(Received 5 December 2024, Accepted 15 January 2025, Scheduled 31 January 2025)

On Selecting Activation Functions for Neural Network-Based Digital Predistortion Models

Mostapha Ouadefli[1, *], Mohamed Et-tolba[1], Abdelwahed Tribak[1], and Tomas Fernandez Ibanez[2]

[1] *Institut National des Postes et Télécommunications (INPT), Rabat, Morocco*
[2] *Universidad de Cantabria (UNICAN), Santander, Spain*

ABSTRACT: Neural networks have become a focal point for their ability to effectively capture the complex nonlinear characteristics of power amplifiers (PAs) and facilitate the design of digital predistortion (DPD) circuits. This is accomplished through the utilization of nonlinear activation functions (AFs) that are the cornerstone in a neural network architecture. In this paper, we delve into the influence of eight carefully selected AFs on the performance of the neural network-based DPD. We particularly explore their interaction with both the depth and width of neural network. In addition, we provide an extensive performance analysis using two crucial metrics: the normalized mean square error (NMSE) and adjacent channel power ratio (ACPR). Our findings highlight the superiority of the exponential linear unit activation function (ELU AF), particularly within deep neural network (DNN) frameworks, among the AFs under consideration.

1. INTRODUCTION

Power amplifiers (PAs) are critical components in communication systems since they amplify the signal, at the transmitter output, to be suitable for transmission. In fact, an increase in the signal power can compensate for losses that occur in the transmission channel. However, in modern communication, achieving higher spectral efficiency imposes stringent requirements on the modulation and multiplexing techniques at the expense of power efficiency. For instance, the orthogonal frequency division multiplexing (OFDM), which has been selected for the long term evolution (LTE), and the fifth generation (5G) mobile communication systems, generate signals with high peak-to-average power ratio (PAPR) [1]. Transmitting such signals without introducing nonlinear distortion constrains the PA to operate in its linear region. This is not in concordance with the fact that a PA provides its maximum power efficiency when it operates in its saturation (nonlinear) region [2]. Consequently, transmitting signals with high PAPR values is challenging since it introduces a conflict between power efficiency and spectral efficiency. To deal with this issue, digital predistortion (DPD) has been widely applied [3]. DPD is a linearization technique that can significantly increase both power and bandwidth efficiency of a power amplifier, hence taking full benefit of its capabilities.

Traditionally, DPD design has relied on polynomial-based models like memoryless polynomial [4], memory polynomials (MP) [5], dynamic deviation-reduction-based Volterra (DDR-Volterra) [6], and generalized memory polynomials (GMP) [7], derived from Volterra series. Nevertheless, these models often face limitations due to high correlation between their basis functions, especially with higher model orders.

Recently, artificial neural networks (ANNs) have garnered attention for their ability to model nonlinear functions effectively [8]. This is achieved through the use of nonlinear activation functions (AFs), which are considered the heart of any NN, giving it the essence of artificial intelligence. The significance of AFs has been explored in various domains, such as facial expression recognition [9], visual pattern recognition [10], and image classification [11]. Regarding the DPD design, the authors in [12] have briefly discussed and analyzed different AFs, concluding that the hyperbolic tangent function provides better performance. However, the considered NN consists of only one hidden layer with a limited number of coefficients (1000 or less). In addition, this analysis has been done disregarding the latest AFs. In [13], another study has found sigmoid AF to be superior when the coefficients count is below 2000, but it is outperformed by ReLU AF when this count exceeds 2000. Notably, this analysis solely focuses on two AFs and their effect on the adjacent channel leakage ration (ACLR) for signal with bandwidth less than 6.6 MHz. A similar analysis is performed in [14], where the ELU activation function outperforms the sigmoid activation function. In [15], an adaptive activation function is proposed to improve the performance of the DPD. However, this study is limited to a shallow neural network.

In this paper, we aim to demonstrate the pivotal role of selecting the right activation function in bolstering the performance of the NN-based DPD model. Our study investigates the influence of eight distinct activation functions on the NN-DPD model across various configurations of depth (number of hidden layers) and length (number of neurons per layer). Evaluation of performance is conducted using two primary metrics: the normalized mean square error (NMSE) for quantifying in-band distortion and the adjacent channel power ratio (ACPR) for measuring out-of-band distortion.

[*] Corresponding author: Mostapha Ouadefli (ouadefli.mostapha@doctorant.inpt.ac.ma).

doi:10.2528/PIERC24120508

Published by THE ELECTROMAGNETIC ACADEMY

The remainder of this paper is organized as follows. Section 2 provides a brief introduction to PA behavioral modeling and DPD design. Section 3 delves into the architecture of the neural network used for designing the DPD and testing different AFs. Section 4 highlights the significance of AFs in modeling nonlinear systems and introduces several common AFs. Section 5 presents numerical results and their analysis. Finally, Section 6 concludes the paper.

2. POWER AMPLIFIER BEHAVIOR AND DIGITAL PREDISTORTION

As mentioned previously, even a PA increases the power of its input signal to levels appropriate for transmission, it causes signal distortion and can dramatically increase the power consumption at the transmitter. This is due to the behavior of a PA, consisting of nonlinearity and memory effects caused by the blocks it comprises. It is worth noting that the memory effect is nonlinear in modern communication systems as the transmitted signal has a wider bandwidth. Accordingly, one should take into account these aspects when modeling the PA behavior.

There are various approaches to PA behavioral models including memoryless models, linear memory models, and nonlinear memory models. The most common approach to nonlinear behavioral modeling of a PA is the polynomial model with Volterra series.

Let $x[k]$ and $y[k]$ be the discrete-time signal at the input and output of the PA. According to the Volterra model, the input-output relationship of the PA is formulated as,

$$y[k] = \sum_{m=0}^{\infty} \sum_{q_{2m+1}=0}^{Q_{2m+1}} h_{2m+1}[q_{2m+1}] \prod_{r=1}^{m+1} x[k-q_r] \prod_{r=m+2}^{2m+1} x^*[k-q_r] \qquad (1)$$

where $K = 2m+1$ and $M = q$ are the order of nonlinearity and the memory depth, respectively. h_k represents the kernels of the system. Although the model given in (1) can accurately represent the nonlinear behavior of a power amplifier, it is computationally expensive, especially when the nonlinear order K and memory depth M are high. To deal with this limitation, some improvements are made on the Volterra model, resulting in variants known as the modified Volterra series [16].

2.1. Digital Predistortion

DPD is primarily motivated by the need to mitigate nonlinear distortions that arise from PAs within communication systems. As mentioned earlier, PAs exhibit nonlinear behavior, especially when being driven at high power levels, contributing to distortions in the transmitted signal. These distortions, including intermodulation distortion (IMD), spectral regrowth, and out-of-band emissions, degrade signal quality. DPD aims to compensate for these nonlinear effects, enabling more efficient and higher-quality transmission in communication systems.

DPD functions by preemptively applying an inverse distortion to the signal before it passes through the PA. This inverse distortion is specifically tailored to cancel out the nonlinear effects introduced by the PA, resulting in a cleaner output signal. The process involves modeling the nonlinear behavior of the PA using techniques such as memory polynomial models and lookup tables [17]. Once the distortion characteristics are known, DPD algorithms generate a pre-distorted signal that, when being passed through the PA, will produce the desired undistorted output.

Conventional DPD methods commonly rely on mathematical models of the PA, which might not fully grasp all the complexities of its nonlinear behavior. In contrast, NN-based approaches offer a more flexible and adaptable solution for DPD [18–20]. They have the capability to learn the nonlinear relationships between input and output signals directly from training data, eliminating the need for explicit models. This flexibility enables NN-based DPD to adjust more effectively to variations in amplifier characteristics and operating conditions. Furthermore, NN-based DPD can potentially achieve superior performance and efficiency compared to traditional techniques, particularly in situations where the amplifier's behavior is highly nonlinear or challenging to model accurately.

3. STRUCTURE OF NEURAL NETWORK-BASED APPROACH TO DPD

To model the PA behavior and design the DPD in a neural network perspective, we exploit the fact that the transfer function of the DPD is the inverse of the PA's transfer function. Then, the actual input of the PA serves as the output for NN-DPD, and the output of the PA is employed as the input for NN-DPD. The structure of the neural network used, in this work, for emulating the PA behavior and the DPD technique is depicted in Fig. 1, and training process is based on the indirect learning architecture. It consists of L ($L \geq 3$) layers: an input layer, an output layer, and multiple $(L-2)$ hidden layers.

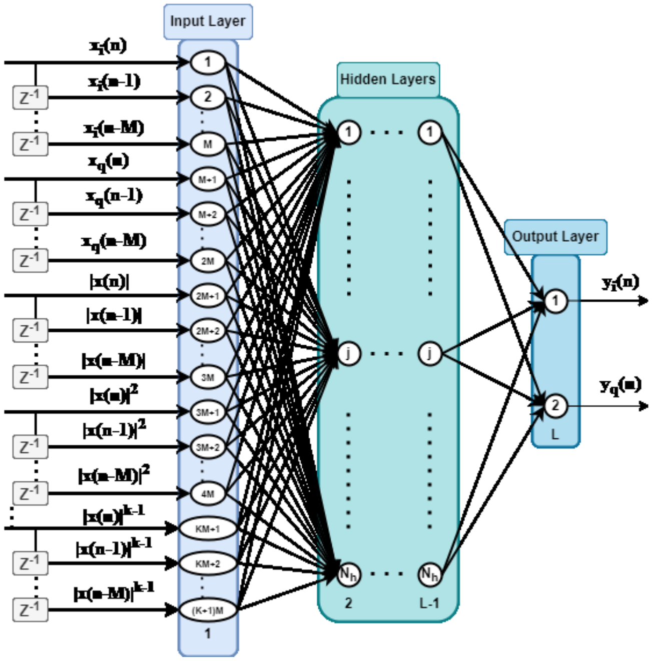


FIGURE 1. Block diagram of NN-DPD.

The input layer of the neural network comprises passive neurons or units transmitting their assigned values to each neuron in the first hidden layer. Moreover, to enhance the modeling accuracy, the input signal includes current and past samples of the in-phase (I) and quadrature (Q) components, along with envelope-dependent terms, as described in [12]. The vector, denoted as $\mathbf{x_1^{(1)}}$, representing the signal at the input of the neural network, is split into $K+1$ sub-vectors, $\mathbf{x_1^{(1)}}, \mathbf{x_2^{(1)}}, \ldots, \mathbf{x_{K+1}^{(1)}}$, each of size M. It is expressed as follows,

$$\mathbf{x^{(1)}} = \left[\mathbf{x_1^{(1)}}, \mathbf{x_2^{(1)}}, \ldots, \mathbf{x_{K+1}^{(1)}}\right] \tag{2}$$

where the two first sub-vectors represent the in-phase (real part) and the quadrature (imaginary) components of the complex-valued input signal, and their past samples. They are given by,

$$\begin{aligned}\mathbf{x_1^{(1)}} &= [\mathbf{x_i(n)}, \mathbf{x_i(n-1)}, \ldots, \mathbf{x_i(n-M)}] \\ \mathbf{x_2^{(1)}} &= [\mathbf{x_q(n)}, \mathbf{x_q(n-1)}, \ldots, \mathbf{x_q(n-M)}]\end{aligned} \tag{3}$$

The remaining $K-1$ sub-vectors represent the powers of the amplitudes of the input complex-valued samples. For $k = 3, \ldots, K+1$, we have,

$$\mathbf{x_k^{(1)}} = \left[|\mathbf{x(n)}|^{\mathbf{k-2}}, |\mathbf{x(n-1)}|^{\mathbf{k-2}}, \ldots, |\mathbf{x(n-M)}|^{\mathbf{k-2}}\right] \tag{4}$$

It is worth noting that the optimal values of the nonlinearity order K and memory depth M will be selected through an optimization process that we will explain later in this paper.

Hidden layers of an NN are located between the input layer and output layer. They play a vital role in learning complex features and representations. When an NN consists of only one hidden layer ($L = 3$), it is called shallow neural network. Otherwise, if it has more than one hidden layer ($L \geq 3$), it is named deep neural network (DNN).

In our approach, we opted for fully-connected layers where all neurons from the preceding $(l-1)$th layer are connected to those of the lth current layer. The output of the ith neuron in the lth layer is given by,

$$x_i^{(l)} = g^{(l)}\left(\sum_{j=1}^{N_h} w_{i,j}^{(l)} x_j^{(l-1)} + b_i^{(l)}\right) \tag{5}$$

where $x_j^{(l-1)}$ is the output of the jth neuron in the $(l-1)$th layer; $w_{i,j}^{(l)}$ represents the weight associated with the connection between the ith neuron of the lth layer and jth neuron of the $(l-1)$th layer; $b_i^{(l)}$ and $g^{(l)}(\cdot)$ are the bias and activation function of the lth layer, respectively. The importance of this function will be thoroughly explained in the next section.

The final layer in the NN is the output layer, which consists of two neurons. The first neuron generates the predicted in-phase component of the output signal, denoted as $y_{i(n)}$, while the second neuron computes the predicted quadrature component, denoted as $y_q(n)$. It is worth mentioning that the activation function of this layer is linear, which makes it typical for regression problems. Accordingly, the output components can be expressed as follows,

$$y_i(n) = x_1^{(L)} = \sum_{j=1}^{N_h} w_{1,j}^{(L)} x_j^{(L-1)} + b_1^{(L)} \tag{6}$$

$$y_q(n) = x_2^{(L)} = \sum_{j=1}^{N_h} w_{2,j}^{(L)} x_j^{(L-1)} + b_2^{(L)} \tag{7}$$

4. ACTIVATION FUNCTIONS

4.1. Importance of Activation Functions

Consider the NN presented in Fig. 1 and examine the output of the ith neuron of the lth hidden layer. As shown in Fig. 2, the inputs to this neuron are the weighted outputs computed by the neurons of the $(l-1)$th hidden layer. If the activation function $g(\cdot)$ is not involved in the computation of the hidden layers' outputs, the NN is equivalent to a linear regression model, where each node's output is a linear combination of its inputs plus a bias. Accordingly, Equation (5) becomes,

$$x_i^{(l)} = \sum_{j=1}^{N_h} w_{i,j}^{(l)} x_j^{(l-1)} + b_i^{(l)} \tag{8}$$

Obviously, the number of hidden layers becomes irrelevant in this scenario, as the composition of two linear functions remains linear. This model is not suitable to emulate the behavior of complex systems such as PAs, and the DPD technique, which are inherently nonlinear. To deal with this problem, it is imperative to use an activation function to introduce the nonlinearity aspect in the computations within the NN layers. This enables it to capture intricate relationships in the data and extract relevant features. With the AF, the output of the ith neuron in the lth layer can be calculated using Equation (5). For an accurate modeling, AFs should be continuous and differentiable to facilitate back-propagation optimization, enabling the computation of errors or losses with respect to weights. This makes it easy to apply gradient-based techniques for weights optimization. Furthermore, desirable attributes of AFs include boundedness within a specific range, monotonic behavior, and computational efficiency. In the subsequent subsection, we introduce various AFs for NN-based DPD modeling. To the best of our knowledge, some of them have been previously employed in the DPD design, while the others have not been utilized in this context.

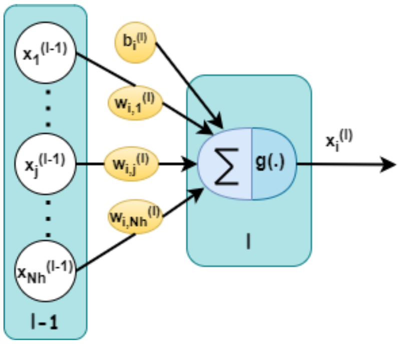


FIGURE 2. The importance of AF in NN.

4.2. Study of Activation Functions

Due to their importance in neural networks, hereafter we provide an extensive analysis of various activation functions.

- Sigmoid AF [21], also known as logistic sigmoid, stands as one of the earliest activation functions employed in neural networks. It is expressed as,

  $$g(x) = sigmoid(x) = \frac{1}{1+e^{-x}} \tag{9}$$

  This function transforms any real-valued input into a range between 0 and 1. When the output value approaches 1, the neuron becomes active, facilitating the flow of information, whereas a value closer to 0 signifies inactivity. Despite its advantages, sigmoid function has the main drawback of vanishing gradient. This issue arises when the gradient of the loss function with respect to the weights of early layers diminishes significantly, resulting in inadequate updates to these weights.

- The tangent hyperbolic AF [22], often denoted as tanh, is a shifted version of the sigmoid function that maps real-valued inputs to outputs within a range spanning from -1 to 1. Mathematically, this is formulated as,

  $$g(x) = \tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} = \frac{e^{2x}-1}{e^{2x}+1} \tag{10}$$

  One notable characteristic of this function is its zero-centered nature, which ensures faster convergence. However, similar to the sigmoid function, tanh does not solve the vanishing gradient problem.

- Symmetric Elliot AF [23] is a high-speed approach to the sigmoid AF, with an output ranging from -1 to 1, similar to tanh. Unlike tanh, it is smoother and grows polynomially rather than exponentially, which mitigates issues with vanishing gradients. However, it has a higher computation complexity than tanh AF because it involves complex derivatives. It is expressed as,

  $$g(x) = SElliot(x) = \frac{x}{1+|x|} \tag{11}$$

  It is noteworthy that this function is a special case of the parametric Elliot function, with a hyperparameter a, which is given by,

  $$g(x) = PElliot(x) = \frac{x(1+a|x|)}{1+|x|(1+a|x|)} (a \geq 0) \tag{12}$$

  This function is explored alongside its parametric counterpart in the design of the DPD system discussed in this paper.

- The rectified linear function [24], or ReLU, is a straightforward calculation that returns the input value directly if it is positive, or 0 if it is negative. Its equation is presented as:

  $$g(x) = ReLU(x) = \max(x,0) = \begin{cases} 0 & \text{if } x \leq 0 \\ x & \text{if } x > 0 \end{cases} \tag{13}$$

  Unlike sigmoid and tanh AFs, ReLU does not encounter the vanishing gradient problem for positive inputs, allowing for more stable and efficient training of deep neural networks. Nevertheless, one of its drawbacks is the occurrence of 'dead' neurons. When dealing with negative input values, the gradient flowing through the neuron always remains 0 during back-propagation. Consequently, this prevents weight updates, rendering the node useless.

- Exponential Linear Unit (ELU) [25] represents a variant of the ReLU function. While retaining the identity operation for positive inputs, ELU employs an exponential nonlinearity for negative inputs. This unique characteristic ensures ELU's smoothness and differentiability across all values, including around 0, effectively mitigating the 'dying ReLU' problem and enhancing training stability. However, this improved functionality comes at a computational cost compared to the simple thresholding operation of ReLU. Moreover, ELU introduces an additional hyperparameter, α, which governs the negative slope for negative inputs. Proper tuning of α is essential for optimizing performance. The mathematical expression for ELU is as follows:

  $$g(x) = ELU(x) = \begin{cases} x & \text{if } x > 0 \\ \alpha(e^x - 1) & \text{if } x \leq 0 \end{cases} \tag{14}$$

  where α is an hyperparameter controlling the negative slope for negative inputs. Note that a proper tuning of this hyperparameter is essential for optimizing performance.

- GELU stands for Gaussian Error Linear Unit [26]. It is designed to overcome some of the limitations of ReLU, such as the 'dying ReLU' problem and its inability to effectively model negative values. GELU accomplishes this by allowing small negative values when the input is less than 0, thereby providing a richer gradient for backpropagation. Its mathematical expression is defined as follows,

  $$\begin{aligned} g(x) &= GELU(x) = x\phi(x) \\ &= x \cdot \frac{1}{2}\left(1 + erf\left(\frac{x}{\sqrt{2}}\right)\right) \end{aligned} \tag{15}$$

  where ϕ is the cumulative distribution function of the standard normal distribution. Additionally, we can approximate GELU using Equation (16) if the speed of feedforward computation outweighs the necessity for exactness. Then, we have,

  $$\begin{aligned} g(x) &= GELU(x) \\ &= 0.5x\left[1+\tanh\left(\sqrt{\frac{2}{\pi}}\left(x+0.044715x^3\right)\right)\right] \end{aligned} \tag{16}$$

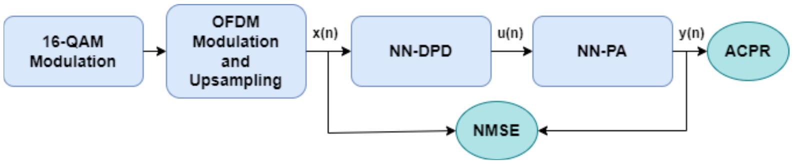


FIGURE 3. Block diagram of the test setup.

where $erf(\frac{x}{\sqrt{2}}) \approx \tanh(\sqrt{\frac{2}{\pi}}(x + 0.044715x^3))$.

- Swish has attracted considerable attention in deep learning circles due to its advantageous characteristics, including smoothness and non-monotonicity. Originally introduced by Hendrycks and Gimpel [26] under the name SiLU, the term 'swish' gained more popularity after its explicit introduction and naming in [27]. This function is a slight modification of the sigmoid function. Mathematically, it is expressed as,

$$g(x) = Swish(x) = x * sigmoid(\beta x) \qquad (17)$$

where β serves as a flexible and trainable parameter. Swish effectively addresses the vanishing gradient issue often encountered with sigmoid functions, although this improvement may come with a higher computational cost than ReLU. Moreover, its performance can be sensitive to the initialization of β. In numerous scenarios, this parameter is commonly set to 1, which is the case in this particular study.

4.3. Computational Complexity of Activation Functions

In addition to their impact on linearization performance, the computational complexity of AFs is a critical factor in the real-time implementation of neural network-based DPD systems. The AFs considered in this study vary in their computational demands due to differences in their mathematical formulations.

Functions like sigmoid and tanh involve exponential calculations, which are computationally intensive and may slow down processing speed. ELU and GELU functions include exponential and error function components, respectively, adding to their complexity.

Moreover, AFs that incorporate adjustable parameters, such as ELU, parametric Elliot, and Swish, require fine-tuning of these parameters (the parameter α for ELU, a for parametric Elliot, and β for Swish) to achieve optimal performance. This fine-tuning process can increase the computational load during the training phase, as it involves additional iterations and validation steps to find the best parameter values. In contrast, AFs like ReLU and symmetric Elliot rely on basic arithmetic operations such as addition, multiplication, and comparison, and do not require parameter tuning, making them more efficient for implementation.

Considering these complexities and the need for parameter tuning is essential when selecting an AF. It affects not only the performance metrics like NMSE and ACPR but also the feasibility of deploying the DPD system in resource-constrained environments where computational resources and time are limited.

5. NUMERICAL SIMULATIONS AND RESULTS

To evaluate and analyze the performance of the NN-based approach to PA modeling and DPD design, we run numerous computer simulations on MATLAB platform. The utilized dataset is provided by MathWorks [28]. It comprises measured input and output signals obtained from an NXP Airfast LDMOS Doherty PA operating within the frequency band of 3.6–3.8 GHz, delivering a gain of 29 dB and is suitable and commonly used for LTE and 5G applications. The test signal utilized is a 100 MHz 5G-like OFDM waveform, incorporating 16-QAM symbols for each subcarrier. It is worth noting that our study employs a PA model to evaluate the performance of NN-based DPD systems instead of using an actual PA. This approach has inherent limitations, as simulated models may not capture all real-world nonlinearities, environmental factors, and hardware-specific impairments such as temperature variations and component aging.

5.1. PA Behavioral Modeling

To obtain an accurate model of the PA behavior, it is important to select the best input combination, as provided in (2). This combination depends directly on the values of nonlinearity (K) and memory depth (M) [29]. To select the values of these two parameters, we assume that an NN with only two hidden layers is sufficient to model the behavior of the PA. In fact, according to [8], an NN with just one hidden layer containing enough hidden neurons (or nodes) can approximate any measurable function. Consequently, the number of hidden neurons N_h per the l hidden layer is set to be large enough to efficiently handle all possible combinations. It is noteworthy that the activation function employed in this process is ReLU AF.

Afterward, we conducted an extensive training of the network with various combinations of K and M. These values are varied from 1 to 7, resulting in 49 training sessions. For each combination, the network is trained, and the corresponding root mean square error (RMSE) is calculated and saved for selecting the optimal values of K and M that gives the best modeling of the PA behavior.

Figure 4 illustrates the RMSE values as a function of K and M. It is clearly observed that the lowest RMSE value is achieved when $K = 3$ and $M = 7$. Consequently, these values were chosen to determine the optimal input combination for training the NN-DPD and testing different activation functions.

5.2. DPD Design

The procedure for training the NN-DPD before it is deployed alongside the PA is outlined in Fig. 5. Initially, we select the activation function for the hidden layers. Afterwards, we set

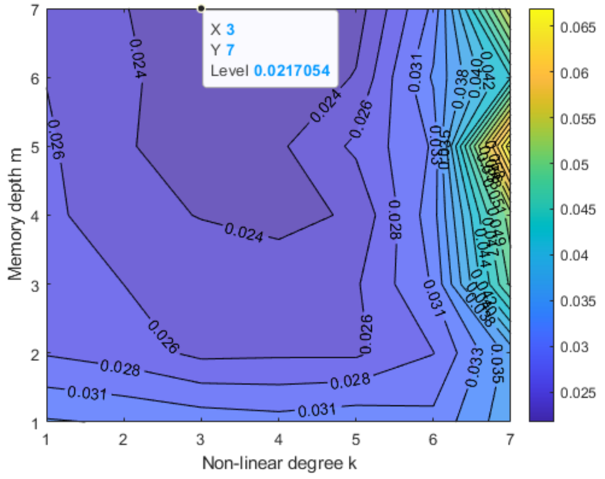


FIGURE 4. RMSE vs nonlinear degree k and memory depth m.

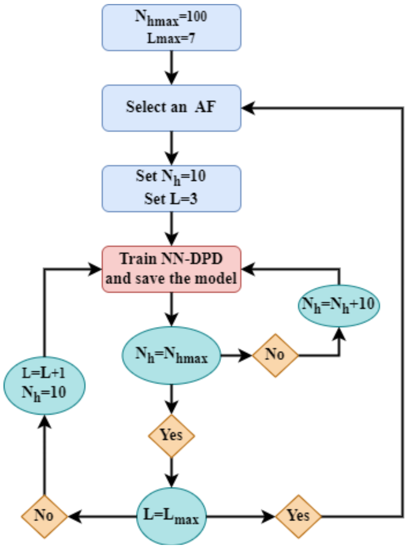
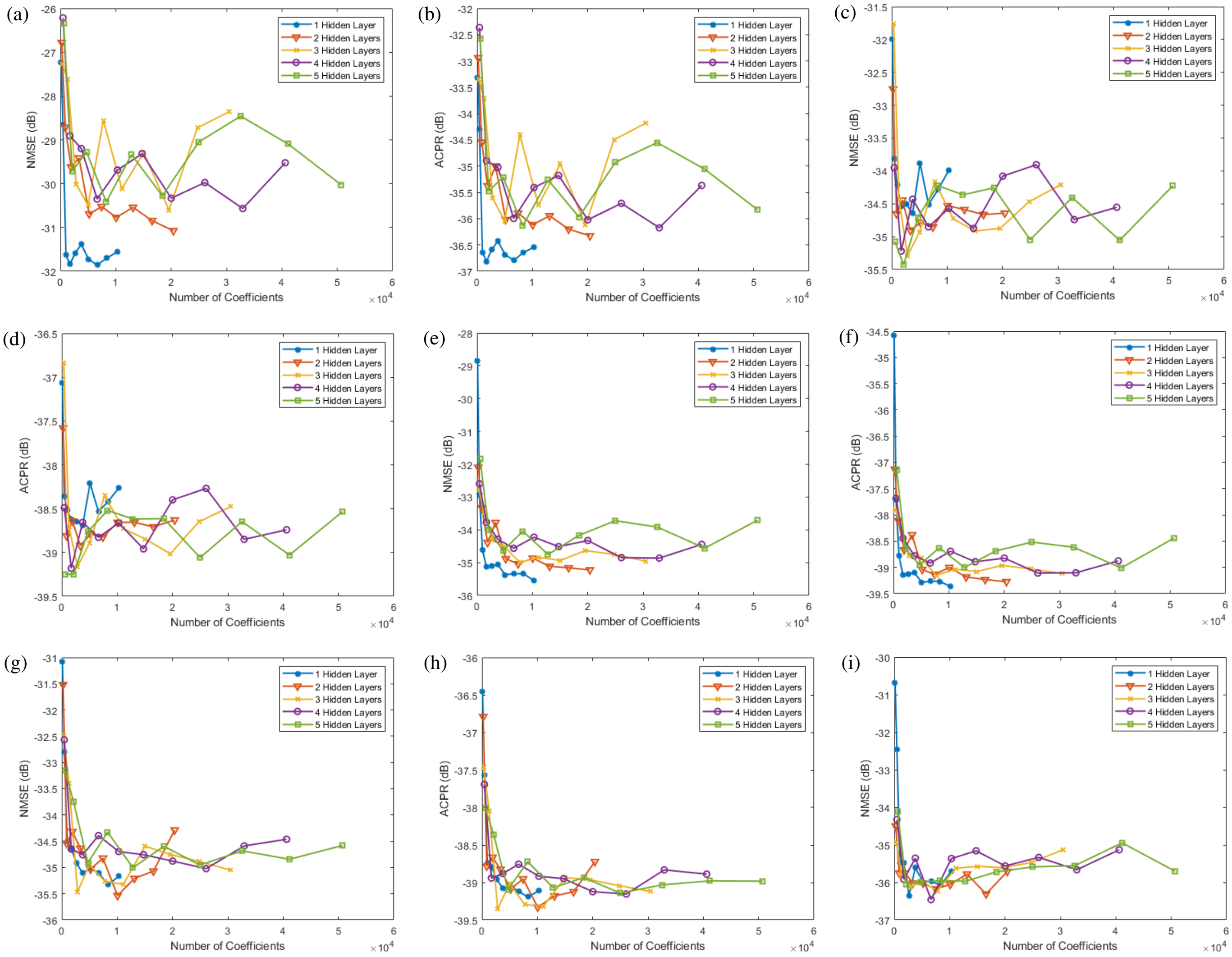


FIGURE 5. Algorithm of NN-DPD models training.



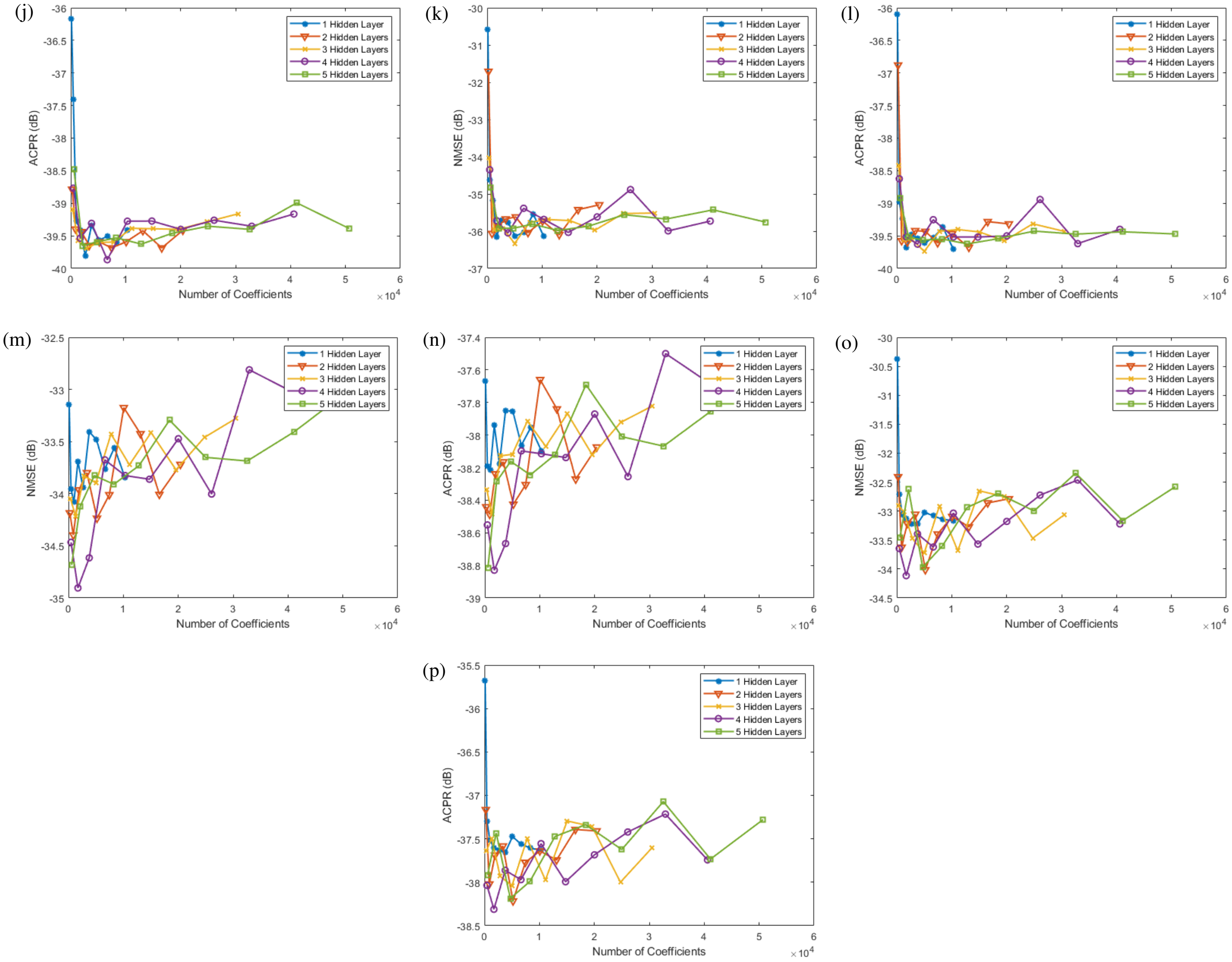


FIGURE 6. NMSE and ACPR vs. number of coefficients for each AF: (a)–(b) Sigmoid, (c)–(d) Tanh, (e)–(f) SElliot, (g)–(h) PElliot, (i)–(j) ReLU, (k)–(l) ELU, (m)–(n) GELU, and (o)–(p) Swich.

the number of hidden layers to one and systematically vary the number of neurons from 10 to 100, with an increment of 10. Once the maximum number of neurons is reached, we introduce an additional hidden layer and repeat the process, gradually increasing the number of neurons per layer until the maximum number of hidden layers is attained. This process is then repeated with another activation function. Each trained configuration of the NN-DPD, including the activation function, the number of hidden neurons N_h, and the number of hidden layers $L-2$, is saved for subsequent cascade deployment with the PA, as illustrated in Fig. 3. Finally, the performance of each model is evaluated using two key metrics, namely normalized mean square error (NMSE) and adjacent channel power ratio (ACPR), which are mathematically given by,

$$NMSE = \frac{\sum_n |y(n) - x(n)|^2}{\sum_n |x(n)|^2} \tag{18}$$

$$ACPR = \frac{\int_{adj.} Y(f)df}{\int_{ch.} Y(f)df} \tag{19}$$

where $y(n)$ is the PA's output signal, $x(n)$ the NN-DPD's input signal, and $Y(f)$ the Fourier transform of PA output signal. Notably, the training process of the neural network was configured with the following hyperparameters: 500 maximum epochs, a mini-batch size of 256, an initial learning rate of 10^{-4}, and the Adam optimization algorithm.

Figure 6 illustrates the relationship among NMSE, ACPR, the number of neurons N_h, and the number of hidden layers $L-2$. In Table 1, we present the best model for each AF, considering both ACPR and NMSE. These findings underscore that the performance of the NN-DPD depends not only on the configuration of layers and neurons but also on the selected AF. It could be noticed that the linearity requirements for the downlink imposed by the 5G NR standard are satisfied in terms of

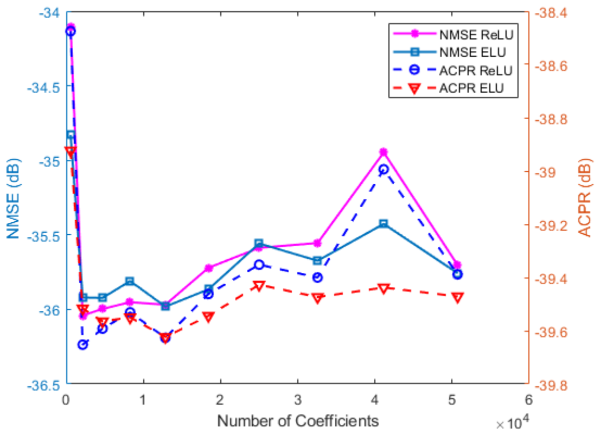


FIGURE 7. ACPR and NMSE using ReLU and ELU.

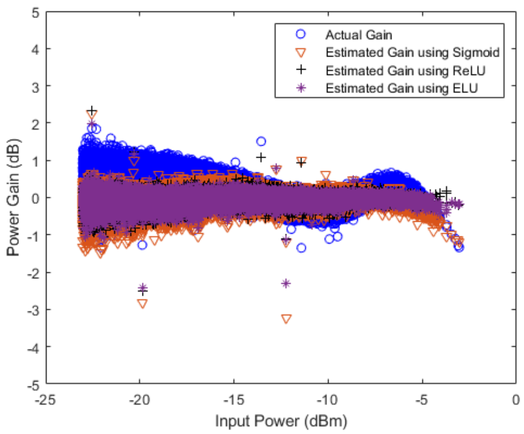


FIGURE 8. Gain characteristics before and after linearization.

TABLE 1. The best model obtained using each AF.

AF	Best Model	ACPR (dB) with DPD	ACPR (dB) No-DPD	NMSE (dB) with DPD	NMSE (dB) No-DPD
Sigmoid	$N_h = 40$ & $L - 2 = 1$	-36.81	-28.83	-31.84	-22.06
Tanh	$N_h = 20$ & $L - 2 = 5$	-39.25	-28.83	-35.42	-22.06
SElliot	$N_h = 100$ & $L - 2 = 1$	-39.36	-28.83	-35.54	-22.06
PElliot	$N_h = 70$ & $L - 2 = 1$	-39.1	-28.83	-35.1	-22.06
ReLU	$N_h = 50$ & $L - 2 = 1$	-39.8	-28.83	-36.45	-22.06
ELU	$N_h = 30$ & $L - 2 = 2$	-39.78	-28.83	-36.43	-22.06
GELU	$N_h = 20$ & $L - 2 = 4$	-38.8	-28.83	-34.12	-22.06
Swish	$N_h = 20$ & $L - 2 = 2$	-38.21	-28.83	-33.62	-22.06

NMSE but not in terms of ACPR. The latter could be improved with the combination of DPD and some power back-off. However, the authors wanted to compare the different AFs based on their performance without adding OPBO, for instance.

Building upon these findings, across most of the examined AFs, optimal results were achieved with two or less hidden layers, except for Tanh, GELU, and Swish. However, the most favorable outcomes emerged with ReLU and ELU. Notably, for models requiring less than 12850 coefficients, both ReLU and ELU demonstrated comparable performance. Nevertheless, with an increase in coefficients beyond this threshold, ELU surpassed ReLU by up to 0.5 dB, as depicted in Fig. 7. This discrepancy may be attributed to quantization-like errors stemming from an inadequate number of binary-like switch-on ReLU AF. It is worth noting that the optimal AF may vary with the type of NN architecture employed. For instance, different neural network types, such as convolutional neural networks (CNNs) or recurrent neural networks (RNNs), might benefit from distinct AF due to their structural differences.

In contrast to the findings in [13], our study identified models utilizing the sigmoid AF as the poorest performers. While an increase in the number of layers improves the modeling performance, these models still fall short of meeting the expected accuracy.

It is noteworthy that the AF SElliot could be a valuable option for designing a DPD-NN. While its performance slightly trails behind that of ReLU and ELU, it still outperforms other alternatives, including PElliot, and its modified version. This could be attributed to the selection of the hyperparameter a, which might require additional fine-tuning.

Figure 8 displays the AM/AM feature both without DPD (in blue) and with DPD based on ReLU, ELU, and Sigmoid. For clarity, we opted not to include other activation functions. However, it is notable that most activation functions exhibit similar behavior to Sigmoid, which performs poorly, especially at higher input power levels, unlike ReLU and ELU.

Figure 9 shows the power spectrum (PS) of the PA's output, comparing scenarios with and without DPD, using a 100 MHz source signal. Clearly, the choice of AF significantly influences the PA's linearization, with ELU and ReLU outperforming all other aforementioned AFs.

To benchmark the performance of the neural network-based DPD against traditional models, we conducted a comparative analysis using the Memory Polynomial (MP) model. The MP model is a widely adopted conventional DPD technique derived from the Volterra series. While the Volterra series provides a comprehensive framework for modeling nonlinear systems with memory effects, it is often computationally intensive due

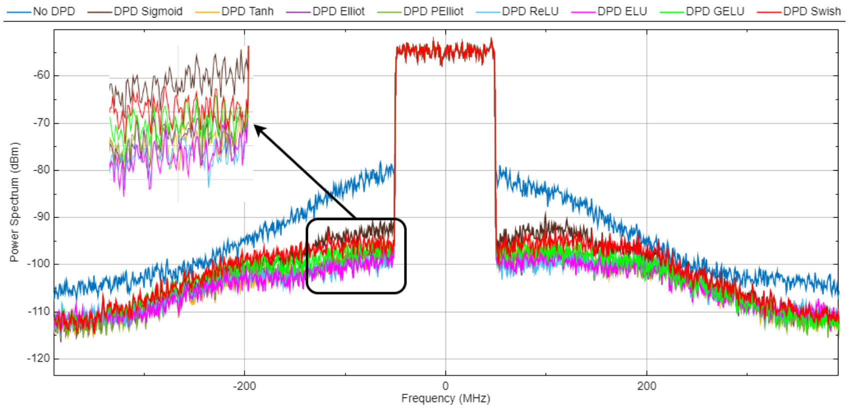


FIGURE 9. PS of linearized output of the different DPD models.

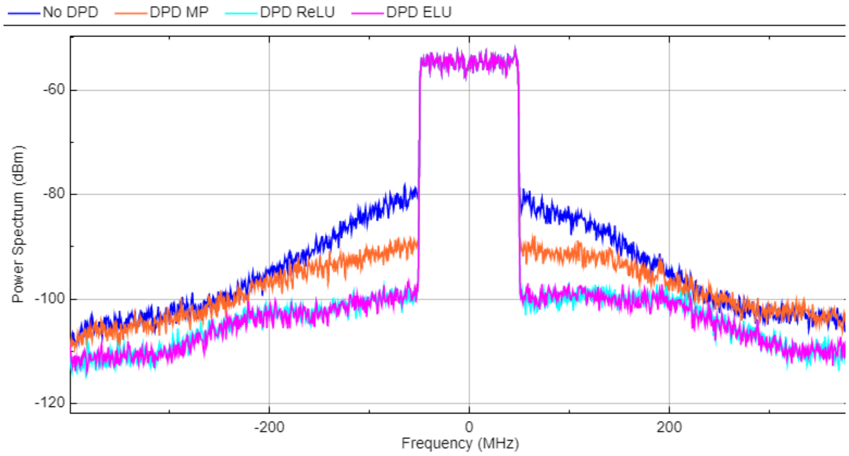


FIGURE 10. PS comparison of the PA output after linearization using MP-based DPD and NN-based DPD (with ReLU and ELU AFs).

to its complexity. The MP model simplifies this by considering only the most significant terms, making it more practical for implementation in DPD systems.

Figure 10 presents the power spectral density of the power amplifier output after linearization using two different DPD approaches.

As illustrated in Fig. 10, the NN-based DPD significantly outperforms the traditional MP-based DPD in suppressing spectral regrowth. The NN-based method reduces the spectral regrowth by approximately 10 dB compared to the MP-based DPD.

6. CONCLUSION

In this paper, we have investigated the effectiveness of various nonlinear AFs within NN-based DPD circuits for capturing the complex nonlinear characteristics of PAs. By examining eight carefully selected AFs and their interaction with neural network depth and width, we have conducted a comprehensive performance analysis using metrics such as NMSE and ACPR. Our results highlight the exponential linear unit activation function (ELU AF) as particularly advantageous, especially within deep neural network (DNN) architectures, compared to other considered AFs. This emphasizes the significance of AF selection in optimizing NN-based DPD systems for enhancing signal quality and mitigating nonlinear distortions in PA-driven communication systems.

REFERENCES

[1] Mohammady, S., R. Farrell, D. Malone, and J. Dooley, "Performance investigation of peak shrinking and interpolating the PAPR reduction technique for LTE-advance and 5G signals," *Information*, Vol. 11, No. 1, 20, 2019.

[2] Lopez-Bueno, D., T. Wang, P. L. Gilabert, and G. Montoro, "Amping up, saving power: Digital predistortion linearization strategies for power amplifiers under wideband 4G/5G burst-like waveform operation," *IEEE Microwave Magazine*, Vol. 17, No. 1, 79–87, 2015.

[3] Wood, J., *Behavioral Modeling and Linearization of RF Power Amplifiers*, Artech House, Boston; London, 2014.

[4] Ceylan, N., J.-E. Mueller, and R. Weigel, "Optimization of EDGE terminal power amplifiers using memoryless digital predistortion," *IEEE Transactions on Microwave Theory and Techniques*, Vol. 53, No. 2, 515–522, 2005.

[5] Kim, J. and K. Konstantinou, "Digital predistortion of wideband signals based on power amplifier model with memory," *Electronics Letters*, Vol. 37, No. 23, 1417–1418, 2001.

[6] Zhu, A., J. C. Pedro, and T. J. Brazil, "Dynamic deviation reduction-based Volterra behavioral modeling of RF power am-

plifiers," *IEEE Transactions on Microwave Theory and Techniques*, Vol. 54, No. 12, 4323–4332, 2006.

[7] Morgan, D. R., Z. Ma, J. Kim, M. G. Zierdt, and J. Pastalan, "A generalized memory polynomial model for digital predistortion of RF power amplifiers," *IEEE Transactions on Signal Processing*, Vol. 54, No. 10, 3852–3860, 2006.

[8] LeCun, Y., Y. Bengio, and G. Hinton, "Deep learning," *Nature*, Vol. 521, No. 7553, 436–444, 2015.

[9] Wang, Y., Y. Li, Y. Song, and X. Rong, "The influence of the activation function in a convolution neural network model of facial expression recognition," *Applied Sciences*, Vol. 10, No. 5, 1897, 2020.

[10] Liew, S. S., M. Khalil-Hani, and R. Bakhteri, "Bounded activation functions for enhanced training stability of deep neural networks on visual pattern recognition problems," *Neurocomputing*, Vol. 216, 718–734, 2016.

[11] Pedamonti, D., "Comparison of non-linear activation functions for deep neural networks on MNIST classification task," *ArXiv Preprint ArXiv:1804.02763*, 2018.

[12] Wang, D., M. Aziz, M. Helaoui, and F. M. Ghannouchi, "Augmented real-valued time-delay neural network for compensation of distortions and impairments in wireless transmitters," *IEEE Transactions on Neural Networks and Learning Systems*, Vol. 30, No. 1, 242–254, 2018.

[13] Hongyo, R., Y. Egashira, T. M. Hone, and K. Yamaguchi, "Deep neural network-based digital predistorter for Doherty power amplifiers," *IEEE Microwave and Wireless Components Letters*, Vol. 29, No. 2, 146–148, 2019.

[14] Yu, X., X. Fang, J. Shi, G. Lv, C. Wei, and J. Sui, "Deep neural network based stable digital predistortion using ELU activation for switchless class-G power amplifier," in *2024 IEEE MTT-S International Wireless Symposium (IWS)*, 1–3, Beijing, China, 2024.

[15] Jiang, Y., A. Vaicaitis, J. Dooley, and M. Leeser, "Efficient neural networks on the edge with FPGAs by optimizing an adaptive activation function," *Sensors*, Vol. 24, No. 6, 1829, 2024.

[16] Filicori, F. and G. Vannini, "Mathematical approach to large-signal modelling of electron devices," *Electronics Letters*, Vol. 27, No. 4, 357–359, 1991.

[17] Gilabert, P. L. and G. Montoro, "Look-up table implementation of a slow envelope dependent digital predistorter for envelope tracking power amplifiers," *IEEE Microwave and Wireless Components Letters*, Vol. 22, No. 2, 97–99, 2012.

[18] Fischer-Bühner, A., L. Anttila, M. D. Gomony, and M. Valkama, "Phase-normalized neural network for linearization of RF power amplifiers," *IEEE Microwave and Wireless Technology Letters*, Vol. 33, No. 9, 1357–1360, 2023.

[19] Lima, E. G., T. R. Cunha, and J. C. Pedro, "A physically meaningful neural network behavioral model for wireless transmitters exhibiting PM-AM/PM-PM distortions," *IEEE Transactions on Microwave Theory and Techniques*, Vol. 59, No. 12, 3512–3521, 2011.

[20] Zhang, Y., Y. Li, F. Liu, and A. Zhu, "Vector decomposition based time-delay neural network behavioral model for digital predistortion of RF power amplifiers," *IEEE Access*, Vol. 7, 91 559–91 568, 2019.

[21] Bawa, V. S. and V. Kumar, "Linearized sigmoidal activation: A novel activation function with tractable non-linear characteristics to boost representation capability," *Expert Systems with Applications*, Vol. 120, 346–356, 2019.

[22] Lohani, H. K., S. Dhanalakshmi, and V. Hemalatha, "Performance analysis of extreme learning machine variants with varying intermediate nodes and different activation functions," in *Cognitive Informatics and Soft Computing: Proceeding of CISC 2017*, 613–623, 2019.

[23] Koçak, Y. and G. U. Şiray, "New activation functions for single layer feedforward neural network," *Expert Systems with Applications*, Vol. 164, 113977, 2021.

[24] Nair, V. and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, 807–814, 2010.

[25] Trottier, L., P. Giguere, and B. Chaib-Draa, "Parametric exponential linear unit for deep convolutional neural networks," in *2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA)*, 207–214, Cancun, Mexico, 2017.

[26] Hendrycks, D. and K. Gimpel, "Gaussian error linear units (GELUS)," *ArXiv Preprint ArXiv:1606.08415*, 2016.

[27] Ramachandran, P., B. Zoph, and Q. V. Le, "Searching for activation functions," *ArXiv Preprint ArXiv:1710.05941*, 2017.

[28] MathWorks, "Power amplifier characterization," June 2024. [Online]. Available: https://www.mathworks.com/help/comm/ug/power-amplifier-characterization.html.

[29] Fawzy, A., S. Sun, T. J. Lim, and Y. X. Guo, "An efficient deep neural network structure for RF power amplifier linearization," in *2021 IEEE Global Communications Conference (GLOBECOM)*, 1–6, Madrid, Spain, 2021.