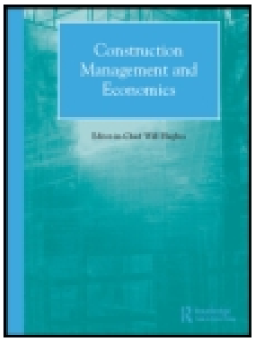

Construction Management & Economics

ISSN: 0144-6193 (Print) 1466-433X (Online) Journal homepage: https://www.tandfonline.com/loi/rcme20

A neural networks approach for cost flow forecasting

A.H. Boussabaine & A.P. Kaka

To cite this article: A.H. Boussabaine & A.P. Kaka (1998) A neural networks approach for cost flow forecasting, Construction Management & Economics, 16:4, 471-479, DOI: 10.1080/014461998372240

To link to this article: https://doi.org/10.1080/014461998372240

Published online: 21 Oct 2010.

Construction Management and Economics (1998) **16**, 471–479

A neural networks approach for cost flow forecasting

A.H BOUSSABAINE and A.P KAKA

School of Architecture and Building Engineering, The University of Liverpool, Liverpool, P.O. 147, Liverpool L69 3BX, UK

Received 19 August 1996; accepted 20 March 1997

Artificial neural networks, which simulate neuronal systems of the brain, are useful methods that have attracted the attention of researchers in many disciplinary areas. They have many advantages over traditional methods in situations where the input-output relationship of the system under study is not explicitly known. This paper investigates the feasibility of using neural networks for predicting the cost flow of construction projects, explains the need for cost flow forecasting, and demonstrates the limitation of the existing models. It then introduces neural networks as an alternative approach to those mathematical and statistical methods. The method used in collecting data and modelling the cost flow is described. Results of the testing are presented and discussed.

Keywords: neural networks, cost flow, forecasting, artificial intelligence, cost modelling

Introduction

The construction industry suffers the largest number of bankruptcies of any sector of the economy, with many companies failing because of poor financial management, especially inadequate attention to cash flow forecasting. Project cost flow forecasting has been inspiring the theoreticians and the practitioners for a long time. Over the years, extensive alternative modelling approaches have been developed. Practitioners speak of ample opportunities at various points in time where forecasting is useful or beneficial. Planning and prediction of cash flow needs and arranging support finance are fundamental management responsibilities.

Many models have been developed to assist contractors in their cost flow forecasts. The majority of these have been based on standard cost flow *S*-curves, developed using actual past construction projects. The accuracy of a cash flow forecast generated from standard cost curves depends on whether the adopted *S*-curve accurately represents the project to be constructed. A large number of mathematical and statistical models have been applied to forecast typical *S*-curves (Hardy, 1970; Balkau, 1975; Bromilow and Henderson, 1977; Drake, 1978; Hudson, 1978; Peer, 1982; Oliver, 1984; Singh and Woon, 1984 ; Miskawi, 1989; Khosrowshahi, 1991; Navon, 1995). Bromilow and Henderson (1977) used four general building projects to develop their value *S*-curve. Hardy (1970) analysed 25 different types of project and found that there was no close correlation between the values considered, even when separating them into different categories. Oliver (1984) analysed projects collected from three construction companies. He concluded that, although the number of projects analysed was statistically small, construction projects are unique and follow such diverse routes that value curves based on historical data are not capable of providing the accuracy needed for individual project control. Drake (1978) collected projects from regional health authorities and further classified them into different categories. Unfortunately, no figures were published of the number of projects analysed or the level of accuracy of the fitted functions. Singh and Woon (1984) fitted envelopes of *S*-curves for high-rise commercial, industrial and residential buildings. The envelopes contained half of the values considered in each category. Although they did not quote the number of projects analysed, the graphs plotted

0144–6193

through the scatter points show that the sample was small and the values outside the envelopes were not relatively close. Most of these models are based on regression techniques whereby the best fit is sought. For many construction projects the factors that determine the shape of an *S*-curve are very difficult to quantify and may not lend themselves to curve fitting since the needed, representation cannot nicely fit into a quantitative description. Further, often it is not clear which factors the project *S*-curve depends on and the degree of effect such factors may have. The relationship between inputs and outputs (in the regression models) are very complex since there could be some unknown combined effects. One of the difficulties in these models is accounting for the existing correlation among cash flow variables modelled as random variables, even if the correlation among random variables is known. Hence, it is difficult to perform such multi-attribute nonlinear mappings by using a regression model. Also the regression models lack the ability to learn by themselves, generalize solutions, and respond adequately to highly correlated, incomplete, or previously unknown data (Shaw, 1992). For this type of environment neural networks are superior to the mathematical and statistical models for forecasting projects cost flow. Neural networks are suitable for solving complex cognitive problems of cost flow (Boussabaine, 1996). It has been proved that problems with multi-attributes can be solved better by neural networks than by conventional methods (Masters, 1993). Neural networks are suited to such problems because of their adaptivity owing to their structure; i.e. nonlinear activation functions. It has been proved that any arbitrarily irregular patterns can be mapped by artificial networks (Widrow *et al.*, 1994). The artificial neural approach to cost flow forecasting uses an approximation technique that handles non-unique cases either by returning one of the possible solutions, or by taking an average over all possible solutions. This generally results in qualitatively good results, as will be demonstrated through out this paper. Project cost flow provides a data rich environment for neural network development. A novel forecasting technique that is capable of making a large number of computations in a data-rich environment and of adapting and learning to track the patterns underlying the cost flow is presented and described in this paper.

Overview of neural networks

Artificial neural networks (ANN) offer an approach to computations that is different from conventional analytical methods. There is a diverse range of ANN models in terms of topology and mode of operation (Fausett, 1994). Figure 1 illustrates a simple three-layer neural network. 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) provide a statistical interpretation of the methods used to train ANNs. Nerrand *et al.* (1993) show that ANNs can be considered as general nonlinear filters that can be trained adaptively. A clear summary of the feed-forward network and the back-propagation models can be found in Karnarthi *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. Unlike expert systems and traditional modelling methods, where knowledge is made explicit in the form of rules, neural networks generate their own rules by learning from examples (Gallant, 1993).

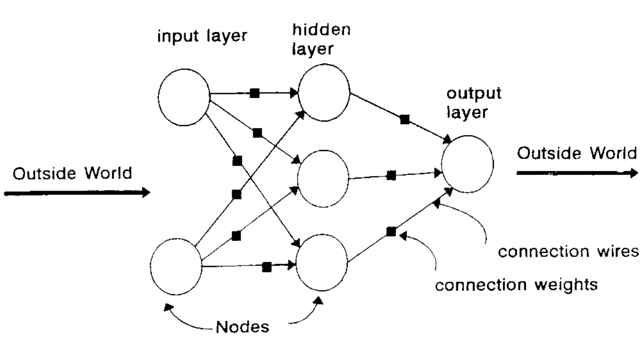

Figure 1 Simple three-layer neural network

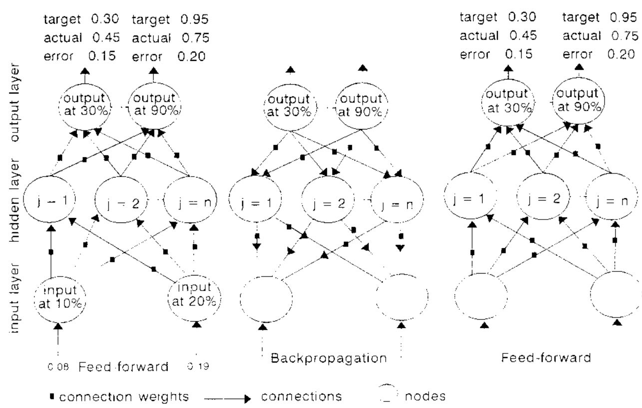

Figure 2 Typical neural network structure (net 2 at 20% completion)

In theory, a network can be put to work in any application where a substantial amount of data is used to predict the outcome. The problem of learning in neural networks is simply the problem of finding a set of connection strengths that allow the network to carry out the desired computation (Hammertrom, 1993). The learning method used in this project is back-propagation (BP). Back-propagation is the most widely and successfully used algorithm in neural networks. The main mechanism in a BP network is to propagate the input forward through the layers to the output layer and then to propagate the errors back through the network from the output layer to the input layer (Boussabaine, 1996), as demonstrated in Fig. 2. Input data (i.e. progress periods) are presented to the ANN model for each sample, with their forecasting targets. The system is then prompted to learn the underlying patterns between these sets of input and output. The working principles of a network trained with BP to predict the cost flow for 30% to 90% completion of projects (see the next section) are outlined in Fig. 2. The process of feed-forward and BP shown in Fig. 2 continue until the weight (knowledge) is stabilized (Boussabaine, 1996) and the learning error is reduced to 0.001%. After this the neural model is tested to predict the outcome of cost flow curves not seen previously by the system. Each of the neurons in this model is characterized by (Rumelhart *et al.*, 1994): (i) an activity level (representing the state of polarization of the neuron); (ii) an output value (representing the firing state of the neuron); (iii) a set of input connections (representing synapses on the cell and its dendrite); (iv) a bias value (representing an internal resting level of the neuron); and (v) a set of output connections (representing a neuron's axonal projections). Each of these attributes of the neuron is represented mathematically by real numbers, and each connection has an associated weight (synaptic) which determines the effect of incoming input on the activation level of the neuron. The weights may be positive (excitatory) or negative (inhibitory). The output of a neuron is determined by the activities of its input wires. A back-propagation neuron transfers its inputs as follows:

$$\text{Output(node)}_i = \sigma[\Sigma w_{ij} x_j(t) - \beta_\text{I}]$$

where σ is the sigmoid function, w_{ij} is the strength of the connection (weight) from node j to node i, x_j is the output value of node j; and β_I is the node threshold value. So, when a neuron is activated, the new output is equal to the sigmoid function of the sum of the products of the weights and the activities of the input wires (connections), minus the threshold of the node. The effect of node I output on the activity of node j is jointly determined by its output level and the strength and sign of its connection to node j. If the sign is negative, it lowers the activation; if the sign is positive it raises the activation. The sigmoid function used in this project is defined by the following equation:

$$\sigma(x) = \frac{1}{1 + \exp(-\beta x)}$$

where β is the steepness of the sigmoid function. The threshold of a neuron determines how large the total input to a node must be before its output becomes significant. As can be seen from the above equation, if the total input to a node is greater than its threshold, then the output of the node is greater than one half The global error function used here to propagate the error back through the network is

$$E = \sum_\text{i}(d_i - o_\text{i})^2$$

where d_i is the desired output and o_i is the actual output produced by the network. The main objective here is to minimize this function. That is to change the weights of the system in proportion to the derivative of the error with respect to the weights according to the following formula (Rumelhart *et al.*, 1994):

$$\frac{\partial E}{\partial o_\text{i}} \frac{\partial o_\text{i}}{\partial w_{ij}}$$

The error correction learning procedure is simple enough in conception, and is as follows: during training input is provided for the network and flows through the network generating a set of values on the output neurons. Then, the actual output is compared with the desired target, and a match is computed. If the output and target match, no change is made to the net. However, if the output differs from the target a change must be made to some of the connections (see Fig. 2). This simple procedure works remarkably well on a wide variety of problems (Masters, 1993). A detailed description of how neural networks work, their advantages and disadvantages, and their application to construction management are discussed elsewhere (Boussabaine, 1996).

Modelling cost-flow forecasting using neural networks

Standard cumulative cost–time curves are used to model cash flow forecasting. Each project curve is divided into nine time intervals (periods) corresponding to 10%, 20%, 30%, . . . 90% completion. A typical curve contains ($n \times m$) periods, where n is the number of inputs provided to the network to predict the next m periods of the project cost flow. Each project cost flow curve is sampled at n periods and desired predictions are made for the next *m* periods. The modelling has been carried out following the logic shown in Fig. 3. A sample of cumulative cost–time curves used in this research is shown in Fig. 4. Table 1 shows the input periods and their corresponding *m* prediction periods. In total six neural networks were trained using *n* periods as input to predict *m* periods as output, see Table 1. Figure 2 shows a typical example of the six nets structure. The first net takes as input the 10% completion periods and forecasts the next eight periods. The second net is trained to take as input the first two periods (i.e. 10% and 20% completion) for forecasting the next seven periods; and likewise for the remaining four nets as shown in Table 2. Thus, each project cost curve is predicted at a different stage of the project progress using six neural programs, which are then grouped under one control program that can be used to activate the program of the required cost flow prediction. The forecast modelling strategy, used here, gives more credence to the future direction of the cost flow movement as an important indicator of a cost flow problem rather than the exact actual forecast values. This will help managers to take corrective measures in order to optimize their resources and to avoid bankruptcy.

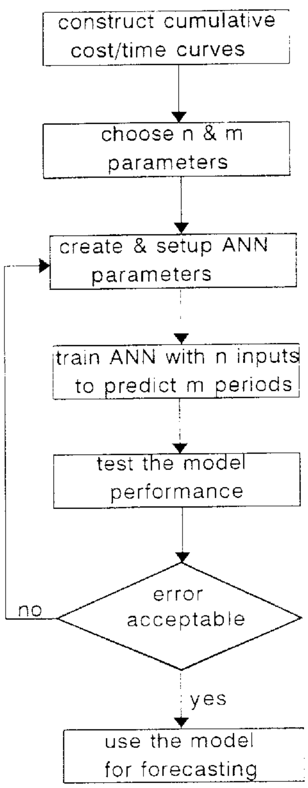

Figure 3 Strategy for developing the model

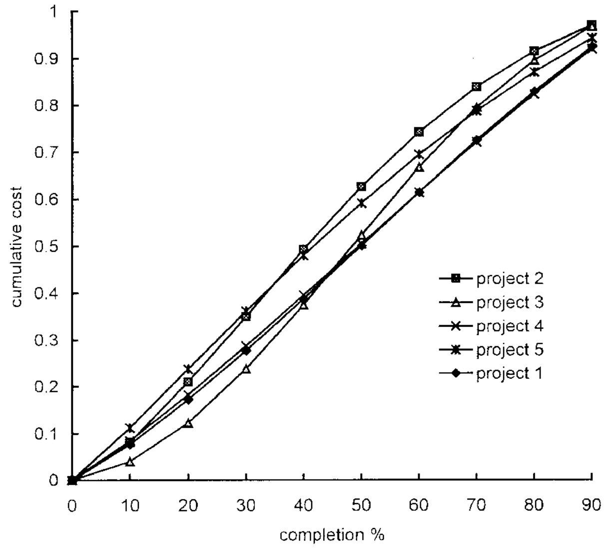

Figure 4 Sample of cost – time curves used in training

Data collection

Table 2 shows a sample of training data. The data used for the training of the six nets were the actual cumulative cost commitment for individual projects at *m* periods (i.e. 9 periods, see Table 1). Only data for projects that were 100% completed were used for training the nets. All the projects used in this study are considered medium size. Their durations ranged from 7 months to 12 months. The monthly cost commitment values of 50 completed building projects were used in the training. A further 15 cases were used in the testing and verification of the system. All the projects were of traditional contract type, in the sense that they were executed under a JCT 80 form of contract. Before data are fed into the system, each individual contract cost was fitted to an *S*-curve. This was done in order to calculate cost entry for the nine period intervals. The logit transformation technique was used, since it has been confirmed to be reliable and flexible (Kenly and Wilson, 1986; Kaka and Price, 1993).

Table 1 Input periods versus output periods

Input periods (*n*)	Forecast periods (*m*)
1	2–9
1–2	3–9
1–3	4–9
1–4	5–9
1–5	6–9
1–6	7–9

Table 2 Sample of training data

		Input periods						Predicted periods (output)							
Completion		10%	20%	30%	40%	50%	60%	20%	30%	40%	50%	60%	70%	80%	90%
Net 1	project 1	0.078						0.185	0.305	0.43	0.553	0.67	0.777	0.871	0.948
	project 2	0.071						0.171	0.286	0.407	0.53	0.65	0.761	0.86	0.943
	project 3	0.056						0.183	0.352	0.529	0.686	0.809	0.898	0.955	0.988
Net 2	project 1	0.078	0.185						0.305	0.43	0.553	0.67	0.777	0.871	0.948
	project 2	0.071	0.171						0.286	0.407	0.53	0.65	0.761	0.86	0.943
	project 3	0.056	0.183						0.352	0.529	0.686	0.809	0.898	0.955	0.988
Net 3	project 1	0.078	0.185	0.305						0.43	0.553	0.67	0.777	0.871	0.948
	project 2	0.071	0.171	0.286						0.407	0.53	0.65	0.761	0.86	0.943
	project 3	0.056	0.183	0.352						0.529	0.686	0.809	0.898	0.955	0.988
Net 4	project 1	0.078	0.185	0.305	0.43						0.553	0.67	0.777	0.871	0.948
	project 2	0.071	0.171	0.286	0.407						0.53	0.65	0.761	0.86	0.943
	project 3	0.056	0.183	0.352	0.529						0.686	0.809	0.898	0.955	0.988
Net 5	project 1	0.078	0.185	0.305	0.43	0.553						0.67	0.777	0.871	0.948
	project 2	0.071	0.171	0.286	0.407	0.53						0.65	0.761	0.86	0.943
	project 3	0.056	0.183	0.352	0.529	0.686						0.809	0.898	0.955	0.988
Net 6	project 1	0.078	0.185	0.305	0.43	0.553	0.67						0.777	0.871	0.948
	project 2	0.071	0.171	0.286	0.407	0.53	0.65						0.761	0.86	0.943
	project 3	0.056	0.183	0.352	0.529	0.686	0.809						0.898	0.955	0.988

Training

The back-propagation learning method was used for training the six nets. Each network was trained to simulate the standard cumulative curve of 45 projects. Only one hidden layer that contained 40 neurons was used in each network. This was selected on an ad hoc basis. The supervised learning mechanism was used to train the nets. In this approach, each net is presented with n inputs and the desired output is set so that the nets can learn the relationship between the inputs and outputs. The six neural networks were trained until the RMS error was reduced to 0.001. Fifteen past cumulative cost–time curves were used to test the six networks. Each net was presented with n inputs and asked to forecast the next m periods of the standard cost flow curve. A sample of the forecast values is given in Table 3. The last row at the foot of the table shows the mean absolute deviation (MAD) of the forecasting error for each cumulative curve. The low MAD values shown in Table 3 could be attributed to the fact that the system has learned the relationship between inputs and outputs and the system also can generalize from data.

Measuring prediction error

The validation tests performed on the networks were a comparison between their accuracy of prediction with the actual costs of projects at desired completion periods. This involves statistical verification that the six nets can generalize. Generalization, which is an important feature of neural networks, means that the network has the ability to predict the cost flow of the desired periods using input data that are not present in the training set. In order to perform this test, two statistical error measurement methods are used. The root-mean-square error (RMS) is the most widely used method for error measurement, and is computed using the formula (Fausett, 1994)

$$\mathrm{RMS} = \sqrt{\left\{ \frac{1}{n} \sum_{i=0}^{n-1} (t_i - o_i)^2 \right\}}$$

where t_i and o_i are the actual and predicted cumulative costs at period i, and n is the number of periods. The RMS error is an absolute number in that it is not related directly to the predicted values, and is easy to explain in that it is linear. For example, doubling all individual errors will double the RMS error. However, the problem in using this method to measure the performance of the system is that the scale of the 5 RMS is tied to the measurement unit of data (Masters, 1993). Therefore, care must be taken in the interpretation and comparison of the results with other cash flow forecasting models. To avoid misinterpretation of the results and facilitate comparison with other forecasting methods the RMS error is expressed as a unitless relative quantity. The following relative RMS error formula is used along with the RMS to measure the prediction error of the six networks (Masters, 1993).

Table 3 Samples from network testing

(a)	Network 1 (10% completion)														
0%	Project 1			Project 2			Project 3			Project 4			Project 5		
Completed	Actual	Predicted	Error	Actual	Predicted	Error	Actual	Predicted	Error	Actual	Predicted	Error	Actual	Predicted	Error
0	0			0			0			0			0		
10	0.08			0.08			0.04			0.08			0.11		
20	0.17	0.19	−0.02	0.21	0.20	0.01	0.12	0.12	0.00	0.18	0.20	−0.02	0.24	0.24	0.00
30	0.28	0.30	−0.02	0.35	0.31	0.04	0.24	0.19	0.05	0.29	0.32	−0.03	0.36	0.36	0.00
40	0.39	0.38	0.01	0.49	0.40	0.09	0.37	0.29	0.08	0.40	0.41	−0.02	0.48	0.46	0.02
50	0.50	0.50	0.00	0.63	0.52	0.11	0.52	0.38	0.14	0.50	0.53	−0.03	0.59	0.58	0.01
60	0.61	0.65	−0.04	0.74	0.66	0.08	0.67	0.51	0.16	0.61	0.66	−0.05	0.69	0.69	0.00
70	0.72	0.78	−0.06	0.84	0.79	0.05	0.79	0.73	0.06	0.72	0.79	−0.07	0.79	0.79	0.00
80	0.83	0.88	−0.05	0.91	0.88	0.03	0.90	0.88	0.02	0.82	0.89	−0.07	0.87	0.88	−0.01
90	0.92	0.95	−0.03	0.97	0.95	0.02	0.97	0.96	0.01	0.92	0.95	−0.03	0.94	0.94	0.00
1	1	1		1	1		1	1		1	1		1	1	
Average error	0.55		−0.03	0.64		0.05	0.57		0.06	0.56		−0.04	0.62		0.00
SD error			0.02			0.03			0.06			0.02			0.01
(b) Network 2 (20% completion)															
20	0.17			0.21			0.12			0.18			0.24		
30	0.28	0.27	0.01	0.35	0.35	0.00	0.24	0.23	0.01	0.29	0.28	0.01	0.36	0.36	0.00
40	0.39	0.38	0.01	0.49	0.48	0.01	0.37	0.37	0.00	0.40	0.39	0.01	0.48	0.48	0.00
50	0.50	0.49	0.01	0.63	0.62	0.01	0.52	0.51	0.01	0.50	0.49	0.01	0.59	0.59	0.00
60	0.61	0.61	0.00	0.74	0.73	0.01	0.67	0.66	0.01	0.61	0.60	0.01	0.69	0.70	−0.01
70	0.72	0.71	0.01	0.84	0.83	0.01	0.79	0.78	0.01	0.72	0.71	0.01	0.79	0.79	0.00
80	0.83	0.82	0.01	0.91	0.91	0.00	0.90	0.89	0.01	0.82	0.81	0.01	0.87	0.87	0.00
90	0.92	0.92	−0.01	0.97	0.96	0.01	0.97	0.96	0.01	0.92	0.92	0.00	0.94	0.94	0.00
1	1	1		1	1		1	1		1	1		1	1	
Average error	0.61		0.01	0.70		0.01	0.64		0.01	0.61		0.01	0.67		0.00
SD error			0.01			0.00			0.00			0.01			0.00
(c) Network 3 (30% completion)															
30	0.28			0.35			0.24			0.29			0.36		
40	0.39	0.39	0.00	0.49	0.50	−0.01	0.37	0.38	−0.01	0.40	0.40	−0.01	0.48	0.48	0.00
50	0.50	0.5	0.00	0.63	0.63	−0.01	0.52	0.53	−0.01	0.50	0.51	−0.01	0.59	0.60	−0.01
60	0.61	0.61	0.00	0.74	0.74	0.00	0.67	0.67	0.00	0.61	0.61	0.00	0.69	0.70	−0.01
70	0.72	0.72	0.00	0.84	0.84	0.00	0.79	0.80	−0.01	0.72	0.72	0.00	0.79	0.79	0.00
80	0.83	0.82	0.01	0.91	0.92	−0.01	0.90	0.90	−0.01	0.82	0.82	0.00	0.87	0.87	0.00
90	0.92	0.92	0.00	0.97	0.97	0.00	0.97	0.97	0.00	0.92	0.92	0.00	0.94	0.94	0.00
Average error	0.66		0.00	0.76		0.00	0.70		−0.01	0.66		0.00	0.73		0.00
SD error			0.00			0.00			0.00			0.00			0.00
(d) Network 4 (40% completion)															
40	0.39			0.49			0.37			0.40			0.48		
50	0.50	0.50	0.00	0.63	0.63	−0.01	0.52	0.52	0.00	0.50	0.51	−0.01	0.59	0.59	0.00
60	0.61	0.61	0.00	0.74	0.74	0.00	0.67	0.67	0.00	0.61	0.62	−0.01	0.69	0.70	−0.01
70	0.72	0.72	0.00	0.84	0.84	0.00	0.79	0.79	0.00	0.72	0.72	0.00	0.79	0.79	0.00
80	0.83	0.83	0.00	0.91	0.91	0.00	0.90	0.90	−0.01	0.82	0.82	0.00	0.87	0.87	0.00
90	0.92	0.92	0.00	0.97	0.97	0.00	0.97	0.97	0.00	0.92	0.92	0.00	0.94	0.94	0.00
Average error	0.72		0.00	0.82		0.00	0.77		0.00	0.72		0.00	0.78		0.00
SD error			0.00			0.00			0.00			0.00			0.00
(e) Network 5 (50% completion)															
50	0.50			0.63			0.52			0.50			0.59		
60	0.61	0.62	−0.01	0.74	0.75	−0.01	0.67	0.67	0.00	0.61	0.62	−0.01	0.69	0.70	−0.01
70	0.72	0.72	0.00	0.84	0.84	0.00	0.79	0.80	−0.01	0.72	0.72	0.00	0.79	0.79	0.00
80	0.83	0.83	0.00	0.91	0.91	0.00	0.90	0.90	−0.01	0.82	0.82	0.00	0.87	0.87	0.00
90	0.92	0.93	−0.01	0.97	0.97	0.00	0.97	0.97	0.00	0.92	0.92	0.00	0.94	0.94	0.00
Average error	0.77		0.00	0.87		0.00	0.83		0.00	0.77		0.00	0.82		0.00
SD error			0.00			0.00			0.00			0.00			0.00

$$\text{REL(RMS)} = \sqrt{\left[\frac{\sum_{i=0}^{n-1}(t_i - o_i)^2}{\sum_{i=0}^{n-1}(t_i - \bar{t})^2}\right]}$$

The mean of the actual values, $\bar{t}$, is subtracted from each period so that the true variance can be used to measure the degree of error with the overall degree of variance.

As can be seen from Figs 5 and 6, RMS and REL (RMS) measurements provide different types of information about the predictive capabilities of the model. Figure 5 is a plot of the RMS errors for a sample of projects, and shows that the RMS is a good measure for indicating large errors. Figure 6 is a plot at 10% completion of the REL(RMS) for a sample of projects, and provides a more balanced perspective of the goodness of fit for all periods. Figures 5 and 6 show that RMS and REL(RMS) errors of the nets at 10% completion are significantly larger than those at 20%, 30% and 40% completion. This could be attributed largely to the variability of the data at this stage of the project's progress and/or to the topology chosen for this network which may not be optimum and need more refinement. Note that the REL(RMS) error increases slightly at the 50% completion period. This might be due to the fact that this network's parameters were not fully optimized, and further trials are required to find the optimum network settings. Table 3 presents a summary of the prediction results. The first two columns for each project represent the actual

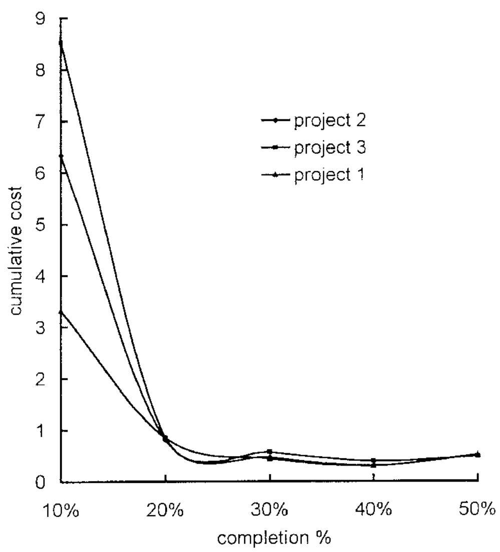

Figure 5 RMS errors for a sample of test projects

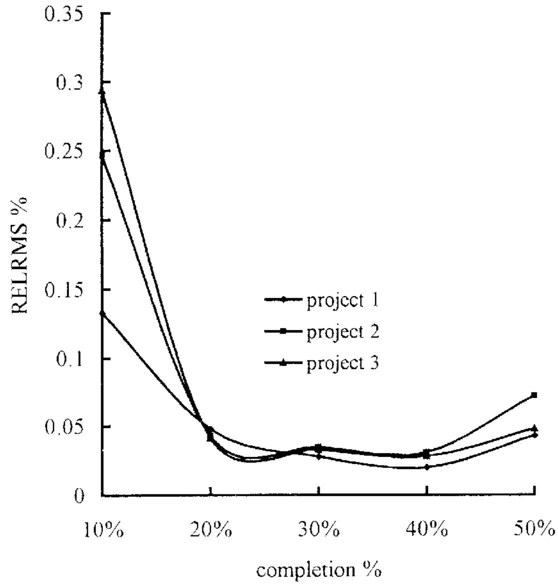

Figure 6 REL (RMS) for a sample of test products

and predicted cumulative cost values for m periods using n inputs. The third column represents the absolute error. The difference in error in all the six nets is not very significant. The forecast errors obtained for the six nets show that the networks achieved an adequate level of accuracy for different periods. The forecasting errors are only slightly different from the actual values, as shown in Table 3 and Fig. 7. Therefore, this statistical test illustrated the network capability to classify inputs into appropriate clusters under a wide range of information (i.e., variability of input and output data). Thus, it may be concluded that the six nets have learned to generalize from the data presented to them. These results show that a neural network approach to cost flow forecasting problems is a promising alternative to more traditional methods.

Cost flow based on forecasts obtained

Cost forecasting is an essential tool for cash flow management, and many construction companies have bankrupted due to inadequate cash flow management. Simple rules of thumb relating to standard cost flow have been used extensively in cost flow management in construction projects. Rules of thumb essentially use standard curves for passive cost flow management strategy of a reactive type. Managers are more interested in the direction of movement of cost flow rather than its forecast value. Managers would be interested not only in novel efforts in forecasting but also in

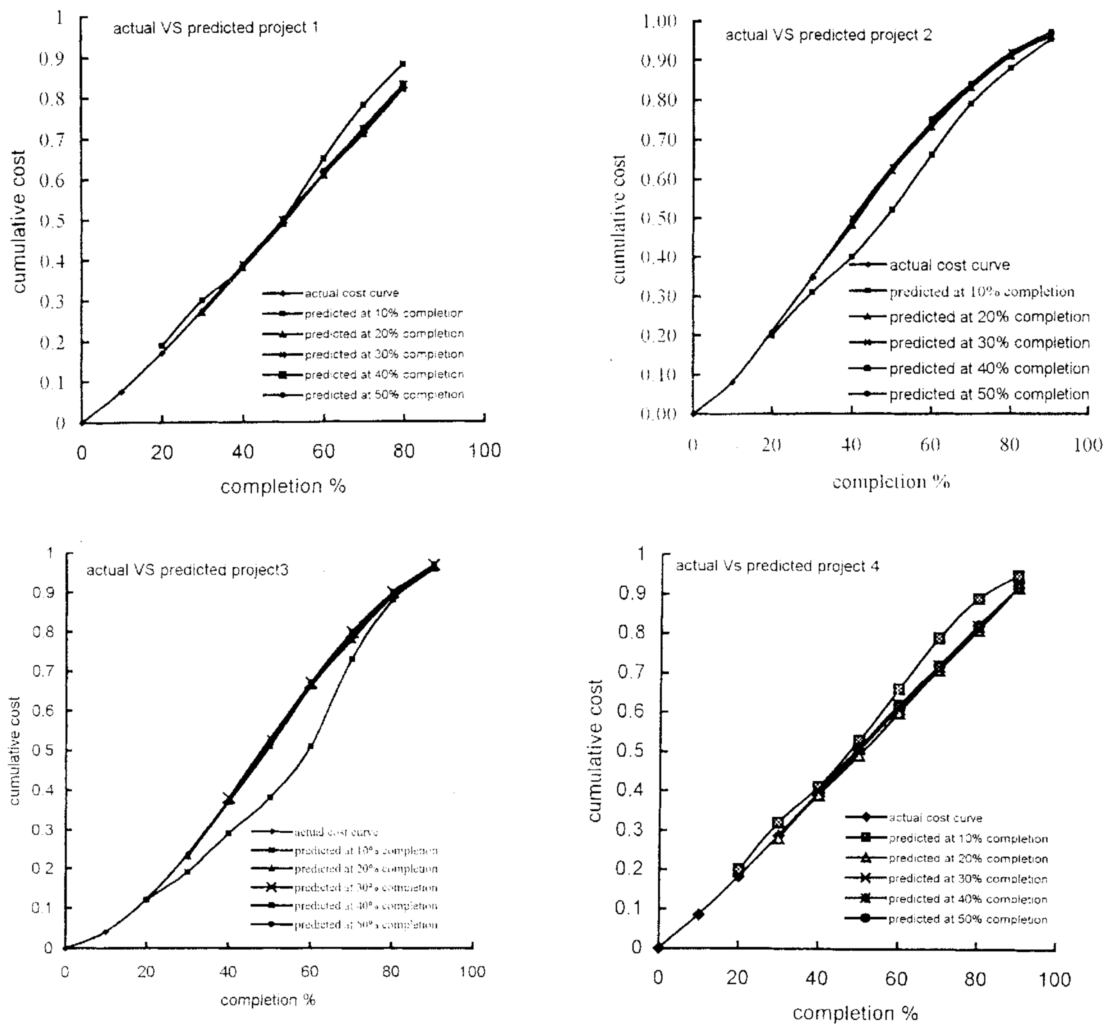

Figure 7 Test results

proactive forecasting strategies. The ANN system can be used to help in this process. The output from the system can be used to analyse the cost flow curve of projects for m periods to make sure it is reasonable. For example, the actual curve, which must be derived from the schedule activities, should be compared with the predicted curve from the neural network system. As a rule of thumb, a typical S-curve for mid-rise buildings would show 1/4 of the cost committed at the end of the first third of the project, and 3/4 of the cost at the end of the second third. Analysis of the cost distribution is required throughout the project $(n \times m)$ periods to determine if there is evidence of cost flow front-end loading. Are there any periods that are likely to overrun? If so, does the cumulative cost for these periods appear to be reasonable in relation to other similar projects? Is the cumulative percentage of cost committed at periods 1/3 and 2/3 of the project duration greater than at 1/4 and 3/4, respectively? All of these concepts plus other rules of thumb can be combined with the neural network system to help managers to perform variance analyses on the actual versus the predicted cost flow, to determine the cash flow periods that fall outside acceptable standard cash flow rules of thumb.

Conclusion

This paper discusses the need for cost flow forecasting and demonstrates the shortcomings of the existing forecasting methods. The conclusion reached is that a new cost flow forecasting method based on nonlinear techniques is required to deal with the complex problem of cost flow at project level. This paper proposes an artificial neural network to solve this problem. A model is developed to forecast cash flow for 9 periods of the project duration. The model takes n inputs and produces a forecast of the next *m* periods. The cost curves of 50 projects of medium size ranging in duration from 7 months to 12 months were used in the training. A further 15 cases were used in the testing and verification of the system. The comparison between the actual and forecast cost curves showed very little difference. The testing results are very encouraging, but further testing is required before concluding that a neural networks approach is more accurate than traditional methods. The authors are currently working on using statistical methods for updating cost flow curves. Results on the accuracy of the statistical model will then be confirmed with those given in this paper.

References

Balkau, B.J. (1975) A financial model for public works programmes, National ASOR Conference, Sydney, August 25–27.

Barron, A.R. and Barron R. (1989) Statistical learning networks: a unifying view, *Computing Science and Statistics*, Wegman E.J., Gantz D.I., and Miller J.J. (eds), pp. 192–202.

Boussabaine, A.H. (1996) The use of artificial neural networks in construction management: a review, *Construction Management and Economics*, **14**(5), 427–36.

Bromilow F.J. and Henderson, J.A. (1977) *Procedures for Reckoning the Performance of Building Contracts*, 2nd Edn. CSIRO, Division of Building Research, Highett, Australia.

Drake, B.E. (1978) A mathematical model for expenditure forecasting post contract, in *Proceedings of the Second International Symposium on Organisation and Management of Construction*, Vol. 2, Technion Israel Institute of Technology, Haifa, Israel, pp. 163–83.

Fausett, L. (1994) *Fundamentals of Neural Networks: Architecture, Algorithms and Applications*, Prentice Hall, Englewood Cliffs, NJ.

Gallant, S.I. (1993) *Neural Network Learning and Expert Systems*, MIT Press, Cambridge, MA.

Hammertrom, D (1993) Neural networks at work, *IEEE Spectrum*, June, pp. 26–32.

Hardy, J.V. (1970) Cash flow forecasting for the construction industry, M.Sc. Report, Department of Civil Engineering, Loughborough University of Technology, UK.

Hudson, K.W. (1978) DHSS expenditure forecasting method, *Chartered Surveyor – Building and Quantity Surveying Quarterly*, **5**,42–5.

Hush, P.R. and Horne, B.G. (1993) Progress in supervised neural networks, *IEEE Signal Processing Journal*, January, 8–39.

Kaka, A.P. and Price, A.D.F. (1993) Modelling standard cost commitment curves for contractors' cash flow forecasting, *Construction Management and Economics*, **11**(4), 271–83.

Karmarthi, 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.

Kenley, R. and Wilson, O. (1986) A construction project cash flow model – an idiographic approach, *Construction Management and Economics*, **4**(3), 213–32.

Khosrowshahi, F.(1991). Simulation of expenditure patterns of construction projects, *Construction Management and Economics*, **9**(2), 113–32.

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, *IEEE Proceedings*, **78**(10), 1568-74.

Lippman, R.P. (1987) An introduction to computing with Neural Nets, *IEEE, Ass. Press*, Journal, **4**(2), 4–22.

Masters, T. (1993). Practical neural networks recipes in C^{++}, Academic Press, London.

Miskawi, Z. (1989) An *S*-curve equation for project control, *Construction Management and Economics*, **7**(2), 115–25.

Navon, R, (1995) Resource based model for automatic cash-flow forecasting, *Construction Management and Economics*, **13**(6), 501–10.

Nerrand, O., Roussel-Rugot, P.. Personnaz. L., Dreyfus. G. and Marcos. S. (1993) Neural networks and nonlinear adaptive filtering: unifying concepts and new algorithms. *Neural Computations*, **5**(2), 165–99.

Oliver, J.C. (1984) Modelling cash flow projections using a standard micro computer spreadsheet program, M.Sc. Construction Management Project, Loughborough University of Technology.

Peer, S. (1982) Application of cost flow forecasting models, *Journal of the Construction Division ASCE*, **108**, (CO2), 226–32.

Rumelhart, D., Widrow, B. and Lehr, A. (1994) The basic ideas in neural networks, *Communications of the ACM*, **37**(3), 87–91.

Shaw, J. (1992) Neural network resource guide, *AI Expert*, **8**(2), 48–54.

Singh, S. and Woon, P.W. (1984) Cash flow trends for high-rise building projects, in *Organising and Managing Construction*, Proceedings of the 4th International Symposium on Organisation and Management of Construction, University of Waterloo, Canada.

Widrow, B., Rumelhart, D. and Lehr, A. (1994) Neural networks: application in industry, business and science, *Communications of the ACM*, **37**(3), 93–105.