A Hierarchical Neural Network Approach to the Development of a Library of Neural Models for Microwave Design

Fang Wang, *Student Member, IEEE*, Vijaya Kumar Devabhaktuni, *Student Member, IEEE*, and Qi-Jun Zhang, *Senior Member, IEEE*

***Abstract*—Neural networks recently gained attention as a fast and flexible vehicle to microwave modeling, simulation, and optimization. This paper addresses a new task in this area, namely, the development of libraries of neural models for passive and active components, a task with a potential significance to many microwave simulators. However, developing libraries of neural models is very costly due to massive data generation and repeated neural network training. A new hierarchical neural network approach is presented in this paper, allowing both microwave functional knowledge and library inherent structural knowledge to be incorporated into neural models. The library models are developed through a set of base neural models, which capture the basic characteristics common to the entire library, and high-level neural modules which map the information from base models to the library model outputs. The proposed method substantially reduces the cost of library development through reduced need for data collection and shortened time of training. The technique is demonstrated through transmission line and FET library examples.**

***Index Terms*—Circuit simulation, modeling, neural network.**

Summary of Notations for the Proposed Hierarchical Neural Method for Microwave Library Development

$\boldsymbol{B}_j(.)$	jth base model in the library.
$\boldsymbol{D}^{n,k}$	Desired outputs of the kth training sample for the nth library model.
$\boldsymbol{D}_B^{j,k}$	Desired outputs of the kth training sample for the jth base model.
$\phi^n(i)$	Base model index of the ith low-level module for the nth library model.
$\boldsymbol{H}^n$	High-level neural module for the nth library model.
$\boldsymbol{L}_i^n$	ith low-level neural module for the nth library model.
M_B^j	Total number of training samples for the jth base model.
M^n	Total number of training samples for the nth library model.
N_B	Total number of base models in the library.
$N_{B_j}^n$	Number of times the jth base model is reused in the low-level neural modules of the nth library model.
N_C	Total number of models in a library.
N_L^n	Total number of low-level neural modules in the nth library model.
$\boldsymbol{U}^n$	Input vector of all the low-level neural modules of the nth library model.
$\boldsymbol{u}_i^n$	Input vector of the ith low-level neural module for the nth library model.
$\boldsymbol{U}_i^n(.)$	ith knowledge hub for the nth library model.
$\boldsymbol{V}^n$	Weight vector of the high-level neural module for the nth library model.
$\boldsymbol{W}_j$	Weight vector for the jth base model.
$\boldsymbol{X}^n$	Input vector of the nth library model.
$\boldsymbol{X}^{n,k}$	Input part of the kth training sample for the nth library model.
$\boldsymbol{X}_B^j$	Input vector of the jth base model.
$\boldsymbol{X}_B^{j,k}$	Input part of the kth training sample for the jth base model.
$\boldsymbol{Y}^n$	Output vector of the nth library model.
$\boldsymbol{Y}_B^j$	Output vector of the jth base model.
$\boldsymbol{z}_i^n$	Output vector of the ith low-level neural module for the nth library model.
$\boldsymbol{Z}^n$	Input vector of the high-level neural module of the nth library model.
$\boldsymbol{Z}^{n,k}$	Input part of the kth training sample for the high-level neural module of the nth library model.

Manuscript received April 1, 1998; revised August 28, 1998. This work was supported through the Natural Sciences and Engineering Research Council of Canada and by Micronet: a Canadian Network Center of Excellence on Microelectronic Devices, Circuits, and Systems.

The authors are with the Department of Electronics, Carleton University, Ottawa, Ont., Canada, K1S 5B6 (e-mail: fwang@doe.carleton.ca; vijay@doe.carleton.ca; qjz@doe.carleton.ca).

Publisher Item Identifier S 0018-9480(98)09193-5.

I. Introduction

RECENTLY, a new computer-aided design (CAD) approach based on neural networks has been introduced for modeling of passive and active microwave components [1]–[5] and microwave circuit design [2], [4], [6], [7]. A neural network can be developed by learning and abstracting from microwave data, a process called training. Once trained, the neural network can then be used during microwave design to provide instant answers to the task it learned [2]. The recent work by microwave researchers demonstrated the ability of neural networks to learn and to model a variety of microwave components, such as microstrip interconnects [1], [3], [8], vias [3], spiral inductors [5], FET devices [1], [9], coplanar waveguide (CPW) circuit components [4], and packaging and interconnects [10]. Neural networks have also been used in circuit simulation and optimization [2], [9], [11], microstrip circuit design [12], IC modeling [13] and process design [14], synthesis [6], Smith Chart representation [7], and microwave impedance matching [15]. Neural models can be much faster than original detailed EM/physics models, more accurate than polynomial and empirical models, allow more dimensions than table lookup models, and are easier to develop when a new device/technology is introduced [11]. The costs for developing neural models are mainly data collection and neural network training. Techniques addressing microwave neural model accuracy and efficient model development have been proposed, e.g., [1]–[5].

The success of these works opened the door for an even more exciting possibility, i.e., developing massive neural network models for libraries of microwave components. This is of practical significance, since the realistic power of many CAD tools depends upon the richness, speed, and the accuracy of their library models. For neural models, while the cost for individual model development has been made manageable, e.g., [1]–[5], massively developing neural models for libraries requires massive data generation and repeated model training. This is a highly intensive process, and practically very expensive using today's neural model technique.

In the neural network research community, a recent development called combining neural networks [16] is taking place, addressing issues of network accuracy and training efficiency. Two categories of approaches have been developed: the ensemble-based approach and the modular approach. In the ensemble-based approach [17], [18], several networks are trained such that each network approximates the overall task in its own way. The outputs from these networks are then combined to produce a final output for the combined network. The second category, e.g., [16], [19]–[21], features a modular neural network structure consisting of several neural networks, each optimized to perform a particular subtask of an overall complex operation. An integrating unit then selects or combines the outputs of the networks to form the final output of the modular neural network. Using combined neural network structures instead of a single network, problem knowledge can be incorporated into network structures, and the complexity of the overall problem can be divided and conquered more effectively [22]. This leads to improved overall network reliability or training efficiency [21], [23]. However, most of the existing network structures were motivated from signal processing, classification, and pattern recognition applications. The specific challenges in developing microwave libraries and the suitable structures for incorporating RF/microwave library knowledge remain unanswered.

Motivated by the concept of combining neural networks, we propose a new hierarchical neural network approach for the development of a library of microwave neural models. In the approach, a distinctive set of base neural models is established. The basic microwave functional characteristics common to various models in a library are first extracted and incorporated into base neural models. Then a hierarchical neural network is constructed for each model of the library with low-level modules realized by base neural models. A high-level neural module is trained to map the low-level module solution to the final output of the microwave component model for each model in the library. Examples of transmission line neural model libraries, useful for design of high-speed VLSI interconnects, and a library of physics-based MESFET devices, are developed. Compared to standard neural model techniques, the proposed hierarchical neural network approach substantially reduces the cost of library development due to less data collection and shorter training time, and at the same time improves model reliability.

II. Proposed Hierarchical Neural Network Approach for Library Development

A. Problem Statement: Library Development

The objective is to develop libraries of passive and active microwave component models. Suppose a library consists of N_C microwave component models. For each model, say, the nth model in the library, the input and output parameters are represented by vectors $\boldsymbol{X}^n$ and $\boldsymbol{Y}^n$, respectively. The library development is to create models to represent the multidimensional nonlinear relationship of

$$\boldsymbol{Y}^n = \boldsymbol{Y}^n(\boldsymbol{X}^n) \tag{1}$$

for each value of $n, n = 1, 2, \cdots, N_C$. We call the spaces spanned by $\boldsymbol{X}^n$ and $\boldsymbol{Y}^n$ as the $\boldsymbol{X}^n$ space and the $\boldsymbol{Y}^n$ space, respectively.

For example, to model a multiconductor transmission line for use in designing high-speed VLSI interconnects, $\boldsymbol{Y}^n$ could represent self and mutual inductances of the coupled conductors. $\boldsymbol{X}^n$ could represent the physical/geometrical parameters of the transmission line such as conductor width, separation between coupled conductors, substrate height, and dielectric constants. Many such neural models would be needed in order to cover a variety of transmission lines in a VLSI interconnect design, such as single-conductor line, dual strip lines, three-conductor coupled lines, etc., leading to the need for a library of transmission line models, such as the strip line library of Fig. 1. Using the standard neural model approach, e.g., multilayer perceptron structure (MLP), costly data collection and extensive model training have to be performed for each model in the library. The total cost for library development will be very high.

B. Base Models

In the proposed approach, we first develop a set of base models to capture the basic electrical or microwave characteristics common to the entire set of models in the library. For example, in a library of various multiconductor transmission line models, the self-inductance of a conductor is one of the common characteristics needed for all the models in the library. Let $\boldsymbol{X}_B^j$ and $\boldsymbol{Y}_B^j$ be vectors representing the inputs and outputs of the jth base model, $j = 1, \cdots, N_B$, where N_B is the total number of base models in the library. Let the two spaces spanned by $\boldsymbol{X}_B^j$ and $\boldsymbol{Y}_B^j$ be called the $\boldsymbol{X}_B^j$ space and the $\boldsymbol{Y}_B^j$ space, respectively. The jth base model, realized by a neural network, relates $\boldsymbol{X}_B^j$ and $\boldsymbol{Y}_B^j$ by

$$\boldsymbol{Y}_B^j = \boldsymbol{B}_j(\boldsymbol{X}_B^j, \boldsymbol{W}_j) \tag{2}$$

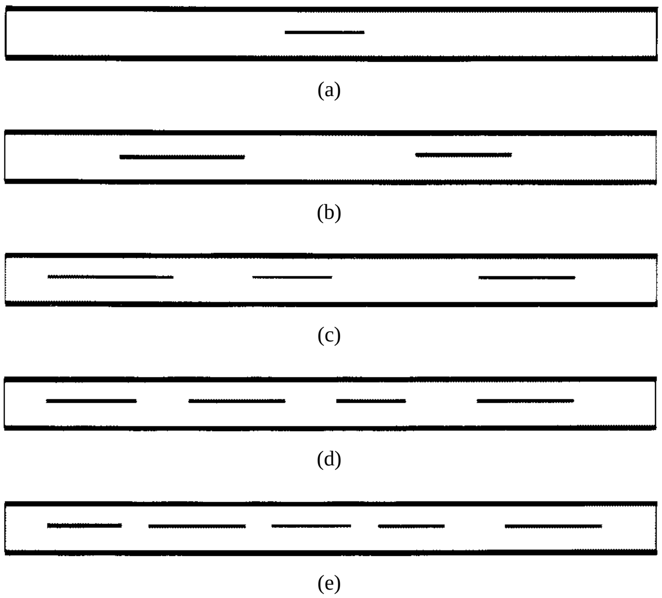

Fig. 1. A library of stripline models. The nth model in the library represents an n-conductor coupled stripline component. (a) $n = 1$, (b) $n = 2$, (c) $n = 3$, (d) $n = 4$, (e) $n = 5$.

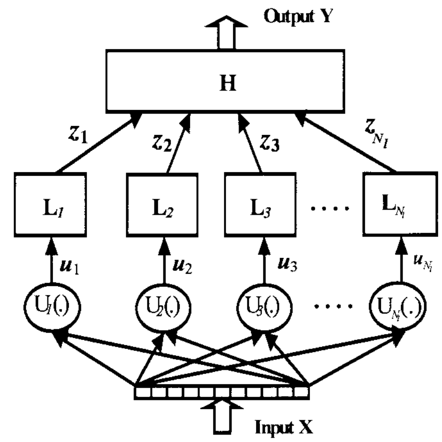

Fig. 2. The proposed hierarchical neural network structure. $\boldsymbol{X}$ and $\boldsymbol{Y}$ represent the inputs and outputs of the overall network. $\boldsymbol{L}_i$ is the ith low-level module with an associated the ith knowledge hub $U_i(\cdot)$. $\boldsymbol{u}$ and $\boldsymbol{z}$ represent the inputs and outputs of low-level modules. This structure can be used for each model in a library. For example, for the nth model, $\boldsymbol{Y} = \boldsymbol{Y}^n$, $\boldsymbol{H} = \boldsymbol{H}^n$, $Z_i = Z_i^n$, $L_i = L_i^n$, $\boldsymbol{u}_i = \boldsymbol{u}_i^n$, $\boldsymbol{U}_i = \boldsymbol{U}_i^n$, and $\boldsymbol{X} = \boldsymbol{X}^n$.

where $\boldsymbol{B}_j$ represents the jth base model and $\boldsymbol{W}_j$ is a vector including all the neural network weights of the jth base model.

The definition of the $\boldsymbol{X}_B^j$ (or $\boldsymbol{Y}_B^j$) spaces is done by choosing a form of space conversion between $\boldsymbol{X}_B^j$ and $\boldsymbol{X}^n$ (or, $\boldsymbol{Y}_B^j$ and $\boldsymbol{Y}^n$), and/or examining the common characteristics in the library. Examples of typical forms of space conversion are: same-space mapping, subspace mapping, or linear transformation. Notice that the terminology "space mapping" used here is in a totally different environment from that in [24] where the terminology was used for a new optimization algorithm. In our paper, the mapping between $\boldsymbol{X}_B^j$ and $\boldsymbol{X}^n$ (or, $\boldsymbol{Y}_B^j$ and $\boldsymbol{Y}^n$) is called same-space mapping if they contain the same set of parameters with equal or nonequal values. Subspace mapping means that $\boldsymbol{X}_B^j$ is a subspace of $\boldsymbol{X}^n$. Linear transformation means that $\boldsymbol{X}_B^j$ is obtained by linearly transforming $\boldsymbol{X}^n$. Examples of each of these cases will be illustrated in Section III of the paper.

For microwave design, empirical formulas often exist approximating such base relationship; for example, the empirical approximation of self-inductance of a transmission line (i.e., $\boldsymbol{Y}_B^j$) as a function of physical/geometrical parameters (i.e., $\boldsymbol{X}_B^j$) [25]. In this case, functional knowledge-based neural networks (KBNN) [1] could be used incorporating such empirical formulas into base neural models, further enhancing their reliability.

Suppose $(\boldsymbol{X}_B^{j,k}, \boldsymbol{D}_B^{j,k})$ are pairs of training samples for the jth base model, where $k = 1, \cdots, M_B^j$, and M_B^j is the total number of training samples. Base neural models should be trained such that

$$\min_{W_j} \sum_{k=1}^{M_B^j} \left\| \boldsymbol{B}_j(\boldsymbol{X}_B^{j,k}, \boldsymbol{W}_j) - \boldsymbol{D}_B^{j,k} \right\|^2 \tag{3}$$

for $j = 1, \cdots, N_B$. Each base model is trained by sufficient number of samples to a high accuracy. The training of all the base models can be considered as an overhead for the library development. This task is done only once in the beginning of library development. The benefit of creating these base models is realized when we subsequently reuse them and combine them in developing many component models for the library.

C. Hierarchical Neural Model

For each model in the library, a hierarchical neural network structure is defined as shown in Fig. 2. The purpose of this structure is to construct an overall model from several modules so that the library base relationship can be maximally reused for every model throughout the library. This structure consists of a high-level neural module denoted as $\boldsymbol{H}^n$ and several low-level neural modules denoted as $\boldsymbol{L}_i^n$, $i = 1, \cdots, N_L^n$. The low-level modules are realized by directly using the base models. Let index function $j = \phi^n(i)$ be defined such that base model $\boldsymbol{B}_j$ is selected as the ith low-level neural module and

$$j = \phi^n(i) = \begin{cases} 1, & \text{for } 0 < i \le N_{B_1}^n \\ 2, & \text{for } N_{B_1}^n < i \le N_{B_1}^n + N_{B_2}^n \\ \cdots, & \cdots \\ N_B, & \text{for } \sum_{k=1}^{N_B - 1} N_{B_k}^n < i \le \sum_{k=1}^{N_B} N_{B_k}^n \end{cases} \tag{4}$$

where $N_{B_j}^n$ is the number of times $\boldsymbol{B}_j$ is reused in the low level of the nth library model. Let $\boldsymbol{u}$ and $\boldsymbol{z}$ be vectors representing the inputs and outputs of low-level modules. Since the ith low-level module of the nth library model is realized by the $\phi^n(i)$th base model,

$$\boldsymbol{L}_i^n = \boldsymbol{B}_j, \; \boldsymbol{u}_i^n = \boldsymbol{X}_B^j, \; \boldsymbol{z}_i^n = \boldsymbol{Y}_B^j, \quad \text{where } j = \phi^n(i) \tag{5}$$

where $\boldsymbol{u}_i^n$ and $\boldsymbol{z}_i^n$ represent the input and output vectors of low-level module i in library model n. For each $\boldsymbol{L}_i^n$, i.e., the ith low-level module, we define a structural knowledge hub $\boldsymbol{U}_i^n(.)$, such that it extracts inputs only relevant to the $\phi^n(i)$th base model out of $\boldsymbol{X}^n$ based on the particular configuration of the nth library component, i.e.,

$$\boldsymbol{u}_i^n = \boldsymbol{X}_B^j = \boldsymbol{U}_i^n(\boldsymbol{X}^n), \text{ where } j = \phi^n(i). \tag{6}$$

The low-level neural modules produce $\boldsymbol{z}^n$ by recalling the trained base models in the library

$$\boldsymbol{z}_i^n = \boldsymbol{L}_i^n(\boldsymbol{u}_i^n) = \boldsymbol{B}_{\phi(i)}(\boldsymbol{u}_i^n, \boldsymbol{W}_{\phi(i)}), \; i = 1, 2, \cdots, N_L^n \tag{7}$$

where $\boldsymbol{W}_{\phi(i)}$ are the weights of the $\phi(i)$th base model, and $\phi(i) = \phi^n(i)$. Let vectors $\boldsymbol{U}^n$ and $\boldsymbol{Z}^n$ be defined by concatenating the $\boldsymbol{u}_i^n$ and the $\boldsymbol{z}_i^n$ for all i, $i = 1, 2, \cdots, N_L^n$, respectively, i.e.,

$$\boldsymbol{U}^n = \begin{bmatrix} \boldsymbol{u}_1^n \\ \boldsymbol{u}_2^n \\ \vdots \\ \boldsymbol{u}_{N_L{}^n}^n \end{bmatrix}, \quad \boldsymbol{Z}^n = \begin{bmatrix} \boldsymbol{z}_1^n \\ \boldsymbol{z}_2^n \\ \vdots \\ \boldsymbol{z}_{N_L{}^n}^n \end{bmatrix}. \tag{8}$$

All the low-level modules combined provide a map from the $\boldsymbol{U}^n$ space to the $\boldsymbol{Z}^n$ space. A high-level module $\boldsymbol{H}^n$ is defined mapping the $\boldsymbol{Z}^n$ space to the $\boldsymbol{Y}^n$ space for each nth model in the library. The high-level module is realized by a neural network

$$\boldsymbol{Y}^n = \boldsymbol{H}^n(\boldsymbol{Z}^n, \boldsymbol{V}^n) \tag{9}$$

where $\boldsymbol{V}^n$ includes all neural network weights for module $\boldsymbol{H}^n$. The relationship in (9) is much easier to model than the original $\boldsymbol{Y}^n = \boldsymbol{Y}^n(\boldsymbol{X}^n)$ relationship since much information is already contained in the base models in the low level. For example, even a linear two-layer perceptron for $\boldsymbol{H}^n$ might be sufficient to produce the final $\boldsymbol{Y}^n$. Consequently, the amount of data needed to train $\boldsymbol{H}^n$ is much less than that for training standard MLP to learn original $\boldsymbol{Y}^n = \boldsymbol{Y}^n(\boldsymbol{X}^n)$.

Suppose ($\boldsymbol{X}^{n,k}$, $\boldsymbol{D}^{n,k}$) are pairs of training samples for the nth library model, where $k = 1, \cdots, M^n$, and M^n is the total number of training samples. The $\boldsymbol{X}^{n,k}$ data are mapped to the $\boldsymbol{U}^n$ space through knowledge hubs and then feed-forwarded through the low-level modules (i.e., various reuse of base neural models) into the $\boldsymbol{Z}^n$ space. Consequently, a new set of training samples, denoted by pairs of ($\boldsymbol{Z}^{n,k}$, $\boldsymbol{D}^{n,k}$) is obtained, where $\boldsymbol{Z}^{n,k}$ is the vector constructed by concatenating $\boldsymbol{z}_i^{n,k} = \boldsymbol{L}_i^n \; (\boldsymbol{U}_i^n(\boldsymbol{X}^{n,k}))$, for all i, $i = 1, \cdots, N_L^n$. The high-level neural module $\boldsymbol{H}^n$ should be trained such that

$$\min_{\boldsymbol{V}^n} \sum_{k=1}^{M^n} \left\| \boldsymbol{H}^n(\boldsymbol{Z}^{n,k}, \boldsymbol{V}^n) - \boldsymbol{D}^{n,k} \right\|^2, \quad \text{for each } n, \; n = 1, \cdots, N_c. \tag{10}$$

With a linear two-layer perceptron neural network as the high-level module, this optimization is simply a quadratic programming problem. In this case, any training method will easily and quickly lead to a globally optimal training solution. This is in contrast to standard MLP approach with the original nonlinear $\boldsymbol{Y}^n(\boldsymbol{X}^n)$ relationship, where training usually takes a long time and might end at a local optimal solution of the neural network, further prolonging the training process.

Under the proposed method, the training of the high-level module is the only training needed for each model in the library. No training is needed for the low-level modules because all such modules are a reuse of the same set of base models which were trained only once in the beginning of library development.

D. Algorithm for Overall Library Development

The overall library development is summarized in the following steps:

Step 1: Define the input and output spaces of base models, i.e., $\boldsymbol{X}_B^j$ and $\boldsymbol{Y}_B^j$, for $j = 1, 2, \cdots, N_B$, and extract basic characteristics from library, using microwave empirical knowledge if available.

Step 2: Collect training data corresponding to each base model input and output, i.e., generate sample data ($\boldsymbol{X}_B^{j,k}$, $\boldsymbol{D}_B^{j,k}$) for the jth base model, where $k = 1, 2, \cdots, M_B^j$ and $j = 1, 2, \cdots, N_B$.

Remark: Training data for base models should be adequate in order to obtain reliable base models.

Step 3: Construct and train base neural models incorporating the knowledge from Step 1. Specifically, solve the optimization problem of (3) to find $\boldsymbol{W}_j$ such that $\boldsymbol{B}_j(\boldsymbol{u}, \boldsymbol{W}_j)$ matches base model training data, for $j = 1, \cdots, N_B$. Let $n = 1$.

Remark: Steps 1–3 are done in the beginning of library development and are considered overhead effort for the library. The next several steps, i.e., Steps 4–8, are the incremental effort for each component model in the library.

Step 4: According to the base model input space definition in Step 1, set up the structural knowledge hubs $\boldsymbol{u}_i^n = \boldsymbol{U}_i^n(\boldsymbol{X}^n)$, which map the model input space $\boldsymbol{X}^n$ into base model input space $\boldsymbol{X}_B^j$, where $j = \phi^n(i)$ as defined in (4), and $i = 1, \cdots, N_L^n$. This automatically sets up the low-level modules.

Step 5: Collect training data corresponding to the model in the library, i.e., generate sample data ($\boldsymbol{X}^{n,k}$, $\boldsymbol{D}^{n,k}$) for the nth model, where $k = 1, 2, \cdots, M^n$.

Remark: Only a small amount of training data is needed here under the proposed technique.

Step 6: Map the $\boldsymbol{X}^{n,k}$ data into the $\boldsymbol{Z}^n$ space through knowledge hubs and low-level modules following (6) and (7).

Step 7: Train the high-level neural module $\boldsymbol{H}^n$, i.e., solve the optimization problem of (10) to find $\boldsymbol{V}^n$ such that the outputs of the high-level module match training data.

Remark: This training step is very easy and fast since the module $\boldsymbol{H}^n$ is very simple and in most cases, a simple linear two-layer perceptron network. Therefore, only a small and incremental effort is needed to obtain each model in the library.

Step 8: If $n = N_C$, then stop, otherwise proceed to train the next library model by setting $n = n + 1$ and go to Step 4.

The algorithm described above permits the hierarchical neural models to be developed systematically and enables the library development process to be maximally automated.

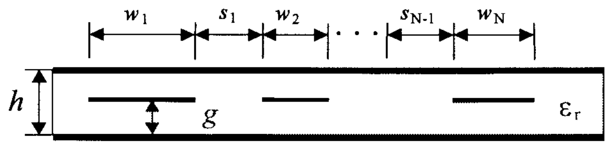

Fig. 3. Details of a typical N-conductor stripline component showing the physical and geometrical parameters.

E. Discussions

Our formulation allows the standard MLP approach to library development as an extreme special case in our theory. To illustrate this case, consider each library model as a base model, and $N_B = N_C$. The base model input and output spaces are defined the same as the library model input and output spaces, i.e.,

$$\boldsymbol{X}_B^j = \boldsymbol{X}^n,\ \boldsymbol{Y}_B^j = \boldsymbol{Y}^n,\ j = n \ \text{ and } n = 1, 2, \cdots, N_c. \tag{11}$$

There is only one low-level module in each library model. The knowledge hub is simply a relay block passing $\boldsymbol{X}^n$ directly to the $\boldsymbol{U}^n$ space

$$\boldsymbol{u}_1^n = \boldsymbol{U}_1^n(\boldsymbol{X}^n) = \boldsymbol{X}^n. \tag{12}$$

The high-level module $\boldsymbol{H}^n$ will also perform a relay from $\boldsymbol{Z}^n$ space to the $\boldsymbol{Y}^n$ space. Therefore, in the worst extreme case where basic characteristics common to various models in the same library are difficult to identify, our technique falls back to the standard MLP approach.

However, in many practical cases, models are grouped into a library due to certain common features. The proposed approach becomes very advantageous when a few base models can capture the common characteristics in a library of many models as demonstrated through the examples in the next section.

III. Examples

A. Example 1—Library of Stripline Models

Multiconductor transmission line models are essential for delay and crosstalk analysis in high-speed VLSI interconnect design [11]. EM simulation of transmission line responses is slow especially if it needs to be repetitively evaluated. Neural models, trained off-line from EM data, can be used online during VLSI interconnect design providing instant solutions of the original EM problem. For practical VLSI interconnect design, libraries of one-conductor, two-conductor, . . . , N-conductor transmission line models are needed. A brute force approach is to train each library model separately, requiring massive data generation and training. Here we apply the proposed hierarchical approach to the development of a library of neural models for N-conductor striplines shown in Fig. 3 for different values of N. In this example, the modeling of self and mutual inductances is presented for illustration. There are five models in the library, $n = 1, 2, 3, 4, 5$ as shown in Fig. 1. And for each nth model, the number of conductors $N = n$. Table I defines the notations for input and output parameters of stripline neural models and the effective range of their input parameters. The detailed list of input and output parameters of each model in the stripline library is shown in Table II. Training and test data were obtained using LINPAR [26] simulator which is based on the method of moments.

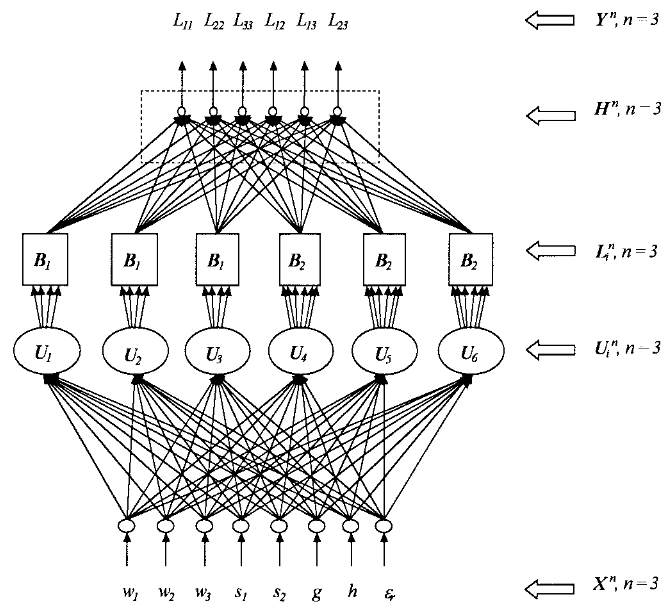

Fig. 4. The hierarchical neural model for the third model in the stripline library, i.e., $n = 3$.

1) Base Model Selections: Two base models, $\boldsymbol{B}_1$ for self-inductance and $\boldsymbol{B}_2$ for mutual inductance are defined. The inputs to the base models include physical/geometrical parameters such as conductor width (w), conductor height (g), substrate height (h), separation between conductors (s), and relative dielectric constant (ε_r). The outputs of $\boldsymbol{B}_1$ and $\boldsymbol{B}_2$ are self and mutual inductances, respectively. Since for any model in the library, shown in Fig. 1, the relation between the self-inductance of a single conductor (and the mutual inductance between two conductors) and the corresponding physical/geometrical parameters is always a useful partial solution to the modeling problem, these two base models do represent basic characteristics useful to all the five stripline models in the entire library. The stripline empirical formulas in [25] are adopted as functional knowledge incorporated into the KBNN's [1], which are the realizations of the base models $\boldsymbol{B}_1$ and $\boldsymbol{B}_2$. The KBNN structural parameters are represented by number of boundary and knowledge neurons, e.g., b2z3 representing two boundary and three knowledge neurons [1]. The base models $\boldsymbol{B}_1$ and $\boldsymbol{B}_2$ are trained to an average testing accuracy of 0.39 and 0.16%, respectively, as shown in Table III. Linear transformation is used as the form of space mapping between $\boldsymbol{X}_B^j$ (of Table III) and $\boldsymbol{X}^n$ (Table II). Subspace mapping is used between $\boldsymbol{Y}_B^j$ (of Table III) and $\boldsymbol{Y}^n$ (Table II). The number of times base models $\boldsymbol{B}_1$ and $\boldsymbol{B}_2$ are reused in each library model, i.e., N_{Bj}^n, is shown in Table II.

2) Example of Library Model $(n = 1)$: For $n = 1$, the library model is for a single conductor transmission line and is directly the base model $\boldsymbol{B}_1$. Therefore, $N_{B1}^n = 1$, $N_{B2}^n = 0$. Knowledge hub is $\boldsymbol{U}_1^n(\boldsymbol{X}^n) = \boldsymbol{X}^n$. Low-level module is $\boldsymbol{L}_1^n = \boldsymbol{B}_1$ and high-level module is $\boldsymbol{H}^n(\boldsymbol{Z}^n) = \boldsymbol{Z}^n$.

TABLE I
NOTATIONS FOR INPUT AND OUTPUT PARAMETERS OF STRIPLINE NEURAL MODELS AND THE EFFECTIVE RANGE OF THEIR INPUT PARAMETERS

Parameters	Notation	Range
the *ith* conductor width	w_i	0.05 *mm* – 0.25 *mm*
the separation between the *ith* and *i+1 th* conductors	s_i	0.1 *mm* – 0.82 *mm*
conductor height above ground	g	0.08 *mm* – 0.25 *mm*
substrate height	h	0.16 *mm* – 0.5 *mm*
relative dielectric constant	ε_r	2 – 10.2
self inductance of the *ith* conductor	L_{ii}	N/A
mutual inductance between the *ith* and *jth* conductors	L_{ij}	N/A

TABLE II
STRIPLINE LIBRARY MODELS

Library Model index n	Model name	Neural model inputs X^n	Neural model outputs Y^n	Number of times each base model (B_1, B_2) is used N_{Bj}^n x B_j
$n=1$	1 conductor stripline model	$w\ g\ h\ \varepsilon_r$	L_{11}	1 x B_1
$n=2$	2 conductor stripline model	$w_1\ w_2\ s\ g\ h\ \varepsilon_r$	$L_{11}, L_{12}, L_{22},$	2 x B_1 , 1 x B_2
$n=3$	3 conductor stripline model	$w_1\ w_2\ w_3\ s_1\ s_2\ g\ h\ \varepsilon_r$	$L_{11}, L_{12}, L_{13}, L_{22}, L_{23}, L_{33}$	3 x B_1 , 3 x B_2
$n=4$	4 conductor stripline model	$w_1\ w_2\ w_3\ w_4\ s_1\ s_2\ s_3\ g\ h\ \varepsilon_r$	$L_{11}, L_{12}, L_{13}, L_{14}, L_{22}, L_{23}, L_{24}, L_{33}, L_{34}, L_{44}$	4 x B_1 , 6 x B_2
$n=5$	5 conductor stripline model	$w_1\ w_2\ w_3\ w_4\ w_5\ s_1\ s_2\ s_3\ s_4\ g\ h\ \varepsilon_r$	$L_{11}, L_{12}, L_{13}, L_{14}, L_{15}, L_{22}, L_{23}, L_{24}, L_{25}, L_{33}, L_{34}, L_{35}, L_{44}, L_{45}, L_{55}$	5 x B_1 , 10 x B_2

TABLE III
BASE MODELS FOR STRIPLINE LIBRARY

Base model index j	Base model symbol B_j	Base model inputs $X_B{}^j$	Base model Outputs $Y_B{}^j$	Base model structure (KBNN)	Model accuracy (average error)
1	B_1	$w\ g\ h\ \varepsilon_r$	self inductance L	*b2z3*	0.39%
2	B_2	$w_1\ w_2\ s\ g\ h\ \varepsilon_r$	mutual inductance L_m	*b4z6*	0.16%

3) Example of Library Model $(n = 3)$*:*

For $n = 3$, the library model is for a three-conductor coupled stripline component. We reuse the base models as the lower level neural modules shown in Fig. 4. There are six low-level neural modules. The knowledge hubs for this library model are defined in Table IV. The six low-level neural modules make a preliminary prediction of self-inductance for each of the three conductors and mutual inductance between each pair of conductors, i.e., conductors 1 and 2, conductors 2 and 3, and conductors 1 and 3. The high-level neural module $\boldsymbol{H}^3$ is realized by a two-layer perceptron with six inputs (i.e., the preliminary predictions of self and mutual inductances from the six low-level modules) and six outputs (i.e., the final and refined predictions of self and mutual inductances of the overall three-conductor stripline model) and with linear functions in all output neurons. This is a linear combination

TABLE IV
LOW-LEVEL MODULES AND STRUCTURAL KNOWLEDGE HUBS FOR THREE-CONDUCTOR STRIPLINE, I.E., LIBRARY MODEL $n = 3$

Low level modules L_i^n	Inputs to module/Knowledge hub $u_i^n = U_i^n(X^n)$	Index function $\phi^n(i) = j$	Base model used $B_{\phi(i)}$
L_1^3	$u_1^3 = U_1^3(X^3) = [w_1\ g\ h\ \varepsilon_r]$	$\phi^3(1) = 1$	B_1
L_2^3	$u_2^3 = U_2^3(X^3) = [w_2\ g\ h\ \varepsilon_r]$	$\phi^3(2) = 1$	B_1
L_3^3	$u_3^3 = U_3^3(X^3) = [w_3\ g\ h\ \varepsilon_r]$	$\phi^3(3) = 1$	B_1
L_4^3	$u_4^3 = U_4^3(X^3) = [w_1\ w_2\ s_1\ g\ h\ \varepsilon_r]$	$\phi^3(4) = 2$	B_2
L_5^3	$u_5^3 = U_5^3(X^3) = [w_2\ w_3\ s_2\ g\ h\ \varepsilon_r]$	$\phi^3(5) = 2$	B_2
L_6^3	$u_6^3 = U_6^3(X^3) = [w_1\ w_3\ s_1{+}s_2\ g\ h\ \varepsilon_r]$	$\phi^3(6) = 2$	B_2

TABLE V
MODEL ACCURACY COMPARISON (AVERAGE ERROR ON TEST DATA) BETWEEN STANDARD MLP AND THE PROPOSED MODEL FOR THREE-CONDUCTOR STRIPLINE MODEL

No. of training samples	MLP (8-8-6)	MLP (8-12-6)	MLP (8-16-6)	Proposed model
15	15.40%	14.25%	14.17%	0.52%
25	10.61%	9.66%	9.96%	0.48%
50	4.01%	1.79%	5.30%	0.41%
100	1.36%	0.96%	1.80%	0.39%
300	0.87%	0.83%	0.86%	0.38%
500	0.84%	0.73%	0.79%	0.39%

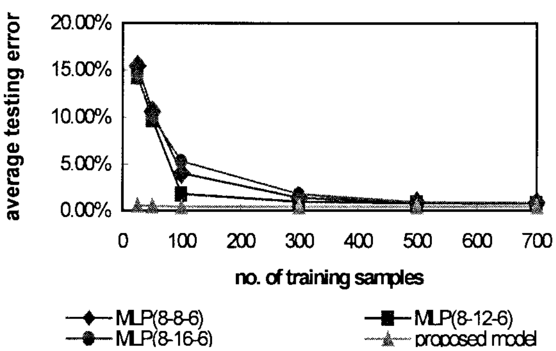

Fig. 5. Model accuracy comparison (average error on test data) between standard MLP and the proposed model for three-conductor stripline model.

of low-level neural modules with no gating functions taking advantage of modular neural network concept. Each low-level neural module provides a portion of the inductance prediction contributing to the overall inductance prediction at the high-level neural module. Only a small amount of training data (15 samples) is needed to train this high-level module of a three-conductor stripline model since the preliminary relationships of the model have already been captured in the base models. However, with the conventional MLP neural model (8, 12, and 16 hidden neurons), 500 samples are needed to achieve a model of similar accuracy, shown in Table V.

The tendency of library model accuracy versus the amount of training data is plotted in Fig. 5. As available training data becomes less and less, the error of standard MLP grows quickly, but the proposed hierarchical library model remains reasonable and reliable.

4) All Library Models: All library models, $n = 2, 3, 4, \cdots$ in the library, can be developed systematically in a similar way

TABLE VI
COMPARISON OF NUMBER OF TRAINING SAMPLES NEEDED AND LIBRARY MODEL ACCURACY FOR STRIPLINE LIBRARY WHEN DEVELOPED BY STANDARD MLP AND THE PROPOSED HIERARCHICAL NEURAL STRUCTURE, RESPECTIVELY

Library model index	Stripline model name	Number of training samples needed, (and corresponding model accuracy)	
		Standard MLP	Proposed model
Overhead for base models		0	264[1] + 300[2]
$n = 1$	1 conductor stripline model	**264**, (0.42%)	**0**, (0.39%)
$n = 2$	2 conductor stripline model	**400**, (0.75%)	**10**, (0.56%)
$n = 3$	3 conductor striplinemodel	**500**, (0.73%)	**15**, (0.52%)
$n = 4$	4 conductor stripline model	**700**, (0.78%)	**25**, (0.74%)
$n = 5$	5 conductor stripline model	**900**, (0.99%)	**35**, (0.63%)
	stripline library	Total = **2764**	Total = **649**

[1]: base model B_1 training
[2]: base model B_2 training

as model #3. It should be noted that efforts in developing those additional library models are small and incremental, since only few training data is needed, and only the high-level neural module $\boldsymbol{H}^n$ needs to be trained for each n.

5) Overall Library Accuracy and Development Cost—A Comparison: Using standard MLP for each library model, the total amount of training data needed for the library is 2764 samples, and using the proposed approach the amount is only 649 (including 564 samples for base models, and 85 samples for subsequent library models) as shown in Table VI. The total training time for all library models using standard MLP approach is 2 h and 10 min on Sun Ultra 1 Workstation for such an illustrative library example. Using the proposed approach, the total training time is only 12 min.

B. *Example 2—Library of Microstrip Models*

In this example, a library of neural models for N-conductor lossless microstrip lines is developed, $N = 1, 2, 3, \cdots, 5$, i.e., a library of five models as shown in Fig. 6. Fig. 7 shows the details of a typical microstrip line from the library with the physical/geometrical parameters of the model defined. In

TABLE VII
MICROSTRIP LIBRARY MODELS

Library model index n	Model name	Neural model inputs X^n	Neural model outputs Y^n	No. of times each base model (B_1, B_2, B_3, B_4) is used $N_{Bj}^{\,n}$ x B_j
$n=1$	1 condcutor microstrip model	$w\ h\ \varepsilon_r$	L_{11}, C_{11}	1 x B_1, 1 x B_3
$n=2$	2 conductor microstrip model	$w\ s\ h\ \varepsilon_r$	$L_{11}, L_{12}, L_{22}, C_{11}, C_{12}, C_{22}$	1 x B_1, 1 x B_2 1 x B_3, 1 x B_4
$n=3$	3 conductor microstrip model	$w\ s_1\ s_2\ h\ \varepsilon_r$	$L_{11}, L_{12}, L_{13}, L_{22}, L_{23}, L_{33}, C_{11}, C_{12}, C_{13}, C_{22}, C_{23}, C_{33}$	1 x B_1, 3 x B_2 1 x B_3, 3 x B_4
$n=4$	4 conductor microstrip model	$w\ s_1\ s_2\ s_3\ h\ \varepsilon_r$	$L_{11}, L_{12}, L_{13}, L_{14}, L_{22}, L_{23}, L_{24}, L_{33}, L_{34}, L_{44}, C_{11}, C_{12}, C_{13}, C_{14}, C_{22}, C_{23}, C_{24}, C_{33}, C_{34}, C_{44}$	1 x B_1, 6 x B_2 1 x B_3, 6 x B_4
$n=5$	5 conductor microstrip model	$w\ s_1\ s_2\ s_3\ s_4\ h\ \varepsilon_r$	$L_{11}, L_{12}, L_{13}, L_{14}, L_{15}, L_{22}, L_{23}, L_{24}, L_{25}, L_{33}, L_{34}, L_{35}, L_{44}, L_{45}, L_{55}, C_{11}, C_{12}, C_{13}, C_{14}, C_{15}, C_{22}, C_{23}, C_{24}, C_{25}, C_{33}, C_{34}, C_{35}, C_{44}, C_{45}, C_{55}$	1 x B_1, 10 x B_2 1 x B_3, 10 x B_4

TABLE VIII
BASE MODELS FOR MICROSTRIP LIBRARY

Base model index j	Base model symbol B_j	Base model inputs $X_B^{\,j}$	Base model Outputs $Y_B^{\,j}$	Base Model structure (KBNN)	Model accuracy (average error)
1	B_1	$w\ h\ \varepsilon_r$	self inductance L	b2z3	0.16%
2	B_2	$w\ s\ h\ \varepsilon_r$	mutual inductance L_m	b2z3	0.13%
3	B_3	$w\ h\ \varepsilon_r$	self capacitance C	b1z2	0.18%
4	B_4	$w\ s\ h\ \varepsilon_r$	mutual capacitance C_m	b4z6	0.31%

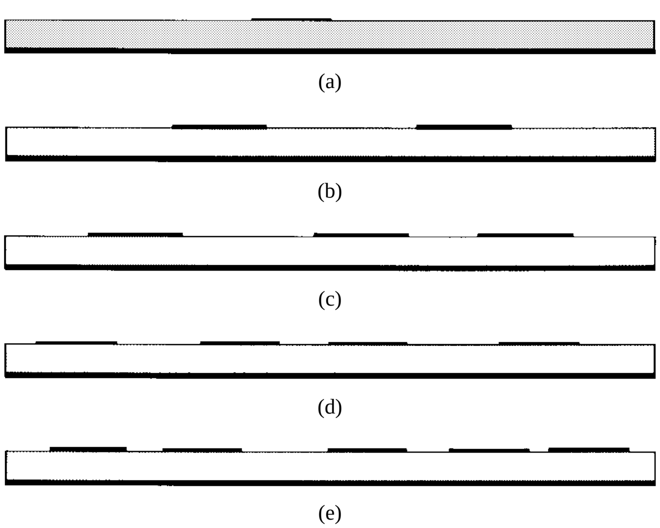

Fig. 6. The microstrip library. The nth model in the library represents an n-conductor coupled microstrip model. (a) $n = 1$, (b) $n = 2$, (c) $n = 3$, (d) $n = 4$, (e) $n = 5$.

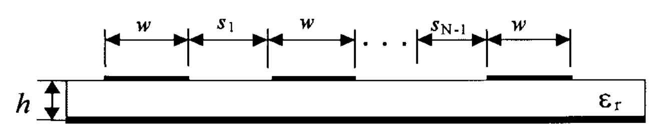

Fig. 7. Details of a typical N-conductor microstrip component showing the physical and geometrical parameters.

this library, we model the self and mutual inductance and capacitance as neural model outputs. All conductors have equal width, which is a reasonable assumption in many situations of signal integrity analysis and design of VLSI interconnects. The notations of parameters and parameter ranges of library neural models are defined similarly as those in Table I.

Table VII shows the detailed list of input and output parameters of each model in the microstrip library. Training and test data again were obtained using LINPAR [26] simulator which is based on the method of moments.

1) Base Model Selections: In this library, the most important basic characteristics of all models can be the re-

TABLE IX
LOW-LEVEL MODULES AND STRUCTURAL KNOWLEDGE HUBS FOR THREE-CONDUCTOR MICROSTRIP, I.E., LIBRARY MODEL $n = 3$

Low level modules L_i^n	Inputs to module/Knowledge Hub $u_i^n = U_i^n(X^n)$	Index function $\phi^n(i) = j$	Base model Used $B_{\phi(i)}$
L_1^3	$u_1^3 = U_1^3(X^3) = [w\ h\ \varepsilon_r]$	$\phi^3(1) = 1$	B_1
L_2^3	$u_2^3 = U_2^3(X^3) = [w\ s_1\ h\ \varepsilon_r]$	$\phi^3(2) = 2$	B_2
L_3^3	$u_3^3 = U_3^3(X^3) = [w\ s_2\ h\ \varepsilon_r]$	$\phi^3(3) = 2$	B_2
L_4^3	$u_4^3 = U_4^3(X^3) = [w\ s_1+s_2\ h\ \varepsilon_r]$	$\phi^3(4) = 2$	B_2
L_5^3	$u_5^3 = U_5^3(X^3) = [w\ h\ \varepsilon_r]$	$\phi^3(5) = 3$	B_3
L_6^3	$u_6^3 = U_6^3(X^3) = [w\ s_1\ h\ \varepsilon_r]$	$\phi^3(6) = 4$	B_4
L_7^3	$u_7^3 = U_7^3(X^3) = [w\ s_2\ h\ \varepsilon_r]$	$\phi^3(7) = 4$	B_4
L_8^3	$u_8^3 = U_8^3(X^3) = [w\ s_1+s_2\ h\ \varepsilon_r]$	$\phi^3(8) = 4$	B_4

TABLE X
MODEL ACCURACY COMPARISON (AVERAGE ERROR ON TEST DATA) BETWEEN STANDARD MLP AND THE PROPOSED MODEL FOR THREE-CONDUCTOR MICROSTRIP COMPONENT, I.E., LIBRARY MODEL $n = 3$

No. of training samples	MLP (5-25-12)	MLP (5-30-12)	MLP (5-35-12)	Proposed model
15	6.16%	6.63%	7.10%	0.42%
25	4.01%	4.57%	5.28%	0.40%
50	1.34%	2.33%	2.87%	0.38%
100	1.45%	1.67%	1.93%	0.38%
300	0.53%	0.43%	0.42%	0.35%

lationship between electrical parameters of self-inductance and capacitance of a conductor (and mutual inductance and capacitance between two conductors) and the microstrip physical/geometrical parameters. Four base models, $\boldsymbol{B}_1, \boldsymbol{B}_2, \boldsymbol{B}_3$, and $\boldsymbol{B}_4$ are created to represent these characteristics, respectively. The inputs and outputs of the base models are defined in Table VIII.

There exist empirical formulas for these characteristics in [25], approximating the relation between the self and mutual inductance and capacitance for single or dual microstrip lines. The base neural models are constructed incorporating such functional knowledge through a KBNN [1] structure combining the empirical information with the learning power of neural networks. The base models $\boldsymbol{B}_1, \boldsymbol{B}_2, \boldsymbol{B}_3$, and $\boldsymbol{B}_4$ are trained to an average testing accuracy of 0.16, 0.13, 0.18, and 0.31%, respectively, as shown in Table VIII. Linear transformation is used as the form of space mapping between $\boldsymbol{X}_B^j$ (of Table VIII) and $\boldsymbol{X}^n$ (of Table VII). Subspace mapping is used between $\boldsymbol{Y}_B^j$ (of Table VIII) and $\boldsymbol{Y}^n$ (of Table VII). The number of times base models are reused in each library model is shown in Table VII.

2) Example of Library Model $(n = 1)$: For $n = 1$, the library model is constructed simply by putting base models $\boldsymbol{B}_1$ and $\boldsymbol{B}_3$ together without any further training. Therefore, $\boldsymbol{L}_1^n = \boldsymbol{B}_1, \boldsymbol{L}_2^n = \boldsymbol{B}_3$. $\boldsymbol{H}^n$ relays from the $\boldsymbol{Z}^n$ space to the $\boldsymbol{Y}^n$ space.

3) Example of Library Model $(n = 3)$: For library model $n = 3$, we reuse the base models as the lower level neural modules following Fig. 2. There are eight low-level modules; four are for inductance prediction and four for capacitance prediction. The knowledge hubs for the model are defined in Table IX. The high-level neural module $\boldsymbol{H}^3$ is realized by a nonfully connected two-layer perceptron with eight inputs (i.e., preliminary inductance and capacitance prediction from low level) and 12 outputs (i.e., final inductance and capacitance of the overall library model). This example takes advantage of the modular neural network feature such that the overall library model is a linear combination with gating functions connecting four inductance (four capacitance) predictions from low-level modules to six inductance (six capacitance) outputs at the high level. Only a small amount of training data (15 samples) is needed to train this three-conductor microstrip model since the preliminary relationships of the model have already been captured in the base models. However, with the conventional MLP neural model (25, 30, 35 hidden neurons), 300 samples are needed to achieve a model of similar accuracy, shown in Table X.

Fig. 8 shows the tendency of model accuracy as the amount of available training data is reduced. The error for the proposed hierarchical model goes up only slowly, whereas the error for the standard MLP models grows very quickly as the amount of available training data is reduced.

4) Overall Library Accuracy and Development Cost—A Comparison: All library models, $n = 2, 3, 4, \cdots$ in the library, can be developed systematically in a similar way as model #3. The total amount of training data needed by standard MLP for the library is 1700 samples collected through electromagnetic simulations. The total amount of training data required by the proposed approach is only 550 (including 400 samples for base models and 150 samples for all subsequent library models) as shown in Table XI. Using standard MLP for each library model, the total training time for all library

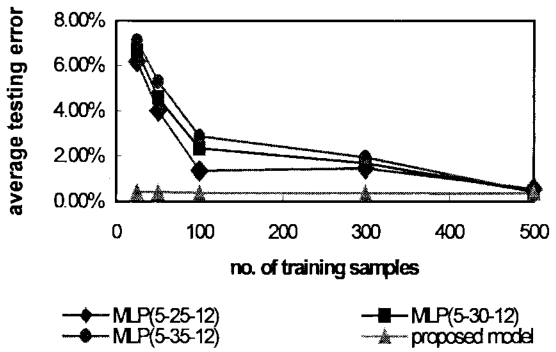

Fig. 8. Model accuracy comparison (average error on test data) between standard MLP and the proposed model for three-conductor microstrip model.

TABLE XI
COMPARISON OF NUMBER OF TRAINING SAMPLES NEEDED AND MODEL ACCURACY FOR MICROSTRIP LIBRARY WHEN DEVELOPED BY STANDARD MLP AND THE PROPOSED HIERARCHICAL NEURAL STRUCTURE, RESPECTIVELY

Library model index n	Microstrip model name	Number of training samples needed, (and corresponding model accuracy, training time) Standard MLP	Proposed model
Overhead for base models		0	100[1] (0.17%, 22 sec) 300[2] (0.22%, 14 min)
$n = 1$	1 conductor microstrip model	**100**, (0.45%, 2.5 min)	**0**, (0.17%, 0 sec)
$n = 2$	2 conductor microstrip model	**300**, (0.42%, 14.8 min)	**10**, (0.37%, 2 sec)
$n = 3$	3 conductor microstrip model	**300**, (0.42%, 73.2 min)	**15**, (0.42%, 4.8 sec)
$n = 4$	4 conductor microstrip model	**500**, (0.45%, 495 min)	**50**, (0.56%, 35 sec)
$n = 5$	5 conductor microstrip model	**500**, (0.37%, 416 min)	**75**, (0.51%, 4.1 min)
	microstrip library	Total = **1700** (16.7 hr)	Total = **550** (19.2 min)

[1]: base models $\boldsymbol{B}_1$ and $\boldsymbol{B}_3$ training
[2]: base model $\boldsymbol{B}_2$ and $\boldsymbol{B}_4$ training

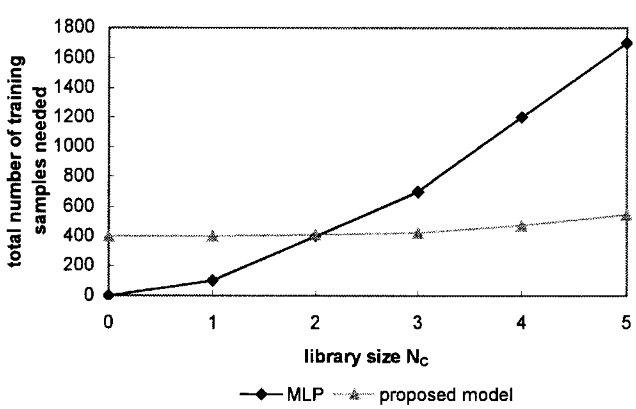

Fig. 9. The total amount of training data required for developing neural model library of microstrip lines versus the total number of models in the library. The overhead data of 400 required for the proposed approach due to base model training is represented by the nonzero value when $Nc = 0$. But the incremental amount of data needed for training each new model in the library is very small under the proposed approach. As the total number of models in the library increases, the total amount of training data required by the proposed approach becomes substantially less than that required by the standard MLP approach.

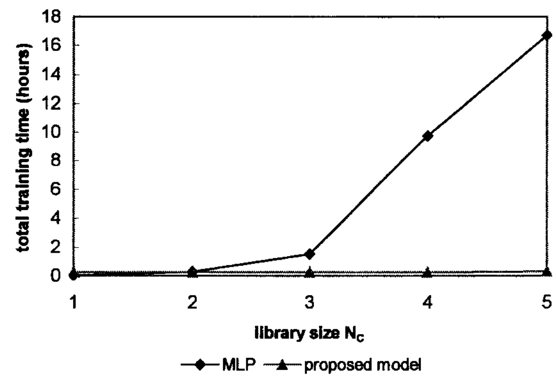

Fig. 10. The total training time for developing neural model library of microstrip lines versus the total number of models in the library. The overhead training time of 14 min for the proposed approach due to base model training is represented by the nonzero value at $Nc = 0$. But the incremental training time for adding each new model to the library is very small under the proposed approach. As the total number of models in the library increases, the total training time required by the proposed approach becomes substantially less than that of the standard MLP approach.

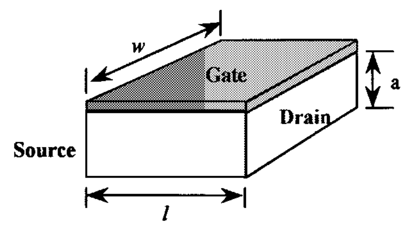

Fig. 11. Physics-based intrinsic MESFET device model following [27].

TABLE XII
EFFECTIVE RANGES OF NEURAL MODEL INPUT PARAMETERS FOR MESFET LIBRARY

Parameters	Notation	Range
gate length	l	0.35 – 0.80 μm
gate width	w	1 mm
channel thickness	a	0.28 – 0.42 μm
frequency	f	0.5 – 25 GHz
gate bias voltage	V_G	-5 – 0 V
drain bias voltage	V_D	0.5 – 4.5 V

models is 16.7 h on Ultra SparcStation, and using the proposed approach the total training time is only 19.2 min. Fig. 9 shows the tendency of amount of required training data versus the size of library. Fig. 10 shows the tendency of total training time required versus the size of the library.

C. Example 3: Library of MESFET Models

The drive for first-pass-success in designing active microwave circuits leads to the need of physics-based transistor device models which give more accurate predictions of device behavior than empirical or equivalent models. However, such physics-based models are too slow when used repetitively in circuit design. Neural models, trained from physics-based FET data, can be used to instantly predict physics-level device behavior for repetitive use during simulation and optimization [2]. Here we demonstrate the proposed hierarchical approach for a set of FET device models. For this specific example, we assume that the library consists of bias dependent S-

TABLE XIII
BASE MODELS FOR MESFET LIBRARY

Base model index j	Base model symbol B_j	Base model inputs $X_B{}^j$	Base model outputs $Y_B{}^j$	Base model structure (MLP)	Model accuracy (average error)
1	B_1	$a\ f\ V_G V_D$	S_{11} of $l = 0.4\mu m$	4-60-2	0.17%
2	B_2	$a\ f\ V_G V_D$	S_{12} of $l = 0.4\mu m$	4-60-2	0.20%
3	B_3	$a\ f\ V_G V_D$	S_{21} of $l = 0.4\mu m$	4-60-2	0.21%
4	B_4	$a\ f\ V_G V_D$	S_{22} of $l = 0.4\mu m$	4-80-2	0.18%
5	B_5	$a\ f\ V_G V_D$	S_{11} of $l = 0.8\mu m$	4-80-2	0.28%
6	B_6	$a\ f\ V_G V_D$	S_{12} of $l = 0.8\mu m$	4-80-2	0.38%
7	B_7	$a\ f\ V_G V_D$	S_{21} of $l = 0.8\mu m$	4-80-2	0.36%
8	B_8	$a\ f\ V_G V_D$	S_{22} of $l = 0.8\mu m$	4-80-2	0.29%

parameter models for MESFET's with different gate length values. A typical MESFET model in the library represents the intrinsic FET structure following Khatibzadeh and Trew [27], as shown in Fig. 11. The library contains ten models, $n = 1, 2, 3, 4, 5, 6, 7, 8, 9, 10$, and each model corresponds to a FET with a fixed gate length of 0.35, 0.4, 0.45, 0.5, 0.55, 0.6, 0.65, 0.7, 0.75, 0.8 μm, respectively. The library neural models are trained to predict the scattering parameters from physical and electrical parameters of the device. $\boldsymbol{Y}^n$ includes real and imaginary parts of S_{11}, S_{12}, S_{21}, and S_{22}. $\boldsymbol{X}^n$ includes frequency (f), channel thickness (a), gate bias voltage (V_G), and drain bias voltage (V_D). Training and test data were obtained by using OSA90[1] with Khatibzadeh and Trew models [27]. In this library, all transistors have assumed gate width of 1 mm. The model parameters and their ranges are shown in Table XII.

1) Base Model Selections: In this library, the relationships between the real and imaginary parts of the scattering parameters, namely S_{11}, S_{12}, S_{21}, and S_{22}, and model input parameters f, V_D, V_G, and a, are taken as the common characteristics required for all transistor models. To represent these common characteristics, eight base models, $\boldsymbol{B}_1, \boldsymbol{B}_2, \cdots$, and $\boldsymbol{B}_8$ are defined corresponding to four scattering parameters of two MESFET's, one with small gate length ($l = 0.40\,\mu$m) and another with large gate length ($l = 0.80\,\mu$m) as shown in Table XIII. Same-space mapping is used between $\boldsymbol{X}_B^j$ (of Table XIII) and $\boldsymbol{X}^n$, the inputs to both base models being the same as those for the other transistor models in the library. Subspace mapping is used between $\boldsymbol{Y}_B^j$ (of Table XIII) and $\boldsymbol{Y}^n$. The outputs of the base models are the real and imaginary parts of individual S-parameters of the transistor. In this example, we demonstrate that conventional MLP structure can also be used to construct the base models, with testing accuracy shown in Table XIII.

2) Example of Library Model ($n = 2$): For $n = 2$, the library model is constructed by four base models, $\boldsymbol{B}_1, \boldsymbol{B}_2, \boldsymbol{B}_3$, and $\boldsymbol{B}_4$, directly, without any further training, i.e., $\boldsymbol{L}_1^2 = \boldsymbol{B}_1, \boldsymbol{L}_2^2 = \boldsymbol{B}_2, \boldsymbol{L}_3^2 = \boldsymbol{B}_4, \boldsymbol{L}_4^2 = \boldsymbol{B}_4, \boldsymbol{H}^2(\boldsymbol{Z}^2) = \boldsymbol{Z}^2$.

[1] *OSA90 Version 3.0,* Optimization Systems Associations Inc., Dundas, Ont., Canada L9H 5E7, now HP EEsof, Santa Rosa, CA 95403.

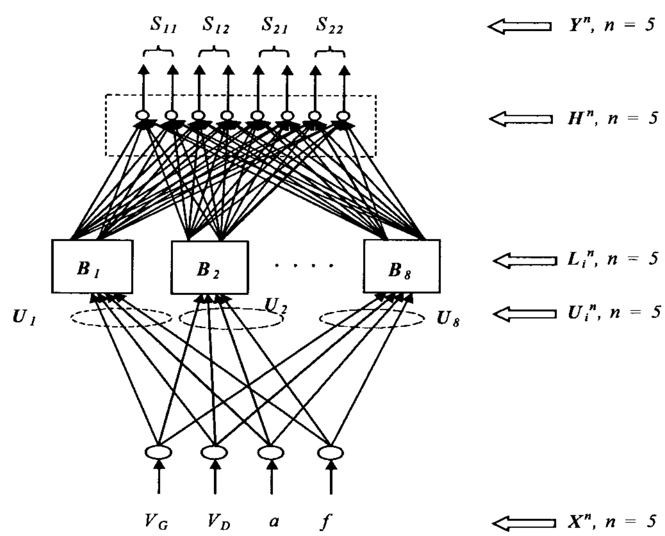

Fig. 12. The hierarchical neural model for FET library model #5, i.e., $n = 5$.

3) Example of Library Model ($n = 5$): For library model $n = 5$, the input and output definition of the model is the same as that of the base models. The difference is that the gate length is equal to 0.55 μm. The overall model structure is shown in Fig. 12. There are eight low-level modules, i.e., $N_L^5 = 8$, and $\boldsymbol{L}_1^5 = \boldsymbol{B}_1, \boldsymbol{L}_2^5 = \boldsymbol{B}_2, \cdots, \boldsymbol{L}_8^5 = \boldsymbol{B}_8$. Base models are used in the low-level neural modules to predict the S-parameter pattern for different model inputs. Since model input space is exactly the same as that of base models, knowledge hubs simply perform relay operations, i.e., $\boldsymbol{u}_i^5 = \boldsymbol{U}_i^5(\boldsymbol{X}^5) = \boldsymbol{X}^5$, $i = 1, 2, \cdots, 8$. The high-level neural module $\boldsymbol{H}^5$ is a two-layer perceptron with 16 inputs and eight outputs (real and imaginary parts of S_{11}, S_{22}, S_{12}, and S_{21}). Out of the 16 inputs, eight inputs correspond to the predictions from base models $\boldsymbol{B}_1, \boldsymbol{B}_2, \boldsymbol{B}_3$, and $\boldsymbol{B}_4$, while the other eight inputs correspond to the predictions from base models $\boldsymbol{B}_5, \boldsymbol{B}_6, \boldsymbol{B}_7$, and $\boldsymbol{B}_8$. This example takes advantage of the modular network concept without any gating function. Table XIV and Fig. 13 show the comparison of model accuracy when this transistor is modeled by standard MLP (with 60, 80, and 100 hidden neurons) and the proposed model.

TABLE XIV
MODEL ACCURACY COMPARISON (AVERAGE ERROR ON TEST DATA) BETWEEN STANDARD MLP AND THE PROPOSED MODEL FOR LIBRARY MODEL, $n = 5$, OF MESFET LIBRARY

No. of training samples	MLP (4-60-8)	MLP (4-80-8)	MLP (4-100-8)	Proposed model
25	13.97%	15.15%	14.78%	2.14%
50	4.51%	4.30%	4.97%	0.99%
100	2.29%	2.25%	2.57%	0.87%
150	1.71%	1.57%	1.62%	0.82%
200	1.46%	1.35%	1.37%	0.79%
300	0.96%	0.86%	0.94%	0.74%

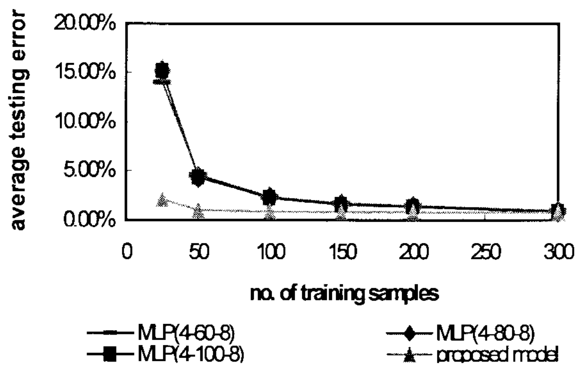

Fig. 13. Model accuracy comparison (average error on test data) between standard MLP and the proposed model for the MESFET library model, $n = 5$, whose gate length equals 0.55 μm.

TABLE XV
COMPARISON OF NUMBER OF TRAINING SAMPLES NEEDED AND TRAINING TIME USED FOR MESFET LIBRARY WHEN DEVELOPED BY STANDARD MLP AND THE PROPOSED NEURAL NETWORK STRUCTURE, RESPECTIVELY

Library model index n	MESFET model	Number of training samples needed, (and corresponding model accuracy, training time)	
		Standard MLP	Proposed model
Overhead for base models		0	300[1], (0.23%, 4.22hrs) 300[2], (0.33%, 4.54hrs)
$n = 1$	MESFET ($l = 0.35\ \mu m$)	300, (0.81%, 2.52 hrs)	50, (0.70%, 1.4 min)
$n = 2$	MESFET ($l = 0.4\ \mu m$)	300, (0.88%, 2.64 hrs)	0, (0.23%, 0 min)
$n = 3$	MESFET ($l = 0.45\ \mu m$)	300, (0.86%, 2.48 hrs)	50, (0.66%, 1.4 min.)
$n = 4$	MESFET ($l = 0.5\ \mu m$)	300, (0.88%, 2.32 hrs)	50, (0.71%, 1.4 min.)
$n = 5$	MESFET ($l = 0.55\ \mu m$)	300, (0.86%, 2.66 hrs)	50, (0.99%, 1.4 min.)
$n = 6$	MESFET ($l = 0.6\ \mu m$)	300, (0.89%, 2.18 hrs)	50, (1.22%, 1.4 min)
$n = 7$	MESFET ($l = 0.65\ \mu m$)	300, (0.82%, 2.45 hrs)	50, (1.08%, 1.4 min)
$n = 8$	MESFET ($l = 0.7\ \mu m$)	300, (0.87%, 2.58 hrs)	50, (1.05%, 1.4 min)
$n = 9$	MESFET ($l = 0.75\ \mu m$)	300, (0.79%, 2.50 hrs)	50, (0.77%, 1.4 min)
$n = 10$	MESFET ($l = 0.8\ \mu m$)	300, (0.88%, 2.78 hrs)	0, (0.33%, 0 min)
	MESFET Library	Total = 3000,(25.11hrs)	Total = 1000,(8.95hrs)

[1]: base models B_1, B_2, B_3, and B_4 training
[2]: base models B_5, B_6, B_7, and B_8 training

4) Overall Library Accuracy and Development Cost—Comparison: Model $n = 10$ can be developed similarly as model $n = 2$. All other library models, $n = 1, 3, 4, 6, 7, 8,$ and 9, can be developed easily in a similar fashion as library model #5. Using the proposed library approach for each library model, the training time and training data required are much less as compared to the standard MLP approach as shown in Table XV.

IV. CONCLUSIONS

A new problem, i.e., library of microwave neural model development, is addressed. A new hierarchical neural model approach is developed exploiting the inherent base relations between library models and incorporating both functional and structural knowledge. This approach can be applied to any microwave neural model library development in which basic electrical/microwave characteristics common to the library exist. The efficiency of the proposed approach increases when the library size increases, i.e., when a small set of base models can be extracted to represent basic information of a large number of library models. A significant cost reduction of neural model library development has been achieved, due to faster training and reduced need for data generation.

REFERENCES

[1] F. Wang and Q. J. Zhang, "Knowledge based neural models for microwave design," *IEEE Trans. Microwave Theory Tech.*, vol. 45, pp. 2333–2343, 1997.
[2] A. H. Zaabab, Q. J. Zhang, and M. S. Nakhla, "Neural network modeling approach to circuit optimization and statistical design," *IEEE Trans. Microwave Theory Tech.*, vol. 43, pp. 1349–1358, 1995.
[3] P. M. Watson and K. C. Gupta, "EM-ANN models for microstrip vias and interconnects in dataset circuits," *IEEE Trans. Microwave Theory Tech.*, vol. 44, pp. 2495–2503, 1996.
[4] ——, "Design and optimization of CPW circuits using EM-ANN models for CPW components," *IEEE Trans. Microwave Theory Tech.*, vol. 45, pp. 2515–2523, 1997.
[5] 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, 1997.
[6] M. Vai, S. Wu, B. Li, and S. Prasad, "Creating neural network based microwave circuit models for analysis and synthesis," in *Proc. Asia Pacific Microwave Conf.*, Hong Kong, Dec. 1997, pp. 853–856.
[7] M. Vai and S. Prasad, "Microwave circuit analysis and design by a massively distributed computing network," *IEEE Trans. Microwave Theory Tech.*, vol. 43, pp. 1087–1094, 1995.
[8] Q. J. Zhang and M. S. Nakhla, "Signal integrity analysis and optimization of VLSI interconnects using neural network models," in *IEEE Int. Symp. Circuits Systems*, London, England, 1994, pp. 459–462.
[9] G. Kothapalli, "Artificial neural networks as aids in circuit design," *Microelectron. J.*, vol. 26, pp. 569–678, 1995.
[10] 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.
[11] Q. J. Zhang, F. Wang, and M. S. Nakhla, "Optimization of high-speed VLSI interconnects: A review," *Int. J. Microwave and Millimeter-Wave CAD*, vol. 7, pp. 83–107, 1997.
[12] T. Horng, C. Wang, and N. G. Alexopoulos, "Microstrip circuit design using neural networks," in *IEEE Int. Microwave Symp. Dig.*, Atlanta, GA, 1993, pp. 413–416.
[13] 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.
[14] M. D. Baker, C. D. Himmel, and G. S. May, "In-situ prediction of reactive ion etch endpoint using neural networks," *IEEE Trans. Comp., Packaging, Manufact. Technol. A*, vol. 18, pp. 478–483, Sept. 1995.

[15] M. Vai and S. Prasad, “Automatic impedance matching with a neural network,” *IEEE Microwave Guided Wave Lett.,* vol. 3, pp. 353–354, 1993.
[16] A. J. Sharkey, “On combining artificial neural nets,” *Connection Sci.,* vol. 8, pp. 299–314, 1996.
[17] S. Hashem and B. Schmeiser, “Improving model accuracy using optimal linear combinations of trained neural networks,” *IEEE Trans. Neural Networks*, vol. 6, pp. 792–794, 1995.
[18] S. Hashem, “Algorithms for optimal linear combinations of neural networks,” in *Proc. IEEE Int. Conf. Neural Networks,* Houston, TX, June 1997, pp. 242–247.
[19] W. J. Jansen, M. Diepenhorst, J. A. G. Nijhuis, and L. Spaanenburg, “Assembling engineering knowledge in a modular multi-layer perceptron neural network,” in *Proc. IEEE Int. Conf. Neural Networks,* Houston, TX, June 1997, pp. 232–237.
[20] S. Haykin, *Neural Networks.* New York, NY: IEEE Press, 1994.
[21] G. G. Lendaris, A. Rest, and T. R. Misley, “Improving ANN generalization using a priori knowledge to pre-structure ANN’s,” in *Proc. IEEE Int. Conf. Neural Networks,* Houston, TX, June 1997, pp. 248–253.
[22] G. Auda, M. Kamel, and H. Raafat, “A new neural network structure with cooperative modules,” in *Proc. IEEE Int. Conf. Neural Networks,* Orlando, FL, June 1994, pp. 1301–1306.
[23] L. C. Wang, N. M. Nasrabadi, and S. Der, “Asymptotical analysis of a modular neural network,” in *Proc. IEEE Int. Conf. Neural Networks,* Houston, TX, June 1997, pp. 1019–1022.
[24] J. W. Bandler, R. M. Biernacki, S. H. Chen, P. A. Grobelny, and R. H. Hemmers, “Space mapping technique for electromagnetic optimization,” *IEEE Trans. Microwave Theory Tech.,* vol. 42, pp. 2536–2544, 1994.
[25] C. S. Walker, *Capacitance, Inductance and Crosstalk Analysis.* Boston, MA: Artech, 1990.
[26] A. Djordjevic, R. F. Harrington, T. Sarkar, and M. Bazdar, *Matrix Parameters for Multiconductor Transmission Lines: Software and User’s Manual.* Boston, MA: Artech, 1989.
[27] M. A. Khatibzadeh and R. J. Trew, “A large-signal, analytical model for the GaAs MESFET,” *IEEE Trans. Microwave Theory Tech.,* vol. 36, pp. 231–239, 1998.
[28]

Fang Wang (S’94) received the B.Eng. degree from Xi’an Jiaotong University, Xi’an, China, in 1993 and the M.Eng. degree from Carleton University, Ottawa, Canada, in 1995, both in electrical engineering She is currently working toward the Ph.D. degree at the Department of Electronics at Carleton University, Ottawa, Ont., Canada.

Her research interests include neural networks, modeling, and their applications in CAD for electronics circuits.

Ms. Wang is a recipient of a 1997–1999 Postgraduate Scholarship from the Natural Sciences and Engineering Research Council of Canada (NSERC). She is also a two-time recipient of the Ontario Graduate Scholarship Award in 1996 and 1997, respectively. She has received the Travel Fellowship twice from the IEEE International Conference on Neural Networks in 1997 and 1998, respectively.

Vijaya Kumar Devabhaktuni (S’97) received the B.Eng. degree in electrical and electronics and the M.Sc. degree in physics, both from the Birla Institute of Technology and Science, Pilani, India, in 1996. He is currently working toward the Ph.D. degree at Carleton University, Ottawa, Ont., Canada.

His research interests include neural networks, microwave device and circuit modeling, and computer-aided design of VLSI and telecommunication circuits.

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 from McMaster University, Hamilton, Canada, in 1987, both in electrical engineering.

He was with the Institute of Systems Engineering, Tianjin University, Tianjin, China, from 1982 to 1983. He was a Research Engineer with Optimization Systems Associates Inc., Dundas, Ont., Canada, from 1988 to 1990. During 1989–1990, he was also an Assistant Professor (part-time) of Electrical and Computer Engineering at McMaster University. He joined the Department of Electronics, Carleton University, Ottawa, Canada, in 1990, where he is presently an Associate Professor. His research interests are in the area of computer-aided design of high-speed/high-frequency circuits, with emphasis on neural networks, optimization, modeling, simulation, and statistical design of microwave circuits and high-speed VLSI interconnections. He has more than 100 publications in international journals and conferences. He is a contributor to *Analog Methods for Computer-Aided Analysis and Diagnosis,* (Marcel Dekker, 1988), a Coeditor of *Modeling and Simulation of High-Speed VLSI Interconnects,* (Kluwer, 1994), and a Guest-Editor for *The International Journal of RF and Microwave Computer-Aided Engineering,* Special Issue on Applications of Artificial Neural Networks to RF and Microwave Design.