AUGMENTED IFN LEARNING MODEL

By Shih-Lin Hung,[1] Member, ASCE, and J. C. Jan[2]

ABSTRACT: Solving engineering problems is a creative, experiential process. An experienced engineer generally solves a new problem by recalling and reusing some similar instances examined before. According to such a method, the integrated fuzzy neural network (IFN) learning model was developed and implemented as a computational model for problem solving. This model has been applied to design problems involving a complicated steel structure. Computational results indicate that, because of its simplicity, the IFN model can learn the complicated problems within a reasonable computational time. The learning performance of IFN, however, relies heavily on the values of some working parameters, selected on a trial-and-error basis. In this work, we present an augmented IFN learning model by integrating a conventional IFN learning model with two novel approaches—a correlation analysis in statistics and a self-adjustment in mathematical optimization. This is done to facilitate the search for appropriate working parameters in the conventional IFN. The augmented IFN is compared with the conventional IFN using two steel structure design examples. This comparison reveals a superior learning performance for the augmented IFN learning model. Also, the problem of arbitrary trial-and-error selection of the working parameters is avoided in the augmented IFN learning model.

INTRODUCTION

Solving engineering problems—such as those of analysis and design—is a creative, experiential process in which the experiences and combined knowledge of engineers serve as resources. An experienced engineer generally solves a new problem in the following stages. First, he or she recalls instances that were similar but produced resolution, while properly considering the functional requirements of those instances. Then, the engineer attempts to derive the solution from these similar instances through adaptation or synthesis. After the problem is solved, the new instance is then stored in her/his memory as an additional knowledge resource for solving future problems.

The described stages can be implemented as a computational model for problem solving, one that utilizes a case base of previously solved problems when solving a new one. In symbolic artificial intelligence (AI), case-based reasoning (Carbonell 1981) is an effective means of facilitating computer program development. It attempts to solve problems by directly accessing the case base. This approach relies on the explicit symbolic representation of a case base established from experience. With a given case base, case-based reasoning uses a representation involving specific episodes of problem solving, not only to solve a new problem, but also to learn *how* to solve the new problem. Based on the approach of case-based reasoning, the way to solve engineering problems has received considerable attention (Maher et al. 1995).

Artificial neural networks (ANNs), on the other hand, constitute a different AI approach, one that has made rapid advances in recent years. Such networks have the ability to develop, from training instances, their own solutions to a class of problems. The method of representation used by ANNs, essentially a continuous function, is conducive to generalization beyond the original set of training instances used in their development. The feasibility of applying ANN computing to engineering problems has received increasing interest, particularly on supervised neural networks with the back-propagation (BP, Rumelhart et al. 1986) learning algorithm. Vanluchene and Sun (1990) applied the back-propagation learning algorithm to structural engineering. Gunaratnam and Gero (1994) discussed the effect of the representation of input/output pairs on the learning performance of a BP neural network. Also, several other researchers have applied neural networks to engineering design and related problems (Hajela and Berke 1991; Ghaboussi et al. 1991; Kang and Yoon 1994; Stephen and Vanluchene 1994; Elkordy et al. 1994).

The learning processes of back-propagation (BP) supervised neural network learning models, however, always take a long time. Therefore, several different approaches have been developed to enhance the learning performance of the BP learning algorithm. Such an approach is to develop parallel learning algorithms on multiprocessor computers to reduce the overall computational time (Hung and Adeli 1991b, 1992, 1994b; Adeli and Hung 1993a). Another approach is to develop more effective neural network learning algorithms to reduce learning cycles (Adeli and Hung, 1993a, 1994a; Hung and Lin, 1994). A third approach involves the development of a hybrid learning algorithm—for instance, by integrating a genetic algorithm with neural network algorithms to improve the overall learning performance (Hung and Adeli 1991b, 1994b).

Another category of learning in ANN includes unsupervised neural network learning models that are generally used in classification problems (Carpenter and Grossberg 1988; Adeli and Hung 1993b). In structural engineering, Adeli and Park (1995) employed a counterpropagation neural network (CPN), which combines supervised and unsupervised neural networks, to solve complicated engineering design problems. That investigation concluded that a CPN learning model can learn how to solve complicated structural design problems within a reasonable computation time. Recently, authors Hung and Jan (1997, 1999) presented an integrated fuzzy neural network (IFN) learning model in structural engineering. The IFN learning model combined a novel unsupervised fuzzy neural network (UFN) reasoning model with a supervised neural network learning model using the adaptive L-BFGS learning algorithm (Hung and Lin 1994). The IFN learning model was applied to steel beam design problems. That work contended that the IFN learning model is a robust and effective ANN learning model. In addition, the IFN model can interpret a large number of instances for a complicated engineering problem within a reasonable computational time, owing to its simplicity in computation. However, the performance of the IFN learning model is heavily affected by some working parameters that are problem dependent and obtained via trial and error.

In this work, we present a more effective neural network

[1]Assoc. Prof., Dept. of Civ. Engrg., National Chiao Tung Univ., 1001 Ta Hsueh Rd., Hsinchu, Taiwan 300, R.O.C.

[2]PhD Candidate, Dept. of Civ. Engrg., National Chiao Tung Univ., 1001 Ta Hsueh Rd., Hsinchu, Taiwan 300, R.O.C.

Note. Editor: Sivand Lakmazaheri. Discussion open until June 1, 2000. To extend the closing date one month, a written request must be filed with the ASCE Manager of Journals. The manuscript for this paper was submitted for review and possible publication on June 1, 1999. This paper is part of the ***Journal of Computing in Civil Engineering***, Vol. 14, No. 1, January, 2000. ©ASCE, ISSN 0887-3801/00/0001-0015–0022/$8.00 + $.50 per page. Paper No. 21139.

J. Comput. Civ. Eng. 2000.14:15-22.

learning model, called the augmented IFN learning model, by integrating a conventional IFN learning model with two newly developed approaches. The first approach, correlation analysis in statistics, is employed to assist users in determining the appropriate working parameter to be used in the fuzzy membership function. The second approach, self-adjustment in mathematical optimization, is used to obtain appropriate weights, systematically, for each decision variable required in the input of training instances. The novel model is implemented in C language on a DEC3000/600 workstation. The augmented IFN learning model proposed herein is applied to two structural engineering problems to verify its learning performance. The first example is a steel beam design problem. The second example involves the preliminary design of steel structure buildings. The two examples are also used to train a conventional IFN learning model, for the sake of comparison.

REVIEW OF IFN LEARNING MODEL

This section briefly reviews the integrated fuzzy neural network (IFN) learning model (Hung and Jan 1997, 1999). The IFN learning model combines two sub-ANN learning models. One is a novel, unsupervised fuzzy neural network (UFN) reasoning model: a single-layered laterally connected network with an unsupervised competing learning algorithm. The other is an offline assistant model: a supervised neural network learning model with the adaptive L-BFGS learning algorithm (Hung and Lin 1994). The IFN learning model is schematically depicted in Fig. 1.

Assume that **U** is an associated instance base, with N solved

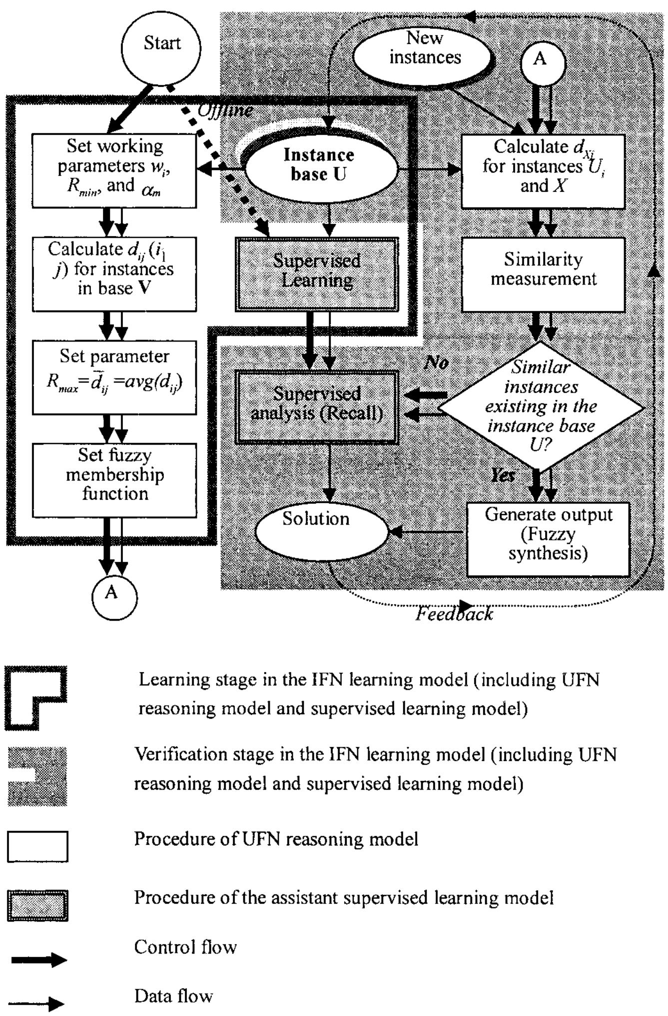

FIG. 1. Conventional IFN Learning Model

instances $U_1, U_2, \ldots, U_N$ and X is a new instance. Instance U_j is defined as a pair, including input $U_{j,i}$ and its corresponding output $U_{j,o}$. If there are M decision variables in the input and K items of data in the output, the input $U_{j,i}$ and output $U_{j,o}$ of instance U_j are represented as vectors of the decision variables and data, and are denoted as $U_{j,i} = \{u_j^1, u_j^2, \ldots, u_j^M\}$ and $U_{j,o} = \{o_j^1, o_j^2, \ldots, o_j^K\}$. Similarly, the new instance X can also be defined as a pair including input X_i and unsolved output X_o, respectively. The input X_i is a set of decision variables as $X_i = \{x^1, x^2, \ldots, x^M\}$. The output X_o is currently a null vector.

The learning stage in the IFN model is performed in two sub-ANN models concurrently. First, the offline assistant supervised neural network model is trained, based on the adaptive L-BFGS supervised learning algorithm using these N given instances. In the UFN reasoning model, however, the learning process simply involves selecting appropriate working parameters for the fuzzy membership function and weights for each decision variable in the input. The process is implemented in the following steps. The first step attempts to determine the degree of difference between any two distinct instances in base **V**, which contains P instances randomly selected from instance base **U**. Therefore, a total of $T = C_2^P = P(P-1)/2$ degrees of difference must be computed. The function of degree of difference, $diff(U_{i,i}, U_{j,i})$, is employed to measure the difference, d_{ij}, of two inputs $U_{i,i}$ and $U_{j,i}$ for instances U_i and U_j in **V**. The function is defined as the modified square of Euclidean distance and represented as

$$d_{ij} = diff(U_{i,i}, U_{j,i}) = \sum_{m=1}^{M} \alpha_m (w_i u_i^m - w_j u_j^m)^2 \qquad (1)$$

where w_i and α_m denote predefined weights and are used to represent the degree of importance for the ith instance in the instance base for the mth decision variable in the input. The weights w_i are generally set as constant one. The weights α_m, however, are set by trial and error. After the values of d_{ij} for all instances in base **V** are computed, the average of the sum of d_{ij}, denoted as $\bar{d}_{ij} = avg(d_{ij}) = \sum_{t=1}^{T}(d_{ij})/T$, can be computed.

The second step entails determining the fuzzy membership function. The relationship of "similarity" between any two instances is represented using a fuzzy membership function. In the UFN reasoning model, a quasi-Z-type membership function is used and defined as

$$\mu(d_{ij}) = f(R_{\max}, R_{\min}, d_{ij}) = \begin{cases} 0 & \text{if } d_{ij} \ge R_{\max} \\ \dfrac{R_{\max}R_{\min} - R_{\min}d_{ij}}{(R_{\max} - R_{\min})d_{ij}} & \text{if } R_{\min} < d_{ij} < R_{\max} \\ 1 & \text{if } d_{ij} \le R_{\min} \end{cases} \qquad (2)$$

The terms $R_{\max}$ and $R_{\min}$ are two working parameters that define the upper and lower bounds of "degree of difference." The lower bound $R_{\min}$ is set as a constant 10^{-5}. The upper bound, however, is set as $R_{\max} = \eta \bar{d}_{ij}$. The term η is a real number between zero and one and it was set by trial and error. Consequently, the degree of similarity for instances U_i and U_j can be determined from the fuzzy membership function.

After learning in the UFN and in the assistant supervised learning model is completed, any new instance X can be solved via the IFN learning method. The reasoning in UFN is performed through a single-layered, laterally connected network with an unsupervised competing algorithm, and it is implemented in three steps. The first step involves searching for some instances from the instance base **U** that resemble the new instance X according to their inputs; that is, the degree of difference, d_{Xj}, between the inputs, X_i and $U_{j,i}$, for instance X and instance U_j in base **U** is calculated. The input X_i of instant X is presented to the first node and the input $U_{j,i}$ of instance U_j

is presented to the $(j + 1)$th node simultaneously. The degree of difference d_{Xj} is then computed using (1) as $diff(X_i, U_{j,i}) = \sum_{m=1}^{M} \alpha_m (w_x x^m - w_j u_j^m)^2$. The second step entails representing the relationships among the new instance and its similar instances, with d_{Xj} less than $R_{\max}$. The fuzzy set, $S_{\sup,X}$, of "similar to X" was then expressed as follows:

$$S_{\sup,X} = (\mu_1/S_1) + (\mu_2/S_2) + \cdots + (\mu_p/S_p) + \cdots \quad (3)$$

where S_p = pth similar instance to instance X; the term "+" denotes a union operator; and μ_p = corresponding fuzzy membership value.

The final step involves generating the output X_o vector of instance X by synthesizing the outputs of its similar instances according to their associated fuzzy membership values using center of gravity (COG) or mean of maximum (MOM) methods (Hung and Jan 1999). For the given unsolved instance X, assume that P similar instances are in fuzzy set $S_{\sup,x}$; they are classified into C distinct clusters according to their outputs. Then, the output X_o of instance X yielded via these two methods are defined, respectively, as follows:

$$X_o = \frac{\sum_{k=1}^{p} \mu_k S_{k,o}}{\sum_{k=1}^{p} \mu_k} \quad \text{(COG)} \quad (4)$$

$$X_o = \frac{\sum_{k=1}^{C} \mu_k^{\max} S_{\max,o}^{k}}{\sum_{k=1}^{C} \mu_k^{\max}} \quad \text{(MOM)} \quad (5)$$

where μ_k denotes the membership value for the kth similar instance in fuzzy set $S_{\sup,X}$. Correspondingly, $\mu_k^{\max}$ denotes the membership value for the most similar instance $S_{\max}^{k}$ in the kth cluster.

The reasoning process of the UFN depends on determining the degree of similarity among X and U_j. Consequently, no solution can be generated by the UFN reasoning model if the new instance entirely differs from all instances in the instance base, e.g., all d_{Xi} are greater than $R_{\max}$. In addition, using inappropriate working parameters would allow for the possibility that no similar instances can be derived. For the above issues, the undertrained adaptive L-BFGS supervised neural network is used as an assistant system to generate an approximate output for the new instance.

AUGMENTED IFN LEARNING MODEL

In the conventional IFN learning model, working parameters (such as w_j, α_m, $R_{\max}$, and $R_{\min}$) are selected subjectively by users, and generally, on a trial-and-error basis. Consequently, the learning performance is highly affected by these parameters, especially $R_{\max}$ and α_m. In this work, two novel approaches are employed for assisting the users to determine these parameters and weights systematically. One approach, correlation analysis in statistics, is used to determine the adequate $R_{\max}$ value in the fuzzy membership function. The other approach, self-adjustment based on mathematical optimization theory, is employed to find proper values for weights α_m.

Correlation Analysis for $R_{\max}$ in Fuzzy Membership Function

In conventional IFU, the similarity measurement between two instances heavily depends on the value of parameter $R_{\max}$. A small value of $R_{\max}$ implies that a strict similar relationship between instances is utilized. Consequently, most of the instances are sorted as dissimilar. As a result, few similar instances to the new instance can be found and, ordinarily, no solution can be generated via the UFN reasoning model. On the other hand, a loose similar relationship is adopted under the case of a larger $R_{\max}$. Accordingly, a large number of instances are taken to be "similiar instances" and the solution generated via these similar instances is inferior. Here, the linear correlation analysis in statistics is employed to facilitate the determination of appropriate value of $R_{\max}$ in the fuzzy membership function. The analysis is a process that aims to measure the strength of the association between two sets of variables that are assumed to be linearly related.

For the above instance base **U** with N instances, the correlation analysis in the fuzzy membership function is implemented in the following steps. The first step is to determine the degree of difference between any two instances in the base **U** using the aforementioned function of degree difference in (1). Hence, a total of $(C_2^N + N)$ resembling samples $S_{ij}(U_{i,o}, U_{j,o}, d_{ij})$ can be compiled. A resembling sample contains two instances' outputs ($U_{i,o}$ and $U_{j,o}$) and the corresponding degree of difference (d_{ij}). Thereafter, two arrays, A_t and B_t, can be assorted from resembling samples in the case of d_{ij} less than or equal to a prescribed value, say t. The elements in A_t and B_t are the first and second items, respectively, of these resembling samples. Next, the accumulative correlation coefficient, $Ac\_CORREL(A_t, B_t, t)$, is calculated for arrays A_t and B_t with the degree of difference less than or equal to t. Assume that for a total P resembling samples with d_{ij} less than a prescribed t, the arrays A_t and B_t can be denoted as

$$A_t = \{a_k | a_k = U_{i,o} \in S_{ij}, \quad \text{for} \quad d_{ij} \le t\} = \{a_1, a_2, \ldots, a_p\}$$

$$B_t = \{b_k | b_k = U_{j,o} \in S_{ij}, \quad \text{for} \quad d_{ij} \le t\} = \{b_1, b_2, \ldots, b_p\}$$

The value of the accumulative correlation coefficient equals

$$Ac\_CORREL(A_t, B_t, t) = \frac{\text{Cov}(A_t, B_t, t)}{\sigma_{A_t}\sigma_{B_t}} \quad (6)$$

$$\text{Cov}(A_t, B_t, t) = \frac{1}{p} \sum_{k=1}^{p} (a_k - \mu_{A_t})(b_k - \mu_{B_t}) \quad \text{s.t.} \quad d_{ij} \le t \quad (7)$$

where σ_{A_t} and σ_{B_t} are standard errors of arrays A_t and B_t; μ_{A_t} and μ_{B_t} are the means of A_t and B_t. The formulas expressed in (6) and (7) represent the relationship between the accumulative correlation coefficient to any value of t. An accumulative correlation curve can be plotted as a function of t and $Ac\_CORREL(A_t, B_t, t)$. Fig. 2 demonstrates an accumulative correlation curve for two arrays $A_t = \sin(x)$ and $B_t = 0.95(x - x^3/3! + x^5/5!)$ such that $x = -\pi + (\pi/15)i$, $i = 0$ to 30. Shown in Fig. 2, the curve falls from one to zero as the value of t increases, and it resembles the quasi-Z-type fuzzy membership function defined in (2) for the case of $R_{\min}$ equal to zero. Note that the

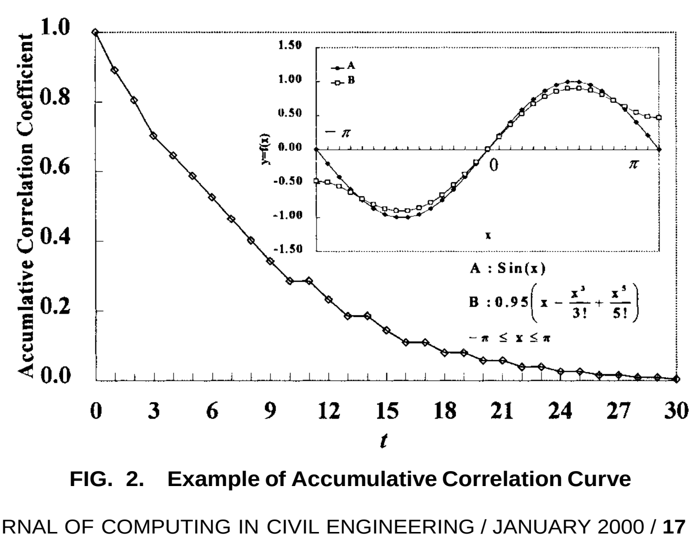

FIG. 2. Example of Accumulative Correlation Curve

J. Comput. Civ. Eng. 2000.14:15-22.

appropriate $R_{\max}$ equals a certain value of t, such that instances in the instance base **U** have a certain degree of correlation.

Obviously, the smaller the t implies a larger accumulative correlation coefficient, indicating a strong relationship between the two arrays, e.g., the strongest correlation, $t = 0$, between the two arrays refers to the case in which the instances in the two sets are identical and the value of $Ac\_CORREL(A_t, B_t, t)$ equals one. In such a case, no solution to a new instance can be generated via the UFN reasoning model except for when identical instances exist in the instance base. In order to avoid this issue here, we set $Ac\_CORREL(A_t, B_t, t)$ equal to 0.8 as the lower bound for similarity measurement. The value of t corresponding to this lower bound is adopted as the appropriate value of $R_{\max}$.

Self-Adjustment Approach for Selecting Weights α_m

Except for $R_{\max}$, the selected weights α_m also significantly affect the learning performance for the conventional IFN. This occurrence has been investigated in the earlier work (Hung and Jan 1999). The learning results indicated that significant improvements were achieved as the weights were gradually updated via a basis of heuristic knowledge associated with learning problems. In this study, a more systematical approach —self-adjustment based on mathematical optimization—is adopted to facilitate the search for appropriate weights.

For the above instance base **U** with N instances, the self-adjustment approach can be briefly stated as consisting of the following steps. First, set up the corresponding working parameters, $R_{\max}$, $R_{\min}$, and w_j, where $R_{\max}$ is determined using the aforementioned correlation analysis approach and where parameter $R_{\min}$ and weights w_j are set as constants in this work. Meanwhile, weights α_m for each decision variable in the input are directly initialized as one. Then, based on these working parameters, the outputs for training instances are found via the UFN reasoning model. Then the error, E_i, between the computed and desired outputs, Y and $U_{i,o}$, for training instance U_i, is calculated. The system error, E, for a total N instances is then defined as half of the average sum of errors and denoted as

$$E = \frac{1}{2N}\sum_{i=1}^{N} E_i \tag{8}$$

$$E_i = \sum_{k=1}^{K} (y^k - o_i^k)^2 \tag{9}$$

where o_i^k and y^k = kth items of data in desired and computed outputs, U_i and Y, respectively. Note that the system error is an implicit function of the weights α_m as $E(\alpha_m)$.

Weights α_m in the UFN reasoning model are adjusted to reduce the system error as much as possible. This goal can be achieved if a set of appropriate weights, α_m, are used. The problem, then, can be considered as an unconstrained optimization problem, that is, searching a set of optimum weights by iteration to minimize the system error. In the mathematical optimization approaches, the conjugate gradient (CG) method has been proved an efficient means of solving the problem. The weights α_m are updated in each iteration, say the $(s + 1)th$ iteration, as $\alpha_m^{(s+1)} = \alpha_m^{(s)} + \lambda \mathbf{d}^{(s)}$. The term λ is step length and is set as a constant in this work. The search direction $\mathbf{d}^{(s)}$ is defined as

$$\begin{cases} \mathbf{d}^{(s)} = \mathbf{g}^{(s)} = -\nabla E(\alpha_m^{(s)}) & \text{if } s = 0 \\ \mathbf{d}^{(s)} = \mathbf{g}^{(s)} + \beta^{(s)}\mathbf{d}^{(s-1)} & \text{if } s > 0 \end{cases} \tag{10a,b}$$

where

$$\beta^{(s)} = \frac{(\mathbf{g}^{(s)})^T\mathbf{g}^{(s)}}{(\mathbf{g}^{(s-1)})^T\mathbf{g}^{(s-1)}}, \quad \text{and} \quad \mathbf{g}^{(s)} = -\nabla E(\alpha_m^{(s)})$$

The iteration is terminated as the value of $\|\mathbf{g}^{(s+1)}\|$ or $|E(\alpha_m^{(s+1)}) - E(\alpha_m^{(s)})|$ is sufficiently small. The term $\mathbf{g}^{(s)}$ is the negative gradient vector of function $E(\alpha_m)$. For simplicity, the superscript (s), denoted as the sth iteration, is ignored. Hereinafter, vector **g** is derived using chain rule and denoted as

$$\mathbf{g} = \frac{\partial E(\alpha_m)}{\partial \alpha_m} = \frac{1}{2N}\sum_{i=1}^{N}\frac{\partial E_i}{\partial \alpha_m} \tag{11}$$

$$\frac{\partial E_i}{\partial \alpha_m} = 2\sum_k (y^k - o_i^k)\left(\sum_p f(d_{ip})\right)^{-2} \times \left(\sum_p o_p^k f'(d_{ip})(u_i^m - u_p^m)^2 \sum_p f(d_{ip}) - \sum_p f(d_{ip}) o_p^k \sum_p f'(d_{ip})(u_i^m - u_p^m)^2\right) \tag{12}$$

where index p denotes the pth similar instance in fuzzy set $S_{\sup, U_i}$, and indices m and k denote the mth decision variable in the input and the kth data item in the output vectors, respectively.

Augmented IFN Learning Model

In this section, we present an augmented IFN learning model. Fig. 3 schematically depicts the procedures of the augmented IFN learning model. Instead of using a constant working parameter $R_{\max}$, as in conventional IFN, the appropriate parameter is determined using correlation analysis. Meanwhile, the appropriate weights α_m are adapted during the self-adjustment process through a mathematical approach. These two approaches, called self-organized learning, are used to enhance the learning capability of the conventional IFN learning model. The procedure of the augmented IFN learning model can be summarized as follows:

- *Self-organized learning phase*

  Step 0. Train the adaptive supervised L-BFGS learning neural network model *offline*.

  Step 1. Initialize parameter $R_{\min}$ as constant 10^{-5} and

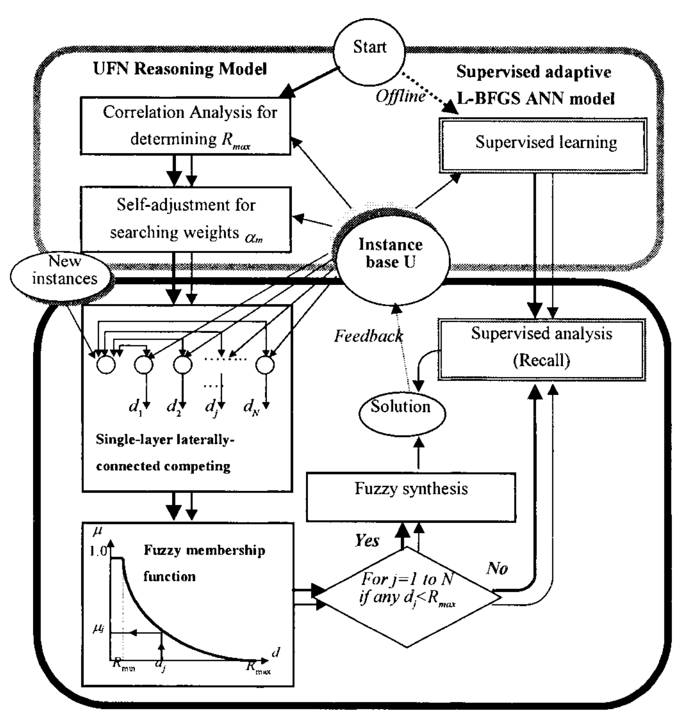

FIG. 3. Augmented IFN Learning Model

J. Comput. Civ. Eng. 2000.14:15-22.

weight α_m for each decision variable, on a heuristic basis or by trial and error.

Step 2. Calculate the degree of difference $d_{ij}(i \neq j)$ among all instances in base **V**.

Step 3. Determine the parameter $R_{\max}$ using correlation analysis and set $R_{\max} = t$ such that $Ac\_CORREL(A_t, B_t, t) = 0.8$.

Step 4. Set the fuzzy membership function, $\mu(d_{ij}) = f(R_{\min}, R_{\max}, d_{ij})$, defined in (2).

Step 5. Adjust the weight α_m for each decision variable in input, using the self-adjustment approach.

- *Analysis phase* (after learning phase is completed)

Step 6. Present the new (unsolved) instance X to the UFN reasoning model, and perform a similarity measurement between X and instance U_i in the base **U** using a single-layered, lateral-connected competing network.

Step 7. If more than one similar instance is found in Step 6, generate the solution X_o for the new instance using fuzzy synthesis approaches in (4) or (5), and go to Step 9. Otherwise go to next step.

Step 8. Compute the solutions via the undertrained assistant adaptive L-BFGS supervised learning model and go to next step.

Step 9. Feedback the new instance into the base **U**. Meanwhile, further learning in the assistant supervised learning model is launched offline.

APPLICATIONS

Here the novel augmented IFN learning model is applied to problems of engineering design. Two structural design examples are presented to verify the learning performance of the augmented IFN learning model. The first example is a steel beam design problem. The second is the preliminary design of steel structure buildings. The two examples are also used to train a conventional IFN learning model for comparison.

Steel Beam Design Problem

The problem is the design of the lightest W-section simply supported steel beam with compact section, under LRFD Specification (*Manual* 1994). The problem was studied in our earlier work (Hung and Jan 1999), with those results demonstrating that the conventional IFN learning model is superior to a stand-alone supervised neural network with L-BFGS learning algorithm when the number of training instances was large enough. This work investigates only the learning performances of the conventional IFN and the novel augmented IFN learning models.

The LRFD design method, applied to the effect of lateral torsional buckling for beam with compact section, is summarized in the following steps:

1. Determine the factored moment M_u at midspan of a given beam.
2. Select one potential member with plastic section modulus Z so that the required nominal moment $M_n = Zf_y$ satisfies $0.9M_n \geq M_u$, where f_y is the specified yield strength of the beam.
3. Estimate whether the lateral supports are close enough to design the beam using plastic analysis or whether we should use the fully plastic moment strength M_p without plastic analysis, for a given unbraced length L_b. The nominal moment M_n is then obtained according to the following formulas with different conditions:

   a. If $L_b < L_p$, then $M_n = M_p$

   b. If $L_p < L_b < L_r$, then M_n

   $$= C_b[M_p - (M_p - M_r)((L_b - L_r/L_r - L_p))] \leq M_p$$

   c. If $L_b > L_r$, then M_n

   $$= M_{cr} = C_b \frac{\pi}{L_b} \sqrt{\left(\frac{\pi E}{L_b}\right)^2 C_w I_y + EI_y GJ}$$

   where M_p = plastic moment strength; L_p and L_r = limited laterally unbraced length for full plastic bending capacity and for inelastic lateral-torsional buckling, respectively; C_b = moment gradient coefficient; M_r = limited buckling moment; E = elasticity modulus of steel; C_w = warping constant; I_y = moment of inertia about the minor axis; G = shear modulus of elasticity of steel; and J = torsional constant.
4. Finally, confirm whether or not the section of selected member satisfies the requirements of flexural design strength, $0.9M_n \geq M_u$, and shear design strength, $0.9V_n \geq V_u$, where V_n is the nominal shear strength.

The above steps are repeated until a satisfactory lightest member is obtained. Eight hundred instances, created according to the aforementioned design process, are used in the present example to train and verify the conventional and augmented IFN learning models. Of these, 600 are used as training instances and the remaining two hundred (200) are used as verification instances. Seven decision variables are used as inputs in order to determine the plastic modulus Z of a lightest W-section steel beam with compact section. The seven decision variables are the yielding strength of steel f_y, factored maximum bending moment M_u, the live load w_l, the factored maximum shear force V_u, the span of the beam L, the moment gradient coefficient C_b, and the unbraced length L_b. Notably, the first decision variable, f_y, was not used in the earlier work (Hung and Jan 1999), as the yielding strength of steel was identical for each instance.

The parameters $R_{\min}$ and w_i are set as constant 10^{-5} and one, respectively. The weights α_m, however, are initialized as [1, 9, 1, 1, 1, 1, 1] for the seven decision variables. Of these, the second weight, α_2, for decision variable, factored maximum bending moment M_u, is set based on the finding investigated in the earlier work (Hung and Jan 1999). Using these parameters and weights, the augmented IFN learning model is trained in four different cases with different numbers of training instances in base **V**. The number of training instances is increased from 100 to 400 with an increment of 100. After performing the correlation analysis in four different bases of **V**, the accumulative correlation coefficients can be computed with respect to any specified value t. With the value of $Ac\_CORREL(A_t, B_t, t)$ equal to 0.8, the values of $t(=R_{\max})$ for four different cases are obtained. Those values are 0.044, 0.034, 0.033, and 0.030, respectively, for 100, 200, 300, and 400 training instances in base **V**, and are displayed in Fig. 4. Interestingly, according to this figure, the greater the number of instances in bases **V** implies a higher correlation among these instances. Restated, the value of $R_{\max}$ decreases with an increase of the number of training instances. After the value of $R_{\max}$ is determined, the self-adjustment approach is launched to search the appropriate weights α_m for each decision variable in the input.

After self-adjustment is achieved, a set of weights α_m is obtained, e.g., the weights α_m are adjusted to values [1.42, 6.93, 1.02, 1.01, 1.01, 1.51, 0.73] for the case of **V** with 400 training instances. Note that the first, second, and sixth weights (α_1, α_2, and α_6) are changed by more than the other weights. These adjustments illustrate that the first and sixth decision

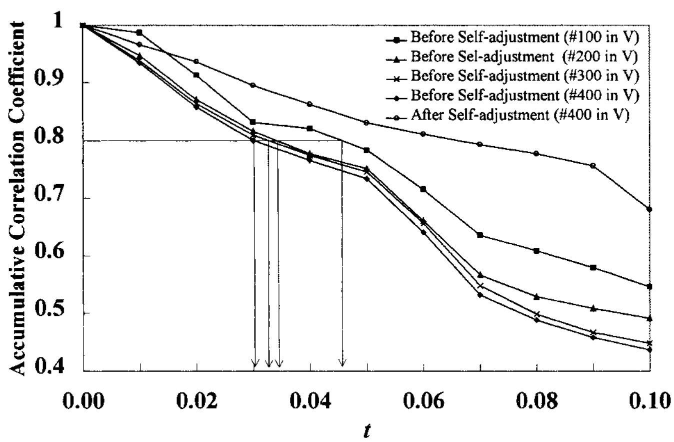

FIG. 4. Accumulative Correlation Curve for Steel Beam Design Example

variables, f_y and C_b, are more important than we would be led to believe by a process of trial and error. However, the weight α_2 for the second decision variable, M_u, is adjusted from 9 to 6.93. The occurrence reveals that the factored moment, M_u, is the most important decision variable in the input, as compared with other decision variables. Then, the 200 verification instances are used to verify the learning performance of the conventional and augmented IFN learning models. The verification in the augmented IFN is based on the newly adjusted weights α_m with the appropriate $R_{\max}$. The working parameters and weights used in the conventional IFN, however, are selected by trial and error. Fig. 5 compares the computed and desired outputs of the 200 verification instances for the augmented and conventional IFN learning models. The correlation coefficients for the computed and desired outputs are 0.997 and 0.992 for the augmented and conventional IFN learning model, respectively. Table 1 summarizes the computing results

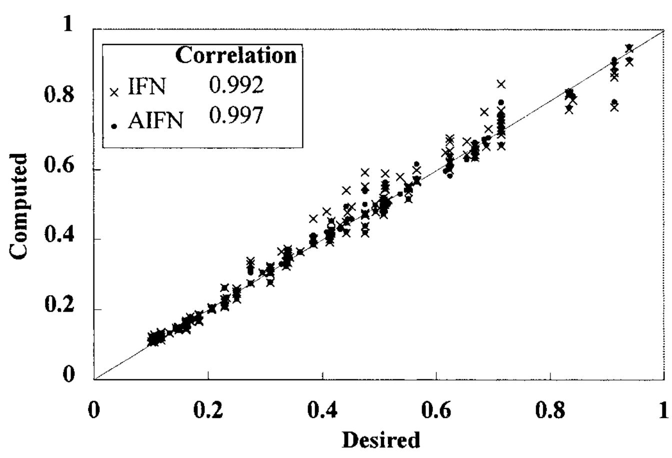

FIG. 5. Comparison of Computed and Desired Outputs for Verification Instances: Steel Beam Design Example

TABLE 1. Results of Steel Beam Design Problem

Number of training instances in **U**(**V**) (1)	Average Error ($R_{\max}$) for 200 Verification Instances: Conventional IFN learning model (2)	Average Error ($R_{\max}$) for 200 Verification Instances: Augmented IFN learning model (3)
100 (100)	9.33% (0.040)	8.31% (0.044)
200 (200)	8.31% (0.040)	5.50% (0.034)
300 (300)	5.57% (0.040)	4.47% (0.033)
400 (400)	4.92% (0.040)	3.82% (0.030)
600 (400)	4.47% (0.040)	3.47% (0.030)

for the 200 verification instances. According to this table, the learning performance is improved as the number of training instances increases for the augmented and conventional IFN learning models. This same table reveals that the average errors for verification instances are 3.47% and 4.47% for the augmented and conventional IFN learning models, respectively. In sum, the learning performance of the augmented IFN learning model is much better than that of the conventional IFN learning model in this example.

Based on the newly adjusted weights α_m, correlation analysis is next performed for the case of 400 instance bases in **V** again. According to those results, the accumulative correlation coefficients are higher for any specified value t than those obtained from the initial set of weights. For example, the accumulative correlation coefficient is from 0.8 to 0.9 at the value t equal to 0.03 (Fig. 4).

Preliminary Design of Steel Structural Buildings

In the complete design of a structure, the preliminary design stage is mainly a creative, experiential process that involves the choice of structure type, selection of the material, and determination of the sections of beams and columns in the structure. An experienced engineer is likely to carry out this stage more quickly than does an inexperienced one. The basic configuration of the structure at this stage should satisfy the specified design code, such as LRFD for steel structures. To satisfy the prescribed constrains and achieve minimum expenditure for materials and construction, this stage becomes a looped optimization decision-making process. Hence, a good initial development of the basic form, with sections of beams and columns satisfying the aforementioned constraints, will reduce the number of redesign cycles. After a basic structure is determined, the structural analysis stage involves analyzing the initial guessed structure and computing the service loads on the members. Also, the maximum lateral drift of the structure and the drifts between floors are computed if lateral loads are considered. The present example involves a complete design structure that satisfies the conditions that the service loads should not exceed the strength of the members; the drifts should be within the prescribed limits, and the structure should be economical in material (e.g., minimum weight), construction, and overall cost. In this example, the augmented IFN learning model is trained to learn how to implement the preliminary design of buildings satisfying the conditions of utility, safety, and economy in only one design cycle. For simplicity, only regular buildings with a rectangular layout—such as most factory buildings—are considered herein. Also, the beams in every floor have the same sectional dimensions, as do the columns.

In this example, 416 instances are used. They are randomly divided into 380 training instances and 36 verification instances. Seven decision variables are used as inputs to determine the sections of beams and columns of a building that satisfies the given specifications. The seven decision variables and their limits are described as follows:

1. Number of stories = [9, 10, 11, 12, 13, 14, 15]
2. Bay length in long-span direction (X direction) = 9 to 12 m
3. Bay length in short-span direction (Y direction) = 6 to 9 m
4. Number of bays in both directions = [3, 4, 5]
5. Seismic zone coefficient = [0.18, 0.23, 0.28, 0.33]
6. Live load (kgw/m^2) = 200 to 350
7. Wall load (kgw/m) = [100, 200]

Other corresponding decision variables used in the stage of preliminary design are assumed to be constant. In practice, the

J. Comput. Civ. Eng. 2000.14:15-22.

sections of beams and columns of a building are classified into certain groups for convenience in construction, instead of separate consideration being given to each element. Here, a building with three groups of steel elements in both beams and columns is considered. The three groups are upper, medium, and lower. For a building with N stories, these three groups are defined as upper group: floors from $2N/3$ to N; medium group: floors from $N/3$ to $2N/3$; and lower group: floors from one to $N/3$. An instance contains seven decision variable inputs and six data items as outputs.

The parameters R_{min} and w_i are set as constant 10^{-5} and one, respectively. The weights α_m, however, are initialized as [1, 1, 1, 1, 1, 1, 1] for the seven decision variables. Using these parameters and weights, correlation analysis in the augmented IFN learning model is performed first to determine the working parameter R_{max}. Fig. 6 displays the result of correlation analysis. With the value of $Ac\_CORREL(A_t, B_t, t)$ equal to 0.8, the values of $t(=R_{max})$ are obtained. They are 0.12 for beams and 0.178 for columns, respectively.

After the fuzzy membership function is defined, the self-adjustment approach is launched to obtain the adequate weights α_m for each decision variable in the input. The weights α_m for each decision variable are updated from [1, 1, 1, 1, 1, 1, 1] to [1.471, 1.369, 0.416, 0.008, 1.104, 0.825, 0.513] for beams and to [1.106, 0.997, 0.542, 0.009, 1.232, 0.999, 0.997] for columns. Interestingly, the weights of the fourth decision variable in beams and columns are both self-adjusted close to zero. This observation indicates that this decision variable (number of bays in both directions) is insignificant in the input. Consequently, this decision variable can be neglected. The 36 verification instances are used to verify the learning performance of the augmented and conventional IFN learning models, respectively. Notably, the augmented IFN is verified on the basis of the newly adjusted weights α_m with the adequate R_{max}. The working parameters and weights used in the conventional IFN, however, are selected on a trial-and-error basis. Fig. 7 depicts the correlation between the computed and desired outputs of beams for the 36 verification instances for the augmented as well as conventional IFN learning models. Similarly, Fig. 8 displays the correlation between the computed and desired outputs for columns for the thirty-six verification instances using the two IFN learning models. Table 2 summarizes the learning results for thirty-six verification instances. According to this table, the average percentage errors for beams and columns are 13.81 and 9.36 for the conventional IFN learning model. However, these errors are reduced to 6.17 and 6.1 for beams and columns, respectively, for the augmented IFN learning model. The augmented IFN learning model significantly improves in terms of learning. This example also illustrates that the augmented IFN learning model yields a substantially better learning performance than that of the conventional IFN learning model.

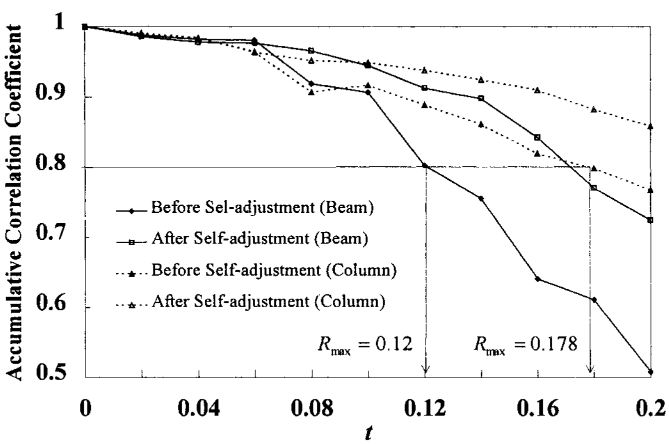

FIG. 6. Accumulative Correlation Curves for Example of Preliminary Design of Steel Structural Buildings

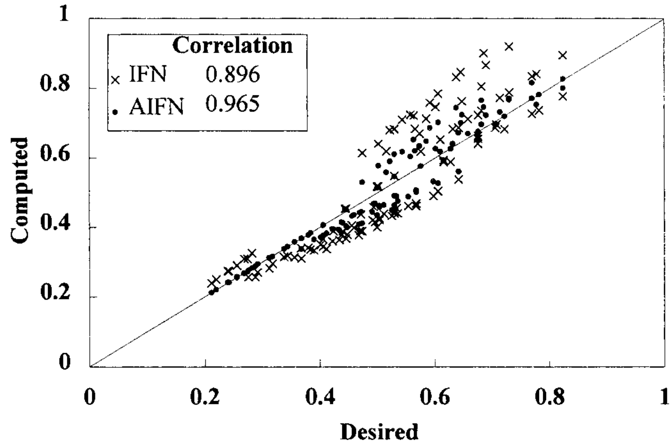

FIG. 7. Comparison of Computed and Desired Outputs (Beams) for Verification Instances: Preliminary Design of Steel Structural Buildings

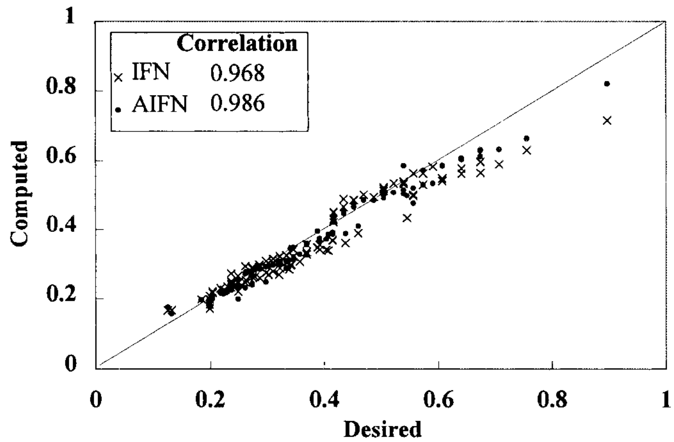

FIG. 8. Comparison of Computed and Desired Outputs (Columns) for Verification Instances: Preliminary Design of Steel Structural Buildings

TABLE 2. Results of Preliminary Structural Design Problem

Error (%) (1)	NUMBER OF ITEMS IN OUTPUT FOR VERIFICATION INSTANCES: Conventional IFN Learning Model — Beam (2)	Conventional IFN Learning Model — Column (3)	Augmented IFN Learning Model — Beam (4)	Augmented IFN Learning Model — Column (5)
0 ~ 5	12	43	52	51
5 ~ 10	29	29	39	31
10 ~ 15	21	22	13	20
15 ~ 20	29	10	3	6
>20	17	4	1	0
Average error for 36 verification instances	13.81%	9.36%	6.10%	6.17%

Note: Each instance has six items, three beams, and columns, in output.

CONCLUDING REMARKS

This work presents an augmented IFN learning model by integrating two newly developed approaches into a conventional IFN learning model. These approached are a correlation analysis in statistics and self-adjustment in mathematical op-

J. Comput. Civ. Eng. 2000.14:15-22.

timization, which collaboratively enhance the learning capability of the conventional IFN. The first approach, correlation analysis in statistics, assists users in determining the appropriate working parameter used in the fuzzy membership function. The second approach, self-adjustment in mathematical optimization, obtains appropriate weights, systematically, for each decision variable in the input of training instance. The augmented IFN learning model proposed herein is applied to engineering design problems. Two structural design problems were addressed to assess the learning performance of the augmented IFN learning model. Based on the results of this work, we can conclude the following:

1. The problem of arbitrary trial-and-error selection of the working parameter (R_{max}) in fuzzy membership function, encountered in the conventional IFN learning model, is avoided in the newly developed augmented IFN learning model. Instead of arbitrary R_{max}, the appropriate value of R_{max} is determined using correlation analysis in statistics. Thus, the new learning model provides a more solid systemic foundation for IFN learning than the conventional IFN learning model.
2. In the conventional IFN learning model, the weights α_m, denoting the importance of the *m*th decision variable in the input, are set on a trial-and-error basis. For complicated problems, the appropriate weights are difficult to obtain because of a lack of the relevant heuristic knowledge. In due course, the value is commonly initialized as one for most of the examples. This problem is avoided by the newly developed learning model. Instead of an assumed constant, the appropriate weights are determined through the self-adjustment approach in mathematical optimization. Therefore, the augmented IFN learning model provides a more solid mathematical foundation for neural network learning.
3. For each training instance, decision variables in the input are generally selected subjectively by users. As a result, some trivial decision variables may be adopted in the input for some complicated examples. Based on the self-adjustment approach, the importance of a decision variable in an input can be derived systematically. Therefore, insignificant or redundant decision variables in the input can be neglected.
4. The results illustrate that the value of the appropriate R_{max} gradually falls with an increase in the number of instances. Consequently, not only is the learning performance for training instances enhanced, but also the performance for verification instances is improved. Notably, a small value of R_{max} indicates that a strict similarity measurement is utilized in the UFN reasoning model and the possibility that no similar instances can be derived for any new instance also increases.

ACKNOWLEDGMENTS

The writers would like to thank the National Science Council of the Republic of China for financially supporting this research under Contract No. NSC 86-2221-E-009-070.

APPENDIX. REFERENCES

Adeli, H., and Hung, S. L. (1993a). ''A concurrent adaptive conjugate gradient learning on MIMD machines.'' *J. Supercomputer Appl.*, 7(2), 155–166.

Adeli, H., and Hung, S. L. (1993b). ''A fuzzy neural network learning model for image recognition.'' *Integrated Computer-Aided Engrg.*, 1(1), 43–55.

Adeli, H., and Hung, S. L. (1995). *Machine learning—neural networks, genetic algorithms, and fuzzy systems.* Wiley, New York.

Adeli, H., and Park, H. S. (1995). ''Counterpropagation neural network in structural engineering.'' *J. Struct. Engrg.*, ASCE, 121(8), 1205–1212.

Carbonell, J. G. (1981). ''A computational model of analogical problem solving.'' *Proc., 7th Int. Joint Conf. on Artificial Intelligence*, William Kaufmann, Los Altos, Calif., 1, 147–152.

Carpenter, G. A., and Grossberg, S. (1988). ''The ART of adaptive pattern recognition by a self-organizing neural network.'' *IEEE Comp.*, 21, 77–88.

Elkordy, M. F., Cheng, K. C., and Lee, G. C. (1994). ''A structural damage neural network monitoring system.'' *Microcomp. in Civ. Engrg.*, 9(2), 83–96.

Ghaboussi, J., Garrett, J. H., and Wu, X. (1991). ''Knowledge-based modeling of material behavior and neural networks.'' *J. Engrg. Mech.*, ASCE, 117(1), 132–153.

Gunaratnam, D. J., and Gero, J. S. (1994). ''Effect of representation on the performance of neural networks in structural engineering applications.'' *Microcomp. in Civ. Engrg.*, 9(2), 97–108.

Hajela, P., and Berke, L. (1991). ''Neurobiological computational modes in structural analysis and design.'' *Comp. and Struct.*, 41(4), 657–667.

Hung, S. L., and Adeli, H. (1991a). ''A model of perceptron learning with a hidden layer for engineering design.'' *Neurocomputing*, 3(1), 3–14.

Hung, S. L., and Adeli, H. (1991b). ''A hybrid learning algorithm for distributed memory multicomputers.'' *Heuristics—J. of Knowledge Engrg.*, 4(4), 58–68.

Hung, S. L., and Adeli, H. (1992). ''Parallel backpropagation learning algorithms on Cray Y-MP8/864 supercomputer.'' *Neurocomputing*, 5(6), 287–307.

Hung, S. L., and Adeli, H. (1994a). ''An adaptive conjugate gradient learning algorithm for effective training of multilayer neural networks.'' *Appl. Math. and Computat.*, 62(1).

Hung, S. L., and Adeli, H. (1994b). ''A parallel genetic/neural network learning algorithm for MIMD shared memory machines.'' *IEEE Trans. on Neural Networks*, 5(6), 900–909.

Hung, S. L., and Lin, Y. L. (1994). ''Application of an L-BFGS neural network learning algorithm in engineering analysis and design.'' *Proc., 2nd Nat. Conf. on Struct. Engrg.*, Vol. III, Chinese Society of Structural Engineering, Taipei, Taiwan, 221–230 (in Chinese).

Hung, S. L., and Jan, J. C. (1997). ''Machine learning in engineering design—an unsupervised fuzzy neural network learning model.'' *Proceedings of Intelligent Information Systems (IIS'97)*, IEEE Computer Society, California, 156–160.

Hung, S. L., and Jan, J. C. (1999). ''Machine learning in engineering analysis and design—an integrated fuzzy neural network learning model.'' *Computer-Aided Civ. and Infrastruct. Engrg.*, 14(3), 207–219.

Kang, H.-T., and Yoon, C. J. (1994). ''Neural network approaches to aid simple truss design problems.'' *Microcomp. in Civ. Engrg.*, 9(3), 211–218.

Kishi, N., Chen, W. F., and Goto, Y. (1997). ''Effective length factor of columns in semirigid and unbraced frames.'' *J. Struct. Engrg.*, ASCE, 123(3), 313–320.

Kitipornchai, S., Wang, C. M., and Trahair, N. S. (1986). ''Buckling of monosymmetric I-beams under moment gradient.'' *J. Struct. Engrg.*, ASCE, 112(4), 781–799.

Kruse, R., Gebhardt, J., and Klawonn, F. (1994). *Foundations of fuzzy systems.* Wiley, New York.

Maher, M. L., Balachandran, M. B., and Zhang, D. M. (1995). *Case-based reasoning in design.* Lawrence Erlbaum Assoc., Mahwah, N.J.

Manual of steel construction—load and resistance factor design. (1994). American Institute of Steel Construction, Inc., Chicago, Ill.

Rumelhart, D., Hinton, G., and Williams, R. (1986). ''Learning representations by back-propagation errors.'' *Parallel distributed processing*, Vol. 1, D. Rumelhart et al., eds., MIT Press, Cambridge, Mass., 318–362.

Stephen, J. E., and Vanluchene, R. D. (1994). ''Integrated assessment of seismic damage in structures.'' *Microcomp. in Civ. Engrg.*, 9(2), 119–128.

Vanluchene, R. D., and Sun, R. (1990). ''Neural networks in structural engineering.'' *Microcomp. in Civ. Engrg.*, 5(3), 207–215.

Zadeh, L. A. (1978). ''Fuzzy set as a basics for a theory of possibility.'' *Fuzzy Sets and Sys.*, 1(1), 3–28.

J. Comput. Civ. Eng. 2000.14:15-22.