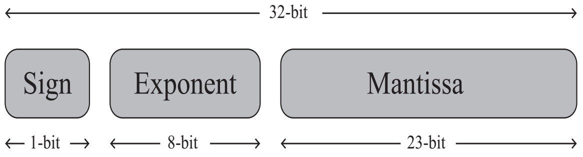

Fig. 2. Bitmap of the FP32 representation.

Based on the binary number system, the FP32 format represents a real number x as follows [6]:

$$x = (-1)^s \cdot 2^e \cdot \left(1 + \frac{m}{2^{23}}\right). \tag{2}$$

In this case, the exponent e and the mantissa m can have values in the following ranges:

$$e \in \{-128, -127, \ldots, -1, 0, 1, \ldots, 126, 127\}, \tag{3}$$

$$m \in \{0, \ldots, 2^{23} - 1\}. \tag{4}$$

Similar to the fixed-point 32-bit representation, the single precision, which is the FP32 representation gives discrete values. However, the difference is that the FP32 representation has variable step size. A smaller step size makes the FP32 representation very efficient to represent small values, while a large step size makes the presentable range wide [6].

B. Floating-Point Quantization Noise

Scientists mostly overlook roundoff errors in computations because of the success of the IEEE double precision standard. We usually expect that a simple personal computer's precision is infinite. However, roundoff errors can readily ruin a calculation's outcome, even if it appears fair. As a result, even with IEEE double precision representation, it is worthwhile to investigate them.

Using floating-point numbers to represent physical quantities enables a vast dynamic range to be covered with a small number of digits. Roundoff errors are usually proportional to the amplitude of the depicted quantity when using this type of representation. In most applications, floating-point representation is better than fixed-point representation from a point of numerical accuracy. The adoption of floating-point numbers is speeding up as the speed of floating-point calculations improves and the cost of implementation decreases. As a result, having a modeling approach for the propagation of Quantization Noise Power (QNP) is needed to simulate or predict the accuracy of the applied implementations.

The difference between the quantizer output (x') and input (x) is the roundoff error of the floating-point quantizer v_{FL}, which is represented by the following equation [12]:

$$v_{\mathrm{FL}} = x' - x. \tag{5}$$

An approximation has been derived in [12] to calculate the QNP, and it is demonstrated by the following equation:

$$E\{v_{\mathrm{FL}}^2\} = 0.18 \cdot 2^{-2p} \cdot E\{x^2\}, \tag{6}$$

where $E\{x^2\}$ is the expected value of the quantizer input power, and p is the number of mantissa bits plus the hidden bit, which in total is 24 bits for FP32 format. When the mantissa is 16 bits or more, the QNP is given by (6); otherwise, the following theoretical bounds present it [12]:

$$\frac{1}{12} \cdot 2^{-2p} \cdot E\{x^2\} \leq E\{v_{\mathrm{FL}}^2\} \leq \frac{1}{3} \cdot 2^{-2p} \cdot E\{x^2\} \tag{7}$$

IV. SIMULATION RESULTS AND DISCUSSION

A small ANN was trained and built in simulation that has one input, one output, and one hidden layer with three neurons. Hyperbolic Tangent Sigmoid (TanSig) activation function has been used in the proposed analysis. The SSPA model of (1) has been built using an ANN for prediction application. The practical range and values for the SSPA considered in this study are [0,1] for x, 0.8 for p, and 1 for A_o.

An FP32 quantizer was used for each operation and for the input of the ANN. The double precision floating-point format was assumed to be the reference to our calculations and simulations. The quantization of the weights and the biases is not included in this study as all of them have been set to FP32 representation. By this means, we can obtain the actual roundoff error during each operation. Coefficient quantization is a subject of further investigation.

Table 1. Propagation of error formulas.

Function	Variance
$F = A + constant$	$\sigma_F^2 = \sigma_A^2$
$F = A \cdot constant$	$\sigma_F^2 = constant^2 \cdot \sigma_A^2$
$F = e^A$	$\sigma_F^2 = F^2 \cdot \sigma_A^2$
$F = \frac{1}{A}$	$\sigma_F^2 = F^2 \cdot (\sigma_A^2/A^2)$

Table 2. QNP after each quantizer taking the error propagated from previous operations into consideration.

No.	Designator	Theoretical	Simulation
1	Input	2.13E-16	1.69E-16
2	m1	5.21E-16	4.72E-16
3	m2	2.42E-16	2.21E-16
4	m3	3.84E-15	3.54E-15
5	s1	6.00E-16	6.41E-16
6	s2	9.03E-16	1.20E-15
7	s3	1.22E-14	1.71E-14
8	Ag1	2.40E-15	2.56E-15
9	Ag2	3.61e-15	4.82E-15
10	Ag3	4.87e-14	6.85E-14
11	Ae1	1.13E-14	2.30E-14
12	Ae2	1.14E-16	8.82E-17
13	Ae3	2.03E-06	5.89E-06
14	As1	1.64E-14	2.98E-14
15	As2	9.63E-16	1.33E-15
16	As3	2.05E-06	5.89E-06
17	Ar1	5.61E-16	4.39E-16
18	Ar2	1.04E-15	1.06E-15
19	Ar3	1.36E-18	5.77E-19
20	Agg1	2.24E-15	1.76E-15
21	Agg2	4.16E-15	4.25E-15
22	Agg3	5.42E-18	2.31E-18
23	Af1	2.31E-15	1.82E-15
24	Af2	4.52E-15	4.25E-15
25	Af3	6.37E-16	2.97E-16
26	mm1	5.83E-17	4.64E-17
27	mm2	3.11E-14	2.97E-14
28	mm3	9.31E-16	5.14E-16
29	ss	3.27E-14	3.12E-14
30	y	3.29E-14	3.12E-14

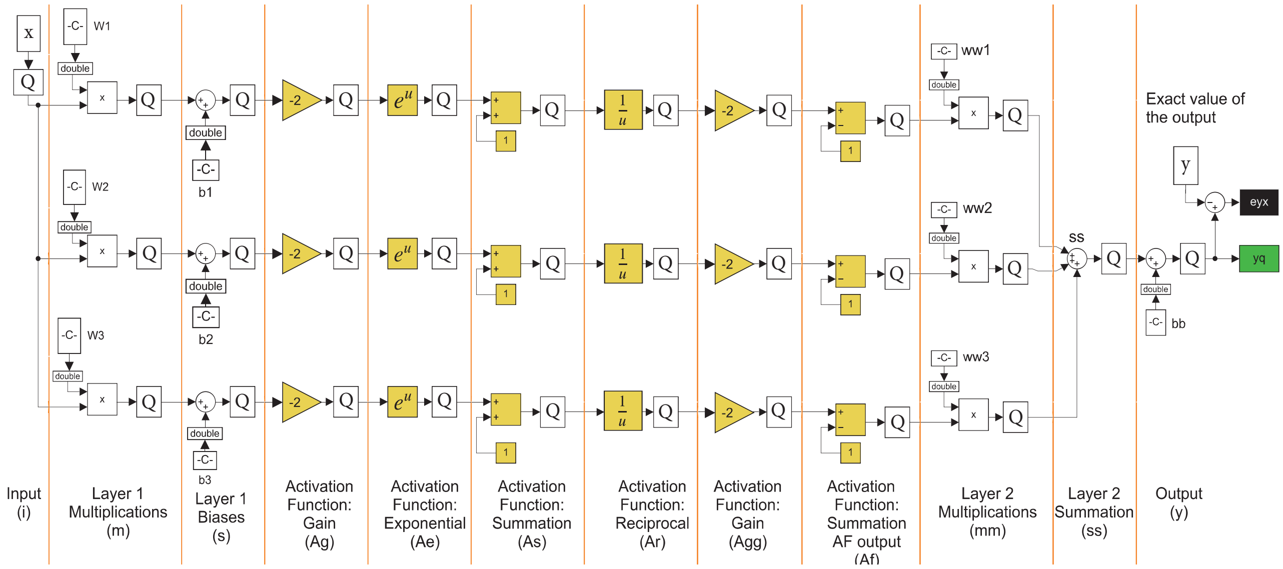

Fig. 3. Block diagram of the SSPA ANN model including the quantizers.

For statistical calculations, 10000 samples were injected into the ANN that spread in the range [0,1]. Fig. 3 shows the ANN architecture including the quantizers denoted by the letter Q that are placed after each operation.

TanSig activation function have been used for each neuron in the ANN. The QNP was calculated for each quantizer, using (6), and added to the propagated QNP from previous operations. The variances, which apply for the QNP, of the error transformation through a function are presented in Table 1. Table 2 shows the cumulative QNP after each operation.

The QNP was also calculated in the simulation by finding the difference between the input and the output of each quantizer. The QNP propagates through stages, and it can increase or decrease according to the operation. If the operation is, for example, a division or a reciprocal, the QNP will decrease while it increases if the operation is a summation or a multiplication operation. The same designator abbreviations were used in Table 2 and in Fig. 3.

For the inserted inputs, the QNP of each quantizer has been investigated considering the bounds of (7). The investigation showed that every QNP value was in the theoretically given range. However, due to the limits of this paper, these results are not presented in detail. The quantizer stages Ag and Agg have zero QNP because the input values are already quantized and a multiplication by two of a quantized number introduces no extra quantization error by this operation quantizer. But the increase that can be seen from Table 2 is caused by a multiplication of the QNP from previous stages by a constant according to the second function in Table 1.

The total QNP values of the model have been calculated theoretically and in simulation. They are 3.29E-14 and 3.12E-14, respectively. The theoretical QNP at the output of the ANN using the FP64 format is 1.14E-31. This small value indicates that the FP64 format is a valid choice to be considered as a reference. Since the theoretical and actual values are really close to each other, the acquired results show the effectiveness of the presented method. Furthermore, it can be seen that these errors are sufficiently small so that the FP32 arithmetics can be used instead of an FP64 arithmetics. By this means, the cost and size of implementation can be decreased significantly, as it was highlighted in Section I.

V. CONCLUSION

Analysis of the 32-bit Floating-Point (FP32) quantization in an Artificial Neural Network (ANN) has been presented in this study taking the double precision floating-point representation as a reference. The ANN have been trained to model a Solid-State Power Amplifier (SSPA) as a practical example. The Quantization Noise Power (QNP) after each operation was calculated theoretically and in simulation.

It can also be concluded that when the QNP propagates through the different stages of the ANN, it can increase or decrease. We encountered two cases when it increased: the exponential and the multiplication operations. It can also decrease when multiplying by a number smaller than one or when it is a reciprocal operation.

Results show that the presented method is effective in giving an estimation of the error that would be generated when using the FP32 quantization. The total error calculated for the presented example is small compared to the data values. Therefore, using half the number of bits, the FP32 representation, for such small ANNs is a valid choice.

VI. ACKNOWLEDGMENT

This research was funded by the National Research, Development, and Innovation Fund of Hungary under Grant TKP2021-EGA-02.

REFERENCES

[1] M. Kiyama, Y. Nakahara, M. Amagasaki, M. Iida, “A quantized neural network library for proper implementation of hardware emulation”, 2019 Seventh International Symposium on Computing and Networking Workshops (CANDARW), 2019, pp.136-140.

[2] A. Gholami, S. Kim, Z. Dong, Z. Yao, M. W. Mahoney, K. Keutzer,“A survey of quantization methods for efficient neural network inference”, arXiv preprint arXiv:2103.13630, 2021.

[3] F. Gregorio, T. Laakso, “The performance of OFDM-SDMA systems with power amplifier non-linearities”, Proceeding of the 2005 Finnish Signal Processing Symposium-FINSIG, vol.5, 2005.

[4] K. R. Nichols, M. A. Moussa, S. M. Areibi, “Feasibility of floating-point arithmetic in FPGA based artificial neural networks”, In CAINE Conference, 2002.

[5] X. Lian, Z. Liu, Z. Song, J. Dai, and W. Zhou, X. Ji, “High-performance FPGA-based CNN accelerator with block-floating-point arithmetic”, IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol.27, No.8, 2019, pp.1874-1885.

[6] Z. Peric, M. Savic, M. Dincic, N. Vucic, D. Djosic, S. Milosavljevic, “Floating Point and Fixed Point 32-bits Quantizers for Quantization of Weights of Neural Networks”, IEEE, 2021 12th International Symposium on Advanced Topics in Electrical Engineering (ATEE), 2021, pp.1-4.

[7] K. Du, M. Swamy, “Neural Networks and Statistical Learning”, 2nd edition, Springer, London, 2019.

[8] I.A.D. Williamson, T. W. Hughes, M. Minkov, B. Bartlett, S. Pai, S. Fan, “Reprogrammable Electro-Optic Nonlinear Activation Functions for Optical Neural Networks”, IEEE Journal of Selected Topics in Quantum Electronics, vol.26, No.1, pp.1-12, Jan.-Feb. 2020.

[9] P. Gautam, P. Lohani, B. Mishra, “Peak-to-Average Power Ratio reduction in OFDM system using amplitude clipping”, 2016 IEEE Region 10 Conference (TENCON), 2016, pp.1101-1104.

[10] D. Harris, S.L. Harris, “Digital design and computer architecture”,Morgan Kaufmann, 2016.

[11] “IEEE Standard for Floating-Point Arithmetic”, IEEE Std 754, 2019.

[12] B. Widrow, I. Kollár, “Quantization Noise: Roundoff Error in Digital Computation, Signal Processing, Control, and Communications”, Cambridge University Press, Cambridge, 2008.