Role of Artificial Neural Networks in Multidisciplinary Optimization and Axiomatic Design

Jongsoo Lee †

Key Words : back-propagation neural networks, causality analysis, problem decomposition, multidisciplinary design optimization, design matrix, independence axiom

Abstract

Artificial neural network (ANN) has been extensively used in areas of nonlinear system modeling, analysis and design applications. Basically, ANN has its distinct capabilities of implementing system identification and/or function approximation using a number of input/output patterns that can be obtained via numerical and/or experimental manners. The paper describes a role of ANN, especially a back-propagation neural network (BPN) in the context of engineering analysis, design and optimization. Fundamental mechanism of BPN is briefly summarized in terms of training procedure and function approximation. The BPN based causality analysis (CA) is further discussed to realize the problem decomposition in the context of multidisciplinary design optimization. Such CA is also applied to quantitatively evaluate the uncoupled or decoupled design matrix in the context of axiomatic design with the independence axiom.

1. Introduction

An artificial neural network (ANN) can be described as a massively parallel, interconnected network of computing elements. There are a number of distinctive features of neurobiological methods of computation which are different from traditional numerical computing and the more recently emergent strategies of symbolic processing and computation. There have been significant applications in adapting such computational model to various fields of engineering such as image processing, pattern recognition, fault detection and diagnostic systems. Basically, ANN has its distinct capabilities of implementing system identification and/or function approximation using a number of input/output patterns that can be obtained via numerical and/or experimental manners. The use of neural networks for function approximations and formulations has been successfully used in the context of engineering analysis and design optimization [1-3].

The paper discusses the back-propagation neural network (BPN) based causality analysis (CA) which can identify the quantitative relation between input design and output response values for a rational engineering design formulation. The causality analysis is first explored to represent the problem decomposition in the context of decomposition based multidisciplinary design optimization (MDO) [2,4,5]. There have been recent considerable advances in development of MDO in areas of aerospace and mechanical engineering. MDO has been shown to be an efficient tool for design and optimization in cases where the coupled interactions among participating engineering disciplines should be considered during the analysis and/or design in parallel. A general structure of MDO may contain the multi-criterion and/or multi-objective design formulation, the problem decomposition of design domain, the sensitivity and approximation techniques, and the coordination and integration of locally optimized design solutions. Furthermore, the speed-up of information technology enables to produce practical MDO frameworks that could utilize the CAD/CAE interface and intelligent data base management and communication on different hardware and software systems under the distributed computing environment. The decomposition based design approach is one of the most efficient MDO methods, which decomposes the overall design system into a number of sub-problems/sub-systems/sub-spaces based on participating engineering disciplines or product

† Member, School of Mechanical Engineering, Yonsei University, Seoul 120-749 Korea
E-mail : jleej@yonsei.ac.kr
TEL: (02)2123-4474, FAX: (02)362-2736

components. The decomposition based design is especially effective when the computing process is performed by parallel processing and distributed systems, thereby resulting in the decentralization of computing resource requirements.

The present study describes how BPN based causality analysis can be utilized to evaluate dependencies among design variables and design objectives, and which can be used as a guideline for problem decomposition.

In the design approach, the concept of independence axiom is implemented in the context of axiomatic design. Such rational design method has received the considerable attention in areas of product design and manufacturing. The design matrix in the independence axiom should be described by the uncoupled or decoupled relationship between functional requirements (FRs) and design parameters (DPs). The independence axiom has been applied to structural design optimization problems, wherein design variables are grouped according to analysis of variance (ANOVA) based sensitivity evaluations. In the present study, the sensitivity, that is, the design matrix between FRs and DPs are evaluated via back-propagation neural network (BPN) based causality analysis. After determining the uncoupled/decoupled relation between FRs and DPs based on independence axiom, the design solution can be logically obtained by examining design data.

As another aspect of BPN based CA, therefore, such problem decomposition technique is applied to the identification of the uncoupled or decoupled design matrix in the context of axiomatic design under the principle of the independence axiom [6,7].

2. Back-Propagation Neural Network

BPN refers to a class of feed-forward networks containing at least one hidden layer of artificial neurons whose weights are set by a supervised learning. The network with two hidden layers is sufficient to represent any arbitrary functions of any number of continuous arguments; moreover, it can actually be shown that networks with even a single hidden layer of neurons are flexible enough to represent any continuous function. The effectiveness of multilayer feed-forward networks in such function approximation has played a role in the emergence of neural networks as a tool for solving engineering analysis and design problems.

At the most basic level, BPN can be viewed as providing a mapping between input and output vectors of interest. There is an input layer of neurons which serves as a fan-out layer for the input signal, a number of hidden layers of neurons, and a layer of output neurons where the output vector is available. A number of forms of activation function have been proposed; an activation function for BPN should exhibit some important characteristics. It should be continuous, easily differentiable, and monotonically non-decreasing. The principal advantage of this function is its ability to handle both large and small magnitude f the input signal; large positive and negative values of the input signal yield the vanishingly small gain, while intermediate levels of the input signal generate the finite gain. However, the objective should be to have an appropriate level of gain for a wide range of input signals. All of input-output pairs of training patters are scaled within the finite range of 0.1 and 0.9, and these scaled values are then used to train the network. The training procedure is very similar to constructing a polynomial based response surface, where the sigmoid-type activation function provides a very rich nonlinear behavior for the surface manipulation without the need for having to specify the order of polynomials.

The training algorithm of BPN involves three stages; the feed-forward of the input training pattern, the calculation associated with errors between the output predicted by the network and the actual output, and the adjustment of weights. During the first stage of the feed-forward operation, each input neuron receives the input signal and relays this signal to all neurons in the hidden layer. In general, the strength of the interconnection between and *i*-th neuron of the *k*-th layer and the *j*-th neuron of the *l*-th layer is represented by the interconnection weight w_{ij}^{kl}. For network architecture with a single hidden layer, the input to the *i*-th neuron can be written as follows:

$$z_j = \sum_i w_{ij} X_i \tag{1}$$

where, w_{ij} is the interconnection weight between the *i*-th neuron of the input layer and the *j*-th neuron of the hidden layer. This weighted sum of inputs is processed through an activation function $F(z_j)$ to generate its signal,

$$Z_j = F(z_j) \tag{2}$$

which is then sent to all neurons in the output layer. The network output Y_k is then computed by an activation function after summing its weighted input signals through hidden layers.

$$y_k = \sum_j w_{jk} Z_j \tag{3}$$

$$Y_k = F(y_k) \tag{4}$$

Here w_{jk} is the interconnection weight between the *j*-th neuron of the hidden layer and the *k*-th neuron of the output layer. During the second stage associated with the back-propagation of error, each output layer receives a target pattern T_k, which is corresponding to input training pattern, and then compute its weight correction term Δw_{jk} in terms of its error information terms δ_k as follows:

$$\delta_k = (T_k - Y_k)F'(y_k) \tag{5}$$

$$\Delta w_{jk} = \alpha \delta_k Z_j \tag{6}$$

where, α is a learning rate. Also the weight correction term between a hidden layer and the input layer is obtained from the following procedure:

$$\delta_j = \sum_k \delta_k w_{jk} \tag{7}$$

$$\hat{\delta}_j = \delta_j F'(Z_j) \tag{8}$$

$$\Delta w_{ij} = \alpha \hat{\delta}_j X_i \,. \tag{9}$$

In case of network architecture with multiple hidden layers, the training algorithm is processed within a loop of hidden layers. The process of updating weights is repeated until the network has a specified accuracy; the update equations in this process are given as follows:

$$w_{ij}(t+1) = w_{ij}(t) + \Delta w_{ij}(t) \tag{10}$$

$$w_{jk}(t+1) = w_{jk}(t) + \Delta w_{jk}(t) \,. \tag{11}$$

During the training of BPN, the learning rate has been used to update interconnection weights. A convergence is sometimes faster if a momentum parameter is added to the update of weights. In order to implement the momentum, weights from one or more previous training patterns must be provided. Such momentum allows the network to generate large weight adjustments as long as the corrections are in the same general direction for several patterns, while using a smaller learning rate to prevent a large response to the error from any one training pattern. This also reduces the likelihood that the network will find weights which are a local, and not a global minimum. From another perspective, the momentum parameter helps the network proceed not in the direction of the gradient, but in the direction of a combination of the current gradient and the previous direction of the weight correction. Other variations in updating weights such as an adaptive learning rate of 'delta-bar-delta' and adaptive slopes of sigmoid function can be adopted in improve the network training performance in some environments.

3. Causality Analysis

Decomposition principles in multidisciplinary design require a rational approach to efficiently handle the large-scale problem in optimization. The present study discusses how BPN can be used to identify dependencies among design variables and design objectives, and which can be used as a guideline for problem decomposition. Consider the strengths of interconnections between neurons of the various layers of a well-trained BPN. For a network with a single hidden layer of neurons where the interconnection weights between the input layer and the hidden layer and between the hidden and output layers are denoted as w_{ij} and u_{jk}, respectively. If the magnitude of interconnection weights is used to determine flow-paths of information, then the fraction of signal, f_j, received at the *j*-th hidden layer neuron which can be attributed to the *i*-th input neuron is given as follows:

$$S_j = \sum_{i=1}^{N} | w_{ij} | \tag{12}$$

$$f_j = \frac{| w_{ij} |}{S_j} \tag{13}$$

where, S_j is the sum of the magnitude of weights between all input layer neurons and the *j*-th neuron of the hidden layer. If the output signal is that a node k of the output layer, then the signal from the *j*-th hidden layer node is further multiplied by $| u_{jk} |$ and a sum of all such signals through the different neurons *J* in the hidden layer can be written as follows:

$$\eta_{ik} = \sum_{j=1}^{J} \frac{| w_{ij} | \cdot | u_{jk} |}{S_j} \,. \tag{14}$$

If the sum of all η_{ik} taken over all input layer neurons is denoted by Θ_k, then the fraction of an output weight which is attributable to the *i*-th input neuron can be expressed as the elements t_{ik} of a transition matrix [T] as follows:

$$\Theta_k = \sum_{i=1}^{N} \eta_{ik} \tag{15}$$

$$t_{ik} = \frac{\eta_{ik}}{\Theta_k} \,. \tag{16}$$

This can be further normalized in the following manner:

$$\bar{t}_{ik} = \frac{t_{ik}}{\sum_{i=1}^{N} t_{ik}} \,. \tag{17}$$

The elements of this matrix show the contribution of the *i*-th input to the *k*-th output. It is noted that the above operation is equivalent to a matrix multiplication of normalized or partitioned interconnection weight matrices of successive layers, taken in order; the extension to multiple hidden layer network is straightforward. Inspection of a particular column of this matrix (e.g., the *k*-th column), the influence of each input component on the *k*-th output component can be qualitatively assessed. Since the sum of all elements in a column is unity, this influence is represented by a fractional quantity, and allows for easy identification of dominant input variables, and hence a topology for decomposing the design problem. This transition is referred to as the absolute causality analysis (ABS-CA) matrix. It is noted that a change in the transition matrix resulting from a change in size of the training set can be used to determine if an adequate number of training patterns has been considered.

In the jargon of optimization methods, this matrix yields an aggregate sensitivity which is valid over a design range defined by the domain over which the neural network is trained. If the region is large, a number of networks can be developed, each trained one for a cluster of the design region with similar characteristics. An alternative approach for computing the aggregate sensitivity matrix has been developed; this approach simply considers the matrix product of the interconnection matrices without taking an absolute value of each weight coefficient. This approach, therefore, incorporates the effect of the sign of interconnection weights in the analysis, and is useful in identifying both the magnitude and direction of input-output interactions. This matrix is referred to as the alternative causality analysis (ALT-CA) matrix. In this approach, the matrix product and normalization of the interconnection weight matrices are described as follows:

$$[T] = \prod_{n=1}^{N-1} W^n \tag{18}$$

$$\bar{T}_{ij} = \frac{T_{ij}}{\max_i |T_{ij}|}. \tag{19}$$

In the above, W^n is the n-th weight matrix, the coefficients w_{ij}^{kl} of which represents the interconnection weight between the *i*-th neuron of the *k*-th layer and the *j*-th neuron of the *l*-th layer; N denotes the total number of layers of neurons in the BPN architecture. This normalized matrix w_{ij} incorporates the effect of sign of interconnection weight in the analysis.

4. Decomposition in MDO

The neural network based causality analysis provides valuable information about the extent of coupling in the large-scale multidisciplinary system. In order to implement a decomposition based design strategy, the problem must be partitioned into a appropriate number of sub-problems depending upon available computing machines or parallel processors. Optimal partitioning schemes for system decomposition have been widely used in design and manufacturing applications for process and scheduling. A reasonable and logical approach for partitioning is one where balanced subsets of design variables would be assigned to different sub-problems, and where each sub-problem would be responsible for meeting the system level design objectives and for satisfying constraints most critically affected by the design variables of the corresponding sub-problem. This approach is implemented in the context of interconnection weights of BPN based transition matrix (i.e., ABS-CA) as follows:

For a transition matrix $[\bar{T}]$ obtained from a trained BPN, partition this matrix into a total of K (where $2 \le K \le NCON$, and $NCON$ is the total number of constraints in the design problem) different groups denoted as G_k; each group contains design variables x_i which have the strongest influence on constraints belonging to the group G_k. To formalize the partitioning procedure, define a grouping identification matrix with element V_{ij} such that

$$V_{ij} = 1 \quad if \quad x_i \in G_k \tag{20}$$

$$V_{ij} = 0 \quad if \quad x_i \notin G_k$$

In the above, the subscript '*j*' refers to the *j*-th constraint. To obtain an optimal partitioning, a performance index *PI* is determined as follows:

$$PI = \frac{1}{\alpha} \sum_{i=1}^{NDV} \sum_{j=1}^{NCON} \frac{B_{ij}}{z_i} \tag{21}$$

$$B_{ij} = V_{ij} \cdot |\bar{T}_{ij}| \tag{22}$$

$$\alpha = \sum_{i=1}^{NDV} \sum_{j=1}^{NCON} |\bar{T}_{ij}| \tag{23}$$

where, *NDV* is the number of design variables, Note that the element B_{ij} are obtained as a scalar product, and have a value of either zero or the absolute value of the coefficient $\bar{T}_{ij}$ of the transition matrix; z_i is the number of non-zero value in the *i*-th row of the matrix B_{ij}. The mathematical statement of optimal problem partitioning can be now formulated as follows:

$$\text{Minimize} \quad f = \frac{1}{PI} + \frac{1}{2} \sum_{i=1}^{K} \sum_{j=1}^{K} (1 - \delta_{ij}) \, |N_{x_i} - N_{x_j}|$$

$$\text{Subject to} \quad 1 \le N(g_j)_{G_k} \le N(g_j)^{Upper} \tag{24}$$

where, δ_{ij} is a Kronecker delta, N_{x_k} is the number of design variables to be assigned to a group G_k, and $N(g_j)_{G_k}$ is the number of output responses (i.e., constraint functions) in a group G_k. The objective function has two components – the first term leads to a maximization of the performance index *PI* while the second term ensures a minimal difference between the numbers of design variables in each group. The constraint condition in the above optimal partitioning problem is necessary to limit the number of design constraints denoted as $N(g_j)^{Upper}$ is used. In this optimal partitioning scheme, the design variables are the allocation of elements of the grouping identification matrix V_{ij}. This is an integer programming problem which can be conventionally solved using genetic algorithm or evolutionary strategies.

5. Design Matrix in Independence Axiom

The paper presents a logical design process for the application of the axiomatic design, an efficient method useful in developing new products. The axiomatic design with independence axiom indicates that the independence should be maintained between functional requirements (FRs) and design parameters (DPs) in an uncoupled or decoupled manner. The design matrix [A] explains the relation between FRs and DPs as follows:

$$\{FR_s\} = [\mathrm{A}]\{DP_s\} \tag{25}$$

If the design matrix by the analysis of independence axiom comes up with a diagonal matrix, it is called an uncoupled design, which is an ideal design matrix as follows:

$$\begin{Bmatrix} FR_1 \\ FR_2 \end{Bmatrix} = \begin{bmatrix} X & O \\ O & X \end{bmatrix} \begin{Bmatrix} DP_1 \\ DP_2 \end{Bmatrix} \tag{26}$$

where, ‘X’ represents connection between FR and DP while ‘O’ means there is no relation. If the design matrix is a triangular matrix, it is called a decoupled design as follows:

$$\begin{Bmatrix} FR_1 \\ FR_2 \end{Bmatrix} = \begin{bmatrix} X & O \\ X & X \end{bmatrix} \begin{Bmatrix} DP_1 \\ DP_2 \end{Bmatrix} \tag{27}$$

Other matrixes are coupled designs. In case of coupled designs, there exists a feedback in the relationship of FRs and DPs. Therefore, the coupled designs should be avoided in the paradigm of the axiomatic design. In other words, the uncoupled design, which is an ideal design method in maintaining independent relationship between FRs and DPs, is the best way in approaching problems. However, pursuing such an ideal design has its limitations in reality so that a decoupled design method, through which the design factors of coupled problems are determined one by one, is suggested. The present study takes logical approaches according to decoupled design methods.

$$[\mathrm{A}] = \begin{bmatrix} \mathrm{X} & \mathrm{X} \\ \mathrm{X} & \mathrm{X} \end{bmatrix} \rightarrow \begin{bmatrix} \mathrm{X} & \mathrm{O} \\ \mathrm{X} & \mathrm{X} \end{bmatrix} \tag{28}$$

The first step during the design matrix is to select the design parameters which affect functional requirements in a uncoupled or decoupled manner in the context of axiomatic design with independence axiom. The sensitivity, that is, the design matrix between FRs and DPs would be evaluated via both analysis of means (ANOM) and BPN based causality analysis.

After determining the uncoupled/decoupled relation between FRs and DPs based on independence axiom, the design solution is locally obtained by looking at design data. Upon such uncoupled/decoupled design matrix between design parameters and functional requirements, the formal optimization process would be conducted.

6. Closing Remarks

The paper describes the benefit of BPN based causality analysis in the context of multidisciplinary design optimization and axiomatic design. In engineering design problems, the problem decomposition is to multidisciplinary design optimization and the uncoupled or decoupled design matrix is to the axiomatic design. The causality analysis directs how the design problem is effectively decomposed into a number of sub-problems which are strongly related between input variables and output responses.

Acknowledgments

This research is supported by the second stage of Brain Korea 21 (BK21) Program.

References

[1] Lee, J., Genetic Algorithms in Multidisciplinary Design of Low Vibration Rotors, Ph.D. Dissertation in Mechanical Engineering, Rensselaer Polytechnic Institute, Troy, NY, 1996.

[2] Lee, J., and Hajela, P., “Parallel Genetic Algorithm Implementation in Multidisciplinary Rotor Blade Design,” Journal of Aircraft, Vol. 33, No. 5, pp. 962-969, 1996.

[3] Lee, J., and Kim, S., “Optimal Design of Engine Mount Rubber Considering Stiffness and Fatigue Strength,” Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering, Vol. 221, No. 7, pp. 823-835, 2007.

[4] Lee, J., and Jeong, H., “A Decomposition Based Design Method Coordinated by Disciplinary Subspace Optimization,” JSME International Series C-Mechanical System, Machine Element Design, Vol. 49, No. 3, pp. 935-941, 2006.

[5] Kim, W.-S., and Lee, J., “System Decomposition Techniques in Multidisciplinary Design Optimization Problems Using Genetic Algorithms and Neural Networks,” Journal of Computational Structural Engineering Institute of Korea, Vol. 12, pp. 619-627, 1999.

[6] Kim, J., Lee, J., and Cha, S., “Optimization of Injection Molding System Using Axiomatic Design,” Transactions of KSME (A), Vol. 27, No. 6, pp. 1020-

1027, 2003.
[7] Lee, J., and Shin, H., “Parameter Design of Water Jet Nozzle Utilizing Independence Axiom,” Proceedings of the Institution of Mechanical Engineers, Part E: Journal of Process Mechanical Engineering, Vol. 222, No. 3, pp. 157-169, 2008.