Short Papers

Reverse Modeling of Microwave Circuits with Bidirectional Neural Network Models

Mankuan M. Vai, Shuichi Wu, Bin Li, and Sheila Prasad

***Abstract*— Neural networks have been developed into an alternative modeling approach for the microwave circuit-design process. In this paper, we describe a neural network-based microwave circuit-design approach that implements the solution-searching optimization routine by a modified neural network learning process. Both the development of a microwave circuit model and the searching of a design solution can thus take advantage of a hardware neural network processor, which is significantly faster than a software simulation. In addition, a systematic simulation-based approach to convert conventional circuit models into neural network models for this design process will be described. The development of a heterojunction bipolar transistor (HBT) amplifier model and its applications are demonstrated.**

***Index Terms*— Neural network applications, modeling, optimization methods.**

I. Introduction

Neural networks have been demonstrated to be a robust modeling approach to predict the behavior of microwave circuits [1]. In comparison with various statistical methods and curve-fitting approaches for predicting system behavior, the neural network approach features a learning process which fine tunes neural network parameters to interrelate the variables being modeled.

In the microwave circuit-design process, a solution is generated and its corresponding behavior is predicted by a circuit model—either traditional or neural network. The difference between the desired and simulated behavior is used to guide the generation of a new solution. This solution-searching optimization routine iterates until no further improvement can be achieved.

Neural network computation is a distributed process in which all the neurons operate in parallel. A software implementation of a neural network is sluggish since the neurons have to be updated sequentially. Only hardware implementations, which explore the parallelism of neural network computing, can fully realize its potential [2], [3]. In prior efforts at applying neural network models in microwave design, while the neural network training and modeling operations could be accelerated by a neural network processor, the solution-searching optimization (e.g., a gradient method) remained as a software routine external to the neural network model.

This paper describes a microwave circuit-design approach that replaces the sequential solution-searching optimization routine by a modified neural network learning process. This approach allows all three main components of a neural network-based design process (i.e., the training, modeling, and solution searching) to be implemented in a hardware neural network co-processor.

After a brief review of the background of the modeling of microwave circuits with neural networks, the new approach will be described. The application of this approach to the design of heterojunction bipolar transistor (HBT) microwave amplifiers will be demonstrated.

Manuscript received June 18, 1997.

The authors are with the Department of Electrical and Computer Engineering, Northeastern University, Boston, MA 02115 USA.

Publisher Item Identifier S 0018-9480(98)07237-8.

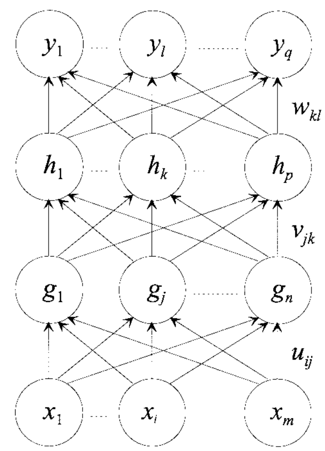


Fig. 1. A feedforward neutral network with two hidden layers.

II. Background

A series of design solutions have to be evaluated in a microwave circuit-design process. In order to predict the result of employing certain design parameters, the circuit has to be simulated. It is beyond the scope of this paper to describe circuit-simulation techniques, but suffice it to say that circuit simulation is always difficult and time consuming.

The contribution of neural network models is the replacement of the circuit model by a fast black-box model. A brief review of neural networks used in microwave circuit modeling is provided in Section II-A.

A. Neural Network Modeling

Unlike most modeling and simulation methods, the complexity of a neural network does not increase exponentially with the number of components in the circuit being simulated. This renders the neural network modeling approach very efficient.

Due to the availability of a powerful training algorithm called back propagation, multilayer feedforward neural networks are most popular for modeling applications. A multilayer neural network with four layers (one input layer, two hidden layers, and one output layer) used for modeling purposes is shown in Fig. 1. Referring to the notation in Fig. 1, $\boldsymbol{X} = (x_1, \cdots, x_i, \cdots, x_m)$ is the input vector, $\boldsymbol{G} = (g_1, \cdots, g_j, \cdots, g_n)$, $\boldsymbol{H} = (h_1, \cdots, h_k, \cdots, h_p)$, and $\boldsymbol{Y} = (y_1, \cdots, y_l, \cdots, y_q)$ are the outputs of the first hidden layer, second hidden layer, and output layer, respectively), u_{ij} is the weight between the ith input and the jth neuron in the first hidden layer, v_{jk} is the weight between the jth neuron in the first hidden layer and the kth neuron in the second hidden layer, and w_{kl} is the weight between the kth neuron in the second hidden layer and the lth neuron in the output layer. The output of the neural network can be computed as

$$y_l = \frac{1}{1 + e^{-\gamma_l}} \tag{1}$$

where γ_l is the weighted total input to the output neuron l, which is defined as

$$\gamma_l = \sum_{k=1}^{p} h_k w_{kl} \tag{2}$$

and p is the number of neurons in the second hidden layer. Similarly, the output of the second hidden layer $\boldsymbol{H}$ can be expressed as a function of the output of the first hidden layer $\boldsymbol{G}$, which can, in turn, be expressed as a function of the input vector $\boldsymbol{X}$.

The back propagation training algorithm aims to adjust the weights of a feedforward neural network in order to minimize the sum-squared error of the network, which is defined as

$$E = \sum_{m=1}^{S} \left[\frac{1}{2} \sum_{l=1}^{q} (d_{ml} - y_{ml})^2 \right] \tag{3}$$

where S is the number of training data points, q is the number of output variables, and $\boldsymbol{d_m} = [d_{m1} \quad d_{m2} \cdots d_{mq}]$ and $\boldsymbol{y_m} = [y_{m1} \quad y_{m2}, \cdots, y_{mq}]$ are the mth desired and calculated output vectors, respectively. This is typically done by continually changing the values of the weights in the direction of steepest descent with respect to the error function E (for a detailed description of the learning algorithm see [4]).

B. Design Process

The function of a circuit model is to predict the circuit response corresponding to given design parameters. Since the objective of a design process is to determine design parameters that produce the desired outcome, it would be ideal if a circuit model can be used in a reverse direction to generate design parameters that will produce the desired response. For example, it has been proposed to create reverse neural network model training data by specifying design parameters for a given circuit response [5].

While the need for a reverse model is apparent, microwave circuits do not have reverse functions. Any attempt to create a reverse model for a microwave circuit unavoidably captures only a portion of the system relations. This is because, in general, a design problem does not have a unique solution. While it presents no problem in the development of a model that maps n sets of possible circuit parameters into the same response, a reverse model can only capture one of these n relations and, thus, must discard the other $(n-1)$ circuit parameters–response relations. The practice of microwave circuit design commonly requires a number of solutions to be generated for a given design target so that the one which is least sensitive to parameter deviations can be chosen for the purpose of a better yield rate. A reverse model that is forced to leave out all but one of the circuit parameters–response relations cannot support this design requirement.

III. Design by a Neural Network Learning Process

Instead of pursuing an explicit reverse model of the microwave circuit under design, a novel design approach in which the searching of a solution is performed with a modified neural network learning process is developed. This approach begins by training a neural network to model the circuit under design. As described in Section II-A, the weights of the neural network are adjusted at this stage to minimize its error function given by (3). The solution searching is then performed by applying a modified backpropagation learning rule to the trained network. An initial solution of design parameters is generated and the trained neural network model is used to predict the outcome of this solution. The difference between the desired outcome and the one corresponding to the current solution is calculated and back propagated through the layers in the neural network. Instead of adjusting the neural network weights, as originally done in the training of the neural network, the input variables are modified to minimize the error function defined in (3), while the weights are kept unchanged.

This is a very simple modification of the learning process because we can simply exchange the roles of weights and inputs in the backpropagation learning rule. This modified learning rule can be described as

$$x_i' = x_i - \eta \frac{\partial E}{\partial x_i} \tag{4}$$

and

$$\frac{\partial E}{\partial x_i} = \sum_{l=1}^{q} \sum_{k=1}^{p} \sum_{j=1}^{n} \Big[(y_l - d_l) y_l (1 - y_l) w_{kl} h_k (1 - h_k) \cdot v_{jk} g_j (1 - g_j) u_{ij} \Big] \tag{5}$$

where x_i and x_i' are the current and next input variables, respectively, and η is the learning rate. The other variables are as defined in Section II-A. It is evident that the operations described in (5) can be carried out in a distributed fashion. Each neuron can utilize values propagated back from the next layer to calculate its associated terms and, in turn, send the results to the previous layer. This feature allows the solution-searching routine to be implemented in a hardware neural network processor along with the training and modeling operations.

By comparing this approach with the reverse model proposed in [5], it is seen that both allow the solution searching to be performed in a neural network processor without the need of an external optimization routine. However, this approach has the advantage that a regular forward model is directly used and there is no need to explicitly define a reverse model. Since the forward model is used, all the relations between design parameters and outcomes are retained. The design process is not forced to make a predetermined selection among the relations.

Another significant property of this design approach is that multiple solutions, if they exist in the modeled system, can be found typically with different initial solutions. This allows a yield analysis to be performed on the solutions to determine a design that is least sensitive to parameter deviations.

IV. Implementation

Since the performance of this design approach depends on a trained neural network model, the accuracy of the forward training is critical. Generally, we would like to use the fewest training patterns that statistically represent the problem on hand. The aim of training the neural network model is to achieve a model that responds correctly to the design parameters that are used for training (memorization) and the ability to give good responses to design parameters that are similar, but not identical, to those used in training (generalization). Apparently, a neural network model trained with insufficient or inadequate data cannot be expected to function as a valid model. On the other hand, a training process can be overwhelmed if it is fed with training data indiscriminately.

Another problem with training-data preparation is in the source for the data. Traditionally, data collection is performed by measuring a real circuit set up under different conditions and is thus often time consuming and expensive. This type of data collection may even be impossible in many microwave applications.

A systematic way to prepare the training data and develop the neural network for the design process is developed. Instead of performing real circuit measurements, simulation techniques are used to generate training data with a simulator such as LIBRA[1] and

[1] HP EEsof, Westlake, CA.

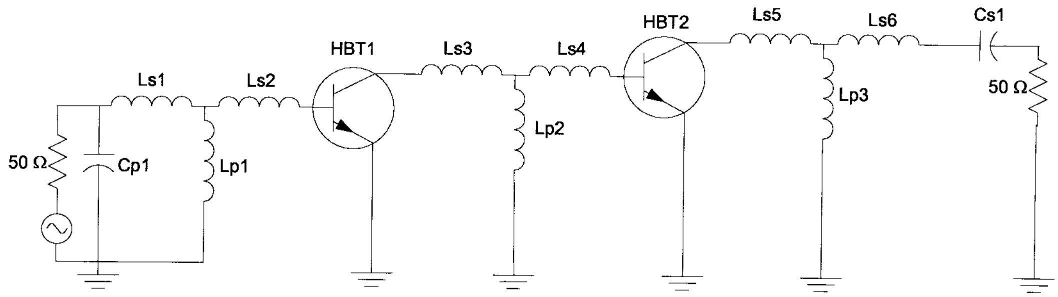
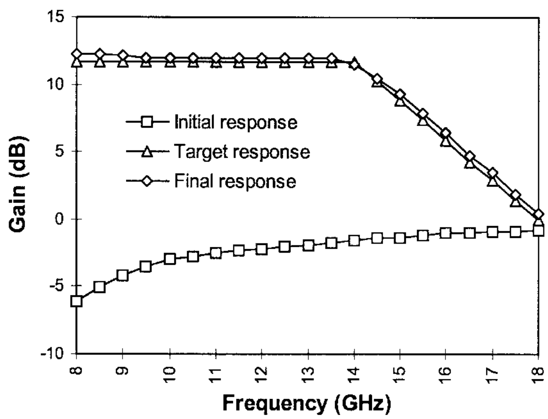
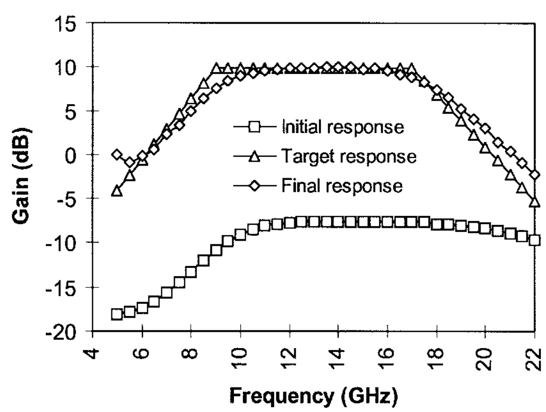


Fig. 2. An HBT microwave amplifier.



(a)



(b)

Fig. 3. The frequency responses of (a) a flat-band amplifier. (b) A bandpass fitler designed by this approach.

HSPICE.[2] A major portion (e.g., 80%) of these data are used to train the neural network, and the rest are used to verify the model. The goal is to achieve similar modeling errors for both the training and verification of data. Additional data can be provided to reduce the discrepancy between training and verification. This simulation-based approach of generating training data also has the advantage of providing a means to convert a conventional circuit model into a neural network model for design purposes.

V. Application Example

The capability of this new design approach has been demonstrated with the design of HBT amplifiers. The circuit diagram of a two-stage HBT amplifier is shown in Fig. 2. A neural network trained to model this amplifier was used to synthesize circuits for different design goals. The design parameters are the nine inductors ($Ls1$–$Ls6$, $Lp1$–$Lp3$) and two capacitors ($Cp1$ and $Cs1$). The circuit response to be modeled is the frequency response of this amplifier. A commercial microwave computer-aided design (CAD) tool LIBRA[1] was used to generate 5000 sets of design parameters versus amplifier frequency-response records. 4000 sets were used to train a neural network model of the HBT amplifier and the remaining 1000 sets were used to verify the accuracy of the resulting model.

The application of the amplifier neural network model was tested on a variety of design cases. Fig. 3 shows the frequency responses of two circuits designed with the neural network model. It can be seen in both design cases that acceptable results were produced by this design approach. The design parameters and their corresponding frequency responses were verified with LIBRA.

The solution searching (i.e., the modified learning) process is very fast since the number of adjustable variables is significantly reduced from that of forward training. It takes several hours to complete the training of this amplifier neural network model on a typical workstation, but the design result can be found in a few seconds ($<$15 s). The fast design time allows a number of solutions to be generated from different initial conditions for the same design goal. A sensitivity analysis process can then be applied to select higher yield solutions that are less sensitive to design parameter deviations.

VI. Summary

In summary, a design process is developed so that neural networks assume an active role in microwave circuit design. Instead of being used passively as fast black-box circuit models, as is conventionally done, neural networks can be used to create a design by a modified neural network learning process. A systematic simulation-based approach is developed to create neural network models from conventional circuit models for the purpose of designing. The performance of this design approach is demonstrated by the creation of a neural network model for a two-stage HBT amplifier and the use of the resulting model to design a few circuits with different frequency responses.

[2] Advant!, Fremont, CA.

References

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

[2] C. Lehmann, M. Viredaz, and F. Blayo, "A generic systolic building block for neural networks with on-chip learning," *IEEE Trans. Neural Networks*, vol. 4, pp. 400–407, Mar. 1993.

[3] C. E. Cox and W. E. Blanz, "Ganglion—A fast hardware implementation of a connectionist classifier," in *Proc. Custom Integrated Circuits Conf.*, San Diego, CA, May 1991, pp. 6.5.1–6.5.4.

[4] R. P. Lippmann, "An introduction to computing with neural nets," *IEEE ASSP Mag.*, vol. 4, pp. 4–22, Apr. 1987.

[5] R. Ferguson and D. J. Roulston, "Neural networks for accelerating the transistor design process," in *Proc. Artificial Neural Networks Eng. Conf.*, vol. 6, St. Louis, MO, Nov. 10–13, 1996, pp. 995–1000.