Application of Neural Networks in the Development of Nonlinear Error Modeling and Test Point Prediction[1]

Xiaolian Han*, Gerard N. Stenbakken*, Fernando J. Von Zuben**, Hans Engler***
*National Institute of Standards and Technology, Gaithersburg, MD 20899, USA
**Department of Computer Engineering and Industrial Automation
State University of Campinas, Sao Paulo-SP-Brazil
*** Department of Mathematics, Georgetown University,
Washington, DC 20057, USA

Abstract *- This paper explores a neural network approach for empirical nonlinear error modeling. For systems that have a significant amount of nonlinearity, nonlinear error models require fewer parameters compared to linear models and require fewer test points to achieve the same prediction accuracy. A neural network with a five-layer structure is investigated. The test point error predictions from nonlinear modeling are compared with the results of linear modeling for an artificial nonlinear model, a circuit with nonlinearity, and an instrument with suspected nonlinearity. The nonlinear modeling shows more improvement when the data set contains more nonlinearity.*

I. INTRODUCTION

For many electronic devices and instruments, over 50 percent of the product's cost can be attributed to testing costs. Thus, to perform the tests efficiently and economically is very important. To meet the requirement of formulating economical test plans, model-based approaches with optimized testing tools have been developed by using an optimized small set of selected test points to predict the overall performance of the tested products [1]. This is based on the fact that the behavior of many devices is governed by a relatively small set of underlying variables, which consequently determine the results of a large number of measurements [2].

The linear approach [1,3] optimizes the testing process by developing a linear model for the type of devices being tested. This is done by analyzing data taken on a large number of similar devices. The singular value decomposition (SVD) is used to determine the principal components of the data set, which become the model for this device type. A minimal subset of test points is determined from this model.

The test data of interest are the deviations of the measurements from their nominal values for each test point. If a full set of test data T on a device is given by m measurements (n test points) and this is done for r independent devices, it is assumed that T can be related to an underlying model A and parameters X by

$$T_{m\times r} = A_{m\times n} * X_{n\times r} + \varepsilon , \qquad (1)$$

where the model matrix $A_{m\times n}$ relates the n parameters X to the m measured deviations for each of the r devices. The data is corrupted by random measurement error ε, which adds noise to each measurement. The model is specific to the device type and incorporates information that depends on the device design, its components, its production process, etc. $X_{n\times r}$ consists of parameters that vary for each individual device [3].

A model is linear if for any device j all of the parameters X_{ij} $(i=1,2,\ldots n)$ are independent. The model is nonlinear if the parameters can be chosen such that for any device j one or more of the parameters X_{ij} can be expressed as a nonlinear function of one or more of the other parameters. This type of nonlinear model can be used to approximate most types of nonlinear models. To determine the behavior of any device the linear model requires n independent parameters or data from at least n selected test points. This is true even if the underlying model is nonlinear. However, when the model is

[1] Electricity Division, Electronics and Electrical Engineering Laboratory, Technology Administration, U.S. Department of Commerce, Official Contribution of National Institute of Standards and Technology; not subject to copyright in the U.S.A.

0-7803-5890-2/00/$10.00 © 2000 IEEE

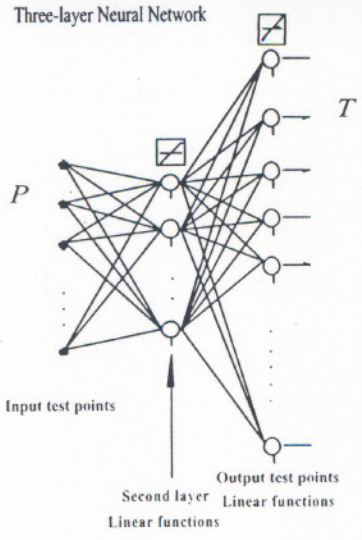

Fig. 1. The structure of three-layer NN with linear functions in second layer—linear modeling.

nonlinear and the nonlinear relations are known, less than n parameters or data from less than n test points are sufficient to determine the device behavior. Thus, to realize the benefits of any nonlinear behavior requires a model that makes use of the nonlinear relations between parameters. A neural network (NN) of two layers, where the first layer is sigmoid and the second layer is linear, can be trained to approximate any function (with a finite number of discontinuities) arbitrarily well [4]. This nonlinear capability is applied in this paper to empirical error modeling.

Nonlinear modeling has the potential of giving more economical test plans than linear modeling and of applying to a broader range of devices. With the more economical test plans, the number of measured test points can be traded off for more accuracy in the error predictions.

The next section describes the relation between a three-layer NN model and linear models. Section III introduces the five-layer nonlinear NN used in this paper and applies it to an artificial model with known nonlinearity. Section IV compares the application of this model with the linear model for a circuit with nonlinearity and data from an instrument that may have nonlinearity. Finally, the conclusions from this study are summarized.

II. LINEAR MODELING WITH NN

The NN analog of the linear model given by (1) is the three-layer NN shown in fig.1. The k nodes or neurons of the first or input layer are the reduced measurement data P for a device. The outputs of the second layer nodes are the model parameters X for that device and the outputs of third layer nodes are the full set of m test point results T for the device. The lines between nodes are weights W that are multiplication factors. All the inputs to a node are summed, a threshold factor B is subtracted, and the result transformed by a function f. Thus, the node output values T for layer l are given by

$$T_t^{(l)} = f^{(l)}\left(\sum_{j=0}^{q-1} W_{tj}^{(l)} * T_j^{(l-1)} - B_t^{(l)}\right) \qquad (2)$$

where $f^{(l)}$ is the function for all layer l nodes, $T_j^{(l-1)}$ is the output of layer l-1 node j, $W_{tj}^{(l)}$ is the connecting weight from layer l-1 node j to layer l node t, $B_t^{(l)}$ is the threshold value of layer l node t, and q is the number of nodes at layer l-1. The index t varies over the nodes at layer l. For this NN the functions f at both the second and third layers are identity functions, i.e., $f(x) = x$.

In the analogy with the linear model, the weights between the second layer and the output layer are related to an estimate for the model matrix A. The weights between the first and second layer are the inverse of the reduced model that solves for the device parameters based on the reduced measurements at the selected test points.

Even though this NN is equivalent to the linear model, the methods used to solve it differ. Both methods use the modeling (training) data set T. The linear approach uses the SVD to determine the principal components, which estimate model A. The NN approach iteratively adjusts the weights W in the model to minimize the differences between the model output data and the training data set. The linear approach is more efficient but only produces linear models. The NN approach can be extended to nonlinear models as described in the next section.

III. NONLINEAR MODELING

For the purpose of nonlinear modeling a five-layer NN is used which has sigmoidal (hyperbolic tangent) functions in the second and fourth layers and linear functions in the third and fifth layers; thus, for $l = 3$ or 5, $f^{(l)}(x) = x$, and for $l = 2$ or 4, $f^{(l)}(x) = tanh(x)$. The structure of the five-layer NN is shown in fig. 2. The outputs of the third layer are the nonlinear principal component coefficients for the devices [5].

To test this approach an artificial nonlinear model was computed from random matrices. Devices with 120 test points and four parameters were simulated using a 120x4 model matrix A. A training matrix T was computed that

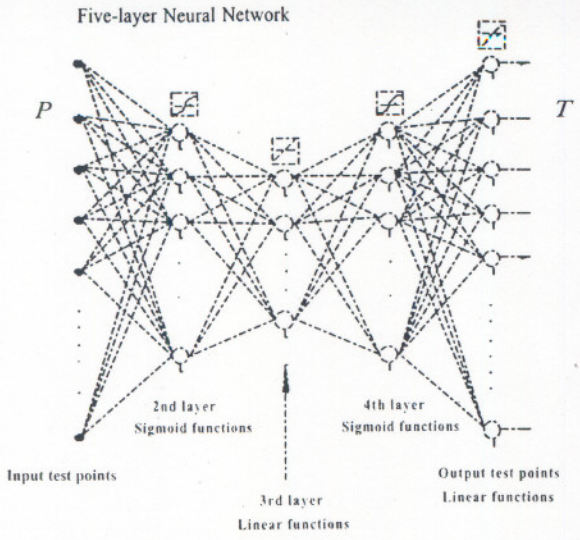

Fig. 2. Structure of five-layer NN for nonlinear modeling.

represents data taken on 300 such devices with a 4x300 matrix X of device parameters; $T = (A*X)/4$. The data was scaled by the factor four so that the average data is about unity. The entries of A and the first two rows of X were randomly sampled from a uniform distribution on the interval -1 to +1. The third and fourth rows of X are obtained from the first two rows by the nonlinear relations $X_{3j} = (X_{1j})^2$ and $X_{4j} = (X_{2j})^3$. Each device j is characterized by two parameters.

First, the five-layer NN was trained using the full training data set T with 120 input nodes and 120 output nodes to determine the model performance under the best of conditions. The training process used feed-forward, supervised learning [5] (fig. 2) for the nonlinear principal component analysis (NLPCA). The training iteration process is sped up with the use of a second order conjugate gradient learning algorithm based on nonlinear unconstrained optimization techniques [6]. The number of neurons in the nonlinear layers was chosen equal to the number of linear independent principal components in the training set, which can be computed from the SVD of T, four in this case. The number of linear center layer nodes estimates the number of nonlinear independent components of the data set. The training of the NN will not converge when the number of the linear center layer nodes is less than two, which is the number of nonlinear independent parameters in this case.

Figure 3 gives the comparison of the center layer output $y = T^{(3)}$ and the original parameters, x_1, x_2, after the model has converged. It is seen that the first parameter is reproduced quite well and that there is qualitative agreement for the second parameter. In general one

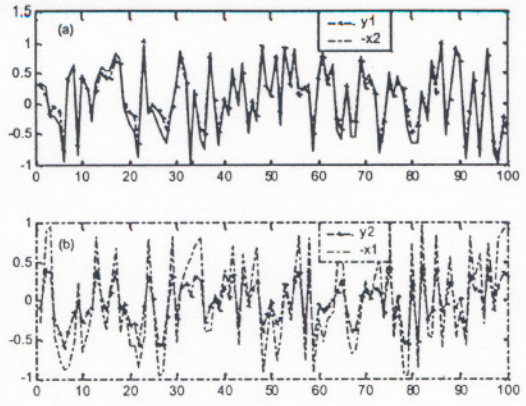

Fig. 3. Center layer output of the five-layer NN y_1,y_2 and the original parameters x_1,x_2 (first 100 of 300 shown).

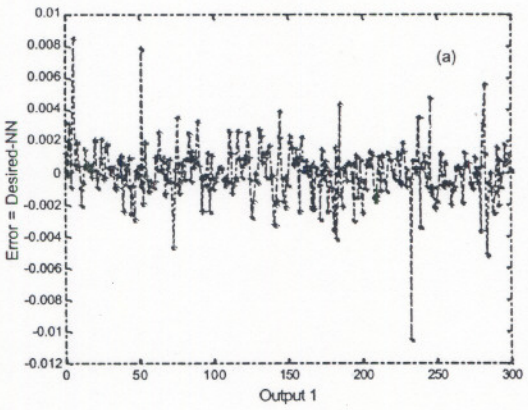

Fig. 4(a). Error of first output neuron (the difference between desired output and NN output).

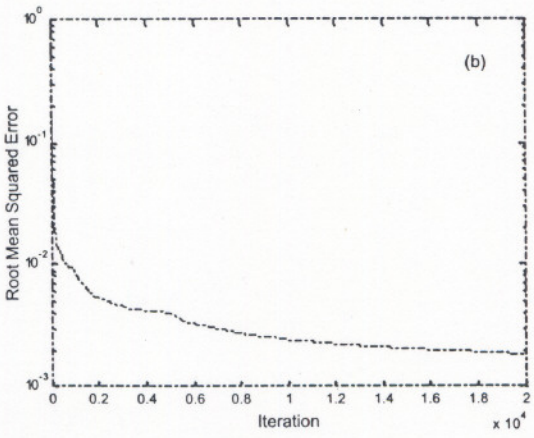

Fig. 4(b). Root mean squared error of the predicted values versus training iterations.

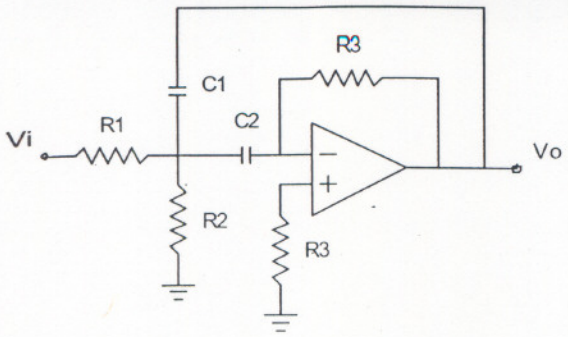

Fig. 5(a). The circuit of a bandpass filter with five parameters R_1, R_2, R_3, C_1, C_2.

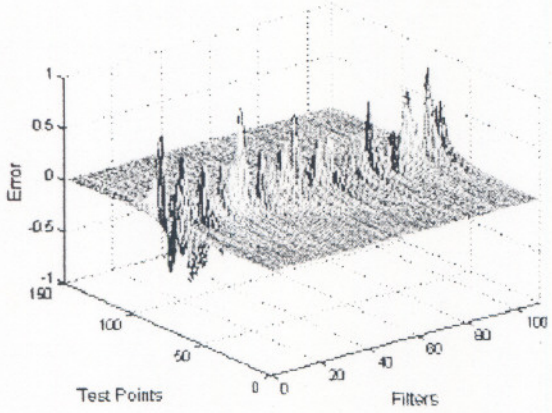

Fig. 5(b). Bandpass filter data set of 150 test points and 110 filters (80 filters are chosen as a model set, 30 are chosen as a validation set).

would expect the parameter estimates from the model, y_1 and y_2, to be a linear combination of the original x_1, and x_2.

Then the five-layer NN was modified to have only two input test points and 120 output test points. The two test points selected were the ones that most closely matched the fitted parameters, y_1 and y_2. The model was trained using the same techniques mentioned above. In this case, for each device, its entire 120 test points were being predicted by data from only two selected test points. Figure 4(a) shows the difference between the predicted and true results for an arbitrarily chosen test point for all 300 devices. The differences are quite small relative to the size of the signal being unity. The convergence behavior for this two input NN is shown in fig. 4(b). That figure shows how the root mean squared error from all 300 by 120 test points decreases during the training iterations. It is seen that the error is reduced by a factor of 20 during the first 2000 iteration steps. It then is halved in the next 5000 iterations and halved again in another 10,000 steps.

IV. NONLINEAR CIRCUITS

Model of Bandpass Filter

The use of NN to empirically model a nonlinear circuit is illustrated with a bandpass filter. The circuit schematic is shown in fig. 5(a). The filter has five passive components whose nominal values are 100 kΩ, 503Ω, and 20 kΩ for R_1, R_2, and R_3, and 1 μF for capacitors C_1, and C_2. The circuit response is a nonlinear function of these parameters as shown by the filter transfer function given as

$$T(s) = \frac{-\left(\frac{1}{R_1 C_1}\right)s}{s^2 + \left(\frac{C_1 + C_2}{C_1 C_2 R_3}\right)s + \left(\frac{R_1 + R_2}{C_1 C_2 R_1 R_2 R_3}\right)}. \qquad (3)$$

The data sets were calculated to represent measurements on a number of such filters with random variations of the nominal component values. No measurement error was considered. For each filter the test point data are calculated as the deviation from the filter's nominal response at 150 frequencies at equal log spacing from 10 Hz to 10 kHz. To explore the differences between mildly nonlinear and strongly nonlinear data, two sets were generated. The mildly nonlinear set was generated by randomly varying the five parameters by 5percent, and the strongly nonlinear set by varying them by 20 percent. For each case, a training set of measurement data on 80 filters was calculated, and the resultant models checked by an independent set of 30 filters. Figure 5(b) shows the data for the strongly nonlinear sets.

Both linear and nonlinear models were used to model the data sets. In each case the number of parameters was set to five. The linear model was used to select varying numbers of test points used for both model types. The five-layer NN structure shown in fig. 2 was used with the input layer the selected reduced test points k.

Figure 6 gives the root mean squared error of the NN results compared with the linear modeling results for the 30 validation filters. The number of selected test points is varied from 5 to 50 (i.e. the number of the first layer neurons for the NN). For the NN the number of the second and fourth layer neurons is always 30. This number was chosen to equal the number of principal components determined from an SVD of the training set data. The results in fig. 6 show a clear difference

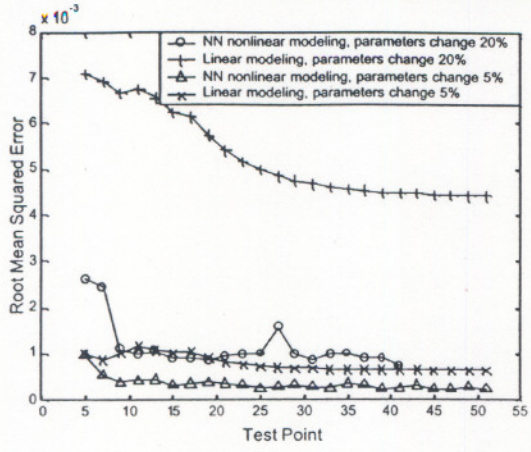

Fig. 6. Root mean squared error of the nonlinear NN modeling and linear modeling of the bandpass filter.

between the linear and nonlinear models. Note that when the data set contains more nonlinearity (parameters change 20 percent) the difference between the linear and nonlinear models is more significant.

Model of a Thermal Voltmeter

The five-layer NN nonlinear model has been applied to the same commercial-thermal-voltmeter data set that was modeled by the linear approach [8]. This data set appears to contain a large number of small parameters. The reason for the large number was thought to be the need to approximate nonlinear functions with a number of linear functions. Thus, it was hoped that applying a nonlinear model to this data set would reduce the number of parameters needed to describe the model. This in turn could lead to the need for fewer test points for the same prediction accuracy. The following describes some preliminary results in applying NN to modeling this data set.

Figure 7(a) shows both the NN and linear modeling results with 45 parameters. The NN modeling shows significantly better results when the number of test points is below 62. Beyond 62 test points the NN and linear modeling results are approximately equal.

The reason the NN modeling is slightly poorer than the linear modeling, when the number of test points is higher than 62, is that the NN modeling results here are not optimized. The NN results depend on a number of parameters, which require simultaneous optimization by means we do not currently know how to accomplish efficiently. These parameters include the number of third layer neurons (i.e., the number of the nonlinear parameters), the number of second and fourth layer

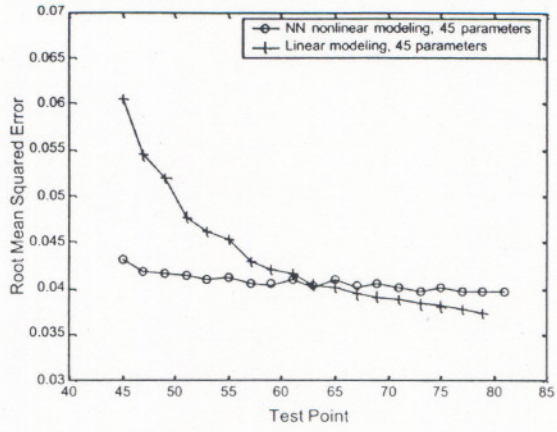

Fig. 7(a). Root mean squared error of the nonlinear NN modeling and linear modeling of a thermal voltmeter.

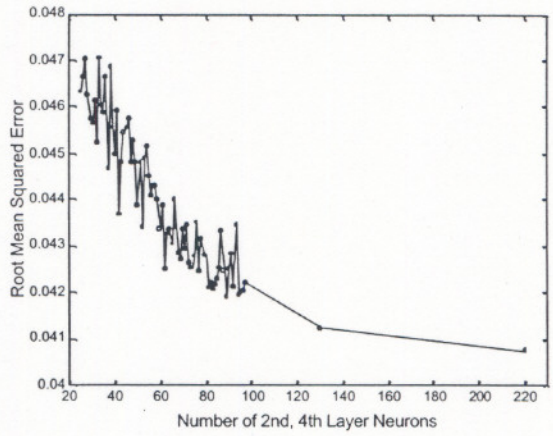

Fig. 7(b). Root mean squared error of the nonlinear NN modeling of a thermal voltmeter with the second and fourth layer changing from 25 to 220 (the number of test points is 80, number of center layer neurons is 25).

neurons, the number of NN training iterations, and the test point selection.

Effects of optimized parameters can be examined by varying one parameter at a time. For example, fig. 7(b) shows the change in the root mean squared errors with the number of second and fourth layer neurons. The number of input test points was held constant at 25. Since this data set appears to contain many small parameters, it requires a large number of second and fourth layer neurons. However, an increase in the number of these neurons makes the NN training process more time consuming. The results show that the gain in accuracy is only modest.

In this approach, test points for NN modeling have been selected with the same method as for linear modeling. A selection method that is better suited to NN models is not known.

V. CONCLUSIONS AND DISCUSSION

An approach for modeling nonlinear data sets has been developed that uses a five-layer NN. The results for a number of artificial and real circuit examples have been compared. When the data contains a significant amount of nonlinearity, the NN is able to develop a model that is more efficient than a linear model. Although the NN approach takes much more computer time to develop the model, this process is normally done offline relative to the testing process. Thus, if this process can produce a more efficient test plan the extra cost of developing the model may be worth while.

The process of selecting the number of parameters and making good test points selections for the linear modeling were previously developed. For the NN approach additional parameters must be selected. These include the number of nonlinear functions in the model, the number of nonlinear principal components, the number of training iterations, and the test points to select. A method for making a good selection for some of these parameters has been demonstrated. For example, the use of SVD to determine the number of nonlinear functions works very well. Also, the use of the same set of test points selected for linear models works well. We have not explored whether a better set can be determined.

So far, the selection of the number of nonlinear principal components has been determining by finding the smallest number of third layer neurons for which the NN still converges well. The optimum number of training iterations can be less than the largest for some models. That is, the overall error for a separate validation set versus iteration count will first decrease, reach a minimum, then start to climb. How to prevent this or to find the optimum number of iterations is being explored.

REFERENCES

[1]. G.N. Stenbakken and T.M. Souders, "Developing linear error models for analog devices", IEEE Trans. Instrum. Meas., vol. 43, no. 2, pp.157-163, 1994.

[2]. T.M. Souders and G.N. Stenbakken, "Cutting the high cost of testing", IEEE Spectrum, March 1991.

[3]. A.D. Koffman, T.M. Souders, G.N. Stenbakken, and H. Engler, "High-dimensional Empirical Linear Prediction (HELP) Toolbox" User's Guide, Version 2.2, NIST Technical Note 1428, May 1999.

[4]. Neural Network Toolbox—For use with MatLAB, The Math Works, Inc., p2-14, 1998.

[5]. D. Fotheringhame, R. Baddeley, "Nonlinear Principal Components Analysis of Neuronal Spike Train Data", Biological Cybernetics, 77: (4) pp283-288 Oct.1997.

[6]. L.N. de Castro, F.J. Von Zuben, " Optimized Training Techniques for Feed-forward Neural Networks", State University of Campinas, Sao Paulo-SP-Brazil, Technical Report, DCA-RT 03/98, July 1998.

[7]. F.J. Von Zuben, M.L. de Andrade Netto, " Unit-growing Learning Optimizing the Solvablility Condition for Model-free Regression", Proceedings of IEEE International Conference on Neural Networks, vol.2, pp.795, 27 Nov.-1 Dec.1995.

[8]. A.D. Koffman, T.M. Souders, G.N. Stenbakken, T.E. Lipe, and J.R. Kinard, "Empirical Linear Prediction Applied to a NIST Calibration Service," Proc. 1996 Natl. Conf. of Standards Laboratories (NCSL) Workshop and Symposium, Aug 25-29, 1996, Monterey, CA, pp. 207-212, Aug 1996.