AIAA JOURNAL
Vol. 35, No. 4, April 1997

Neural Network Approximator with Novel Learning Scheme for Design Optimization with Variable Complexity Data

Srinivas Kodiyalam*
Lockheed Martin Advanced Technology Center, Palo Alto, California 94304
and
Ram Gurumoorthy†
General Electric Corporate R&D Center, Schenectady, New York 12301

A novel learning scheme for training neural networks is proposed. The trained network is then used for function approximation during the numerical optimization process. The learning scheme trains the network with data of varying complexity, including data that have only zeroth-order information, and when the data include first-order information (gradients of responses with respect to the design parameters) it uses the combined information (response values and its gradients) for a better approximation. The learning scheme and its function approximation capability for design optimization are demonstrated on two realistic examples.

I. Introduction

THE application of neural networks as universal approximators has been the focus of a tremendous amount of activity in the past few years.[1–3] Neural networks have been used in design optimization for approximating and modeling responses of both memoryless systems and dynamic systems with memory. These neural networks primarily use function values (zeroth-order information) while learning the design response surfaces. The authors have demonstrated that a neural network with a modified backpropagation learning scheme, using design sensitivity or gradient information, in addition to function values, can provide for more accurate approximations of the response surface and quicker convergence of the learning process.[4]

Detailed design optimization of realistic structural systems, for example, aerospace and automotive systems, usually involves multiple disciplines and data of varying complexity. Not all of the physical discipline and manufacturing process solvers provide for both zeroth-order (response values) and first-order (response gradients) information for use with the optimization process. This paper addresses such variable complexity data problems encountered in design optimization of structural systems.

A novel learning scheme for training neural networks is proposed. This scheme trains the network with data of varying complexity, including data that have only zeroth-order information, and when the data include first-order information (gradients of responses with respect to the design parameters), it uses the combined information (response values and its gradients) for a better approximation. The network trained using the hybrid learning scheme then is used for function approximation during the numerical optimization process.

This hybrid learning scheme exploits the assets of the standard backpropagation learning scheme, such as robustness of approximation and block presentation of input data for improving convergence, when training the network with data that represent just zeroth-order information, while exploiting the advantages of the modified backpropagation learning scheme,[4] and when training the net for those data that in addition include the gradient information. The hybrid learning scheme facilitates the use of test/experimental data as part of network learning.

II. Network Architecture and Learning Scheme

This paper considers a multilayer feedforward network. For data with just the zeroth-order information, a multilayer feedforward network with standard backpropagation learning has been shown to be an efficient universal approximator.[1,5] There have been many additions to, or adaptations of, the basic backpropagation scheme, such as adaptive learning rate, learning with momentum,[1] improving convergence of the learning scheme, but this paper does not go into the details of these additions. The authors restrict themselves to a single hidden-layer feedforward network, as shown in Fig. 1. The restriction to a single hidden layer is not a limitation of the hybrid learning scheme and the new training scheme also can be used on networks with multiple hidden layers.

A. Hybrid Learning Scheme

To illustrate the philosophy of the proposed learning scheme, a net with a single neuron is considered. Figure 2 shows a single tansigmoid neuron net. The output of the net is given by

$$o = f(wu)$$

where

$$f(x) = \frac{2}{1 + e^{-2x}} - 1$$

An energy function P is defined as

$$P = 0.5(T - o)^2 = 0.5e^2$$

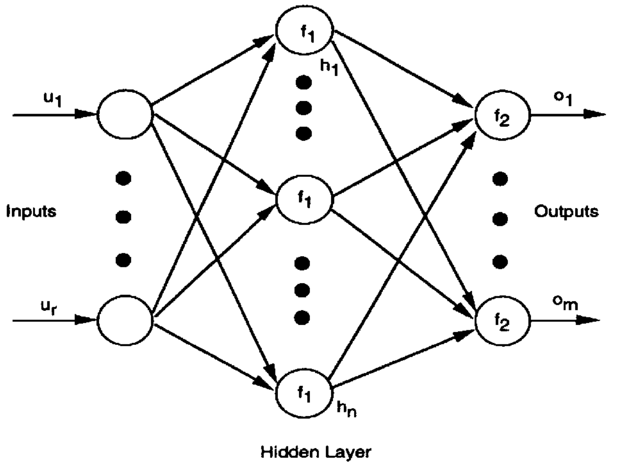

Fig. 1 Feedforward neural net with single hidden layer.

Received Feb. 28, 1996; presented as Paper 96-1339 at the AIAA/ASME/ASCE/AHS/ASC 37th Structures, Structural Dynamics, and Materials Conference, Salt Lake City, UT, April 15–17, 1996; revision received Jan. 8, 1997; accepted for publication Jan. 13, 1997.

*Senior Staff Scientist, Structures Laboratory; currently Director of Advanced Technology, Engineous Software Inc., Morrisville, NC 27560. Senior Member AIAA.

†Staff Engineer, Control Systems and Electronic Technologies Laboratory. Member AIAA.

where T is the target value of the output. The objective of the training scheme is to make the difference between the target output and the actual output to be zero. This can be achieved by maintaining a negative gradient of the energy function, as this drives P to zero, because P is always positive:

$$\mathrm{d}P = (T - o)\left(\frac{\partial e}{\partial w}\,\mathrm{d}w + \frac{\partial e}{\partial u}\,\mathrm{d}u\right)$$

$$= (T - o)\left\{-(1 - o^2)u\,\mathrm{d}w + \left[\frac{\partial T}{\partial u} - (1 - o^2)w\right]\mathrm{d}u\right\}$$

To make $\mathrm{d}P$ negative, that is, to have the training scheme reduce the difference between the target output and the actual output, we can use different weight update laws. Under the standard backpropagation learning scheme, the weight update is defined as

$$\mathrm{d}w = (T - o)(1 - o^2)u$$

Under the modified backpropagation learning scheme,[4] which uses the gradient of the response to the input parameters, the weight update is redefined as

$$\mathrm{d}w = (T - o)(1 - o^2)u + \left[-w + \frac{1}{1 - o^2}\frac{\partial T}{\partial u}\right]\frac{\mathrm{d}u}{u}$$

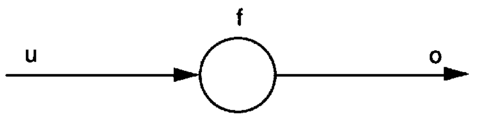

Fig. 2 Neural net with a single neuron.

The term inside the square bracket in the weight update law is the modification to the standard backpropagation. This modification has been shown to speed up the learning process convergence and also improve the accuracy of the approximation.[4]

In situations with varying complexity of data, let us assume that the data are categorized into two families: u^1 (data with zeroth- and first-order information) and u^0 (data with just zeroth-order information), T^1 and T^0 the corresponding target outputs, and o^1 and o^0 the corresponding neural network outputs. With the hybrid backpropagation learning scheme, the weight update is defined as

$$\mathrm{d}w = (T^0 - o^0)[1 - (o^0)^2]u^0 + (T^1 - o^1)[1 - (o^1)^2]u^1$$

$$+ \left[-w + \frac{1}{1 - (o^1)^2}\frac{\partial T^1}{\partial u^1}\right]\frac{\mathrm{d}u^1}{u^1}$$

This update exploits the advantages of the modified training scheme when the first-order information is available. In addition, an adaptive learning rate l_r is implemented in both the normal backpropagation term as well as the modification term:

$$\mathrm{d}w = l_{r1}(T^0 - o^0)[1 - (o^0)^2]u^0 + l_{r2}(T^1 - o^1)[1 - (o^1)^2]u^1$$

$$+ l_{r3}\left[-w + \frac{1}{1 - (o^1)^2}\frac{\partial T^1}{\partial u^1}\right]\frac{\mathrm{d}u^1}{u^1}$$

The learning rates are adapted independently, based on the change in the sum squared error in any epoch.

In the case of multiple neurons and multiple layers, there is an additional hybrid structure to the training scheme—a block and sequential treatment in the presentation of data for weight update in each iteration. This is explained in Sec. II.B.

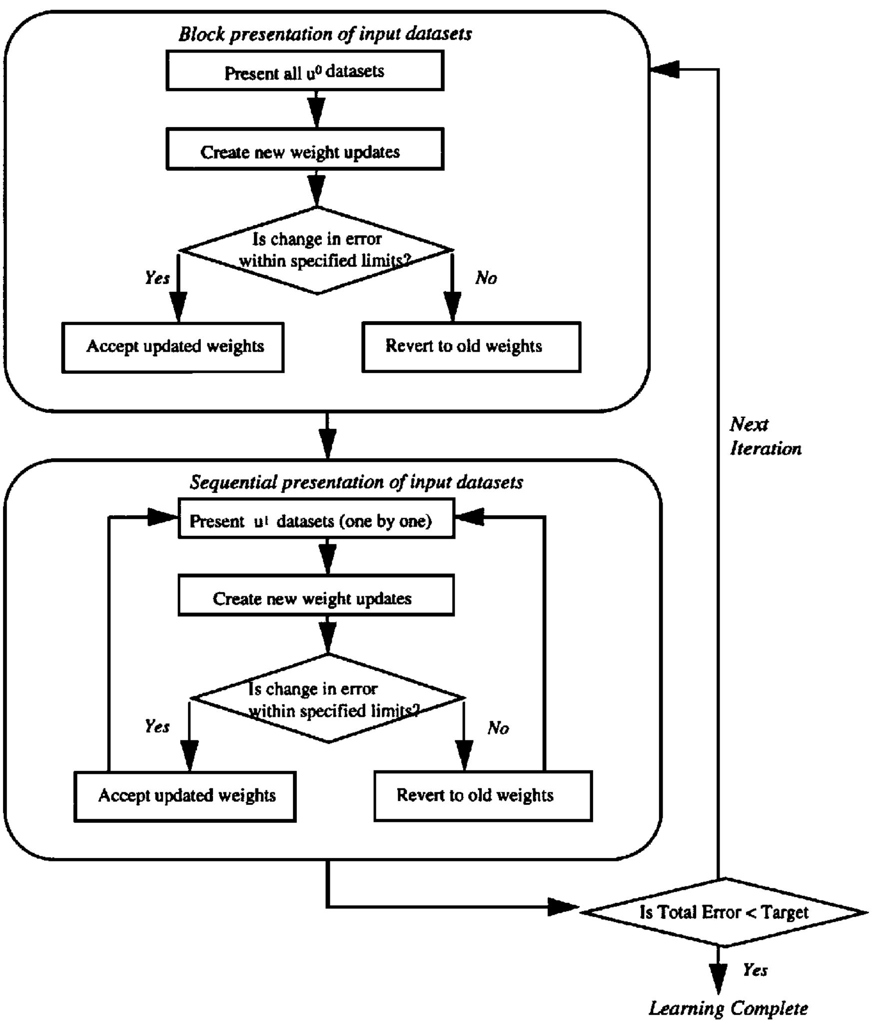

Fig. 3 Flow of the hybrid neural network learning scheme.

B. Hybrid Learning Scheme: Data Presentation

It has been shown that in a standard backpropagation learning scheme, a block presentation of the data during weight update is more efficient (faster convergence) than a sequential presentation of data and update of weights.[1,2] In the modified learning scheme, to utilize the gradient information efficiently, the data have to be presented sequentially.

In the hybrid algorithm, both features are utilized: all data with just zeroth-order information are processed in block whereas the data with zeroth- and first-order information are processed in a sequential manner. The flow chart in Fig. 3 shows the method of data presentation used while training with the hybrid learning scheme.

C. Choice of Initial Weights

The Nguyen and Widrow random generator[6] is used for the choice of initial weights. The weights are initialized such that the linear region of tan-sigmoid neurons lies near the region where inputs are likely to occur. The same initial weights are used while training with the standard, modified, and, hybrid learning schemes for comparison of performances.

D. Neural Network Approximation Example: Rosenbrock's Valley Function

The example considered here is the approximation of the Rosenbrock's valley function, given as

$$F(x_1, x_2) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2$$

The domain of operation was chosen to be [−1.3 to 1.3] for x_1 and [−0.5 to 1.3] for x_2. The training data set consists of 165 data points evenly spaced, 83 of which contain just the function value, whereas the rest also contain the gradient information. The network architecture consists of a two-layer feedforward network with two inputs (x_1 and x_2) and one output (F). Layer 1 consists of 18 tan-sigmoid neurons, and layer 2 consists of 1 linear neuron. An adaptive learning rate with an initial value of 0.075 is used.

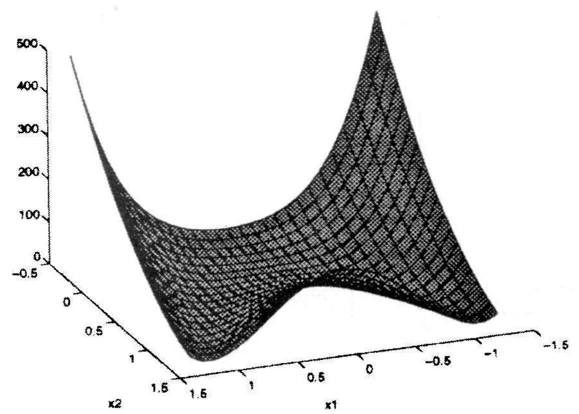

Fig. 4 Rosenbrock valley function.

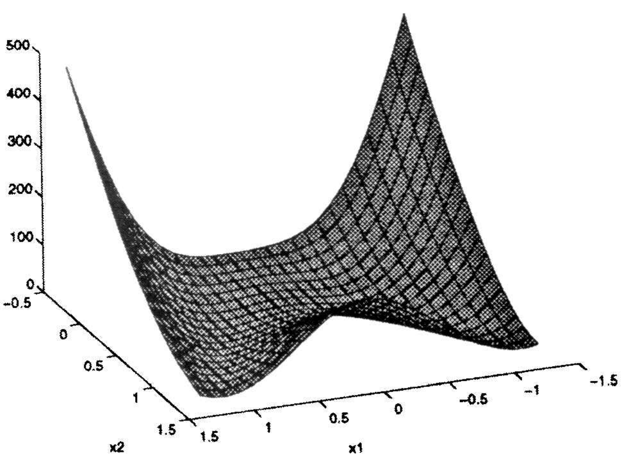

Fig. 5 Standard backpropagation trained network approximation.

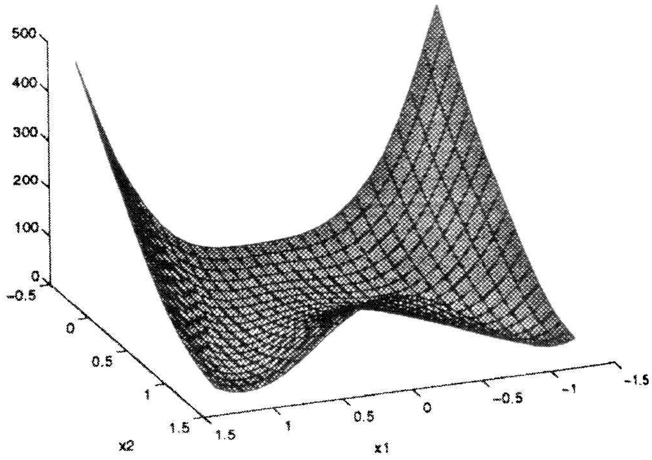

Fig. 6 Hybrid backpropagation trained network approximation.

Figures 4–6 show a surface plot of the actual Rosenbrock valley function, the standard neural network approximations (NNA), and the hybrid NNA, respectively. This is after the networks had converged to under 4% total sum squared error (TSSE). As can be seen, both neural networks approximate the Rosenbrock valley function accurately. The neural network with standard backpropagation training took 88,200 epochs to converge to under 3% TSSE, whereas the network with the hybrid back-propagation training took 79,200 epochs to converge to the same error.

III. Design Optimization Methodology

The NNA-based optimization methodology is shown in Fig. 7. This methodology is similar to the well-established Taylor-series approximation (TSA) based optimization.[7] The network is trained on both test data that include just zeroth-order information and analysis data that include both zeroth- and first-order information. The neural network trained with the hybrid learning scheme that uses the gradient information when available is used in the optimization for the response evaluations. This is then compared with a neural network trained with standard backpropagation scheme that uses just the zeroth-order information in all of the data.

The design optimization problem is posed as constrained optimization problem of the following form:

To find the set of design variables, X, that
Minimizes/Maximizes:

$$F(X)$$

Subject to:

$$g_j(X) \leq 0; \qquad j = 1, \quad \text{number of inequality constraints}$$

$$x_i^l \leq x_i \leq x_i^u; \quad \text{bounds on design variables}$$

The method of feasible directions, programmed in ADS code, is used to solve the constrained optimization problem.

IV. Design Optimization Example: Aircraft Engine Guide Vane

The aircraft engine composite guide vane[4] is used as the design example. In this example, both test and analysis data are used for training the neural network. The design problem addressed here is to determine the composite ply orientations of the guide vane that would produce a stiffer structure. The stiffness requirements are specified in terms of frequency requirements, and producibility requirements of the composite ply layup are also an integral part of the guide vane design.

The guide vane is modeled using 990, eight-noded isoparametric finite elements. The composite ply layup is symmetrical and is given by

$$[(\theta_1, \theta_2, \theta_1, \theta_3, \theta_4, \theta_3, \theta_1), (\theta_2, \theta_1, \theta_1, \theta_4, \theta_1, \theta_1)_n]_{\text{sym}}$$

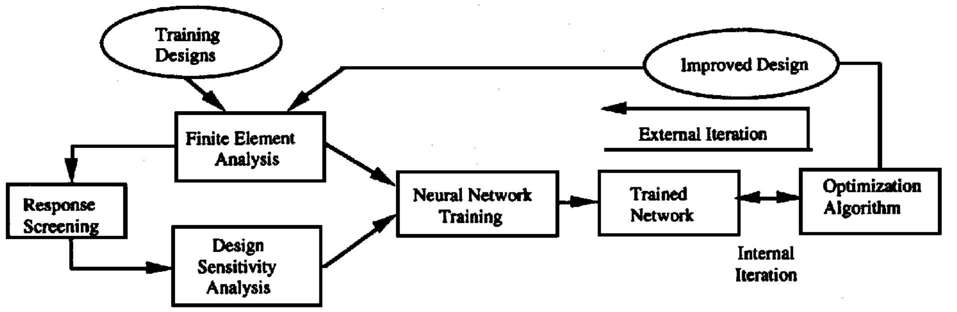

Fig. 7 Neural network approximation based design optimization methodology.

Table 1 Aircraft engine guide vane results (normalized values)

		NN-based option		
Data	Baseline	Standard	Modified	Hybrid
θ_1	0	−5	−5	−5
θ_2	+45	+30	+30	+30
θ_3	+90	+90	+90	+90
θ_4	−45	−25	−15	−15
ω_{1F}	1.0	1.058	1.062	1.062
ω_{3F}	5.7	5.52	5.39	5.39
ω_{3T}	6.0	5.92	5.87	5.87
ω_{2S}	9.9	9.96	9.98	9.98
No. of datasets used in NN training		$29u^0$	$29u^1$	$22u^1$ $+7u^0$
No. of epochs for NN training convergence		233,822	244,658	200,891
NN error goal		0.05	0.05	0.05

The ply layup consists of the two generation sets with four unique ply angles ($\theta_1, \theta_2, \theta_3, \theta_4$,). Some of the producibility requirements on the composite ply orientations include 1) that the plies be layered as a three-ply packs, 2) that the first and third ply orientation angle in each pack be the same (...), and 3) that all plies be of constant thickness of approximately 5 mils.

The design requirements on the guide vane frequencies include

Mode 1 (first flex frequency):

$$\omega_{1F} > 1.0 \text{ Hz}$$

Mode 6 (third flex frequency):

$$4.0 < \omega_{3F} < 6.7 \text{ Hz}$$

Mode 7 (third torsion frequency):

$$4.0 < \omega_{3T} < 6.7 \text{ Hz}$$

Mode 14 (two-stripe frequency):

$$\omega_{2S} > 9.8 \text{ Hz}$$

The frequencies are normalized with respect to first flex frequency corresponding to the baseline layup.

The neural network architecture consists of a two-layer feed-forward network with 10 tan-sigmoid neurons in the hidden layer (layer 1) and 4 linear neurons in the output layer (layer 2). The network inputs consist of the four unique ply angles ($\theta_1, \theta_2, \theta_3, \theta_4$) and the network outputs consists of the four guide vane frequencies of interest. A feedforward network with 10 nodes in the hidden layer is used. For the network with standard backpropagation learning, all 29 datasets (only zeroth-order information) are used, whereas for the modified backpropagation learning, only the 29 datasets also include the first-order (gradients of frequencies with respect to ply angles) information. The gradients are evaluated using a semianalytical sensitivity procedure. For the hybrid backpropagation learning scheme, 7 datasets only contain zeroth-order information whereas the remaining 22 datasets include zeroth- and first-order information. The results for the three different training schemes and the associated design optimization are presented in Table 1.

The hybrid training presents the zeroth-order data in block and the first-order data sequentially. As noted previously, block presentation of the data during weight update is more efficient (faster convergence) than a sequential presentation of data and update of weights. In the hybrid algorithm, the data with just zeroth-order information are processed in block whereas the data with zeroth- and first-order information are processed in a sequential manner. The results provided in Table 1 for the standard and modified backpropagation present all data (both zeroth and first order) sequentially. This accounts for the hybrid training requiring slightly fewer training epochs than the modified backpropagation training.

V. Conclusions

This paper investigates a hybrid backpropagation learning algorithm for handling variable-complexity data commonly encountered in multidisciplinary design optimization problems. Not all of the disciplinary solvers can provide for both zeroth- and first-order (gradient) data. The hybrid learning algorithm can accommodate a combination of datasets, with and without first-order information, for training the network.

It is reasonable to conclude that the quality of approximations is better with the learning schemes using gradient data. Within the schemes using the gradient data, the efficiency of approximation is improved by the hybrid training scheme, compared to the modified backpropagation training scheme, and this is due primarily to the presentation of zeroth-order data in block.

The hybrid learning scheme requires more I/O datasets for training than the number of function evaluations required by the well-established TSA using analytical or semianalytical gradient calculations. However, the region of applicability of the proposed hybrid NNA is larger.

References

[1]Zurada, J. M., *Introduction to Artificial Neural Systems*, West Publishing, St. Paul, MN, 1992, Chap. 3.

[2]Kosko, B., *Neural Networks for Signal Processing*, Prentice–Hall, Englewood Cliffs, NJ, 1992, Chap. 1.

[3]Hunt, K. J., Sbarbaro, D., Zbikowski, R., and Gawthrop, P. J., "Neural Networks for Control Systems: A Survey," *Automatica*, Vol. 28, 1992, pp. 1083–1122.

[4]Kodiyalam, S., and Gurumoorthy, R., "Neural Networks with Modified Backpropagation Learning Applied to Structural Optimization," *AIAA Journal*, Vol. 34, No. 2, 1996, pp. 408–412.

[5]Hajela, P., and Berke, L., "Neurobiological Computational Models in Structural Analysis and Design," *Computers and Structures*, Vol. 41, No. 4, 1991, pp. 657–667.

[6]Nguyen, D., and Widrow, B., "Improving the Learning Speed of Two Layer Networks by Choosing Initial Values of the Adaptive Weights," *Proceedings of the IEEE International Joint Conference of Neural Networks*, Vol. 3, Inst. of Electrical and Electronics Engineers, 1990, pp. 21–26.

[7]Miura, H., Vanderplaats, G. N., and Kodiyalam, S., "Experiences in Large Scale Structural Sign Optimization," *Applications of Supercomputers in Engineering: Fluid Flow and Stress Analysis Applications*, Elsevier, Amsterdam, 1989, pp. 251–268.

A. D. Belegundu
Associate Editor