FACTA UNIVERSITATIS
Series: **Automatic Control and Robotics** Vol. 21, Nº 2, 2022, pp. 95 - 105
https://doi.org/10.22190/FUACR220626008D

Regular Paper

OPTIMIZATION AND PERFORMANCE ANALYSIS OF THE 30-BIT FIXED-POINT DIGITAL FORMAT

UDC ((681.586+621.391):004.4)

Milan Dinčić[1], Zoran Perić[2], Dragan Denić[1]

[1]University of Niš, Faculty of Electronic Engineering, Department of Measurements, Republic of Serbia
[2]University of Niš, Faculty of Electronic Engineering, Department of Telecommunications, Republic of Serbia

Abstract. *The 32-bit floating-point format (FP32) is standardly used for digital representation of data in computers, providing high quality of digital representation in a very wide dynamic range of data. However, the FP32 format has a very high computational complexity, requiring the use of expensive and powerful hardware, as well as high energy consumption. Hence, the implementation of the FP32 format on devices such as smart sensors, embedded and edge devices that have limited hardware resources becomes very problematic. On the other hand, the fixed-point format has significantly less computational complexity, consumes less power, requires less area on chip and provides faster calculations than the floating-point format, being much more suitable for implementation on devices with limited hardware resources.*
The main goal of this paper is to find a fixed-point format that will be a good replacement for the FP32 format, in the sense that it provides the same performance as the FP32 format and at the same time significantly reduces the computational complexity. Therefore, the paper considers the 30-bit fixed-point format, optimizes the value of its parameters and evaluates its performance, using the analogy between the fixed-point digital representation and uniform quantization. As the main result, the paper shows that the 30-bit fixed-point format can achieve a better quality (i.e. higher SQNR) of digital representation for 3.352 dB compared to the FP32 format, saving at the same time 2 bits per each piece of data (which can be a significant saving for a large amount of data) and significantly reducing the complexity of the implementation. Therefore, the proposed 30-bit fixed-point format can be successfully used as a replacement for the FP32 format on devices with limited resources.

Key words: *Fixed-point digital format, floating-point digital format, uniform quantization, piecewise uniform quantization, smart sensors, resource-constrained devices*

Received July 26, 2022 / Accepted October 17, 2022
Corresponding author: Milan Dinčić
University of Niš, Faculty of Electronic Engineering, Department of Measurements, Aleksandra Medvedeva 14, 18000 Niš, Republic of Serbia
E-mail: milan.dincic@elfak.ni.ac.rs

1. INTRODUCTION

Today there is the dominance of digital systems that use digital representation of data. The main condition that the digital representation must satisfy is to ensure the appropriate accuracy of the digitized data (usually increasing the number of bits in the digital representation increases the accuracy of the representation). Also, it is important for the digital representation to be adapted to the dynamic range of data, i.e. to ensure the required accuracy in the entire dynamic range of data (this is very important for data with a wide dynamic range, i.e. with a large difference between the smallest and the largest value). However, due to the constantly increasing amount of data, it is very important for the digital representation to be efficient in terms of providing the required accuracy of the representation with as few bits as possible.

There are two basic types of digital formats for data representation: the floating-point format [1, 2] and the fixed-point format. The floating-point format, defined by the IEEE 754 standard [1], is dominantly used for the digital representation of data in computers since it provides a high quality of digital representation in a very wide dynamic range of data (from very small to very large data values). However, the main disadvantage of the floating-point format is the high computational complexity, which requires the use of expensive and powerful hardware. Another negative consequence of the floating-point processing is high energy consumption [3]. While powerful computers can easily support the floating-point format, its implementation on devices such as smart sensors, embedded and edge devices that have limited hardware resources (limited processing power, limited energy since they are mostly battery-powered, as well as limited memory capacity) becomes very problematic. In fact, many embedded and edge devices do not support the floating-point format at all [4].

On the other hand, the fixed-point format has significantly less computational complexity, consumes less power, requires less area on chip and provides faster calculations than the floating-point format [5, 6, 7], being much more suitable for implementation on devices with limited hardware resources.

A particularly current trend is the implementation of DNN (deep neural networks) [8] on smart sensors and edge devices. Since DNN parameters are standardly represented in the 32-bit floating-point format (FP32), the possibility of implementing DNN on these devices is significantly limited. One very effective way to overcome this problem is to represent the DNN parameters in the fixed-point format. Based on all the above, the analysis and optimization of the fixed-point format becomes a very important research topic.

The main goal of this paper is to find a fixed-point format that will be a good replacement for the FP32 format, in the sense that it provides the same performance as the FP32 format and at the same time significantly reduces the computational complexity.

In the paper [9], an analogy was established between the floating-point digital format and piecewise uniform quantization, showing that the floating-point format can be considered as a piecewise uniform quantizer. Also, in the paper [10], an analogy between the fixed-point format and uniform quantization was established, showing that the fixed-point digital format can be considered as a uniform quantizer. These analogies between digital formats and quantizers are very important because they allow us to express the performance of digital formats through an objective quantizer performance, such as distortion and signal-to-quantization noise ratio (SQNR).

As the performance of the quantizers depends on the probability density function (PDF) of the input data, the accuracy of the digital representation also depends on the PDF of the input data. This paper considers the Laplacian PDF, which is widely used for statistical modeling of different types of data [11, 12].

In this paper, the 32-bit floating-point (FP32) digital format is considered first. The analogy between the FP32 format and the 32-bit piecewise uniform quantizer is explained; then, using this analogy, the quality of the FP32 format is expressed by the SQNR of the 32-bit piecewise uniform quantizer. It is shown that the quality of digital representation of the FP32 format corresponds to the SQNR value of 151.934 dB. Since the FP32 format is robust [9], this value of SQNR is constant in a very wide range of variance of the input data.

The key part of the paper is the optimization of the 30-bit fixed-point format in terms of determining the optimal value of the parameter n which represents the number of bits used to encode the integer part of real numbers, using an analogy with the 30-bit uniform quantizer. An iterative algorithm for the calculation of the optimal value of the parameter n is defined and it is shown that the optimal value is $n = 5$ for data with the unit variance. Using the mentioned analogy, the performance of the 30-bit fixed-point format is analyzed, showing that it achieves a quality of digital representation equivalent to the SQNR value of 155.286 dB for data with the unit variance. As the fixed-point format is not robust [10], the SQNR value will decrease if the variance of the input data deviates from 1. However, if an adaptation is performed as suggested in the paper, the 30-bit fixed-point format will achieve constant SQNR of 155.286 dB in a very wide range of variance, similar to the FP32 format.

Hence, the main contribution of this paper is the design of the 30-bit fixed-point format that achieves a better quality (i.e. higher SQNR) of digital representation for 3.352 dB in a wide range of data variance compared to the FP32 format, saving at the same time 2 bits per each piece of data (which can be a significant saving for a large amount of data) and significantly reducing the complexity of the implementation. Therefore, the proposed 30-bit fixed-point format can be successfully used as a replacement for the FP32 format on devices with limited resources.

2. THE 32-BIT FLOATING POINT QUANTIZER

We will firstly consider the 32-bit floating-point (FP32) binary format, standardly used for the binary representation of data. A real number x can be represented in the FP32 format as:

$$x = (se_1e_2...e_8\ m_1m_2...m_{23})_2 . \tag{1}$$

In the binary representation (1) we have one bit ‘s’ intended for encoding the sign of the real number x, 8 bits ($e_1e_2...e_8$) intended for encoding the exponent and 23 bits ($m_1m_2...m_{23}$) representing the significand. The exponent E is calculated as:

$$E = \sum_{i=1}^{8} e_i 2^{8-i} \tag{2}$$

and can take values from 0 to 255. A real number x represented in the FP32 format (1) is calculated as:

$$x = (-1)^s \cdot (1.m_1 ... m_{23})_2 \cdot 2^{E-127} = (-1)^s \cdot (1.m_1 ... m_{23})_2 \cdot 2^{E^*}, \tag{3}$$

where $E^* = E - 127$ represents the biased exponent that can take values from -127 to 128. However, according to the IEEE 754 standard [1], the two end values ($E^* = -127$ and $E^* = 128$) are reserved for special purposes, hence the values from $E^* = -126$ to $E^* = 127$ are available for the binary representation of numbers. It is valid that:

$$(1.m_1 ... m_{23})_2 = 1 + \sum_{i=1}^{23} m_i 2^{-i} = 1 + 2^{-23} \sum_{i=1}^{23} m_i 2^{23-i} = 1 + \frac{M}{2^{23}}, \tag{4}$$

whereby the parameter M, defined as:

$$M = \sum_{i=1}^{23} m_i 2^{23-i} \tag{5}$$

can take values from 0 to $2^{23} - 1$. Based on (3) and (4), a real number x represented in the FP32 format is calculated as:

$$x = (-1)^s \cdot (1.m_1 ... m_{23})_2 \cdot 2^{E-127} = (-1)^s \cdot 2^{E^*} \left(1 + \frac{M}{2^{23}}\right). \tag{6}$$

For each positive number represented in the FP32 format, there is a corresponding negative counterpart, meaning that the FP32 format is symmetrical about zero. The maximal positive number that can be represented in the FP32 format (for $E^* = 127$ and $2^{23} - 1$) is

$$x_{\max} = 2^{127}\left(1 + \frac{2^{23} - 1}{2^{23}}\right) = 2^{127}\left(2 - \frac{1}{2^{23}}\right) = 2^{128}\left(1 - \frac{1}{2^{24}}\right) \approx 2^{128}. \tag{7}$$

Due to the symmetry, the maximal negative number represented in the FP32 format is -2^{128}.

Let us consider the positive numbers in the FP32 format. There are 254 values of E^* ($-126 \le E^* \le 127$) and for each value of E^* there are 2^{23} values of M ($0 \le M \le 2^{23} - 1$). For each value of E^* we have a group of 2^{23} numbers (each of them corresponding to one value of M). Therefore, in total we have 254 groups (whereby each group corresponds to one value of E^*) with 2^{23} numbers within each of them. Let us consider a group G_{E^*} of 2^{23} adjacent numbers for some arbitrary value of E^*. Let $x_{E^*,i}$ denote the i-th number in that group obtained for $M = i$ ($0 \le i \le 2^{23} - 1$). According to (6) we have that:

$$x_{E^*,i} = 2^{E^*}\left(1 + \frac{i}{2^{23}}\right). \tag{8}$$

The distance between two adjacent numbers within the group G_{E^*} is

$$\Delta_{E^*} = x_{E^*,i+1} - x_{E^*,i} = 2^{E^*}\left(1 + \frac{i+1}{2^{23}}\right) - 2^{E^*}\left(1 + \frac{i}{2^{23}}\right) = 2^{E^*-23}. \tag{9}$$

Numbers from the group G_{E^*} belongs to the interval $S_{E^*} = [2^{E^*},2^{E^*+1})$.

In total, we have 254 groups of 2^{23} numbers in the positive part; the numbers from one group have the same value of E^*, being uniformly distributed in the segment $S_{E^*} = [2^{E^*},2^{E^*+1})$ with the step-size $\Delta_{E^*} = 2^{E^*-23}$. Since the step-size Δ_{E^*} depends on E^*, it is obvious that the step-size has different values in different groups. A symmetrical structure exists in the negative part.

We can see that the structure of the FP32 format corresponds to the structure of a symmetrical 32-bit piecewise uniform quantizer with the support region $[-x_{max}, x_{max}]$, which consists of 254 segments $[2^{E^*},2^{E^*+1})$ in the positive part ($-126 \le E^* \le 127$), whereby the uniform quantization with 2^{23} quantization levels and with quantization step-size $\Delta_{E^*} = 2^{E^*-23}$ is performed within each segment. This 32-bit piecewise uniform quantizer that corresponds to the structure of the FP32 format will be called *the 32-bit floating-point quantizer*. This analogy between the FP32 representation and the 32-bit floating-point quantizer will allow us to determine the quality of the FP32 binary representation, based on the objective performance (distortion D and SQNR) of the 32-bit floating-point quantizer.

During the quantization, an irreversible error is made, which is expressed by distortion. For the 32-bit piecewise floating-point quantizer, the distortion D is calculated as:

$$D = 2\sum_{E^*=-126}^{127} \frac{\Delta_{E^*}^2}{12} P_{E^*} + 2\int_{x_{\max}}^{+\infty} (x - x_{\max})^2 \, p(x)dx \,. \tag{10}$$

Multiplication by 2 in the expression (10) serves to incorporate the distortion in the negative part, which is the same as the distortion in the positive part due to the symmetry. The first term in (10), expressed in the form of a sum, represents the granular distortion that corresponds to the quantization error caused by quantizing data belonging to the support region of the quantizer. The granular distortion of the 32-bit piecewise uniform quantizer is expressed in the form of a sum, where each member of the sum represents the distortion of the uniform quantization in one of 254 segments. P_{E^*} represents the probability that the input data belongs to the segment $[2^{E^*},2^{E^*+1})$. The probability P_{E^*} is defined as:

$$P_{E^*} = \int_{2^{E^*}}^{2^{E^*+1}} p(x)dx, \tag{11}$$

where $p(x)$ represents the probability density function (PDF) of the input data.

The second term in the expression (10) represents the overload distortion that corresponds to the quantization error caused by quantizing data outside the support region of the quantizer. We can see that the overload distortion also depends on the PDF of the input data. In general, the performance of any quantizer depends on the PDF of the input data, i.e. we need to know the PDF of the input data to determine the performance of the quantizer. In this paper, we will consider the Laplacian PDF which is widely used for statistical modeling of many types of data [11, 12]. The Laplacian PDF is defined by the following expression [11]:

$$p(x) = \frac{1}{\sqrt{2}\sigma} \exp\left(-\frac{|x|\sqrt{2}}{\sigma}\right), \tag{12}$$

where σ^2 represents the variance of the input data. Quantizers are usually designed for the unit variance ($\sigma^2 = 1$), hence we will below consider the Laplacian PDF with the unit variance defined as:

$$p(x) = \frac{1}{\sqrt{2}} \exp\left(-|x|\sqrt{2}\right). \tag{13}$$

The probability P_{E^*} for $p(x)$ defined by (13) is calculated as:

$$P_{E^*} = \frac{1}{2}\left(\exp\left(-2^{E^*+\frac{1}{2}}\right) - \exp\left(-2^{E^*+\frac{3}{2}}\right)\right). \tag{14}$$

The expression for the overload distortion for $p(x)$ defined by (13) becomes:

$$D_{ov} = \exp(-\sqrt{2}x_{\max}) = \exp(-\sqrt{2}\cdot 2^{128}). \tag{15}$$

Substituting (11), (14) and (15) into (10), the expression for the distortion of the 32-bit floating-point quantizer becomes:

$$D = \sum_{E^*=-126}^{127} \frac{1}{12}\cdot 2^{2E^*-46}\cdot\left(\exp\left(-2^{E^*+\frac{1}{2}}\right) - \exp\left(-2^{E^*+\frac{3}{2}}\right)\right) + \exp(-\sqrt{2}\cdot 2^{128}). \tag{16}$$

Quality of the quantization is expressed by SQNR (signal-to-quantization noise ratio), which is defined as

$$\text{SQNR [dB]} = 10\log_{10}\frac{\sigma^2}{D}. \tag{17}$$

For $\sigma^2 = 1$ and for the distortion defined by (16), the expression for the SQNR of the 32-bit floating-point quantizer becomes:

$$\text{SQNR [dB]} = -10\log_{10}\left[\sum_{E^*=-126}^{127} \frac{2^{2E^*-46}}{12}\cdot\left(\exp\left(-2^{E^*+\frac{1}{2}}\right) - \exp\left(-2^{E^*+\frac{3}{2}}\right)\right) + \exp(-\sqrt{2}\cdot 2^{128})\right]. \tag{18}$$

After calculation we obtain that SQNR = 151.934 dB.

The 32-bit floating-point quantizer is very robust since it has a constant SQNR over a very wide range of variance [9]. This means that even if the variance of the input data deviates significantly from $\sigma^2 = 1$, the SQNR will keep the same value, i.e. the quality of the FP32 representation will remain the same.

3. THE 30-BIT FIXED POINT QUANTIZER

A real number x is represented in the 30-bit fixed-point binary representation as:

$$x = (s a_{n-1} a_{n-2} \ldots a_1 a_0 \,.\, a_{-1} \ldots a_{-m})_2. \tag{19}$$

The binary representation (19) consists of one bit '*s*' intended for encoding of the sign of *x* (*s* = 0 if *x* > 0 and *s* = 1 if *x* < 0), *n* bits ($a_{n-1}a_{n-2}...a_1a_0$) intended for encoding of the integer part of *x* and *m* bits ($a_{-1}...a_{-m}$) intended for encoding of the fractional part of *x*. It holds that *n* + *m* + 1 = 30, therefore, we have that:

$$m = 29 - n \,. \tag{20}$$

The main issue related to the fixed-point format is how to choose values of parameters *n* and *m*. According to (20), if we find the optimal value of *n*, we can easily calculate the optimal value of *m*.

The fixed-point binary format is a weighted format, whereas each bit a_i ($i = -m, \ldots, n-1$) has the weight of 2^i. Using this fact, we can calculate the real number *x* represented in the 30-bit fixed-point binary format as:

$$x = (-1)^s \sum_{i=-m}^{n-1} a_i 2^i \,. \tag{21}$$

For each positive number represented in the fixed-point format, there is a negative counterpart. Hence, the fixed-point format is symmetrical about zero. The number 0 is represented with all bits equal to 0. The largest positive number that can be represented in the 30-bit fixed-point format is:

$$x_{\max} = (1...1.1...1)_2 = \sum_{i=-m}^{n-1} 2^i = 2^{-m} \sum_{i=0}^{n+m-1} 2^i = 2^{-m}(2^{n+m} - 1) = 2^n - 2^{-m} \,. \tag{22}$$

Using (20) it is obtained that:

$$x_{\max} = 2^n - 2^{n-29} = 2^n \left(1 - \frac{1}{2^{29}}\right) \approx 2^n \,. \tag{23}$$

Due to the symmetry, the largest negative number that can be represented is - 2^n.

Let us consider the first few positive numbers represented in the 30-bit fixed point format:

$$(0...0.0...01)_2 = 2^{-m} \,,$$
$$(0...0.0...010)_2 = 2^{-(m-1)} = 2 \cdot 2^{-m} \,,$$
$$(0...0.0...011)_2 = 2^{-(m-1)} + 2^{-m} = 3 \cdot 2^{-m} \,, \text{ etc.}$$

It is obvious that the numbers represented in the 30-bit fixed point format are uniformly distributed (i.e. equidistant) discrete numbers, whereas the distance between adjacent numbers is $\Delta = 2^{-m}$. Hence, we can conclude that the 30-bit fixed-point format represents uniformly distributed discrete numbers from the range $[-x_{\max}, x_{\max}] = [-2^n, 2^n]$, with the step-size $\Delta = 2^{-m}$. According to (20), the step-size Δ can be written as:

$$\Delta = 2^{n-29} \,. \tag{24}$$

Based on the above, the 30-bit fixed-point representation can be considered as a 30-bit uniform quantizer with the following parameters: the maximal amplitude $x_{max} = 2^n$, the support region $[-2^n, 2^n]$ and the quantization step-size $\Delta = 2^{n-29}$. This uniform quantizer that corresponds to the 30-bit fixed-point representation will be called *the 30-bit fixed-point quantizer*. This analogy between the 30-bit fixed-point representation and the 30-bit

fixed-point uniform quantizer will allow us to assess the quality of the 30-bit fixed-point representation based on the performance of the 30-bit fixed-point quantizer. Therefore, we will analyze the performance of the 30-bit fixed-point quantizer below.

Based on the quantization theory, the distortion of the 30-bit fixed-point quantizer can be expressed as [11]:

$$D = \frac{\Delta^2}{12} + 2\int_{x_{\max}}^{+\infty} (x - x_{\max})^2 \, p(x)dx \,. \tag{25}$$

The first term in the expression (25) represents the granular distortion while the second term in (25) represents the overload distortion. For the unit-variance Laplacian PDF defined with (13), the overload distortion becomes:

$$D_{ov} = 2\int_{x_{\max}}^{+\infty} (x - x_{\max})^2 \, p(x)dx = \exp(-\sqrt{2} \cdot x_{\max}) \,. \tag{26}$$

Substituting (26) in (25), we obtain the following expression for the distortion of the 30-bit fixed-point quantizer:

$$D = \frac{\Delta^2}{12} + \exp(-\sqrt{2} \cdot x_{\max}) \,. \tag{27}$$

Using (24) and (23), the distortion D can be expressed as a function of the parameter n, in the following way:

$$D(n) = \frac{2^{2n-58}}{12} + \exp(-\sqrt{2} \cdot 2^n) = \frac{2^{2n-58}}{12} + \exp\left(-2^{n+\frac{1}{2}}\right). \tag{28}$$

The quality of the 30-bit fixed-point quantization is expressed by an objective measure SQNR (signal-to-quantization noise ratio), which is defined in the following way, being also a function of the parameter n:

$$\mathrm{SQNR}(n)\,[\mathrm{dB}] = -10 \cdot \log_{10}(D(n)) = -10 \cdot \log_{10}\left(\frac{2^{2n-58}}{12} + \exp\left(-2^{n+\frac{1}{2}}\right)\right). \tag{29}$$

The performance of the 30-bit fixed-point quantizer is expressed by objective measures: by the distortion D and by SQNR.

Our goal is to optimize the value of the parameter n in a way to maximize the quality of the 30-bit fixed-point representation. Because of the analogy between the 30-bit fixed-point representation and the 30-bit fixed-point quantizer, maximizing the quality of the 30-bit fixed-point format is equivalent to achieving the best performance of the 30-bit fixed-point quantizer. Therefore, the optimization of the value of the parameter n can be performed by maximizing the SQNR or minimizing the distortion D of the 30-bit fixed-point quantizer. Thus, the analogy between the 30-bit fixed-point format and the 30-bit fixed-point quantizer allows us to optimize the 30-bit fixed-point representation using objective quality measures (D and SQNR) of the 30-bit fixed-point quantizer.

In order to minimize the distortion D, we will find the first derivative of the function $D(n)$:

$$\frac{dD(n)}{dn} = 2^{n+\frac{1}{2}} \ln 2 \cdot \left(\frac{2^{-58}}{12} 2^{n+\frac{1}{2}} - \exp\left(-2^{n+\frac{1}{2}} \right) \right). \tag{30}$$

From the condition $\frac{dD(n)}{dn} = 0$ the following equation is obtained:

$$\frac{2^{-58}}{12} 2^{n+\frac{1}{2}} - \exp\left(-2^{n+\frac{1}{2}} \right) = 0 . \tag{31}$$

Let us define the following substitution:

$$t = 2^{n+\frac{1}{2}} . \tag{32}$$

The equation (31) becomes:

$$\exp(-t) = \frac{2^{-58}}{12} t . \tag{33}$$

By logarithmization of both sides of the equation (33), it is obtained that:

$$t = -\ln\left(\frac{2^{-58}}{12} t \right) = \ln\left(\frac{12 \cdot 2^{58}}{t} \right) = \ln(12) + 58 \cdot \ln 2 - \ln t = 42.6874 - \ln t . \tag{34}$$

Based on (34), we can define the following iterative process for calculating the optimal value of the parameter t:

$$t_{i+1} = 42.6874 - \ln t_i . \tag{35}$$

where t_i denotes the value of t in the i-th iteration.
Let us arbitrarily choose the value of the starting point of the iterative process as $t_0 = 20$. The values of the parameter t in the first few iterations are: $t_1 = 39.6917$, $t_2 = 39.0063$, $t_3 = 39.0237$, $t_4 = 39.0233$, $t_5 = 39.0233$, $t_6 = 39.0233$. We can see that the iterative process very quickly (after 4 iterations) reaches the value 39.0233 and remains at that value. We can say that the iterative process (35) converges to the value 39.0233, therefore the optimal value of the parameter t is $t_{opt} = 39.0233$. Based on (32), the optimal value of the parameter n is calculated as:

$$n_{opt} = \log_2(t_{opt}) - \frac{1}{2} = 4.786 . \tag{36}$$

However, the parameter n must be an integer. Since n_{opt} is between 4 and 5, it is clear that the optimal integer value of the parameter n must be 4 or 5. According to (29), SQNR($n = 4$) = 98.270 dB and SQNR($n = 5$) = 155.286 dB. Since SQNR($n = 5$) > SQNR($n = 4$), it is obvious that the optimal value of the parameter n for the 30-bit fixed-point format is $n = 5$. Based on (20), (23) and (24), the optimal values of the parameters of the 30-bit fixed-point quantizer are:

$$n = 5,\ m = 24,\ x_{\max} = 2^n = 32 \text{ and } \Delta = 2^{-m} = 2^{-24} . \tag{37}$$

The SQNR of the 30-bit fixed-point quantizer with parameters defined by (37) is 155.286 dB.

Let us calculate SQNR of the 30-bit fixed-point quantizer for several values of the parameter n close to the optimal value $n = 5$, which is shown in Table 1. We can see from Table 1 that the wrong choice of the value of the parameter n drastically reduces the SQNR, therefore reducing the 30-bit fixed-point representation. This fact confirms the importance of determining the optimal value of the parameter n.

Table 1 SQNR of the 30-bit fixed-point quantizer for different values of the parameter n

n	SQNR [dB]
3	49.135
4	98.270
5	155.286
6	149.266
7	143.245

The fixed-point format has significantly less robustness than the floating-point format [10], so the SQNR value will decrease as the value of the data variance moves away from 1. However, by applying well-known adaptation methods (such as the forward adaptation, which involves grouping data into blocks, calculating the variance σ^2 of each block, dividing data in the block by σ to normalize the variance of the block to 1, and storing σ in order to subsequently reconstruct the original data values) [11], the SQNR of the 30-bit fixed-point format can easily maintain a constant value of 155.286 dB over a very wide range of variance of the input data. Increasing of the processing time due to adaptation depends on the size of the block of data for adaptation, therefore the size of the block should be optimized taking into account the allowed processing time and the required quality of adaptation for a specific application. The proposed adaptation techniques have been successfully used in real-time applications for a long time (e.g. speech and image transmission [11]), showing that the increasing of complexity and processing time due to the adaptation techniques is not critical.

We can see that the adaptive 30-bit fixed-point format achieves for 3.352 dB higher SQNR compared to the 32-bit floating-point format (FP32), with a saving of 2 bits per each data element. At the same time, the 30-bit fixed-point format has significantly less complexity compared to the FP32 format. Increasing of SQNR contributes to the quality of digital data representation. In the context of DNN, there is a direct correlation between SQNR of data representation and prediction/classification accuracy; hence, the increasing the SQNR of 3,352 dB may have a certain positive effect on the prediction/classification accuracy.

7. CONCLUSION

The paper considered the 30-bit fixed-point format using an analogy with the 30-bit uniform quantizer in terms of determining the optimal value of the parameter n which represents the number of bits used to encode the integer part of real numbers, using an analogy with the 30-bit uniform quantizer. An iterative algorithm was defined for optimization of the parameter n (the number of bits used to encode the integer part of real numbers). It was shown that the optimal value of n is 5 for data with the unit variance. Also, it was shown that the 30-bit fixed-point format could achieve a quality of digital representation equivalent to the SQNR

value of 155.286 dB. By performing some of the well-known adaptation techniques (such as the forward adaptation), the mentioned value of SQNR can be kept constant in a very wide range of variance.

The main conclusion of the paper is that the 30-bit fixed-point format can achieve a better quality (i.e. higher SQNR) of digital representation for 3.352 dB in a wide range of data variance compared to the FP32 format, saving at the same time 2 bits per each piece of data (which can be a significant saving for a large amount of data) and significantly reducing the complexity of the implementation. Therefore, the proposed 30-bit fixed-point format can be successfully used as a replacement for the FP32 format on devices with limited resources.

Acknowledgement: *This work has been supported by the Science Fund of the Republic of Serbia (Grant No. 6527104, AI- Com-in-AI) as well as by the Ministry of Education, Science and Technological Development of the Republic of Serbia (projects TR-32045 and III-42009).*

REFERENCES

[1] IEEE Standard for Floating-Point Arithmetic IEEE 754-2019, https://standards.ieee.org/ieee/754/6210/.

[2] D. Zoni, A. Galimberti and W. Fornaciari, "An FPU design template to optimize the accuracy-efficiency-area trade-off", *Sustainable Computing: Informatics and Systems*, vol. 29, part A, March 2021, doi: 10.1016/j.suscom.2020.100450.

[3] G. Tagliavini, S. Mach, D. Rossi, A. Marongiu and L. Benini, "A transprecision floating-point platform for ultra-low power computing", *2018 Design, Automation Test in Europe Conference Exhibition (DATE)*, March 2018, pp. 1051–1056.

[4] D. Cattaneo, A. Di Bello, S. Cherubin, F. Terraneo and G. Agosta, "Embedded Operating System Optimization through Floating to Fixed Point Compiler Transformation", *2018 21st Euromicro Conference on Digital System Design (DSD)*, Prague, Czech Republic, 29-31 August 2018, doi: 10.1109/DSD.2018.00042.

[5] MathWorks. Benefits of fixed-point hardware. [Online]. Available: https://de.mathworks.com/help/fixedpoint/gs/benefitsof-fixed-point-hardware.html.

[6] NI. (2019) Advantages of fixed-point numbers on hardware. [Online]. Available: https://www.ni.com/documentation/en/labview/latest/datatypes/advantages-fixed-point-numbers/.

[7] R. T. Syed, M. Ulbricht, K. Piotrowski, and M. Krstic, "Fault Resilience Analysis of Quantized Deep Neural Networks", *2021 IEEE 32nd International Conference on Microelectronics (MIEL)*, Niš, Serbia, September 12-14, 2021, pp. 275-279, doi: 10.1109/MIEL52794.2021.9569094.

[8] A. Zhang, Z. –C. Lipton, M. Li and A. -J. Smola, "Dive into Deep Learning", Amazon Science, (2020).

[9] Z. Peric, M. Savic, M. Dincic, N. Vucic, D. Djosic and S. Milosavljevic, "Floating point and fixed point 32-bits quantizers for quantization of weights of neural networks", *12th International Symposium on Advanced Topics in Electrical Engineering (ATEE)*, Bucharest, Romania, March 2021, pp. 1-4, doi: 10.1109/ATEE52255.2021.9425265.

[10] Z. Perić, A. Jovanović, M. Dinčić, M. Savić, N. Vučić and A. Nikolić, "Analysis of 32-bit Fixed Point Quantizer in the Wide Variance Range for the Laplacian Source", *2021 15th International Conference on Advanced Technologies, Systems and Services in Telecommunications (TELSIKS)*, Niš, Serbia, 20-22 October 2021, doi: 10.1109/TELSIKS52058.2021.9606251.

[11] N. C. Jayant and P. Noll, "Digital Coding of Waveforms: Principles and Applications to Speech and Video", Prentice Hall, (1984).

[12] J. Nikolić, D. Aleksić, Z. Perić and M. Dinčić, "Iterative Algorithm for Parameterization of Two-Region Piecewise Uniform Quantizer for the Laplacian Source", *Mathematics*, vol. 9, no. 23: 3091, 2021, doi: 10.3390/math9233091.