CIRCUIT MODELING USING EXTRAPOLATED NEURAL NETWORKS

VIKAS PALIWAL AND QI-JUN ZHANG
Dept. of Electronics, Carleton University,
1125 Colonel by Drive, Ottawa (ON), Canada, K1S5B6
Email: {vpaliwal, qjz}@doe.carleton.ca

A new extrapolation technique for trained neural models for RF and microwave circuits is presented enhancing their performance outside the trained domain. A neural model developed after training it with data in a particular range of inputs shows poor and arbitrary performance outside the training range. The proposed technique exploits the information available on the boundary of the trained domain to extrapolate outside it. The extrapolated neural models show a marked improvement over the original ones outside the trained domain.

1 Introduction

Artificial neural networks have emerged as a useful tool to address the demand for fast and accurate models for RF and microwave crcuits [1] [2]. A neural network, after being trained for a particular range of input data, is computationally very fast compared to EM simulators like *Sonnet-Lite* [5] and physics based simulators like *Minimos-NT* [6]. However, the accuracy of a neural model of a circuit component decreases very fast outside its training region due to saturation of the activation functions used in the hidden layer of neural network structure. So, even though a well-trained neural model yields accurate results in its trained region, it loses most of the information about actual component behavior outside the trained region.

This paper presents a novel extrapolation technique for improving the performance of a trained neural model outside its training range for cases where boundary characteristics are useful in predicting the component behaviour outside the training range. It utilizes the concept of gradient-based extrapolation using multi-dimensional Taylor series expansion from a suitable point on the training boundary. Examples of embedded resistor modeling using EC-SSE-NN approach [3] and GaAs-FET modeling demonstrate that the extrapolated neural models developed using the proposed technique are more useful than the models developed using existing techniques over an extended range of inputs.

2 Proposed Extrapolation Technique for Trained Neural Models

A neural model of a circuit or a circuit component characterizes a set of output behaviours for some set of inputs like physical and geometrical parameters. More precisely, a neural model evaluates an m–dimensional output vector, $\mathbf{y} = [y_1, y_2, \ldots, y_j, \ldots, y_m]^T$ for n–dimensional input vector, $\mathbf{x} = [x_1, x_2, \ldots, x_i, \ldots, x_n]^T$. Training of a neural network is done using a set of input-output data in a desired range of inputs, say $\{l_1 \leq x_1 \leq u_1, l_2 \leq x_2 \leq u_2, \ldots, l_i \leq x_i \leq u_i, \ldots, l_n \leq x_n \leq u_n\}$, by optimizing the weight parameters associated with the neurons to match the actual behaviour in the best possible manner. A trained neural model that is an optimized linear combination of shifted and scaled forms of activation functions like sigmoid function [1] tends to saturate fast and carries little information about actual behaviour beyond the trained range and fails to provide even rough tendencies. In this paper, we present an extrapolation technique that preserves the behavior on the training boundary of the neural model in the form of first and second order derivatives and uses it to improve the performance of a trained model over an extended region.

A *training box* can be defined as the set of $2n$ possible surfaces of types $x_i = l_i$ or $x_i = u_i$ in the n-dimensional space that enclose the finite n-dimensional space corresponding to the region for which the model is trained. Many powerful training algorithms like backpropagation are available for achieving any desired level of accuracy in a finite *training box*. Assuming that the neural network has been trained well in the *training box* and the developed model faithfully follows the actual behaviour, for points lying inside the *training box*, we simply evaluate the trained model and for a point, $\mathbf{p} = [p_1, p_2, \ldots, p_n]^T$, lying outside, we extrapolate from the point of intersection of the line joining $\mathbf{p}$ and the centre of the *training box*, $\mathbf{c}(\mathbf{c} = \frac{1}{2}\{[l_1, l_2, \ldots, l_n]^T + [u_1, u_2, \ldots, u_n]^T\})$, with the *training box*. Let **t** be this point which can be easily found using space geometry.

A gradient-based extrapolation approach can then be used to extrapolate the behaviour of the neural model outside the *training box*. Since neural networks evaluate known activation functions like sigmoid function, their analytical derivatives at point **t** on the training boundary can be easily calculated. For each of the output, y_i ,with a simple modification of replacing the weight parameters with the inputs, the gradient vector and the Hessian matrix can be accurately evaluated at any point in the trained domain in a similar manner as done during the training process.

$$\mathbf{g}_i(\mathbf{x}) = [\frac{\partial y_i}{\partial x_1}, \frac{\partial y_i}{\partial x_2}, \ldots, \frac{\partial y_i}{\partial x_n}]^T \tag{1}$$

$$\mathbf{H}_i(\mathbf{x}) = \begin{bmatrix} \frac{\partial^2 y_i}{\partial x_1^2} & \frac{\partial^2 y_i}{\partial x_2 \partial x_1} & \cdots & \frac{\partial^2 y_i}{\partial x_n \partial x_1} \\ \frac{\partial^2 y_i}{\partial x_1 \partial x_2} & \frac{\partial^2 y_i}{\partial x_2^2} & \cdots & \frac{\partial^2 y_i}{\partial x_n \partial x_2} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 y_i}{\partial x_1 \partial x_n} & \frac{\partial^2 y_i}{\partial x_2 \partial x_n} & \cdots & \frac{\partial^2 y_i}{\partial x_n^2} \end{bmatrix} \tag{2}$$

Then each of the outputs can be evaluated for first and second orders of extrapolation using Equation (3a) and (3b), respectively.

$$y_i(\mathbf{p}) = y_i(\mathbf{t}) + (\mathbf{p} - \mathbf{t})^T \cdot \mathbf{g}_i(\mathbf{t}) \tag{3a}$$

$$y_i(\mathbf{p}) =$$

$$y_i(\mathbf{t}) + (\mathbf{p} - \mathbf{t})^T \cdot \mathbf{g}_i(\mathbf{t}) + \frac{1}{2}(\mathbf{p} - \mathbf{t})^T \cdot \mathbf{H}_i(\mathbf{t}) \cdot (\mathbf{p} - \mathbf{t}) \tag{3b}$$

To illustrate this approach, a simple relation of type, $y_1 = \sqrt{(x_1 - 1)^2 + (x_2 - 1)^2}$, is considered.The contour plot for this relation is shown in Fig. 1(a). A neural network with inputs, $\mathbf{x} = [x_1, x_2]^T$, and output, $\mathbf{y} = [y_1]^T$, is trained in the region $\{-2 \leq x_1 \leq 2, -2 \leq x_2 \leq 2\}$. The contour plot using neural network output is shown in Fig. 1(b). The *training box*, as defined earlier, is also shown. Significant improvement is observed using first and second orders of extraploation as shown in Fig. 1(c) and 1(d).

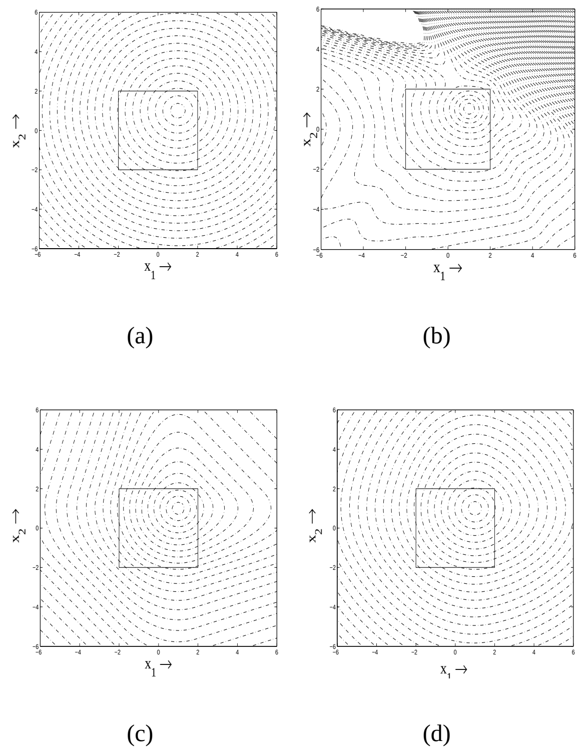


Figure 1: Contour Plots (*Training Box* is marked in each case) (a) Actual function (b) Output of neural network developed using existing technique (c) First-order extrapolated neural network output (d) Second-order extrapolated neural netwok output.

3 Examples

The proposed approach can be effectively used to improve trained neural models for circuit components. After an accurate neural model for a particular range of parameters is developed using the existing techniques available in softwares like *NeuroModeler* [7], extrapolation can be used to improve the accuracy of the model outside the trained range as shown by the following examples.

3.1 GaAs FET Modeling

To illustrate this approach, an example of a GaAs FET model is presented here. Data generated from a model of GaAs FET available in *Minimos-NT* [6] is used to train a neural network with inputs, $\mathbf{x} =$

$[V_{gs}, V_{ds}]$ and output $\mathbf{y} = [I_{ds}]$ in the range $\{-3\text{V} \leq V_{gs} \leq -2\text{V}, 0\text{V} \leq V_{ds} \leq 2\text{V}\}$. Fig. 2 shows the poor performance of the unextrapolated neural model and subsequent improvement after first and second order extrapolation using the proposed approach over the extended region, $\{-3\text{V} \leq V_{gs} \leq 0\text{V}, 0\text{V} \leq V_{ds} \leq 8\text{V}\}$.

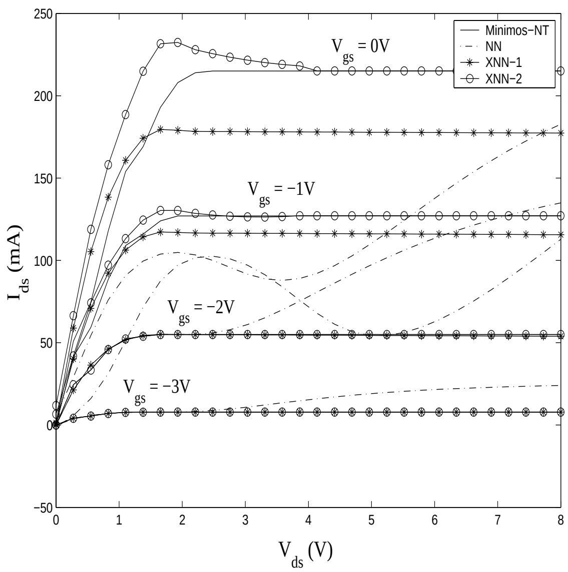
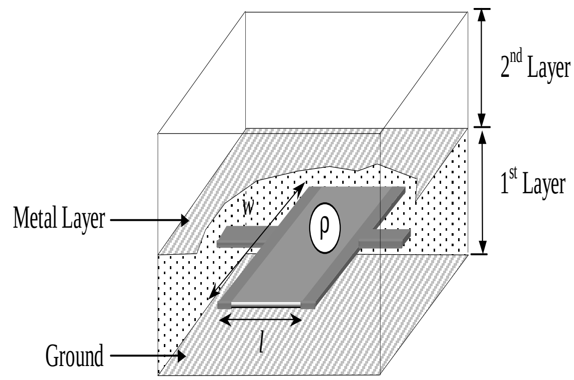
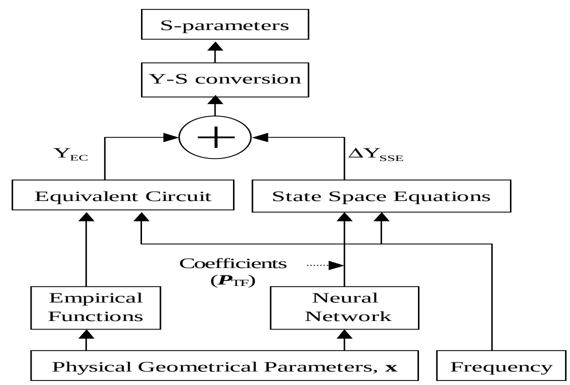


Figure 2: Comparison of extrapolated models of FET with original neural model (NN stands for original neural model, XNN-1 for first-order extrapolated neural model and XNN-2 stands for second-order extrapolated neural model)



(a)

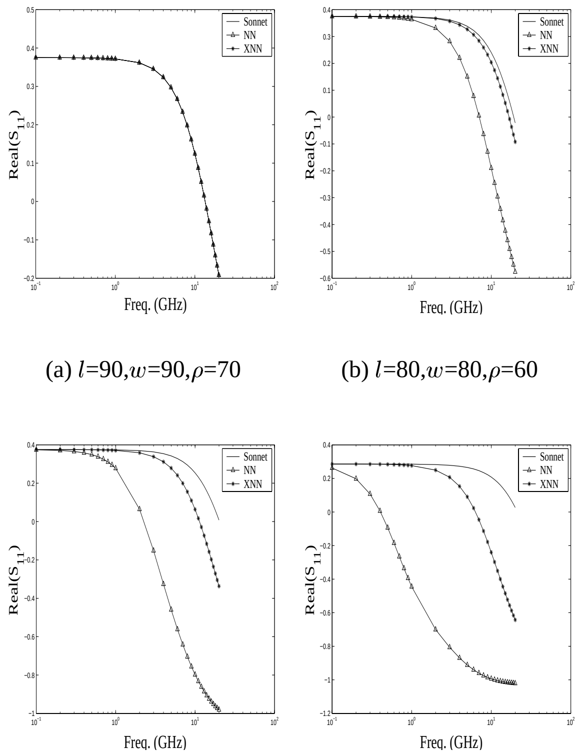


(b)

Figure 3: (a) Geometrical sructure of embedded resistor in multi-layer PCBs, (b)EC-SSE-NN structure for embedded resistor modeling

3.2 EC-SSE-NN Model for Embedded Resistor

EM based models of embedded resistors are important for high-speed PCB design. It has been shown that neural networks, in conjunction with equivalent circuits (EC) and state-space equations (SSE), can be used to develop models for the embedded resistor shown in Fig. 3(a) in the high-frequency range [3] [4]. The structure of an EC-SSE-NN model is shown in Fig. 3(b). In EC-SSE-NN structure, neural network is used to calculate the vector of coefficients of second-order state space equations, $\mathbf{p}_{tf}$, for input vector of physical and geometrical paramaters, $\mathbf{x} = [w, l, \rho]^T$, where w, l, ρ are width, length and surface resistivity of embedded resistor, respectively. A model trained with high level of accuracy (testing error = 1.22%) for the region $\{90\text{mil} \leq w \leq 120\text{mil}, 90\text{mil} \leq l \leq 120\text{mil}, 60\ \Omega/\text{sq} \leq \rho \leq 80\ \Omega/\text{sq}\}$ is taken and extrapolation technique is used to improve its performance beyond the trained region. Fig. 4 shows the improvement shown by the first-order extrapolated model as we progressively go outside the trained range. The average testing errors when compared with EM data generated using *Sonnet-Lite* [5] for the extrapolated and original model are compared for all the possible regions beyond the trained range of the neural model. For this purpose, *testing boxes* are constructed by expanding the input data ranges beyond the trained ranges so that they always contain the *training box*. For example, a *testing box* of volume $8V_T$ (V_T is the volume of *training box*) corresponds to doubling the range for each of the three inputs. The average testing errors for different volumes of *testing boxes* are calculated and the results are summarized in Table 1.



(a) l=90,w=90,ρ=70 (b) l=80,w=80,ρ=60

(c) l=70,w=70,ρ=50 (d) l=60,w=60,ρ=40

Figure 4: Comaprison of *Sonnet-Lite* simulated frequency response with unextrapolated and extrapolated neural models (l and w are in mils and ρ is in Ω/sq).

4 Conclusion

The proposed extrapolation technique is found to be useful in improving the performance of the trained neural models for EM circuits. It can be a useful tool to see the approximate behaviour of components outside the training domain of the neural models. Considering the generic approach followed in the proposed technique, the performance of any trained neural network can be improved in this fashion.

Table 1: Average percentage testing errors for embedded resistor model

Testing Box Volume	**Original Model**	**Extrapolated Model**
V_T	1.22	1.22
1.33 V_T	2.93	1.36
1.7 V_T	4.03	1.84
2.4 V_T	5.25	2.54
8 V_T	13.91	5.19
27 V_T	27.81	10.04

References

[1] Q. J. Zhang and K.C. Gupta, Neural Networks for RF and Microwave Design, Norwood, MA, 2000.

[2] P. M. Watson and K.C. Gupta, “EM-ANN models for microstrip vias and interconnects in multilayer circuits”, IEEE Trans. Microwave Theory Tech., vol. 44, pp.2495-2503, Dec. 1996.

[3] X. Ding, V. K. Devabhaktuni, M. C. E. Yagoub, B. Chattaraj, M. Deo, J. J. Xu and Q. J. Zhang, “Neural Network Approaches to Electromagnetic Based Modeling of Passive Components and Their Applications to High-Frequency and High-Speed Nonlinear Circuit Optmization”, accepted for IEEE Trans. Microwave Theory Tech.

[4] Q.J. Zhang, M.C.E. Yagoub, X. Ding, D. Goulette, R. Sheffield and H. Feyzbakhsh, “Fast and Accurate Modeling of Embedded Passives in Multi-Layer Printed Circuits”, IEEE Electronic Components and Technology Conference, pp. 700-703, 2002.

[5] *Sonnet-Lite v. 6.0b*, Sonnet Software Inc., Liverpool, NY.

[6] *Minimos-NT*, Institute of MicroElectronics, Technical University, Vienna, Austria.

[7] *NeuroModeler v. 1.2*, Q. J. Zhang, Dept. of Electronics, Carleton University, Ottawa, Canada.