



## The use of artificial neural networks in construction management: a review

A. Halim Boussabaine

**To cite this article:** A. Halim Boussabaine (1996) The use of artificial neural networks in construction management: a review, *Construction Management & Economics*, 14:5, 427-436, DOI: [10.1080/014461996373296](https://doi.org/10.1080/014461996373296)

**To link to this article:** <https://doi.org/10.1080/014461996373296>



Published online: 21 Oct 2010.



Submit your article to this journal [↗](#)



Article views: 869



View related articles [↗](#)



Citing articles: 16 View citing articles [↗](#)

# The use of artificial neural networks in construction management: a review

A. HALIM BOUSSABAIN

*School of Architecture and Building Engineering, University of Liverpool, PO Box 147, Liverpool L69 3BX, UK*

Received 11 July 1996; accepted 22 December 1995

Artificial neural networks (ANNs) are systems that can learn. A neural system can be trained on a set of input and output data belonging to a particular problem. If new data of the same problem, but not in the training set, are presented to the system, the ANN can use the learned data to predict outcomes without any specific programming relating to the category of events involved. The fields of application of ANNs have increased dramatically in the past few years. A large variety of possible ANN applications now exist for non-computer specialists. Therefore, with only a very modest knowledge of the theory behind ANNs, it is possible to tackle complicated problems in a researcher's own area of speciality with the ANN technique. This is true in the field of construction management, the topic of this review. The review is divided into three sections: a background introduction to ANN methods, a review of the most important application of the ANN techniques to this point and a selection of construction management topics that have potential application.

**Keywords:** Artificial neural networks, information technology, automation, neural systems, intelligence.

## Background introduction

Artificial neural networks (ANNs) offer an approach to computation that is different from conventional analytic methods. ANNs are an information processing technology that simulate the human brain and the nervous system. Like the human brain, neural networks learn from experience, generalize from previous examples to new ones and abstract essential characteristics from inputs containing irrelevant data. Network components with names such as neurons (sometime referred to as cells, units or nodes) and synaptic transmissions with weight factors are used to mimic the nervous system (analogous to synaptic connections in the nervous system) in a way which allows signals to travel through the network in parallel as well as serially. Although neural networks have some qualities in common with the human brain, this resemblance is only superficial. The ANN will most likely never be able to duplicate completely the functions of the human brain. Various ANN models have been

proposed over the past decades and impressive results have been obtained with some of the designs. The most popular model for many applications is the non-linear multilayered network. This method is an outgrowth of the single-neuron linear neural network approach that started in the 1960s. The original models consisted of two layers of computational neurons: input and output neurons. These models are easier to train and have found widespread commercial application (Fausett, 1994). Unfortunately, only a limited number of applications could be solved with this technique. The introduction of non-linear multi-element ANNs and algorithms for calculating and correcting errors of the network's performance (e.g. the back-propagation method) made ANNs very productive in the 1980s. Several publications describe the development and theory of ANNs from the introductory level to more advanced stages. For example, Lippman (1987) and more recently Hush and Horne (1993) published updated reviews of several ANN models. Barron and Barron (1989) and Levin *et al.* (1990) provided a

statistical interpretation of the methods used to train ANNs. Nerrand *et al.* (1993) showed that ANNs can be considered as general non-linear filters that can be trained adaptively. A clear summary of the feed-forward network and the back-propagation models can be found in Kamarthi *et al.* (1992). Kohonen (1988) reported interesting and useful results from his research on self-organizing feature maps used for pattern recognition and signal processing. It has been proven that problems which involve complex non-linear relationships can be better solved by neural networks than by conventional methods (Rumelhart *et al.* 1994). ANNs are suited to such problems because of their adaptivity owing to their structure; that is, non-linear activation functions (Flood and Kartam, 1994a and b). Adaptivity allows the neural network to perform well even when the environment or the system being modelled varies with time. ANNs have many advantages over traditional methods of modelling in situations where the process to be modelled is complex to the extent it cannot be explicitly represented in mathematical or statistical terms or that explicit formation causes loss of sensitivity due to over-simplification. Traditional models lack the ability to learn by themselves, generalize solutions and respond adequately to highly correlated, incomplete or previously unknown data. For this type of environment ANN models are superior to other models. The most important advantage of ANNs over mathematical and statistical models is their adaptivity. ANN systems can automatically adjust their weights to optimize their behaviour as decision makers, predictors, etc. Self-optimization allows the neural to design itself. In the construction management field ANNs will probably be seen as components of larger systems which make use of expert-given rules or statistical inference techniques as required. Such systems, in turn, will be able to provide decision support for experts, help decision makers perform at a higher level, assist in the training of inexperienced personnel and help scenario planning (i.e. what if?) by managers. A useful ANN decision support system must be robust, easy to use and it should enhance the process of decision making. An ANN that simply models the decision-making behaviour of the user is likely to be of limited use.

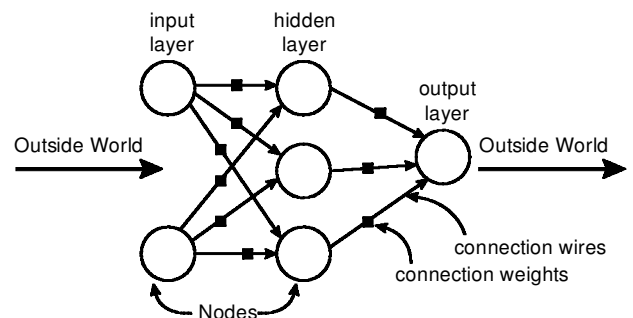
This review will concentrate on the ANN techniques and concepts useful in construction management and help researchers to identify opportunities where this new technology is applicable in assisting decision makers. In addition, this paper provides guidance and tips for the development of successful applications in construction management.

## Neural network concepts

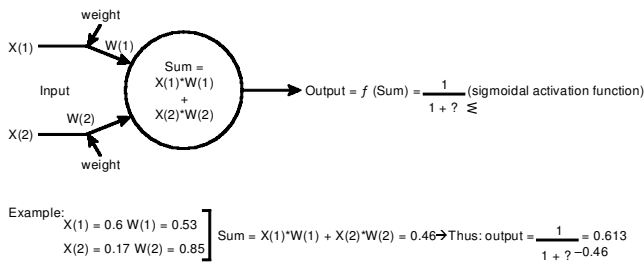
There is a diverse range of ANN models in terms of topology and mode of operation. However, each model can be specified by the following seven major concepts (Lippman, 1987; Hall, 1992; Hush and Horne, 1993).

1. A set of processing neurons.
2. A state of activation for each neuron.
3. A pattern of connectivity among the neurons or topology of the network.
4. A propagation method to propagate the activities of the neurons through the network.
5. An activation rule to update the activities of each node.
6. An external environment that provides information to the network and interacts with it.
7. A learning method to modify the pattern of connectivity by using information provided by the external environment.

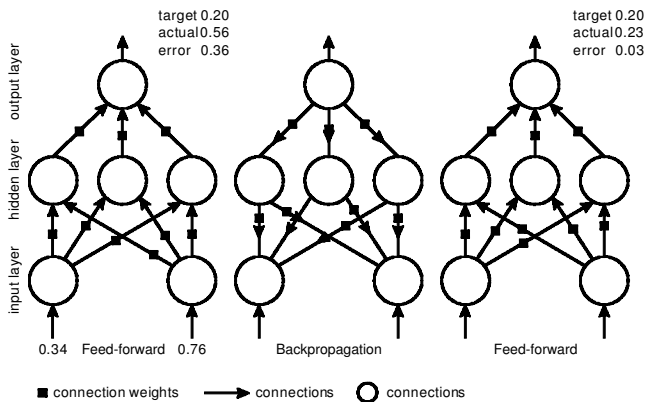
Figure 1 illustrates a multilayered ANN with three layers. These consist of a number of nodes with each of the nodes in one layer linked to each node in the next layer. The communication with the outside world occurs through the nodes of the input and output layers. The middle layer, which is hidden from the outside, gives a critical computational ability to the system. The functioning of the nodes is illustrated in Figure 2. In a simple case the node receives only two inputs  $X(1)$  and  $X(2)$ , respectively, with corresponding weight factors  $W(1)$  and  $W(2)$ . The node calculates the sum,  $X(1)W(1) + X(2)W(2)$  and delivers an output value obtained from a special sigmoidal function, the activation functions, which can take on a variety of forms. The output reached in this fashion is delivered to nodes in the next layer, where a computation similar to the one described above takes place. If the node is in the output layer the obtained value has reached its final destination. The pattern of connectivity or the network topology specifies how each node is connected to the other units in the network. The strength of each connection is represented by a real number (weight).



**Figure 1** An artificial neural network with three layers



**Figure 2** Activities at the neural network node



**Figure 3** Phases in neural network working procedure

The weights represent the knowledge that is encoded into the network. As the network learns, the numerical values of the weights may change according to the new information that is circulating in the network. A learning method (for example, back-propagation) is used to change the weight of the network and other adaptable parameters. The working principles of the three-layered network with back-propagation are outlined in Figure 3.

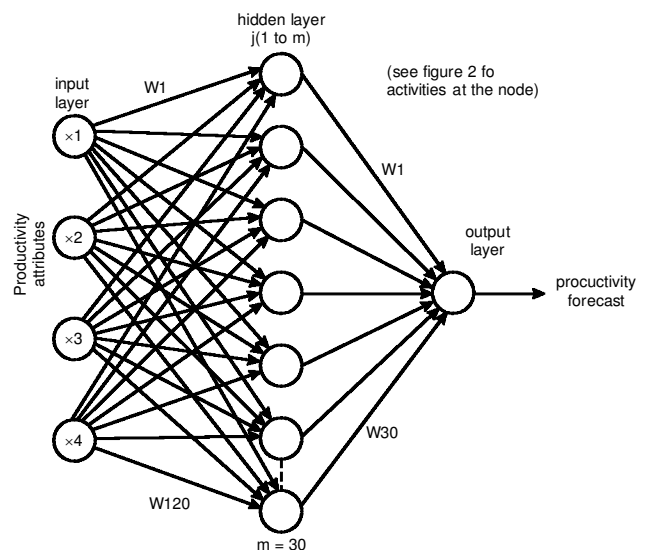
### How do neural networks work?

The working principles of the three-layered network with back-propagation are shown in Figure 3. Input information is presented to the ANN for each sample and the specified target number given, if supervised training is used. During training, the input layer broadcasts a pattern to all the hidden nodes. The system is then asked to calculate an output value in a feed-forward way following the specification stated in Figure 2. The hidden nodes broadcast their results to all output nodes. Each output node then calculates a weighted sum (Figure 2) and passes it to the output node to generate an actual result. The result is compared with the target value, which the trainer has

established at the onset of a training session. The difference yields the system output error. At this stage the system has to decide whether further learning is required. This is accomplished by comparing the obtained total difference with a specified acceptable error given by the system developer. If the decision is to continue, the output nodes calculate the derivatives of the error with respect to the weights and the result is sent back through the system to all the hidden nodes. Each hidden node calculates the weighted sum of the error. Then, each hidden-layer node and output-layer node change their weights to compensate for the corrections. Once the weights have been changed, the feed-forward computation starts all over again. New output values are obtained and the cycle continues until a desired result is obtained. At this stage we can say that the training of the system is complete and the testing phase can start. The system can now be used to predict the outcome of an input not previously seen by the ANN.

### An example of a neural network system

To illustrate the concepts of ANNs, consider an example of a neural network system that predicts a construction project's productivity. The systems' topology is shown in Figure 4. The input values to the network are in a range in the interval of (0–1), as continuous values. The values of these inputs are a range of likely effects on operations productivity. Table 1 shows a sample of data used to train the system. The number of nodes in the first layer is equal to the number of problem attributes and the number of hidden layers and nodes are heuristically chosen as



**Figure 4** Structure of the system

**Table 1** A sample from training data: labour factors (Boussabaine and Cheetham 1995)

Low effect		Mean effect		High effect	
Loss	Prob-ability	Loss	Prob-ability	Loss	Prob-ability
%	%	%	%	%	%
2	80	3.5	28.6	5	10
5	64	8	53.6	7	12.14
10	47	10	60	15	19.36

shown in Table 2. The last layer consists of a single node that is used to predict the percentage loss on productivity. The system was then asked to learn and map the pattern between the inputs. The procedure was repeated for each pattern of inputs. A series of learning exercises was designed to find the optimal choice of the number of hidden layers and the number of nodes in each hidden layer. The results are presented in Table 2. A few observations relating to these results are of general interest. First, as noted above, the number of nodes and hidden layers must be determined through experimentation. In the present case 30 nodes with one hidden layer seems to be optimal for the mean square error. Increasing the number of hidden layers and nodes would cause a corresponding increase in the training time and might not be a suitable solution in a practical situation.

After successful training, the performance of the network was tested on a sample of data. The standard method for predicting generalization performance is called cross-validation. This method works by splitting the data into two sets, a training set and a test set. Learning is performed on the training set, and network performance is evaluated on the test set. To achieve statistically significant results it is generally necessary to perform several independent splits and then average the results to obtain an overall estimate of performance. While cross-validation is a widely accepted method, it can be extremely time-consuming in neural networks

because of the lengthy learning times required for each of the splits. In this example, 70% of the sample ( $n = 324$ ) were randomly selected as training cases. The remaining 30% of the sample ( $m = 120$ ) comprised the testing cases. The system is trained until the root mean square error (RMS) is reduced to 0.0011% loss on productivity. The results of the testing are shown in Figure 5. This clearly shows that the system has learned to generalize from the training set. However, there is a need to test the reliability of this generalization performance of the network. A predicted square error (PSE) technique is used to test the system. This technique relies on statistical analysis methods to derive an expression for the generalization performance of a system as a function of its performance on the training set, the number of free parameters (weights) in the system and the training size (Moody, 1992). The predicted mean square error is given by

$$PSE = RMS + (2Nw/P)V^2 \quad (1)$$

$$V^2 = [P/(P - Nw)]RMS \quad (2)$$

where  $RMS$  is the mean square error on the training set,  $Nw$  is the number of weights (free parameters),  $P$  is the number of training samples and  $V$  is the variance. It has been proven that Equation 1 provides an unbiased estimate of the predicted RMS for systems that are non-linear in their parameters. By using Equations 1 and 2 the PSE of the system is estimated. The PSE of the system is 0.0019. This shows that the system can generalize from the training sample and that the training set is large enough and the performance of the network on the training data is an accurate measure of its performance on future data.

## Developing neural network applications

Much of the proposed neural network development methodology is adapted from conventional software and expert systems development. Neural network

**Table 2** Experimenting with network topology

Hidden layers	Neurons		Learning rate		RMS error	Time (min)	Learning iterations
	Layer 1	Layer 2	Rate	Momentum			
1	5	0	0.6	0.9	0.026582	10	1000
1	10	0	0.6	0.9	0.007898	35	1000
1	20	0	0.6	0.9	0.002245	45	1000
1	30	0	0.6	0.9	0.001103	70	1000
1	40	0	0.6	0.9	0.023121	105	1000
2	10	5	0.6	0.9	0.026738	50	1000
2	30	10	0.6	0.9	0.051692	110	1000
2	30	15	0.6	0.9	0.043933	140	1000
2	20	10	0.6	0.9	0.039705	90	1000

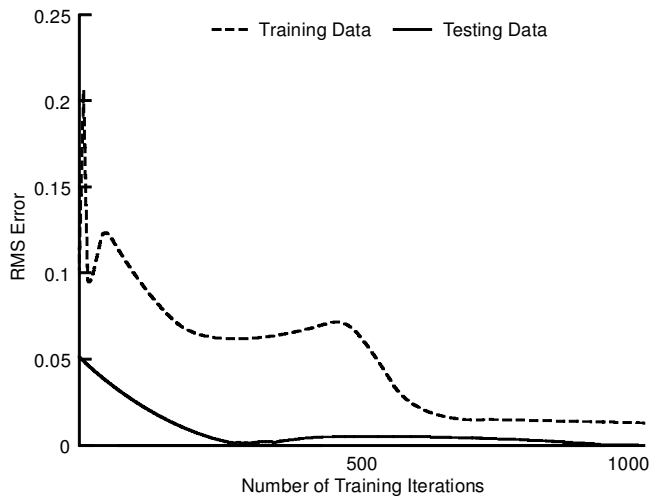


Figure 5 Training and testing results of the system

development places a stronger emphasis on experimentation and multiple simultaneous development methods. This systematic approach is shown in Figure 6.

### Selecting the domain of application

This stage involves the study of the existing situation with a view to formulating a feasibility assessment for selecting and validating the proposed application. The following heuristic rules are used for selecting successful applications (Bailey and Thompson, 1990; Hammertrom, 1993; Widrow *et al.*, 1994; Boussabaine, 1995).

1. Conventional statistical and mathematical methods are inadequate.
2. Conventional computer technology is inadequate.
3. The problem requires qualitative or complex quantitative reasoning.
4. Solutions are derived from highly interdependent parameters that have no precise quantification.
5. Data is multivariate and intrinsically noisy or error prone.

### Selecting the system attributes

This stage involves the choice of the system variables and indicators and an understanding of their significance, correlation and their possible values that describe the problem being analysed and characterize its input and output patterns. It is very useful to have networks which provide output values that are reason-

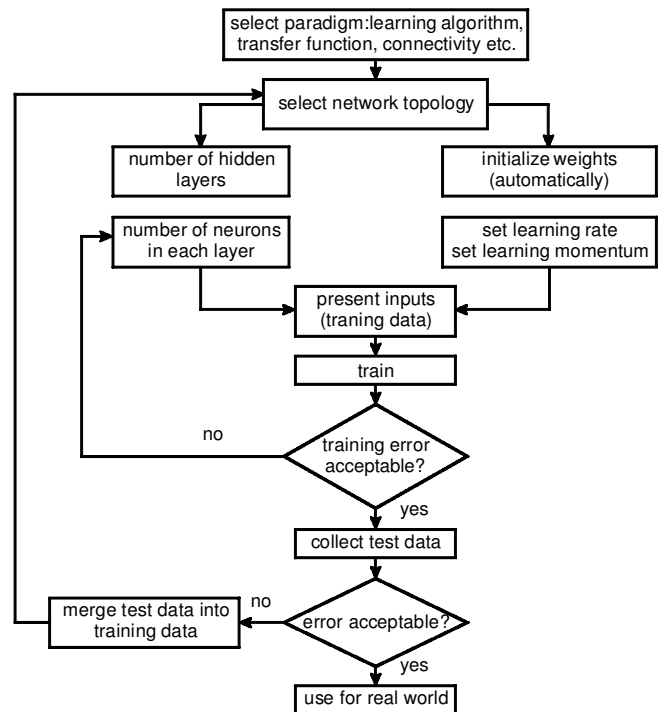


Figure 6 A systematic development process

ably interpreted as probabilities (Masters, 1993). These probabilities can then be used to determine levels of confidence and combine with other sources of evidence.

### Selecting neural paradigms

Several paradigms have recently been developed to deal with different problems. These paradigms vary a great deal with regard to the type of input patterns they accept, the output patterns they produce and the learning characteristics and the limitations they exhibit (Kohonen, 1988; Barron and Barron, 1989; MacKay, 1992; Masters 1993). Therefore, this stage is concerned with experimenting on these different paradigms to select a network topology and to fine-tune the network structure and attributes to obtain a suitable paradigm. Table 3 specifies the relationship between the application requirements and the capabilities of selected neural paradigms. Several of these paradigms (e.g. back-propagation and Boltzmann machine) offer flexibility in terms of size (number of layers), connectivity (how nodes are linked together), learning algorithm and training methods, whereas others (e.g. ART, Kohonen network, Hopfield, etc.) are more tightly contained in the nature of their topology or learning algorithm. There are no algorithms for determining the best paradigm for a particular application. However, the selection should be based on the comparison of

**Table 3** Neural network paradigms (Bailey and Thompson, 1990)

Paradigm	Transfer function	Number of layers	Type of input accepted	Size of hidden layers	Connectivity	Learning algorithms	Learning parameters
Back-propagation	Sigmoid	2 or more	Continuous	Small to medium	Fully interconnected random, layers	Generalized data rule	Learning constant momentum
Counter propagation	Kohonen and sigmoid	2 or 3	Continuous	Same as Kohonen	Fully interconnected	Kohonen and Grossberg	Kohonen and Outstar
Madaline	Signum	2	Bipolar	Small to medium	Fully interconnected	Delta rule and max, min, majority	Learning constant
Outstar	Sigmoid	1 continuous	Binary	N/A application	Varies by	Outstar function	Decay time attack
ART2	Sigmoid	1	Grey scale	Increases b data type	Fully interconnected	ATR2	Vigilance, gain
Kohonen network	Competitive learning	1	Continuous data type	Equals number of interconnected	Fully	Kohonen size, alpha	Neighbourhood
Boltzmann machine	Varies	2 or more	Binary continuous	Small to medium	Fully interconnected random	Boltzmann	Temperature
Hopfield network	Hard limiting	1	Binary bipolar	N/A	Fully interconnected	Hopfield	None
Adaline	Signum	1	Bipolar	N/A	Fully Interconnected random	Delta rule	Learning constant
Bam	Clamped linear	1	Bipolar	N/A	Fully interconnected	Bam	Retention, gain
Perceptron	Perceptron	1	Continuous	N/A	Random	Perceptron convergence	Learning constant

the application requirements (e.g. the type of input required by the application limits the neural paradigms that can be selected, see Table 3, column 4) to neural paradigm capabilities.

### Training the system

The most widely used training methods are supervised, unsupervised and reinforcement learning (Bailey and Thompson, 1990; Masters, 1993). If the supervised learning method is used then the system is presented with a set of inputs and a desired output the network is expected to provide. In unsupervised learning the external environment does not provide the desired network output nor classifies it into categories. By using the correlation of the input, the learning method changes the network weights to group the input into 'clusters' such that similar inputs will produce similar network outputs since they will belong to the same cluster. Reinforcement learning requires a set of inputs and only a grade as output. The selection of a training method depends on the data available, the problem being analysed and timing. However, in each method, the number of training samples must be sufficiently large to guarantee good generalization. The training data can be designed for a good generalization by observing the following two rules (Moody, 1992).

1. The training inputs must span the whole range of expected inputs. This means that the network will not have to extrapolate.
2. The distribution of training inputs must be sufficiently dense over the whole range of expected inputs to produce an accurate interpolation.

After the selection of training and testing data, the network topology and attributes are varied and their impact on the network is monitored. This process is repeated until the network's overall accuracy is improved by the aggregate corrections during training.

### Validating the system

The validation process starts by splitting the data into two sets, a training set and a test set. Learning is performed on the training set and the network performance is evaluated on the test set. To achieve statistically significant results it is generally necessary to perform several independent splits and then average the results to obtain an overall estimate of performance (Moody, 1991). When a neural network is trained, the measure of performance that is optimized is usually the mean square error of the outputs (Masters, 1993). A low RMS error can be misleading and a low training error need not infer good performance. For example,

if a network system has too many weights (because of too many nodes or layers) relative to the number of samples in the training set, it can overfit data. The system learns irrelevant details about the individual samples rather than the basic structure of the data presented to it (Boussabaine and Cheetham, 1995). If the difference between the training error and testing error is large, then either the training set and test set is not representative of the same population or the system has been overfitted. The RMS method has no ability to differentiate between minor and serious errors. There are also some error measures closely related to the RMS that can be used for some problems. The absolute error, maximum absolute error and median error statistical measures are more robust and reliable than measures based on the RMS (Masters, 1993). An alternative method that requires less calculation and addresses the shortcomings of the RMS method is described above (see an example of a neural network system).

The following qualitative and quantitative criteria can also be used to test the validity of ANN systems.

1. Consistency: a system is consistent if repeated executions with the same data lead to the same output.
2. Generality: generality is measured in terms of the ability of the ANN system to be used with a range of similar problems. A system is broad when the range and conditions within which it operates reliably are wide.
3. Accuracy: this is measured by comparing the number of correct predictions with known data.
4. Sensitivity: this method specifically determines the sensitivity of results to each input parameter by varying that parameter incrementally, holding all other parameters constant and observing the output results.

### Problems in validating neural network systems

Neural network systems provide no obvious explanation of how they solve a problem, so results cannot be accounted for. Neural networks are not unlike human experts in this sense: both express opinions that they cannot easily explain. Neural networks can pinpoint certain factors, that were thought to have been irrelevant or which conflict with traditional theories, as important for decision making. This aspect can be extremely frustrating due to the fact that there is no way to determine whether the network has incorrectly identified these factors or if, by chance, traditionally accepted methods are wrong. There is also no assurance that the network will train to the best configuration possible. Even if the neural network is functioning

correctly, it can still be prone to errors. However, even so, all networks suffer from limitations in their ability to learn and to recall. The importance of the degree of accuracy must be assessed and then it must be decided if it is worthwhile to use neural networks.

### Construction management applications

Successful commercial applications of non-linear multielement neural networks, have found widespread commercial applications (Asakawa and Takagi, 1994; Widrow *et al.*, 1994). Many banks are currently using ANNs to study patterns of credit card usage and to detect transactions that are potentially fraudulent. ANN systems are also used for financial forecasting in a large number of investment and financial firms. Using ANN systems trained with genetic algorithms, some financial firms claim to be able to earn 25% returns per year investing in the currency markets. ANN models are also used to optimize marketing strategies and cut marketing costs by removing unlikely future customers from a list of potential customers. This is only a sample of successful commercial applications of ANNs the list is long and impressive and growing rapidly (see Widrow *et al.*, 1994). In contrast, so far little has been published on the application of neural networks to construction management. Applications in construction management go back to the early 1990s, but already cover a range of topics. For example, Flood (1989) described the development of an ANN-based method for optimizing the sequencing of construction tasks with the objective of minimizing production time. The basic idea behind this work is that when a matrix of times spent by each job at each process is presented to the input layer, the network will respond by producing an optimal job sequence across the output layer. The sequence is dictated by the relative levels of activation of the output neurons. The testing results proved that the network had formed a valid model of the problem. However, this model lacks a rigorous analysis to assess the performance of the model in relation to variance between patterns and alternative network topologies. Moselhi *et al.* (1991) reported on the development of a trial neural network system for optimum mark-up estimation under different bid situations. The system uses as its input the number of typical competitors, the mean of the distribution of the ratio of the competitors' bid prices to the contractor's estimated cost in previous encounters and the standard deviation of the latter distribution. Three bidding strategy models were used to provide the desired network output (optimum mark-ups) in response to the different bid situations. The authors claimed that the model was able to generalize solutions and capture



the probabilistic nature of ten bid situations used in training. This seems to contradict the fact that the data sample used for training is too small which may have contributed to an overfitting problem. If overfitting had occurred then the model cannot generalize (interpolate) from new data because every training item has been learned perfectly at the expense of learning the underlying regularities of the domain. Therefore, data or even some errors in the input data may not lead to an incorrect forecast in well-trained networks because ANNs have some properties of generalization and fault tolerance. Williams (1994) developed a back-propagation ANN model to predict the changes in construction cost indexes. The system uses as input the recent trends, the prime lending rate, housing starts and the month of the year. The model produces three outputs: prediction of the percentage change of the construction cost index 1 month ahead, prediction of the percentage change in the construction index 6 months ahead and one prediction. The output from the ANN system was compared with predictions made by an exponential smoothing and simple linear regression models. The authors found that the prediction produced by the ANN model gave a greater error than statistical models. They concluded that the movement of the cost indexes is a complex problem that cannot be predicted accurately by a back-propagation ANN model. These findings contradict the advantages claimed for ANNs over statistical methods (see background). The failure of this model could be attributed to the selection and design of input data or failure to find an optimum network topology and fine-tune the network structure (weights, nodes and layers) to obtain a suitable model. Chao and Skibniewski (1994) developed two ANN modules for estimating excavation capacity based on job conditions and estimating excavator efficiency based on the attributes of operation elements. The training data were generated from a desk-top excavator model and a simulation program. The output of the first module, excavator cycle time, is used as an input to the second module. The outputs of the second module include hourly productivity plus the mean and standard deviation. The work of Chao and Skibniewski (1994) is limited because the number of hidden layers was fixed and there was no search for the optimum set-up of ANN parameters. However, it demonstrated the potential for applying neural networks to construction operation productivity estimation. Mirza and Fisher (1994) described a model for decision making about construction modularization using ANNs. The decision is based on factors such as plant location, project risks etc. The system is trained using 40 cases and the performance of the model is tested on ten separate cases. The validation tests showed that the ANN decisions were accurate. The

training set and testing cases are limited and too small, therefore, this might have lead to the problem of overfitting and inaccurate generalization of decisions by the model. All of these applications are still in their infancy. None of them has been fully developed to implementation stage.

The potential benefits of ANNs as decision support tools in construction management can seem obvious to a researcher in the field of information technology (IT) but they are often less clear to construction industry users and even some academics in construction management are suspicious of the artificial intelligence systems. Another concern is the 'black-box' nature of ANNs. Data go in and a prediction comes out but the user has no understanding of what happens in-between. Therefore, it is essential that the benefits of ANNs to construction management over conventional methods are clearly demonstrated in randomized pilot studies. The following are some areas where these pilot studies can be of potential importance to construction management.

### **Predicting project cash flow and costs**

An ANN system can be used to provide assistance to construction contractors in predicting, updating and managing (planning and control) project cash flow and cost. This will assist the contractor in taking necessary managerial action to avoid a shortage of cash, bankruptcy and to give early warning of cost and time overrun. The input of such a system would be based on the data collected by the cost system, cost estimating and planning.

### **Risk analysis**

An ANN for risk analysis as a tool to assist with decision making in financial investments and in assessing situations where opportunities for alternative contractual arrangements are available involving different allocations of risk and reward among the construction companies involved, can have beneficial results from its application. In addition, it is possible to develop an ANN system to assess risks of project cost overruns and assist in developing contingency management strategies. Such a system could have as an input the sources of risk (for example, project complexity, unrealistic estimates, poor project specification, etc.) and as the outputs a classification of risk and the range of likely estimates.

## Decision making

Many decisions take place under conditions of uncertainty. Construction managers take decisions with incomplete knowledge or knowing that the outcomes of these decisions are at best uncertain. While there are a number of mathematical and statistical models for assisting managers to take decisions, these models are based on probability and regression techniques whereby the best fit is sought. In addition, these models lack the ability to learn or generalize solutions from incomplete or previously unseen data. Here, the characteristics of ANNs described above offer a chance to match the problem's complexity.

## Resource optimization

In construction management many problems are encountered that require minimizing costs by optimizing the contending factors. For example, one of the problems in construction is resource allocation. Resource allocation has been a major concern of researchers and practitioners since the development of CPM/PERT network planning techniques. Many methods have been used to solve this problem. However, none of these models have the ability to predict the effect of factors that affect the allocation of resources (e.g. design change, site conditions, plant conditions, etc.) and learn from experience. For such a complex problem, a neurofuzzy system that combines fuzzy concepts and neural network technology to predict and determine the priority ranking of resources would be of a great benefit to construction managers and planners.

## Prediction of tendering outcomes

A neural network system can be developed to assist with the deliberation of the circumstances in which competitive or other type of bids should or should not be made and in helping contractors to predict the results of a bid before embarking on the costly process of estimating and costing of the bid, as a result of assessing accurately the risk involved in a particular bid or contract.

## Classification and selection

The selection and classification of construction material, plant, construction methods, etc., is a complex problem and has to be based on experience and judgement. Information and knowledge about the

processes of selecting and classifying these resources together with data about the strengths and weaknesses in meeting operational requirements and the general priorities and preference of industry uses of existing comparable construction technologies could be used to train a probabilistic neural network system to assist in the selection and classification of these expensive resources.

## Conclusion

In view of the extensive applications of ANN analysis, this approach will continue to make impressive gains. In fields requiring analytical tools for integration into the decision-making process, the predictive nature of ANN analysis, in particular for non-linear cases, will be an invaluable assistant. The complex nature of construction management will be fertile ground for additional growth.

## References

- Asakawa, K. and Takagi, H. (1994) Neural networks in Japan, *Communication of the ACM*, **37**(3), 107–13.
- Bailey, D. and Thompson, D. (1990) How to develop neural-network applications?, *AI Expert*, **June**, 38–47.
- Barron, A.R. and Barron, R. (1989) Statistical learning networks: a unifying view, in *Computing Science and Statistics: Proceedings of the 20th Symposium on the Interface*, Wegman, E.J., Gantz, D.I. and Miller, J.J. (eds) pp.192–202.
- Boussabaine, A.H. (1995) Neural networks for productivity forecasting, *Proceedings of the 12th International Symposium on Automation and Robotics in Construction Conference*, June, Poland, pp.375–83.
- Boussabaine, A.H. and Cheetham, D.W. (1995) Artificial neural networks as a tool for predicting project durations, in *Proceedings of the Association of Researchers in Construction Management*, September, York, pp.551–8.
- Chao, L. and Skibniewski, M.J. (1994) Estimating construction productivity: neural network based approach, *ASCE Journal of Computing in Civil Engineering*, **8**(2), 221–33.
- Fausett, L. (1994) *Fundamentals of Neural Networks: Architecture, Algorithms and Applications*. Prentice-Hall International, Englewood Cliffs, NJ.
- Flood, I. (1989) A neural network Approach to the sequencing of construction tasks, *Proceedings of the 6th International Symposium on Automation and Robotics in Construction*, Austin, Tex, USA, 204–11
- Flood, I. and Kartam, N. (1994a). Neural networks in civil engineering I: principles and understanding, *ASCE Journal of Computing in Civil Engineering*, **8**(2), 131–48.
- Flood, I. and Kartam, N. (1994b) Neural networks in civil engineering II: principles and systems and application, *ASCE Journal of Computing in Civil Engineering*, **8**(2), 149–62.

- Hall, C. (1992) Neural net technology ready for prime time?, Institute of Electrical and Electronic Engineers (IEEE) *Expert*, **December**, 2–4.
- Hammertrom, D. (1993) Neural networks at work, IEEE *Spect.*, **June**, 26–32.
- Hush, P.R. and Horne, B.G. (1993) Progress in supervised neural networks, *IEEE Signal Processing Journal*, **January**, 8–39.
- Kamarthi, S., Sanvido, V. and Kumara, R. (1992) Neuroform – neural network system for vertical formwork selection, *ASCE Journal of Computing in Civil Engineering*, **6** (2), 178–99.
- Kohonen, T. (1988) An introduction to neural computing, *Neural Networks*, **1**(1), 3–16.
- Levin, E., Tishby, N. and Solla, S.A. (1990) A statistical approach to learning and generalization in layered neural networks, *Proceedings of the IEEE*, **78**(10), 1568–74.
- Lippman, R.P. (1987) An introduction to computing with neural nets, *IEEE, Ass. Press, Journal*, **4**(2), 4–22.
- MacKay, D.J.C. (1992) A practical Bayesian framework for back-propagation networks, *Advances in Neural Information Processing Systems*, **4**, 839–46.
- Masters, T. (1993) *Practical Neural Networks Recipes in C++*. Academic Press, London.
- Mirza, B. and Fisher, D.J. (1994) Neuromodex – neural network system for modular construction decision making. *ASCE Journal of Computing in Civil Engineering*, **8**(2), 221–33.
- Moody, J.E. (1991) Note on generalization and regulation and architecture selection in non-linear learning systems. Neural networks for signal processing, in *Proceedings of the 1991 IEEE Workshop*, Juang, B.H., Kung, S.Y. and Kamm, C.A. (eds) IEEE Press, pp.1–10.
- Moody, J.E. (1992) The effective number of parameters: an analysis of generalization and regulation in non-linear systems, *Advances in Neural Information Processing Systems*, **4**, 847–54.
- Nerrand, O., Roussel-Rugot, P., Personnaz, L., Dreyfus, G. and Marcos, S. (1993) Neural networks and non-linear adaptive filtering: unifying concepts and new algorithms, *Neural Computations*, **5**(2), 165–99.
- Rumelhart, D., Widrow, B. and Lehr, A. (1994) The basic ideas in neural networks, *Communication of the ACM*, **37**(3), 87–91.
- Widrow, B., Rumelhart, D. and Lehr, A. (1994) Neural networks: application in industry, business and science, *Communication of the ACM*, **37**(3), 93–105.
- Williams, T.P. (1994) Predicting changes in construction cost indexes using neural networks. *ASCE Journal of Construction Engineering and Management*, **120**(2), 306–20.