A Backpropagation Neural Network for Sales Forecasting

J H L Kong & G P M D Martin
Department of Computer Technology
Monash University
900 Dandenong Road, Caulfield East Victoria, Australia 3145
gmartin@ct.monash.edu.au

Abstract

This study uses a backpropagation neural network (BPN) to forecast future sales volumes of a food product for a large Victorian food wholesaler. The study compares the results obtained using different parameter values, and discusses network performance. The BNP results are compared with the methods currently used by the company which involve trend and market analysis using a simple linear regression model. The BPN appears to give better forecasts than the statistical methods. Specific factors such as advertising, and competition from other competitors were not included. It is believed that some of these factors may be important. The results obtained suggest that the BPN model may provide a useful tool for generating sales forecasts, however poor selection of parameter settings can lead to slow convergence and/or incorrect output

Introduction

Backpropagation neural network (BPN) is one of the most popular neural network models for use in business applications, especially with time series prediction problems such as sales forecasting. Sales forecasting is the prediction of future sales based on past historical data.

There are several methods which have been used for forecasting. The formulation of a deterministic mathematical model using past data is a possible approach. However, this method can not include uncertainties which affect sales. It is often not possible to fully understand or know what to include in the model of a frequently complex system. In sales forecasting applications the relationships between the various parameters are non-linear and unclear and some of the useful parameters are not easily quantifiable. It is often impossible to get enough relevant parameters to specify the system adequately [6]. There is a lack of mathematical algorithms for the analysis of data which incorporates imprecise and uncertain knowledge data such as the effect of festive seasons on sales.

There are several statistical techniques which have been applied to this problem. Most are based on linear regression models. As noted above, the relationships between the sales volume and measurable parameters are complex and non-linear and this causes the statistical techniques to be of limited effectiveness.

Neural network systems are a possibly more fruitful approach. The neural network model can cope with the non-linear relationships between the various parameters. Apart from past statistical and numerical data, neural networks allow the inclusion of important influential factors. For example a product's promotion, and its estimated market share can be included and translated into neural network system readable data.

Neural networks models such as BPN may also outperform statistical analysis models and other conventional methods including knowledge-based systems (Dutta and Shekhar, 1988; [8].

This study used a BPN model to forecast future sales volumes for a food product for a large Victorian food wholesaler. The study also compared the results (sales volume) obtained using different parameter values, and measured network performance. The best set of parameters for the particular sales product and then the BNP results were compared with the methods currently used by the company which involve trend and market analysis using a simple linear regression model.

BPN is said to be simple to implement, solves most problems correctly, and is "unfussy" about how it is used. BPN is characterised by its robustness, its ability to generalise, learn, and to be trained. Neural networks allows dynamic modification which incorporates learning. What makes a system 'intelligent' is that it is able to learn and modify by experiences. The connection weights are modified or trained until the network behaves in the desired manner when a data set consisting of inputs and outputs is presented to it. BPN can help "data miners" understand new events and make predictions about future events by generalising from prior events. It has the ability to connect experiences that are not so obvious. It can also interrelate both linear and non linear data sets. Apart from being creative, BPN is also able to generalise well from a set of

noisy and irrelevant data. For example, it is able to produce a reasonable solution when it has never experienced or encountered the given data, or even when the given data is not complete.
Previous work which was used in the development of this study included that of Lapedes [5] which predicted the closing price for the following week sales using the closing price from the Standard & Poor (S&P) 500 for the ten previous weeks; the Columbia River Flow Prediction study of Harris and Davis [2]; White's stock market prediction study [11], and a study of sales prediction for a gaming operator by Jagielska and Jacob [3].

BPN Architecture

The process by which BPN works is well explained by Rumelhart et. al. [7].

For the j th neurode of a layer, each of the input (Xi) values are multiplied by a weight (Wij), and then summed:

$$Sj = \Sigma_N \; Wij \bullet Xi$$

where Wij is the value of connection between neurodes i and j, X_i is the i th input value, and N is the number of input neurodes.

The output of each neurode is determined by applying the following formula:

$$Oj = \sigma (Sj)$$

where one of the most common transfer functions used is the sigmoid function,

$\sigma (x) = 1 / (1 + \exp(-x))$. This transfer function will produce output values ranging between 0 and 1.

The training of BPN involves two processes: a feedforward and a backpropagation. During training, the interconnection weights are adjusted. When a training set vector is presented to the network, the vector will propagate to the output layer. The resultant output vector is then compared with the desired output vector which is supplied by the "teacher" in supervised training. The backpropagation process involves the propagation of the difference error back through lower layers so that the weight matrix of each lower layer is adjusted [1]

For training the BPN for forecasting we use earlier portion of historical sales data as inputs and the later ('future') values as the desired output. The effectiveness of training is measured by applying additional test data to the trained network operating in analysis mode (rather than training mode) and statistically comparing the output obtained against the known historical behaviour. We used the sum of squared error (SSE) between the two sets of results as a goodness of training measure.

BPN learns by making changes to its weights in a direction which minimises the sum of squared errors between its predictions and a training data set using a steepest descent algorithm. A typical BPN has three layers of neurodes (input, hidden and output). There may be more than one hidden layer.

The experiments in this study were conducted using NeuralWorks Explorer (NeuralWare, Inc.) on an IBM compatible 486 PC using a standard spreadsheet (Microsoft Excel) and a word processor (Microsoft Word). Training was repeated with different parameter values and sliding window sizes until the minimum error margin was found between the desired output and predicted output. The experiment also made use of other available tools in NeuralWorks Explorer to determine the optimal performance of the system. The experiments followed the guidelines stated in the Reference Guide and other NeuralWorks Explorer manuals listed in the reference.

There were four input variables included in the study (past sales volumes, school holidays, festive seasons and climate seasons). Monthly sales data over a period of 3 years was used.

A number of network topologies with different parameters and learning procedures were experimented with. Input variables values were presented sequentially to the network together with the corresponding output values identifying the expected sales volume. After the entire training data set had been input to the network, that is after a training epoch had been completed, a new epoch was started using the same data set and presentation sequence.

The transfer function used was sigmoid because of its non-linear nature which plays an important role in the performance of the neural network. Other transfer functions such as the hyperbolic tangent were also applied but they showed no significant impact on the results. The training rule applied was the delta rule. The hetero-associative network was chosen instead of an autoassociative one as both the training input and the corresponding output were available.

The first experiment was aimed at evaluating the performance of BPN by varying the size of epoch value. Other parameters were also varied to find the optimal forecast.

Training evaluation is generally a process of measuring how well a network is performing on the training data set. A 'good' training set is crucial as it will affect the network capacity to generalise. Supervised learning of BPN also helps in minimizing the error between the desired output and the actual output. Wasserman states that the objective of training is "to adjust the weights so that application of a set of inputs produces the desired set of outputs"[10]. Training is stopped when a specified tolerance is achieved or the learn count is reached. and the final weights are stored for future use.

Training using different parameters with same input data is required to find the best results (that is, in terms of learning rate, number of hidden neurodes and other related parameters).

The NeuralWorks Explorer provide instruments such as Root Mean Square (RMS) Error chart and Weight Histograms to help in optimizing the network parameters. The system displays outputs and measures network performance while the network is running. The system probes allow the editing and display of weights, hidden activations and other parameters of the processing elements or neurodes.

Predictions can be done almost instantaneously once the network is trained.

Results and problems

Monitoring the network was done by displaying the state of weights and activations. During training, the RMS error of individual sales patterns which was initially small, decreased slightly. The errors become even across the neurodes during training. The weights histogram also showed that the weights gradually spread out as learning progress. The confusion matrix was not used as it is not beneficial in determining the performance of the network.

The experiments showed that the network achieved a better learning capacity using the following parameter values :

NUMBER OF PROCESSING ELEMENTS FOR INPUT LAYER : 12

NUMBER OF PROCESSING ELEMENTS FOR HIDDEN LAYER : 6

NUMBER OF PROCESSING ELEMENTS FOR OUTPUT LAYER : 1

LEARNING COEFFICIENT : 0.3

MOMENTUM : 0.7

EPOCH : 36

TOLERANCE : 0.001

The current error field of each processing element in the output buffer of the network was checked as it provides the error between desired and actual output for that particular processing element only. Most of the values of the current error fields were less than 0.15.

The two layers of BPN (input and hidden) are crucial because of its nonlinear relationship between inputs and outputs. Increasing the number of processing elements and hidden layers did not seem to improve the results but instead it increases the training time and computer memory required. This would not be a problem if only one or two products are being forecasted, but if thousands of products items are to be forecasted, storage could be problem. Increase of sliding window size tends to increase the storage and complexity as well.

The SSE and mean SSE was minimised through the training process and various experiments.

The results indicated that the number of processing elements and hidden layers together with the values of learning coefficient and momentum contribute significantly to the training results and the overall performance of BPN.

Figure 1 shows the plots of actual sales volume, BPN predicted sales volume and the company's conventional method prediction sales volume.

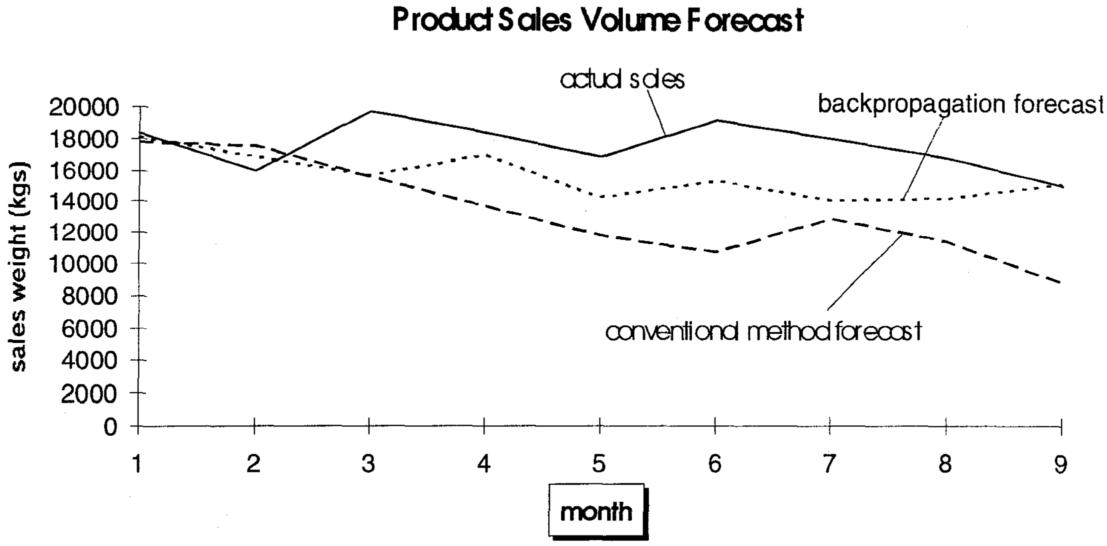

Figure 1 : *Comparison of the desired forecast (actual sales) with the conventional and BPN methods forecast.*

The study indicates that BPN gave a better result than the conventional method used by the company. The error is different from month to month when comparison is made between the desired and predicted sales volume of each method. The BPN model has an SSE significantly lower than that of the conventional method. The best parameters were decided on the basis of the overall system performance.

Choosing the right value of the learning rate or momentum was difficult. If the value is too small, a very slow convergence will occur. On the other hand, if the value is too large, the optimization process will diverge [9]. Pruning by removing all neurodes whose weight is negligible can enhance the BPN performance. The level of experience required to manage the neural network appears to depend on the complexity of the problem to be solved.

The effectiveness of the network for a given application is hard to measure. It is not easy to examined a trained network and made inferences from it. Network weights histogram and RMS error provided little insight into the performance of the trained network. Having to do many experiments is troublesome and time consuming.

The NeuralWorks Explorer does not do the postprocessing tasks which are often necessary to measure the performance of the sales forecasting system. A separate spreadsheet like Microsoft Excel was needed to analyse the results and plot graphs.

Conclusions

The BPN gave better forecasts than the statistical methods for this scenario but the training data set was not large enough for the network to generalise at its best. Specific factors such as advertising, and competition from other competitors were not included. It is believed that some of these factors may have a significant influence on the results. A small learning coefficient was used to ensure that the network settled to a stable solution but a large number of iterations were required. A compromise value of learning coefficient and momentum is often required to help speed up the convergence and hence the training time.

Determining a good set of input variables and network parameters are crucial in generating good prediction as well as reducing training time to a reasonable level. The results obtained in this study suggested that BPN model may provide a useful tool for generating sales forecast. however, poor parameter settings will lead to slow convergence and/or incorrect output.

True indicators of the neural network's performance are not predictable. For example, leaving out some factors of influence in the input data may or may not lead to lower accuracy of results. One needs to experiment with the different variables and factors of influence.

Addition of more neurodes to the network layer may not necessarily increase the performance of the system. However, in general, the robustness of the system in processing missing or incomplete data is increased when more neurodes are added.

Reference

[1] Ahalt, S.C., Jung, T-P and Krishnamurthy, A.K. (1991). 'A comparison of MLP and FSCL-LVQ neural networks for vowel classification', in Neural Networks: Concepts, Applications, and Implementations. (eds.) Antognetti, P. and Milutinovic, V. Vol. III, Prentice-Hall, Englewood Cliffs, New Jersey. 124- 143.

[2] Harris, B. and Davis, A. (1991). 'Columbia River Flow Prediction using a Neural Network Model and Time Series Data' in HNC Inc., San Diego.

[3] Jagielska, I. and Jacob, A. (1993). A Neural Network Model for Sales Forecasting. Working Paper Series, Department of Information Systems, Faculty of Computing and Information Technology, Monash University, Melbourne, Australia.

[4] Kempka, A. (1994). 'Activating Neural Networks : Part 1'. AI Expert, 33-37. (June).

[5] Lapedes, A. and Farber, R. (1987). 'Non-linear Signal Processing using Neural Networks : Prediction and System Modelling' in Los Alamos National Laboratory report.

[6] Rogers, R.D. and Vemuri, V. (1994). 'Time Series and Forecasting Problem', in Artificial Neural Networks : Forecasting Time Series. (eds.) V. Vemuri and R.D. Rogers. IEEE Computer Society Press, Los Alamitos, CA.

[7] Rumelhart, D. and McClelland, J.L. and the PDP Research Group (1986). Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Vol. 1, Chap. 8, Cambridge, Mass.: MIT Press.

[8] Sharda, R. and Patil, R.B. (1990). 'Neural networks as forecasting experts: an empirical test, 'International Joint Conference on Neural Networks, II, 491- 494.

[9] Silva, F.M. and Almeida, L.B. (1990). 'Speeding up Backpropagation,' in Advanced Neural Computers. (ed.) R. Eckmiller. Elsevier Science Publishers B.V. (North-Holland). 151-158.

[10] Wasserman, P.D. (1989). Neural Computing : Theory and Practice. USA: Van Nostrand Reinhold.

[11] White, H. (1988). 'Economic Prediction Using Neural Networks : The Case of the IBM Daily Stock Returns' in IEEE International Conference on Neural Networks, II451-8.

[12] ----- (1993). Neural Computing : A Technology Handbook for Professional II/PLUS and Network Explorer. NeuralWare, Inc. USA.

[13] ----- (1991). Reference Guide : NeuralWorks Professional II/Plus and NeuralWorks Explorer. NeuralWare, Inc.

are two types of security analyses: fundamental analysis and technical analysis. To maximize profits from the stock market, more and more 'best' forecasting techniques are used by different traders, or so to speak. Nowadays, traders no longer rely on a single technique to provide information about the future of markets. They use a variety of techniques to obtain multiple signals. Neural networks are often trained by both technical and fundamental indicators to produce trading signals. Fundamental and technical analysis could be simulated in neural networks. For fundamental methods, retail sales, gold price, industrial production index, and foreign currency exchange rate etc. could be used as inputs[6]. For technical methods, the delayed time series data could be used as inputs. In this paper, a mixed technical method which takes not only the delayed time series data as inputs but also the technical indicators. Neural networks are trained to approximate the thinking and behavior of some stock market traders. Different indicators are used as the inputs to a neural network and the index of stock is used to supervise the training process. Finally, the trained neural network is used to predict the future levels of the KLSE index .

3. Neural Network And Its Usage In Stock Market

A typical backpropagation neural network[14] is used to capture the relationship between the stock prices of today and the future. The relationship can be obtained through a group of mappings of constant time interval. Assume that u_i represents today's price, v_i represents the price after ten days. If the prediction of a stock price after ten days could be obtained using today's stock price, then there should be a functional mapping from u_i to v_i, where

$$v_i = \Gamma_i(u_i) \tag{1}$$

Using all (u_i, v_i) pairs of historical data, a general function $\Gamma()$ which consists of $\Gamma_i()$ could be obtained.

$$v = \Gamma(u) \tag{2}$$

More generally, $\vec{u}$ which consists of more information in today's price could be used in function $\Gamma()$. Neural networks can simulate all kinds of functions, so they also can be used to simulate this $\Gamma()$ function. The $\vec{u}$ is used as the inputs of the neural network.

There are five major steps in the neural network based forecasting. First, the information that could be used as the inputs and outputs of neural network are collected. Second, these data are normalized and scaled in order to reduce the fluctuation and noise. Third, a neural network model that could be used to capture the relationship between the data of inputs and outputs is built. Fourth, variations of the models, i.e., different models and configurations with different training, validation and testing data sets are experimented with. Finally, the best model measured by out-of-sample hit rates, for example, is chosen for use in forecasting.

4. A Case Study on the Forecasting of the KLSE Index

The technical analysis method is used commonly to forecast the KLSE index, the buying and selling point, turning point, and the highest, lowest point etc. When forecasting by hand, different charts will be used by analysts in order to get the ideas of things happening in the future. Neural networks could be used to recognize the patterns of the chart and the value of index.

4.1. Data Choice And Pre-processing

The daily data from Jan 3, 1984 to Oct 16, 1991(1911 data) are used on the first trial. The mean of these data is 383.73, the standard deviation is 122.66. After normalization they are 0.080466 and 0.527310 respectively. These data are preprocessed to be used in the prediction of the daily, weekly, half monthly, monthly, bimonthly indices. Technical analysts usually use indicators to predict the future. The major types of indicators are *moving average(MA), momentum(M), Relative Strength Indicator(RSI), stochastics(%K), and moving average of stochastics(%D).* These indicators can be derived from the real stock composite index. The target for training the neural network is the actual index. The forecast of the weekly and monthly stock indices is experimented, with the closing price of the calendar week and month being used as weekly and monthly data. In general, the stock price data have bias due to differences in name and spans. Normalization can be used to reduce the range of the data set to values appropriate for inputs to the activation function being used.

The inputs of the neural network model are $I_{t-1}, I_t, MA_5, MA_{10}, MA_{50}, RSI, M, \%K$ and $\%D$. The output is I_{t+1}. Here I_t is the index of *t-th* period, MA_j is the moving average after *j-th* period, and I_{t-1} is the delayed time series. Other indicators are defined as follows,

$$M = CCP - OCP \tag{3}$$

where
CCP = Current closing price
OCP = Old closing price for a predetermined period (5 days)

$$RSI = 100 - \frac{100}{1 + \frac{\Sigma(Positive\_changes)}{\Sigma(Negative\_changes)}} \tag{4}$$