On the performance of quantized neural networks based digital predistortion for PA linearization in OFDM systems

Alexis Falempin, Johan Laurent, Jean-Baptiste Dore, Rafik Zayani, Emilio Calvanese Strinati

On the Performance of Quantized Neural Networks based Digital Predistortion for PA linearization in OFDM systems

Alexis Falempin, Johan Laurent, Jean-Baptiste Doré, Rafik Zayani, Emilio Calvanese Strinati
CEA, Leti, Univ. Grenoble Alpes, F-38000 Grenoble, France
alexis.falempin@cea.fr

***Abstract*—Neural networks (NNs) based digital predistortion (DPD) have been shown to be a very promising technique to enhance power amplifier (PA) linearity. However, studies consider high level of quantization NNs (32-bits) whose implementation is not feasible in practice. Therefore, this paper investigates few-bits quantization of NNs based DPD for PA linearization in OFDM communication systems, in order to improve inference time, resources and energy efficiency. To perform quantization, we first operate a post training quantization to assess the impact of quantization on the NN. Second, we perform a quantization aware training (QAT) to cope with the quantization noise. Numerical simulations show that, via the QAT approach, 4-bits quantization of the NN parameters and activation functions, can offer excellent linearization performance, with slight degradation on the error vector magnitude (EVM) performance compared to the ideal case using 32-bits quantization. Consequently, considering 4-bits, the resource usage is reduced by 62% compared to the 32-bits quantization, with slight EVM loss. Thus, the proposed quantization approach puts forward an efficient transmission chain using NN DPD for OFDM based wireless communication systems.**

***Index Terms*—OFDM, Energy-efficiency, Power amplifier, Digital pre-distortion, Machine learning, Neural networks, Quantization.**

I. Introduction

Future communications systems such as 6G systems are leading the path to new services requiring high data rate and ultra low-latency [1]. Nevertheless, the major challenge is to offer a sustainable and cost-effective design of wireless transmitter, leading to green communications. To that extent, it becomes increasingly important to improve the energy consumption and computing resources of wireless transceivers. Particularly, we break down the hardware complexity of neural network (NN) based digital pre-distortion using quantization. Since, the radio-frequency (RF) power amplifier (PA) is the most critical power-hungry component in a transceiver, representing about 60% and 80% of the energy consumed respectively, in the BS and UE RF chains. Thus, its efficiency must be maximized in order to improve the global energy-efficiency transmitter and lessen the overall carbon footprint of the system. Nonetheless, RF PAs exhibits a high level of nonlinearities close to its saturation level, where its power efficiency is high [2]. Indeed, the PA induces nonlinear amplitude-to-amplitude (AM/AM) and amplitude-to-phase (AM/PM) distortions on the transmitted signal affecting the performance at the receiver. Hence, one must find a trade-off between linearity and power efficiency of the system.

To fulfill both aspects, digital pre-distortion has been shown to be the most effective PA linearization technique [3]. It consists in adding a module before the PA, such that the resulting system is linear. However, estimating a DPD module can be challenging due to the severity of nonlinearities induced by the PA.

Thus, machine learning techniques have been investigated in order to solve the issue of deriving a DPD module. Indeed, NNs are proven to be efficient at solving nonlinear problems. Nevertheless, NN based DPD has been widely investigated in the literature. In [4] and [5], authors have proposed deep learning architectures which exhibit high performance. However, those algorithms present a high computational complexity and are difficult to implement. In [6], we designed low-complexity neural network architectures to perform DPD while being highly efficient to derive the DPD function. Nonetheless, even with low-complexity design, implementation of neural networks is still challenging because of computational cost and energy consumption. Hence, to propose efficient implementation of NNs, one must consider quantizing, *i.e.* lowering the number of bits to perform matrix multiplication and activation functions. Quantization of NNs is highly active in the literature since it permits optimizing computational resources and energy consumption [7]. In [8] and [9], authors have studied the different approaches to perform quantization on NNs. However, to the best authors' knowledge, there is no use of quantization regarding state-of-the-art NN based DPD solutions.

The main contribution of this work is to enable quantized NN based DPD to enhance the resource usage of the transmission chain and by extent its energy efficiency. Thus, we analyze the effect of quantization on the performance of our developed solution in [6]. Specifically, we first study the impact of quantization on the parameters of a pre-trained NN based DPD using post-training quantization (PTQ). Using PTQ, numerical evaluation shows that a severe degradation occurs on the Error Vector Magnitude (EVM) and spectrum due to the quantization noise. Thus, we also use quantization aware training (QAT) on our NN to correct the quantization noise. Through numerical simulations, we show that QAT allows to use 4-bits quantization instead of the 32-bits one with

a slight degradation on the EVM and spectrum. In addition, we give en estimation of the resource usage of our solution on FPGA regarding several quantization levels.

The remainder of this paper is structured as follows. Sec. II recalls the system model used in [6]. We introduce the quantization scheme and algorithms employed in Sec. III. While Sec. IV presents our simulation results, Sec. V draws the conclusion and some future perspectives of this work. For the sake of clarity, in this paper, lowercase bold symbols represent vectors, uppercase bold symbols represent matrices and regular symbols represent scalars.

II. System Model

A. Communication model

In this work, we consider a communication system pictured in Fig. 1 using an OFDM transmitter, a DPD based on NNs and a PA. To derive the DPD, a loop back link is necessary and is modeled as a complex perturbation coefficient h and additive noise $\mathbf{n}$. Then, the received symbols are given by

$$\mathbf{y} = h\mathbf{x} + \mathbf{n}, \tag{1}$$

where $\mathbf{z}, \mathbf{y} \in \mathbb{C}^N$, $N = N_{OFDM}\,(N_{fft} + N_{CP})$, $h \in \mathbb{C}$ and $\mathbf{n} \sim \mathcal{CN}(0, \sigma^2)$ with σ^2 the noise variance.

The PA model is characterized by an amplitude distortion function (AM/AM) denoted by f_ρ and a phase distortion function (AM/PM) denoted by f_Φ. The output characteristics, modulus and phase, are given by

$$|\mathbf{y}| = f_\rho(|\mathbf{x}|) \text{ and } \arg(\mathbf{y}) = f_\Phi(|\mathbf{x}|) + \arg(\mathbf{x}) \tag{2}$$

where $\mathbf{x}$ and $\mathbf{y}$ are the PA input and output signals, respectively. Assuming a PA derived from a 3GPP Rapp model for communication [10], the two functions are defined as follows:

$$f_\rho(u) = \frac{Gu}{\left(1 + \left|\frac{Gu}{V_{sat}}\right|^{2p}\right)^{\frac{1}{2p}}},\ f_\Phi(u) = \frac{Au^q}{\left(1 + \left(\frac{u}{B}\right)^q\right)} \tag{3}$$

where u denotes the magnitude of the input signal. G represents the gain in linear region, p the "knee" factor and V_{sat} the saturation voltage level. A, B and q are fitting parameters. Thereafter, we consider the following input back-off (IBO) definition :

$$IBO = \frac{P_{sat,in}}{P_{avg,in}}, \tag{4}$$

where $P_{sat,in}$ corresponds to the input power for which the PA reaches its saturation level and $P_{avg,in}$ the average input power.

B. Low-Complexity Neural Network for DPD (LCDPDNN)

In this section, we recall the architecture of our designed DPD introduced in [6]. A slight modification of this architecture has been done to make it more robust to quantization effects.

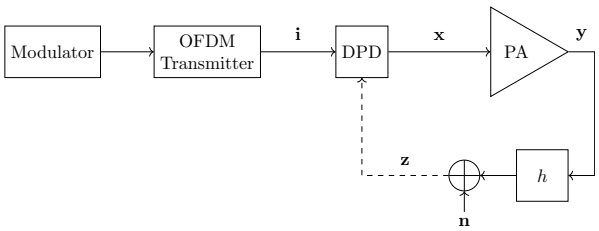
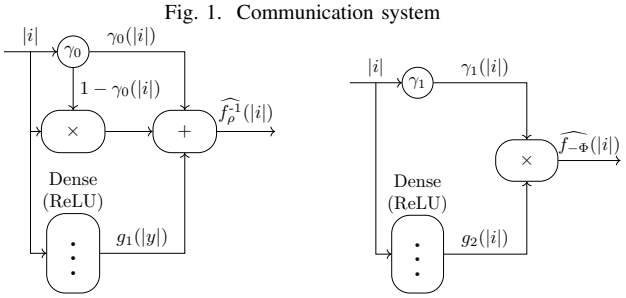


Fig. 1. Communication system



(a) DPD for AM/AM distortion (b) DPD for AM/PM distortion

Fig. 2. Architecture of the LCDPDNN for AM/AM and AM/PM distortions

1) Architecture: We employ two NNs to perform DPD, tackling separately AM/AM and AM/PM distortions for better performance. Fig. 2 presents the architecture of the NN performing DPD.

Each neural network allows to correct respectively AM/AM and AM/PM distortions. Fig. 2a presents the NN estimating the function $\hat{f}_\rho^{-1}$ such that $(f_\rho \circ \widehat{f_\rho^{-1}})(|i|) = G|i|$.

$$\widehat{f_\rho^{-1}}(u) = \text{ReLU}\left[g_0(u) + g_1(u) + b_z\right], \tag{5}$$
$$\text{where} \begin{cases} g_0(u) = \gamma_0(u) + (1 - \gamma_0(u))\,u, \\ \gamma_0(u) = \left(1 + e^{-\alpha(u-\omega_\rho)}\right)^{-1}, \\ g_1(u) = \sum_{j=1}^{N_n^\rho} \omega_j^\rho \left[\text{ReLU}(\mathbf{W}_\rho u + \mathbf{b}_\rho)\right]_j, \end{cases}$$

where $b_z, \alpha, \omega_\rho, \omega_j^\rho \in \mathbb{R}$, $\mathbf{W}_\rho \in \mathbb{R}^{1\times N_n^\rho}$ and $\mathbf{b}_\rho \in \mathbb{R}^{1\times N_n^\rho}$ are trainable variables optimized during the learning phase. N_n^ρ denotes the number of neurons. ReLU function is defined by $f(x_i) = \max(0, x_i), x_i = [\mathbf{x}]_i$.

Similarly, Fig. 2b presents the NN estimating the function $\hat{f}_{-\Phi}$ such that $f_\Phi(|i|) + \hat{f}_{-\Phi}(|i|) = 0$.

$$\hat{f}_{-\Phi}(u) = \gamma_1(u) g_2(u), \tag{6}$$
$$\text{where} \begin{cases} \gamma_1(u) = \left(1 + e^{-\beta(u-\omega_\Phi)}\right)^{-1}, \\ g_2(u) = \sum_{j=1}^{N_n^\Phi} \omega_j^\Phi \text{ReLU}\left[(\mathbf{W}_\Phi u + \mathbf{b}_\Phi)\right]_j, \end{cases}$$

where $\beta, \omega_\Phi, \omega_j^\Phi \in \mathbb{R}$, $\mathbf{W}_\Phi \in \mathbb{R}^{1\times N_n^\Phi}$ and $\mathbf{b}_\Phi \in \mathbb{R}^{1\times N_n^\Phi}$ are trainable variables optimized during the training phase. N_n^Φ denotes the number of neurons.

2) Training: To train both NNs, we consider using indirect learning architecture (ILA). It consists in deriving a postdistorter and placing it before the PA. To do so, we build two datasets $\mathcal{D}^\rho(|\mathbf{z}|, |\mathbf{x}|)$ and $\mathcal{D}^\Phi(|\mathbf{x}|, f_\Phi(\mathbf{z}))$ for training, respectively, AM/AM DPD and AM/PM DPD NNs. θ_ρ and θ_Φ denote all the trainable parameters required to compute respectively $\hat{f}_\rho^{-1}$ and $\hat{f}_\Phi$. θ_ρ and θ_Φ are obtained through the optimization of a mean squared error loss function using gradient descent as,

$$\theta_\rho^k = \theta_\rho^{k-1} - \nabla_{\theta_\rho^{k-1}} \mathcal{L}_\rho(\theta_\rho^{k-1}, \mathcal{D}_\rho), \quad (7)$$
$$\theta_\Phi^k = \theta_\Phi^{k-1} - \nabla_{\theta_\Phi^{k-1}} \mathcal{L}_\Phi(\theta_\Phi^{k-1}, \mathcal{D}_\Phi),$$

where $k \in \{1, \dots, N\}$ with N the number of gradient steps. $\mathcal{L}(\theta^{k-1}, \mathcal{D})$ represents the loss function between the training dataset $\mathcal{D}$ and the predictions of the NNs using the parameters θ^{k-1}. Once the training his finished, we store the optimal parameters for inference and quantization processing.

III. Quantization of Neural Networks

For cost-effective implementation of NNs, quantization is mandatory in order to minimize latency and energy consumption. Thereafter, we present the quantization scheme and techniques applied to the LCDPDNN presented in Sec. II-B.

A. Quantization Scheme

There are two different ways to represent a fractional number, floating-point and fixed-point. In this paper, we consider the fixed-point arithmetic because it is easier to implement and less resource-intensive than floating-point arithmetic.

1) Fixed-Point representation: We define $S(i, f)$ being a signed fixed point number representation with i bits for the integer part including the sign bit and f bits for the fractional part. Using this representation, the range of a number with $i + f$ bits will be $\left[-2^{i-1}, 2^{i-1} - 2^{-f}\right]$ and a step of 2^{-f}. As a toy example, let suppose having $S(1, 1)$ to represent a number r. Then $r \in \{-1, 0.5, 0, 0.5\}$.

2) Fractional number to fixed-point number: Now, let's suppose that we have a given fractional number r to represent using a scheme $S(i, f)$. The resulting fixed point number is given by

$$Q(r, i, f) = clip\left(\frac{\lfloor r2^f \rceil}{2^f}, \left[-2^{i-1}, 2^{i-1} - 2^{-f}\right]\right), \quad (8)$$

where $\lfloor . \rceil$ performs rounding half to even and $clip(u, [min, max])$ returns min if $u < min$, max if $u > max$ and u otherwise. We denote $N_{bits} = i + f$, the total number of bits for the considered scheme.

B. Post Training Quantization (PTQ)

PTQ is the fastest technique to quantize the weights and activation functions of a NN. As its name indicates, the quantization is performed on a pre-trained NN. Regarding our model defined in Sec. II-B, we suppose that we have performed a training leading to optimal set of weights θ_ρ^N and θ_Φ^N.

Then, we apply PTQ on both weights and activation functions using the function Q described in Eq. (8). Each parameter of the set θ_ρ^N and respectively θ_Φ^N can use a different quantization scheme $S(i, f)$. This allows some flexibility to build a trade-off between performance and quantization levels. Finally, each neuron output is quantized the same way as the weights using the Q function.

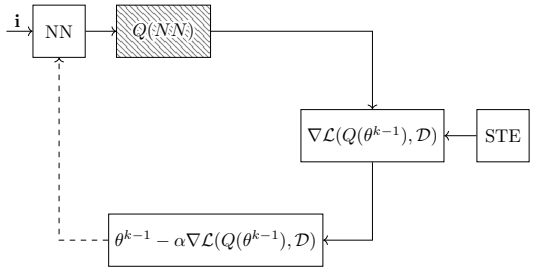


Fig. 3. QAT algorithm for one gradient step

This method is straightforward to enable quantization of NNs. However, we may expect severe performance degradation due to the noise introduced by the quantization. To cope with this issue, we introduce in the next paragraph an algorithm able to mitigate the added quantization noise.

C. Quantization Aware Training (QAT)

QAT is a training algorithm that takes into account the error brought by the quantization of weights and activation functions. Fig. 3 presents the QAT algorithm for one gradient step. We perform a forward pass using the Q function to quantize the weights and activations of the NN. Then, we calculate the gradients of the loss w.r.t. to the weights and finally we update the weights using a regular gradient descent or an optimizer such as Adam [11]. It must be underlined that the quantization only occurs in the forward pass, *i.e.* on the hatched box in Fig. 3. Gradients computation and gradient descent are performed in floating-point over 32 bits to avoid zero gradient issue.

Besides, to compute the gradient of the loss w.r.t. to the quantized weights, one may encounter an issue due to the rounding function. The rounding function is a step function and its gradient is almost zero everywhere. This will leads to a convergence problem during the training. Thus, we use a straight through estimator (STE) [12] which considers the rounding function as a linear function. Then, its derivative equals 1. It is a biased estimator but allows to escape from the zero gradient issue.

Finally, once QAT has fully converged, we stores the quantized weights for further implementation on-chip or field programmable gate array (FPGA).

IV. Numerical simulations

In this section, we evaluate the impact of both PTQ and QAT on the performance of our NN based DPD. Regarding the system model, we consider using a 64-QAM and OFDM with $N_{fft} = 1024$ and $N_{CP} = 72$. The PA follows the model given in Eq. (3) with $p = 0.7$, $A = -345$, $B = 0.17$ and $q = 4$. $N_n^\rho = 6$ and $N_n^\Phi = 3$ are the number of neurons used respectively for the AM/AM and AM/PM DPDs. We evaluate the EVM and conduct a spectrum analysis to assess the impact of different quantization schemes. Besides, the baseline model denotes the DPD w/o quantization.

A. Choice of quantization schemes

In order to perform PTQ and QAT, we must choose coherent quantization schemes regarding the activation functions and weights.

TABLE I
QUANTIZATION OF NN PARAMETERS AND ACTIVATIONS

Parameters	Quantization scheme
α, β	$S(8,5)$
b_z	$S(1,f)$
ω_ρ, ω_Φ	$S(1,8)$
$\mathbf{W}_\rho, \mathbf{W}_\Phi$	$S(3,f)$
$\mathbf{b}_\rho, \mathbf{b}_\Phi$	$S(1,f)$
$\omega_j^\rho, \omega_j^\Phi$	$S(1,f)$
γ_0, γ_1	$S(0,12)$
ReLU	$S(2,10)$

Regarding the activation functions, since the γ function is the most complex, we use the scheme $S(0,12)$ which is an unsigned fixed point number. γ ranges between 0 and 1 hence no bits are required for the integer part. 12 bits are needed to have an efficient sigmoid, otherwise it leads to an unusable DPD. For the ReLUs, we employ the scheme $S(2,10)$ because the output value may exceed 1. Choosing those schemes will ensure no loss on the performance. Moreover, on a real system, input data of the NN will be quantized on 12 bits. Concerning the weights, we are mainly interested in changing the fractional bits f. Indeed, the weights $\omega_j^\rho, \omega_j^\Phi \in [-1,1)$, $\mathbf{b}_\rho, \mathbf{b}_\Phi, b_z \in [-1,1)$ and $\mathbf{W}_\rho, \mathbf{W}_\Phi$ $in[-4,4)$.

B. EVM performance

Here, we analyze the impact of the quantization on the EVM. Fig. 4 presents the EVM performance of our baseline model and using PTQ and QAT. The green line corresponds to the DPD without quantization, the orange and magenta bars are respectively the EVM performance using PTQ and QAT. The number of bits corresponds to the variation of the quantization scheme $S(1,f)$ used for parameters described in Table I.

First, we can notice that the PTQ exhibits high performance loss when the number of bits is lower than 7. We also remark that for quantization levels higher than 7 bits, the DPD reaches an EVM threshold around -30dB which is still far from the baseline model reaching -40dB.

Second, we observe that QAT allows to almost reach the baseline model performance. The EVM always reaches -35dB. For quantization levels higher than 7 bits, QAT permits reaching the same performance as the baseline model. Besides, for low-bit quantization, *i.e.* lower than 6 bits, we have up to 30dB gain in terms of EVM performance.

C. Spectrum analysis

In this paragraph, we conduct a spectrum analysis concerning the impact of quantization on the our NNs. All the spectrums presented are obtained using a digital up converter (DUC) function in order to design a realistic RF chain. This function is composed of multiple filters which design and coefficients can be found in [13]. Fig. 5 and Fig. 6 present power spectral densities (PSDs) using respectively $S(1,9)$ and $S(1,3)$ quantization schemes for the DPD.

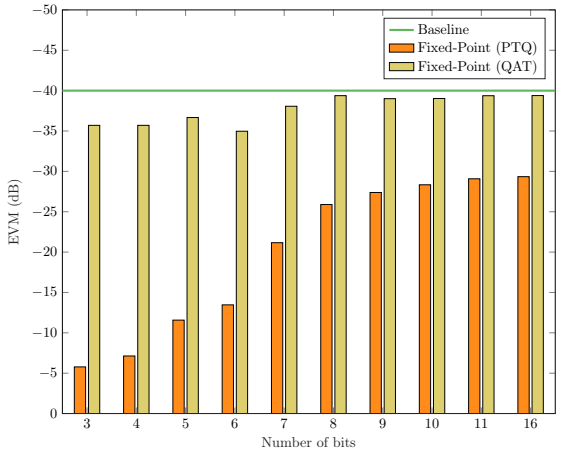


Fig. 4. EVM performance using PTQ and QAT

First, in Fig. 5, concerning the PTQ, we observe that we have a small degradation which is correct. Using QAT allows to have the same PSD as the baseline model. Second, in Fig. 6, the PTQ produces a PSD which is worse than using the PA w/o DPD. However, QAT will almost cancel the quantization noise. All the nonlinearities from the PA are canceled. Even if a small degradation is still present compared to the baseline model, the noise level is below -40dB which is sufficient for practical systems. Thus, we can state that QAT is mandatory to improve the performance of the LCDPDNN while using quantization.

D. Trade-off between implementation complexity and performance

To evaluate the benefits of the quantization, we perform an analysis of the resource consumption using a FPGA synthesis for implementation on a system on chip from Xilinx [14]. Specifically, the resource usage is given in terms of Configurable Logic Blocks (CLB) that constitute the FPGA. Fig. 7 presents the resource usage of the FPGA for the quantization scheme $S(1,f)$. The resource usage is presented in terms of

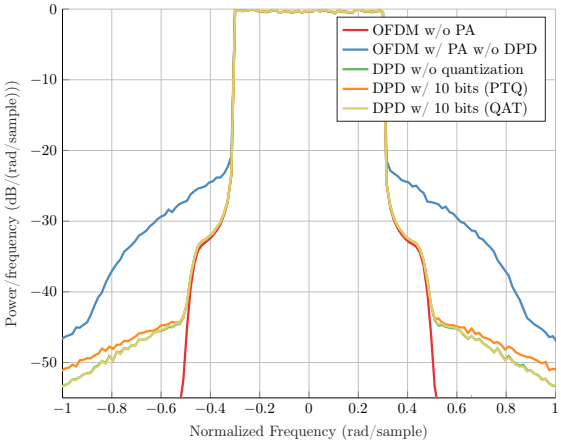


Fig. 5. Spectrum using PTQ and QAT, $N_{bits} = 10$

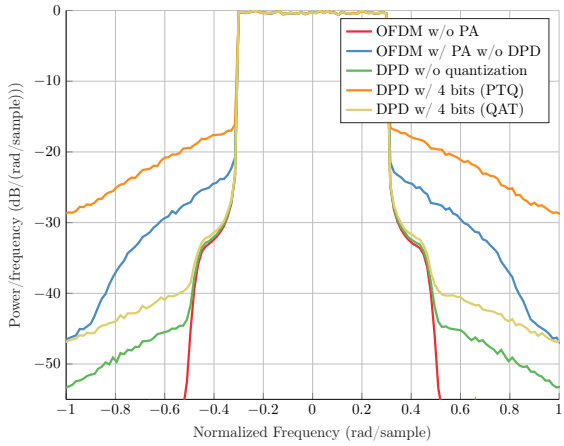


Fig. 6. Spectrum using PTQ and QAT, $N_{bits} = 4$

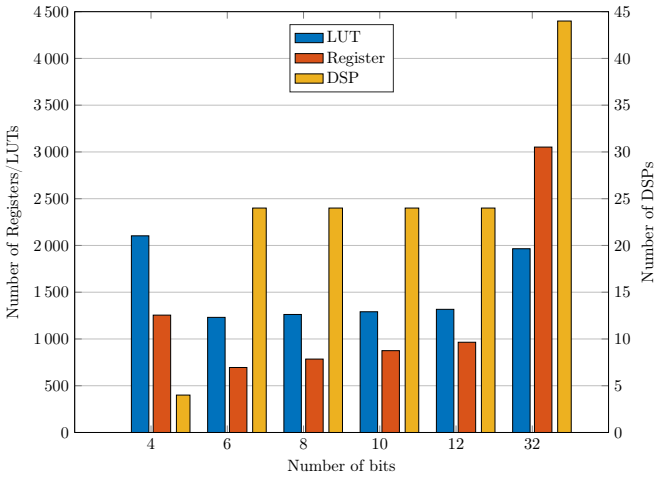


Fig. 7. Resource consumption on FPGA

lookup tables (LUTs) performing logical functions, digital signal processing (DSP) units mainly performing multiplication and accumulation operations and Registers.

In Fig. 7, the blue bar presents the number of required LUTs to compute LCDPDNN on FPGA. We observe that the LUTs usage is the same between 6 and 12 bits but is higher with 32 bits. The same interpretation can be given regarding the registers usage represented by the red bars. It may seem that the FPGA exhibits higher resource usage for 4 bits regarding the number of LUTs and registers. However, judging by the yellow bar, it can be noted that the FPGA uses less DSPs than other quantization schemes which justify the rise of LUTs and registers. As a comparison, the implementation of a 4096 points FFT (12 bits input/output) requires 3160 LUTs, 6230 registers and 48 DSPs. The proposed implementation is therefore of a slightly lower order of magnitude. Next, we can approximate the whole resource usage by considering that 1 LUT is equivalent to 2 registers and 1 DSP is equivalent to 109 LUTs. Thus, using this approximation, 4-bits quantization reduces the resource usage by 62% compared to the 32-bits one, with a slight degradation on the EVM, leading to less than -35dB.

V. Conclusion

In this paper, we studied the use of quantization on NNs performing DPD for PA linearization. The goal is to reduce resource usage and to improve its energy efficiency. We performed a fixed-point quantization on the parameters and outputs of the NNs using PTQ to first assess the impact of the latter. Numerical evaluations show that PTQ is severely degrading the performance when the number of bits is low which is undesirable. Thus, using QAT allows to reduce the quantization noise and then retrieve the baseline performance in terms of EVM and spectrum considerations. We showed that we can use only 4 bits for most parameters and 12 bits for the outputs of the NNs to get excellent performance compared to the baseline model. In addition, through a FPGA synthesis of our NN, it appears that quantization highly reduce the resource usage. Hence, through fixed-point quantization and QAT, we provided a low-complexity and low-cost NN based DPD that can be used in real time systems. Eventually, future work will enable the implementation of the proposed solution using a FPGA taking into account the quantization aspects for low-cost design.

References

[1] E. Calvanese Strinati, S. Barbarossa, J. L. Gonzalez-Jimenez, D. Ktenas, N. Cassiau, L. Maret *et al.*, "6G: The Next Frontier: From Holographic Messaging to Artificial Intelligence Using Subterahertz and Visible Light Communication," *IEEE Veh. Technol. Mag.*, vol. 14, no. 3, pp. 42–50, 2019.

[2] S. Cripps, "Nonlinear Effects in RF Power Amplifiers," in *RF Power Amplifiers for Wireless Communications, Second Edition*. Artech House, 2006, pp. 231–283.

[3] M. A. Hussein, O. Venard, B. Feuvrie, and Y. Wang, "Digital predistortion for RF power amplifiers: State of the art and advanced approaches," in *IEEE 11th International New Circuits and Systems Conference (NEWCAS)*, 2013, pp. 1–4.

[4] C. Tarver, L. Jiang, A. Sefidi, and J. R. Cavallaro, "Neural Network DPD via Backpropagation through a Neural Network Model of the PA," in *53rd Asilomar Conf. Signals, Systems, and Computers*, 2019, pp. 358–362.

[5] Y. Wu, U. Gustavsson, A. G. i. Amat, and H. Wymeersch, "Residual Neural Networks for Digital Predistortion," in *IEEE Global Communications Conf. (GLOBECOM)*, 2020, pp. 01–06.

[6] A. Falempin, R. Zayani, J.-B. Doré, and E. C. Strinati, "Low-Complexity Adaptive Digital Pre-Distortion with Meta-Learning based Neural Networks," in *IEEE 19th Annual Consumer Communications and Networking Conference (CCNC)*, 2022, pp. 453–453.

[7] M. Horowitz, "1.1 Computing's energy problem (and what we can do about it)," in *IEEE International Solid-State Circuits Conference (ISSCC)*, 2014, pp. 10–14.

[8] A. Gholami, S. Kim, Z. Dong, Z. Yao, M. W. Mahoney, and K. Keutzer, *A Survey of Quantization Methods for Efficient Neural Network Inference*. Chapman and Hall, 2022.

[9] M. Nagel, M. Fournarakis, R. A. Amjad, Y. Bondarenko, M. van Baalen, and T. Blankevoort, *A White Paper on Neural Network Quantization*. Chapman and Hall, 2022.

[10] Nokia, "Realistic power amplifier model for the New Radio evaluation," 3GPP TSG-RAN WG4, Tech. Rep., 2016.

[11] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *3rd Int. Conf. Learning Representations (ICLR)*, 2015.

[12] Y. Bengio, N. Léonard, and A. C. Courville, "Estimating or Propagating Gradients Through Stochastic Neurons for Conditional Computation," *CoRR*, vol. abs/1308.3432, 2013.

[13] Xilinx, "Zynq UltraScale+ RFSoC RF Data Converter v2.6 Gen 1/2/3," Tech. Rep., 2021.

[14] ——, "Zynq UltraScale XCZU9EG MPSoC Data Sheet," Tech. Rep., 2021.