Exact Adjoint Sensitivity Analysis for Neural-Based Microwave Modeling and Design

Jianjun Xu, *Student Member, IEEE*, Mustapha C. E. Yagoub, *Member, IEEE*, Runtao Ding, and Qi Jun Zhang, *Senior Member, IEEE*

***Abstract*—For the first time, an adjoint neural network method is introduced for sensitivity analysis in neural-based microwave modeling and design. The proposed method is applicable to generic microwave neural models including variety of knowledge-based neural model embedding microwave empirical information. Through the proposed technique, efficient first- and second-order sensitivity analysis can be carried out within the microwave neural network infrastructure using neuron responses in both the original and adjoint neural models. A new formulation of simultaneous training of original and adjoint neural models allows robust model development by learning not only the input/output behavior of the modeling problem, but also its derivative data. The proposed technique is very useful for neural-based microwave optimization and synthesis, and for analytically unified dc/small-signal/large-signal device modeling and circuit design. Examples of high-speed very large scale integration system interconnect modeling and optimization, large-signal FET modeling, and three-stage power-amplifier simulation utilizing the proposed sensitivity technique are demonstrated.**

***Index Terms*—Design automation, modeling, neural networks, sensitivity.**

I. INTRODUCTION

ARTIFICIAL neural networks have been recently recognized as a useful vehicle for RF and microwave modeling and design [1]. Neural network models can be trained from electromagnetic (EM)/physics simulation or measurement data and subsequently used during circuit analysis and design. The models are fast and can represent EM/physics behaviors it learned, which otherwise are computationally expensive. Various types of input–output information in linear and nonlinear microwave design have been used for neural network learning, such as EM solutions versus geometrical/physical parameters [2]–[4], signal integrity solutions versus electrical parameters [5], transistor electrical parameter versus electrical parameters [6], transistor electrical versus physical parameters [7], and more. The learning ability of neural networks is very useful when an analytical model for a new device is not available, e.g., modeling of a new transistor. A neural network can also generalize, meaning that the model can respond to new data that has not been used during training. Neural models can be more accurate than polynomial regression models, handle more dimensions than lookup table models, and allow more automation in model development than conventional circuit models. Microwave researchers have demonstrated this approach in a variety of applications such as modeling and optimization of high-speed very large scale integration (VLSI) interconnects [2], coplanar waveguide (CPW) circuits [8], spiral inductors [9], microwave FETs and amplifiers [10], [11], CMOS and HBTs [12], [13], global modeling [14], and yield optimization and circuit synthesis [10], [15], [16]. Knowledge-based approaches combining microwave empirical or equivalent-circuit models together with neural network learning have also been studied [7], [17], [18] to further improve the training efficiency and model reliability.

This paper addresses a new task in this area, i.e., neural-based sensitivity analysis. Sensitivity information is very important for circuit optimization [19], [20], and for unified dc/small-signal/large-signal modeling and circuit design [21]. In the case of neural networks, first-order sensitivity analysis has been studied, e.g., for networks with binary responses for signal-processing purposes [22] and for multilayer perceptron structures used in microwave modeling and design [16], [23]. However, to perform sensitivity analysis in more generic neural model structures including embedded microwave knowledge, and to train the networks to learn from sensitivity data that arise during microwave modeling, remain an unsolved task.

For the first time, a novel adjoint neural network sensitivity analysis technique is presented in this paper, which allows exact sensitivity to be calculated in a general neural model accommodating microwave empirical functions, equivalent circuit, as well as conventional switch-type neurons in an arbitrary neural network structure. The adjoint neural network structure is excited by a unit excitation corresponding to the output neurons in the original neural network. A new formulation allows the training of the adjoint neural models to learn from derivative training data. An elegant derivation is presented where the first- and second-order derivative calculation are carried out using the neural network infrastructure through a combination of back-propagation processes in both the original and adjoint neural networks. Using the second-order derivative, we are able to train a neural network model to learn not only microwave input/output data, but also its derivative information, which is very useful in simultaneous dc/small-signal/large-signal device modeling.

In Section II, the microwave neural modeling problem is summarized. In Section III, we formulate the new adjoint sensitivity technique and present its structure including descriptions of adjoint neurons and links, and element derivative neurons (EDNs). We then formulate a combined training of original and

Manuscript received September 19, 2001.

J. Xu and Q. J. Zhang are with the Department of Electronics, Carleton University, Ottawa, ON, Canada K1S 5B6.

M. C. E. Yagoub is with the School of Information Technology and Engineering, University of Ottawa, Ottawa, ON, Canada K1N 6N5.

R. Ding is with the School of Electronics and Information Engineering, Tianjin University, Tianjin 300072, China.

Digital Object Identifier 10.1109/TMTT.2002.806910

adjoint neural networks and derive the sensitivity formulas for both first- and second-order derivatives using the neural network error propagation infrastructure. In Section IV, the proposed sensitivity analysis technique is applied to high-speed VLSI interconnect modeling and optimization, large-signal FET modeling, and three-stage power-amplifier simulation examples.

II. MICROWAVE NEURAL MODELING: PROBLEM STATEMENT

Let $\boldsymbol{x}$ represent an N_x-vector containing parameters of a microwave device/circuit, e.g., gate length and gatewidth of an FET, or width and spacing of transmission lines. Let $\boldsymbol{y}$ represent a N_y-vector containing the responses of the device/circuit under consideration, e.g., drain current of an FET, or mutual inductance between transmission lines. The physics/EM relationship between $\boldsymbol{y}$ and $\boldsymbol{x}$ can be highly nonlinear and multidimensional. The theoretical model for this relationship may not be available (e.g., a new semiconductor device), theory may be too complicated to implement, or the theoretical model may be computationally too intensive for online microwave design and repetitive optimization (e.g., three-dimensional full-wave EM analysis inside a Monte Carlo statistical design loop). We aim to develop a fast and accurate neural model by teaching/training a neural network to learn the microwave problem. Let the neural network model be defined as

$$\boldsymbol{y} = \boldsymbol{y}(\boldsymbol{x}, \boldsymbol{w}) \tag{1}$$

where $\boldsymbol{w}$ represents the parameters inside the neural network, also called the weight vector. A neural network realization of the $\boldsymbol{x}$–$\boldsymbol{y}$ relationship is through a network of interconnected neurons. The most widely used neural network structure is the feed-forward multilayer perceptrons [1], [22], where neurons are grouped into layers, and each neuron in a layer acts as a smooth switch that produces a response between the low and high state according to the weighted responses of all neurons from the preceding layer. The neural network structure allows the ability to represent multidimensional nonlinear input/output mappings accurately, and to evaluate $\boldsymbol{y}$ from $\boldsymbol{x}$ quickly. To enable a neural network to represent a specific microwave $\boldsymbol{x}$–$\boldsymbol{y}$ relationship, we first train the neural network to learn the microwave data pairs $(\boldsymbol{x_S}, \boldsymbol{d_S})$, where $\boldsymbol{x}_s$ is a sample of $\boldsymbol{x}$, $\boldsymbol{d}_s$ is a vector representing the $\boldsymbol{y}$ data generated from microwave simulation or measurement under given sample $\boldsymbol{x}_s$, and s is the sample index. For training purpose, we define an error function $E(\boldsymbol{w})$ as

$$E(\boldsymbol{w}) = \frac{1}{2} \sum_{s \in Tr} \sum_{k=1}^{N_y} \left(y_k(\boldsymbol{x}_s, \boldsymbol{w}) - d_{ks} \right)^2 \tag{2}$$

where d_{ks} is the kth element of $\boldsymbol{d}_s$, $y_k(\boldsymbol{x}_s, \boldsymbol{w})$ is the kth output of the neural network for input sample $\boldsymbol{x}_s$ and Tr is an index set of all training samples. The objective of neural network training is to adjust neural network connection weights $\boldsymbol{w}$ such that $E(\boldsymbol{w})$ is minimized. A trained neural model can then be used online during microwave design stage providing fast model evaluation replacing original slow EM-device simulators. The benefit of the neural model is especially significant when the model is highly repetitively used in design processed such as optimization, Monte Carlo analysis, and yield maximization [24].

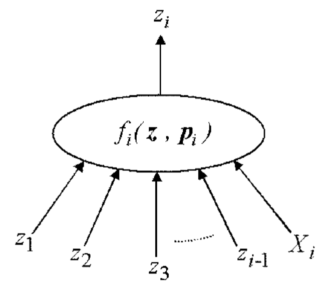


Fig. 1 Typical neuron, say, the ith neuron, in the original neural network. The neuron receives stimulus from responses of other neurons z_j, $j < i$, processes the stimulus using a processing function $f_i(\boldsymbol{z}, \boldsymbol{p}_i)$, and produces a response z_i. $\boldsymbol{p}_i$ is a vector of parameters for the processing function. X_i is an external stimulus.

Our present task is to develop a technique for sensitivity analysis of $\boldsymbol{y}$ with respect to $\boldsymbol{x}$ of a microwave neural model for neural-based circuit optimization, and to develop a training technique to allow the neural model to simultaneously learn the $\boldsymbol{x}$–$\boldsymbol{y}$ relationship and derivative information of $\boldsymbol{y}$ with respect to $\boldsymbol{x}$ in microwave design. The second task involves the derivation of second-order sensitivity information. In addition, we require that the technique be applicable not only for multilayer perceptron structure, but also for general structures and knowledge-based neural networks including microwave empirical functions in neurons.

III. PROPOSED ADJOINT NEURAL NETWORK APPROACH

A. Formulation of Two Neural Models: Original and Adjoint Neural Model

Two models, one called the original neural network model and the other defined as the adjoint neural network model, are utilized in the proposed sensitivity analysis technique. Each model consists of neurons and connections between neurons. Each neuron receives and processes stimuli (inputs) from other neurons and/or external inputs, and produces a response (output). Here, we introduce a generic framework in which microwave empirical and equivalent models can be coherently represented in the neural network structures, and connections between neurons can be arbitrary allowing different types of microwave neural structures to be included.

Suppose for a generic neuron, say, neuron i in the original model, the response is z_i and the external input to this neuron is X_i. Let N be the total number of neurons in the original neural network and $\boldsymbol{z} = [z_1, z_2, \ldots, z_N]^{\mathrm{T}}$. In order to accommodate microwave empirical knowledge, we use a notation $f_i(\boldsymbol{z}, \boldsymbol{p}_i)$ to represent the processing function for neuron i where $\boldsymbol{p}_i$ could represent either the neuron connection weights or parameters in a microwave empirical/equivalent model, as shown in Fig. 1. The collection of $\boldsymbol{p}_1, \boldsymbol{p}_2, \ldots, \boldsymbol{p}_N$ forms the weight vector $\boldsymbol{w}$ for the overall neural model. For example, if neuron i is a sigmoid switch neuron, then $f_i(\boldsymbol{z}, \boldsymbol{p}_i) = 1/(1 + e^{-p_i^T \cdot z})$, where $\boldsymbol{p}_i$ is an N-vector and its elements represent connection weights between neuron i and other neurons. For another example, $f_i(\boldsymbol{z}, \boldsymbol{p}_i)$ could represent an empirical formula for an FET drain current versus terminal voltages and physical/geometrical parameters [7]. If df_i/dz_j is nonzero (or zero), then neuron i is (or is not) connected from neuron j. In such a way, this formulation allows us to represent not only multilayer perceptrons, but also arbitrary connections between neurons and knowledge-based neural networks.

A neuron who receives stimulus from outside the neural network is called an input neuron. A neuron whose response becomes the output of the overall neural network model is called an output neuron. A neuron whose stimulus is from responses of other neurons and whose response becomes stimulus to other neurons is called a hidden neuron. Let I and K be defined as index sets containing indexes of input neurons and output neurons, respectively, i.e.,

$$I = \{\, i \,|\text{if stimulus to neuron}\, i \,\text{is from neural model external inputs, i.e.,}\, \boldsymbol{x} \,\}$$
$$K = \{\, k \,|\, \text{if response of neuron}\, k \,\text{is an output of the overall neural model, i.e.,}\, \boldsymbol{y} \,\}$$

Assuming the neuron indexes are numbered consecutively starting from the input neurons, through hidden neurons, and to the outputs neurons. The feed-forward calculation of the original model can be defined as

$$z_i = f_i(\boldsymbol{z}, \boldsymbol{p}_i) + X_i, \quad X_i = \begin{cases} x_i & i \in I \\ 0, & i \notin I \end{cases} \tag{3}$$

calculated sequentially for $i = 1, 2, \ldots, N$. The outputs of the original neural model $\boldsymbol{y}$ will be the neuron responses at the output neurons, i.e., $y_i = z_k$, $k = i + N - N_y$, $k \in K$.

Now we introduce the adjoint neural model, which consists of N adjoint neurons. Let $\hat{z}_j^k$ be the response of the jth adjoint neuron. We interpret $\hat{z}_j^k$ as the gradient of the original neural model output with respect to the local response of the jth neuron in the original neural model, i.e.,

$$\hat{z}_j^k = \frac{\partial z_k}{\partial z_j} \tag{4}$$

where k and $k \in K$ indicates an output neuron of interest for which sensitivity is to be computed. In most of the follow presentation, we use $\hat{z}_j$ to represent $\hat{z}_j^k$ for simplicity.

The processing function for this adjoint neuron is defined as a linear function

$$\hat{z}_j = \sum_{i=j+1}^{N} \frac{\partial f_i}{\partial z_j} \cdot \hat{z}_i + \delta_{kj} \quad \delta_{kj} = \begin{cases} 1, & k = j \\ 0, & k \neq j \end{cases} \tag{5}$$

where $(\partial f_i)/(\partial z_j)$, which could be derivatives from microwave empirical functions, are the local derivatives of original neuron functions.

Let $\boldsymbol{J}$ be the Jacobian matrix $((\partial \boldsymbol{f}^T)/(\partial \boldsymbol{z}))^T$, where $\boldsymbol{f} = [f_1, f_2, \ldots, f_N]$. For generic feed-forward neural networks with neuron indexes numbered consecutively starting from the input neurons, through hidden neurons to the outputs neurons, we have $(\partial f_i)/(\partial z_j) = 0$ if $j \geq i$.

Equation (5) is equivalent to

$$(\mathbf{1} - \boldsymbol{J})^T \cdot \hat{\boldsymbol{z}} = [\delta_{k1}\ \delta_{k2} \cdots \delta_{kN}]^T \tag{6}$$

where $\mathbf{1}$ is an $N \times N$ identity matrix. Since $(\mathbf{1} - \boldsymbol{J})^T$ is upper diagonal, to perform "feed-forward" computation in the adjoint model, we first initialize the last several adjoint neurons (corresponding to the output neurons in the original neural model) by Kronecker functions $\hat{z}_j = \delta_{kj}$, $j \in K$. We then calculate (5) *backward* according to the neuron sequence $j = N-1, N-2, \ldots, 1$ without solving equations. The final desired sensitivity solution of the original $\boldsymbol{x}$–$\boldsymbol{y}$ model can now be obtained explicitly from the adjoint model as $(\partial y_i)/(\partial x_j) = (\partial z_k)/(\partial z_j) = \hat{z}_j$, $k \in K, j \in I, i = k + N_y - N$. Notice that the adjoint neurons receiving nonzero external excitation (i.e., δ_{kj}) correspond to the output neurons in the original neural model, i.e., $j = k \in K$.

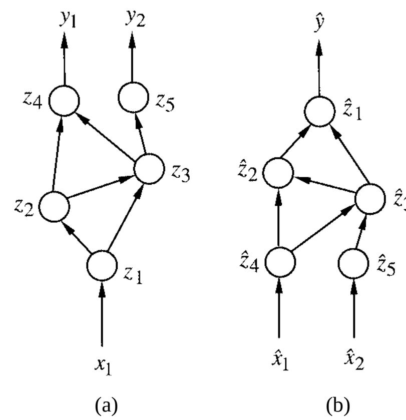


Fig. 2 Example illustrating: (a) original neural model and (b) basic adjoint neural model for sensitivity analysis. The input (output) neurons in the adjoint model correspond to the output (input) neurons in the original model. The neuron processing sequence in the adjoint model is the reverse of that in the original model.

B. Basic Adjoint Neural Model Structure

As formulated in (5), the input (output) neurons in the adjoint model correspond to the output (input) neurons in the original model, and the sequence of the neuron processing in the adjoint model is exactly the reverse of that in the original neural model. With this concept, a basic adjoint neural network structure can be created by flipping the original neural model between input and output. The connections between the adjoint neurons i and j has a weight value equal to $(\partial f_j)/(\partial z_i)$ (to be referred to as local derivative), and processing functions for all adjoint neurons are linear.

Here, we use an example to show how to setup a basic adjoint neural model from a given original neural model. The original model is given in Fig. 2(a) The total number of neurons in the original model is $N = 5$.

Knowing that the adjoint neural model is the"reverse" version of original neural model, we realize the adjoint model structure shown in Fig. 2(b), where, from (5), we have

$$\begin{aligned} \hat{z}_3 &= \hat{z}_4 \cdot \frac{\partial f_4}{\partial z_3} + \hat{z}_5 \cdot \frac{\partial f_5}{\partial z_3} \\ \hat{z}_2 &= \hat{z}_3 \cdot \frac{\partial f_3}{\partial z_2} + \hat{z}_4 \cdot \frac{\partial f_4}{\partial z_2} \\ \hat{z}_1 &= \hat{z}_2 \cdot \frac{\partial f_2}{\partial z_1} + \hat{z}_3 \cdot \frac{\partial f_3}{\partial z_1}. \end{aligned} \tag{7}$$

By providing values of $(\partial f_4)/(\partial z_3)$, $(\partial f_5)/(\partial z_3)$, $(\partial f_4)/(\partial z_2)$, $(\partial f_3)/(\partial z_2)$, $(\partial f_3)/(\partial z_1)$, and $(\partial f_2)/(\partial z_1)$ as the connection weights in Fig. 2(b), we obtain the basic adjoint neural network.

This basic adjoint neural model can be used for first-order sensitivity analysis and for optimization such as physical/geometry optimization of EM problems. When the neural model

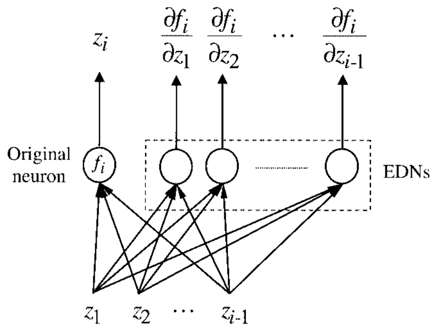


Fig. 3 Relationship between the ith original neuron and the fictitious EDNs.

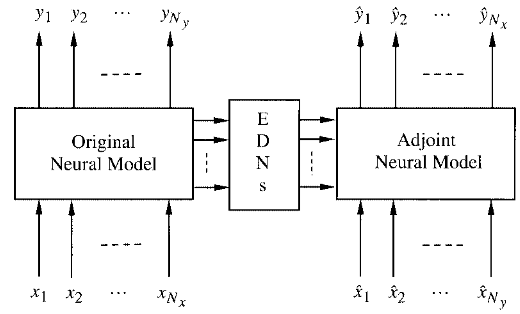


Fig. 4 Original neural model, adjoint neural model, and EDNs. The adjoint model in this setup is trainable.

structure is multilayer perceptron, our technique becomes equivalent to the existing sensitivities in [16] and [23]. Our method described above expands sensitivity analysis to general microwave neural models such as knowledge-based neural models embedding microwave empirical information.

C. Trainable Adjoint Neural Model Structure

Here, we consider a novel and advanced neural modeling requirement, i.e., to use sensitivity as target data for learning. This can be useful for enhancing reliability of models and for addressing challenges in microwave modeling involving different domains, e.g., large-signal versus small-signal domains because small-signal parameters embed the derivative information of the large-signal model. Here, we propose to train adjoint neural models to achieve this task.

If the adjoint neural model is to be trained, the connection weights in the adjoint neural model will vary with respect to (dependent upon) training parameters in both the adjoint and original models. In order to derive a training technique using the neural model framework, we add a set of fictitious neurons, called EDNs, whose processing functions are exactly the local derivatives $(\partial f_j)/(\partial z_i)$. These EDNs are stimulated by (dependent upon) neurons in the original neural model, and the responses of the EDNs become the stimulus to the adjoint neural model. In general, the EDNs can be created from each neuron in the original neural model shown in Fig. 3. The EDNs share the same stimuli and parameters as their corresponding original neurons.

The overall sensitivity analysis framework is shown in Fig. 4 including the original model, adjoint model, and EDNs, where $\{x_1x_2,\ldots,x_{N_x}\}$, $\{y_1y_2,\ldots,y_{N_y}\}$ and $\{\hat{x}_1\hat{x}_2,\ldots,\hat{x}_{N_y}\}$, $\{\hat{y}_1\hat{y}_2,\ldots,\hat{y}_{N_x}\}$ are the inputs and outputs of the original and adjoint neural models, respectively.

For the adjoint model, the relationships between inputs and outputs are decided in Table I.

Here, we use the example from Fig. 2 to show the setup of a trainable adjoint neural model from the given original neural model. The EDNs are created from the original model shown in Fig. 5(a) and are connected to the adjoint neural model illustrated in Fig.5 (b), where the EDNs are defined as

$$z_{3''} = \frac{\partial f_3}{\partial z_2}$$
$$z_{3'} = \frac{\partial f_3}{\partial z_1}$$

TABLE I

RELATIONSHIP BETWEEN INPUTS AND OUTPUTS OF ADJOINT NEURAL-MODEL. INPUTS TO THE ADJOINT MODEL ARE UNIT EXCITATIONS APPLIED TO AN ADJOINT INPUT NEURON, WHICH CORRESPONDS TO AN OUTPUT NEURON IN THE ORIGINAL NEURAL MODEL

Input $\hat{x}$	Output $\hat{y}$
$[1\ 0\ 0\ \cdots\ 0]$	$\left[\frac{\partial y_1}{\partial x_1}\ \frac{\partial y_1}{\partial x_2}\ \cdots\ \frac{\partial y_1}{\partial x_{N_x}}\right]$
$[0\ 1\ 0\ \cdots\ 0]$	$\left[\frac{\partial y_2}{\partial x_1}\ \frac{\partial y_2}{\partial x_2}\ \cdots\ \frac{\partial y_2}{\partial x_{N_x}}\right]$
$[0\ \cdots\ 1\ \cdots\ 0]$ $1\ 2\cdots k\ \cdots\ N_y$	$\left[\frac{\partial y_k}{\partial x_1}\ \frac{\partial y_k}{\partial x_2}\ \cdots\ \frac{\partial y_k}{\partial x_{N_x}}\right]$
$[0\ 0\ 0\ \cdots\ 1]$	$\left[\frac{\partial y_{N_y}}{\partial x_1}\ \frac{\partial y_{N_y}}{\partial x_2}\ \cdots\ \frac{\partial y_{N_y}}{\partial x_{N_x}}\right]$

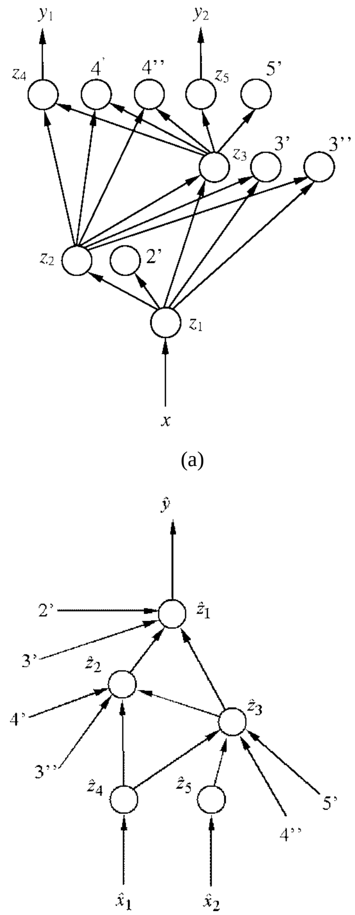
$$z_{2'} = \frac{\partial f_2}{\partial z_1}$$
$$z_{4'} = \frac{\partial f_4}{\partial z_2}$$
$$z_{4''} = \frac{\partial f_4}{\partial z_3}$$
$$z_{5'} = \frac{\partial f_5}{\partial z_3}.$$

D. Combined Training of the Adjoint and the Original Neural Models

Let g_j^k represent the derivative training data (i.e., desired target value) for the derivative $(dz_k)/(dx_j)\ k \in K$. Let $\boldsymbol{g}$ represent the derivative training data including g_j^k for all $k \in K, j \in I$. We formulate a new training task such that the neural model $\boldsymbol{y(x,w)}$ fits not only the $\boldsymbol{x}$–$\boldsymbol{y}$ relationship, but simultaneously also the required derivative relationship of $\boldsymbol{y}$ with respect to $\boldsymbol{x}$.

To achieve this goal, we utilize the adjoint neural network model such that the training task becomes simultaneous training of the original and the adjoint neural models. Let the per sample training error be defined as

$$\tilde{E} = \tilde{E}_o + \tilde{E}_a = \frac{1}{2}\left[W_1 \sum_{k\in K}(z_k - d_k)^2 + W_2 \sum_{i\in I, k\in K}\left(\frac{dz_k}{dx_i} - g_i^k\right)^2\right] \quad (8)$$



(a)

(b)

Fig. 5 Illustration of EDNs and trainable adjoint model for the example in Fig. 4. (a) Original neural model with EDNs created. (b) Trainable adjoint neural model. The single or double prime denote EDNs, e.g., 4′ and 4′′ represent EDNs created from neuron 4 of (a) and used in the adjoint neural model of (b).

where $\tilde{E}_o$ and $\tilde{E}_a$ represent the training error from original and adjoint neural models, respectively, subscripts i and k (used for x, z, d, and g) indicate original input neuron i and original output neuron k, respectively, and W_1 and W_2 are weights used to balance emphasis between training original and adjoint models. We also call $\tilde{E}_o$ and $\tilde{E}_a$ the original training error and adjoint training error, respectively.

The overall training error will be this per sample error $\tilde{E}$ accumulated over various samples in training data set Tr. During training, both the original and adjoint neural models share a common set of parameters $\boldsymbol{p}_i$, $i = 1, 2, \ldots, N$ to ensure consistency between original and adjoint models, and/or to ensure that training original and adjoint models reinforce each other's accuracy.

Our formulation can accommodate the following three types of training situations.

1) Train the original neural model using input/output data $(\boldsymbol{x}, \boldsymbol{d})$. After training, the outputs of the adjoint model automatically become an explicit derivative of the original input/output relationship.
2) Train the adjoint model to learn derivative data $(\boldsymbol{x}, \boldsymbol{g})$. The original model will then give original input/output (i.e., $\boldsymbol{x}$–$\boldsymbol{y}$) relationship, which has the effect of providing an integration solution over the derivative training data.
3) Train both the original and adjoint models together to learn $(\boldsymbol{x}, \boldsymbol{d})$ and $(\boldsymbol{x}, \boldsymbol{g})$ data, which will help the neural model to be trained more accurately and robustly.

Fig. 6 shows those types of training. We can achieve these three training cases using our general formulation of (8) by setting: 1) $W_2 = 0$; 2) $W_1 = 0$; or 3) $W_1 \neq 0$ and $W_2 \neq 0$, respectively.

E. Second-Order Sensitivity Analysis

Training is to adjust neural network internal parameters $\boldsymbol{p}_i$ for each neuron such that the accumulated training error of $\tilde{E}$ is minimized. The training algorithms, such as the conjugate gradient method, quasi-Newton method, and back-propagation [1] typically require the derivative of $\tilde{E}$ with respect to $\boldsymbol{p}_i$.

First, considering the training errors due to training of the adjoint neural model to learn input/output derivative data, it becomes necessary to perform second-order sensitivity analysis.

Let

$$\tilde{E}_a = \sum_{k \in K} \tilde{E}_{a,k} \tag{9}$$

be defined as the derivative training error for each sample data, where $\tilde{E}_{a,k}$ is the training error between the adjoint model and the sensitivity data for the kth output neuron in the original model. Let ψ_i represent an element in vector $\boldsymbol{p}_i$, which are the parameters of the ith neuron in the original model. To find the derivatives required to train the adjoint model for each sample, we first differentiate $\tilde{E}_{a,k}$ as

$$\frac{\partial \tilde{E}_{a,k}}{\partial \psi_i} = \sum_{i \in I} W_2 \cdot (\hat{z}_i - g_{ki}) \cdot \frac{\partial \hat{z}_i}{\partial \psi_i} = \boldsymbol{G}^T \cdot \frac{\partial \hat{\boldsymbol{z}}}{\partial \psi_i} \tag{10}$$

where $\boldsymbol{G}$ is a column vector of size N with elements

$$G_i = \begin{cases} W_2 \cdot (\tilde{z}_i - g_{ki}), & i \in I \\ 0, & i \notin I \end{cases}$$

which is the training error at the output neuron in the adjoint model, i.e., adjoint neuron i. To obtain $(\partial \hat{\boldsymbol{z}})/(\partial \psi_i)$, we can differentiate (6) with respect to parameter ψ_i as

$$(\boldsymbol{1} - \boldsymbol{J})^T \cdot \frac{\partial \hat{\boldsymbol{z}}}{\partial \psi_i} - \left(\frac{\partial \boldsymbol{J}}{\partial \psi_i} + \sum_{n=i+1}^{N} \frac{\partial \boldsymbol{J}}{\partial z_n} \cdot \frac{\partial z_n}{\partial \psi_i} \right)^T \cdot \hat{\boldsymbol{z}} = 0 \tag{11}$$

Equation (10) can now be replaced by

$$\frac{\partial \tilde{E}_{a,k}}{\partial \psi_i} = \boldsymbol{G}^T \cdot \left[(\boldsymbol{1} - \boldsymbol{J})^{-1} \right]^T \cdot \left(\frac{\partial \boldsymbol{J}}{\partial \psi_i} + \sum_{n=i+1}^{N} \frac{\partial \boldsymbol{J}}{\partial z_n} \cdot \frac{\partial f_n}{\partial \psi_i} \right)^T \cdot \hat{\boldsymbol{z}}. \tag{12}$$

Let $\hat{\hat{z}}$ be defined as a vector solution for

$$(\boldsymbol{1} - \boldsymbol{J}) \cdot \hat{\hat{z}} = \boldsymbol{G}. \tag{13}$$

$\hat{\hat{z}}$ can be interpreted as error propagation signal in the adjoint neural model, which is solved from back-propagation in the adjoint model according to the neuron processing sequence $j = 2, 3, \ldots, N$ by initializing $\hat{\hat{z}}_1 = W_2 \cdot (\hat{z}_1 - g_{k1})$, and

$$\hat{\hat{z}}_j = \sum_{m=l}^{j-l} \frac{\partial f_j}{\partial z_m} \cdot \hat{\hat{z}}_m + G_j. \tag{14}$$

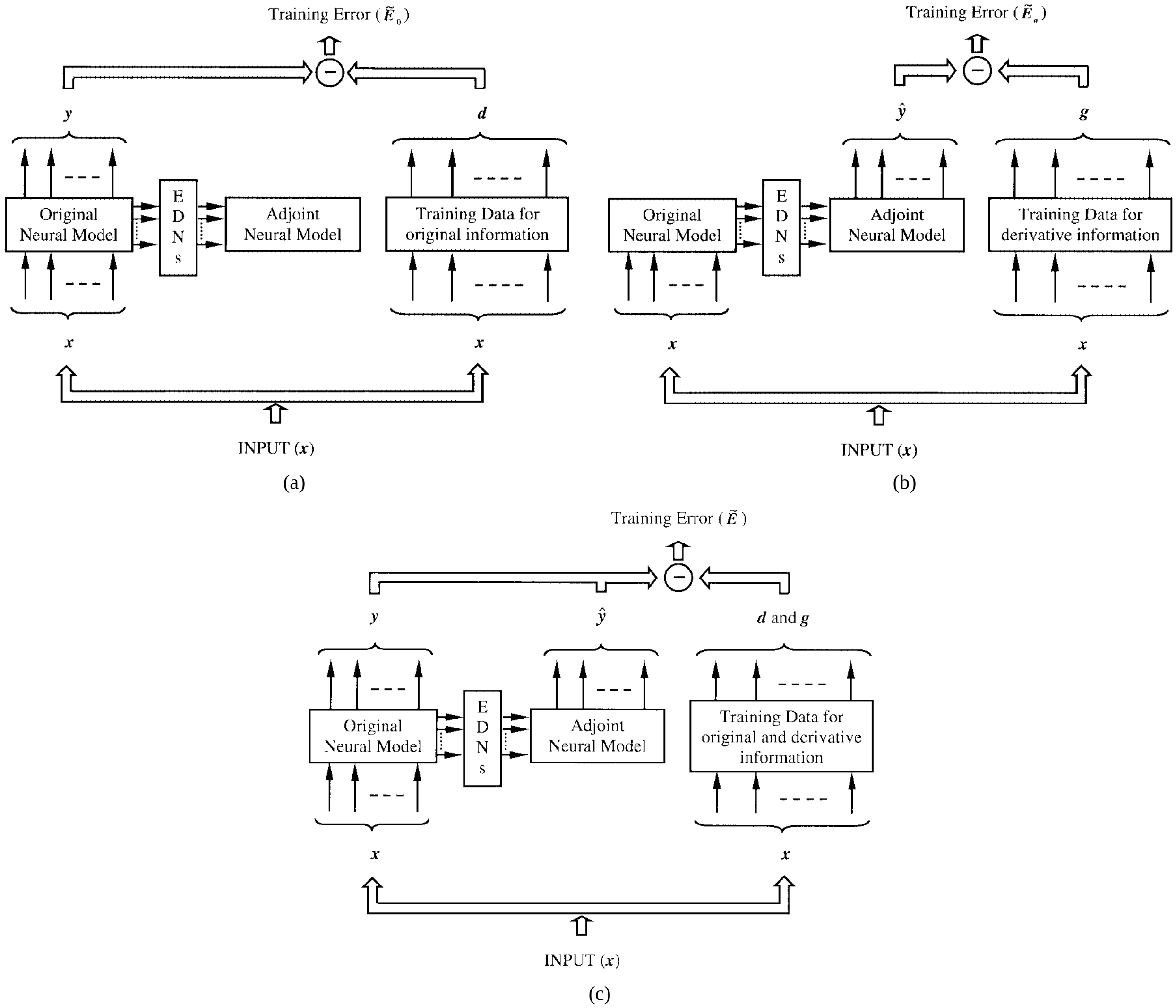


Fig. 6 (a) Training to learn original $(\boldsymbol{x}, \boldsymbol{y})$ input–output relationship, i.e., to learn from data $(\boldsymbol{x}, \boldsymbol{d})$. After training, the adjoint model automatically provides explicit sensitivity information of $\boldsymbol{y}$ versus $\boldsymbol{x}$. (b) Training to learn derivation information of $\boldsymbol{y}$ with respect to $\boldsymbol{x}$, i.e., to learn from data $(\boldsymbol{x}, \boldsymbol{g})$. After training, the original model provides an $(\boldsymbol{x}, \boldsymbol{y})$ relationship with an integration effect on training data $\boldsymbol{g}$. (c) Training to simultaneously learn both input–output and its derivative information, enhancing reliability of the neural model.

Equation (12) now becomes

$$\begin{aligned}\frac{\partial \tilde{E}_{a,k}}{\partial \psi_i} &= \hat{\hat{z}}^{\tau} \cdot \left(\frac{\partial \boldsymbol{J}}{\partial \psi_i} + \sum_{n=i+1}^{N} \frac{\partial \boldsymbol{J}}{\partial z_n} \cdot \frac{\partial z_n}{\partial \psi_i} \right)^{\tau} \cdot \hat{z} \\ &= \sum_{j=1}^{i-1} \hat{\hat{z}}_j \cdot \frac{\partial^2 f_i}{\partial z_j \partial \psi_i} \cdot \hat{z}_i \\ &\quad + \sum_{n=i+1}^{N} \left(\sum_{j=1}^{N-1} \sum_{m=j+1}^{N} \hat{\hat{z}}_j \frac{\partial^2 f_m}{\partial z_j \partial z_n} \cdot \hat{z}_m \right) \cdot \frac{\partial z_n}{\partial \psi_i} \end{aligned} \tag{15}$$

where $(\partial^2 f_i)/(\partial z_j \partial \psi_i)$ represents second-order derivative information in the individual neurons.

Next, we define a new back-propagation from the adjoint neural model into the original neural model through EDNs as

$$\overline{\hat{z}}_n = \sum_{j=1}^{N-1} \sum_{\substack{m=\max \\ (j+1, n+1)}}^{N} \hat{\hat{z}}_j \frac{\partial^2 f_m}{\partial z_j \partial z_n} \cdot \hat{z}_m. \tag{16}$$

The last term in (15) can be handled by injecting $\overline{\hat{z}_n}$ into the original neural model as an additional error propagation to be merged together with the error propagation in the original model. Notice that $\hat{z}$, $\hat{\hat{z}}$, and $\overline{\hat{z}}$ are all defined corresponding to the sensitivity of selected neuron k, as in (4), which means $\hat{z}_j = \hat{z}_j^k$, $\hat{\hat{z}}_j = \hat{\hat{z}}_j^k$, and $\overline{\hat{z}}_j = \overline{\hat{z}}_j^k$.

Now we include $\tilde{E}_o$ to consider the derivative for the total training error per sample of (8). Utilizing (15) and (16), we have

$$\begin{aligned}\frac{\partial \tilde{E}}{\partial \boldsymbol{p}_i} &= \frac{\partial \tilde{E}_o}{\partial \boldsymbol{p}_i} + \frac{\partial \tilde{E}_a}{\partial \boldsymbol{p}_i} \\ &= \sum_{k \in K} \left[W_1 \cdot (z_k - d_k) \cdot \frac{\partial z_k}{\partial z_i} \frac{\partial f_i}{\partial \boldsymbol{p}_i} + \sum_{n=i+1}^{N} \overline{\hat{z}}_n^k \right. \\ &\quad \left. \cdot \frac{\partial z_n}{\partial z_i} \frac{\partial f_i}{\partial \boldsymbol{p}_i} + \sum_{j=1}^{i-1} \hat{\hat{z}}_j^k \cdot \frac{\partial^2 f_i}{\partial z_j \partial \boldsymbol{p}_i} \cdot \hat{z}_i^k \right] \end{aligned} \tag{17}$$

where $\tilde{E}_o$ is the original training error for each sample data.

According to (17), there are three concurrent back-propagation paths in our task, corresponding to the three terms in the equation. The first path is that of the training error in the original network, i.e., $W_1 \cdot (z_j - d_j)\ j \in K$, which starts from the output neurons in the original model and back-propagates through the original hidden neurons toward the original input neurons. The second path is that of the adjoint training error, i.e., $G_i = W_2 \cdot (\hat{z}_i - g_{ki}),\ i \in I$, which starts from the adjoint output neurons through the EDNs and into the original neural network toward the original input neurons. The third path is that of the adjoint training error, i.e., $G_i, i \in I$, which starts from output neurons in the adjoint neural model and back-propagates toward the EDNs.

To formulate our training into an efficient and concurrent original/adjoint neural network back-propagation scheme, we further process the first and second paths as follows.

Let $\sigma_j = (\partial \tilde{E})/(\partial z_j)$ be the new combined local gradient representing the original and derivative training error back-propagated to neural j in the original model, i.e., the back-propagation of path 1 and path 2 merged together at neuron j in the original neural model. This combined back-propagation continues toward the original input neurons, merges again with $\bar{\hat{z}}$ (which is the back-propagation from the adjoint model through EDNs to the original model), and arrives at every neuron the combined back-propagation encounters along the way as follows:

$$\sigma_j = \sum_{m=j+1}^{N} \sigma_m \cdot \frac{\partial f_m}{\partial z_j} + D_j + \sum_{k \in K} \bar{\hat{z}}_j^k,$$
$$D_j = \begin{cases} W_1(z_j - d_j), & j \in K \\ 0, & j \notin K \end{cases} \tag{18}$$

where D_j is the training error for back-propagation path 1 at the original output neurons. The derivative required by training due to the first two parts in (17) is then $\sigma_i \cdot (\partial f_i)/(\partial \boldsymbol{p}_i)$.

The final derivative for training the combined original and adjoint model is now

$$\frac{\partial \tilde{E}}{\partial \boldsymbol{p}_i} = \sigma_i \cdot \frac{\partial f_i}{\partial \boldsymbol{p}_i} + \sum_{k \in K} \sum_{j=1}^{i-1} \hat{\hat{z}}_j^{k} \cdot \frac{\partial^2 f_i}{\partial z_j \partial \boldsymbol{p}_i} \cdot \hat{z}_i^k \tag{19}$$

which includes the first- and second-order derivatives. Notice that even though the derivation process is complicated, the final result of (19) is surprisingly simple and elegant, fully compatible with the neural network concept of error propagation. Also notice that the first-order sensitivity analysis in Section III-A requires only one back-propagation, whereas the combined first- and second-order sensitivity technique of (19) requires three error propagation paths with paths 1 and 2 merged as the propagation continues along the way. The proposed method is suitable for incorporation into a microwave neural modeling software.

IV. Examples

A. Example 1

Fast and accurate sensitivity analysis of coupled transmission lines is important for high-speed VLSI interconnect optimization [24] and statistical design. This example illustrates the proposed sensitivity technique for an arbitrary neural network structure where microstrip empirical formulas are used as part of a knowledge-based neural network structure shown in Fig. 7(a). The inputs to our model $(\boldsymbol{x})$ are conductor widths (w_1, w_2), spacing between coupled interconnects (s), substrate thickness (h), dielectric constant (ε_r), and frequency (f). The output of model (y) is mutual inductance L_{12}.

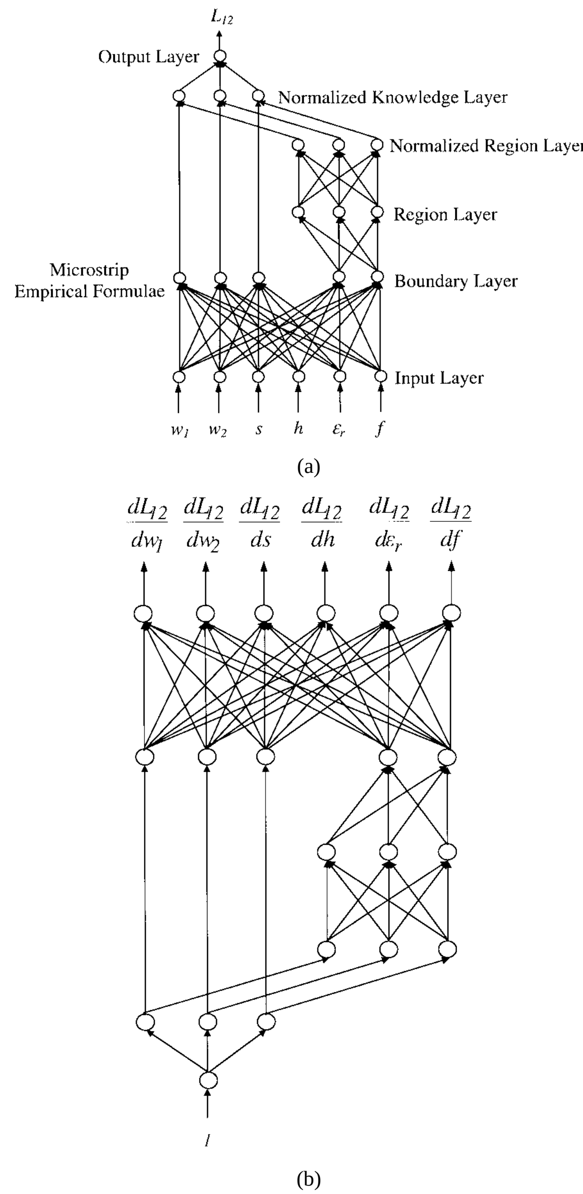


Fig. 7 Knowledge-based coupled transmission-line neural model of mutual inductance (L_{12}) for VLSI interconnect optimization. $w_1, w_2, s, h, \varepsilon_r$, and f are conductor widths, spacing between coupled interconnects, substrate thickness, dielectric constant, and frequency, respectively. (a) Original neural model. (b) Basic adjoint neural model, which will be used by optimization to perform solution space analysis and synthesis of this element.

After training the original model of Fig. 7(a) using *NeuroModeler*[1] with accurate EM-based microstrip data (100 samples) obtained by LINPAR [25], we use the proposed method to provide exact derivatives of electrical parameters of the transmission line with respect to the physical–geometrical parameters needed in VLSI interconnect optimization. The sensitivity solution from the basic adjoint neural model of Fig. 7(b) is verified with brute-force perturbation in Table II. Fig. 8 compares our sensitivity versus that from perturbation as a continuous

[1]*NeuroModeler*, ver. 1.2, Q. J. Zhang, Carleton. Univ., Ottawa, ON, Canada.

TABLE II
EXAMPLE OF SENSITIVITY BETWEEN PERTURBATION TECHNIQUE AND ADJOINT TECHNIQUE FOR THE VLSI INTERCONNET MODELING EXAMPLE. GOOD AGREEMENT IS ACHIEVED

Sensitivity	Perturbation Technique	Adjoint Technique	Difference (%)
dL_{12}/dw_1	-0.1440	-0.1435	0.354
dL_{12}/dw_2	0.0620	0.0616	0.645
dL_{12}/ds	-0.8462	-0.8514	0.610
dL_{12}/dh	0.5338	0.5337	0.018
$dL_{12}/d\varepsilon_r$	-0.0010	-0.0010	0.001
$dL_{12}/dfreq$	-0.0037	-0.0037	0.001

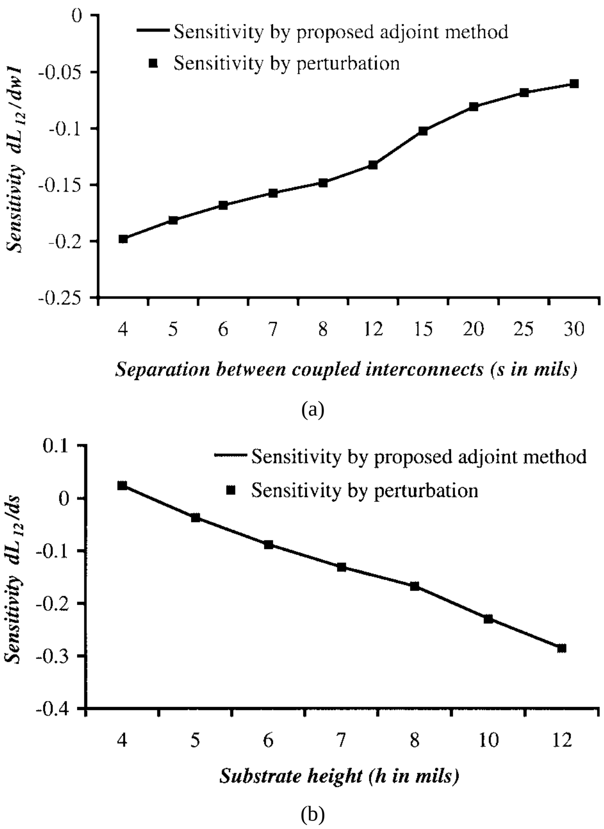


Fig. 8 Sensitivity verification for VLSI interconnect modeling example. (a) dL_{12}/dw_1 versus s. (b) dL_{12}/ds versus h. Good agreement is observed between our sensitivity solution and EM perturbation sensitivity.

function in s and h sub-spaces, respectively. The good agreement in those figures verifies our adjoint model. Notice that the exact sensitivity is obtained through the adjoint neural model without extra training. Without the neural model, such sensitivity would have been computed by perturbation in EM simulators. The computation time for the proposed method compared to EM perturbation solution is 3 s versus 2660 s for sensitivity analysis of 1000 microstrip models, which are typically needed in optimization of a network of VLSI interconnects.

Now we consider an advanced use of the neural model just trained. The purpose is to find the solution of feasible regions of interconnect geometry ($\boldsymbol{x}$ of the neural model) from the given budget on electrical parameters ($\boldsymbol{y}$ of the neural model). This is

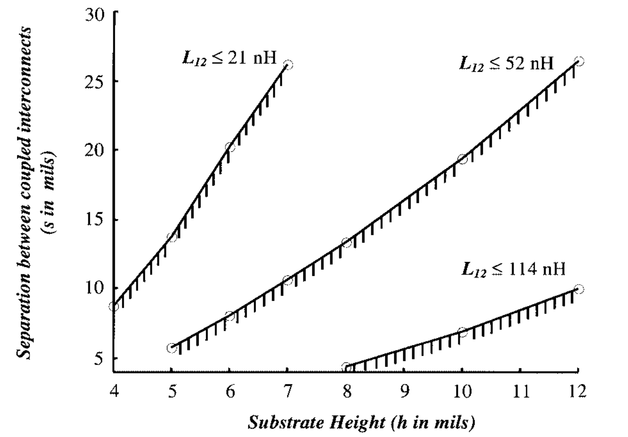


Fig. 9 Solution space analysis. Feasible regions of s–h of VLSI interconnect design for given design budgets on L_{12}. This solution space is obtained after 40 separate optimizations, where gradient information required by optimization is supplied by the adjoint model of Fig. 7(b).

also called design solution space analysis, which is very useful for synthesis of VLSI interconnects and for making tradeoff decisions during early design stages of VLSI systems. A basic step is to use optimization to find inputs $\boldsymbol{x}$ of the neural model from given specifications on $\boldsymbol{y}$. The overall solution space is solved by repeatedly performing such optimization for a variety of $\boldsymbol{y}$ specifications and a variety of $\boldsymbol{x}$ patterns. Fig. 9 shows a solution of feasible space of s versus h for the various given design budgets of mutual inductance L_{12}. This solution is obtained with 40 optimizations of the trained neural model and the gradient information required by optimization are provided by the adjoint model of Fig. 7(b).

B. Example 2

This example illustrates the integration effect of the adjoint neural model. We first train only the adjoint neural model to learn the nonlinear capacitor data, which is generated from *Agilent-ADS*.[2] After training the adjoint model with 41 data samples, we then use the original neural model without re-training (with internal parameters updated according to Section III-E) as a nonlinear charge model (i.e., Q-model). The charge model is compared with analytical integration of the ADS capacitor formula (Fig. 10). The good agreement in this figure verifies the integration effect of training the adjoint neural model. This example shows an interesting solution to one of the frequently encountered obstacles in developing a charge model for nonlinear capacitors required for harmonic balance simulators with only capacitor data available.

C. Example 3

This example shows large-signal device modeling using dc and small-signal training data. The model used is a knowledge-based approach, where the existing intrinsic electrical equivalent-circuit model is combined with neural network learning. In practice, manually creating formulas for the nonlinear currents and charge sources in an FET model could be very time consuming. Here, we use neural networks to automatically learn the unknown relationship of the gate–source charge Q_{gs}, gate–drain

[2] *Agilent-ADS*, ver. 1.5, Agilent Technol., Palo Alto, CA.

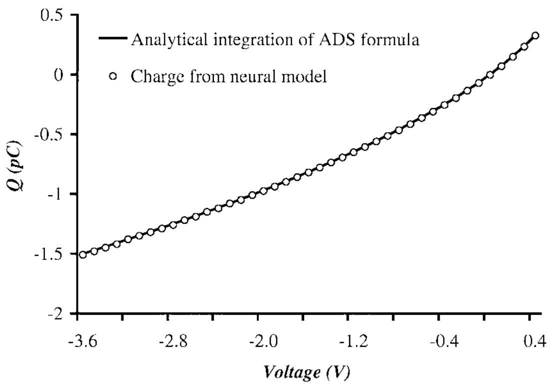


Fig. 10 Comparison of the charge model trained from nonlinear capacitance date with that from analytical integration of ADS capacitance formula. Training was done by training the adjoint neural model from the capacitance data. After training, the original neural model automatically produces the charge model achieving an integration effect of training data. The charge model for nonlinear capacitors is useful for harmonic balance simulation.

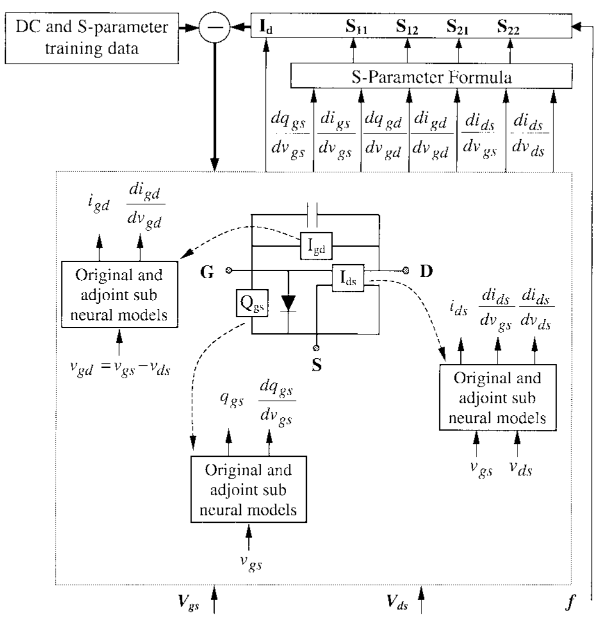


Fig. 11 Large-signal FET modeling including adjoint neural networks trained by dc and bias-dependent S-parameters. Here, adjoint neural networks complement an intrinsic FET equivalent circuit by providing the unknown nonlinear currents ($I_{\rm ds}, I_{\rm gd}$) and charge ($Q_{\rm gs}$). The small-signal S-parameters imply the derivative information of the large-signal model. This example shows combined training of the original neural model to learn dc data and simultaneously the adjoint neural modal to learn the small-signal S-parameter data. Microwave knowledge of a basic equivalent circuit is combined with sub-neural models leading to a knowledge-based approach for FET modeling.

current $I_{\rm gd}$, and drain–source current $I_{\rm ds}$ as nonlinear functions of gate–source and drain–source voltages $V_{\rm gs}$ and $V_{\rm ds}$, respectively. However, we do not have explicitly the charge data $Q_{\rm gs}$ and dynamic currents data $I_{\rm gd}$ and $I_{\rm ds}$ for training the model. The available training data is the dc and bias-dependent S-parameters of the overall FET, which, in our example, is generated using *Agilent-ADS* with the Statz model [26]. Therefore, the neural models and the rest of the FET equivalent circuit are combined into a knowledge-based model and together they are trained to learn the training data, shown in Fig. 11. Both

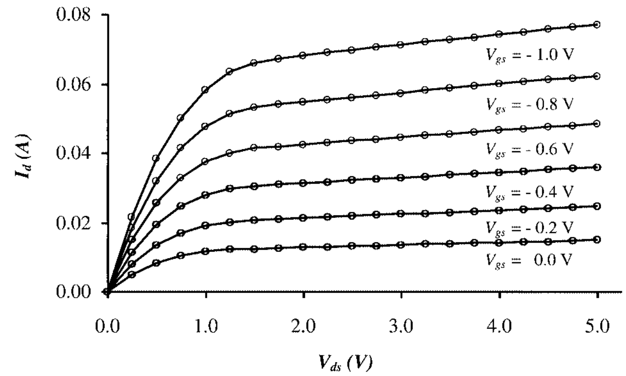


Fig. 12 Comparison between dc curves of the Statz Agilent ADS model (—) and our knowledge-based neural FET model (o).

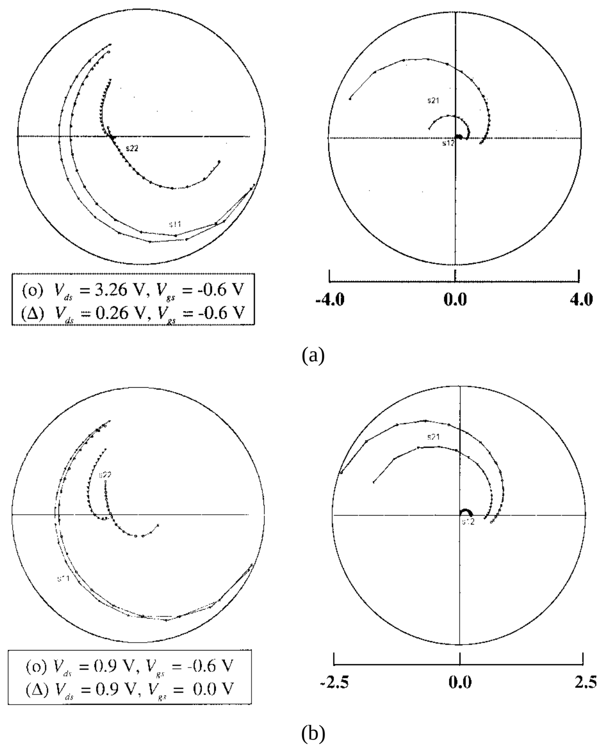


Fig. 13 Comparison between S-parameters of the ADS Statz model (—) and our knowledge-based neural FET model at four of the 90 bias points. (a) $\{V_{\rm ds} = 3.26\ {\rm V},\ V_{\rm gs} = -0.6\ {\rm V}\}$ and $\{V_{\rm ds} = 0.26\ {\rm V},\ V_{\rm gs} = -0.6\ {\rm V}\}$. (b) $\{V_{\rm ds} = 0.9\ {\rm V},\ V_{\rm gs} = -0.6\ {\rm V}\}$ and $\{V_{\rm ds} = 0.9\ {\rm V},\ V_{\rm gs} = -0.0\ {\rm V}\}$

S-parameter data and all the dc-bias data are used for simultaneous training involving all the original and adjoint neural models. Notice that learning S-parameters means learning the derivative information of the large-signal model. After training, a good agreement of dc and small-signal responses at all of the 90 bias points between our knowledge-based neural FET model and those given by the ADS solution is observed, as shown in Figs. 12 and 13.

We then used our complete knowledge-based neural FET model in a three-stage power amplifier, shown in Fig. 14, for large-signal harmonic balance simulation. The large-signal response of the amplifier using our model agrees well with that using the original ADS model illustrated in Fig. 15.

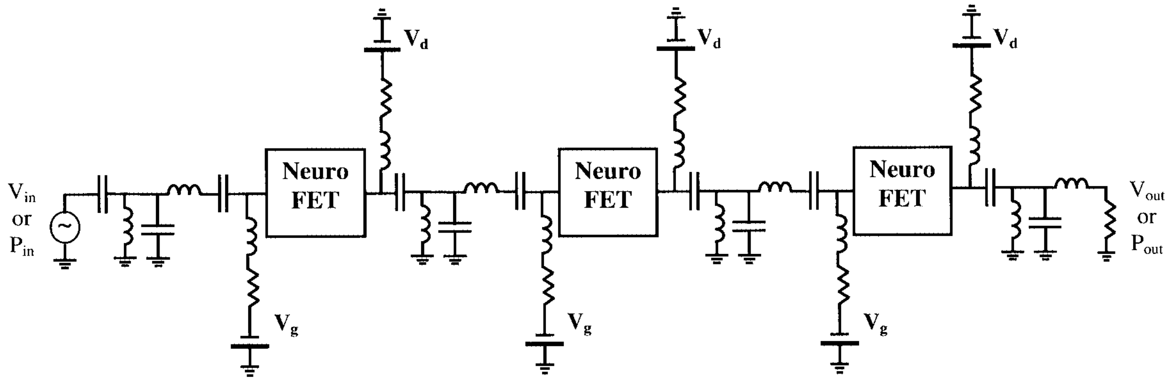
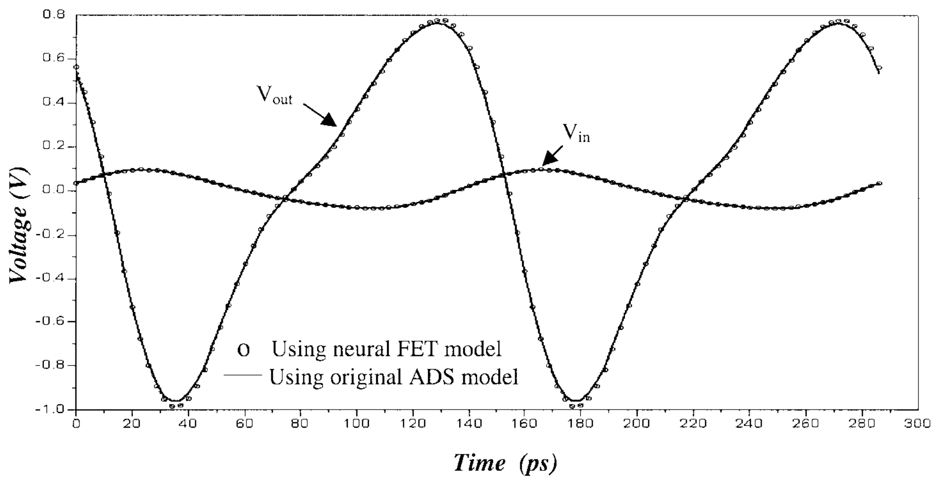
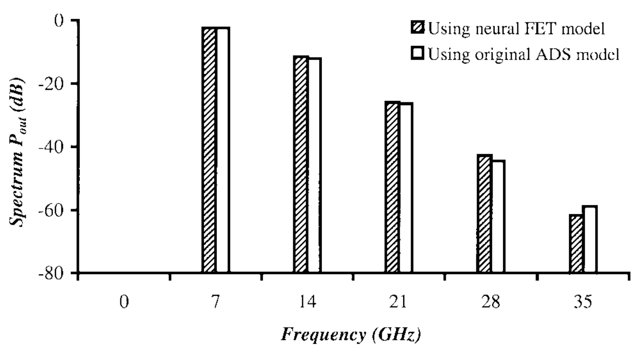


Fig. 14 Three-stage amplifier where the FET models used are knowledged-based neural FET models trained from the proposed method following Fig. 11.



(a)



(b)

Fig. 15 Comparison of the power amplifier large-signal responses. (a) Time-domain amplifier responses using the ADS Statz model and our knowledge-based neural FET model. (b) Output spectrum of the amplifier using the ADS Statz model and our model. The neural model trained with dc and S-parameter data is used here for harmonic balance-based amplifier design, made possible by the proposed approach of training the adjoint model.

Our example demonstrates the capability of the adjoint neural networks in enhancing conventional FET models through adding *trainable* nonlinear current or charge relationships to the model. Such a trainable nonlinear relationship is especially beneficial when analytical formulas in the FET problem are unknown or available formulas are not suitable. By combining adjoint neural networks with the existing FET models, one can improve the models efficiently without having to go through the trial-and-error process typically needed during manual creation of empirical functions. The proposed method provides a new alternative for efficient generation of nonlinear device models for use in large-signal simulation and design.

V. Conclusion

This paper has presented a unified framework for neural-based modeling and sensitivity analysis for generic types of microwave neural models including knowledge-based models. The proposed method provides continuous and differentiable models with analytically consistent derivatives from raw information present in the original training data. A novel and elegant first- and second-order sensitivity analysis scheme allows the training of neural models to learn not only input–output relationships in a microwave component, but also its derivatives. The technique is an important contribution to further realizing the flexibility of neural-based approaches in linear and nonlinear microwave modeling, simulation, and optimization.

REFERENCES

[1] Q. J. Zhang and K. C. Gupta, *Neural Networks for RF and Microwave Design*. Boston, MA: Artech House, 2000.

[2] A. Veluswami, M. S. Nakhla, and Q. J. Zhang, "The application of neural networks to EM-based simulation and optimization of interconnects in high-speed VLSI circuits," *IEEE Trans. Microwave Theory Tech.*, vol. 45, pp. 712–723, May 1997.

[3] M. H. Bakr, J. W. Bandler, M. A. Ismail, J. E. Rayas-Sáanchez, and Q. J. Zhang, "Neural space mapping optimization for EM-based design," *IEEE Trans. Microwave Theory Tech.*, vol. 48, pp. 2307–2315, Dec. 2000.

[4] P. M. Watson and K. C. Gupta, "EM-ANN models for microstrip vias and interconnects in dataset circuit," *IEEE Trans. Microwave Theory Tech.*, vol. 44, pp. 2495–2503, Dec. 1996.

[5] A. Veluswami, Q. J. Zhang, and M. S. Nakhla, "A neural network model for propagation delays in systems with high-speed VLSI interconnect networks," in *Proc. IEEE Custom Integrated Circuits Conf.*, Santa Clara, CA, May 1995, pp. 387–390.

[6] K. Shirakawa *et al.*, "Structural determination of multilayered large-signal neural network HEMT model," *IEEE Trans. Microwave Theory Tech.*, vol. 46, pp. 1367–1375, Oct. 1998.

[7] F. Wang and Q. J. Zhang, "Knowledge based neural models for microwave design," *IEEE Trans. Microwave Theory Tech.*, vol. 45, pp. 2333–2343, Dec. 1997.

[8] P. M. Watson and K. C. Gupta, "Design and optimization of CPW circuits using EM-ANN models for CPW components," *IEEE Trans. Microwave Theory Tech.*, vol. 45, pp. 2515–2523, Dec. 1997.

[9] G. L. Creech, B. J. Paul, C. D. Lesniak, T. J. Jenkins, and M. C. Calcatera, "Artificial neural networks for fast and accurate EM-CAD of microwave circuits," *IEEE Trans. Microwave Theory Tech.*, vol. 45, pp. 794–802, May 1997.

[10] H. Zaabab, Q. J. Zhang, and M. S. Nakhla, "A neural network modeling approach to circuit optimization and statistical design," *IEEE Trans. Microwave Theory Tech.*, vol. 43, pp. 1349–1358, June 1995.

[11] Y. Harhouss, J. Rousset, H. Chéhadé, E. Ngoya, D. Barataud, and J. P. Teyssier, "The use of artificial neural networks in nonlinear microwave devices and circuits modeling: An application to telecommunication system design," *Int. J. RF Microwave Computer-Aided Eng. (Special Issue)*, vol. 9, pp. 198–215, 1999.

[12] V. B. Litovski, J. I. Radjenovic, Z. M. Mrcarica, and S. L. Milenkovic, "MOS transistor modeling using neural network," *Electron. Lett.*, vol. 28, pp. 1766–1768, 1992.

[13] V. K. Devabhaktuni, C. Xi, and Q. J. Zhang, "A neural network approach to the modeling of heterojunction bipolar transistors from S-parameter data," in *Proc. 28th Eur. Microwave Conf.*, Amsterdam, The Netherlands, Oct. 1998, pp. 306–311.

[14] S. Goasguen, S. M. Hammadi, and S. M. El-Ghazaly, "A global modeling approach using artificial neural network," in *IEEE MTT-S Int. Microwave Symp. Dig.*, Anaheim, CA, 1999, pp. 153–156.

[15] P. M. Watson, C. Cho, and K. C. Gupta, "Electromagnetic-artificial neural network model for synthesis of physical dimensions for multilayer asymmetric coupled transmission structures," *Int. J. RF Microwave Computer-Aided Eng. (Special Issue)*, vol. 9, pp. 175–186, 1999.

[16] M. Vai and S. Prasad, "Neural networks in microwave circuit design—Beyond black box models," *Int. J. RF and Microwave Computer Eng. (Special Issue)*, vol. 9, pp. 187–197, 1999.

[17] P. M. Watson, K. C. Gupta, and R. L. Mahajan, "Application of knowledge-based artificial neural network modeling to microwave components," *Int. J. RF and Microwave Computer-Aided Eng. (Special Issue)*, vol. 9, pp. 254–260, 1999.

[18] J. W. Bandler, M. A. Ismail, J. E. Rayas-Sánchez, and Q. J. Zhang, "Neuromodeling of microwave circuits exploiting space mapping technology," *IEEE Trans. Microwave Theory Tech.*, vol. 47, pp. 2417–2427, Dec. 1999.

[19] J. W. Bandler, Q. J. Zhang, and R. Biernacki, "A unified theory for frequency-domain simulation and sensitivity analysis of linear and nonlinear circuits," *IEEE Trans. Microwave Theory Tech.*, vol. 36, pp. 1661–1669, Dec. 1988.

[20] J. W. Bandler and S. Chen, "Circuit optimization: The state of the art," *IEEE Trans. Microwave Theory Tech.*, vol. 36, pp. 424–443, Feb. 1988.

[21] J. W. Bandler, R. M. Biernacki, S. H. Chen, J. Song, S. Ye, and Q. J. Zhang, "Analytically unified DC/small-signal/large-signal circuit design," *IEEE Trans. Microwave Theory Tech.*, vol. 39, pp. 1076–1082, July 1991.

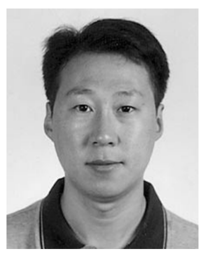
[22] S. Haykin, *Neural Networks, A Comprehensive Foundation*. Englewood Cliffs, NJ: Prentice-Hall, 1994.

[23] G. Antonini and A. Orlandi, "Gradient evaluation for neural-networks-based electromagnetic optimization procedures," *IEEE Trans. Microwave Theory Tech.*, vol. 48, pp. 874–876, May 2000.

[24] Q. J. Zhang, F. Wang, and M. S. Nakhla, "Optimization of high-speed VLSI interconnects: A review," *Int. J. Microwave Millimeter-Wave Computer-Aided Eng.*, vol. 7, pp. 83–107, 1997.

[25] A. Djordjević, R. F. Harrington, T. Sarkar, and M. Baẑdar, *Matrix Parameters for Multiconductor Transmission Lines: Software and User's Manual*. Boston, MA: Artech House, 1989.

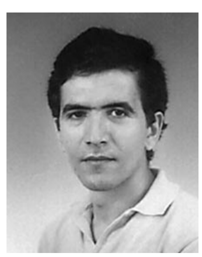
[26] H. Statz, P. Newman, I. W. Smith, R. A. Pucel, and H. A. Haus, "GaAs FET device and circuit simulation in SPICE," *IEEE Trans. Electron Devices*, vol. ED-34, pp. 160–169, Feb. 1987.



Jianjun Xu (S'00) was born in Liaoning, China, in 1975. He received the B.Eng. degree in electrical and electronics engineering from Tianjin University, Tianjin, China, in 1998, and is currently working toward the Ph.D. degree in electronics at Carleton University, Ottawa, ON, Canada.

His research interests include neural network modeling and their applications in computer-aided design for electronics circuits.

Mr. Xu was the recipient of the Student Paper Award at the 2001 IEEE Microwave Theory and Techniques Society (IEEE MTT-S) International Microwave Symposium (IMS), the 2002–2003 Ontario Graduate Scholarship in Science and Technology, and the 2002 Ontario Graduate Student Scholarship.



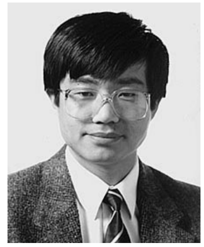
Mustapha C. E. Yagoub (M'96) received the Diplôme d'Ingénieur degree in electronics and the Magister degree in telecommunications from the Ecole Nationale Polytechnique, Algiers, Algeria, in 1979 and 1987, respectively, and the Ph.D. degree from the Institut National Polytechnique, Toulouse, France, in 1994.

He was with the Institute of Electronics, Université des Sciences et de la Technologie Houari Boumédienne, Algiers, Algeria, initially as an Assistant from 1983 to 1991, and then as an Assistant Professor from 1994 to 1999. From 1999 to 2001, he was with the Department of Electronics, Carleton University, Ottawa, ON, Canada, where he was involved with neural network applications in microwave areas. He is currently an Assistant Professor with the School of Information Technology and Engineering, University of Ottawa, Ottawa, ON, Canada. His research interests include neural networks for microwave applications, computer-aided design (CAD) of linear and nonlinear microwave devices and circuits, and applied electromagnetics. He has authored over 70 publications in international journals and conferences. He coauthored *Conception de Circuits Linéaires et Non Linéaires Micro-Ondes* (Toulouse, France: Cépadues, 2000).

Runtao Ding was born in Shanghai, China, in 1938. He received the Diploma degree from Tianjin University, Tianjin, China, in 1961.

Since 1961, he has been with the Department of Electronic Engineering, School of Electronic Information Engineering, Tianjin University, where he is currently a Professor. From 1991 to 1996 and from 1996 to 1999, he was the Chairman of the Department of Electronic Engineering and the Dean of the School of Electronic Information Engineering, respectively. His research interests include nonlinear signal processing, image processing, neural networks, and circuit design.

Prof. Ding was a co-chair of the Technical Program Committee (TPC) of the IEEE APCCAS'2000.



Qi Jun Zhang (S'84–M'87–SM'95) received the B.Eng. degree from the East China Engineering Institute, Nanjing, China, in 1982, and the Ph.D. degree in electrical engineering from McMaster 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, Canada, where he developed advanced microwave optimization software. In 1990, he joined the Department of Electronics, Carleton University, Ottawa, ON, Canada, where he is currently a Professor. His research interests are neural network and optimization methods for high-speed/high-frequency circuit design. He has authored over 150 papers on these topics. He coauthored *Neural Networks for RF and Microwave Design* (Boston, MA: Artech House, 2000), and co-edited *Modeling and Simulation of High-Speed VLSI Interconnects* (Boston, MA: Kluwer, 1994). He was a contributor to *Analog Methods for Computer-Aided Analysis and Diagnosis* (New York: Marcel Dekker, 1988). He was a guest co-editor for the "Special Issue on High-Speed VLSI Interconnects" of the *International Journal of Analog Integrated Circuits and Signal Processing* and was twice a guest editor for the "Special Issues on Applications of ANN to RF and Microwave Design" for the *International Journal of RF and Microwave Computer-Aided Engineering*.

Dr. Zhang is a member of the Professional Engineers of Ontario, Canada.