JOURNAL OF GUIDANCE, CONTROL, AND DYNAMICS
Vol. 22, No. 5, September–October 1999

Robustness Analysis of Neural Networks with an Application to System Identification

K. KrishnaKumar* and K. Nishita†
University of Alabama, Tuscaloosa, Alabama 35487-0280

The robustness characteristics of multilayered feedforward neural networks (NNs) are analyzed using linear systems theory. Robustness of NN is defined as the NN's ability to perform within certain bounds of its nominal (without uncertainty) performance in the presence of bounded uncertainty. An induced Euclidean matrix norm is used to derive error bounds for NN with activation functions that exhibit predominantly linear behavior. Lyapunov stability theory is used to derive bounds on the nonlinear variations in NNs with activation functions that exhibit nonlinear behavior. A Monte Carlo simulation analysis is conducted to examine the robustness characteristics of fully and sparsely connected networks. The following conclusions are drawn on the basis of the analysis: 1) sparsity in the NN connection topology is highly desirable to achieve robustness; 2) two-hidden-layer networks with an equal number of neurons in each layer exhibit very poor robustness; 3) a fully forward connected network with sparsity is the most robust and accurate for a given number of neurons; and 4) for NNs with many neurons, activation functions with highly nonlinear regions exhibit poor robustness. A system identification of a nonlinear model is presented to reinforce the results of the robustness analysis.

Nomenclature

A	= first-order derivative matrix
$h[\,]$	= nonlinear perturbation
I_n	= identity matrix
P	= matrix that satisfies the Lyapunov equation
S	= frequency scale
w_{ij}	= weight connecting the ith neuron to the jth neuron
x_i	= input to the ith neuron
y_i	= output of the ith neuron
β	= some positive constant
ξ	= fixed gain
ρ	= largest singular value of a matrix
ρ_s	= degree of stability
$\hat{\rho}_s$	= estimated degree of stability
σ	= singular value
Φ	= system matrix
$\phi_i(\,)$	= nonlinear activation function
$\|\cdots\|$	= norm
$\|\cdots\|_F$	= Frobenius norm
$\|\cdots\|_s$	= spectral norm

Introduction

ROBUSTNESS of neural networks (NNs) is an important characteristic for many applications, ranging from simple function mapping to more complicated nonlinear control problems. Robustness of any system can be defined as the system's ability to perform within certain bounds of its nominal (without uncertainty) performance in the presence of bounded uncertainty. In the NN context, robustness relates bounded output error of the NN to bounded input errors. The inputs are seen as inputs to any neurons and the outputs are seen as outputs from any neurons. By considering inputs and outputs from individual neurons instead of the whole NN, we can include uncertainties related to structural failures (such as loss of nodes and weights), internal noise (hardware related) in the NN, and also examine uncertainties in the inputs to the NN system.

Where does generalization fit into all of these? Generalization of an NN is its ability to sensibly interpolate input patterns that are new to the network.[1] Generalization is a loosely used term. Training an NN for a desired generalization is impossible because we do not know the function to be mapped a priori. To illustrate this, we present in Fig. 1 a set of training data pairs $[x, y = f(x)]$. It is obvious to see that one can have an infinite number of lines go through these points. Which one is the best? Because this is unknown, how can one predict a priori the generalization characteristics of the NN? This can be done only approximately by using test data after the training is done. In conclusion, generalization depends to a great extent on the training set used and is also difficult to define a priori. This is not true about robustness. Robustness is easy to define using input–output uncertainty bounds as shown in Fig. 1.

Most popular NNs today use multilayered feedforward networks with connections going from one layer to the next. Several investigators have shown the approximating capabilities of these networks. Another type of network that has received some attention is the fully forward connected network.[2,3] Using empirical results, KrishnaKumar[3] has shown that if this type of network is made sparse, the networks are more accurate and robust for a given number of neurons. Many other researchers[4–6] have shown similar empirical results using sparse multilayered feedforward networks.

We address the general question of how to define relative robustness of different NN structures. We assume that the network has been trained to provide a desired accuracy for the training data set. The question then is simply one of how to relate the given NN structure to certain bounds on its mapping error for untrained data. We first define robustness given an NN structure and then examine the robustness of layered networks, fully forward connected networks, and sparse networks.

We begin with the definitions of NN structures that are discussed in this paper. Next, we state the linear systems theory results that will be used to define robustness. We then present an interpretation of NN equations in terms of linear systems theory and present a Monte Carlo simulation analysis to examine the robustness characteristics. The following conclusions are drawn on the basis of the analysis: 1) sparsity in the NN connection topology is highly desirable to achieve robustness; 2) two-hidden-layer networks with an equal number of neurons in each layer exhibit very poor robustness; 3) a fully forward connected network with sparsity is the most robust and accurate for a given number of neurons; and 4) for NN with many neurons, activation functions with highly nonlinear regions exhibit poor robustness. Finally, a system identification problem is presented to reinforce the results of the robustness analysis.

Received 13 July 1998; revision received 1 March 1999; accepted for publication 1 March 1999.

*Associate Professor and Interim Department Head, Department of Aerospace Engineering and Mechanics; kkumar@coe.eng.ua.edu. Associate Fellow AIAA.

†Graduate Research Assistant, Department of Aerospace Engineering and Mechanics.

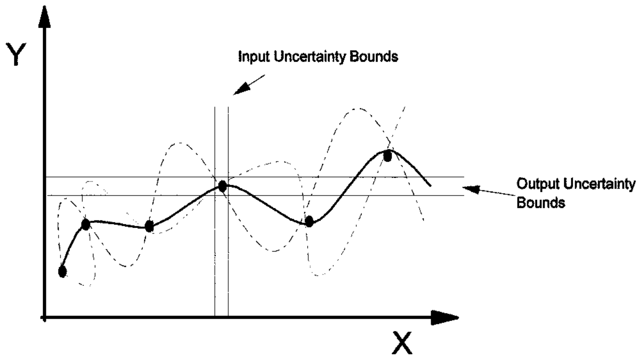

Fig. 1 Generalization vs robustness: ——, NN mapping; - - -, unknown function; and •, training samples.

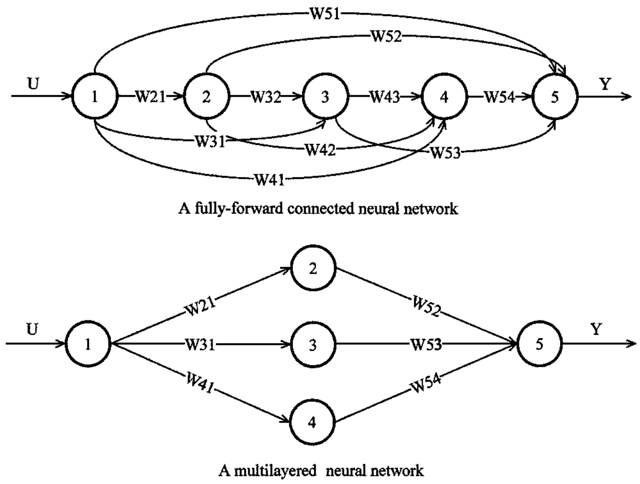

Fig. 2 NN arrangements.

NN Preliminaries

Without loss of generality, we assume one input/one output NNs with $N-2$ sigmoidal hidden neurons. The hidden neurons could be arranged either in a layered form or in a fully forward connected form (Fig. 2). Also, the fully forward connected NN with the proper connections removed is equivalent to the layered form shown in Fig. 3. Because our main concern is one of robustness as it relates to NN structures, it is assumed that the user has made a choice of the NN weights via some learning scheme. The equation for a fully forward connected NN is given as

$$x_i = \phi_i\left(\sum_{j=1}^{i-1} w_{ij}x_j\right), \qquad y_i = x_i, \qquad 2 \le i \le N \tag{1}$$

where x_N is NN output and x_1 is NN input and the input neuron is a linear neuron.

Equation (1) can be used to arrive at layered networks by simply zeroing out appropriate elements of the weight matrix. For example, Eq. (1) can be expanded as follows:

$$X_1 = X_1, \qquad X_2 = \phi_2(W_{21}X_1), \qquad X_3 = \phi_3(W_{31}X_1 + W_{32}X_2)$$
$$X_4 = \phi_4(W_{41}X_1 + W_{42}X_2 + W_{43}X_3) \tag{1a}$$
$$X_5 = \phi_5(W_{51}X_1 + W_{52}X_2 + W_{53}X_3 + W_{54}X_4)$$

This is essentially the case for the fully forward connected NN shown in Fig. 2. The multilayered NN shown in Fig. 2 also can be obtained using Eq. (1) by simply zeroing out the appropriate elements of the weight matrix:

$$X_1 = X_1, \qquad X_2 = \phi_2(W_{21}X_1), \qquad X_3 = \phi_3(W_{31}X_1) \tag{1b}$$
$$X_4 = \phi_4(W_{41}X_1), \qquad X_5 = \phi_5(W_{52}X_2 + W_{53}X_3 + W_{54}X_4)$$

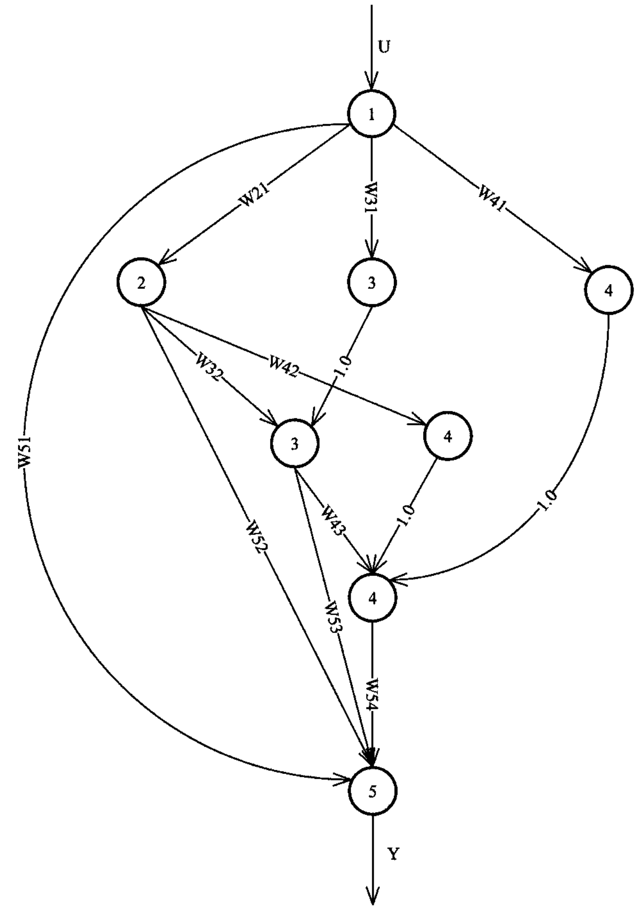

Fig. 3 Fully forward connected NN arranged as a layered NN.

The weight elements, W_{32}, W_{42}, W_{43}, and W_{51} are zeroed out in this case.

Mathematical Preliminaries

Vector Norm

For any vector $\boldsymbol{x} \in R^n$, the Frobenius (Euclidean) norm is given as

$$\|\boldsymbol{x}\|_F = \|\boldsymbol{x}\|_2 = \sqrt{\sum_j |x_j|^2} \tag{2}$$

Spectral Matrix Norm

A spectral matrix norm $\|G\|_s$ of matrix $G \in R^{n\times n}$ is a matrix norm induced by the Euclidean vector norm and is given as

$$\|G\|_s = \sqrt{\lambda_{\max}[G^TG]} = \sigma_{\max}[G] \tag{3}$$

i.e., the spectral norm equals the largest singular value of matrix G. On the basis of the preceding definitions and using Holder's inequality, we have, for a given $\boldsymbol{y} = G\boldsymbol{x}$,

$$\|\boldsymbol{y}\| \le \|G\|\|\boldsymbol{x}\| \Rightarrow \|\boldsymbol{y}\| \le \sigma_{\max}[G]\|\boldsymbol{x}\| \tag{4}$$

Examining the preceding equations, we see that the $\sigma_{\max}[G]$ represents the least upper bound for the ratio of the output and input norms.

Lyapunov Stability Theory

Let a nonlinear perturbed discrete system be

$$\boldsymbol{x}(\boldsymbol{k}+1) = \Phi\boldsymbol{x}(\boldsymbol{k}) + \boldsymbol{h}[k, \boldsymbol{x}(\boldsymbol{k})] \tag{5}$$

where $\Phi \in R^{n\times n}$ is the state transition matrix, $\boldsymbol{x}(\boldsymbol{k})$ is the state vector at time-step $\boldsymbol{k}$, and $\boldsymbol{h}[\boldsymbol{k}, \boldsymbol{x}(\boldsymbol{k})]$ is the nonlinear perturbation in the system. Let the nonlinear perturbation be bounded by

$$\|\boldsymbol{h}[k, \boldsymbol{x}(\boldsymbol{k})]\| \le \xi\|\boldsymbol{x}(\boldsymbol{k})\|, \qquad \xi > 0 \tag{6}$$

i.e., the norm of $\boldsymbol{h}$ does not exceed the norm of $\boldsymbol{x}$ multiplied by a fixed gain ξ. The system given in Eq. (5) is exponentially stable if the solution $\boldsymbol{x}(\boldsymbol{k})$ satisfies the condition[7]

$$\|\boldsymbol{x}(\boldsymbol{k})\|_F \leq \beta\|\boldsymbol{x}(0)\|_F \exp(\boldsymbol{k}\, ln\rho_s) = \beta\|\boldsymbol{x}(0)\|_F \rho_s^k$$

$$\beta > 0, \quad 0 \leq \rho_s < 1 \qquad (7)$$

An estimate of ρ_s can be derived for the nonlinearly perturbed system using norm-like Lyapunov functions in terms of the nonlinear perturbation bound ξ as

$$\xi < \xi_N = 1 - \hat{\rho}_s = \frac{1.0}{\sigma_{\max}[P] + \sqrt{\sigma_{\max}[P]\sigma_{\max}[P - I_n]}} \qquad (8)$$

where $P \in R^{n\times n}$ satisfies the Lyapunov equation

$$\Phi^T P \Phi - P = -I_n \qquad (9)$$

The preceding equations imply that the stability margin of the nonlinearly perturbed system is governed by the system matrix Φ and the gain ξ_N. Larger ξ_N implies greater stability.

Robustness Analysis of NNs

The NN equation (1) can be interpreted as a nonlinear system with x_i representing the inputs to the system and y_i representing the outputs of the system. Now, let $\tilde{\boldsymbol{X}}$ be a vector corresponding to inputs not used during the training. Then, the corresponding outputs are given by

$$\tilde{y}_i = \phi_i\left(\sum_{j=1}^{i-1} w_{ij}\tilde{x}_j\right), \qquad 1 \leq i \leq N \qquad (10)$$

Now, let y_i^* be the desired output and x_i^* be the input such that

$$y_i^* = \phi_i\left(\sum_{j=1}^{i-1} w_{ij}x_j^*\right), \qquad 1 \leq i \leq N \qquad (11)$$

Expanding the right-hand side of Eq. (10) in a Taylor-series form about x^*, we get the vector equation

$$\tilde{\boldsymbol{Y}} - \boldsymbol{Y}^* = A(\tilde{\boldsymbol{X}} - \boldsymbol{X}^*) + \boldsymbol{h}[\tilde{\boldsymbol{X}} - \boldsymbol{X}^*] \qquad (12)$$

where $h[\tilde{\boldsymbol{X}} - \boldsymbol{X}^*]$ = the higher-order nonlinear terms.

It is clear from Eq. (12) that if $\|\tilde{\boldsymbol{Y}} - \boldsymbol{Y}^*\|$ is small for a given $\|\tilde{\boldsymbol{X}} - \boldsymbol{X}^*\|$, the NN will be considered robust. Also, in Eq. (12), A is the first-order derivative matrix. Depending on the type of NN structure, the matrix will have different entries. Given below are examples of three first-order derivative matrices for three types of NN structures. In the matrices given below, $\Phi' = [\partial\Phi(\,)/\partial(\,)]$. Note that the sparsity in these matrices can be created by zeroing either the weights or the slopes of the activation functions.

First order derivative matrix for a fully-forward connected NN (input, 7 hidden neurons, and 1 output):

$$\begin{bmatrix}
0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00\\
\Phi'W_{21} & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00\\
\Phi'W_{31} & \Phi'W_{32} & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00\\
\Phi'W_{41} & \Phi'W_{42} & \Phi'W_{43} & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00\\
\Phi'W_{51} & \Phi'W_{52} & \Phi'W_{53} & \Phi'W_{54} & 0.00 & 0.00 & 0.00 & 0.00 & 0.00\\
\Phi'W_{61} & \Phi'W_{62} & \Phi'W_{63} & \Phi'W_{64} & \Phi'W_{65} & 0.00 & 0.00 & 0.00 & 0.00\\
\Phi'W_{71} & \Phi'W_{72} & \Phi'W_{73} & \Phi'W_{74} & \Phi'W_{75} & \Phi'W_{76} & 0.00 & 0.00 & 0.00\\
\Phi'W_{81} & \Phi'W_{82} & \Phi'W_{83} & \Phi'W_{84} & \Phi'W_{85} & \Phi'W_{86} & \Phi'W_{87} & 0.00 & 0.00\\
\Phi'W_{91} & \Phi'W_{92} & \Phi'W_{93} & \Phi'W_{94} & \Phi'W_{95} & \Phi'W_{96} & \Phi'W_{97} & \Phi'W_{98} & 0.00
\end{bmatrix}$$

First order derivative matrix for a single hidden layer NN (1 input, 7 neurons in the hidden layer, and 1 output):

$$\begin{bmatrix}
0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00\\
\Phi'W_{21} & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00\\
\Phi'W_{31} & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00\\
\Phi'W_{41} & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00\\
\Phi'W_{51} & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00\\
\Phi'W_{61} & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00\\
\Phi'W_{71} & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00\\
\Phi'W_{81} & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00\\
0.00 & \Phi'W_{92} & \Phi'W_{93} & \Phi'W_{94} & \Phi'W_{95} & \Phi'W_{96} & \Phi'W_{97} & \Phi'W_{98} & 0.00
\end{bmatrix}$$

First order derivative matrix for a two-hidden layer NN (1 input, 4 neurons in the first hidden layer, 3 neurons in the second hidden layer, and 1 output):

$$\begin{bmatrix}
0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00\\
\Phi'W_{21} & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00\\
\Phi'W_{31} & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00\\
\Phi'W_{41} & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00\\
\Phi'W_{51} & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00\\
0.00 & \Phi'W_{62} & \Phi'W_{63} & \Phi'W_{64} & \Phi'W_{65} & 0.00 & 0.00 & 0.00 & 0.00\\
0.00 & \Phi'W_{72} & \Phi'W_{73} & \Phi'W_{74} & \Phi'W_{75} & 0.00 & 0.00 & 0.00 & 0.00\\
0.00 & \Phi'W_{82} & \Phi'W_{83} & \Phi'W_{84} & \Phi'W_{85} & 0.00 & 0.00 & 0.00 & 0.00\\
0.00 & 0.00 & 0.00 & 0.00 & 0.00 & \Phi'W_{96} & \Phi'W_{97} & \Phi'W_{98} & 0.00
\end{bmatrix}$$

Letting $\tilde{Y} - Y^* = \delta Y$ and $\tilde{X} - X^* = \delta X$, we get, from Eq. (12),

$$\delta Y = A\delta X + h[\delta X] \tag{13}$$

Case 1: Predominantly Linear Activations

If $h[\delta X]$ is small, we can approximate Eq. (13) as $\delta Y = A\delta X$. Recalling the Holder's inequality for vector and matrix norms, we have

$$\|\delta Y\|_F \leq \|A\|_s \|\delta X\|_F \tag{14}$$

For a Euclidean norm, $\|A\|_s = \sigma_{\max}[A] = \rho$. This implies that if ρ is small, the output error will be a small proportion of the input deviations. In essence, the numerical value of the norm of A defines the robustness of the mapping, and smaller ρ implies better robustness. Figure 4 presents the results of a Monte Carlo simulation conducted for various types of NN structures. The entries of matrix A were chosen randomly to lie between -1 and $+1$. In all, 500 samples were generated for each type of NN structure. Sparsity in the network was created by randomly zeroing out the nonzero elements of the A matrix. Figures 4 and 5 present the mean and standard deviation of the robustness measure ρ. Figure 6 presents the number of connections (or nonzero elements in the weight matrix) for different NN structures used in this study. The following observations are made from Figs. 4–6:

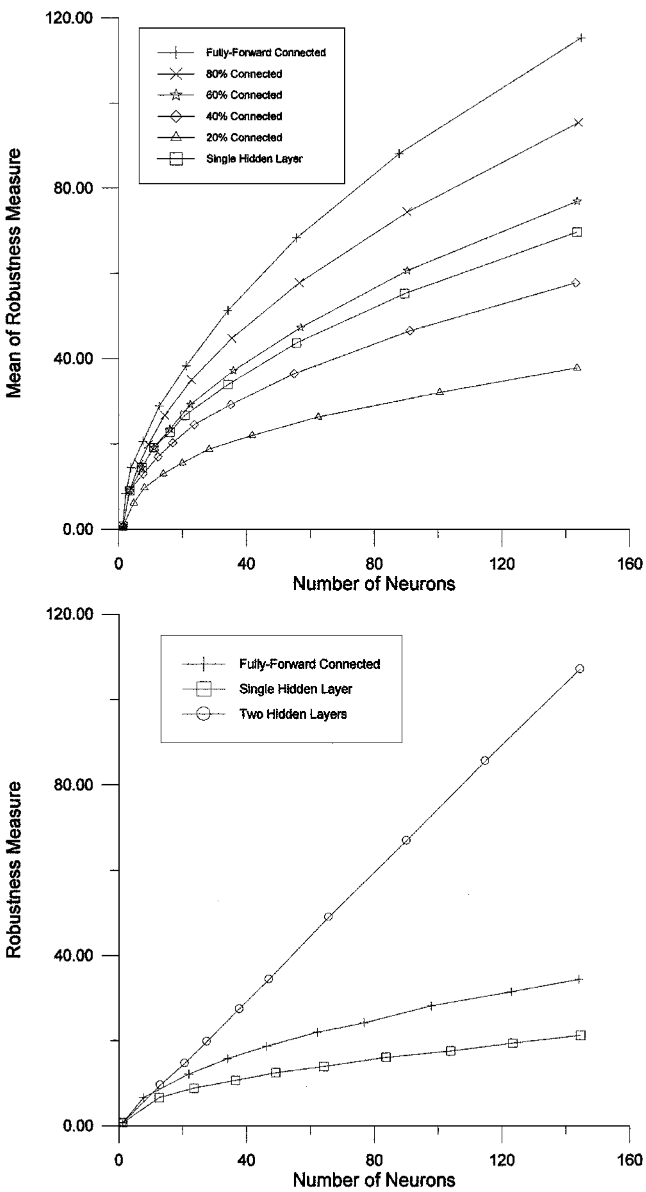

Fig. 4 Mean of ρ for various NN structures.

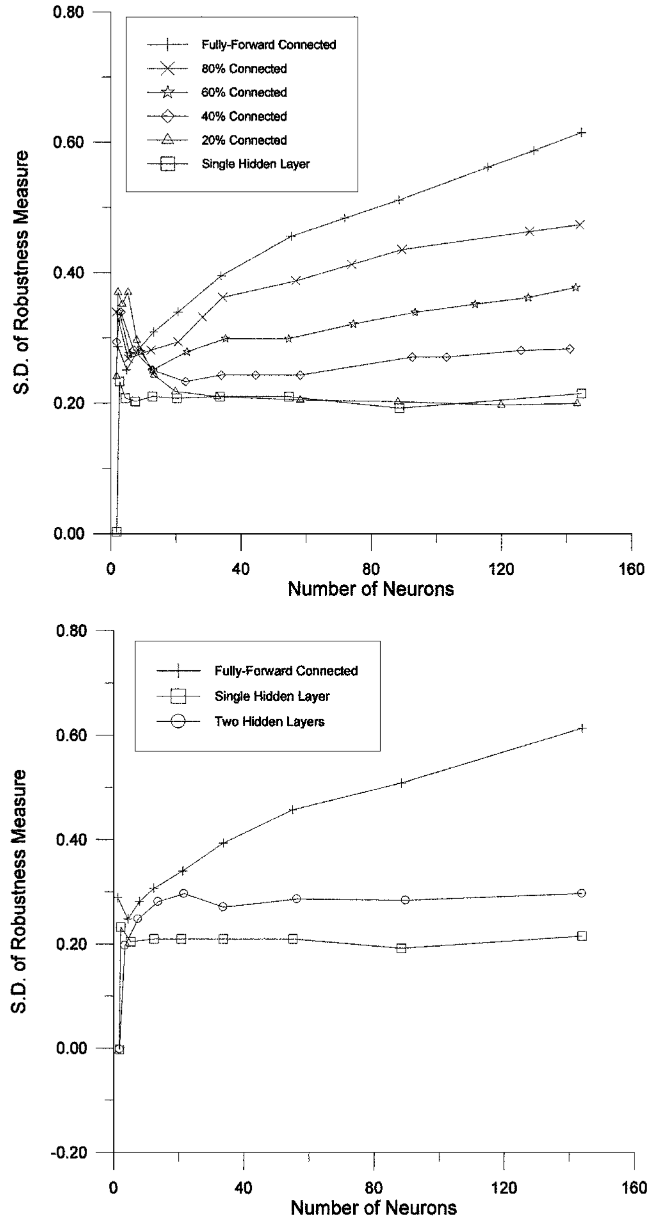

Fig. 5 Standard deviation of ρ for various NN structures.

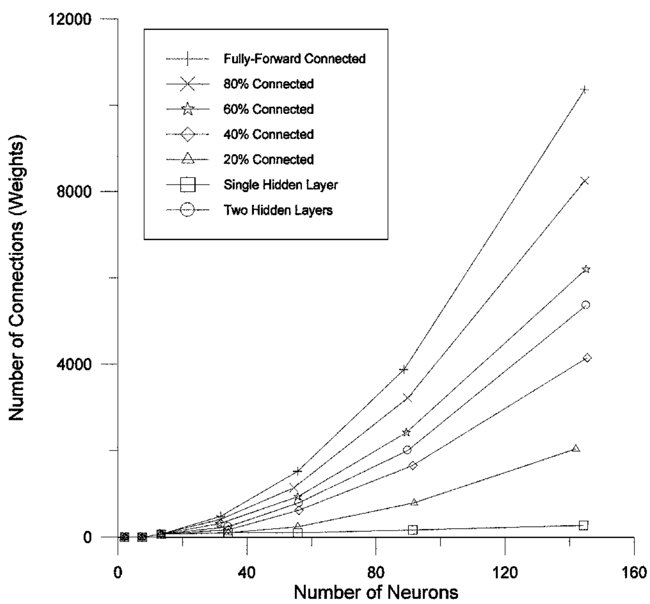

Fig. 6 Number of connections (or weights) for various NN structures.

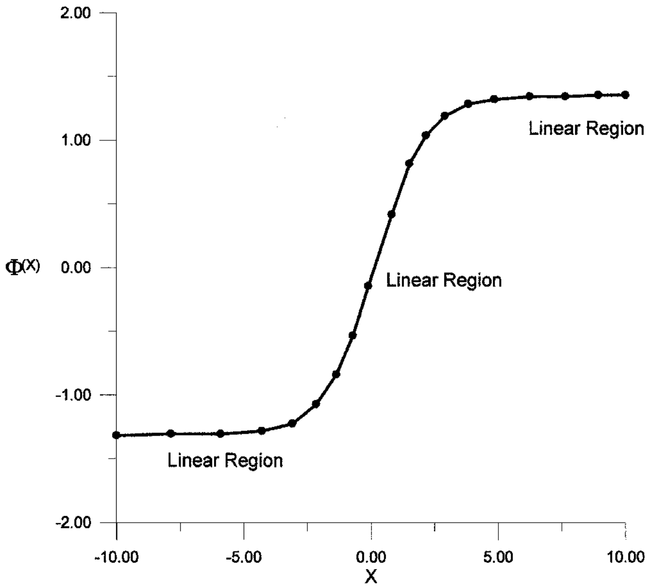

Fig. 7 Sigmoidal activation function.

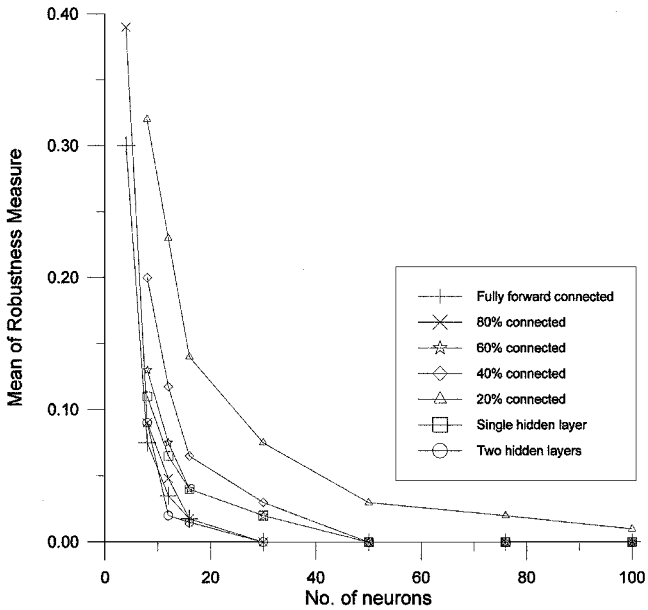

Fig. 8 Mean of ξ_N for various NN structures.

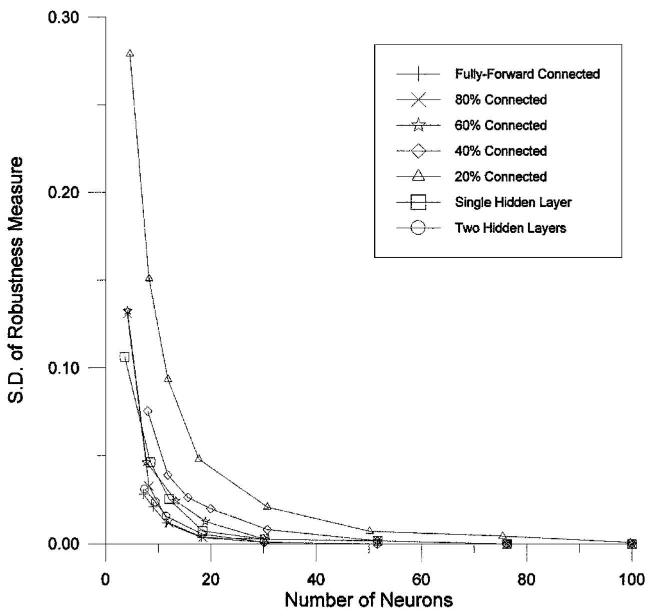

Fig. 9 Standard deviation of ξ_N for various NN structures.

1) The standard deviation of the robustness measure is small and remains a constant for low-sparsity networks. This implies that the mean of the robustness measure is a good indicator of robustness, regardless of the choice of NN weights.

2) As the number of neurons increases, robustness decreases.

3) Two hidden-layer networks with an equal number of neurons in each layer exhibit very poor robustness.

4) When Figs. 4–6 are examined, it can be concluded that the fully forward connected NN with sparsity is the best in terms of both accuracy of mapping (because of higher number of available free parameters or weights; see Fig. 6) and robustness for a given number of neurons. This result was stated in Ref. 3 using empirical data.

Case 2: Nonlinear Activations

Most of the neuronal activation functions in use today exhibit locally linear behavior. For example, the sigmoidal network has three distinctly linear regions, as shown in Fig. 7. The analysis conducted earlier should be applicable to the three linear regions shown in Fig. 7. In the case in which the nonlinearity cannot be ignored, we have the following situation:

$$\boldsymbol{h}[\delta \boldsymbol{X}] \neq 0, \qquad \|\boldsymbol{h}[\delta \boldsymbol{X}]\| \leq \xi \|\delta \boldsymbol{X}\| \tag{15}$$

Now, combining Eqs. (13–15), we get

$$\|\delta \boldsymbol{Y}\| \leq \|A\| \|\delta \boldsymbol{X}\| + \xi \|\delta \boldsymbol{X}\| \tag{16}$$

Because our interest is to compare robustness between different NNs with nonlinear behavior, we define robustness in terms of an upper limit on the allowable nonlinear variations, i.e., an upper limit for ξ. Here we use the Lyapunov stability condition presented in the Mathematical Preliminaries section and define an acceptable upper bound for ξ as

$$\xi < \xi_N = 1 - \rho_s = \frac{1.0}{\sigma_{\max}[P_A] + \sqrt{\sigma_{\max}[P_A]\sigma_{\max}[P_A - I_n]}} \tag{17}$$

In Eq. (17), P_A satisfies the Lyapunov equation

$$A^T P_A A - P_A = -I_n \tag{18}$$

From Eqs. (16) and (17), it is clear that ξ_N is a good measure of robustness for the nonlinear mapping as it relates to the bounds on the NN output presented in Eq. (13). Note also that higher ξ_N implies greater robustness. Figures 8 and 9 present the mean and standard deviation of the robustness measure ξ_N. From these figures, it is seen that the robustness to nonlinear variations in the activation functions is very similar to the linear case; i.e., sparsity is important for robustness and the two-hidden-layer network exhibits very poor robustness. In addition it is seen that, as the number of neurons increases, the mean and the standard deviation of ξ_N approach 0.0. This suggests that, for large NNs, highly nonlinear activation functions are not well suited for robustness.

System Identification Example

Here we consider a system identification of nonlinear models. This application illustrates the benefits of synthesizing sparse NNs. This example is obtained from Ref. 8. The model to be identified is

$$y_p(k+1) = f(y_p(k), y_p(k-1), y_p(k-2), u(k), u(k-1)) \tag{19}$$

Table 1 SSE for robustness test—system identification example

Frequency scale, S	NN arrangement	Phase bias, ϕ, deg 15	30	45	60	90
0.1	Sparse	0.40194	0.45836	0.53856	0.62563	0.89880
	Fully connected	0.78436	0.85138	0.90008	0.94100	1.16392
2.0	Sparse	1.11487	1.25982	1.32078	1.33565	1.26664
	Fully connected	1.29825	1.39719	1.43105	1.42465	1.36371
5.0	Sparse	4.90935	4.93914	4.69217	4.63619	4.74710
	Fully connected	9.41876	9.34034	9.02402	9.11086	9.36692
10.0	Sparse	7.32307	16.3174	23.7026	28.1521	32.6886
	Fully connected	14.1790	23.3422	31.0133	35.3847	39.3141

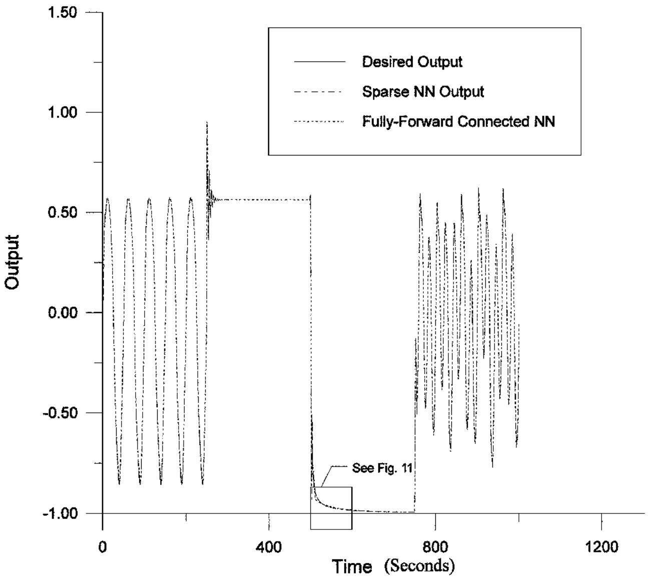

Fig. 10 Comparison of NN performance on the trained data for the system identification problem.

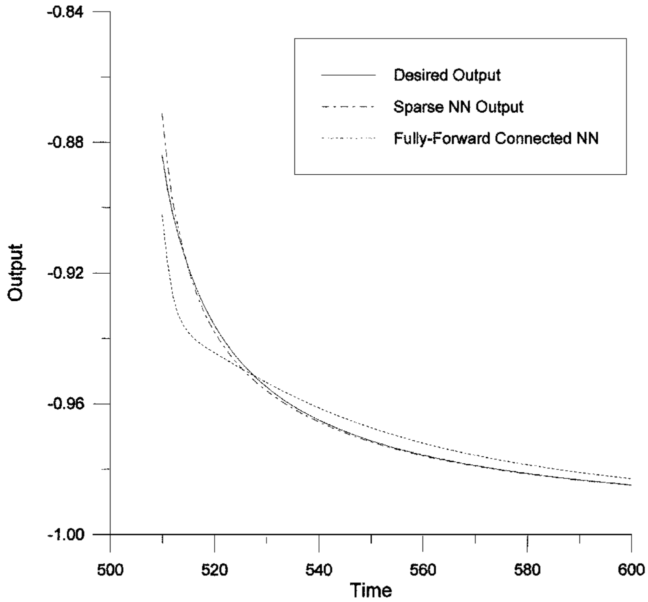

Fig. 11 Comparison of NN performance on the trained data for the system identification problem for 510 (s) < time < 600 (s).

where

$$f(x_1, x_2, x_3, x_4, x_5) = \frac{x_1 x_2 x_3 x_5 (x_3 - 1) + x_4}{1 + x_3^2 + x_2^2}$$

$$\begin{aligned} u(k) &= \sin(\pi k/25), \quad 0 \le k < 250 \\ &= 1.0, \quad 250 \le k < 500 \\ &= -1.0, \quad 500 \le k < 750 \\ &= 0.3\sin(\pi k/25) + 0.1\sin(\pi k/32) + 0.6\sin(\pi k/10) \\ &\qquad 750 \le k < 1000 \end{aligned} \tag{20}$$

The NN inputs included a bias, $y_p(k)$, $y_p(k-1)$, $y_p(k-2)$, $u(k)$, and $u(k-1)$ and the output was $y_p(k+1)$. Also, the NN consisting of 20 hidden neurons were used. The NN was trained till the sum of squared error (SSE), Eq. (21), reached 0.2325 using a technique that creates sparse NN.[9] In the technique reported in Ref. 9, sparse NNs are created using the fully forward connected architecture presented earlier in conjunction with a pruning technique. The resulting NN structure was compared to a fully forward connected NN, which was trained to obtain a final SSE of 0.2325:

$$\text{SSE} = \frac{1}{2}\sum_{i=1}^{n}\left(y_{p_i} - \text{Desired}_i\right)^2 \tag{21}$$

where n = number of outputs.

Figures 10 and 11 show the NN performance for the data used for training. To verify system robustness, we modified the sinusoidal frequency by scaling it by a factor S and shifting the phase by a factor ϕ. Table 1 documents the comparison between the sparse and nonsparse NN. It is clear that the sparse NN performed very well.

Conclusions

Simple robustness measures for NNs with linear and nonlinear behavior are presented. Using these measures, we examined robustness of several NNs, using a Monte Carlo simulation with 500 samples for each type of NN structure. The results of the analysis reinforce the need for sparsity and the need to use fully forward connected NN with sparsity instead of the traditional multilayered NNs. Also, two-hidden-layer NNs with equal number of neurons in the hidden layers were shown to be the least robust among the popular structures currently in use. Another interesting result is that for NNs with a large number of neurons, nonlinear activation functions are not the best for robustness. A system identification problem is used to show the benefit of using sparse networks.

Acknowledgment

This material is based upon work supported by the National Science Foundation under Grant ECS-9113283 and ECG-941535.

References

[1]Zurada, J. M., *Introduction to Artificial Neural Systems*, West, New York, 1992, pp. 54, 55.

[2]Werbos, P. J., "Back-Propagation Through Time: What It Does and How to Do It," *Proceedings of the IEEE*, Vol. 78, No. 10, 1990, pp. 1550–1560.

[3]KrishnaKumar, K., "Optimization of the Neural Net Connectivity Pattern Using a Backpropagation Algorithm," *Journal of Neurocomputing*, Vol. 5, 1993, pp. 273–286.

[4]Mozer, M. C., and Smolensky, P., "Skeletonization: A Technique for Trimming the Fat from a Network via Relevance Assessment," Dept. of Computer Science and Institute of Cognitive Science, Univ. of Colorado, TR CU-CS-421-89, Boulder, CO, 1989.

[5]Rumelhart, D. E., McClelland, J. L., and PDP Research Group, "Parallel Distributed Processing: Explorations in the Microstructure of Cognition," *Foundations*, Vol. 1, MIT Press, Cambridge, MA, 1986, pp. 45–76.

[6]Sietsma, J., and Dow, R. J. F., "Creating Artificial Neural Networks That Generalize," *Neural Networks*, Vol. 4, No. 1, 1991, pp. 67–79.

[7]Weinmann, A., *Uncertainty Models and Robust Control*, Springer–Verlag, New York, 1991, pp. 45–62, 225–248.

[8]Narendra, K. S., and Parthasarathy, K., "Identification and Control of Dynamical Systems Using Neural Networks," *IEEE Transactions on Neural Networks*, Vol. 1, No. 1, 1990, pp. 4–27.

[9]KrishnaKumar, K., and Nishita, K., "BORN—Bama Optimized Recurrent Neural Networks," *Proceedings of the World Congress on Neural Networks*, INNS Press, Washington, DC, 1996, pp. 305–310.