Hyperparameter Optimization of Two-Hidden-Layer Neural Networks for Power Amplifiers Behavioral Modeling Using Genetic Algorithms

Siqi Wang, Morgan Roger, Julien Sarrazin, *Member, IEEE*, and Caroline Lelandais-Perrault

***Abstract*—Neural networks (NNs) are efficient techniques for behavioral modeling of power amplifiers (PAs). This letter proposes a genetic algorithm to determine the optimal hyperparameters of the NN model for a PA. Different activation functions are compared. The necessary number of training epochs is also studied to get an optimal solution with a significantly reduced computational complexity. Experimental measurements on a PA with different signals validate the NN models determined by the proposed method.**

***Index Terms*—Behavioral modeling, genetic algorithm, neural networks (NNs), nonlinear distortion, power amplifiers (PAs).**

I. Introduction

THE power amplifiers (PAs) performance is crucial in modern wireless communication systems [1]. Modeling the behavior of the PA is crucial to estimate and analyze its performance [2]–[4].

Neural networks (NNs) have been widely used in PA behavioral modeling with similar or better performance compared with the classical analytic models [4]–[9]. A real-valued time-delay NN model has been proposed in [5] by decomposing the baseband input signal into in-phase and quadrature component (I/Q) signals. The output signal is buffered to the input layer in [6], which improved the modeling accuracy. Instead, Wang *et al.* [7] added nonlinear terms of the input signal to the input layer. Recently, some studies [8], [9] applied NN models with two or more hidden layers for better modeling accuracy.

The memory depth, the number of layers, and the number of hidden neurons in each layer have a great impact on the performance of the NN model [10]. They are often determined in an empirical way or by sweeping search [6], [7].

Some optimization algorithms, such as hill climbing [11], [12] and particle swarm optimization [13], have been applied to determine the optimal structure of a Volterra-series-based model. However, the hyperparameter optimization of the NN has always been a challenge.

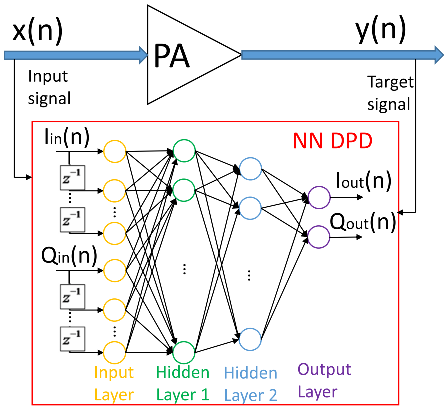

Fig. 1. PA behavioral modeling with NN.

In this letter, we propose to optimize the hyperparameters of a two-hidden-layer (2HL) NN using genetic algorithms (GAs). To the best of our knowledge, this is the first time that the NN hyperparameter problem for the PA behavioral modeling is addressed. The number of training epochs is optimized to reduce the running time.

II. Multilayer Neural Network Model

The PA modeling system architecture is shown in Fig. 1. In order to train the network, we take the PA input signal $x(n)$ as the NN input signal and the PA output signal $y(n)$ as the target signal. They are decomposed into real-valued I/Q signals

$$\begin{aligned} x(n) &= I_{\text{in}}(n) + i\,Q_{\text{in}}(n)\\ y(n) &= I_{\text{out}}(n) + i\,Q_{\text{out}}(n). \end{aligned} \tag{1}$$

The input layer is composed of $I_{\text{in}}(n)$, $Q_{\text{in}}(n)$, and their delayed samples. The memory depth is L which corresponds to the number of delay taps. We denote the number of neurons in the lth hidden layers by N_l. The output $s_{k(l)}$ of the kth neuron in the lth layer can be then expressed as a function of the inputs t_j

$$s_{k(l)} = f\left(\sum_{j=1}^{N_{l-1}} w_{jk} t_j + b_k\right) \tag{2}$$

where w_{jk} is the weight, b_k is the bias, and $f(\cdot)$ is the activation function. In this letter, we test two different activation

Manuscript received September 29, 2019; accepted October 25, 2019. Date of publication November 12, 2019; date of current version December 4, 2019. *(Corresponding author: Siqi Wang.)*

S. Wang, M. Roger, and C. Lelandais-Perrault are with the Group of Electrical Engineering Paris, CNRS, CentraleSupélec, Univ. Paris-Sud, Université Paris-Saclay, Sorbonne Université, 91192 Gif-sur-Yvette, France (e-mail: siqi.wang@centralesupelec.fr; morgan.roger@centralesupelec.fr; caroline.lelandais-perrault@centralesupelec.fr).

J. Sarrazin is with the Laboratory of Electronics and Electromagnetism, Sorbonne Université, F-75005 Paris, France (e-mail: julien.sarrazin@upmc.fr).

Color versions of one or more of the figures in this article are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/LMWC.2019.2950801

Algorithm 1 Overview of GA

```
Set generation counter $q = 1$;
$\mathbf{P}$ is divided into $\mathbf{P}_1$, $\mathbf{P}_2$, $\mathbf{P}_3$;
while $q \leqslant Q$ do
    Compute fitness of all individuals in $\mathbf{P}$;
    Selection: $\mathbf{P}_1$ stays as offspring $\mathbf{O}_1$;
    Crossover: $\mathbf{P}_2$ are subdivided and regrouped to
    create offspring $\mathbf{O}_2$ ;
    Mutation: $\mathbf{P}_3$ are randomly changed to create
    offspring $\mathbf{O}_3$;
    $q = q + 1$;
    $\mathbf{P} = \mathbf{O}_1 + \mathbf{O}_2 + \mathbf{O}_3$;
end
```

functions: tangent sigmoid (tansig) and rectified linear unit (ReLU) as follows [9]:

$$\operatorname{tansig}(x) = \frac{1 - e^{-2x}}{1 + e^{-2x}} \tag{3}$$

$$\operatorname{ReLU}(x) = \begin{cases} x & \text{if } x \geq 0 \\ 0 & \text{otherwise.} \end{cases} \tag{4}$$

The activation function of the output layer is linear.

The training process is realized by backward computation using the Levenberg–Marquardt algorithm [14]. The input and target signals are randomly divided into three segments for training, validating, and testing. The process can be composed of two phases as follows.

1) At each epoch, we take the first segment to compute the values of w_{jk} and b_k and validate the computed NN with the second segment.
2) After the training, we test the NN with the third segment.

The modeling performance is evaluated with the normalized-mean-square error (NMSE) between the measured PA output $y(n)$ and the model output $\hat{y}(n)$

$$\mathrm{NMSE}_{\mathrm{dB}} = 10 \log_{10} \left[\frac{\sum_n |y(n) - \hat{y}(n)|^2}{\sum_n |y(n)|^2} \right]. \tag{5}$$

The hyperparameters of the 2HL NN model to be optimized in this letter are the memory depth L and the number of neurons in the first and the second hidden layers N_1 and N_2, respectively.

III. Genetic Algorithm

The GAs are iterative stochastic search mechanisms based on the idea of Darwinian natural selection theory. After generational evolution, better genes are kept. In this algorithm, fitness is defined to evaluate whether a gene is good. In our implementation, the genes are represented by NN hyperparameters (L, N_1, and N_2) in binary bits. We train the corresponding NN model to evaluate its fitness, which is given by the error defined in (5) between the target signal and the output of the NN model in the test phase (phase 2 in Section II). We use $\mathbf{P}$ to denote the group of parents and q as the number of generations. The maximum generation number is set to Q.

The offspring of each generation is created by the parents of the previous generation through selection, crossover, and mutation as described in Algorithm 1. The parents are listed from the best to the worst and then divided into three groups—the first group $\mathbf{P}_1$ has the best 5% individuals, the second group $\mathbf{P}_2$ has the best 80% of the remaining individuals, and the third group $\mathbf{P}_3$ has the rest. **Selection**: we keep $\mathbf{P}_1$ completely into the following generation as the first offspring group $\mathbf{O}_{1(q)}$. **Crossover**: we mix the genes of $\mathbf{P}_2$ to constitute the second offspring group $\mathbf{O}_{2(q)}$. **Mutation**: we randomly change the genes of $\mathbf{P}_3$ and create the third offspring group $\mathbf{O}_{3(q)}$. The parents of the following generation $\mathbf{P}_{(q+1)}$ is the union of $\mathbf{O}_{1(q)}$, $\mathbf{O}_{2(q)}$, and $\mathbf{O}_{3(q)}$. The iteration continues till the stall condition is met.

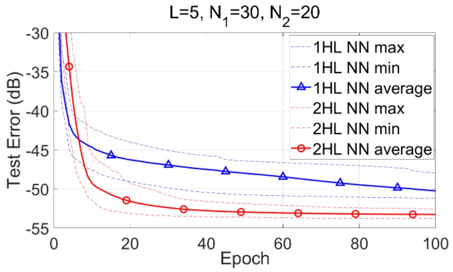

Fig. 2. Test error versus epoch in training NN model with tansig as activation function, HL NN.

IV. Experimental Results

In order to validate the proposed method, we test with data acquired from an HMC409LP4E PA fabricated by Analog Devices, Norwood, MA, USA. Its nominal gain at 3.5 GHz is 31 dB, and the saturated output power is 32.5 dBm. We generate a stimulus at the carrier frequency of 3.5 GHz in the PC Workstation and feed it to the PA through an arbitrary waveform generator with 10-GHz sampling frequency. The input and output baseband signals of 15 000 samples are time-aligned after downsampling to 120 MHz for behavioral modeling. A 20-MHz-long-term evolution (LTE) signal and a 40-MHz wireless local area network (WLAN) signal are used as stimuli in the tests. Their peak-to-average power ratios are 8 and 12 dB, respectively.

To determine the necessary number of training epochs with the aim of significantly reducing the search algorithm running time, we first study the convergence of the NN model training error as a function of the epoch number, as shown in Fig. 2. The training procedure is standard and uses the data randomly divided into a training set (70%), a validation set (15%), and a test set (15%). The activation function is tansig. In order to study the separation of convergences of different NN models, we make the tests of a 1HL NN and a 2HL NN, which have surely different performances [8]. For the sake of simplicity, we choose a 1HL NN with $L = 5$ and $N_1 = 30$ to test. Since the NN is trained with 70% data randomly selected from the signal and the initial values of w_{jk} and b_k are random, its modeling accuracy is a random variable. We independently train and test the NN model for 50 times. With 50 realizations, we calculate the min, max, and average of their test errors at each epoch.

We can see that the performance convergence of these two NN models slows down after 10 epochs, and a 2HL NN almost always outperforms a 1HL NN after 20 epochs. In other words, using a 2HL NN model allows us to significantly reduce the

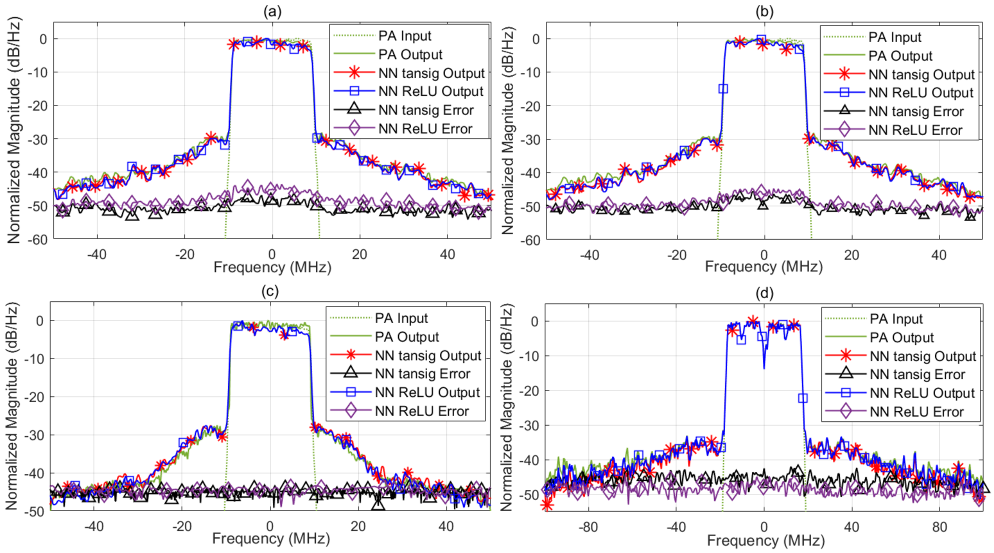

Fig. 3. Experimental results: (a) Output spectra of the PA and the determined NN model M_{20} for 20-MHz LTE signal. (b) Output spectra of the PA and the determined NN model M_{100} for 20-MHz LTE signal. (c) Output spectra of the PA and the determined NN model M_{20} for 20-MHz LTE signal at saturation region. (d) Output spectra of the PA and the determined NN model for 40-MHz WLAN signal.

number of epochs needed to get better performance/training time tradeoff. Since our objective is to find the optimal NN hyperparameters instead of giving the precise minimum error, we propose to reduce the epoch number to 20 in the GA search considering the variance of the NN performance.

In the search algorithm, we set the population size to 10 for each generation and the maximum number of generations to $Q = 6$. When computing fitness, we evaluate the performance by training the corresponding NN model with 20 and 100 epochs separately.

Based on the data measured with the PA, we applied the proposed GA with 20 and 100 training epochs. The final solutions of the NN model determined by the GA are denoted by M_{20} and M_{100}, respectively. We obtain their final performances by training the M_{20} and M_{100} with 1000 epochs. The output spectra of the optimal NN models trained with 1000 epochs are shown in Fig. 3(a) and (b), respectively, and the corresponding modeling errors. The results in Fig. 3(c) validate the NN model performance in the saturation region by increasing the input power 2 dB. We also make a test with a 40-MHz bandwidth WLAN signal, as shown Fig. 3(d), which confirms the effectiveness of our proposed method.

Table I gives the quantitative performances of NN models determined by our proposed method with LTE and WLAN signals. We take a generalized memory polynomial model as a reference whose optimal structure is determined using the method in [11]. With the LTE and WLAN signals, the NMSE values obtained are −34.7 and −37.0 dB, respectively, which are similar to the performance of the NN model determined by a GA. The algorithm running time T is measured under MATLAB with an Intel(R) Core(TM) i7-3770 CPU @3.40 GHz and 16-GB RAM.

The values in "fitness" are the NMSE values between the model output and measured PA output during the training process. The values in "NMSE" are the NMSE values obtained by testing the GA-determined model after 1000-epoch training.

TABLE I

BEHAVIORAL MODELING PERFORMANCE OF NN GIVEN BY ALGORITHM 1

Model	Ac Fcn	L	N_1	N_2	T (min)	Fitness (dB)	NMSE (dB)
M_{20}	tansig	7	46	49	151	-54.1	-33.7
	ReLU	7	48	32	124	-54.5	-33.3
M_{100}	tansig	9	42	44	876	-55.2	-32.4
	ReLU	9	43	42	503	-55.6	-32.1
M_{wlan20}	tansig	9	26	37	121	-48.4	-34.2
	ReLU	10	48	33	110	-49.5	-36.1

Ac Fcn: Activation Function; T: Running time of algorithm

For the solution model determined by the GA, we calculate its average NMSE value with 50 tests. The proposed algorithm with 20 epochs seems to be a good tradeoff between modeling accuracy and running time. Increasing the number of epochs to 100 does not bring sufficient improvement in performance while the running time is more than four times longer.

The activation function has an impact on the algorithm's running time. The search with ReLU needs less time since the computational complexity of ReLU is lower than tansig. The performance of ReLU is better for the WLAN signal.

V. CONCLUSION

In this letter, we propose a GA to optimize the hyperparameter of an NN-based PA behavioral model. To the best of our knowledge, this is the first time this optimization problem is addressed without exhaustive search. The algorithm is validated with measurement data acquired from a PA with different signals. We study the convergence of the NN model training error as a function of the epoch number and significantly reduce the optimization running time while keeping similar performances. Different activation functions of the NN model are tested. ReLU leads to less running time.

References

[1] F. M. Ghannouchi, "Power amplifier and transmitter architectures for software defined radio systems," *IEEE Circuits Syst. Mag.*, vol. 10, no. 4, pp. 56–63, 4th Quart., 2010.

[2] C. Fager, T. Eriksson, F. Barradas, K. Hausmair, T. Cunha, and J. C. Pedro, "Linearity and efficiency in 5G transmitters: New techniques for analyzing efficiency, linearity, and linearization in a 5G active antenna transmitter context," *IEEE Microw. Mag.*, vol. 20, no. 5, pp. 35–49, May 2019.

[3] F. M. Ghannouchi and O. Hammi, "Behavioral modeling and predistortion," *IEEE Microw. Mag.*, vol. 10, no. 7, pp. 52–64, Dec. 2009.

[4] J. Cai, C. Yu, L. Sun, S. Chen, and J. B. King , "Dynamic behavioral modeling of RF power amplifier based on time-delay support vector regression," *IEEE Trans. Microw. Theory Techn.*, vol. 67, no. 2, pp. 533–543, Feb. 2019.

[5] T. Liu, S. Boumaiza, and F. M. Ghannouchi, "Dynamic behavioral modeling of 3G power amplifiers using real-valued time-delay neural networks," *IEEE Trans. Microw. Theory Techn.*, vol. 52, no. 3, pp. 1025–1033, Mar. 2004.

[6] F. Mkadem and S. Boumaiza, "Physically inspired neural network model for RF power amplifier behavioral modeling and digital predistortion," *IEEE Trans. Microw. Theory Techn.*, vol. 59, no. 4, pp. 913–923, Apr. 2011.

[7] D. Wang, 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 Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 1, pp. 242–254, Jun. 2019.

[8] P. Jaraut, M. Rawat, and F. M. Ghannouchi, "Composite neural network digital predistortion model for joint mitigation of crosstalk, I/Q imbalance, nonlinearity in MIMO transmitters," *IEEE Trans. Microw. Theory Techn.*, vol. 66, no. 11, pp. 5011–5020, Nov. 2018.

[9] R. Hongyo, Y. Egashira, T. M. Hone, and K. Yamaguchi, "Deep neural network-based digital predistorter for Doherty power amplifiers," *IEEE Microw. Wireless Compon. Lett.*, vol. 29, no. 2, pp. 146–148, Feb. 2019.

[10] B. Nakisa, M. N. Rastgoo, A. Rakotonirainy, F. Maire, and V. Chandran, "Long short term memory hyperparameter optimization for a neural network based emotion recognition framework," *IEEE Access*, vol. 6, pp. 49325–49338, 2018.

[11] S. Wang, M. A. Hussein, O. Venard, and G. Baudoin, "A novel algorithm for determining the structure of digital predistortion models," *IEEE Trans. Veh. Technol.*, vol. 67, no. 8, pp. 7326–7340, Aug. 2018.

[12] S. Wang, M. A. Hussein, O. Venard, and G. Baudoin, "Optimal sizing of two-stage cascaded sparse memory polynomial model for high power amplifiers linearization," *IEEE Trans. Microw. Theory Techn.*, vol. 66, no. 9, pp. 3958–3965, Sep. 2018.

[13] A. Abdelhafiz, L. Behjat, and F. M. Ghannouchi, "Generalized memory polynomial model dimension selection using particle swarm optimization," *IEEE Microw. Wireless Compon. Lett.*, vol. 28, no. 2, pp. 96–98, Feb. 2018.

[14] M. T. Hagan and M. B. Menhaj, "Training feedforward networks with the Marquardt algorithm," *IEEE Trans. Neural Netw.*, vol. 5, no. 6, pp. 989–993, Nov. 1994.