Earthquake Engineering, Tenth World Conference © 1992 Balkema, Rotterdam. ISBN 90 5410 060 5

Modeling unknown structural systems through the use of neural networks

A.G.Chassiakos
California State University, Long Beach, Calif., USA

S.F.Masri
University of Southern California, Los Angeles, Calif., USA

ABSTRACT: This paper explores the potential of using neural networks to identify the internal forces of typical systems encountered in the field of earthquake engineering and structural dynamics. After formulating the identification task as a neural network learning procedure, the method is applied to a representative chain-like system under deterministic and stochastic excitations. The range of validity of the approach is demonstrated, and some application issues are discussed.

1. INTRODUCTION

System identification refers to any systematic way of deriving or improving models for dynamic systems through the use of experimental data. It is an area of considerable importance in structural engineering which has been gaining increasing attention over the last decade or so. Some representative publications on the subject are available in the work of Beck (1978), Ibanez (1979), Masri and Caughey (1979), Natke (1982), Masri and Werner (1985), and the IMAC Proceedings (1992). The methods of system identification provide a means of utilizing laboratory and field testing to improve dynamic modeling capabilities for the large civil structures, that is, high-rise buildings and large bridges and dams.

For example, by systematically utilizing dynamic test data from a structure, rather than relying on theory alone, models can be derived which provide more accurate response predictions for dynamic loads on the structure which are produced by wind or earthquakes. Another application is to continually update the model through vibration monitoring of the structure to provide a convenient method for defect identification or damage assessment (Natke and Yao (Eds), 1988; Chen, 1988; Garba, 1990).

The potential for using active control approaches to reduce the response of large civil structures under arbitrary dynamic environments, such as earthquakes, has drawn a considerable amount of interest worldwide. Among the key research topics in this area, is the development of system identification approaches that can cope with the challenging nature of physical structures encountered in the structural mechanics and earthquake engineering fields (Housner and Masri, 1990). Since the model structure in many practical dynamics problems is by no means clear, an increasing amount of attention is being devoted to nonparametric identification methods. These methods do not identify the physical parameters of the system (such as mass, stiffness, etc.) but instead identify the parameters of a mathematical model which fits the input/output data.

The present paper introduces a new nonparametric identification method for unknown dynamic systems undergoing arbitrary earthquake-type excitation. The method is based on the use of artificial neural networks as system identifiers. Artificial neural networks are the ideal choice in cases when real time processing of large amounts of data is required because of their inherent massive parallelism, fault tolerance and learning capabilities.

Several issues related to the use of artificial neural networks in dynamic system identification are presented and discussed, such as: the network size and topology, the network training algorithms, the amount and type of required experimental data, validation of the identified model and its prediction capabilities.

2. NEURAL NETWORK FORMULATION

2.1 *Neural Network Approach*

An Artificial Neural Network (ANN) is a system with inputs and outputs, composed of a number of similar nonlinear processing elements. These processing elements operate in parallel and are arranged in patterns similar to the patterns found in biological neural nets. The processing elements or nodes are connected to each other by adjustable weights (Lippmann (1987), Nguyen and Widrow (1990)). Changing these weights will change the input/output behavior of the network, hence the following is a natural goal for such a system: Choose the weights of the net in such a way as to achieve a desired input/output relationship. To achieve this goal, systematic ways of adjusting the weights have to be developed, which are referred to as training or learning algorithms. A neural net is characterized by the following:

- The processing elements
- The network topology
- The learning algorithm

Each of these network characteristics is described next.

2.2 *The Processing Elements (Nodes)*

A typical node sums n weighted inputs $u_1, u_2, \dots u_n$ and a bias term b and passes the result through a nonlinear function $\gamma(.)$ as shown in Fig. (1):

$$v = \gamma(\bar{v}) \qquad ; \qquad \bar{v} = \sum_{i=1}^{n} w_i u_i + b$$

Note that the bias term b can be considered to be either an input $u_0 = b$ connected to the node by a fixed weight $w_0 = 1$, or a constant input $u_0 = 1$ connected to the node by a weight $w_0 = b$. In this case the value of b is adjusted by the same algorithm as the rest of the weights in the net. Typical nonlinearities used in the nodes are:

- Hard-limiters
$$\gamma(x) = \begin{cases} 1, & \text{if } x \geq 0 \\ -1, & \text{if } x < 0 \end{cases}$$
- Threshold operations
$$\gamma(x) = \begin{cases} 1, & \text{if } x \geq 0 \\ 0, & \text{if } x < 0 \end{cases}$$
- The logistic function (sigmoid)
$$\gamma(x) = \frac{1}{1 + e^{-\alpha x}} \, ; \qquad \alpha > 0$$
- The hyperbolic tangent function (sigmoid)
$$\gamma(x) = \frac{1 - e^{-\alpha x}}{1 + e^{-\alpha x}} \, ; \qquad \alpha > 0$$

The choice of the nonlinearity depends on the particular application for which the network is being used. In cases when it is required that the nonlinearity is differentiable, the logistic and the hyperbolic tangent functions are widely used. Moreover, their derivatives are given by:

$$\gamma'(x) = -\alpha\gamma(x)[\gamma(x) - 1]$$

for the logistic function, and by:

$$\gamma'(x) = -(\alpha/2)[\gamma(x) + 1][\gamma(x) - 1]$$

for the hyperbolic tangent.

2.3 *The Network Topology*

The network topology depends on the way the nodes are connected to each other and to the input and output vectors. According to their topology, neural networks can be classified as:

- Single layer networks, when only one layer of nodes is present, the output layer, or
- Multi layer networks, when the nodes are arranged in more than one layer.

Moreover, a network can be characterized as:

- A feedforward network, if there is no feedback to previous layers from the output of the subsequent layers.
- Recurrent network, if such a feedback connection exists.

In Fig. (2) a three-layer feedforward neural network is shown. The various signals present in each layer are:

$u = [u_1, u_2, \dots u_n]^T = n \times 1$ input vector

$v = [v_1, v_2, \dots v_p]^T = p \times 1$ vector, output of layer 1 (first hidden layer)

$z = [z_1, z_2, \dots z_q]^T = q \times 1$ vector, output of layer 2 (second hidden layer)

$y = [y_1, y_2, \ldots y_m]^T = m \times 1$ vector, output of the net

$\bar{v} = [\bar{v}_1, \bar{v}_2, \ldots \bar{v}_p]^T = p \times 1$ vector , intermediate output of layer 1 (first hidden layer)

$\bar{z} = [\bar{z}_1, \bar{z}_2, \ldots \bar{z}_q]^T = q \times 1$ vector , intermediate output of layer 2 (second hidden layer)

$\bar{y} = [\bar{y}_1, \bar{y}_2, \ldots \bar{y}_m] = m \times 1$ vector , intermediate output of layer 3 (output layer)

$W^1 = \{w^1_{ij}\} = p \times n$ matrix of weights connecting the j-th input to the i-th node of layer 1

$W^2 = \{w^2_{ij}\} = q \times p$ matrix of weights connecting the j-th node of layer 1 to the i-th node of layer 2

$W^3 = \{w^3_{ij}\} = m \times q$ matrix of weights connecting the j-th node of layer 2 to the i-th node of layer 3

$\gamma(x)$ is the differentiable nonlinearity

$$\gamma(x) = (1 - e^{-\alpha x})/(1 + e^{-\alpha x}) \; ; \; \alpha > 0$$

The various inputs and outputs in each layer are related by the following equations:

Layer 1: $\bar{v} = W^1 u$; $v_i = \gamma(\bar{v}_i)$, $i = 1, \ldots p$

Layer 2: $\bar{z} = W^2 v$; $z_i = \gamma(\bar{z}_i)$, $i = 1, \ldots q$

Layer 3: $\bar{y} = W^3 z$; $y_i = \gamma(\bar{y}_i)$, $i = 1, \ldots m$

It is seen that the output vector y is a nonlinear function of the input vector u and of the weights $\{w_{ij}\}$. It is precisely this nonlinear dependence that makes the neural net a powerful tool for approximating arbitrary functions, hence a potential tool for system identification.

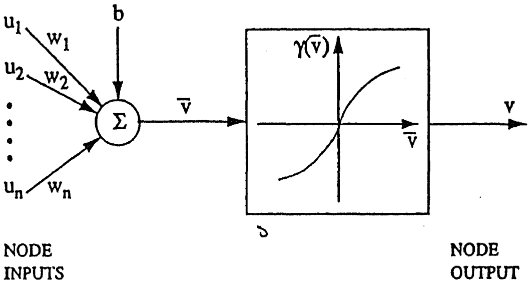

Fig. 1: A processing element.

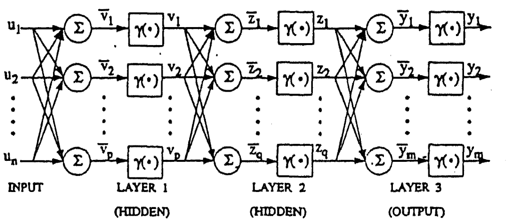

Fig. 2: A feedforward three-layer network.

2.4 *The Training Algorithms*

The term training or learning algorithm refers to a systematic procedure for adjusting the weights in the network in order to achieve a desired input/output relationship. In the case of "supervised" learning the network is being presented pairs of input vectors and desired output vectors (u^r, y^r_d), where the superscript r ranges over all pairs used to train the network and the subscript d stands for "desired" output vector. During training the network learns to associate the input vector u^r with the output vector y^r_d. For a given set of weights, if the network is presented an input u^r it will produce an output y^r, which should be identical or very close to y^r_d if the training was successful.

A general feature of "supervised" learning algorithms is that a performance criterion $E = f(y, y_d)$ is evaluated, where y is the actual network output vector and y_d is the desired output vector. Then the weights w_{ij} of the net are adjusted in such a way as to reduce the value of this error criterion. The various learning algorithms reported in the literature differ in the ways this adjustment is being done.

As an example, the back propagation training algorithm uses a square error criterion defined as:

$$E = \frac{1}{2} \sum_r \sum_{k=1}^{m} (y^r_k - y^r_{kd})^2$$

where the index r ranges over all training patterns.

The performance criterion is a measure of the distance between the output vectors y^r and the desired outputs y^r_d. The criterion E is a function of the actual output values y^r_k, which in turn depend on the values of the network weights $\{w_{ij}\}$ and the input patterns. Because of this dependence of y^r_k on $\{w_{ij}\}$, the criterion E can be reduced if the weights w_{ij} are adjusted appropriately. In order to find a set of parameters (weights) that will reduce E, we need to calculate the gradient of E with respect to the parameters $\{w_{ij}\}$ and then adjust the parameter set in the direction of the negative gradient. The back propagation algorithm calculates this gradient, i.e. the partial derivatives $\partial E/\partial w_{ij}$ of the criterion E with respect to the elements w_{ij} of the weight matrices. The algorithm starts at the output layer and "propagates" the results backwards to the first layer. After the gradient matrices have been computed, the weights are adjusted in the negative gradient direction, so that the error criterion E is reduced. A typical weight w_{ij} (which could belong to any layer) is adjusted from its old value w_{ij}^{old} to its new value w_{ij}^{new} according to

$$w_{ij}^{new} = w_{ij}^{old} - \eta \frac{\partial E}{\partial w_{ij}^{old}}$$

The stepsize η is termed the "learning rate". This η is usually chosen as constant during training, but it could also be adjustable if this facilitates the training process.

Detailed descriptions of the back propagation algorithm can be found in Narendra and Parthasarathy (1990), where a graphical representation of the algorithm is given, and in McLelland and Rumelhart (1986) for an analytical formulation.

3. APPLICATION

3.1 *System Description*

A methodology based on a neural network formulation is developed and applied to identify a structural chain-like system undergoing deterministic as well as random excitation.

Let the N-degree of freedom chain-like system obey the following equation of motion:

$$M\ddot{y}(t) + g(y(t), \dot{y}(t)) = -m\ddot{s}(t) \qquad (1)$$

where $\ddot{s}(t)$ is the base acceleration vector, y is the vector of relative displacements with respect to the base, $g(y, \dot{y})$ is the vector of restoring forces, $M = \text{diag}\{m_1, m_2, \ldots, m_N\}$ is the mass matrix and $m = [m_1, m_2, \ldots, m_N]^T$.

This formulation represents general classes of structural systems, such as: systems with linear force/deflection characteristics, nonlinear systems having polynomial-form nonlinearities etc. Assume that the experimental measurements for $\ddot{s}(t)$ and $\ddot{y}(t)$ are available and that the corresponding relative displacement $y(t)$ and velocity $\dot{y}(t)$ vectors can be found by direct measurements or through integration of $\ddot{y}(t)$. If the measurements are taken at discrete times t_k, then the following notation is used:

$$y_k = y(t_k); \quad \dot{y}_k = \dot{y}(t_k); \quad \ddot{y}_k = \ddot{y}(t_k); \quad \ddot{s}_k = \ddot{s}(t_k)$$

It is assumed that the masses m_i, $i = 1, \ldots N$ are known, or can be accurately estimated, but the restoring force vector $g(y, \dot{y})$ is unknown. Then the neural network is trained to identify a model of $g(y, \dot{y})$. The system identification is done on the basis of input/output data, where inputs are the measured values of $y_k, \dot{y}_k$ and outputs are the calculated values of $g_k = (-M\ddot{y}_k - m\ddot{s}_k)$. During the training phase, the neural network is presented repeatedly with the sequence of input vectors $\{[y_k, \dot{y}_k]^T\}$ and the sequence $\{g_k\}$ of desired output vectors. The training algorithm adjusts the weights of the network in such a way as to reduce the error between the desired output g_k and the actual output of the network $\hat{g}_k$.

After the training procedure is completed, the network is validated as the identifier of system (1): given an *arbitrary* input vector $[y_\alpha, \dot{y}_\alpha]^T$, at time $t = \alpha$, the net should produce an output $\hat{g}_\alpha$ which is very close to the true value of the restoring force vector $g_\alpha = -M\ddot{y}_\alpha - m\ddot{s}_\alpha$.

3.2 *Simulation Results*

To test the validity of this procedure, simulations were performed on a 3-degree-of-freedom system with linear force-deflection characteristics, under stochastic as well as deterministic excitation. A three-layer feed-forward neural network is trained to identify the unknown system and then its modeling capabilities are validated. Training is done by the back propagation algorithm. The network has 6 input nodes (corresponding to the components of the displacement and velocity vectors $y, \dot{y}$), 3 output nodes (corresponding to the components of the vector $g(y, \dot{y})$ and 15 and 10 nodes in the first and second hidden layers, respectively.

During the training phase, system (1) is excited by a swept sine excitation and the structural response is used to obtain the input sequence to the neural net $\{[y_k, \dot{y}_k]^T\}$ and the output sequence $\{g_k\} = \{-M\ddot{y}_k - m\ddot{s}_k\}$. Figs. 3a-3c, show a comparison of the neural network output $\hat{g}_k$ and the actual system output g_k, after training is completed. It is seen that the network is performing very well in matching the system's response.

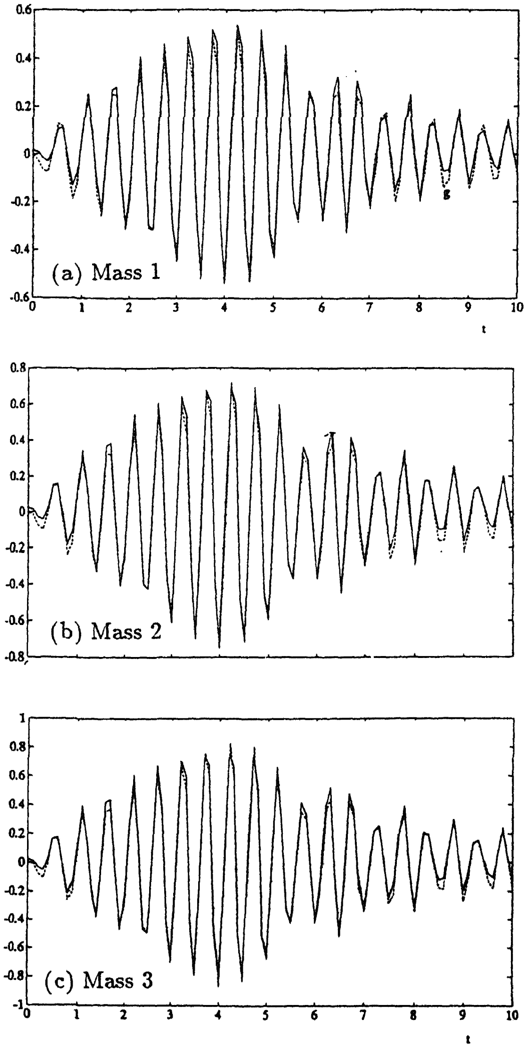

Fig. 3: Identification results;
Solid line: neural net $\hat{g}(y,\dot{y})$
Dashed line: actual system $g(y,\dot{y})$

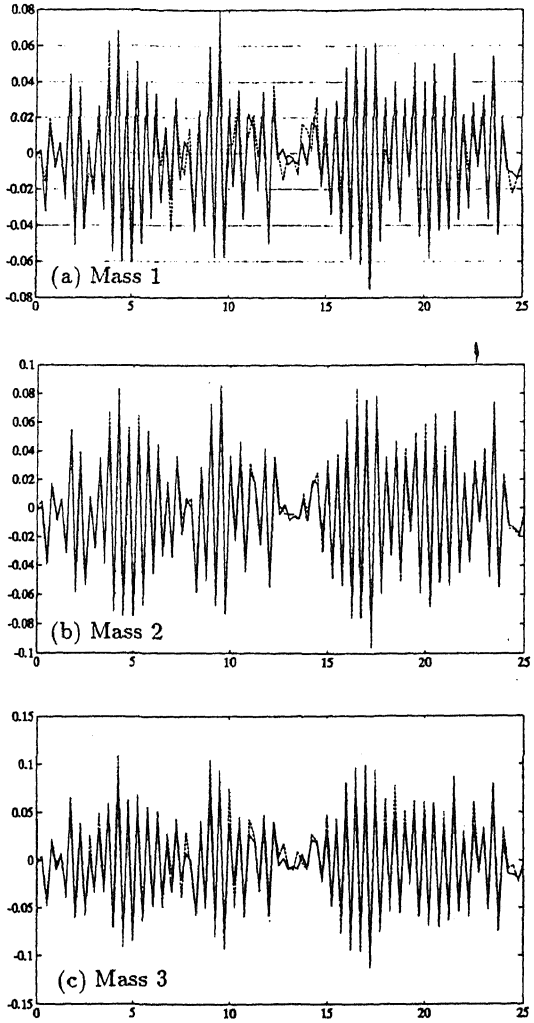

Fig. 4: Validation results;
Solid line: neural net $\hat{g}(y,\dot{y})$
Dashed line: actual system $g(y,\dot{y})$

During the next phase, the neural network is validated as a system identifier. System (1) is excited by a random excitation and the responses $\{y_k, \dot{y}_k\}$ are subsequently used as inputs to the already trained neural network. In Figs. 4a-4c, the outputs $\{\hat{g}_k\}$ of the network are compared to the actual system outputs $\{g_k\}$. It is seen that the neural net performs very well even when given random inputs, on which it has never been trained.

The present paper dealt with with the application of neural networks to a linear structural system. Details on the application of neural networks to nonlinear dynamic systems are available in Masri et al. (1992a, 1992b).

4. CONCLUSIONS

This exploratory study demonstrates the potential of using neural networks for the identification of the internal (restoring) forces of structural systems undergoing deterministic or stochastic excitations. It is shown that employing a three-layer feed-forward net is adequate to characterize the internal forces in a linear three-degree-of-freedom chain-like system. The neural net is validated as a system identifier and is able to predict the system's response to random excitations.

REFERENCES

Beck, J.L., (1978), "Determining Models of Structures from Earthquake Records," Earthquake Engineering Research Laboratory, California Institute of Technology, Pasadena.

Chen, J., (Ed.), (1988), Proc. USAF/NASA Workshop on Model Determination for Large Space Systems, Jet Propulsion Laboratory, California Institute of Technology, Pasadena, March.

Garba, J., (Ed.), (1990), Proc. USAF/NASA Workshop on System Identification and Health Monitoring, Jet Propulsion Laboratory, California Institute of Technology, Pasadena, March.

Housner G.W. & S.F. Masri (eds.) 1990. *Proceedings of the U.S. National Workshop on Structural Control Research*, University of Southern California, Publication CE-9013.

Ibanez, P., (1979), "Review of Analytical and Experimental Techniques for Improving Structural Dynamic Models," Welding Research Council Bulletin, No.249, June.

International Modal Analysis Conference, (1992), Proceedings, Union College, Schenectedy, New York.

Lippmann R.P. 1987. An Introduction to Computing with Neural Nets. *IEEE ASSP Mag.*, April, pp.4-22 (1987).

Masri, S.F. & T.K. Caughey, (1979), "A Nonparametric Identification Technique for Nonlinear Systems", *Journal of Applied Mechanics*, Vol. 46, No. 2, June, pp. 433-447, Trans. ASME.

Masri, S.F & S.D. Werner, (1985), "An Evaluation of a Class of Practical Optimization Techniques for Structural Dynamics Applications," *Earthquake Engineering and Structural Dynamics*, Vol. 13, pp. 635–649.

Masri S.F., A.G. Chassiakos & T.K. Caughey, (1992a). Structure-unknown non-linear dynamic systems: identification through neural networks. *Smart Materials and Structures* Vol. 1, 1, March.

Masri S.F., A.G. Chassiakos & T.K. Caughey (1992b). Identification of nonlinear dynamic systems using neural networks. To appear in the*ASME Transaction on Applied Mechanics.*

McLelland J.L. & D.E. Rumelhart 1986. *Parallel Distributed Processing*, Vol.1, MIT Press.

Narendra K.S. & K. Parthasarathy 1990. Identification and Control of Dynamical Systems Using Neural Networks. *IEEE Transactions on Neural Networks*, Vol.1, 1, March, pp.4-27.

Natke, H.G., (Ed.), (1982), "Identification of Vibrating Structures," Springer-Verlag.

Natke, H.G. & J.T.P Yao, (Eds.), (1988), Proc. of the Workshop on Structural Safety Evaluation Based on System Identification Approaches, Vieweg & Sons, Wiesbaden.

Nguyen D.H. & B. Widrow 1990. Neural Networks for Self Learning Control Systems. *IEEE Control Systems Magazine*, Vol.10,3, April, pp. 18-23.