IEEE Access
Multidisciplinary ⁝ Rapid Review ⁝ Open Access Journal

Received April 13, 2020, accepted April 28, 2020, date of publication May 6, 2020, date of current version May 15, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2991890

A Novel Deep Neural Network Topology for Parametric Modeling of Passive Microwave Components

JING JIN[1,2], (Student Member, IEEE), FENG FENG[2], (Member, IEEE), JIANAN ZHANG[1,2], (Student Member, IEEE), SHUXIA YAN[3], WEICONG NA[4], (Member, IEEE), AND QIJUN ZHANG[2], (Fellow, IEEE)

[1]School of Microelectronics, Tianjin University, Tianjin 300072, China
[2]Department of Electronics, Carleton University, Ottawa, ON K1S5B6, Canada
[3]School of Electronics and Information Engineering, Tianjin Polytechnic University, Tianjin 300387, China
[4]Faculty of Information Technology, Beijing University of Technology, Beijing 100124, China

Corresponding author: Feng Feng (fengfeng@doe.carleton.ca)

This work was supported in part by the Scientific Research Project of Tianjin Education Commission under Grant 2017KJ088, in part by the National Natural Science Foundation of China under Grant 61901010, in part by the China Postdoctoral Science Foundation Funded Project under Grant 2019M650404, in part by the Beijing Municipal Natural Science Foundation under Grant 4204092, in part by the Chaoyang District Postdoctoral Scientific Research Funding under Grant 2019ZZ-9, and in part by the Natural Sciences and Engineering Research Council of Canada under Grant RGPIN-2017-06420.

ABSTRACT Artificial neural network technique has gained recognition as a powerful technique in microwave modeling and design. This paper proposes a novel deep neural network topology for parametric modeling of microwave components. In the proposed deep neural network, the outputs are S-parameters. The inputs of the proposed model include geometrical variables and the frequency. We divide the hidden layers in the proposed deep neural network topology into two parts. Hidden layers in Part I handle both the geometrical inputs and the frequency inputs while hidden layers in Part II only handle the geometrical inputs. In this way, more training parameters are utilized to specifically learn the relationship between the S-parameters and the geometrical variables, which are more complicated than that between the S-parameters and the frequency. The purpose is to reduce the total number of training parameters in the deep neural network model. New formulations are derived to calculate the derivatives of the error function with respect to training parameters in the deep neural network. Taking advantage of the calculated derivatives, we propose an advanced two-stage training algorithm for the deep neural network. The two-stage training algorithm can determine the number of hidden layers in both parts during the training process and guarantee that the proposed deep neural network model can achieve the required model accuracy. The proposed deep neural network can achieve similar model accuracy with less training parameters compared to the commonly used fully connected neural network. The proposed technique is demonstrated by two microwave parametric modeling examples.

INDEX TERMS Deep neural network, microwave components, neural network training, parametric modeling.

I. INTRODUCTION

Parametric modeling of microwave components plays an important role in the area of electromagnetic (EM)-based microwave design. Parametric models can be developed from the relationship between the EM behavior of microwave components and the geometrical variables. The inputs of the parametric model include geometrical variables and the frequency, and the outputs are EM responses (such as S-parameters) of the microwave component. The developed parametric models can provide fast and accurate prediction of the EM responses for different values of geometrical parameters and subsequently can be implemented in high-level circuit and system designs. It can accelerate the EM-based design process by avoiding repetitive EM simulations, which are usually time-consuming.

The associate editor coordinating the review of this manuscript and approving it for publication was Dušan Grujić.

Different modeling methods, such as neural network technique [1], [2], polynomial-based surrogate modeling technique [3], Kriging technique [4], and support vector machine (SVM) [5], can be used to build parametric models for microwave components. Polynomial-based surrogate modeling technique, Kriging technique, and SVM have good generalization capability for the situation where the training data are limited [3]. Neural network technique is applicable to the situation where the size of training data is large. In this paper we develop a neural network-based method for parametric modeling of passive microwave components where extensive training data are used.

Artificial neural network (ANN) modeling technique is an efficient parametric modeling technique for EM behaviors of microwave components. The developed ANN models can provide fast solutions to the tasks they have learned [6]. ANN technique has been utilized as a powerful technique for microwave modeling and design optimization [6]–[12]. In [13]–[15], microwave parametric models are developed by the knowledge-based neural network techniques where the neural networks are combined with prior knowledge. In addition, there are applications reported in nonlinear microwave device modeling [16]–[21], power amplifier modeling [22]–[24], multiphysics parametric modeling [25], and microwave component design [26]–[28].

Most of these reported applications utilize shallow neural networks. In recent years, deep neural network with many hidden layers has gain recognition in the neural network community learning complicated relationships in large data sets [29]–[32]. Outstanding results have been achieved by deep neural networks in various challenging areas, such as speech recognition [33], image recognition [34], sentiment analysis [35], language processing [36], and machine translation [37]. In [28], the deep neural network technique is introduced into the microwave modeling field to address the challenges because of high-dimensional inputs. The application examples presented in [28] show the advantages of the deep neural network over the shallow neural network. The deep neural network technique in [28] can be trained to learn the training data in a high-dimensional space. In the area of parametric modeling of microwave components, there are situations where the relationship between the model outputs and the geometrical parameters is more complex than the relationship between the model outputs and the frequency. In this paper, we propose a novel deep neural network topology for parametric modeling to specifically address this situation. The proposed technique can represent the input-output relationship using less training parameters than the commonly used fully connected neural network.

In this paper, we propose a novel deep neural network topology specifically to reduce the total number of training parameters for neural network-based parametric modeling of microwave components. The inputs of the parametric model include geometrical variables and the frequency, and the outputs are S-parameters. In the proposed deep neural network topology, the inputs are divided into two sets, i.e., geometrical inputs and the frequency input. We divide the hidden layers in the proposed deep neural network into two parts. Hidden layers in Part I handle both the geometrical inputs and the frequency input while hidden layers in Part II only handle the geometrical inputs. In this way, more training parameters (including weights and biases in the proposed deep neural network) are utilized to specifically learn the relationship between the S-parameters and the geometrical variables, which are more complex than that between the S-parameters and the frequency. The purpose is to reduce the total number of training parameters in the deep neural network model while maintain the model accuracy. In order to train the proposed deep neural network, we propose new computations of the derivatives of the error function with respect to training parameters. Taking advantage of the calculated derivatives, we propose an advanced two-stage training algorithm for training the deep neural network. By using the two-stage training algorithm, the number of hidden layers in both parts can be determined during the training process and the model accuracy can reach the required threshold after training. The novel deep neural network topology can achieve similar model accuracy using less training parameters than the fully connected neural network.

This paper is organized as follows. In Section II, we first describe the structure of the proposed deep neural network topology. The feedforward computations from model inputs to outputs are derived for the proposed deep neural network. The computations of the derivatives of the error function with respect to training parameters are derived for the training algorithm. An advanced two-stage training algorithm is proposed to train the novel deep neural network model. The proposed deep neural network technique is applied to two parametric modeling examples of microwave filters in Section III. Finally, Section IV concludes the paper.

II. PROPOSED DEEP NEURAL NETWORK TOPOLOGY FOR PARAMETRIC MODELING OF MICROWAVE COMPONENTS

A. THE PROPOSED DEEP NEURAL NETWORK TOPOLOGY

We propose a novel deep neural network topology to develop parametric models for microwave components more efficiently. Parametric models of microwave components can be developed from the information of EM responses as functions of geometrical parameters and the frequency. The inputs of the parametric model are the geometrical parameters of microwave components and the frequency. The outputs of the parametric model are the EM responses such as the scattering parameters. The relationship between the EM responses and the geometrical parameters is usually more complex while the relationship between the EM responses and the frequency is less complex. In order to represent the relationship between the EM responses and the geometrical parameters and the frequency more efficiently, we propose to use more hidden layers to learn the relationship between the EM responses and the geometrical parameters and use less hidden layers to learn that between the EM responses and the frequency.

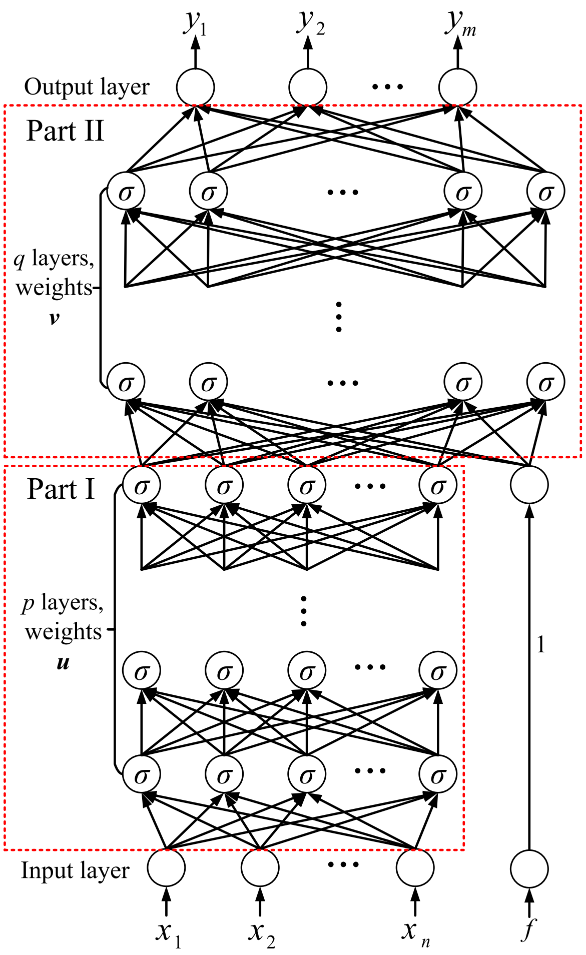


FIGURE 1. The proposed deep neural network topology. Sigmoid function is used as activation function in the neuron with σ, and linear activation function is used in the neuron without σ.

Figure 1 shows the proposed deep neural network topology. In the novel deep neural network topology for parametric modeling of microwave components, the inputs are divided into geometrical inputs and the frequency input. Let $\boldsymbol{x} = [x_1\ x_2\ x_3\ \ldots\ x_n]^T$ represent a vector of the geometrical inputs of the proposed deep neural network model, where n is the number of the geometrical inputs. $\boldsymbol{x}$ contains the physical geometrical variables of a microwave component, such as the length and the width of the EM structure. Let f represent the frequency input of the proposed deep neural network. Let $\boldsymbol{y} = [y_1\ y_2\ y_3\ \ldots\ y_m]^T$ represent a vector of the model outputs, where m is the number of the outputs. The vector $\boldsymbol{y}$ contains the S-parameters of a microwave component. As shown in Figure 1, we divide the hidden layers in the deep neural network into two parts. Sigmoid function is utilized as the activation function for hidden neurons in both parts. We intend to feed the frequency input directly into the first layer in Part I. To make the topology of the proposed deep neural network tidier, we add a hidden neuron that can achieve a unit mapping from its input to its output. Let p be the number of hidden layers in Part I. The vector containing the weights in Part I is defined as $\boldsymbol{u}$. Let q be the number of hidden layers in Part II. The vector containing the weights in Part II is defined as $\boldsymbol{v}$. The proposed deep neural network model can be defined as

$$\boldsymbol{y} = g(\boldsymbol{x}, f, \boldsymbol{u}, \boldsymbol{v}) \tag{1}$$

where g is the function representing the input-output relationship of the proposed deep neural network model. The output of the developed model is a function of geometrical variables $\boldsymbol{x}$ and the frequency variable f. The frequency f is only one input variable for the model. When using the proposed model to represent the EM behavior of a microwave component, the developed model needs to be calculated at multiple different frequency points by changing the value of f.

B. THE FEEDFORWARD COMPUTATION OF THE PROPOSED DEEP NEURAL NETWORK MODEL

Let h and l be the indexes of layers in Part I and Part II, respectively, where $h = 1, 2, \ldots, p$ and $l = 1, 2, \ldots, q$. Let N_1^h and N_2^l represent the number of hidden neurons in the hth layer in Part I and that in the lth layer in Part II, respectively. Let $z_1^{h,i}$ and $z_2^{l,i}$ be the output of the ith hidden neuron in the hth layer of Part I and in the lth layer of Part II, respectively. $z_1^{h,i}$ and $z_2^{l,i}$ can be calculated by the feedforward process.

We first consider the hidden layers in Part I. Let u_{ij}^h represent the weight between the jth neuron in the $(h-1)$th layer and the ith neuron in the hth layer for $h = 2, 3, \ldots, p$, $j = 1, 2, \ldots, N_1^{h-1}$, $i = 1, 2, \ldots, N_1^h$. For $h = 1$, u_{ij}^1 represents the weight between the jth input neuron and the ith neuron in the first layer of Part I. We introduce an extra weight parameter u_{i0}^h to represent the bias for ith neuron of hth layer. Thus, the vector $\boldsymbol{u}$ for Part I include u_{ij}^h, $h = 1, 2, \ldots, p$, $j = 0, 1, \ldots, N_1^{h-1}$, $i = 1, 2, \ldots, N_1^h$. By using the feedforward computation, $z_1^{h,i}$ of Part I can be calculated as

$$z_1^{h,i} = \begin{cases} \sigma(\displaystyle\sum_{j=1}^{n} u_{ij}^h x_j + u_{i0}^h) & \text{if } h = 1 \\ \sigma(\displaystyle\sum_{j=1}^{N_1^{h-1}} u_{ij}^h z_1^{h-1,j} + u_{i0}^h) & \text{if } h = 2, \ldots, p \end{cases} \tag{2}$$

where $\sigma(\cdot)$ is the activation function used in the hidden neurons of Part I, i.e., the sigmoid function.

Then we proceed to Part II. Since the first layer in Part II connect to both the pth layer of Part I and the single hidden neuron connecting with the frequency input, we add the single hidden neuron into the pth layer of Part I for the description convenience. The outputs of the neurons in the extended pth layer of Part I becomes

$$z_1^{p,i} = \begin{cases} \sigma(\displaystyle\sum_{j=1}^{N_1^{p-1}} u_{ij}^p z_1^{p-1,j} + u_{i0}^p) & \text{if } i = 1, 2, \ldots, N_1^p \\ f & \text{if } i = N_2^p + 1 \end{cases} \tag{3}$$

The number of neurons in the extended pth layer of Part I becomes $N_1^p + 1$.

Let v_{ij}^l represent the weight between the jth neuron in the $(l-1)$th layer and the ith neuron in the lth layer for $l = 2, 3, \ldots, q, j = 1, 2 \ldots, N_2^{l-1}, i = 1, 2, \ldots, N_2^l$. For $l = 1$, v_{ij}^1 represents the weight between the jth neuron in the extended pth layer of Part I and the ith neuron in the first layer of Part II. We also introduce an extra weight parameter v_{i0}^l to represent the bias for the ith neuron of lth layer. Thus, the vector $\boldsymbol{v}$ for Part II include v_{ij}^l, $l = 1, 2, \ldots, q$, $j = 0, 1, \ldots, N_2^{l-1}$, $i = 1, 2, \ldots, N_2^l$. By using the feedforward computation, $z_2^{l,i}$ of Part II can be calculated as

$$z_2^{l,i} = \begin{cases} \sigma(\sum\limits_{j=1}^{N_1^p+1} v_{ij}^l z_1^{p,j} + v_{i0}^l) & \text{if } l = 1 \\ \sigma(\sum\limits_{j=1}^{N_2^{l-1}} v_{ij}^l z_2^{l-1,j} + v_{i0}^l) & \text{if } l = 2, \ldots, q \end{cases} \tag{4}$$

where $z_1^{p,j}$, $j = 1, 2, \ldots, N_1^p + 1$ is calculated in Equation (3).

To represent the weights between the qth layer of Part II and the output layer, the definition of v_{ij}^l is extend to the output layer, i.e., v_{ij}^l, $l = q + 1$ represents the weight between the jth neuron in the qth layer of Part II and the ith neuron in the output layer. After the feedforward computation, we can extract the outputs of the proposed deep neural network model from the output layer as

$$y_i = \sum_{j=1}^{N_2^q} v_{ij}^{q+1} z_2^{q,j}, \quad i = 1, \ldots, m \tag{5}$$

Then we need to train the proposed deep neural network model with the training data by adjusting values of the training parameters.

C. PROPOSED TRAINING ALGORITHM FOR THE NEW DEEP NEURAL NETWORK

The deep neural network model need to be trained with the training data. Let $\boldsymbol{d}_{kt}$ be defined as the desired model outputs corresponding to inputs $[\boldsymbol{x}_k^T, f_t]^T$. The training data can be presented as pairs of $([\boldsymbol{x}_k^T, f_t]^T, \boldsymbol{d}_{kt})$, $k = 1, 2, \ldots, N_g$, $t = 1, 2, \ldots, N_f$, where N_g represents the total number of geometric samples and N_f represents the total number of frequency samples in the frequency band of interest. Specifically, for microwave parametric modeling, $\boldsymbol{x}_k$ is a set of geometrical parameters of the microwave component, f_t is a frequency in the designed frequency band, and $\boldsymbol{d}_{kt}$ should be the S-parameters corresponding to $\boldsymbol{x}_k$ at the frequency f_t. The standard error function is used to evaluate the performance of the proposed deep neural network model in the training process [38]. The error function is defined as

$$E(\boldsymbol{u}, \boldsymbol{v}) = \frac{1}{2mN_gN_f} \sum_{k=1}^{N_g} \sum_{t=1}^{N_f} \sum_{j=1}^{m} (y_j(\boldsymbol{x}_k, f_t, \boldsymbol{u}, \boldsymbol{v}) - d_{kt}^j)^2 \tag{6}$$

where $y_j(\boldsymbol{x}_k, f_t, \boldsymbol{u}, \boldsymbol{v})$ is the jth output of the proposed deep neural network model corresponding to $\boldsymbol{x}_k$ and f_t, and d_{kt}^j is the jth element of $\boldsymbol{d}_{kt}$.

Gradient-based training method is one of the most efficient methods to train a neural network model [6]. When training the deep neural network model with the gradient-based training method, the weights are adjusted according to the derivatives of the error function formulated in Equation (6) with respect to the weights. We propose a method to calculate the derivatives by extending the back propagation (BP) concept for multilayer perceptron (MLP) to our novel deep neural network structure [38]. New formulations are proposed to compute the derivatives of the error function with respect to the training parameters. The derivatives can guide the gradient-based training for the proposed deep neural network structure.

The error function in Equation (6) is the average of the errors of the model for all geometric samples at all sampled frequencies. For the kth geometric sample at the tth sampled frequency, the total error of all outputs of the model can be computed by [38]

$$E_{kt} = \frac{1}{2} \sum_{j=1}^{m} (y_j(\boldsymbol{x}_k, f_t, \boldsymbol{u}, \boldsymbol{v}) - d_{kt}^j)^2 \tag{7}$$

Let $\gamma_1^{h,i}$, $h = 1, 2, \ldots, p$ and $\gamma_2^{l,i}$, $l = 1, 2, \ldots, q$ represent the total input to the ith neuron in hth layer of Part I and that to the ith neuron in lth layer of Part II, respectively. For the description convenience, the definition of $\gamma_2^{l,i}$ is extended to the output layer, i.e., $\gamma_2^{q+1,i}$ represents the total input to the ith neuron of output layer. Let $\delta_1^{h,i}$ and $\delta_2^{l,i}$ be the local derivative of E_{kt} with respect to $\gamma_1^{h,i}$ and $\gamma_2^{l,i}$, respectively, i.e., $\delta_1^{h,i} = \frac{\partial E_{kt}}{\partial \gamma_1^{h,i}}$, $h = 1, 2, \ldots, p$ and $\delta_2^{l,i} = \frac{\partial E_{kt}}{\partial \gamma_2^{l,i}}$, $l = 1, 2, \ldots, q + 1$. Starting from $\delta_2^{l,i}$, $l = q + 1$ at the output layer of the deep neural network model, the BP algorithm will propagate this local derivative backward through the hidden layers to the input layer. Since neurons in the output layer utilize linear activation functions, the $\delta_2^{l,i}$, $l = q + 1$ at the output layer can be calculated as

$$\delta_2^{l,i} = (y_i(\boldsymbol{x}_k, f_t, \boldsymbol{u}, \boldsymbol{v}) - d_{kt}^i), \quad l = q + 1 \tag{8}$$

Subsequently, this local derivative of the output layer will be propagated backward to the hidden layers in Part II by the BP algorithm. The local derivative $\delta_2^{l,i}$ in these hidden layers in Part II can be derived as

$$\delta_2^{l,i} = (\sum_{j=1}^{N_2^{l+1}} \delta_2^{l+1,j} v_{ji}^{l+1}) z_2^{l,i} (1 - z_2^{l,i}), \quad l = q, q-1, \ldots, 2, 1 \tag{9}$$

Then, the local derivative will be propagated backward from hidden layers in Part II to those in Part I by the BP algorithm. For the pth hidden layer in Part I that is directly connected to the first hidden layer in Part II, the local derivative $\delta_1^{h,i}, h = p$ can be derived as

$$\delta_1^{h,i} = (\sum_{j=1}^{N_2^1} \delta_2^{1,j} v_{ji}^1) z_1^{h,i}(1 - z_1^{h,i}), \quad h = p \tag{10}$$

For the other hidden layers in Part I (i.e., $h = p - 1, p - 2, \ldots, 1$), the local derivative $\delta_1^{h,i}$ is calculated using similar method as the standard back propagation [38]

$$\delta_1^{h,i} = (\sum_{j=1}^{N_1^{h+1}} \delta_1^{h+1,j} u_{ji}^{h+1}) z_1^{h,i}(1 - z_1^{h,i}), \quad h = p - 1, \ \ldots, \ 2, \ 1 \tag{11}$$

Once we obtain the local derivatives for all layers in the proposed deep neural network, we can compute the derivatives of E_{kt} with respect to the weight parameters by

$$\frac{\partial E_{kt}}{\partial v_{ij}^l} = \begin{cases} \delta_2^{l,i} z_2^{l-1,j}, & l = q + 1, \ q, \ q - 1, \ \ldots, 2 \\ \delta_2^{l,i} z_1^{p,j}, & l = 1 \end{cases} \tag{12}$$

$$\frac{\partial E_{kt}}{\partial u_{ij}^h} = \begin{cases} \delta_1^{h,i} z_1^{h-1,j}, & h = p, \ p - 1, \ \ldots, \ 3, \ 2 \\ \delta_1^{h,i} x_j, & h = 1 \end{cases} \tag{13}$$

where $\delta_1^{h,i}$ and $\delta_2^{l,i}$ are calculated using Equations (8)-(11); $z_2^{l-1,j}$, $z_1^{p,j}$, and $z_1^{h-1,j}$ are calculated using Equations (2)-(4). With the proposed Equations (8)-(13), we can compute the derivative of the error function with respect to each weight parameter. Then the weight parameters are updated based on the calculated derivatives [6].

We propose a two-stage training algorithm for the proposed deep neural network. The derivative information is used to guide the gradient-based training process. To simplify the training algorithm, we predetermine the number of hidden neurons per layer in Part I and Part II according to experience before the training process. Usually, if the geometrical input dimension is relatively high and/or the variation ranges of the model inputs are relatively large, we choose larger number of hidden neurons. If the geometrical input dimension is relatively low and/or the variation ranges of the model inputs are relatively small, we choose smaller number of hidden neurons [28]. Using the proposed two-stage training algorithm, the number of hidden layers in Part I and Part II can be determined during the training process.

In Stage I, the number of layers in Part II is set to be one, i.e., $q = 1$. In this stage, we determine the number of layers in Part I, i.e., the value of p by changing the value of p in the training process. Usually we start from $p = 1$. A deep neural network with one hidden layer in Part II and p hidden layers in Part I is trained to reduce the training error as much as possible. After training, we check the model accuracy. If the model accuracy can satisfy the requirement, we stop the training process and do not need Stages II. If the model accuracy cannot satisfy the requirement, a new hidden layer is added to Part I again and again. We stop adding new layers to Part I until the model accuracy can satisfy the requirement or until the training error cannot be reduced even if a new layer is added in Part I. After each layer is added in Part I, we train the deep neural network model and compute the training error. We define E_{be} and E_{tr} as the training errors of the deep neural network model before and after adding the recently added layer in Part I, respectively. Let E_{re} represent the required error threshold of the model. If $E_{tr} < E_{be}$ and $E_{tr} > E_{re}$, it means that adding a layer in Part I can reduce the training error. However, the reduced training error still cannot reach the required error threshold. In this case, the deep neural network model is in underlearning state. A new hidden layer need to be added to Part I again. If $E_{tr} \geq E_{be}$ even after many training iterations, it means that adding more layers in Part I cannot reduce the training error anymore. In this case, the last added layer in Part I is deleted and the total number of hidden layers in Part I is determined as p. After determining the value of p, we proceed to Stage II.

In Stage II, in order to further reduce the training error of the deep neural network, we begin to add hidden layers in Part II. We add one new hidden layer to Part II again and again until the model accuracy can achieve the accuracy requirement. After training of Stage II, the number of hidden layers in Part II can be determined as q. The final deep neural network has p hidden layers in Part I and q hidden layers in Part II. It can represent the input-output relationships of parametric models of microwave components accurately.

D. PROCESS OF DEVELOPING THE DEEP NEURAL NETWORK PARAMETRIC MODEL

We summarize the development process of the deep neural network parametric model of microwave components as follows.

Step 1) Perform EM simulation to generate training and test data using random sampling method. Fix the number of hidden neurons in each layer of Part I and Part II. Initialize $p = 1, q = 1$.

Step 2) Train the neural network with p hidden layers in Part I and q hidden layers in Part II. Calculate the training error E_{tr} and the test error E_{te}. If $p = 1$, go to Step 4); else, go to Setp 3).

Step 3) Compare the values of E_{tr} and E_{be}. If $E_{tr} < E_{be}$, go to Step 4); else, go to Step 7).

Step 4) Compare the values of E_{tr} and E_{re}. If $E_{tr} \leq E_{re}$, go to Step 5); else if $E_{tr} > E_{re}$, set $E_{be} = E_{tr}$, add one hidden layer to Part I, i.e., $p = p + 1$, and go to Step 2).

Step 5) Compare the value of E_{te} and E_{te}. If $E_{te} > E_{re}$, go to Step 6); else if $E_{te} \leq E_{re}$, stop the training process of the deep neural network model.

Step 6) Train the deep neural network model with more training data, calculate the training error E_{tr} and the test error E_{te}, and go to Step 4).

Step 7) Delete the last added layer in Part I, i.e., $p = p - 1$, and train the deep neural network model with one hidden layer in Part II and p hidden layers in Part I to

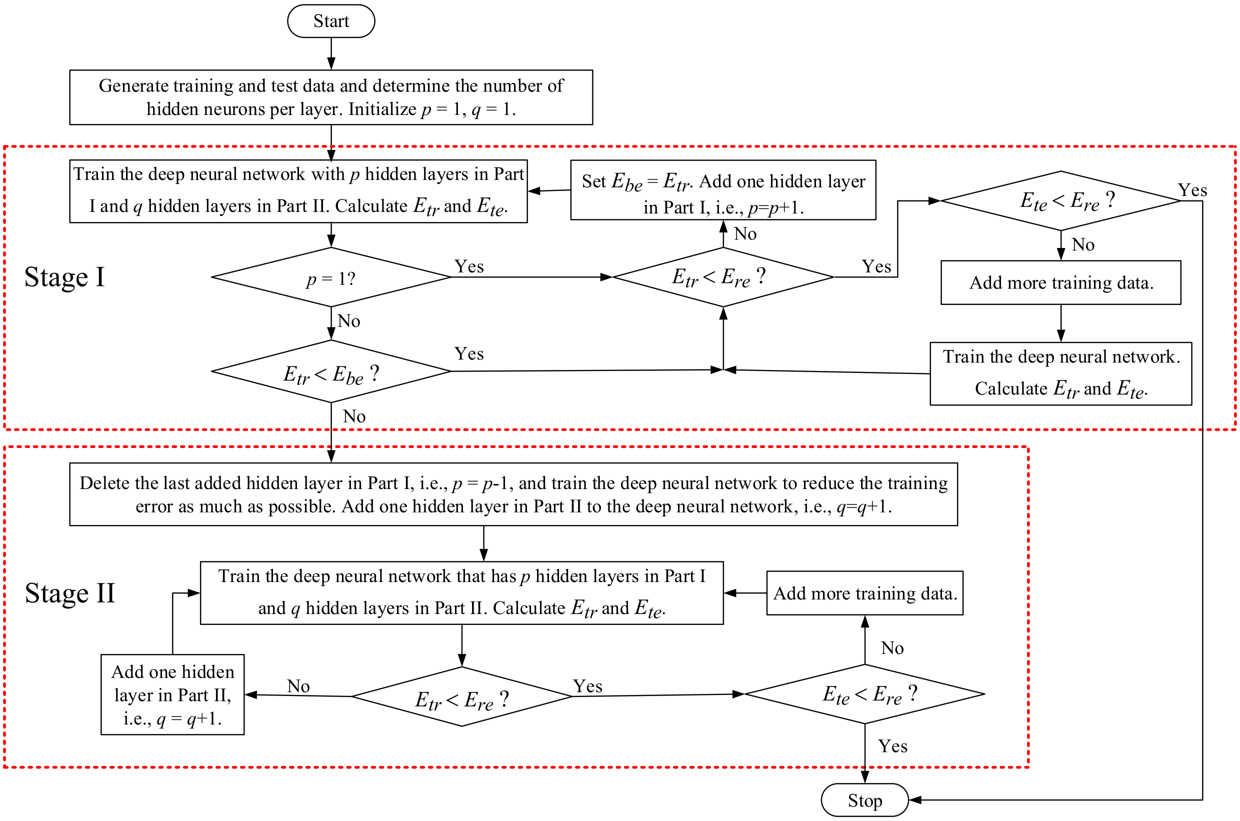

Start
Generate training and test data and determine the number of hidden neurons per layer. Initialize p = 1, q = 1.
Stage I
Train the deep neural network with p hidden layers in Part I and q hidden layers in Part II. Calculate E_{tr} and E_{te}.
Set E_{be} = E_{tr}. Add one hidden layer in Part I, i.e., p=p+1.
$E_{te} < E_{re}$?
Yes
No
p = 1?
Yes
$E_{tr} < E_{re}$?
No
Yes
Add more training data.
No
$E_{tr} < E_{be}$?
Yes
Train the deep neural network. Calculate E_{tr} and E_{te}.
No
Delete the last added hidden layer in Part I, i.e., p = p-1, and train the deep neural network to reduce the training error as much as possible. Add one hidden layer in Part II to the deep neural network, i.e., q=q+1.
Stage II
Train the deep neural network that has p hidden layers in Part I and q hidden layers in Part II. Calculate E_{tr} and E_{te}.
Add more training data.
Add one hidden layer in Part II, i.e., q = q+1.
No
$E_{tr} < E_{re}$?
Yes
$E_{te} < E_{re}$?
No
Yes
Stop

FIGURE 2. The flow diagram of overall development process of the proposed deep neural network parametric model.

reduce the training error as much as possible. Add one hidden layer in Part II to the trained model, i.e., $q = q + 1$.

Step 8) Train the neural network with p hidden layers in Part I and q hidden layers in Part II. Calculate the training error E_{tr} and the test error E_{te}.

Step 9) Compare the values of E_{tr} and E_{re}. If $E_{tr} \leq E_{re}$, go to Step 10); else if $E_{tr} > E_{re}$, add one hidden layer in Part II, i.e., $q = q + 1$, and go to Step 8).

Step 10) Compare the value of E_{te} and E_{te}. If $E_{te} > E_{re}$, add more training data and go to Step 8); else if $E_{te} \leq E_{re}$, stop the training process.

The flow diagram of overall development process of the proposed deep neural network parametric model is shown in Figure 2. According to the two-stage training algorithm, Stage I consists of Steps 2) to 6), and Stage II consists of Steps 7) to 10).

E. DISCUSSION

In addition to the sigmoid function, other activation functions, such as tanh function, the rectified linear unit (ReLU), and so on, can also be adopted in the proposed deep neural network structure. If another activation function is used, the model structure and training algorithm of the proposed technique will not be changed. Only the calculation of model outputs and the derivatives of the error function with respect to the weights need to be revised accordingly. The idea of deriving the calculations of model outputs and derivatives is similar to what we have presented in Section II. In this paper, we choose sigmoid function as the activation function for the hidden neurons because it is one of the most commonly used activation functions for neural network [38].

In the two-stage training, it is necessary to check the test errors in both stages and adjust the training data accordingly. This is because in Stage II, the new layers are added to the neural network trained from Stage I. We keep the trained layers and weights from Stage I and further train the whole neural network after adding new layers in Stage II. The trained weights from Stage I provide a good starting point for Stage II. During the training process, we need to check the test errors in both stages. If we do not check the test error in Stage I, the neural network trained from Stage I may be in the overlearning state. In this case, the weights trained from Stage I cannot provide a good starting point for Stage II. It will make the training of Stage II harder. Therefore, we need to check the test errors in both stages and adjust the training data accordingly.

The number of hidden neurons is changed during the training process since the number of hidden layers is changed. In our proposed two-stage training algorithm, to make the training process simpler, we predetermine the number of neurons per layer and adjust the total number of training parameters by changing the number of layers in the training process. If the randomly initialized weights are close to a local minimum, and the local minimum cannot satisfy the required error threshold, we will add a new layer with random weights and train the neural network with the newly added layer. The added layer makes it possible for the neural network to get out of the current local minimum and achieve a better result.

The neural network topology proposed in this paper is meant to reduce the number of training parameters in the neural network model while maintain the model accuracy. It is specifically designed to address the situation where the relationship between the outputs and geometrical parameters is more complex than the relationship between the outputs and the frequency. For the application situations where the relationship between the model outputs and geometrical parameters is less complex than the relationship between the model outputs and the frequency, the proposed technique may not be that effective. The developed neural network model is accurate within the training range, and is unreliable if it is used outside the training range. An extrapolation technique to guide the neural network outside the training range is reported in [39] to address this issue. The proposed deep neural network method is for parametric modeling of passive microwave components. The parametric modeling of nonlinear components can be a very interesting future direction.

III. EXAMPLES

A. PARAMETRIC MODELING OF A THREE-POLE H-PLANE FILTER

In this example, we develop a parametric model for a three-pole H-plane filter [40], whose structure is shown in Figure 3, using the proposed deep neural network topology. The cross section of the waveguide where the filter is constructed is 19.05 mm × 9.525 mm (WR-75). For this filter, the relationship between the S-parameters and the geometrical parameters is more complex than the relationship between the S-parameters and the frequency. The proposed deep neural network structure is suitable for developing the parametric model of this filter. Four geometrical variables are used as geometrical inputs to the model of this example, i.e., $\boldsymbol{x} = [L_1, L_2, W_1, W_2]^T$, and the real and imaginary parts of S_{21} are used as model outputs, i.e., $\boldsymbol{y} = [Re(S_{21}), Im(S_{21})]^T$.

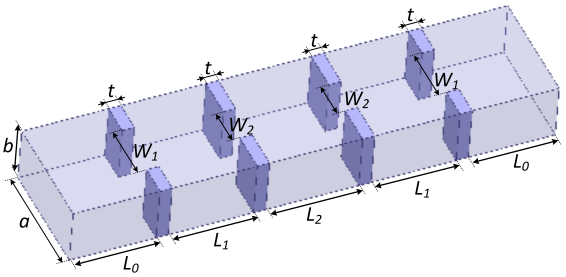


FIGURE 3. Structure of the three-pole H-plane filter. The four geometrical parameters of this filter are $x = [L_1, L_2, W_1, W_2]^T$, with $a = 19.05$ mm, $b = 9.525$ mm, and $t = 2.0$ mm.

When developing parametric models for the microwave components, the outputs are S-parameters of the components. In this filter example, we choose one of the S-parameters (S_{21}) as the model output to demonstrate the proposed technique. The proposed modeling method is also suitable when other S-parameters besides S_{21} are used as model outputs. The model is developed in similar modeling procedure when using other S-parameters as model outputs.

We apply the novel deep neural network topology to two different cases as defined in Table 1. In Case 1, the geometrical parameters change in a narrower range, while in Case 2 the geometrical parameters change in a wider range. For both cases of this example, the frequency range is from 11.5 GHz to 12.5 GHz. For both cases, we perform EM simulations to generate the training and test data using random sampling method. Suppose N_{tr} and N_{te} are the total numbers of training and test data needed for developing the parametric model. The value of N_{tr} changes in the training process according to the two-stage training algorithm. In order to generate the training and test data, we first generate N_{tr} and N_{te} sets of randomly distributed geometrical samples in the variation range of geometrical parameters. For each set of geometrical parameters, we perform full-wave EM simulation using HFSS to simulate the filter at all chosen frequency samples. Suppose we have N_f frequency samples, where N_f equals to 101 for this example. We will get N_f sets of training/test data for each set of geometrical parameters. Each set of training/test data is composed of the geometrical parameters, one frequency, and the S-parameters at the specific frequency. After simulations at all the geometrical samples and frequencies, we will get $N_{tr} \times N_f$ sets of training data and $N_{te} \times N_f$ sets of test data.

TABLE 1. Definition of training and testing data for the three-pole *H*-plane filter example.

Geometrical Variables		Data Range		
		Min	Nominal	Max
Case 1	L_1 (mm)	13.3	14.0	14.7
	L_2 (mm)	14.44	15.2	15.96
	W_1 (mm)	8.55	9.0	9.45
	W_2 (mm)	5.7	6.0	6.3
Case 2	L_1 (mm)	11.2	14.0	16.8
	L_2 (mm)	12.16	15.2	18.24
	W_1 (mm)	7.2	9.0	10.8
	W_2 (mm)	4.8	6.0	7.2

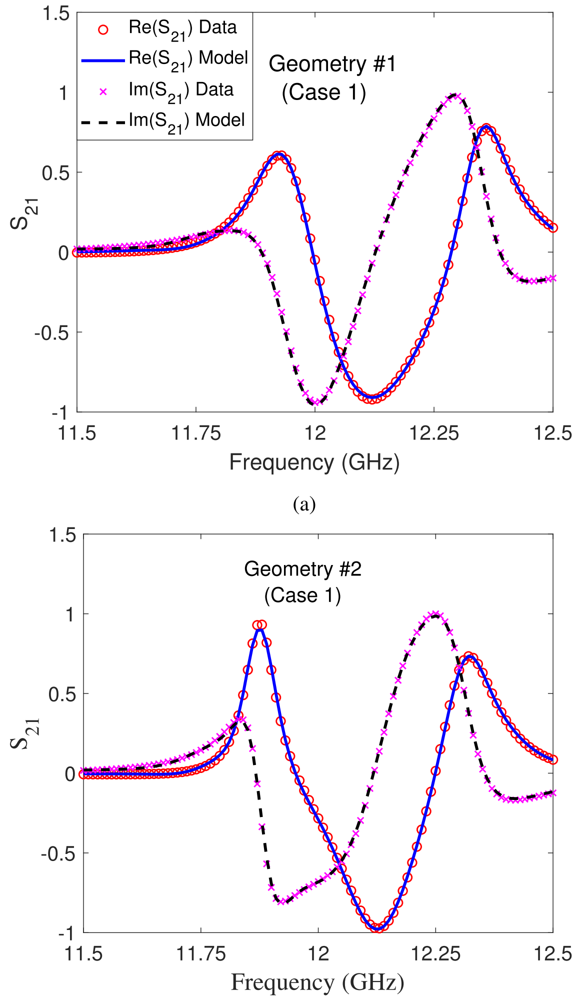
We develop the parametric models in both cases for the three pole H-Plane filter using the proposed deep neural network technique. The training and test error threshold for this example is set to be 2%. The training and test error threshold is a user-defined parameter. It is defined according to the accuracy requirement of the developed model for different application examples. Smaller error threshold means higher model accuracy. We defined the error threshold to be 2% because it is accurate enough for this filter example.

TABLE 2. Model accuracy comparisons of the proposed deep neural network and conventional 3-layer MLP for the three-pole *H*-plane filter example.

Neural Network Topology		Details of Hidden Layers	Total Number of Weight Parameters	Training Error	Test Error
Case 1	Proposed deep neural network	Part I: 2 hidden layers with 15 neurons per layer Part II: 2 hidden layers with 5 neurons per layer	432	1.93%	1.94%
	3-layer MLP	54 hidden neurons	434	2.36%	2.39%
		150 hidden neurons	1202	1.98%	2.00%
Case 2	Proposed deep neural network	Part I: 2 hidden layers with 25 neurons per layer Part II: 2 hidden layers with 10 neurons per layer	1162	1.78%	1.88%
	3-layer MLP	145 hidden neurons	1162	3.47%	3.43%
		200 hidden layers	1602	3.22%	3.21%

In Case 1, the number of hidden neurons in Part I and Part II are determined to be 5 and 15, respectively. We start from one hidden layer in each part and change the number of hidden layers based on the training algorithm. When the training process is finished, the final structure is composed of two layers with 5 hidden neurons per layer in Part I and two layers with 15 hidden neurons per layer in Part II. The total number of training parameters, including weight parameters and biases, is 432. We start from using 20200 training data and gradually add the training data. When the number of training data reaches 80800, the training and test errors achieve the required threshold. The average training error and test error of the proposed deep neural network parametric model are 1.93% and 1.94%, respectively. Two different geometrical samples in Case 1 are used to test the modeling results as shown in Figure 4. The two test samples are randomly chosen from all test data. From Figure 4, we can see that the S-parameters provided by the deep neural network model can match the desired S-parameters very well. For comparison purpose, we also develop a parametric model for this example in Case 1 using the fully connected 3-layer MLP [6]. The comparison of the modeling results are shown in Table 2. A 3-layer MLP with 54 hidden neurons, i.e., MLP: 5-54-2, is used to learn the learn the input-output relationship of the parametric model. The total number of training parameters is similar to the number of those in the proposed deep neural network model. The average training and test errors of the 3-layer MLP with 54 hidden neurons are 2.36% and 2.39%, respectively. Then we increase the number of hidden neurons of the 3-layer MLP to be 150, i.e., MLP: 5-150-2, where there are 1202 training parameters in total. The average training and test errors of the 3-layer MLP with 150 hidden neurons are 1.98% and 2.00%, respectively.

In Case 2, the number of hidden neurons in Part I and Part II are 10 and 25, respectively. The number of hidden layers in each part starts from one and changes based on the training algorithm. When the training process is finished, the final structure is composed of two layers with 10 hidden neurons per layer in Part I and two layers with 25 hidden neurons per layer in Part II. The total number of training parameters is 1162. For this case, we start from using 101000 training data and gradually add the training data. The final size of the training data is 303000 when the two-stage training process is finished. After training the deep neural network, the average training error and test error are 1.78% and 1.88%, respectively. The modeling results at two random geometrical samples in Case 2 are shown in Figure 5. The deep neural network parametric model can provide accurate S-parameter



(a)

(b)

FIGURE 4. Three-pole H-plane filter example in Case 1: modeling results at two different geometrical samples (a) $x = [13.52,\ 14.62,\ 9.30,\ 6.15]^T$ (mm), (b) $x = [13.67,\ 14.90,\ 8.81,\ 6.27]^T$ (mm).

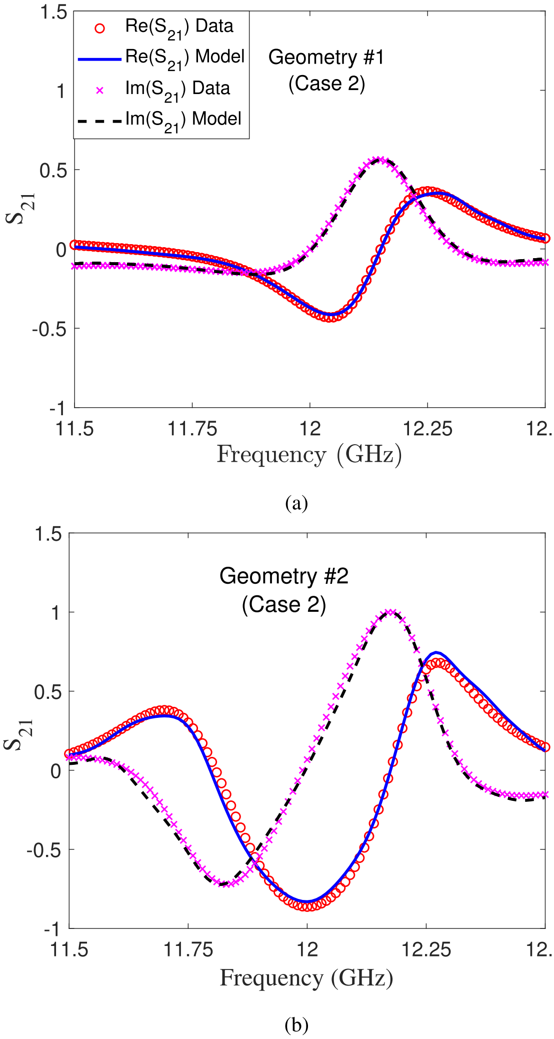


FIGURE 5. Three-pole H-plane filter example in Case 2: modeling results at two different geometrical samples (a) $x = [13.23, 16.33, 9.37, 6.87]^T$ (mm), (b) $x = [13.15, 14.58, 10.51, 6.67]^T$ (mm).

solutions, which match the desired S-parameters very well. For comparison purpose, we also use the traditional fully connected neural network to develop parametric models for this example in Case 2. The comparison of the modeling results are shown in Table 2. Two 3-layer MLPs with different number of hidden neurons are used to build the parametric model for Case 2. The numbers of hidden neurons of these two neural networks are 145 and 200, respectively. The MLP with 145 hidden neurons has 1162 training parameters, which is as same as the number of those in the proposed deep neural network model. The average training and test errors of the 3-layer MLP with 145 hidden neurons are 3.47% and 3.43%, respectively. When the number of hidden neurons is increased to 200, the total number of training parameters in the neural network increases to 1602. The average training and test errors are reduced to 3.22% and 3.21%, respectively.

From the test results shown in Figure 4 and Figure 5, the S-parameters provided by the proposed deep neural network model can match the desired S-parameters very well for both cases. According to the comparisons of the modeling results shown in Table 2, our proposed deep neural network topology can achieve better training and test errors for both cases compared to the 3-layer MLPs with the similar number of training parameters. For the narrower range (Case 1), adding more hidden neurons in the 3-layer MLP can reduce the training and test errors to the required error threshold. For the wider range (Case 2), adding hidden neurons in the 3-layer MLP can reduce the training and test errors slightly. However, it is much harder to reduce the training and test errors to the required error threshold. In other words, for narrower range, both the proposed deep neural network topology and the conventional 3-layer MLP can satisfy the model accuracy requirement. The proposed deep neural network technique is more efficient than the 3-layer MLP because it needs less training parameters than the 3-layer MLP to achieve similar model accuracy. For the wider range, the proposed deep neural network can achieve the required accuracy efficiently while the 3-layer MLP cannot reach the required accuracy even with much more training parameters than the proposed technique.

B. PARAMETRIC MODELING OF A FIFTH-ORDER WAVEGUIDE BANDPASS FILTER

Parametric models of a fifth-order waveguide bandpass filter in two cases are developed using the proposed deep neural network topology in this example. The structure of the filter is shown in Figure 6 [15]. d_1, d_2, and d_3 are the distances from the irises to the waveguide wall. z_1, z_2, and z_3 are the distances between two adjacent irises. The thicknesses of the irises are t_1, t_2, and t_3. The parametric model for this example has nine geometrical inputs, i.e., $\boldsymbol{x} = [d_1, d_2, d_3, z_1, z_2, z_3, t_1, t_2, t_3]^T$, and two outputs, i.e., $\boldsymbol{y} = [Re(S_{21}), Im(S_{21})]^T$.

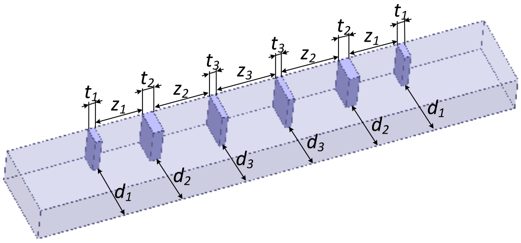


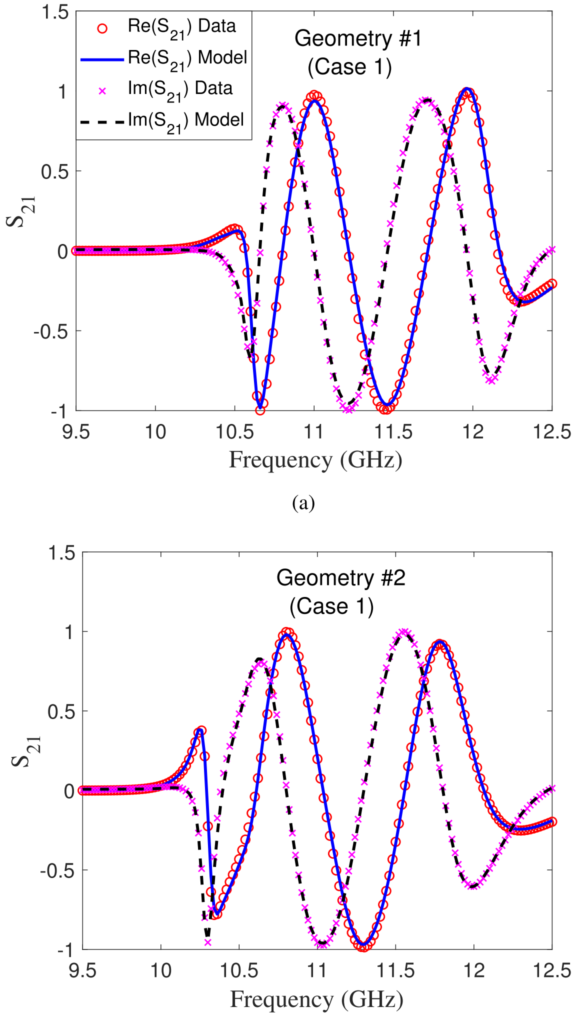
FIGURE 6. Structure of the fifth-order waveguide bandpass filter. The nine geometrical parameters of this filter are $\boldsymbol{x} = [d_1, d_2, d_3, z_1, z_2, z_3, t_1, t_2, t_3]^T$.

We apply the novel deep neural network topology to two different cases as defined in Table 3. The frequency range for both cases of this example is from 9.5 GHz to 12.5 GHz. Total 151 frequency samples in the range are used for this example. For both cases, we perform EM simulations to generate the training and test data using random sampling method.

The training and test error threshold for this example is set to be 2%. In Case 1, we use 8 hidden neurons per layer in Part I and 15 hidden neurons per layer in Part II. The number of

TABLE 3. Definition of training and testing data for the fifth-order waveguide bandpass filter example.

Geometrical Variables		Data Range		
		Min	Nominal	Max
Case 1	d_1 (mm)	13.27	14.745	16.22
	d_2 (mm)	11.9	12.94	13.98
	d_3 (mm)	11.4	11.88	12.36
	z_1 (mm)	10.69	11.88	13.07
	z_2 (mm)	13.33	14.03	14.73
	z_3 (mm)	14.18	14.93	15.68
	t_1 (mm)	0.97	1.61	2.25
	t_2 (mm)	1.82	2.6	3.38
	t_3 (mm)	0.87	1.45	2.03
Case 2	d_1 (mm)	12.53	14.745	16.96
	d_2 (mm)	11.39	12.94	14.49
	d_3 (mm)	11.17	11.88	12.59
	z_1 (mm)	10.1	11.88	13.66
	z_2 (mm)	12.98	14.03	15.08
	z_3 (mm)	13.81	14.93	16.05
	t_1 (mm)	0.64	1.61	2.58
	t_2 (mm)	1.43	2.6	3.77
	t_3 (mm)	0.58	1.45	2.32



(a)

(b)

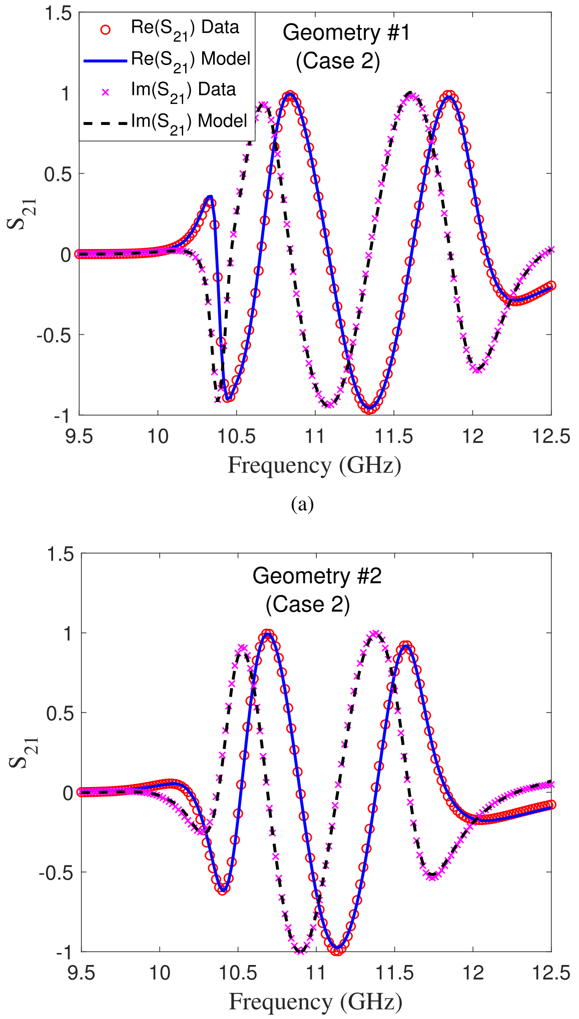
FIGURE 7. Fifth-order waveguide bandpass filter example in Case 1: modeling results at two different geometrical samples (a) $x = [13.70, 11.94, 11.20, 12.22, 13.96, 14.63, 1.29, 3.67, 2.10]^T$ **(mm), (b)** $x = [14.43, 12.09, 11.83, 11.21, 14.36, 15.15, 2.35, 2.79, 2.27]^T$ **(mm).**

hidden layers in each part is adjusted according to the training algorithm. When the training process is finished, the final structure is composed of two layers with 8 hidden neurons per layer in Part I and two layers with 15 hidden neurons per layer in Part II. The initial training data size for Case 1 of this example is 75500. After the two-stage training process, the total number of training data used for this case is 302000. The developed model with 574 training parameters can achieve a 1.72% training error and a 1.75% test error. Training and test errors are the average errors of all training data and test data, respectively. Two random geometrical samples in Case 1 are used to show the modeling results in Figure 7. From Figure 7, we can see that the S-parameters provided by the deep neural network model can match the desired S-parameters very well. The comparison results of the proposed technique and MLPs with different number of hidden neurons are shown in Table 4. The MLP (i.e., MLP: 10-44-2) with same number of training parameters as the proposed technique has a 2.28% training error and a 2.30% test error. When the total number of training parameters increases to 1224, the average training and test errors of the MLP (i.e., MLP: 10-94-2) are reduced to 1.76% and 1.76%, respectively.

In Case 2, the final structure is composed of two layers with 10 hidden neurons per layer in Part I and two layers with 25 hidden neurons per layer in Part II after the two-stage training. There are 1212 training parameters, including weight parameters and biases, in the developed model. The initial training data size for Case 2 of this example is 302000. After the two-stage training process, the total number of training data used for this case is 377500. The average training error and test error are 1.74% and 1.80%, respectively. Figure 8 shows the modeling results at two random geometrical samples in Case 2. The S-parameter solutions from the developed model match the desired S-parameters very well. The comparison results of the proposed technique and MLPs with different number of hidden neurons are shown in Table 4. The 3-layer MLP (i.e., MLP: 10-94-2) with similar number of training parameters as the proposed technique has a 2.88% training error and a 2.94% test error. When the total number of training parameters increases to 1900, the average training and test errors of the MLP (i.e., MLP: 10-146-2) are reduced to 2.70% and 2.76%, respectively. Figure 9 shows training errors versus the the number of training epochs for the proposed neural network model and the 3-layer MLP with 146 hidden neurons. A 4-layer MLP is used to develop the parametric model for Case 2 of this example as a further comparison. The modeling result is shown in Table 4.

TABLE 4. Comparison of model accuracy comparisons of the proposed deep neural network and conventional MLP for the fifth-order waveguide bandpass filter example.

Neural Network Topology		Details of Hidden Layers	Total Number of Weight Parameters	Training Error	Test Error
Case 1	Proposed deep neural network	Part I: 2 hidden layers with 15 neurons per layer Part II: 2 hidden layers with 8 neurons per layer	574	1.72%	1.75%
	3-layer MLP	44 hidden neurons	574	2.28%	2.30%
		94 hidden neurons	1224	1.76%	1.76%
Case 2	Proposed deep neural network	Part I: 2 hidden layers with 25 neurons per layer Part II: 2 hidden layers with 10 neurons per layer	1212	1.74%	1.80%
	3-layer MLP	94 hidden neurons	1224	2.88%	2.94%
		146 hidden layers	1900	2.70%	2.76%
	4-layer MLP	35 hidden neurons per layer	1717	1.77%	1.84%



(a)

(b)

FIGURE 8. Fifth-order waveguide bandpass filter example in Case 2: modeling results at two different geometrical samples **(a)** $x = [14.14, 12.14, 11.12, 11.47, 14.76, 15.26, 2.25, 3.11, 1.04]^T$ (mm), **(b)** $x = [14.76, 12.40, 11.13, 13.06, 14.99, 15.71, 1.13, 2.43, 2.21]^T$ (mm).

There are 35 hidden neurons per layer in the 4-layer MLP model, and the total number of training parameters is 1717. After training, the average training and test errors are 1.77% and 1.84%, respectively.

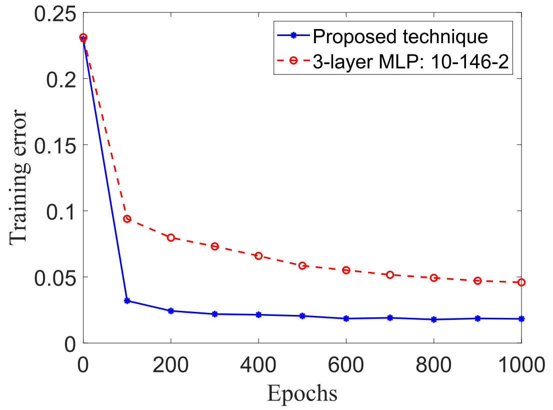


FIGURE 9. Training errors versus the number of training epochs for the proposed neural network model and the 3-layer MLP.

According to the comparisons of the modeling results shown in Table 4, for narrower variation range (Case 1), both the proposed deep neural network technique and the conventional 3-layer MLP can achieve the required model accuracy. The proposed deep neural network technique is more efficient than the 3-layer MLP because it needs less training parameters than the 3-layer MLP to achieve similar model accuracy. For the wider range (Case 2), the proposed deep neural network technique the required accuracy efficiently while the 3-layer MLP cannot reach the required accuracy even with much more training parameters than the proposed technique. Compared to the 4-layer MLP, the proposed neural network technique can achieve similar accuracy with fewer training parameters.

IV. CONCLUSION

This paper has proposed a novel deep neural network topology for parametric modeling of microwave components. The inputs in the proposed deep neural network topology have been divided into geometrical inputs and the frequency input. We have divided the hidden layers in the propose deep neural network into two parts in order to reduce the total number of training parameters. We have derived the feedforward computation from the model inputs to the outputs for the proposed deep neural network. New computations of the derivatives of the error function with respect to training parameters have been proposed. We have proposed an advanced two-stage training algorithm to train the deep neural network. The training algorithm can determine the number of hidden layers in both parts during the training process and guarantee that the proposed deep neural network model can achieve the required model accuracy. The proposed deep neural network topology can achieve similar model accuracy using less training parameters than the fully connected neural network.

REFERENCES

[1] S. A. Sadrossadat, Y. Cao, and Q.-J. Zhang, "Parametric modeling of microwave passive components using sensitivity-analysis-based adjoint neural-network technique," *IEEE Trans. Microw. Theory Techn.*, vol. 61, no. 5, pp. 1733–1747, May 2013.

[2] F. Feng, C. Zhang, J. Ma, and Q.-J. Zhang, "Parametric modeling of EM behavior of microwave components using combined neural networks and Pole-Residue-Based transfer functions," *IEEE Trans. Microw. Theory Techn.*, vol. 64, no. 1, pp. 60–77, Jan. 2016.

[3] J. L. Chavez-Hurtado and J. E. Rayas-Sanchez, "Polynomial-based surrogate modeling of RF and microwave circuits in frequency domain exploiting the multinomial theorem," *IEEE Trans. Microw. Theory Techn.*, vol. 64, no. 12, pp. 4371–4381, Dec. 2016.

[4] N. Leszczynska, I. Couckuyt, T. Dhaene, and M. Mrozowski, "Low-cost surrogate models for microwave filters," *IEEE Microw. Wireless Compon. Lett.*, vol. 26, no. 12, pp. 969–971, Dec. 2016.

[5] G. Angiulli, M. Cacciola, and M. Versaci, "Microwave devices and antennas modelling by support vector regression machines," *IEEE Trans. Magn.*, vol. 43, no. 4, pp. 1589–1592, Apr. 2007.

[6] Q.-J. Zhang, K. C. Gupta, and V. K. Devabhaktuni, "Artificial neural networks for RF and microwave design-from theory to practice," *IEEE Trans. Microw. Theory Techn.*, vol. 51, no. 4, pp. 1339–1350, Apr. 2003.

[7] J. E. Rayas-Sanchez, "EM-based optimization of microwave circuits using artificial neural networks: The state-of-the-art," *IEEE Trans. Microw. Theory Techn.*, vol. 52, no. 1, pp. 420–435, Jan. 2004.

[8] M. B. Steer, J. W. Bandler, and C. M. Snowden, "Computer-aided design of RF and microwave circuits and systems," *IEEE Trans. Microw. Theory Techn.*, vol. 50, no. 3, pp. 996–1005, Mar. 2002.

[9] D. Root, "Future device modeling trends," *IEEE Microw. Mag.*, vol. 13, no. 7, pp. 45–59, Nov. 2012.

[10] H. Kabir, L. Zhang, M. Yu, P. Aaen, J. Wood, and Q.-J. Zhang, "Smart modeling of microwave devices," *IEEE Microw. Mag.*, vol. 11, no. 3, pp. 105–118, May 2010.

[11] T. Khan and A. De, "Modeling of microstrip antennas using neural networks techniques: A review," *Int. J. RF Microw. Comput.-Aided Eng.*, vol. 25, no. 9, pp. 747–757, Nov. 2015.

[12] P. Zhao and K. Wu, "Homotopy optimization of microwave and millimeter-wave filters based on neural network model," *IEEE Trans. Microw. Theory Techn.*, vol. 68, no. 4, pp. 1390–1400, Apr. 2020.

[13] W. Na, F. Feng, C. Zhang, and Q.-J. Zhang, "A unified automated parametric modeling algorithm using knowledge-based neural network and l_1 optimization," *IEEE Trans. Microw. Theory Techn.*, vol. 65, no. 3, pp. 729–745, Mar. 2017.

[14] F. Feng, V.-M.-R. Gongal-Reddy, C. Zhang, J. Ma, and Q.-J. Zhang, "Parametric modeling of microwave components using adjoint neural networks and pole-residue transfer functions with EM sensitivity analysis," *IEEE Trans. Microw. Theory Techn.*, vol. 65, no. 6, pp. 1955–1975, Jun. 2017.

[15] J. Zhang, F. Feng, W. Zhang, J. Jin, J. Ma, and Q.-J. Zhang, "A novel training approach for parametric modeling of microwave passive components using Padé via Lanczos and EM sensitivities," *IEEE Trans. Microw. Theory Techn.*, early access, Mar. 24, 2020.

[16] W. Liu, W. Na, L. Zhu, J. Ma, and Q.-J. Zhang, "A wiener-type dynamic neural network approach to the modeling of nonlinear microwave devices," *IEEE Trans. Microw. Theory Techn.*, vol. 65, no. 6, pp. 2043–2062, Jun. 2017.

[17] A.-D. Huang, Z. Zhong, W. Wu, and Y.-X. Guo, "An artificial neural network-based electrothermal model for GaN HEMTs with dynamic trapping effects consideration," *IEEE Trans. Microw. Theory Techn.*, vol. 64, no. 8, pp. 2519–2528, Aug. 2016.

[18] S. Zhang, J. Xu, Q.-J. Zhang, and D. E. Root, "Parallel matrix neural network training on cluster systems for dynamic FET modeling from large datasets," in *IEEE MTT-S Int. Microw. Symp. Dig.*, San Francisco, CA, USA, May 2016, pp. 1–3.

[19] L. Zhu, Q. Zhang, K. Liu, Y. Ma, B. Peng, and S. Yan, "A novel dynamic neuro-space mapping approach for nonlinear microwave device modeling," *IEEE Microw. Wireless Compon. Lett.*, vol. 26, no. 2, pp. 131–133, Feb. 2016.

[20] Z. Marinkovic, G. Crupi, A. Caddemi, V. Markovic, and D. M. M.-P. Schreurs, "A review on the artificial neural network applications for small-signal modeling of microwave FETs," *Int. J. Numer. Model.*, vol. 33, no. 3, May/Jun. 2020, Art. no. e2668.

[21] A. Jarndal, "Neural network electrothermal modeling approach for microwave active devices," *Int. J. RF Microw. Comput.-Aided Eng.*, vol. 29, no. 9, pp. 1–9, Sep. 2019.

[22] M. Isaksson, D. Wisell, and D. Ronnow, "Wide-band dynamic modeling of power amplifiers using radial-basis function neural networks," *IEEE Trans. Microw. Theory Techn.*, vol. 53, no. 11, pp. 3422–3428, Nov. 2005.

[23] D. Schreurs, M. O'Droma, A. A. Goacher, and M. Gadringer, *RF Power Amplifier Behavioral Modeling*. New York, NY, USA: Cambridge Univ. Press, 2008.

[24] 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.

[25] W. Zhang, F. Feng, V.-M.-R. Gongal-Reddy, J. Zhang, S. Yan, J. Ma, and Q.-J. Zhang, "Space mapping approach to electromagnetic centric multiphysics parametric modeling of microwave components," *IEEE Trans. Microw. Theory Techn.*, vol. 66, no. 7, pp. 3169–3185, Jul. 2018.

[26] J. Monzo-Cabrera, J. L. Pedreno-Molina, A. Lozano-Guerrero, and A. Toledo-Moreo, "A novel design of a robust ten-port microwave reflectometer with autonomous calibration by using neural networks," *IEEE Trans. Microw. Theory Techn.*, vol. 56, no. 12, pp. 2972–2978, Dec. 2008.

[27] C. Zhang, J. Jin, W. Na, Q.-J. Zhang, and M. Yu, "Multivalued neural network inverse modeling and applications to microwave filters," *IEEE Trans. Microw. Theory Techn.*, vol. 66, no. 8, pp. 3781–3797, Aug. 2018.

[28] J. Jin, C. Zhang, F. Feng, W. Na, J. Ma, and Q.-J. Zhang, "Deep neural network technique for high-dimensional microwave modeling and applications to parameter extraction of microwave filters," *IEEE Trans. Microw. Theory Techn.*, vol. 67, no. 10, pp. 4140–4155, Oct. 2019.

[29] G. W. Cottrell, "New life for neural networks," *Science*, vol. 313, no. 5786, pp. 454–455, Jul. 2006.

[30] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436–444, May 2015.

[31] A. Krizhevsky, I. Sutskever, and G. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.

[32] A. Graves, A.-R. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, Vancouver, BC, Canada, May 2013, pp. 6645–6649.

[33] D. Chen and B. Mak, "Multi-task learning of deep neural networks for low-resource speech recognition," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 23, no. 7, pp. 1172–1183, Jul. 2015.

[34] C. Farabet, C. Couprie, L. Najman, and Y. LeCun, "Learning hierarchical features for scene labeling," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 8, pp. 1915–1929, Aug. 2013.

[35] X. Glorot, A. Bordes, and Y. Bengio, "Domain adaptation for large-scale sentiment classification: A deep learning approach," in *Proc. Int. Conf. Mach. Learn.*, Washington, DC, USA, Jul. 2011, pp. 513–520.

[36] R. Collobert and J. Weston, "A unified architecture for natural language processing: Deep neural networks with multitask learning," in *Proc. 25th Int. Conf. Mach. Learn. (ICML)*, Helsinki, Finland, Jul. 2008, pp. 160–167.

[37] K. Cho, B. van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN Encoder–Decoder for statistical machine translation," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, Doha, Qatar, Oct. 2014, pp. 1724–1734.

[38] Q.-J. Zhang and K. C. Gupta, *Neural Networks for RF and Microwave Design*. Norwood, MA, USA: Artech House, 2000.

[39] W. Na, W. Liu, L. Zhu, F. Feng, J. Ma, and Q.-J. Zhang, "Advanced extrapolation technique for neural-based microwave modeling and design," *IEEE Trans. Microw. Theory Techn.*, vol. 66, no. 10, pp. 4397–4418, Oct. 2018.

[40] S. Liao, H. Kabir, Y. Cao, J. Xu, Q.-J. Zhang, and J.-G. Ma, "Neural-network modeling for 3-D substructures based on spatial EM-field coupling in finite-element method," *IEEE Trans. Microw. Theory Techn.*, vol. 59, no. 1, pp. 21–38, Jan. 2011.

JING JIN (Student Member, IEEE) was born in Suizhou, Hubei, China. She received the B.Eng. degree from Wuhan University, Wuhan, China, in 2014. She is currently pursuing the Ph.D. degree with the School of Microelectronics, Tianjin University, Tianjin, China, and the Cotutelle Ph.D. degree with the Department of Electronics, Carleton University, Ottawa, ON, Canada. Her current research interests include modeling of microwave circuits, deep neural network modeling technique, and computer aided optimization, and the tuning of microwave filters.

FENG FENG (Member, IEEE) was born in Huludao, Liaoning, China. He received the B.Eng. degree from Tianjin University, Tianjin, China, in 2012, and the Ph.D. degree from the School of Microelectronics, Tianjin University, and the Department of Electronics, Carleton University, Ottawa, ON, Canada, in 2017.

He is currently a Postdoctoral Fellow with the Department of Electronics, Carleton University. His research interests include microwave circuit design and modeling, optimization theory and algorithms, space mapping and surrogate model optimization, and EM field simulation and optimization.

JIANAN ZHANG (Student Member, IEEE) was born in Tieling, Liaoning, China, in 1991. He received the B.Eng. degree from Tianjin University, Tianjin, China, in 2013, and the Ph.D. degree from the School of Microelectronics, Tianjin University, and the Department of Electronics, Carleton University, Ottawa, ON, Canada, in 2020.

He is currently a Research Associate with the Department of Electronics, Carleton University. His research interests include neural-network-based parametric modeling, polynomial chaos expansion, space mapping-based EM optimization, computational electromagnetics, and model-order reduction.

SHUXIA YAN received the B.Eng. degree in communication engineering from Tianjin Polytechnic University, Tianjin, China, in 2010, and the M.E. and Ph.D. degrees in electromagnetic field and microwave technology from Tianjin University, in 2012 and 2015, respectively.

Since 2015, she has been with the School of Electronics and Information Engineering, Tianjin Polytechnic University. Her current research interests include neural-network-based methods for microwave device modeling and circuit design, and the development of a neural-network-based circuit simulator.

WEICONG NA (Member, IEEE) received the B.Eng. degree from Tianjin University, Tianjin, China, in 2012, the Ph.D. degree from the School of Microelectronics, Tianjin University, and the Department of Electronics, Carleton University, Ottawa, ON, Canada, in 2018.

She is currently a Lecturer with the Faculty of Information Technology, Beijing University of Technology, Beijing, China. Her research interests include microwave circuit modeling and design, automated neural based model generation, EM field knowledge-based modeling and optimization, and deep neural network modeling for microwave applications.

QIJUN ZHANG (Fellow, IEEE) received the B.Eng. degree from the Nanjing University of Science and Technology, Nanjing, China, in 1982, and the Ph.D. degree in electrical engineering from Mc-Master University, Hamilton, ON, Canada, in 1987.

From 1982 to 1983, he was with the System Engineering Institute, Tianjin University, Tianjin, China. From 1988 to 1990, he was with Optimization Systems Associates (OSA) Inc., Dundas, ON, where he developed advanced microwave optimization software. In 1990, he joined the Department of Electronics, Carleton University, Ottawa, ON, where he is currently a Chancellor's Professor. He has authored or coauthored over 300 publications. He has authored *Neural Networks for RF and Microwave Design* (Artech House, 2000), coedited *Modeling and Simulation of High-Speed VLSI Interconnects* (Kluwer, 1994), and contributed to the Encyclopedia of *RF and Microwave Engineering* (Wiley, 2005), *Fundamentals of Nonlinear Behavioral Modeling for RF and Microwave Design* (Artech House, 2005), and *Analog Methods for Computer-Aided Analysis and Diagnosis* (Marcel Dekker, 1988). He was a Guest Co-Editor for the Special Issue on High-Speed VLSI Interconnects for the *International Journal of Analog Integrated Circuits and Signal Processing* (Kluwer, 1994), and twice was a Guest Editor for the Special Issue on Applications of ANN to RF and Microwave Design for the *International Journal of RF and Microwave Computer-Aided Engineering* (Wiley, 1999 and 2002). His research interests are microwave design automation especially on neural-network and optimization methods for high-speed/high-frequency circuit design.

Dr. Zhang is a Fellow of the Electromagnetics Academy and a Fellow of Canadian Academy of Engineering. He is an Associate Editor of the IEEE Transactions on Microwave Theory and Techniques. He is the Chair of the Technical Committee on Design Automation (MTT-2) of the IEEE Microwave Theory and Techniques Society (IEEE MTT-S).