Hindawi
Mathematical Problems in Engineering
Volume 2022, Article ID 1463094, 15 pages
https://doi.org/10.1155/2022/1463094

Research Article

Performance of Post-Training Two-Bits Uniform and Layer-Wise Uniform Quantization for MNIST Dataset from the Perspective of Support Region Choice

Stefan Tomić,[1] **Jelena Nikolić**,[2] **Zoran Perić**,[2] **and Danijela Aleksić**[3]

[1]*School of Engineering and Technology, Al Dar University College, Dubai, UAE*
[2]*Faculty of Electronic Engineering, University of Niš, Niš 18000, Serbia*
[3]*Telekom Srbija, Department of Mobile Network Niš, Voždova 11, Niš, Serbia*

Correspondence should be addressed to Jelena Nikolić; jelena.nikolic@elfak.ni.ac.rs

Received 15 September 2021; Accepted 17 February 2022; Published 7 April 2022

Academic Editor: Hao Gao

This paper contributes to the goal of finding an efficient compression solution for post-training quantization from the perspective of support region choice under the framework of low-bit uniform quantization. The decision to give preference to uniform quantization comes from the fact that support region choice is the most sensitive in the uniform quantization of nonuniform sources (e.g., Laplacian sources). Therefore, in this paper, we analyse in detail how the choice of the support region influences the performance of two-bits uniform quantization, measured with signal to quantization noise ratio (SQNR), and the accuracy of the compressed neural network (NN) model. We provide experimental and theoretical results for a few significant cases of two-bits uniform quantizer design, where we assume that Laplacian source models the distribution of weights in our fully connected NN. We opt for Laplacian distribution since it models well weights of NNs. Specifically, we analyse whether it is possible to apply the simplest uniform quantization in trained NN model weight representation with a bit rate of $R = 2$ bit/sample while preserving the accuracy of the model to a great extent. Also, our goal is to determine whether the choice of the key parameter of two-bits uniform quantizer (support region threshold) equally reflects on both, SQNR and accuracy. Moreover, we extend our analysis to the application of layer-wise two-bits uniform quantization in order to examine whether it is possible to achieve an additional improvement of the accuracy of our NN model for the MNIST dataset. We believe that the detailed analysis of post-training quantization described and conducted in this paper is very useful for all further research studies of this very current topic, especially due to the fact that the problem regarding post-training quantization is addressed from a particularly important perspective of choosing the support region.

1. Introduction

Although neural networks (NNs) have made significant improvements in various real-world solutions, in some scenarios, their usage might be limited or impeded in edge devices with constrained resources, such as IoT sensors and mobile devices [1]. To overcome the aforementioned potential problems with NNs and mitigate aggravating circumstances, various enhancement techniques, such as clipping, pruning, knowledge distillation, and quantization, have been proposed [1–4]. Relying on a plethora of previous conclusions about quantization for traditional network solutions [5–14] and considering the benefits of low-bit presentations [15–30], further enhancements of NNs are intuitively motivated by the application of quantization. Quantization is significantly beneficial for NN implementation on resource-limited devices as it can enable fitting the whole NN model into an on-chip memory of edge devices so that the high overhead occurred by the off-chip memory access can be mitigated [2]. Numerous papers have confirmed that the growing interest in NN optimization is indeed directed towards the utilization of quantization methods during various training phases (e.g., see [1, 21, 29]). It is shown that quantization, as an important compression

technique, can dictate the entire NN performance, especially in the most rigorous case with aggressive bit reduction for extremely low-bit NN, such as in binarized neural networks [15, 16, 23–27, 29]. Although numerous previous papers have given general guidelines on the utilization of different possible quantization methods for NNs [1–4, 13, 15–36], there are many intriguing directions for future research, even in the case of the most exploited uniform quantization model [3, 4, 17–21, 31, 32].

In recent years, a few new quantizer models and quantization methodologies have been proposed in [1, 27, 34–36], with a main objective, to enable quantized NNs to have acceptable accuracy as with using the initial single-precision floating-point (FP32) format. Since in NNs the low-bit quantization implies that floating-point NN parameters are presented with a much lower number of discrete values, reducing the inevitable quantization error is a really great challenge. An even greater challenge is to achieve this goal if simple low-bit quantizer models are used, such as the simple one with low complexity as the two-bits uniform quantizer (UQ), which is the subject of this paper. In very recent related previous works, [31, 36], the goal was to achieve the highest possible signal to quantization noise ratio (SQNR) with the utilized quantizer for NN weight compression. In contrast, in this paper, the predominant goal is not only to achieve the highest possible SQNR but also to maintain a high QNN (quantized neural network) model's accuracy and derive firm conclusions on the effects of different two-bits UQ parameter choices on both SQNR and accuracy, in the case when the MNIST database is brought to the input of the fully connected (FC) NN. Since the support region was recognised as the most striking parameter of a quantizer design [7–13, 36–40], in this paper, we pay special attention to the choice of this parameter for our two-bits UQ, and we show how this choice is of great importance in preserving accuracy with the application of post-training quantization. Post-training quantization is especially convenient as there is no need to retrain the network, while the memory size required for storing the NN model weights can be significantly reduced [1, 40]. We provide the experimental and theoretical results for the quantizer in question, where we assume the Laplacian source to model the distribution of weights in our FC NN, as it provides a good model for NN weight distribution [15, 18, 19, 31, 33, 36, 40]. The choice of the key parameter of our two-bits UQ, that is the support region width ($\mathfrak{R}_g$ width), is the most sensitive in the uniform quantization of nonuniform sources (e.g., Laplacian sources) [7–10], [14, 37, 38, 40]. Therefore, this paper evaluates the influence of this choice on the quantization performance, measured with SQNR, and the accuracy of the compressed NN model.

In brief, the following contributions are offered in this paper:

(i) An analysis of the influence of the two-bits UQ's key parameter choice on the SQNR and resulting accuracy of the QNN model for the MNIST dataset for a few significant cases of two-bits UQ design

(ii) The answer to the question whether the choice of the key parameter of two-bits UQ equally reflects on both, SQNR and QNN accuracy

(iii) The answer to the question whether it is possible to apply the simplest uniform quantization to represent the weights of a trained NN model with a bit rate of $R = 2$ bit/sample while preserving the accuracy of the model to a great extent

(iv) An analysis of a further accuracy improvement achievable by the completely novel approach based on the layer-wise uniform quantization (LWUQ) of weights in NNs for the given framework of the paper

To the best of the authors' knowledge, a similar analysis of the influence of the three-bits UQ's key parameter choice on the SQNR and resulting accuracy of the QNN model trained on the MNIST dataset has only been recently conducted in [40]. However, the analysis presented in this paper outputted completely different formulas for the simpler design and performance assessment of the two-bits UQ. Moreover, as an additional unique contribution of this paper, we propose a layer-wise adapted model of a two-bits uniform quantizer. We should highlight that due to completely different conclusions regarding the accuracy of QNN that can be derived for different bit rates, it is of importance to separately analyse and address the research question regarding the problem of the support region choice and its influence on the performance of the QNN model, for the particular bit rate. In other words, the mentioned problem of selecting the support region of UQ is more prominent in the two-bits case, when compared to the three-bits case addressed in [40], which makes the analysis for the two-bits case presented in this paper particularly significant. Namely, in this paper, we will show that the performance of the QNN model trained on the MNIST is highly dependent on the support region choice, whereas in [40], a weak influence of the support region choice on the performance of the QNN model has been ascertained, while a slightly higher accuracy preservation of the original NN model has been achieved. In order to still benefit from the one-bit compression over the quantization model from [40] and to improve the performance of the QNN model addressed in this paper, we also propose a completely novel approach based on the layer-wise adaptation of the two-bits uniform quantizer and consider the resulting performance of the QNN model determined theoretically and experimentally for the given framework of the paper. We will show that with the layer-wise adaptation of the two-bits UQ, the performance of the QNN model comparable to one of the three-bits UQ from [40] can be achieved, whereas in this paper, the original weights of the trained NN, stored in the FP32 format, are compressed with a higher compression factor compared to [40]. In particular, in this paper, we provide 16 times the compression of the originally trained weights, while the same weights stored in the FP32 format are compressed about 10 times in [40].

This paper is organized as follows: Section 2 describes a symmetrical two-bits UQ design and derivation of formulas for its theoretical performance assessment, when the input data are modelled by a Laplacian distribution, and for its experimental performance assessment when the real NN weights are uniformly quantized. In addition, a concise formulation of layer-wise uniform quantization is presented in Section 2. The description of the post-training quantization procedure and the main results obtained in the case of applying UQ and LWUQ are presented and discussed in Sections 3 and 4, respectively. Finally, in Section 5, we provide a summary of the paper's goals and conclude our research results.

2. Symmetrical Two-Bits UQ Design for the Laplacian Source and Its Layer-Wise Adjusted Version

Quantization specifies mapping a presumably random input to a discrete set of N quantization levels (representation levels). To express the error produced by the quantization process, mean squared error (MSE) distortion is a commonly used measure [37, 40]. The ultimate goal in quantization is to minimize the error between the quantized ($Q_N(X)$) and original data (X), for a given N, that is, for a given number of bits required to represent the data $R = \log_2 N$ (bit rate R) [37].

$$D = E\left[\left(X - Q_N(X)\right)^2\right]. \tag{1}$$

However, the goal in compression is to minimize the bit rate for reducing the memory footprint and computation requirements, which simultaneously increases the quantization error [2, 37]. Due to these conflicting requirements, quantization is a very intriguing area of research. As already mentioned, and highlighted in [40], the choice of the quantizer model itself and its support region (key parameter of every quantizer) affects the amount of quantization error. In particular, the choice of this key parameter is the most sensitive in uniform quantization of nonuniform sources (e.g., Laplacian sources) [7–10], [14, 37, 38, 40], which is why the influence of this choice on the performance of both, the quantizer and the QNN performance, is analysed in depth.

Firstly, we specify the key parameter of an N-level symmetrical quantizer Q_N. During the quantization process, the amplitude range of an input signal is divided into a granular region $\mathfrak{R}_g$ and an overload region $\mathfrak{R}_o$ (see Figure 1 drawn for the symmetric two-bits UQ). In the case of symmetric quantizer, as the one we design in this paper, and as the one we addressed in [40], these regions are separated by the support region thresholds, denoted by $-x_{\max}$ and $x_{\max}$, respectively. The granular region $\mathfrak{R}_g$ is defined by

$$\begin{aligned}\mathfrak{R}_g &= \bigcup_{i=-N/2}^{-1} \mathfrak{R}_i \cup \bigcup_{i=1}^{N/2} \mathfrak{R}_i \\ &= \left[-x_{\max}, x_{\max}\right].\end{aligned} \tag{2}$$

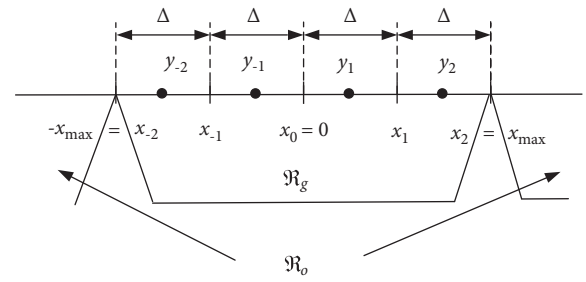

Figure 1: Granular region $\mathfrak{R}_g$ and overload region $\mathfrak{R}_o$ of the symmetric two-bits UQ.

And it consists of N nonoverlapping quantization cells of limited widths, where the i^{th} cell is defined as

$$\begin{aligned}\mathfrak{R}_i = \{x \mid x \in [-x_{\max}, x_{\max}], Q_N(x) = y_i\}, \\ \mathfrak{R}_i \cap \mathfrak{R}_j = \varnothing,\ i \neq j,\end{aligned} \tag{3}$$

while $\{\mathfrak{R}_i\}_{i=-N/2}^{-1}$ and $\{\mathfrak{R}_i\}_{i=1}^{N/2}$ denote the granular cells in the negative and positive in amplitude regions, which are symmetrical. In symmetric quantization, the quantizer's main parameter set is halved since only the positive or the absolute values should be determined and stored [37, 40]. The symmetry also holds for the overload quantization cells, that is, for a pair of cells of unlimited width in the overload region, $\mathfrak{R}_o$, defined as

$$\begin{aligned}\mathfrak{R}_0 = \{x \mid x \notin [-x_{\max}, x_{\max}], Q_N(x) = y_{N/2}, \\ x > 0 \vee Q_N(x) = y_{-N/2},\ x < 0\}.\end{aligned} \tag{4}$$

In low-bit quantization, a very small number of bits per sample is used to represent the data being quantized (less than or equal to 3 bit/sample) [37, 40]. If the quantization cells are of equal widths, then the quantizer is uniform, which is the case with the low-bit uniform quantizer we address here and the one from [40]. The code book of our two-bits UQ, $Y \equiv \{y_{-2}, y_{-1}, y_1, y_2\} \subset \mathbb{R}$, contains $N = 4$ representation levels, while in [40], the number of representation levels is doubled. These representation levels are denoted by y_i (see Figure 1) and specified as the midpoint of the i^{th} cell by

$$y_i = \frac{(x_{i-1} + x_i)}{2},\ y_{-i} = -y_i =, i \in \{1, 2\}, \tag{5}$$

where the cell borders are equidistant and specified by

$$x_i = i\Delta, \quad x_{-i} = -x_i,\ i \in \{0, 1, 2\}, \tag{6}$$

while Δ stands for the step size of our two-bits UQ and is given by

$$\begin{aligned}\Delta &= \frac{2x_{\max}}{N} \\ &= \frac{x_{\max}}{2}.\end{aligned} \tag{7}$$

We recall that $x_{\max}$ denotes the support region threshold of our two-bits UQ, and it is the key parameter of the quantizer in question. From (5)–(7), one can conclude that $x_{\max}$ completely determines the cell borders x_i and the representation levels y_i of the two-bits UQ. In other words, the quantizer in question is completely determined by the support region threshold $x_{\max}$. Therefore, similarly as in [40], we introduce the following notation of our transfer characteristic of the symmetric two-bits UQ, $Q^{\mathrm{UQ}}(x; x_{\max})$ (see Figure 2, where the characteristic of the symmetric two-bits UQ is presented for $x_{\max} = x_{\max}^{[\mathrm{J}]} = 2.1748$. The notation [J] comes from the name of the author of [37]). The choice of this important parameter, $x_{\max}$, will be examined in the rest of the paper.

Let us assume that, as in [40], the unrestricted Laplacian probability density function (pdf) of zero mean and variance σ^2 is

$$p(x) = \frac{1}{\sqrt{2}\sigma}\exp\left\{-\frac{\sqrt{2}\,x}{\sigma}\right\}. \tag{8}$$

The cell borders and the representation levels of our UQ are symmetrical in relation to the mean value. To determine the total distortion of our symmetrical two-bits UQ, composed of the granular and the overload distortion, $D_{\mathrm{g}}^{UQ} = D_{\mathrm{g}}^{UQ} + D_{o}^{UQ}$, we begin with the basic definition of distortion, given by (1) [37], where the granular distortion D_{g}^{UQ} and the overload distortion D_{o}^{UQ} of our symmetric two-bits UQ are defined as

$$\begin{aligned} D_g^{UQ} &= 2\sum_{i=1}^{2}\int_{x_{i-1}}^{x_i}(x-y_i)^2 p(x)dx\,, \\ D_o^{UQ} &= 2\int_{x_2}^{\infty}(x-y_2)^2 p(x)dx\,. \end{aligned} \tag{9}$$

Foremost, in order to simplify our derivation, let us define $x_3 = \infty$, where x_3 denotes the upper limit of the integral in (9). The total distortion of our two-bits UQ is then

$$D^{UQ} = 2\sum_{i=1}^{3}\int_{x_{i-1}}^{x_i} x^2 p(x)dx - 4\left(\sum_{i=1}^{2} y_i\int_{x_{i-1}}^{x_i} x p(x)dx + y_2\int_{x_2}^{\infty} x p(x)dx\right) + 2\left(\sum_{i=1}^{2} {y_i}^2\int_{x_{i-1}}^{x_i} p(x)dx + {y_2}^2\int_{x_2}^{\infty} p(x)dx\right). \tag{10}$$

By further reorganization of the last formula, we have that

$$D^{UQ} = \mathrm{I}^{\mathrm{I}} - 4\left(\sum_{i=1}^{2} y_i \mathrm{I}_i^{\mathrm{II}} + y_2 \mathrm{I}_3^{\mathrm{II}}\right) + 2\left(\sum_{i=1}^{2} {y_i}^2 \mathrm{I}_i^{\mathrm{III}} + {y_2}^2 \mathrm{I}_3^{\mathrm{III}}\right), \tag{11}$$

where for the assumed pdf given by (8), we derive

$$\begin{aligned} \mathrm{I}^{\mathrm{I}} &= 2\sum_{i=1}^{3}\int_{x_{i-1}}^{x_i} x^2 p(x)dx = \sigma^2, \\ \mathrm{I}_i^{\mathrm{II}} &= \int_{x_{i-1}}^{x_i} x p(x)dx = \frac{1}{2}\left[\left(x_{i-1}+\frac{\sigma}{\sqrt{2}}\right)\exp\left\{-\frac{\sqrt{2}\,x_{i-1}}{\sigma}\right\} - \left(x_i+\frac{\sigma}{\sqrt{2}}\right)\exp\left\{-\frac{\sqrt{2}\,x_i}{\sigma}\right\}\right], i = 1,2,3, \\ \mathrm{I}_i^{\mathrm{III}} &= \int_{x_{i-1}}^{x_i} p(x)dx = \frac{1}{2}\left[\exp\left\{-\frac{\sqrt{2}\,x_{i-1}}{\sigma}\right\} - \exp\left\{-\frac{\sqrt{2}\,x_i}{\sigma}\right\}\right], i = 1,2,3. \end{aligned} \tag{12}$$

The substitution of equations (13)–(15) in equation (12) yields

$$D^{UQ} = \sigma^2 + {y_1}^2 - \sqrt{2}\sigma y_1 + \left[(y_2 - y_1)(y_2 + y_1 - 2x_1 - \sqrt{2}\,\sigma)\right]\exp\left\{-\frac{\sqrt{2}\,x_1}{\sigma}\right\}. \tag{13}$$

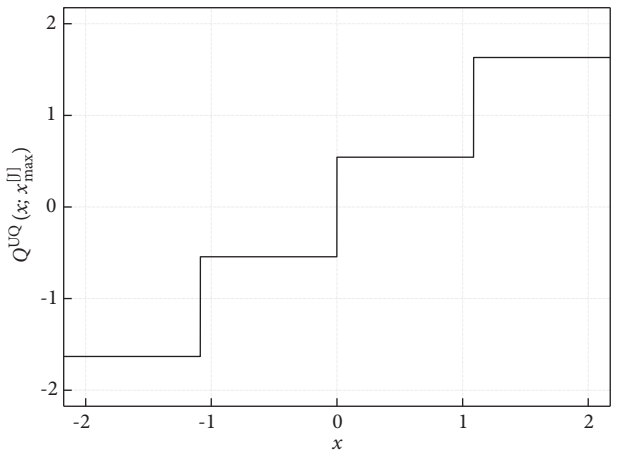

Figure 2: Transfer characteristic of the symmetric two-bits UQ, $Q^{UQ}(x; x_{max}^{[J]})$.

By taking into account (5)–(7), the total distortion of our symmetrical two-bits UQ for the Laplacian pdf of zero mean and arbitrary variance σ^2 as an input can be expressed as

$$D^{UQ} = \sigma^2 + \frac{x_{max}^2}{16} - \frac{\sqrt{2}}{4}\sigma x_{max}\left(1 + 2\exp\left\{-\frac{\sqrt{2}\,x_{max}}{2\sigma}\right\}\right). \tag{14}$$

As in numerous papers in the field of quantization (for instance, [4, 6–9, 11–14, 31, 40]), we are interested in conducting an analysis for the variance-matched case, where the variance of the input signal and the variance for which the quantizer is designed match. Therefore, we further assume the unit variance, $\sigma^2 = 1$, resulting in formula for the distortion of our symmetrical two-bits UQ with the input specified by the Laplacian pdf with zero mean and unit variance:

$$D^{UQ}|_{\sigma^2=1} = 1 + \frac{x_{max}^2}{16} - \frac{\sqrt{2}}{4}x_{max}\left(1 + 2\exp\left\{-\frac{\sqrt{2}\,x_{max}}{2}\right\}\right). \tag{15}$$

Let us further define the theoretical SQNR as

$$\mathrm{SQNR}_{th}^{UQ} = 10\log_{10}\left(\frac{\sigma^2}{D^{UQ}}\right), \tag{16}$$

which will be calculated for $\sigma^2 = 1$. Theoretically obtained values will be compared with the experimentally determined SQNR, obtained for the real weights of the trained NN, defined as

$$\mathrm{SQNR}_{ex}^{UQ} = 10\log_{10}\left(\frac{1/W\sum_{j=1}^{W} w_j^2}{1/W\sum_{j=1}^{W}\left(w_j - w_j^{UQ}\right)^2}\right), \tag{17}$$

where w_j, $j = 1, 2, \ldots, W$ denote the weights represented in FP32 precision, w_j^{UQ}, $j = 1, 2, \ldots, W$ are the weights quantized by our symmetrical two-bits UQ, and W is the total number of weights. We note that in order to distinguish between the theoretical and experimental results we use x for the signal/data being quantized and taken into calculation of the theoretical SQNR value, while we use w for the data (NN model weights) being quantized and taken into calculation of the experimental SQNR value.

Finally, we can define the quantization rule for our symmetrical two-bits uniform quantization of weights. Therefore, we define the transfer characteristics of our symmetrical two-bits UQ as

$$Q^{UQ}(x; x_{max}) = \begin{cases} \mathrm{sgn}(x)\left(\lfloor |x|/\Delta \rfloor + 1/2\right)\Delta, & |x| \le x_{max}, \\ \mathrm{sgn}(x)\left(x_{max} - \Delta/2\right), & |x| > x_{max}, \end{cases} \tag{18}$$

while by taking into account (7), we can eventually define the quantization rule for our symmetrical two-bits uniform quantization of weights as

$$w_j^{UQ} = \begin{cases} \mathrm{sgn}(w_j)\left(\left\lfloor \frac{2|w_j|}{x_{max}} + \frac{1}{2}\right\rfloor\right)\frac{x_{max}}{2}, & |w_j| \le x_{max}, \\ \mathrm{sgn}(w_j)\left(\frac{3x_{max}}{4}\right), & |w_j| > x_{max}. \end{cases} \tag{19}$$

By observing (18)–(22), one can anticipate that setting a suitable support region threshold value is crucial for achieving the best possible performance of the given quantization task. Even in this simple case of two-bits uniform quantization, x_{max} cannot be analytically determined to provide the minimal distortion. More precisely, x_{max} determined by Jayant [37] is a result of numerical distortion optimization, while Hui analytically obtained the following equation for x_{max} of symmetrical N-level asymptotically optimal UQ, designed for high bit rates and an input signal with the Laplacian pdf of zero mean and unit variance [6]:

$$x_{max}^{[H]} = \sqrt{2}\ln(N). \tag{20}$$

Let us highlight here that numerous papers have paved the frameworks on how to interpret the impact of the number of quantization levels and the support region widths on both distortions, the granular and the overload, and at the same time on the SQNR [6, 11–13, 18–20, 38]. The problem is that for the fixed number of quantization levels, with the decrease of the x_{max} value, the granular distortion is reduced at the expense of the overload distortion increase. Namely, the shrinkage of $\mathfrak{R}_g$ can cause a significant granular distortion reduction, while at the same time, it can result in an unwanted but expected increase of the overload distortion in $\mathfrak{R}_o$. Therefore, as highlighted in [40], tuning the value of x_{max} is one of the key challenges when heavy-tailed Laplacian pdf is taken into consideration. In the following sections, we will show that tuning the value of x_{max} is also of great importance in image classification tasks, as the one we describe in this paper.

Let us also specify the transfer characteristic and the quantization rule for our second novel quantizer model, that is LWUQ, composed of M UQs adjusted in terms of the support region threshold to the amplitude dynamic of weights at each of M layers:

$$Q^{\mathrm{UQ}_{L_i}}\left(x; x_{\max}^{L_i}\right) = \begin{cases} \operatorname{sgn}(x)\left(\left\lfloor \dfrac{2|x|}{x_{\max}^{L_i}} \right\rfloor + 1/2\right)\dfrac{x_{\max}^{L_i}}{2}, & |x| \le x_{\max}^{L_i} \\ & , L_i = 1, 2, \ldots, M, \\ \operatorname{sgn}(x)\left(\dfrac{3x_{\max}^{L_i}}{4}\right), & |x| > x_{\max}^{L_i} \end{cases}$$

$$w_j^{\mathrm{UQ}_{L_i}} = \begin{cases} \operatorname{sgn}\left(w_j\right)\left(\left\lfloor \dfrac{2\left|w_j\right|}{x_{\max}^{L_i}} \right\rfloor + \dfrac{1}{2}\right)\dfrac{x_{\max}^{L_i}}{2}, & \left|w_j\right| \le x_{\max}^{L_i} \\ & , L_i = 1, 2, ..., M. \\ \operatorname{sgn}\left(w_j\right)\left(\dfrac{3x_{\max}^{L_i}}{4}\right), & \left|w_j\right| > x_{\max}^{L_i} \end{cases} \tag{21}$$

The second quantizer model is addressed in the paper with the goal to determine whether an additional layer-wise adjustment of the support region can provide an improvement in terms of accuracy and SQNR. The difference between LWUQ and UQ lies only in the layer-wise adjustment of the support region threshold, so in the last two equations, we have $x_{\max}^{L_i}$, which denotes the support region threshold of the UQ adjusted to the amplitude dynamic of weights at each of M layers (L_i, $i = 1, 2, \ldots, M$) of our NN model. Accordingly, we can specify the experimental SQNR for each of M layers as

$$\mathrm{SQNR}_{ex}^{UQ_{L_i}} = 10\log_{10}\left(\frac{1/W_{L_i}\sum_{j=1}^{W_{L_i}} w_j^2}{1/W_{L_i}\sum_{j=1}^{W_{L_i}}\left(w_j - w_j^{\mathrm{UQ}_{L_i}}\right)^2}\right), \quad L_i = 1, 2, \ldots, M, \tag{22}$$

and for all layers as

$$\mathrm{SQNR}_{ex}^{LWUQ} = 10\log_{10}\left(\frac{1/M\sum_{L_i=1}^{M} 1/W_{L_i}\sum_{j=1}^{W_{L_i}} w_j^2}{1/M\sum_{L_i=1}^{M} 1/W_{L_i}\sum_{j=1}^{W_{L_i}}\left(w_j - w_j^{\mathrm{UQ}_{L_i}}\right)^2}\right), \tag{23}$$

where W_{L_i} denotes the number of weights of the layer L_i.

3. Post-Training Two-Bits Uniform Quantization

This section describes the practical application of our post-training quantization. As already mentioned and highlighted in [40], post-training quantization is especially convenient as there is no need for retraining the NN, while the memory size required for storing the NN model weights can be significantly reduced. We evaluate the performance of the compressed QNN model by comparing its accuracy before and after weight compression/quantization. As the originally trained weights are typically stored as 32-bit floating-point values, we have a large potential space for data compression by means of quantization. However, as already mentioned, the choice of the support region width ($\mathfrak{R}_g$ width) is the most sensitive in the uniform quantization of nonuniform sources [7–10], [14, 37, 38, 40], e.g., Laplacian sources, which is why this choice is the topic of our research. For the experimental evaluation of the post-training two-bits UQ performance and its adjusted layer-wise version, we have performed weight compression of a three-layer FC NN model (shortly, our NN model). The NN model has been trained on the MNIST dataset for handwritten digit recognition, consisting of 60,000 images of handwritten digits from 0 to 9 [41]. We have created the block diagram (see Figure 3) to depict our experimental evaluation process.

The NN model training and evaluation are implemented in the TensorFlow framework with Keras API, version 2.5.0 [42], while the code is implemented in Python programming language [43]. By default, TensorFlow stores all the variables with 32-bit precision, where each of 669,706 model parameters is represented and stored by using 32 bits. The main goal of our experiment is to provide the answer to the question whether it is possible to apply the simplest two-bits UQ to represent the trained NN model weights with such a smaller bit rate of $R = 2$ bit/sample while preserving the accuracy of the model to a great extent. Specifically, we examine the performance of two-bits UQ for a few significant cases of its design. Our goal is also to derive conclusions on how the key parameter of two-bits UQ affects its performance and QNN model performance. Moreover, we extend our analysis to the layer-wise two-bits UQ case to examine whether further performance improvements of the utilized two-bits uniform quantization are possible.

The first step in the process shown in Figure 3 is to load the training and testing data into the framework. The

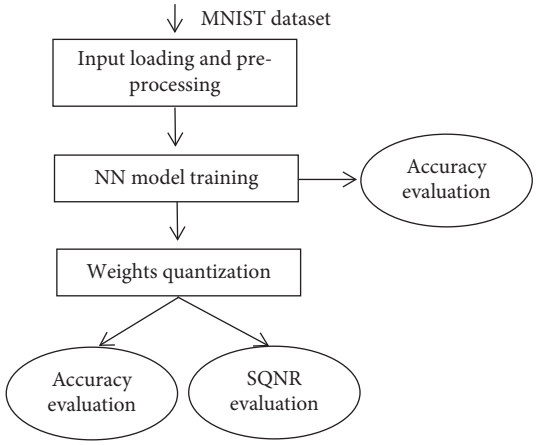

Figure 3: Block diagram for the experimental evaluation of the post-training quantization performance.

MNIST dataset consists of 60,000 training and 10,000 testing black and white images, with pixel values in the range of [0–255]. Each image has the dimensions of 28×28 pixels [41]. After loading, the images have been reshaped (flattened) into 1-dimensional vectors of 784 ($28 * 28$) elements, which is the shape accepted by our NN. Additionally, the inputs have been normalized to the range [0–1], by dividing each of the image samples with the maximum possible amplitude value of 255. Finally, the training and validation sets are prepared to be loaded into the NN model for training.

We have created a flowchart, as shown in Figure 4, to present the block diagram of our FC NN model. The NN model consists of $M = 3$ fully connected (dense) layers, with 2 hidden and 1 output layer. Both hidden layers consist of 512 nodes, where the first one accepts the input shape of (784,). Hidden layers are followed by the rectified linear unit (ReLu) activation, which directly forwards positive activation values, while setting the negative values to zero [44]. The activation function is followed by a dropout regularization with a factor of 0.2, which randomly sets outputs of 20% of the total nodes in the layer to 0, which helps preventing the network to overfit to the training examples. The output of the second hidden layer is fed to the output layer with 10 nodes, which matches the number of target classes. The dense layer is followed by the SoftMax activation, which outputs probabilities that input belongs to any of 10 possible classes. In total, the NN model consists of 669,706 trainable parameters, which are quantized by utilizing the two-bits UQ after the training is completed. The training has been conducted as in [40], in 10 epochs, with the batch size of 128, which resulted in 469 iterations per epoch to complete the training over all the training examples. The validation set accuracy after training amounts to 0.981, meaning that the NN model made accurate predictions for 98.1% of the images in the validation set. The trained weights have been quantized by using the two-bits UQ, and our NN model performance has been evaluated with compressed/quantized weights in order to represent the post-training two-bits UQ performance, and the post-training performance of additionally adapted layer-wise two-bits UQ.

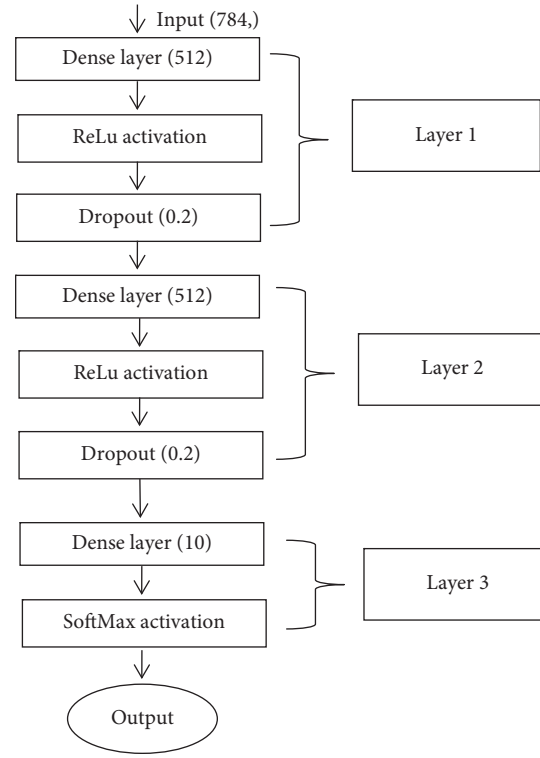

Figure 4: Three-layer FC NN model for handwritten digit recognition.

The first step in processing the trained weights of the NN model is applying normalization. Before being fed into the quantizer, weight values are normalized to a mean value equal to 0 and a standard deviation equal to 1, similarly as presented in [40, 45, 46]. Let us highlight that the weight normalization has been introduced in [46] to replace the batch normalization. Although both normalization methods are reparameterization methods, they are different from each other in that the object of batch normalization is the input of NN and that of the weight normalization is the weight vector [45]. The performance of these two normalization methods has been compared in [45], and it has been shown that the convolutional NN with weight normalization has a better performance than that with batch normalization. In particular, it has been shown that the weight normalization has a smaller loss between the construction pictures and original pictures.

The weight normalization process adjusts the weight's distribution making quantizer's input more suitable, while the reversibility of the process, that is, denormalization after quantization, assures us that we do not introduce additional errors into the process. In other words, the weight normalization is being reversed after the quantization process and before loading the quantized weights back into the model so that quantized weights could go back to the original range of values. After the quantized weights are loaded back into the model, we can evaluate the performance of both two-bits UQ and the QNN model's accuracy. Similarly, we can evaluate the performance of LWUQ and the corresponding compressed NN model's accuracy.

4. Numerical Results and Analysis

In this section, we analyse the accuracy of the presented NN model after applying uniform quantization to its weights (compressed NN model), where UQ is designed for the bit rate of $R = 2$ bit/sample. As already mentioned, with the default precision of FP32, the NN achieves the accuracy of 98.1% on the MNIST validation dataset. Naturally, compressing the weight representations from 32 to 2 bits per sample unavoidably degrades the accuracy of our NN model. To minimize the amount of degradation, we need to fully utilize the available bit rate and keep the quantized weights as close as possible to the original ones. This intuitively indicates that when applying quantization, we need to obtain a high SQNR value to preserve the accuracy of our NN model. In contrast, as NN model weights start from a random distribution and do not have the exact and precise end value, our anticipation is that in the observed case, where only two bits are available, SQNR and accuracy do not necessarily have to be highly related. The previous statement is examined and confirmed in the experimental application process, which is described in the following study.

As mentioned, the performance of the quantizer is observed by assessing its SQNR value in dB, while the performance of the QNN model is evaluated by its accuracy achieved on the validation set. Since, to the best of the authors' knowledge, there is not a direct relation between SQNR and accuracy, we need to compromise and find the best balance between the two. Assuming so, we intuitively anticipated that the choice of the support region width ($\mathfrak{R}_g$ width) would have a high impact on the accuracy of the model. To confirm the intuition, we have analysed multiple choices of $\mathfrak{R}_g$ width, that is, we have designed our two-bits UQ for different $\mathfrak{R}_g$. Among many of the observed cases, we have selected to present 4 different and the most intriguing cases of two-bits UQ design, with the numerical results presented in Table 1. The observed cases differ in the choice of $\mathfrak{R}_g$ width of the implemented quantizer, which highly impacts the performance of both, the QNN model and our two-bits UQ. However, completely different conclusions regarding the accuracy of QNN have been derived in [40]. In particular, the problem of selecting the support region of UQ to compress the weights of the trained neural network is more prominent in the two-bits case, when compared to the three-bits case addressed in [40], which makes the analysis for the two-bits case presented in this paper particularly intriguing. Namely, in this paper, we show that the performance of the QNN model trained on the MNIST is highly dependent on the support region choice. However, in [40], a weak influence of the support region choice on the performance of the QNN model has been ascertained, while a slightly higher accuracy preservation of the original NN model has been achieved, which amounts to about 1% compared to the QNN model we address here. We should also highlight once more that in this paper, we provide 16 times the compression of the originally trained weights, while the same weights stored in the FP32 format are compressed less (about 10 times) in [40]. Therefore, this justifies and explains a slightly lower accuracy preservation achieved here in comparison to the QNN from [40]. In the following study, we show that with the layer-wise adaptation of the two-bits UQ, the performance of the QNN model can be additionally improved.

The layer-wise adaptation of $\mathfrak{R}_g$ is also performed, and the results are presented in Table 2. Specifically, two-bits LWUQ is applied in NN weight compression, where $\mathfrak{R}_g$ is defined according to the statistics of each of the layer weights. The application of LWUQ is considered only for Cases 1 and 2; as in these two cases, we determine $\mathfrak{R}_g$ of the quantizer according to the statistics of the weights. We note that it can be considered that the design of the quantizer in question in Cases 3 and 4 depends only on the bit rate, which in our analysis amounts to $R = 2$ bit/sample. By adapting quantization to the individual layers, we are able to observe the global and local information of the weights, similarly as implemented in [47]. Moreover, we can pay extra attention to both high and low weight values, which might not be properly quantized by utilizing a single quantizer for the whole NN. Therefore, we introduce the local quantization of an individual layer, while utilizing global statistical parameters as in [48], with the purpose of reducing processing time and preventing an increase in bit rate.

In the first two cases, two-bits UQ is designed according to the statistical information of the original model weight values. By observing the statistics of the trained NN weights, we have found that the minimum and maximum weight values in the original 32-bit representation amount to $x_{\min} = -7.063787$ and $x_{\max} = 4.8371024$. In contrast, for comparison purposes, Cases 3 and 4 implement the support region threshold values from the literature [6, 37] determined for $R = 2$ bit/sample, which are not dependent on the input weight statistics as it is in Cases 1 and 2. Along with the experimentally obtained SQNR values, calculated by using (16), Table 1 also presents theoretical SQNR values, calculated from (14) and (15), for our two-bits UQ designed for different $\mathfrak{R}_g$ widths and Laplacian pdf of zero mean and unit variance. The first case represents two-bits UQ symmetrically designed for the maximum amplitude of our trained model weights so that the support region is determined as $[-x_{\max}, x_{\max}]$, which in this case amounts to the range of $[-4.8371024, 4.8371024]$. This implies that Case 1 utilizes a wider $\mathfrak{R}_g$ when compared to Cases 3 and 4, as it includes 99.988% of the weight values. In other words, there is only 0.012% of the weight values outside $\mathfrak{R}_g$, meaning that they are within the remaining range of $[-7.063787, -4.8371024]$. As this relatively large range holds such a small percentage of input values, we can safely assume that designing $\mathfrak{R}_g$ to cover the whole range of weights values would not contribute to the increase of the accuracy and SQNR. Among the observed cases, QNN presented in Case 1 achieves the highest accuracy during validation, equal to 96.97%, indicating that by implementing two-bits UQ, we degrade the accuracy of our NN model by only 1.13%. Considering that the aforementioned accuracy is achieved while using only 2 bits per sample to represent the NN model weights, we can conclude that this result is of great significance. Although our proposal achieves the highest accuracy in Case 1, the obtained SQNR value of 2.8821 dB is fairly small compared

Table 1: SQNR and model accuracy for application of different two-bits UQ designs.

Two-bits UQ, accuracy (FP32) = 98.1%				
x_{min} = -7.063787, x_{max} = 4.8371024 $x_{max}^{(H)}$ = 1.9605, $x_{max}^{(J)}$ = 2.1748	Case 1 $\mathfrak{R}_g$ $[-x_{max}, x_{max}]$	Case 2 $\mathfrak{R}_g$ $[x_{min}, -x_{min}]$	Case 3 $\mathfrak{R}_g$ $[-x_{max}^{(H)}, x_{max}^{(H)}]$	Case 4 $\mathfrak{R}_g$ $[-x_{max}^{(J)}, x_{max}^{(J)}]$
$SQNR_{ex}^{UQ}$ (dB)	2.8821	−1.2402	**8.7676**	8.7639
$SQNR_{th}^{UQ}$ (dB)	1.9360	−2.0066	6.9787	**7.0707**
Accuracy (%)	**96.97**	94.58	96.34	96.74
Within $\mathfrak{R}_g$ (%)	99.988	100	94.787	96.691

Table 2: SQNR and model accuracy for application of different two-bits LWUQ designs.

Two-bits LWUQ, accuracy (FP32) = 98.1%		
	Case 1 $\mathfrak{R}_g$	Case 2$\mathfrak{R}_g$
$(x_{max}^{L1}, x_{max}^{L2}, x_{max}^{L3}) = (4.5150, 4.8371, 3.6784)$ $(x_{min}^{L1}, x_{min}^{L2}, x_{min}^{L3}) = (-7.0638, -5.4354, -6.1979)$	$[-x_{max}^{L1}, x_{max}^{L1}]$ $[-x_{max}^{L2}, x_{max}^{L2}]$ $[-x_{max}^{L3}, x_{max}^{L3}]$	$[x_{min}^{L1}, -x_{min}^{L1}]$ $[x_{min}^{L2}, -x_{min}^{L2}]$ $[x_{min}^{L3}, -x_{min}^{L3}]$
$SQNR_{ex}^{UQ\ L1}$ (dB)	3.1340	−1.7588
$SQNR_{ex}^{UQ\ L2}$ (dB)	3.4507	2.2826
$SQNR_{ex}^{UQ\ L3}$ (dB)	8.3642	4.6137
$SQNR_{ex}^{LWUQ}$ (dB)	**3.3145**	−0.374
Accuracy (%)	**97.26**	93.55

to Cases 3 and 4. This confirms the premise that SQNR and accuracy do not necessarily have to be highly related and that the accuracy of the QNN depends on the proper choice and definition of the support region of the quantizer in question.

One can notice that Case 2 is the least favourable one in terms of SQNR and the model's accuracy. This is due to the choice of overly wide $\mathfrak{R}_g$ of the two-bits UQ, designed to cover the range of $[-x_{min}, x_{min}]$, which amounts to [−7.063787, 7.063787]. Compared to Case 1, $\mathfrak{R}_g$ in Case 2 includes 100% of the weight samples, while including an extra portion of the range in which no weight sample falls to [4.8371024, 7.063787], which makes it unnecessarily wide. In addition, as the model weights tend to have Laplacian distribution (see Figure 5), most of the weight samples are symmetrically concentrated around zero, which makes the middle of our support region the most important part for quantization. In other words, by adjusting the width of $\mathfrak{R}_g$ to match the full range of weight values, we sacrifice the accuracy in the area of $\mathfrak{R}_g$ where most of the weight values lie in. This leads to a very large distortion introduced by the quantizer in question and to SQNR with even a negative value of −1.2402 dB. This is followed by the theoretical analysis as well, with the theoretical SQNR equal to about −2 dB. Since the accuracy of our NN model in Case 2 is the lowest one observed (94.58%), we can conclude that designing $\mathfrak{R}_g$ of the quantizer in question to include all the weight samples, that is to cover the range $[-x_{min}, x_{min}]$, as it was assumed, for instance in [21, 29], is not a suitable solution for post-training two-bits uniform quantization of weights for our targeted application. However, a completely opposite conclusion has been derived in [40] for Case 2, where an additional bit has been utilized in the uniform quantizer design. Specifically, in [40], we have determined equal accuracies of the QNN in Cases 1 and 2 and concluded that the choice of the support region threshold does not have a strong impact on the performance of the QNN. This again proves that conclusions about the performance of QNN for low bit rates (2 bits and 3 bits) should be carefully derived and should be addressed as particular research questions, where results must be compared to derive valid conclusions and to enable tracing further research directions in this intriguing research filed.

Cases 3 and 4 utilize the support region threshold values from the literature, defined by Hui [6] and Jayant [37], respectively. Case 3 utilizes analytically obtained equation (19) for the support region threshold of symmetrical N-level asymptotically optimal UQ, designed for an input signal with the Laplacian pdf of zero mean and unit variance. As we utilize the two-bits quantizer, the number of representation levels ($N = 2^R$) amounts to $N = 4$, and the support region threshold defined by (19) is equal to $x_{max}^{[H]} = 1.9605$. One can notice that the defined $\mathfrak{R}_g$ [−1.9605, 1.9605] is thus much narrower compared to the previously defined ones as it covers 94.787% of all the weight samples. This makes a positive impact on the obtained SQNR performance, which reaches its maximum in this case and amounts to 8.7676 dB. The accuracy of the compressed NN model in this case is also preserved in a great manner with 96.34%, resulting in a degradation of 0.63%, compared to Case 1 and 1.76% compared to the original model weights. Finally, Case 3 also confirms that SQNR and model accuracy are not highly related, as we get the highest experimental SQNR value (the highest values are bolded), while accuracy is lower than in Cases 1 and 4. Case 4 has similar performance characteristics as the previous one, while implementing slightly wider $\mathfrak{R}_g$, with the support limit threshold value of $x_{max}[J] = 2.1748$, specified in [37]. This value is determined for the UQ designed by means of numerical optimization for the Laplacian pdf with zero mean value and unit variance. In this case, $\mathfrak{R}_g$ includes 96.691% of the weight values, providing the experimental SQNR value of 8.7639 dB. The accuracy of the NN model is slightly better compared to Case 3, with

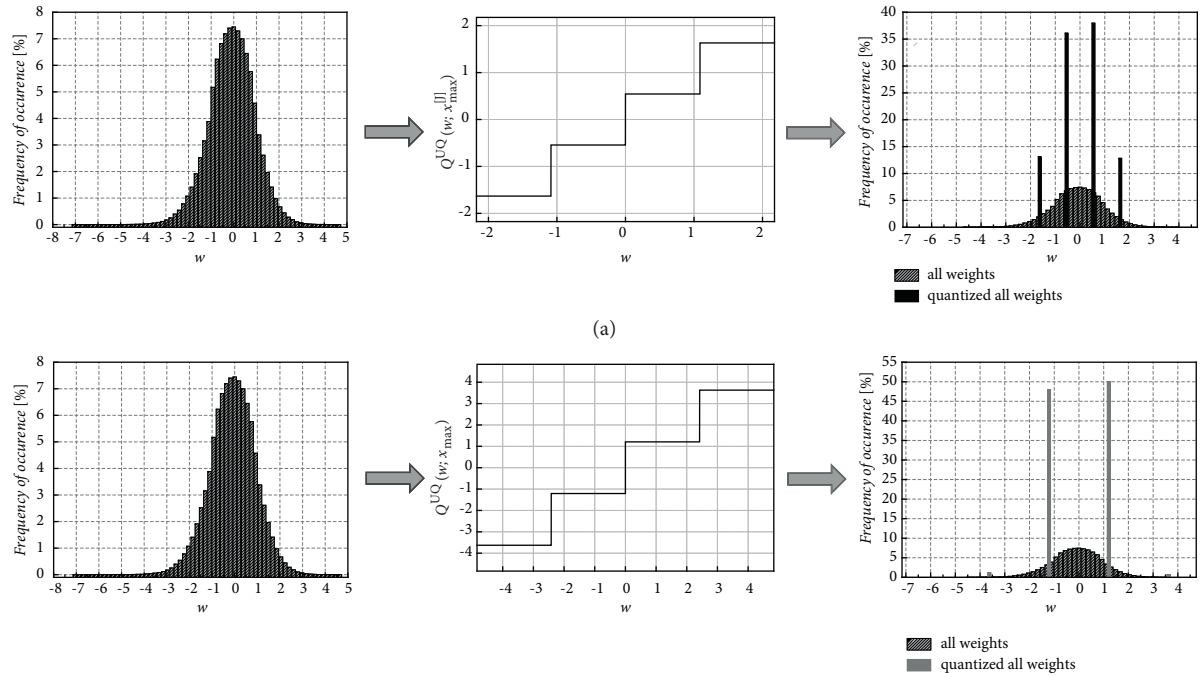

Figure 5: Normalized histogram of all weights (FP32); transfer characteristic of the symmetric two-bits UQ; merged normalized histogram of weights FP32 and uniformly quantized (a) Case 4 ($x_{max}^{[J]} = 2.1748$) and (b) Case 1 ($x_{max} = 4.8371024$).

96.74%, while SQNR values are almost identical. The accuracy is still being lower than the one obtained in Case 1, by a margin of 0.23%.

In Case 3, it is interesting to notice that experimental SQNR is greater than the theoretically obtained one by a margin of 1.7889 dB. As in Case 3, the theoretically determined SQNR value in Case 4 is lower than the experimental one by a margin of around 1.8 dB, due to the similar quantizer design in these two cases. The reason is that in the experimental analysis, the weights from a limited set of possible values [−7.063787, 4.8371024] are quantized; while in the theoretical analysis, the quantization of values from an infinite set of values from the Laplacian source is assumed, resulting in an increase of distortion, that is in the decrease of the theoretical SQNR value. To confirm this conclusion, one can notice that the wider the $\mathfrak{R}_g$, the smaller the deviation. Thus, the smallest deviation of theoretical and experimental results can be observed in Case 2. It is possible to minimize the deviation by scaling the $\mathfrak{R}_g$ with a certain constant, as performed in [31], where a constant is introduced to scale $\mathfrak{R}_g$ of the quantizer utilized in obtaining the experimental results. We chose not to perform the adaptation of the $\mathfrak{R}_g$, as we analyse the importance of a proper choice of $\mathfrak{R}_g$ and its impact on both, the QNN model's accuracy and the obtained SQNR.

To further improve the performance of both, quantizer and NN model, we consider LWUQ, whose performance is presented in Table 2. By adapting $\mathfrak{R}_g$ width of the quantizer for each layer, we expectedly increase the obtained SQNR value. In contrast, QNN model performance benefits the layer-wise adaptation only in Case 1, where the accuracy is increased by 0.29%, and a significant increase in SQNR is provided. It should be pointed out that the same bit rate and quantizer are used, with the difference lying only in applying the adaptation of quantizer's $\mathfrak{R}_g$ for each layer. Case 2 is already presented as an example for an unfavourable method of choosing quantizer's $\mathfrak{R}_g$ so that the layer-wise adaptation even decreased the model performance (accuracy but not SQNR) in this case. This can be considered as a form of error multiplication, as we apply rough quantization to the most significant area of the weight distribution, which is further emphasized by the layer-wise adaptation. The performance of a quantizer can be also assessed by observing the histograms of the quantized weights, which is performed for both, UQ and LWUQ, and is presented in the following study.

Figure 5 presents normalized histograms of all weights in full precision (FP32) with symmetric two-bits uniform quantizer characteristics and a merged normalized histogram of weights in FP32 precision with its uniformly quantized counterparts for Cases 1 and 4. Let us observe and compare the merged histograms in both cases as they display the most important information about the quantization process. One can notice that in Case 4 (the top image), we utilize all the representation levels in a large amount; while in Case 1, we mostly utilize only 2 representation levels $\pm y_1$. From the histograms, we have determined that in Case 1, 98.0657% of all the weights are represented by $\pm y_1$, while in Case 4 that percentage amounts to 74.0703%. Therefore, in Case 1, statistically, the most important part of the input data, located around zero, is roughly quantized by utilizing only 2 representation levels; while in Case 4, this is done by utilizing all 4 available representation levels distributed in

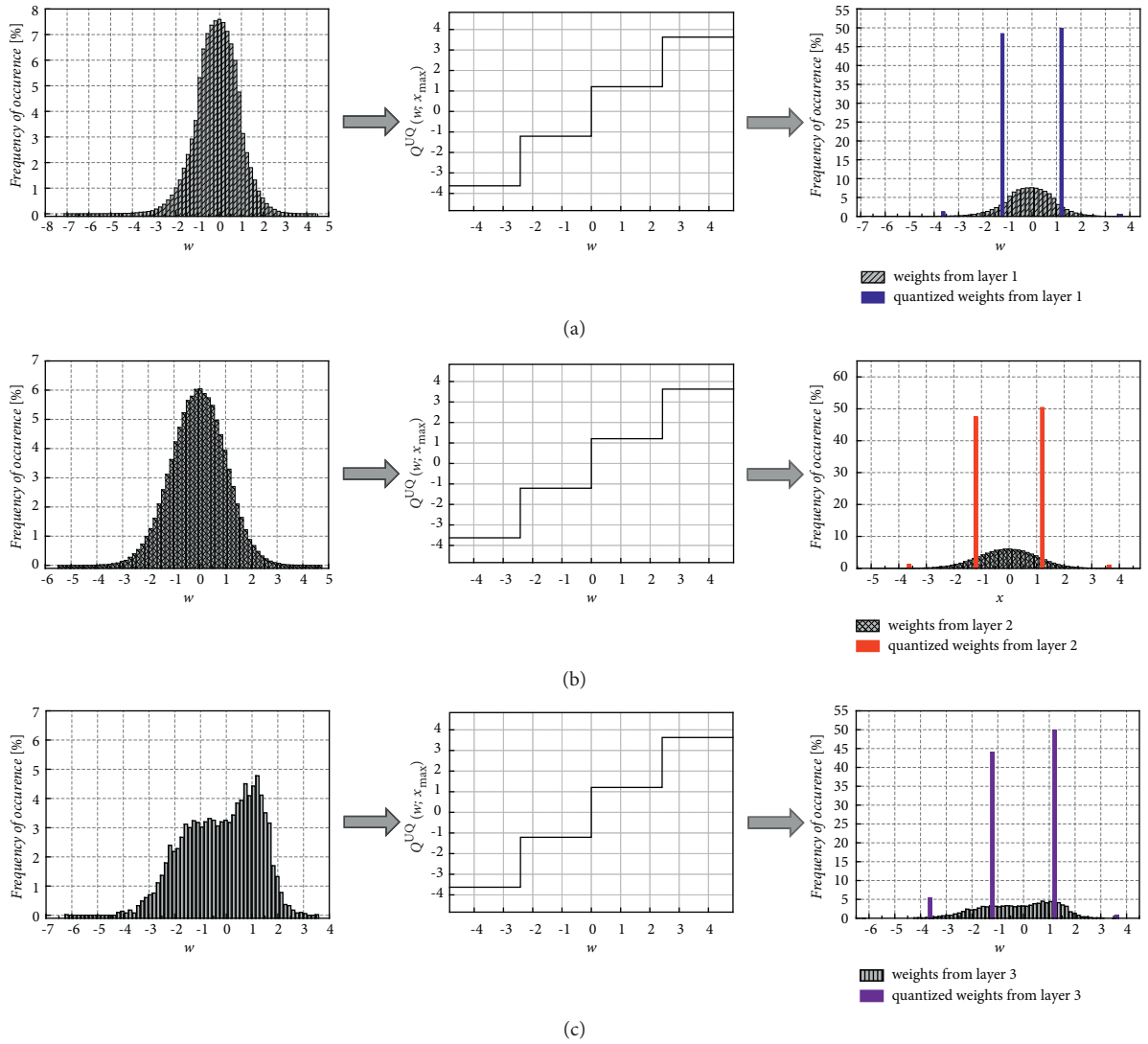

FIGURE 6: Normalized histogram of weights (FP32) from (a) layer 1, (b) layer 2, and (c) layer 3; transfer characteristic of the symmetric two-bits UQ for Case 1; normalized histogram of FP32 and uniformly quantized weights from (a) layer 1, (b) layer 2, and (c) layer 3.

the narrower $\mathfrak{R}_g$. Finally, presented histograms confirm and explain why in Case 4 we obtain much higher SQNR values of 8.7639 dB, compared to Case 1, where the obtained SQNR amounts to only 2.8821 dB. Interestingly, Case 1 provides the highest QNN model accuracy, confirming that accuracy does not solely rely on the SQNR value but on the wise choice of the $\mathfrak{R}_g$ of the quantizer, as in Case 1.

Figure 6 presents the layer-wise analysis of the NN model weights after quantization, as well as the transfer characteristics of the two-bits UQ applied to each layer of our NN model. Transfer characteristics of the quantizers are identical, as the two-bits UQ is not adapted to the individual layers. By observing FP32 layer-wise histograms on the left, one can notice that NN model weights have slightly different probability distributions among different layers. This finding is in line with our intuition that adapting quantization according to the individual layer statistics would in fact increase SQNR obtained after quantization. The idea of adapting quantization to individual layers of the NN model is exploited and presented in Figure 7. One can notice that in this case, along with different distributions of the weight samples among the layers, we have different transfer characteristics of the uniform quantizers applied to different layers, which defines the previously analysed LWUQ. As the presented choice of $\mathfrak{R}_g$ is Case 1, large exploitation of just 2 representation levels $\pm y1$ is still present, as shown in colour on the rightmost histograms, while the SQNR is increased compared to only applying two-bits UQ without layer-wise adaptation, as shown in Table 2.

Finally, we compare the accuracy achieved in our favourable case of the two-bits UQ, denoted by Case 1, to the similar complexity and comparable post-training solutions found in the literature (see Table 3). The accuracy achieved by utilizing our post-training solution is obviously higher compared to the ones presented in [15, 20, 21, 31]. In [15], the bit rate is only 1 bit per sample. In [20, 21], two-bits UQ is utilized, with the difference in the definition of the quantization step size ($\Delta = 2\ x_{max}/(N-1)$) and in the $\mathfrak{R}_g$

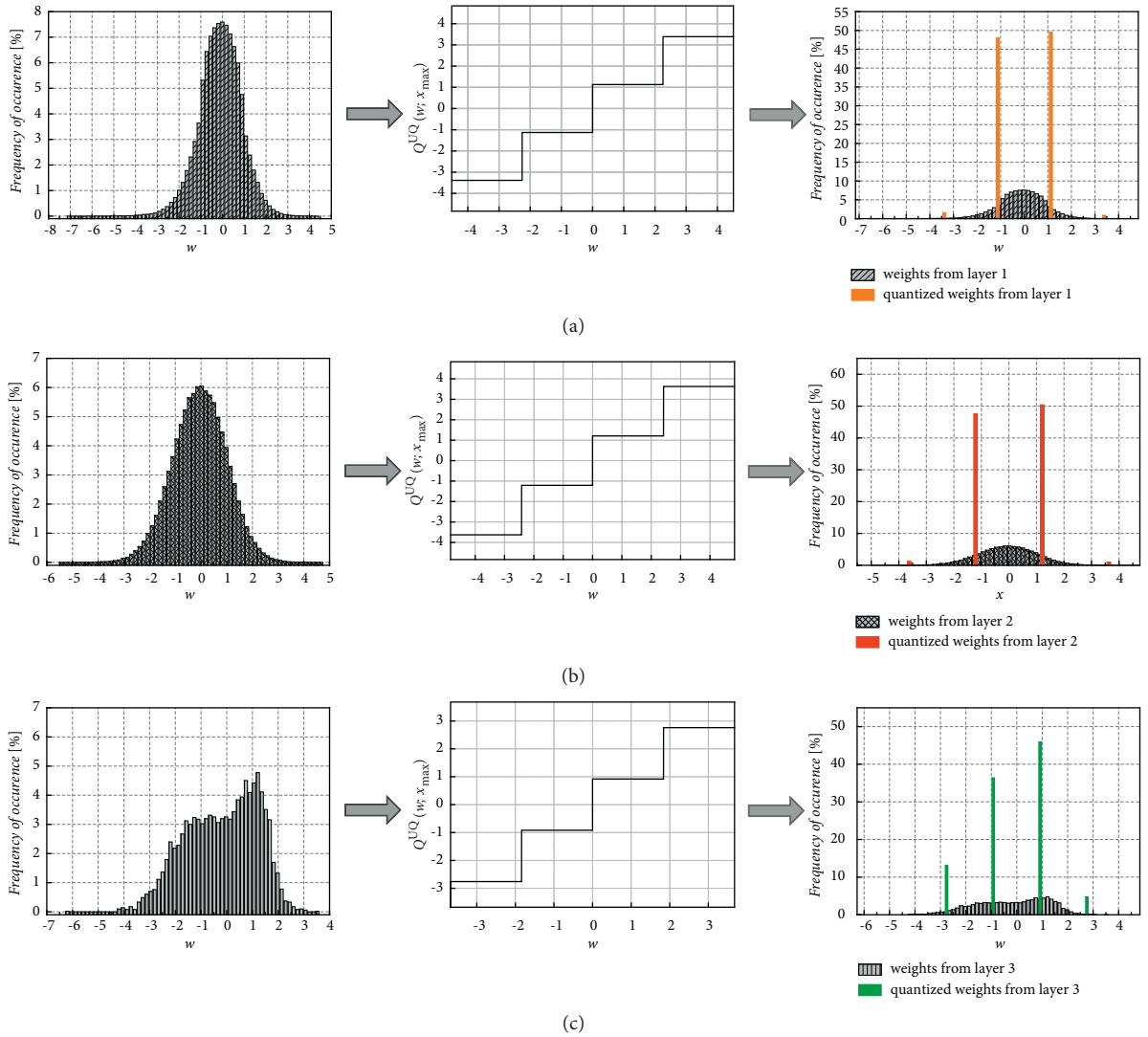

Figure 7: Normalized histogram of weights (FP32) from (a) layer 1, (b) layer 2, and (c) layer 3; layer-adapted transfer characteristic of the symmetric two-bits UQ for Case 1 and (a) layer 1, (b) layer 2, and (c) layer 3; layer-wise FP32 and uniformly quantized weights from (a) layer 1, (b) layer 2, and (c) layer 3.

Table 3: Accuracy comparison of compressed NN models.

	1-bit [15]	2-bits UQ [20]	2-bits UQ [21]	2-bits adaptive UQ [31]	Our Case 1	Our layer-wise Case 1
Accuracy (%)	91.12	94.70	94.49	96.26	96.97	97.26

choice with the quantization model we analyse here. The closest case for comparison is the one presented in [31], where an adaptive two-bits UQ is presented. Our favourable quantizer, denoted by Case 1, provided a higher accuracy of the compressed NN model on the validation set, by a margin of 0.71% and by a margin of 1% in the layer-wise case. Let us highlight that in [31], the full precision accuracy is 96.86%, while the full precision accuracy of our NN model is 98.1%. As the depth of our NN is greater than that of MLP from [31], it is not surprising that the accuracy of our NN model with FP32 parameters is higher. However, our model has a significantly greater number of parameters for quantization, compared to the one from [31], which provides the opportunity of even further degrading the accuracy. Moreover, this once more confirms the importance of the $\mathfrak{R}_g$ choice.

According to the presented results obtained for the considered research framework, one can conclude that the accuracy of the compressed NN model greatly depends on the choice of the support region of the quantizer in question. Additionally, it has been confirmed that SQNR and compressed NN model's accuracy are not highly related, as the performance of the quantizer alone does not affect the

accuracy degradation the most. Finally, by using two-bits UQ to compress the NN model's weights, we have managed to significantly preserve the accuracy of the NN model, which is degraded by 1.13%, while the weights are represented with a 16 times lower bit rate. Eventually, we have ascertained that two-bits LWUQ can provide an additional improvement of the accuracy of our QNN model for the MNIST dataset.

5. Summary and Conclusions

In this paper, we have shown that even when aggressive two-bits uniform quantization is utilized for post-training quantization, the accuracy of NN that we have built can be slightly degraded if we thoughtfully select the width of the quantizer's support region. The impact on the accuracy has been reported relative to the accuracy of the trained NN model with weights represented in the FP32 format. We have intuitively anticipated that the choice of the support region width ($\mathfrak{R}_g$ width) would have a high impact on the NN model's accuracy. To confirm the intuition, we have analysed several choices of $\mathfrak{R}_g$ width, that is, we have designed our two-bits UQ for different $\mathfrak{R}_g$. We have uncovered deficiencies of one common choice of the support region, and we have driven a firm conclusion about the importance of such a choice for the application described within the framework of this paper. One interesting conclusion derived for the given framework of the paper refers to the fact that SQNR and compressed NN model's accuracy are not highly related and that the performance of the quantizer alone does not dominantly affect the accuracy degradation. We have practically shown that the accuracy of the compressed NN model is very dependent on a proper choice and definition of the support region and the step size of the quantizer in question. Similar analyses and conclusions can be easily carried out for some other datasets since it is well known that weights in many NNs follow the Laplacian distribution, which we assumed in our analysis. We have shown that by using two-bits UQ to compress the NN model's weights, we have managed to significantly preserve the accuracy of the NN model, which is degraded by 1.13%, while the weights are represented with 16 times lower bit rate. Moreover, we have ascertained that layer-wise two-bits uniform quantization can provide an additional improvement of the accuracy of our compressed NN model for the MNIST dataset. The simplicity of our proposal along with a great performance in post-training quantization indicates that it can be practically exploited, with great importance in edge computing devices for real-time classification tasks. Moreover, the research presented in this paper opens a large space for future work and improvements. While performing the layer-wise adaptation, we have noticed that the distributions of weights vary across different layers of the NN model, whereas in the case of our NN model, the 3rd layer shows the greatest deviation from the Laplacian distribution. This opens a possibility to perform layer-wise adaptation, not only of $\mathfrak{R}_g$ but also of the probability density function used to model the NN model weights across different layers. While the step size of a uniform quantizer is purely defined by $\mathfrak{R}_g$, such an adaptation could be especially beneficial when utilizing nonuniform quantization, where we can adjust the step size according to the input data distribution.

Data Availability

The data used to support the findings of this study are available at http://yann.lecun.com/exdb/mnist/.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This research was supported by the Science Fund of the Republic of Serbia, 6527104, AI-Com-in-AI.

References

[1] A. Gholami, S. Kim, Z. Dong, Z. Yao, M. W. Mahoney, and K. Keutzer, "A Survey of Quantization Methods for Efficient Neural Network Inference," 2021, https://arxiv.org/abs/2103.13630.

[2] D. Liu, H. Kong, X. Luo, W. Liu, and R. Subramaniam, "Bringing AI to Edge: From Deep Learning's Perspective," 2020, https://arxiv.org/abs/2011.14808.

[3] F. Tung and G. Mori, "Deep neural network compression by in-parallel pruning-quantization," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 3, pp. 568–579, 2020.

[4] D. Lin, S. Talathi, and S. Annapureddy, "Fixed point quantization of deep convolutional networks," in *Proceedings of the 33rd International Conference on Machine Learning*, vol. 48, pp. 2849–2858, New York, NY, USA, June 2016.

[5] J. Rhee and S. Na, "On the characteristics of MSE-optimal symmetric scalar quantizers for the generalized gamma, bucklew-gallagher, and hui-neuhoff sources," *The Journal of Korean Institute of Communications and Information Sciences*, vol. 40, no. 7, pp. 1217–1233, 2015.

[6] D. Hui and D. L. Neuhoff, "Asymptotic analysis of optimal fixed-rate uniform scalar quantization," *IEEE Transactions on Information Theory*, vol. 47, no. 3, pp. 957–977, 2001.

[7] J. Lee and S. Na, "A rigorous revisit to the partial distortion theorem in the case of a Laplacian source," *IEEE Communications Letters*, vol. 21, no. 12, pp. 2554–2557, 2017.

[8] S. Na and D. L. Neuhoff, "On the convexity of the MSE distortion of symmetric uniform scalar quantization," *IEEE Transactions on Information Theory*, vol. 64, no. 4, pp. 2626–2638, 2018.

[9] S. Na and D. L. Neuhoff, "Monotonicity of step sizes of MSE-optimal symmetric uniform scalar quantizers," *IEEE Transactions on Information Theory*, vol. 65, no. 3, pp. 1782–1792, 2019.

[10] S. Na and D. L. Neuhoff, "Asymptotic MSE distortion of mismatched uniform scalar quantization," *IEEE Transactions on Information Theory*, vol. 58, no. 5, pp. 3169–3181, 2012.

[11] Z. Perić, M. Petković, and J. Nikolić, "Optimization of multiple region quantizer for Laplacian source," *Digital Signal Processing*, vol. 27, no. 15, pp. 150–158, 2014.

[12] Z. Perić and J. Nikolić, "High-quality Laplacian source quantisation using a combination of restricted and unrestricted logarithmic quantisers," *IET Signal Processing*, vol. 6, no. 7, pp. 633–640, 2012.

[13] Z. Perić, J. Nikolić, D. Aleksić, and A. Perić, "Symmetric quantile quantizer parameterization for the Laplacian source: qualification for contemporary quantization solutions," *Mathematical Problems in Engineering*, vol. 2021, Article ID 6647135, 12 pages, 2021.

[14] S. Na and D. Neuhoff, "On the Support of MSE-optimal, fixed-rate, scalar quantizers," *IEEE Transactions on Information Theory*, vol. 47, no. 7, pp. 2972–2982, 2001.

[15] Z. Perić, B. Denić, M. Savić, and V. Despotović, "Design and analysis of binary scalar quantizer of Laplacian source with applications," *Information*, vol. 11, no. 11, 2020.

[16] Z. H. Perić, B. D. Denić, M. S. Savić, N. J. Vučić, and N. B. Simić, "Binary quantization analysis of neural networks weights on MNIST dataset," *Elektronika Ir Elektrotechnika*, vol. 27, no. 4, pp. 55–61, 2021.

[17] P. Pham, J. A. Abraham, and J. Chung, "Training multi-bit quantized and binarized networks with a learnable symmetric quantizer," *IEEE Access*, vol. 9, Article ID 47194, 2021.

[18] R. Banner, Y. Nahshan, and D. Soudry, "Postraining 4-bit quantization of convolutional networks for rapid-deployment," in *Proceedings of the 33rd Conference on Neural Information Processing Systems (NeurIPS 2019)*, pp. 7948–7956, Vancouver, Canada, 2019.

[19] J. Choi, S. Venkataramani, V. Srinivasan, K. Gopalakrishnan, Z. Wang, and P. Chuang, "Accurate and efficient 2-bit quantized neural networks," in *Proceedings of the 2nd SysML Conference*, Stanford, CA, USA, March 2019.

[20] Y. Bhalgat, J. Lee, M. Nagel, T. Blankevoort, and N. Kwak, "LSQ+: improving low-bit quantization through learnable offsets and better initialization," in *Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, Seattle, WA, USA, June 2020.

[21] I. Hubara, M. Courbariaux, D. Soudry, R. El-Yaniv, and Y. Bengio, "Quantized neural networks: training neural networks with low precision weights and activations," *Journal of Machine Learning Research*, vol. 18, no. 187, pp. 6869–6898, 2018.

[22] C. Baskin, N. Liss, E. Schwartz et al., "UNIQ: Uniform noise injection for non-uniform quantization of neural networks," *ACM Transaction on Computer System*, vol. 37, pp. 1–15, 2021.

[23] T. Simons and D.-J. Lee, "A review of binarized neural networks," *Electronics*, vol. 8, no. 6, p. 661, 2019.

[24] H. Qin, R. Gong, X. Liu, X. Bai, J. Song, and N. Sebe, "Binary neural networks: a survey," *Pattern Recognition*, vol. 105, Article ID 107281, 2020.

[25] Y. Li, Y. Bao, and W. Chen, "Fixed-sign binary neural network: an efficient design of neural network for Internet-of-Things devices," *IEEE Access*, vol. 8, Article ID 164858, 2020.

[26] W. Zhao, T. Ma, X. Gong, B. Zhang, and D. Doermann, "A review of recent advances of binary neural networks for edge computing," *IEEE Journal on Miniaturization for Air and Space Systems*, vol. 2, no. 1, pp. 25–35, 2021.

[27] K. Huang, B. Ni, and X. Yang, "Efficient quantization for neural networks with binary weights and low bitwidth activations," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 1, pp. 3854–3861, 2019.

[28] P.-E. Novac, G. Boukli Hacene, A. Pegatoquet, B. Miramond, and V. Gripon, "Quantization and deployment of deep neural networks on microcontrollers," *Sensors*, vol. 21, no. 9, p. 2984, 2021.

[29] Y. Guo, "A Survey on Methods and Theories of Quantized Neural Networks," 2018, https://arxiv.org/abs/1808.04752.

[30] X. Long, X. Zeng, Z. Ben, D. Zhou, and M. Zhang, "A novel low-bit quantization strategy for compressing deep neural networks," *Computational Intelligence and Neuroscience*, vol. 2020, Article ID 7839064, 7 pages, 2020.

[31] Z. Perić, M. Savić, N. Simić, B. Denić, and V. Despotović, "Design of a 2-bit neural network quantizer for Laplacian source," *Entropy*, vol. 23, no. 8, 2021.

[32] R. Banner, I. Hubara, E. Hoffer, and D. Soudry, "Scalable methods for 8-bit training of neural networks," in *Proceedings of the 32nd Conference on Neural Information Processing Systems (NeurIPS 2018)*, Montreal, QC, Canada, 2018.

[33] S. Uhlich, L. Mauch, F. Cardinaux et al., "Mixed precision DNNs: all you need is a good parametrization," in *Proceedings of the 8th International Conference on Learning Representations*, pp. 1–20, Addis Ababa, Ethiopia, April 2020.

[34] S. Han, H. Mao, and W. J. Dally, "Deep compression: compressing deep neural networks with pruning, trained quantization and Huffman coding," in *Proceedings of the International Conference on Learning Representations*, San Juan, PR, USA, May 2016.

[35] S. Sanghyun and K. Juntae, "Efficient weights quantization of convolutional neural networks using kernel density estimation based non-uniform quantizer," *Applied Sciences*, vol. 9, no. 12, 2019.

[36] Z. Peric, B. Denic, M. Dincic, and J. Nikolic, "Robust 2-bit quantization of weights in neural network modeled by Laplacian distribution," *Advances in Electrical and Computer Engineering*, vol. 21, no. 3, pp. 3–10, 2021.

[37] S. Jayant and P. Noll, *Digital Coding of Waveforms*, pp. 221–251, Prentice-Hall, New Jersey, NJ, USA, 1984.

[38] A. Jovanović, Z. Perić, and J. Nikolić, "Iterative algorithm for designing asymptotically optimal uniform scalar quantization of the one-sided Rayleigh density," *IET Communications*, vol. 15, no. 5, pp. 723–729, 2021.

[39] M. Dinčić, Z. Perić, M. Tančić, D. B Denić, Z. Stamenković, and B. Denić, "Support region of μ-law logarithmic quantizers for Laplacian source applied in neural networks," *Microelectronics Reliability*, vol. 124, Article ID 114269, 2021.

[40] J. Nikolić, Z. Perić, D. Aleksić, S. Tomić, and A. Jovanović, "Whether the support region of three-bit uniform quantizer has a strong impact on post-training quantization for MNIST dataset?" *Entropy*, vol. 23, no. 12, 1699.

[41] L. Deng, "The MNIST database of handwritten digit images for machine learning research [Best of the Web]," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 141-142, 2012.

[42] M. Abadi, A. Agarwal, P. Barham et al., "Tensorflow: Large-Scale Machine Learning on Heterogeneous Distributed Systems," 2016, https://arxiv.org/abs/1603.04467v2.

[43] Guido van Rossum:, "Python reference manual," CWI Report CS-R9525, 1995.

[44] A. F. Agarap, "Deep Learning Using Rectified Linear Units (ReLU)," 2018, https://arxiv.org/abs/1803.08375v2.

[45] Z. Pan, X. Yi, Y. Zhang, B. Jeon, and S. Kwong, "Efficient in-loop filtering based on enhanced deep convolutional neural networks for HEVC," *IEEE Transactions on Image Processing*, vol. 29, pp. 5352–5366, 2020.

[46] T. Salimans and D. P. Kingma, "Weight normalization: a simple reparameterization to accelerate training of deep neural networks," in *Proceedings of the 30th International Conference on Neural Information Processing Systems (NIPS'16)*, pp. 901–909, Barcelona, Spain, December 2016.

[47] Z. Pan, W. Yu, J. Lei, N. Ling, and S. Kwong, "TSAN: synthesized view quality enhancement via two-stream attention network for 3D-HEVC," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 32, no. 1, pp. 345–358, 2022.

[48] Z. Pan, F. Yuan, J. Lei, W. Li, N. Ling, and S. Kwong, "MIEGAN: mobile image enhancement via A multi-module cascade neural network," *IEEE Transactions on Multimedia*, vol. 1, 2021.