Neural Processing Letters, 25(1):63-78, 2007

Product Demand Forecasting with a Novel Fuzzy CMAC

D.Shi*, C. Quek, R. Tilani and J. Fu

School of Computer Engineering, Nanyang Technological University, Singapore 639798

[*Corresponding author, Email: asdmshi@ntu.edu.sg]

ABSTRACT

Forecasting product demand has always been a crucial challenge for managers as they play an important role in making many business critical decisions such as production and inventory planning. These decisions are instrumental in meeting customer demand and ensuring the survival of the organization. This paper introduces a novel Fuzzy Cerebellar-Model-Articulation-Controller (FCMAC) with a Truth Value Restriction (TVR) inference scheme for time-series forecasting and investigates its performance in comparison to established techniques such as the Single Exponential Smoothing, Holt's Linear Trend, Holt-Winter's Additive methods, the Box-Jenkin's ARIMA model, radial basis function networks, and multi-layer perceptrons. Our experiments are conducted on the product demand data from the M3 Competition and the US Census Bureau. The results reveal that the FCMAC model yields lower errors for these data sets. The conditions under which the FCMAC model emerged significantly superior are discussed.

Keywords: Time series, forecasting, neural networks, fuzzy, CMAC, truth value restriction.

1 Introduction

From government to education and from public utilities to private industries, forecasting is a necessary tool that enables decision makers to predict changes in demand, plans and sales. These forecasts can be used to enhance a variety of purposes in an organization, including production planning, budgeting, sales quota setting, and human resource planning [1]. The importance of sales forecasting for a firm has often been emphasized [2] as more and more organizations recognize the importance of formal forecasting [3]. Without a forecast, firms in the short run can only respond retroactively, which may result in lost orders, poor service and inefficient utilization of resources. In the long run, this misallocation of resources may have drastic consequences on the firm's ability to survive [4].

Accurate estimation of the future demand of a product can be instrumental in meeting customer demand promptly and precisely, improving relationships with suppliers, cutting inventory levels, and developing a more efficient supply chain. Much literature has been written describing the attributes and factors of good forecasting practices [5] and a large number of empirical studies have been conducted on forecasting in general or sales forecasting in particular [6].

The importance of forecasting has seen the emergence of companies specializing in forecasting packages and software. These range from stand-alone, specialized packages to modules in enterprise management systems. Many of these packages use conventional forecasting methods, such as Simple Exponential Smoothing (SES), Holt's Linear Exponential Smoothing (HLES), Holt-Winter's method with Additive Seasonality (HWA)

[7]. Another popular forecasting method is the Box-Jenkin's Autoregressive Integrated Moving Average (ARIMA) model [8,9].

Newer forecasting techniques and innovative modifications of present techniques have also emerged. These include pattern matching of historical data [10], which relies on old structures of data that may be used for matching with current structures to generate a future prediction. Segmentation of time-series data to enable further manipulation and extraction of useful information has also been explored successfully [11]. These new techniques have been compared to the classical techniques and have yielded encouraging results indeed.

Neural networks have attracted more and more interest in time-series forecasting, thanks to their capabilities to easily model any type of parametric or non-parametric process [12,13]. Encouraging results have emerged from various comparisons of neural networks with the abovementioned classical techniques [14]. However one of the main drawbacks of neural networks is that the whole system works like a black box therefore the network knowledge is not easily extracted and comprehended.

On the other hand, fuzzy systems can handle imprecision that is inherent in human reasoning, especially when dealing with complexity. Based on the concepts of fuzzy sets and fuzzy logic, fuzzy systems encode the linguistic samples in a designated numerical matrix, which links input to output through fuzzy membership functions and sets of fuzzy rules. Therefore fuzzy rules are compact, efficient representations of human knowledge. Fuzzy rules therefore provide a human-like thinking ability which allows expert-knowledge to be incorporated into a system. In order to combine the advantages of two models, fuzzy system is introduced into neural network to obtain a new neural fuzzy system [15]. Neural fuzzy systems have the learning capabilities of neural networks and the advantages of fuzzy systems which make it more robust highly intuitive and easily comprehended. Moreover, neural fuzzy systems have the capability of acquiring and incorporating human knowledge into the systems and the capability of processing information based on fuzzy inference rules from the fuzzy system.

The Cerebellar Model Articulation Controller (CMAC) first proposed by Albus [15,16], is a type of associative memory neural network that models how a human cerebellum takes inputs, organize its memory and compute outputs. CMAC is a table-lookup module that represents complex and non-linear functions. It maps inputs from a subset of N members to a scalar output. Input parameters can be selected in such a way so that each set of input parameters will give an output that is close to the desired output.

The CMAC system has the advantages of fast learning, simple computation, local generalization and can be realized by high-speed hardware. The main disadvantage of CMAC is that the input parameters grow exponentially with the input variables. It is not very efficient in terms of data storage and modeling of problem space. Nevertheless, as a trained CMAC is a black box, it is not possible to extract structural knowledge from the system or to incorporate domain expert prior knowledge. To overcome this shortcoming, fuzzy set theory can be combined with CMAC, which is called Fuzzy CMAC, or FCMAC [17]. The main difference between the FCMAC and the original CMAC is that the association layer in the FCMAC is the rule layer and each associative cell represents a fuzzy rule that links input cluster to the output cluster. Also, the inputs and outputs are fuzzy and form the antecedent and precedent of the fuzzy IF-THEN rule respectively.

In this paper, we propose a novel approach to product demand forecasting by FCMAC with a Truth Value Restriction (TVR) inference scheme called FCMAC-TVR. The purpose of our study is to analyze the forecasting capabilities of this novel model. The structure of this paper is outlined as follows. The next section describes a brief introduction to time series forecasting and a brief description of the classical forecasting techniques used in the experiment. This is followed by Section 3, which then describes in detail the FCMAC-TVR architecture and how it can be used for time series forecasting. Section 4 describes the experiment and presents the results. Section 5 provides the conclusion.

2 Time Series Forecasting

By studying the behaviour of time series data, certain features may be identified and these may help in choosing an accurate forecasting method. Generally, historical time series data may reflect one or more of the following features [19]:

(i) **Trends -** A trend is a gradual upward or downward shift in the level of the series or the tendency of the series values to increase or decrease over time.

(ii) **Seasonal and non-seasonal cycles -** A seasonal cycle is a repetitive, predictable pattern in the series values. A non-seasonal cycle is a repetitive, possibly unpredictable, pattern in the series values.

(iii) **Pulses and steps –** A data set may also experience abrupt changes in level. They generally come in two froms. The first is a sudden, temporary shift or pulse in the series level. The second is a sudden, permanent shift or step in the series level.

(iv) **Intermittent demand –** Intermittent demand refers to time series data which have gaps in the series of recorded values.

The classical forecasting techniques used in our experiment are SES, HLES, HWA and ARIMA [6-8]. These popular techniques will provide us with a credible platform upon which we can compare the performance of our model.

Single Exponential Smoothing (SES) method. The SES method is a forecasting technique in which smaller weights are assigned to older historical data and heavier weights to more recent data [2]. This is to reflect the relative significance that the more recent data may have on the forecast. Simple exponential smoothing assumes that there is no trend or seasonal aspects in the data and that the level of the series changes slowly over time. The expression for SES is as follows:

$$F_{t+1} = (1-\alpha)^t F_1 + \alpha \sum_{j=0}^{t-1} (1-\alpha)^j Y_{t-j} \tag{2.1}$$

where F_{t+1} is the forecast at time *t+1*, Y_{t-j} is the actual value at time *t-j* and α is the smoothing constant which has a value between 0 and 1.

Holt's Linear Exponential Smoothing (HLES) method. Suppose that the data series is non-seasonal but does display a trend [2]. The HLES method extends the SES method to estimate both the current level and the current trend. The expressions for HLES are as follows:

$$\begin{aligned} L_t &= \alpha Y_t + (1-\alpha)(L_{t-1} + b_{t-1}) \\ b_t &= \beta(L_t - L_{t-1}) + (1-\beta) b_{t-1} \\ F_{t+m} &= L_t + b_t m \end{aligned} \tag{2.2}$$

where L_t and b_t are exponentially smoothed estimates of the level and linear trend of the series at time t respectively. F_{t+m} is the linear forecast from t for a forecast horizon m, the α and β are the smoothing constants with values between 0 and 1.

Holt-Winter's Additive (HWA). The Holt-Winter's methods extend HLES to cater for data with both trend and seasonal components [7]. For additive seasonality, the seasonal component S_t is subtracted from Y_t in the level component L_t. It is then added to the forecast F_{t+m}. The expressions for seasonality are:

$$\begin{aligned} L_t &= \alpha(Y_t - S_{t-s}) + (1-\alpha)(L_{t-1} + b_{t-1}) \\ b_t &= \beta(L_t - L_{t-1}) + (1-\beta) b_{t-1} \\ S_t &= \gamma(Y_t - L_t) + (1-\gamma) S_{t-s} \\ F_{t+m} &= L_t + b_t m + S_{t-s+m} \end{aligned} \tag{2.3}$$

ARIMA model. ARIMA forecasts [8,9] are based on linear functions of the sample observations. An ARIMA process has three parts, namely the Autoregressive (AR) part, Integrated (I) and the Moving Average (MA) part. The AR part of the model describes how each observation is a function of a certain number of the pervious ones. The I part of the model determines whether the observed values are modeled directly, or whether the differences between consecutive observations are modeled instead. The MA part of the model describes how each observation is a function of a certain number of previous errors. There are

3 main phases involved in identifying an ARIMA model: identification, estimation and testing phases. In the identification phase, the data is made stationary in mean and variance before the appropriate model is selected. In the next two phases, the model parameters are estimated and residuals are tested if they satisfy the requirements of a white noise process. If the hypotheses on the residuals are validated successfully, the model can be used to generate the forecasts. Otherwise, the residuals contain a certain structure that should be studied and refined in the first phase.

3 FUZZY CMAC with TVR Inference

Classical CMAC has the advantages of local generalization, output superposition, easy hardware implementation, faster training and incremental learning because of the limited number of computations per cycle. However, CMAC suffers from inherent disadvantages like its inefficiency in storing data storage, since the memory size required grows exponentially with respect to the number of input variables and its weak performance in classifying inputs which are similar and highly overlapping.

The motivation of advancing classical CMAC to FCMAC is to increase the learning capability for the model. The introduction of fuzzy membership in respective fields has the effect of smoothing the network output and increasing the approximation ability in function approximation. The FCMAC structure also reduces the memory requirement by a great deal as compared to the original CMAC. Our proposed FCMAC model uses Discrete Incremental Clustering (DIC) [20] for the self-organizing phase and the back-propagation learning algorithm for the parameter-learning phase.

3.1 FCMAC-TVR Architecture

As shown in Figure 1, FCMAC-TVR consists of three phases, namely, fuzzification, inference and defuzzification.

Fuzzification. Discrete Incremental Clustering is used for fuzzification. Each fuzzy label (fuzzy set) belonging to the same input/output dimension has little or no overlapping of kernel with its immediate neighbours. In addition, the number of fuzzy sets for each input/output dimension need not be the same. The detailed description of DIC algorithm can be found in [20].

Inference. In this phase, CMAC works together with TVR, while the former is in charge of weight updating and the latter fulfills logic reasoning. There are two critical layers in CMAC – association layer and pos-association layer. Association layer is a *conceptual/logical memory space* and carries out l*ogical AND operation* to combine the outputs of the sensors. The *AND* operation ensures that any cell is active only when *all the inputs* to it are fired. This layer has the generalization feature, i.e., the similar sensor outputs are mapped to similar cells in the associative layer. To address the problem of a large memory size required in the conceptual space, post-association layer maps it to a *physical memory space* by either *linear mapping or hashing.* The *logical OR operation* makes any cell in this layer fired if *any of its connected inputs* is activated. The work mechanism of TVR will be introduced later on.

Defuzzification. The center of Area (COA) method is applied in this research, see Equation (3.9) for details.

[Figure 3.1 Here]

Truth-value restriction uses implication rules to derive the truth-values of the consequents from the truth-value of the antecedents [21,22]. In the TVR methodology, the degree to which the actual given value of $\hat{A}$ of a variable *x* agrees with the antecedent value *A* in the proposition *"IF x is A then y is B"* is represented as a fuzzy subset of truth space $\tau_{A\hat{A}}$. This fuzzy subset of truth space, known as truth-value restriction, is used in a fuzzy deduction process to determine the corresponding restriction on the truth-value of the proposition *"y is B"*. The latter truth value restriction is then 'inverted', which means that a fuzzy proposition *"y is $\hat{B}$"* in the *Y* universe of discourse is found such that its agreement with *"y is B"* is equal to the truth value restriction derived by the fuzzy inference process.

The fuzzy subset $\tau_{A\hat{A}}$ of the truth space is given in the following equation:

$$\tau_{A\hat{A}} = \begin{cases} \sup_{x}\{\mu_{\hat{A}}(x) \mid \mathrm{x} \in \mu_A^{-1}(a)\}, & \mu_A^{-1}(a) \neq \phi \\ 0, & otherwise \end{cases} \tag{3.1}$$

where μ_A and $\mu_{\hat{A}}$ are the respective membership functions of the fuzzy set A and $\hat{A}$ defined on the universe of discourse *X*; *x* denotes a value in *X*; a is the membership value of *x* in fuzzy set *A*; $\mu_A^{-1}(a)$ is the set of values of *x* in *X* that take membership value *a* in the fuzzy set *A*; and ϕ denotes the empty set or null set.

The truth-value $\tau_{B\hat{B}}$ of the consequent "*y* is *B*" can be computed using the following equation:

$$\tau_{B\hat{B}}(b) = \sup_a\{m_I[\tau_{A\hat{A}}(a), I(a,b)]\} \tag{3.2}$$

where m_I is the forward reasoning function (usually a T-norm operation); and *I* is the implication rule.

The truth-value $\tau_{B\hat{B}}$ is subsequently "inverted" to determine the inferred conclusion. That is, a fuzzy proposition "*y* is $\hat{B}$" is computed using the *truth function modification* (TFM) process such that the degree of possibility that the proposition "*y* is B" is true given "*y* is $\hat{B}$" is described by the truth-value $\tau_{B\hat{B}}$. That is, the derivation of the fuzzy set $\hat{B}$ from the truth-value $\tau_{B\hat{B}}$ is performed using $\mu_{\hat{B}}(y) = \tau_{B\hat{B}}(\mu_B(y))$, where μ_B and $\mu_{\hat{B}}$ are the membership functions of the fuzzy sets *B* and $\hat{B}$ respectively.

In the proposed model, the computed truth-values of the antecedents can be effectively propagated in the hybrid structure of a neural fuzzy system. The truth-value of the proposition in the antecedent is computed and allowed to propagate through the network. It is this value that is used to calculate the proposition in the consequent. This treatment makes the TVR a viable inference scheme for implementation in a neural fuzzy system.

3.2 Work mode of FCMAC-TVR

First of all, self-organizing DIC technique is conducted on the input training data set to obtain fuzzy labels. Then, using AND operator in the CMAC association layer (Figure3.1) we get *P* rules corresponding to an input-output data $(x^k, y^k) = (x_1^k, x_2^k, ..., x_n^k, y^k)$, as shown by the following implications [20]:

Rule^p: If "x_1^k is A_1^p and x_2^k is A_2^p and, …, and x_n^k is A_n^p" Then "y is w^p"

where p = 1,2, …, P, x_i^k is the *k*th input variable, *y* is an output variable, *P* is the number of fuzzy rules, A_i^p are fuzzy sets and w^p is the real number of the *p*th fuzzy inference rule.

Step 1: Singleton Fuzzifier

The crisp-valued input variable in each dimension is first fuzzified by a singleton fuzzifier into a fuzzy set denoted by the membership function:

$$\mu_{\overline{X}_i}(\overline{x}_i) = \begin{cases} 1 & \text{if } \overline{x}_i = x_i^k \\ 0 & \text{otherwise} \end{cases} \tag{3.3}$$

where x_i is the input in *i*th dimension and $\overline{x}_i$ is the fuzzified equivalent of crisp input x_i in the fuzzy set $\overline{X}_i$.

Step 2: Antecedent Matching

From (3.1) and (3.3) the following equation can be deduced:

$$\tau_{i,j}(a_i) = \begin{cases} 1 & \text{if } a_i = \mu_i\ (x_i^k), a_i \in [0,1] \\ 0 & \text{otherwise} \end{cases} \tag{3.4}$$

where $\tau_{i,j}(a_i)$ is the truth value derived at the input-label node $IL_{i,j}$, $\mu_{i,j}$ is the membership function depicting the input-label node $IL_{i,j}$ and x_i is the *i*th element of the input vector.

Step 3: Rule Fulfillment

In this step, the responses of the activated sensors in each dimension are aggregated to compute the response of each cell, which is also the degree of fulfillment of the current input with respect to the antecedents of the fuzzy rules. Only a limited number of cells which are fired by all the input data are active and from (3.4) the total truth value η_p of the *p*th firing cell corresponding to *p*th rule is:

$$\eta_p = \min_{i,j} \left\{ \tau_i^p(a_i) \right\} = \begin{cases} 1 & \forall i, a_i = \mu_i^p(x_i^k) \\ 0 & \text{otherwise} \end{cases} \tag{3.5}$$

where $\tau_{i.j}$ is the fuzzy truth value derived at the input label node $IL_{i,j}$ and $m_i^p(x_i)$ is the membership function representing the input-label node $IL_{i,j}$ which is one of conditions of the *p*th rule.

The truth value-deduction function (3.2) can be rewritten:

$$\tau_i^p(b) = \sup_{a_i = \mu_i^p(x_i)} \left\{ m_I \left[1, I\left(\prod_{i=1}^{n} a_i, b \right) \right] \right\} \tag{3.6}$$

where τ_i^p is the truth-value derived by the pth rule, I[.] is implication functions and m_I is triangular norm and has the following properties: $m_I(0,a)=0$ and $m_I(1,a)=a$, thus, we have:

$$\tau_i^p(b) = \prod_{i=1}^{n} \mu_i^p(x_i) \times b \tag{3.7}$$

A simple scalar multiplication operator "×" is used as the implication function $I[.]$.

Step 4: Consequence Derivation

The term $\prod_{i=1}^{n} \mu_i^p(x_i)$ in Equation (3.7) is the fired strength or the total membership value of the antecedent part of pth rule:

$$\omega^p = \prod_{i=1}^{n} \mu_i^p(x_i^k) \qquad \text{for i=1,2, …, n} \tag{3.8}$$

Step 5: Output Defuzzification

The Center of Area (COA) method [27] is used in the defuzzification to map fuzzy space of fuzzy rules into crisp space of output value. The output value of the mth neuron in the CMAC post-association layer P (Figure 3.1) y_m^τ is derived by the following equation:

$$y_m^\tau = \frac{\sum_{p=1}^{P} \omega^p \times w^p}{\sum_{p=1}^{P} \omega^p} \qquad \text{for } p=1,2,\ldots,P \tag{3.9}$$

where w^P is the real number of the pth fuzzy inference rule.

3.3 FCMAC-TVR Training

Similar to CMAC, the Least Mean Square (LMS) approach is adopted to train the FCMAC-TVR network. The output for the input data x_i^k is derived by:

$$y^\tau = \sum_{m=1}^{M} y_m^\tau \tag{3.10}$$

where y_m^τ is the output of the *m*th neuron in the post-association of CMAC and *M* is the total neurons in this layer for the input data x_i^k.

The weights are updated to achieve least mean square errors:

$$w_{t+1}^p = w_t^p + \alpha \times \frac{y^d - y^\tau}{M} \tag{3.11}$$

where $0 < \alpha < 1$, and y^d is the desired output.

The training stops either when the specified number of cycles or completed or the error converges below a certain threshold ε:

$$y^d - y^\tau < \varepsilon \tag{3.12}$$

4 Experimental Results

The general concept behind CMAC is that similar inputs are clustered in the effort to generate the output. Hence, the key point in our model is to increase the accuracy of the FCMAC-TVR model for time-series forecasting, the input data should be stationary in mean and variance. This will then allow the inputs to be clustered in the effort to generate the output. Data that is non-stationary in mean and variance will have trend and seasonal components that will not enable accurate clustering of input data.

In order to make the data stationary in mean the data must undergo preliminary differencing. This will help remove any trend or seasonal component present. To make the data stationary in variance, log transformation can be applied to the data. Log transformations have the effect of dampening exponential growth patterns and reducing the level of heteroscedasticity.

When the data is stationary, it is fed into the CMAC network. The data will be clustered accordingly in the association layer of the CMAC network. Inline with the principle of a CMAC output, which is "the function with similar outputs for similar inputs", the output for time series forecasting therefore represents the most significant cluster of differences that have been identified in the association layer.

We are now in the position to compare the performance of the FCMAC network with the classical forecasting techniques. Our experiments were conducted using benchmark data from two sources: product demand data from the M3-competition [28] data series and retail sales data from the US Census Bureau [29]. The forecasts for the classical models were generated with commercial software [30] that obtained the ideal parameters of each model. This provided an accurate platform with which we could compare the performance of the FCMAC-TVR.

There are a variety of other accuracy measures that are available in the forecasting literature, however each measure having its own advantages and disadvantages. Our study will focus solely on the Mean Absolute Percentage Error (MAPE) statistic [31]. MAPE is a comparative measure that does not have the problem of averaging the positive and negative errors. It is relatively easy to justify a model's effectiveness:

$$MAPE = \left(\sum_{t=1}^{n} \left(\left| \frac{A_t - F_t}{A_t} \right| \right) \times 100 \right) / n \quad (4.1)$$

where n is the number of times series data points, A_t and F_t refer to the actual results and forecast results at time t respectively.

4.1 M3 Domestic Product Demand Data

The M3-competition was conducted in 1997 by the International Journal of Forecasting. It compared a range of forecasting techniques across a range of measures on a holdout set. 3003 sets of historic time series data were collected to cover as wide of a range of data types as possible (e.g., microeconomic, macroeconomic, industrial, financial and demographic) and included monthly, quarterly and annual series. The competition attracted forecasting experts from academic institutions such as the Wharton School, Case Western Reserve, INSEAD and the Imperial College, who tested their forecasting techniques with this data set. The relative performances were then tabulated and published in reputable journals. The M3-competition data set has since been widely available and used by researchers to analyze the performance of forecasting techniques.

From the 3003 time series datasets, *Domestic product demand (N2071)* data were singled out in relevance to product demand forecasting. The values represent monthly statistics pertaining to the demand for a range of goods and the numbers in brackets indicate the number of the series as presented in the M3-competition data set. In our experiment, the most recent 96 monthly values were used to compare the performance of the FCMAC-TVR model. These values are sufficient to train the neural network as well as offered 3 time origins (36, 60 and 84) from which forecasts were generated up to a horizon of 12 (one-year). The *ex-ante* form of forecasting was used. Hence the values not included in the parameter estimation were used as a holdout set over which the MAPE error statistic was measured. For the classical models, the parameters of each method were optimized by the commercial software [30] at every time origin. For instance, the ideal parameters for M3 N2071 data set at time origin 36 were set as $\alpha = 0.56$ for SES (Equation (2.1)), $\alpha = 0.62$ for HLES (Equation (2.2)) and $\alpha = 0.62$ $\beta = 0.24$ $\gamma = 0.14$ for HWA (Equation (2.3)). This ensured that all the available time series information was used in determining the parameters of the method and model prior to the forecast being made.

[Figure 4.1 Here]

Figures 4.1 shows the relative performance of different models with respect to the holdout data of 12 values for a particular time series and time origin. These two instances have been specially singled out as they exhibit the characteristics of the FCMAC-TVR model and the results show that proposed model's performance is better than other techniques. To make our comparison clearer, only the performances of ARIMA and FCMAC-TVR are presented in

Figure 4.1. As a matter of fact, FCMAC-TVR predicts the closest values to the actual line at most of the data points. Both ARIMA and HWA perform better than HLES. Over the respective holdout periods of 12 values in Figure 4.1, it is obvious that the data pattern does not show any stationarity in mean or variance or a very significant semblance of a consistent trend or seasonality. Under such circumstances, the novel approach of clustering inputs in the FCMAC-TVR results in a model that can better adapt to irregular patterns in the data as compared to the classical techniques.

4.2 US Retail Sales

The following data sets were retrieved from the US Census Bureau [29]. They refer to US retail sales and hence reflect product demand of common goods from common industries.

- Grocery stores (NAICS 4451)
- Clothing stores (NAICS 4481)
- Sporting goods, hobby, book & music stores (NAICS 451)
- Household appliance stores (NAICS 443111)

Each of the above four datasets contains monthly retail sales amounts (millions of US Dollars) in the United States since January 1992. We extract the entries from January 1992 to December 2004, i.e., 156 entries in each dataset were extracted for our experiments. Table 4.1 shows some samples of these four datasets, which are described by three attributes: year, month and amounts.

[Table 4.1 Here]

Figure 4.2 is shown to identify the patterns and main traits of these four series.

[Figure 4.2 Here]

The architecture of FCMAC-TVR for these four data sets is: 48 input nodes and 1 output node. Before neural network training, the DIC technique groups the training data into a few clusters (fuzzy sets). The weight updating is carried out by CMAC association layer and post-association layer, whereas the logic inference is conducted by truth value restriction technique.

We run and test FCMAC-TVR for ten times and report the average results, which are compared with SES, HLES, HWA, ARIMA, Radial Basis Function (RBF) [32,33] and Multilayer Perceptron (MLP) [34] in Table 4.2. We assume a *t*-distribution of errors, and the confidence intervals are stated at 95% levels, hence the confidence intervals of MAPE in Table 4.2 are calculated by the following formula:

$$95\%\ \mathrm{C.I.}\ MAPE = MAPE \pm (1.96 \times \frac{SD}{\sqrt{10}}) \qquad (4.2)$$

Where the value 1.96 comes from the understanding of the normal curve, *SD* is the stand deviation of MAPE of the 10 experiments.

[Table 4.2 Here]

Table 4.2 indicates that the proposed FCMAC-TVR model outperforms all the other methods in all forecasting horizons. The RBF comes out the second best due to the fact that its approximate capability of complex non-linear mappings. HWA achieves basically the same level of accuracy as RBF, thanks to the additive seasonality component inherent in the test data. The performance of MLP is not good enough, because of the problems with overfitting as well as determination of the number of hidden units. The ARIMA is able to model a wider range of data patterns, so it performs better than SES and HLES, which cannot do very well with long forecast horizons.

The superior performance of FCMAC-TVR benefits from its DIC fuzzification technique and TVR inference scheme. The DIC technique self-organizes clusters from a set of raw training data. The truth-value restriction inference scheme provides FCMAC with an intuitive fuzzy logic-reasoning framework.

5 Conclusion

This paper introduces a novel FCMAC with TVR inference scheme for product demand forecasting with time series data. This neural network has the characteristic of high learning speed and localization. TVR inference scheme makes the system more understandable and less rigid. It also gives the network a consistent rule base and a strong theoretical foundation. The proposed model uses DIC for the self-organizing phase and the LMS minimization for the parameter-learning phase. Our experiment investigates the performance of the FCMAC-TVR model in comparison to popular time-series forecasting techniques such as SES, HLES, HWA, ARIMA, and some other neural network models such as RBF and MLP. The experiments were conducted using data from the M3 competition and the US Census Bureau. The results show that the FCMAC-TVR model outperforms all the other techniques. This is especially so when the data series does not exhibit significant and consistent characteristics such as stationarity, trend and seasonality. In this respect, the FCMAC-TVR model is able to better accommodate the irregular pattern in the data series than the classical techniques.

References

[1] J. T. Mentzer and J. E. Cox, Jr., *"Familiarity, application and performance of sales forecasting techniques",* Journal of Forecasting, **3(1):**27–36, 1984.

[2] S. Makridakis and S. C. Wheelwright, *"Forecasting Methods for Management",* 5th edition, Wiley, Chichester, 1989.

[3] C. L. Jain, "*Ivy league business schools far behind the time",* Journal of Business Forecasting, **13(4):**2, 1991.

[4] R. Fildes and R. Hastings, "*The organization and improvement of market forecasting*", Journal of the Operational Research Society, 45, Pages 1–16, 1994.

[5] D. H. Drury, *"Issues in forecasting management",* Management International Review, **30(4):**317–329, 1990.

[6] H. Winklhofer, A. Diamantopoulos and S. F. Witt, *"Forecasting practice: A review of the empirical literature and agenda for future research",* International Journal of Forecasting, **12:**193–221, 1996.

[7] P. R. Winters, *"Forecasting sales by exponentially weighted moving averages",* Management Science, Vol. 6, Pages 324–342, 1960.

[8] G. E. P. Box and G. M. Jenkins, "*Time Series Analysis*, Forecasting and Control", San Francisco: 2nd Edition, Holden Day, 1976.

[9] R. D. Snyder, A. B. Koehler and J. K. Ord, *"Forecasting for inventory control with exponential smoothing",* International Journal of Forecasting, **18(1):** 5-18, 2002.

[10] S. Singh and E. Stuart, *"A pattern matching tool for time-series forecasting",* Proceedings of the International Conference on Pattern Recognition, Vol.1, Pages 103-105, 1998

[11] F. L. Chung, T. C. Fu, R. Luk and V. Ng, *"Evolutionary time series segmentation for stock data mining",* Proceedings of the IEEE International Conference on Data Mining (ICDM), Pages 83- 90, 2002.

[12] S. D. Balkin and J. K. Ord, *"Automatic neural network modeling for univariate time series",* International Journal of Forecasting, **16(4):** 509-515, 2000.

[13] J. Teo and D. Wood, *"Neural network protocols and model performance",* Neurocomputing, **55(3):** 747-753, 2003.

[14] C. W. Chu and G. P. Zhang, *"A comparative study of linear and nonlinear models for aggregate retail sales forecasting",* International Journal of Production Economics, **86(3):** 217-231, 2003.

[15] C. Y. Wen and M. Yao, *"A combination of traditional time series forecasting models with fuzzy learning neural networks",* Proceedings of the International Conference on Machine Learning and Cybernetics, Vol. 1, Pages 21- 23, 2002.

[16] J. S. Albus, *"A new approach to manipulator control: the cerebellar model articulation controller (CMAC)",* Transactions of ASME Journal of Dynamic Systems, Measurements, and Control, **97(3):** 220-227, 1975.

[17] J. S. Albus, "*Data storage in the cerebellar model of articulation controller (CMAC)",* Transactions of ASME Journal of Dynamic Systems, Measurements, and Control, **97(3):** 228-233, 1975.

[18] J. Ozawa, I. Hayashi and N. Wakami, "*Formulation of CMAC-Fuzzy system*", IEEE international Conference on Fuzzy Systems - Fuzzy-IEEE, San Diego, CA, pp. 1179-1186, 1992.

[19] S. Makridakis, S. C. Wheelwright and R.J. Hyndman, *"Forecasting methods and applications (3rd ed.)",* New York: Wiley, 1998.

[20] W. L. Tung and C. Quek, "*DIC: A Novel Discrete Instrumental Clustering Technique for the Derivation of Fuzzy Membership Functions*", Proceedings of the 7th Pacific Rim International Conference on Artificial Intelligence: Trends in Artificial Intelligence, 2002.

[21] C. T. Lin and C. S. G. Lee, "*A Neuro-Fuzzy Synergism to Intelligent Systems*", Neural Fuzzy Systems, Upper Saddle River, NJ: Prentice-Hall, 1996.

[22] C. Quek and R. W. Zhou, "*POPFNN: A Pseudo Outer-Product Based Fuzzy Neural Network*", IEEE Transaction, Neural Networks, 9(9): Pages 1569-1581, 1996.

[23] L. A. Zadeh, "*Calculus of fuzzy restrictions*", In: Fuzzy sets and Their Applications to Cognitive and Decision Processes, New York: Academic, Pages 1-39, 1975.

[24] K. Ang, C. Quek and M. Pasquier, "*POPFNN-CRI(S): Pseudo Outer Product based Fuzzy Neural Network using the Compositional Rule of Inference and Singleton Fuzzifier*", IEEE Transactions on Systems, Man and Cybernetics, **33(6):**838-849, 2003.

[25] I. B. Turksen and Z. Zhong, "*An approximate analogical reasoningschema based on similarity measures and interval-valued fuzzy sets*", Fuzzy Sets System, **34:**323–346, 1990.

[26] C. Quek and R. W. Zhou, "*POPFNN-AARS(S): A pseudo outer-product based fuzzy neural network*", IEEE Transaction Systems, Man & Cybernetics, **29(6):**859-870, 1999.

[27] E. S. Lee, Q. Zhu, "*Fuzzy and Evidence Reasoning*", Physica-Verlag, 1995.

[28] M. Hibon M. and S. Makridakis, *"The M3 Competition: results, conclusions and implications",* International Journal of Forecasting, **16(4):**451-476, 2000.

[29] http://www.economagic.com

[30] http://www-marketing.wharton.upenn.edu/forecast/software.html#Commercial_Programs

[31] S. Makridakis, S. C. Wheelwright and V. E. McGee, *"Forecasting: methods and applications",* 2nd Edition, John Wiley, New York, 1983.

[32] C. M. Bishop, "*Improving the generalization properties of radial basis function neural networks*", Neural Computation, **3(4):**579-581, 1991.

[33] D. Shi, D. S. Yeung, J. Gao, "*Sensitivity analysis applied to the construction of radial basis function network*", Neural Networks, **18(7):**951-957, 2005.

[34] C. M. Bishop, *Neural Networks for Pattern Recognition.* Oxford University Press, 1995.

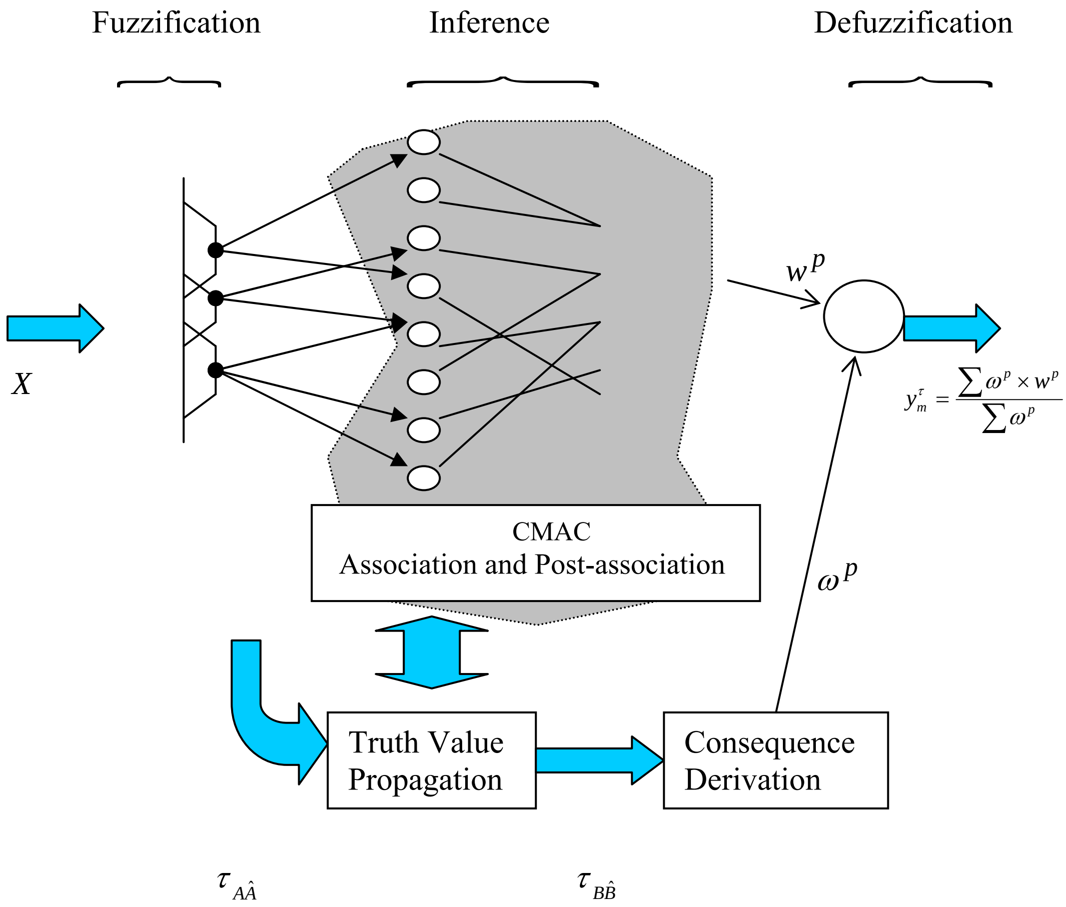

Figure 3.1 Block diagram of FCMAC-TVR

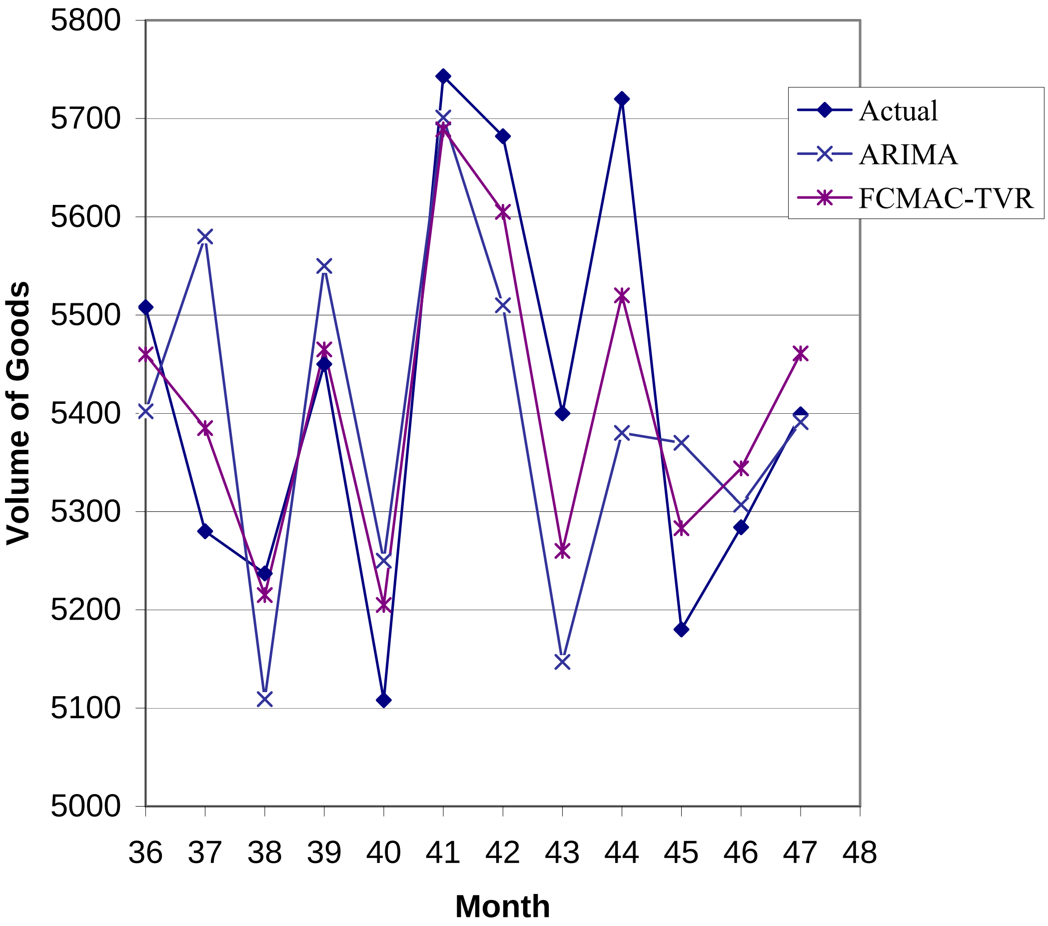

Figure 4.1 Performance of different methods on M3 domestic product demand data

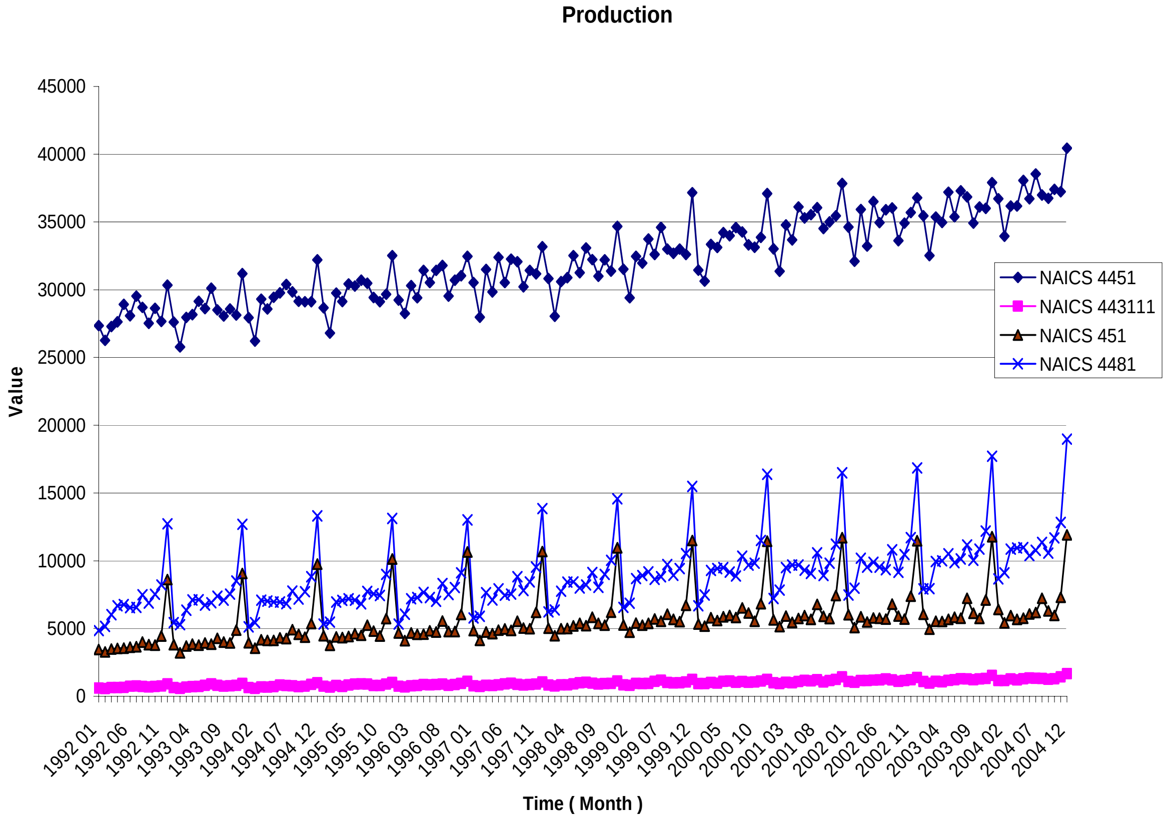

Figure 4.2 Monthly US retail sales from 1992 to 2004

Table 4.1 Sample Data of US Retail Sales

NAICS4451	NAICS4481	NAICS451	NAICS443111
…	…	…	…
1992 10 28317	1992 10 7238	1992 10 4154	1992 10 733
1992 11 28345	1992 11 7145	1992 11 4156	1992 11 754
1992 12 28278	1992 12 7432	1992 12 4314	1992 12 918
1993 01 28236	1993 01 7658	1993 01 4321	1993 01 643
1993 02 28465	1993 02 7245	1993 02 4183	1993 02 594
1993 03 28353	1993 03 7097	1993 03 4174	1993 03 668
1993 04 28384	1993 04 7328	1993 04 4380	1993 04 701
1993 05 28459	1993 05 7401	1993 05 4345	1993 05 726
1993 06 28422	1993 06 7334	1993 06 4351	1993 06 803
1993 07 28517	1993 07 7414	1993 07 4264	1993 07 921
1993 08 28401	1993 08 7297	1993 08 4309	1993 08 818
1993 09 28550	1993 09 7375	1993 09 4340	1993 09 741
…	…	…	…

Table 4.2: MAPE values of the different methods on US Retail Sales datasets

Method	95% Confidence Intervals of Forecasting Results in Different Horizons						
	1	3	6	12	1-3	1-6	1-12
SES	7.13±0.22	7.19±0.29	7.24±0.25	7.26±0.22	7.16±0.21	7.19±0.26	7.20±0.19
HLES	7.48±0.27	7.51±0.26	7.60±0.25	7.62±0.21	7.50±0.24	7.53±0.23	7.56±0.22
HWA	1.29±0.22	1.30±0.24	1.30±0.22	1.31±0.25	1.30±0.19	1.30±0.25	1.30±0.22
ARIMA	1.99±0.13	1.99±0.14	2.01±0.11	2.01±0.12	1.99±0.17	1.99±0.21	2.00±0.20
MLP	3.18±0.25	3.04±0.22	3.53±0.16	3.68±0.17	3.05±0.24	3.17±0.17	3.54±0.21
RBF	1.53±0.19	1.30±0.16	**1.14**±0.12	1.59±0.14	**1.26**±0.12	1.49±0.18	1.33±0.15
FCMAC-TVR	**1.21**±0.17	**1.22**±0.15	1.26±0.15	**1.26**±0.12	1.29±0.18	**1.29**±0.14	**1.30**±0.13