Managing Construction Knowledge in Patterns: A Neural Network Approach

T HEGAZY, O MOSELHI and P FAZIO*

ABSTRACT

Neural networks are AI-based computational tools with powerful capabilities of effective capturing and re-use of domain knowledge that are inherently implicit. This paper describes the modelling capabilities of neural networks with respect to construction problems, emphasizing the advantages associated with their representation of knowledge in the form of patterns. Several aspects related to proper management of knowledge are addressed for the purpose of developing practical and more reliable neural network models of complex construction problems. These aspects include: 1) problem structuring and patterns formation; 2) knowledge acquisition and data validation; 3) preparation and transformation of acquired data; and 4) analysis and interpretation of network state of knowledge. Guidelines pertaining to these aspects are provided along with considerations for modelling with noisy data and under high degree of uncertainty. The issues discussed are illustrated through a case study of a neural network for bidding decision support, developed based on knowledge acquired from contractors in Canada and the U.S. The case study demonstrates neural network modelling and illustrates the benefits gained through better management of acquired knowledge.

Key Words

neural networks; knowledge acquisition; construction; information technology; bidding strategy

PATTERN RECOGNITION IN CONSTRUCTION PRACTICE

Reasoning, deduction, and pattern recognition are among the fundamental aspects of human intelligence. Among those, humans exhibit phenomenal abilities to recognize patterns of information in the environment and respond to them in a speedy and effortless manner, even under extremely difficult conditions (Rothman, 1992). Obvious human experience that involve patterns include the recognition of speech utterances and the understanding of images such as handwriting, despite major distortions or omissions. This outstanding ability of humans, observed also in the decision-making capability of domain experts, have stimulated growing research and developments in statistical pattern recognition and AI systems such as neural networks (NNs). The

* Centre for Building Studies, Concordia University, Montreal, Canada H3G 1M8.

interest in these areas of research has several motivations, including: 1) capturing of scarce and implicit domain knowledge; 2) developing intelligent machines with human-like abilities; and 3) developing effective decision supports for complicated real-life problems. Background material regarding neural network variations, characteristics, and mathematical formulations is well documented elsewhere (Moselhi *et al*, 1991a; Pao, 1989).

Recently, NNs have been suggested for modeling a number of construction engineering and management problems that are solved in practice based primarily on holistic analogy and "gut feeling" rather than detailed deduction and reasoning (Moselhi *et al*, 1991a; 1992). Some construction examples include the prediction of productivity levels achievable under a particular job site conditions, the assignment of a percent markup before submitting the bid price, and day-to-day decisions regarding the allocation of resources, minimizing idle times, and solving disputes. Neural networks derive their analogy-based and pattern recognition capabilities from a process of learning a set of examples representing previous encounters of a certain problem. NNs utilize these holistic examples (without their underlying logic), as patterns, to simulate complicated decision processes and their related knowledge. Much of neural networks' power, as such, stems from their representation and processing of knowledge in the form of patterns. A pattern consists of a group of factors that distinctively characterize the cause (situation) or the effect (consequent decisions) associated with a particular problem. In a typical bidding situation, for example, the cause pattern, on one hand, consists of a number of risk-related factors that influence contractors' decision of an optimum percent markup to add to their cost estimates. The effect pattern, on the other hand, incorporates other factors associated with the decision made and the consequences of such decision on winning/losing the job and actual profitability attained. The advantages of such type of knowledge representation is the holistic and problem-less manner by which knowledge can be acquired from experts and further managed to develop practical models of complex problems. Given a sufficient number of cause and associated effect patterns that well represent the domain knowledge, neural networks can be trained to extract and generalize the implicit relationships linking causes to effects. The trained network, thus, becomes able to predict the effect (solution) given only the causes pertaining to a future or new situation.

KNOWLEDGE MANAGEMENT USING NEURAL NETWORKS

It is apparent from the aforementioned discussion that the practicality and effectiveness of neural network models are highly dependent on: 1) patterns structure and the relevance of their constituent factors to the problem; 2) sufficiency and representativeness of the knowledge acquired to the domain;

and 3) suitability of the method used for transforming the data into format usable by neural networks. This is in addition to a suitable neural network paradigm, an optimum network configuration, and sufficient training. Despite the importance of proper management of knowledge to the powerful modelling capabilities of neural networks, however, the task of structuring the problem patterns and processing their numeric and non-numeric data is a highly problem-dependent task that is not simple and straightforward, and lacks structure and organization.

In an effort to guide the process of capturing of construction knowledge and its successful utilization in the development of neural network models, the following aspects are addressed: 1) problem structuring and patterns formation; 2) knowledge acquisition and data validation; 3) preparation and transformation of acquired data; and 4) analysis and interpretation of network state of knowledge. These issues are discussed in detail in the following sub-sections.

Problem Structuring and Pattern Formation

It is assumed that there is at hand a problem that lends itself well to neural network modelling. As a case study, the problem of deciding a percent markup in competitive bidding situation is considered and the backpropagation NN paradigm (Rumelhart *et al*, 1986) is selected as most suitable for its modelling (Moselhi *et al*, 1991b). In general, one approach to validate an application is to compare its characteristics to those of potential neural network applications. The selection of a neural network paradigm can also be based on a comparison of the application requirements to the paradigm capabilities and limitations (Moselhi *et al*, 1991a).

Problem analysis, structuring, and patterns formation represent major steps in the design of a neural network model. Problem analysis consists of (Bailey and Thompson, 1990): (a) identifying all data that in any way relate to the application area; (b) removing data sources that are regarded as peripheral or unreliable; (c) filtering out data sources that are impractical for technical or economical reasons; and (d) exploring methods of combining or preprocessing data to make it more meaningful (eg, ratios are often more significant than numbers and provide the relevant relationship explicitly). Based on such analysis, the problem governing attributes, representing the system input, and the problem output attributes, representing the system conclusions, can be identified. These attributes, if well selected, normally represent a high order of abstraction from the data. They are independent with little correlation between any two of them. This ensures that the attributes adequately define the problem and require less network design and implementation effort. For the present case study, characteristic factors that need to be considered in formulating a successful bidding strategy were identified through a survey conducted among the top 400 contractors in the U.S. (Ahmad and Minkarah, 1988). Percent markup decisions were predominantly attributed to the group of factors that arise from uncertainty. Accordingly, these factors are grouped to form the risk pattern associated with individual projects.

Once the problem input and output attributes are identified, a number of feasible problem structures has to be determined. This depends on several factors including: (a) the number of input and output attributes identified and their data types; (b) constraints on the acquisition of training examples (number, time needed, and cost needed); (c) clarity or fuzziness of the problem; and (d) availability of domain heuristic knowledge that may guide the search. A direct structure is often done by presenting all inputs at the network input buffer and all outputs at the output layer. This structure works well for small size problems with definitely relevant and consistent data (a non-consistent data would include mixed short term and long term data, for instance). However, more often this is not the case. Problems vary in size (number of input and output attributes), complexity, data type, and solution requirements. There is no one single approach that can be used for direct structuring of a problem. It is important, however, to identify more than one structure since successful implementation of any structure is not guaranteed to suit the problem. This step is necessary before conducting knowledge acquisition since alternate structures may require additional information to be elicited.

For the markup problem, two structures were determined at that stage: 1) a direct structure with one large neural network having all the identified factors as inputs (Figure 1); 2) a five-network hierarchical structure with the factors divided among four small sub-networks that are linked to an additional global network (Moselhi *et al*, 1991b). The hierarchical structure was sought on the assumption that it may require less number of training examples to produce the same level of accuracy of the single-network structure. Such structure takes advantage of an inherent hierarchical structure of the problem itself.

In the single-network structure of Figure 1, thirty input attributes represent the project environment pattern and seven output attributes represent the output pattern desired to be predicted by the model for new situations. Not only a markup value need to be estimated but it is also required to give an indication about the implications of the project pattern and the estimated markup on: the chances of win/lose; difference in ($) between the winner and second bid; project potential for change orders; project potential for claims; duration extension; and actual project profitability. A pattern, as shown in Figure 1, is constructed from a chess-like grid of the network attributes and the possible values of each. The possible values and their associated literal meanings are shown in Table 1. The black spots on the grid represent the attributes' values in a particular training example.

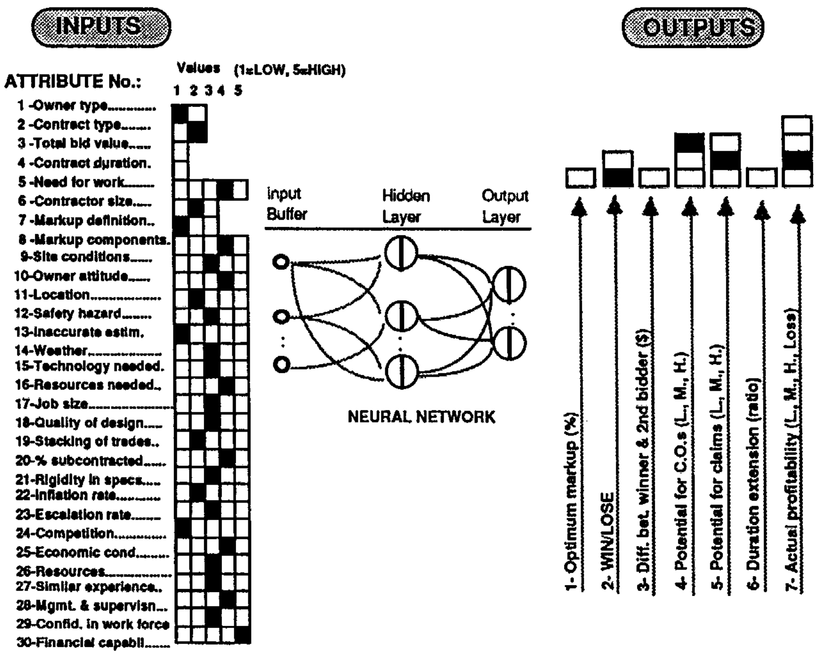

Figure 1. A Single Neural Network for the Markup Problem

Knowledge Acquisition and Data Validation

Once there is a clear idea about some feasible structures and the information needed to be elicited, necessary knowledge could be acquired and the data validated. In this step, focus is on gathering as much training data as possible within practical limits of time, money, and computer resources. More important than the quantity of examples is their quality and representativeness of the domain. A good training set should contain routine, unusual, and boundary-condition cases (Bailey and Thompson, 1990). One measure of the data's representativeness is the breadth of the problem covered by the training cases, including the different types of patterns in a pattern-classification problem and variety of significant cases in a decision-making process. Potential data sources include historical records, test data, case studies, instrument readings, simulation results, and hypothesized results.

In the present case study, necessary knowledge was acquired through a questionnaire survey prepared based on initial interviews conducted among a

Table 1: Description of Markup Attributes

Attribute	Possible values	Scaling operation
INPUTS:		
1- Owner type	- 1: Public; & 2: Private.	- (Value - 1).
2- Contract type	- 1: Lump Sum; 2: Unit Price;	- (Value - 1).
3- Total bid value	- Real Number ($ millions).	- (Value / 20).
4- Contract duration	- Real number (months).	- (Value / 10).
5- Need for work	- 1: Low to 5: High.	- (Value / 5).
6- Contractor size	- 0: Small; 1: Medium; 2: Large.	- (Value / 2).
7- Markup definition	- 0: a % of (DIR*); 1: a % of (DIR+P.O.**); 2: a % of (DIR+P.O.+G.O.***).	- (Value / 2).
8- Markup components	- 0: Profit only; 1: Profit+Cont.****; 2: Profit+G.O.; 3: Profit+G.O.+Cont.; 4: Profit+P.O.+G.O.+Cont.	- (Value / 4).
9 to 30	- 1: Low to 5: High.	- (Value / 5).
OUTPUTS:		
1- Markup	- Real number (%)	- (Value / 50).
2- Win/lose	- 0: Win; 1: Lose.	- (Value).
3- Diff. between winner and second bidder	- Real number ($ * 10E-5).	- (Value / 10).
4- Potential for C.O.s	- 1: High; 2: Medium; 3: Low.	- (Value / 5).
5- Potential for Claims	- 1: High; 2: Medium; 3: Low.	- (Value / 5).
6- Duration extension	- Real number (Ratio = Actual duration / contract duration).	- (Value / 2).
7- Actual profitability	- 1: High; 2: Medium; 3: Low; 4: loss.	- (Value / 5).

* DIR = Direct cost.
** P.O. = Project Overhead cost.
*** G.O. = General Overhead cost.
**** Cont. = Contingency.

number of Montreal area contractors. In an effort to maximize homogeneity and minimize variability, questionnaires were sent to contractors in Canada and the U.S. who appeared to: 1) work as general contractors; 2) specialize mainly in building construction; and 3) obtain a large percentage of their work based on competitive bidding. These criteria were used later to qualify the respondents, ensuring some commonalities among the qualified participants and limiting the markup estimation model to a well defined domain. The survey elicited information on the firm's policy with regard to bidding strategy. This helps in making necessary adjustments to the training examples obtained form different contractors, accounting for the different markup definitions adopted by the participants. Alternatively, as shown in Figure 1, two input attributes (7 and 8) were included to ensure generic model. The survey, furthermore, elicited a number of complete bidding examples from the firm's past records, including successful and unsuccessful bidding situations. For each project, respondents were asked to provide complete data about:

- assessment of the various factors affecting markup, using a score from 1 to 5.
- contractor's bid data (total bid price and percent markup decided);
- bid outcomes (win/lose and difference in ($) between the winner and second lowest bidder); and
- if successful, after construction outcomes (intensity of change orders experienced; intensity of claims experienced; duration extension depicted; and level of actual project profitability attained).

From the survey, 78 responses were qualified as they meet the criteria mentioned above. With respect to the respondents' bidding examples, the total number of cases received were checked for completeness and suitability as training and testing (validation) examples. Based on this scrutiny, 65 and 7 project examples were suitable for training and testing the single-network, respectively. It should be noted, however, that although the learning examples used were complete in terms of information content, it was assumed that they are not free of biased judgements and inconsistencies. The contractor providing the historical example could have been biased towards his recent experience. Such types of errors represent noisy training data to the neural network models.

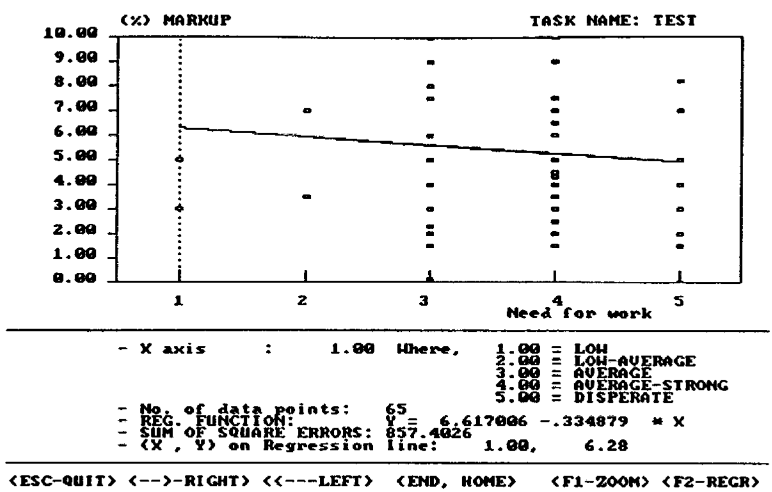

Figure 2. Relationship Between Markup and Need for Work

In order to validate the information content of the acquired examples, a simple test was conducted on the training data. The test is to examine the relationship between the data pertaining to an input attribute and that of an output attribute, depicted in all the training examples. These relationships (or general trends) are established through simple regression analysis and then compared with known industry heuristics to test the information content of the training examples and their compatibility with current industry practice. Examples of these relationships are presented in Figures 2 to 3 showing the results of such analysis on the 65 training examples of the single network model. It is clear that the training examples, exhibit trends that are in good agreement with industry heuristics. This includes:

- Percent markup decreases with the need for work (Figure 2).
- Potential for claims increases with uncertainty due to site conditions (Figure 3).

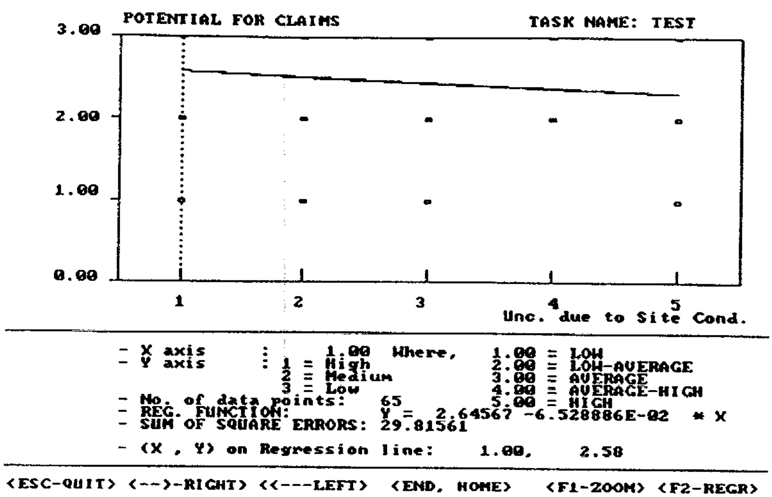

Figure 3. Relationship Between Potential for Claims and Uncertainty Due to Site Conditions

This simple analysis shows that, despite the noise inherent in the data, the training examples used, provide a good training environment that reflects current industry practices and accounts for many quantitative and qualitative factors used by contractors in this domain.

Preparation and Transformation of Acquired Data

Once enough data is elicited, it may need to be prepared and transformed into another format to be meaningful to a neural network. Transformation and scaling of data are techniques that are commonly used to convert input and output data into either binary or continuous formats (Moselhi *et al*, 1992). Binary-value transformation is done by assigning ones for the applicable attributes and zeros elsewhere on the grid of a certain pattern (*eg*, the black spots in Figure 1 become ones and elsewhere become zeros). A binary vector is then constructed from all the zeros and ones in their sequence on the grid and fed to the input buffer of the designed neural network. The number of processing elements (PEs) in the input buffer of that network is determined by the number of elements in the transformed vector. In the continuous-value transformation, the pattern is transformed into a vector of real numbers, each assigned for a given input attribute. The numbers could be made to represent the values of the scores assigned to the attributes. Alternatively, an index identifying the position of the value on the grid could be used. The real numbers assigned could then be pre-processed by a number of possible ways, including: scaling, normalization, and function processing (*eg*, sine). These methods have been described in recent literature providing guidelines for preparing the data and preventing problems associated with the magnitude and variability among the input attributes (Crooks, 1992; Knaus, 1991; Lawrence, 1991; Bailey and Thompson, 1990). Adopting a certain method depends on the characteristics of the neural paradigm used (*eg*, the existence of a bias node), and the type of data available. It is noted that the transformed output vector, in most cases, is constructed from attribute values confined (*ie*, scaled) to a range from 0 to 1.0, as dictated by the transfer function of the network PEs. In continuous transformation, the number of PEs in both the input buffer and the output layer have to be set as the number of attributes that represent the input and output patterns, respectively.

For the example problem at hand, the row data were suitably scaled (Table 1 shows the scaling operation used for the different attributes). These scaling operations transform the row data into continuous, rather than binary, values suitable for neural network processing. The scaled data were then utilized to implement, train, and test the two neural network structures. As a result of the implementation process, the single-network structure was selected for the markup model since it achieved better performance (less estimation errors as opposed to the hierarchical structure). The network was configured to have an input buffer of 30 elements (accepting the inputs), a one hidden layer of 30 PEs, and output layer of 7 PEs (producing the outputs). Details of the implementation process are described elsewhere (Moselhi *et al*, 1991b), along with the heuristics used to overcome the development problems encountered.

Analysis and Interpretation of the Network State of Knowledge

The major property that deems neural networks superior over conventional algorithmic and other AI-based systems is their ability to learn. Once a neural network is trained, the weights and bias values encode the network's state of knowledge gained during training. Utilizing the trained network on new cases is a matter of mathematical manipulation of these values with the outputs produced almost instantaneously. Usually, no logical inferences or explanations are involved in providing the solution, making the solution process non-transparent and non logically traceably. This has contributed to the "Black Box" image of neural networks which researchers have been trying to demystify.

Several researchers have proposed innovative techniques that facilitate the interpretation of the network weights and the understanding of their underlying logic (*eg*, Garson, 1991; Howell, 1990; Touretzky and Pomerleau, 1989). The simplest and most interesting effort to the scope of this study is the technique presented by Garson (1991) to interpret the relative importance of each input attribute to the conclusions reached by the network. The technique evaluates the relative importance of an input attribute (V) through a process of *partitioning output layer connection weights*, for a network with one hidden layer, into components associated with each input attribute. The technique uses the absolute values of all weights, without considering the PEs' biases into the computations. The process can be illustrated as shown in Figure 4, where a weight connected to the output layer O_J can be divided into components each corresponding to one of the input attributes. For a hidden PE (J) the component of O_J associated with V is proportional to the weight incoming from attribute V (I_{VJ}) in relation to the sum of weights incoming from all attributes. This can be expressed as:

$$\textit{Component of } O_J \textit{ associated with } V = \frac{I_{VJ}}{\sum_{K}^{n_V} I_{VJ}} \cdot O_J \tag{1}$$

Performing this calculation for all the hidden layer and summing the shares of each input attribute results in a total share or score for each attribute. The relative importance of each attribute can then be calculated as a percentage of the sum of scores (Eq. 2). The method, though simple, stands in sharp contrast to misleading views of neural networks as "Black Boxes" whose iterative processes are beyond human comprehension, even if predictions are good.

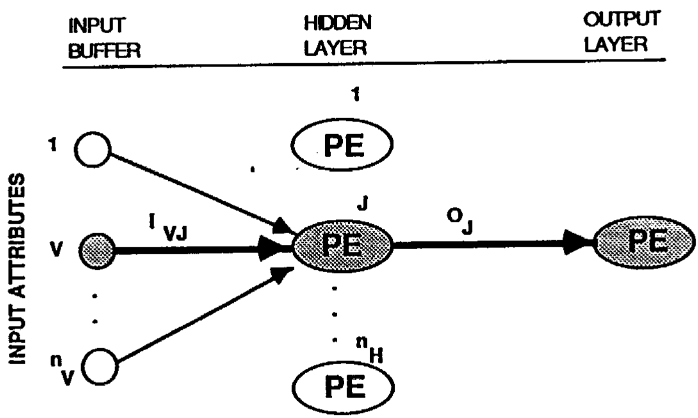

Figure 4. Partitioning of Output Layer Weights into Components Associated with Input Attributes

$$\frac{\sum_{J}^{n_k}\left(\frac{I_{VJ}}{\sum_{K}^{n_V} I_{VJ}} \cdot O_J\right)}{\sum_{i}^{n_V}\left(\sum_{J}^{n_k}\left(\frac{I_{VJ}}{\sum_{K}^{n_V} I_{VJ}} \cdot O_J\right)\right)} \tag{2}$$

With respect to the case study, the technique of Garson was applied. Based on the technique, the relative importance of the 30 input attributes to the individual outputs are calculated for the network that is selected for the markup model (Table 2). From the results obtained, it can be seen that:

- All the input attributes are highly relevant to the model formulation. The values for the relative importance vary between 1.2 for a least important to 5.5 for the most important input attribute. Thus, none of the input attributes could have been eliminated in the model formulation.
- The values for the relative weights determine the input attributes that govern the conclusions derived by the individual outputs. For instance, the network showed that "Competition", "Contract duration", and "Uncertainty due to owner attitude" are the most influential factors on the % markup estimate.
- A certain input attribute impacts the individual system outputs with varying degrees. For instance, "Need for work" influences more the win/loose possibility, % markup, duration extension, potential for change orders,

potential for claims, actual profitability, and the ($) difference between the winner and second bidder.

The results produced by this technique can be further utilized to detect causal rules and provide explanation facilities for neural network models.

Table 2: Relative Importance of the Markup Network Input Attributes

Input Attributes	OUTPUT ATTRIBUTES* (1)	(2)	(3)	(4)	(5)	(6)	(7)
1. Contract type	1.9	3.0	2.3	2.7	3.2	2.3	2.7
2. Owner type	4.5	4.6	4.5	4.8	4.9	4.3	4.2
3. Competition	4.6	4.8	5.8	4.4	4.1	5.0	4.3
4. Need for work	3.5	3.7	2.8	3.1	2.9	3.2	2.8
5. Total bid	4.3	4.7	5.5	4.1	4.0	4.9	5.5
6. Contract duration	4.6	4.3	4.0	4.7	4.5	4.3	3.9
7. Company size	3.9	3.4	3.7	3.6	3.8	3.6	3.9
8. Markup definition	3.7	4.0	3.7	4.0	4.2	3.6	3.7
9. Markup components	4.0	3.7	4.5	4.3	4.2	4.0	3.5
10. Site conditions	4.1	4.6	3.3	4.4	4.6	4.6	4.8
11. Owner attitude	4.7	3.8	4.0	4.1	4.0	4.4	4.2
12. Project location	4.4	2.8	4.1	4.0	3.9	3.8	4.0
13. Safety hazard	4.4	4.4	5.0	5.0	5.3	4.5	5.1
14. Inaccurate estimate	3.1	3.1	3.2	3.1	2.8	2.6	2.9
15. Weather sensitivity	3.8	3.3	2.2	3.6	2.6	3.2	3.3
16. Technology needed	1.2	1.9	1.5	1.4	1.7	1.4	1.7
17. Resources needed	2.5	2.7	2.9	2.8	3.4	3.3	3.4
18. Job size	2.2	2.7	2.6	2.5	2.2	3.0	2.7
19. Quality of Drawings	4.2	3.2	3.4	3.5	3.0	3.6	3.3
20. Stacking of Trades	3.6	3.3	3.5	3.0	3.2	3.6	3.6
21. % Subcontracted	3.0	3.2	3.5	3.0	2.8	2.8	2.8
22. Rigidity in Specs.	4.0	4.0	4.1	3.7	3.0	3.9	3.7
23. Inflation rate	3.4	3.9	3.0	3.8	2.7	3.7	4.0
24. Escalation rate	2.0	2.3	1.8	1.9	2.5	2.0	2.0
25. Economic growth	2.8	2.4	2.2	2.3	2.5	2.4	2.0
26. Resources available	2.3	2.2	2.4	2.5	3.0	2.2	2.6
27. Firm Expertise	2.2	1.9	1.8	2.2	2.2	2.3	2.0
28. Mgmt. experience	2.7	1.9	1.9	2.2	3.0	2.7	2.3
29. Confidence in work	2.3	2.5	3.5	2.4	2.3	2.2	2.2
30. Financial ability	2.3	3.7	2.9	2.8	3.4	2.6	3.0
	100	100	100	100	100	100	100

* (1): Markup Estimate (%). (2): Win/Lose. (3): $ Difference. (4): Potential for C.O.'s. (5): Potential for claims. (6): Duration extension. (7): Project profitability.

SUMMARY AND CONCLUDING REMARKS

The importance of pattern recognition to human problem solving abilities is briefly described and neural networks are used to provide pattern recognition capabilities in the modelling of complex construction problems. As opposed to traditional AI techniques, neural networks process implicit construction knowledge in the form of patterns that can be extracted from experts with relative ease. In order to facilitate proper management of the captured knowledge and the successful development of neural network models, guidelines are provided pertaining to several aspects: 1) problem structuring and patterns formation; 2) knowledge acquisition and data validation; 3) preparation and transformation of acquired data; and 4) analysis and interpretation of the network state of knowledge. The modelling issues discussed are illustrated through a case study of a neural network for bidding decision support. It is apparent from the study that, unlike most other software systems, the performance of a neural network is as much determined by the data in its experience as by the algorithms used to build it. The developments made in this study provide insights into proper NN design and implementation, irrespective of the neural paradigm used. The guidelines proposed could be readily adopted in developing neural network models in other construction domains where solutions are generated based primarily on analogy and traditional algorithmic tools may prove inadequate.

References

Ahmad, I and Minkarah, I (1988), Questionnaire Survey on Bidding in Construction. *Journal of Management in Engineering*, ASCE, Vol 4, No 3, pp 229-243.

Bailey, D and Thompson, D (Sept. 1990), Developing Neural-Network Applications. *AI Expert*, pp 34-41.

Crooks, T (July 1992), Care and Feeding of Neural Networks. *AI Expert*, pp. 36-41.

Garson, G (April 1991), Interpreting Neural-Network Connection Weights. *AI Expert*, pp 47-51.

Howell, J (Nov. 1990), Inside a Neural Network. *AI Expert*, pp 29-33.

Knaus, R (Sept. 1991), Putting Knowledge into Nets. *AI Expert*, pp 19-25.

Lawrence, J (Nov. 1991), Data Preparation for a Neural Network. *AI Expert*, pp 34-41.

Moselhi, O, Hegazy, T and Fazio, P (1992), Potential Applications of Neural Networks in Construction. *Canadian Journal of Civil Engineering*, Vol 19, pp 521-529.

Moselhi, O, Hegazy T and Fazio, P (1991a), Neural Networks as Tools in Construction. *Journal of Construction Engineering and Management*, ASCE, Vol 117, No 4, pp 606-625.

Moselhi, O, Hegazy, T and Fazio, P (1991b), Markup Estimation Using AI Methodology. In *AI and Civil Engineering*, B.H.V. Topping (ed.), Proceedings of the 2nd International Conference on the Application of Artificial Intelligence Techniques to Civil and Structural Engineering, Oxford, England, Civil-Comp Press, pp 257-266.

Pao, Y H (1989), *Adaptive Pattern Recognition and Neural Networks.* Addison-Wesley Publishing Company Inc.

Rothman, P (Oct. 1992), Syntactic Pattern Recognition. *AI Expert*, pp 41-51.

Rumelhart, D E, Hinton, G E and Williams, R J (1986), Learning Internal representations by Error Propagation," In *Parallel Distributed Processing: Explorations in the Microstructures of Cognition*, Rumelhart and McClelland (eds.), Vol 1, MIT press, Cambridge, Mass.

Touretzky, D and Pomerleau, D (Aug. 1989), What's Hidden in the

Hidden Layers ?. *BYTE*, pp 227-233.

c. 1993, Management of Information Technology for Construction, K. Mathur et al (Eds), World Scientific Publishing Co., Singapore.