

An Improved Neural Approaches for Forecasting Demand in Supply Chain Management

Mariem Mrad

Laboratory of Economics and Management
University of Sfax, Tunisia
Airport Road Km 4 Sfax, 3018 Tunisia

Younes Boujelbene

Laboratory of Economics and Management
University of Sfax, Tunisia
Airport Road Km 4 Sfax, 3018 Tunisia

ABSTRACT

Demand forecasting plays a pivotal role for supply chain management. It allows predicting and meeting future demands of the product and expectations of customers. Several forecasting techniques have been developed, each one has its particular benefits and limitations compared to other approaches. This motivates the development of artificial neural networks (ANNs) to make intelligent decisions while taking advantage of today's processing power. Well, this paper deals with an improved algorithm for feedforward neural networks. Initially, the neural modelling process will be discussed. The approach adopted of neural modeling will be presented in a second time; this method is based on mono-network neural modeling and multi-network neural modeling. The results of simulation obtained will be illustrated by a simulated time series data.

General Terms

Backpropagation algorithm; Feedforward Neural Networks; Multilayer Perception; Nonlinear systems.

Keywords

Neural Networks; Supply chain management; Demand Forecasting; Time series forecasting.

1. INTRODUCTION

In the literature of supply chains (SC)s, many authors present diverse definitions. For instance, Chopra and Meindl (2001) define the supply chain as “a dynamic structure involving the constant flow of information, product, and funds between different phases”. On the other hand, Lu and Swaminathan (2015) state that “a supply chain is the set of entities that are involved in the design of new products and services, procuring raw materials, transforming them into semi-finished and finished products, and delivering them to the end customers”. As such, we define the supply chain as a process that encompasses three important phases; supply, production, and distribution containing not only manufacturer and suppliers, but also transporters, warehouses, retailers, and customers themselves. The flow of information, knowledge, resources between and among these entities must be managed appropriately flow between departments is the mainly connection links for SC's success.

Forecasting is the downstream part of supply management picture and directly affects both quantity and delivery. Forecasts of usage, supply, market conditions, technology, prices, and so on, are always necessary to make good decisions (Bousqaoui and al., 2018). Besides, forecasting the expected demand over lead with one or more products is one of the most relevant targets in an organization. It is unenviable to be able to know or predict future demand as close to reality as possible. Thus, accurate demand forecasting could achieve

a lot in terms of maximizing profit, increasing sales, efficient production planning, etc. It is central to the planning and operation of retail business and macro level (Efendigil and al., 2008). But, Despite the need for accurate forecasts to improve commercial competitive advantage in a constantly changing business environment, there is no standard approach.

To achieve this end, this paper suggests a modeling of demand forecasting is provided via a multi-network neural modeling. The rest of this paper is organized as follows. In section 2 the modeling of demand forecasting is described. In section 3 several approaches based on mono-network neural modeling and multi-network neural network are presented. Section 4 illustrates the obtained simulations results. Finally, concluding remarks are presented in section 5.

2. LITERATURE REVIEW

2.1 Supply chain Management and Demand Forecasting

A supply chain management (SCM) —It is the management of flow of goods and services. It involves the movement and storage of raw materials, of work in process inventory, and of finished goods from point of origin to point of consumption (Chawla, Singh, Lamba, Gangwani and Soni, 2019). In brief, it can describe as process that include order generation, order taking, information feedback and the efficient and timely delivery of goods and services.

Demand forecasting plays a key role in SCM. There are several forecasting methods that can be classified into four main categories: (1) Times-series methods extrapolate historical data to provide forecast. (2) Qualitative methods are essentially subjective; they hinge primarily on human judgement and opinion to provide forecast. (3) Causal methods assume that the demand forecast is extremely matched with some factors in the environment. (4) Simulation methods emulate the consumer choices that lead to demand to reach forecasts (Chopra and Meindl, 2001).

Several studies have been involved to predict the customer demand mainly based on time-series models, such as moving-average, Box-Jenkins method, and exponential smoothing, and causal models, such as econometric and regression models. Lee, Padmanabhan, and Whang (1997) described the demand variance amplification while moving through to upstream echelons from downstream echelons as the bullwhip effect. Chen, Drezner, Ryan, and Simchi-Levi (2000) quantified this effect for simple supply chains in two steps into one retailer and one manufacturer. Their models included two factors generally taken to cause the bullwhip effect: demand forecasting and order delays. They extended these results to the multiple supply chains steps with and without demand information centralized customer and the

demonstrated that the bullwhip effect can be reduced but not completely eliminated, by the centralization of information on demand. Chen, Ryan, and Simch-Levi (2000) demonstrated that the use of an exponential smoothing predicted by retailer could result in the bullwhip effect. The writers suggested that magnitude of the increase in variability depends both on the nature of the customer demand process and the forecasting technique used by retailer. Zhao, Xie, and Leung (2002) investigated that the impact of forecast accuracy on the value of supply-side information sharing is strongly moderated by demand variability, which is expressed in different models of demand that retailers face.

Although the quantitative methods described above act as well, but they complain about some shortcomings. First, a rich and diverse database is often required to guarantee an accurate and meaningful forecast. Secondly, it can draw on the lack of experience of a functional defect that links independent and non-independent variables, leading to a low decline. Finally, it is difficult to capture non-linear models. These deficits can be solved using neural networks, which have been mathematically proven to be almost as universal approximation of functions. (Garetti and Taisch, 1999).

2.2 Forecasting and artificial intelligent in supply chain management

Commonly, ANNs are extraordinary computing information and processing methods can be used handle the complicated for many tasks such as function approximation, identification of complex systems and time series forecasting (Abid and al., 2014).

Providing better forecasting approaches to reduce or eliminate inventories affecting the overall cost of SCM is becoming an important issue nowadays. As a new implement, ANN has previously been used in demand forecasting systems or as data pre-processors to refine and classify noisy data to match the relationships between complex functions.

The supply chain can be modeled on different levels, such as materials processing, manufacturing, distribution, customers, etc. In where neural networks are considered a method to solve the problem and assistance problem solving methodology. The areas that are used in supply chain neural networks are: optimization, forecasting, modeling, simulation, globalization and decision support (Leung, 1995).

Given the increasing complexity and ambiguity, the question of demand forecasting has been studied in different ways to achieve satisfactory results. Artificial neural methods have been recently been used and convincing results have been obtained in the areas of demand forecasting. While, in SCM literature, there is no publication that used mono and multi-neural network modeling to forecast demand. There is no evidence that any was applied to the issue of demand forecasting in SCM using a neural technique, specifically. Thus, this study is the first attempt to develop a modular approach. The proposed model including ANN is explained in the following section.

3. NEURAL MODELING AND SYNTHESIS

3.1 Neural networks

Neural networks, as powerful computational tools, have been generally applied to a wide range of tasks, such as function approximation, pattern recognition, identification of complex systems and times series forecasting (Benkachcha and al.,

2013). Modeling with neural networks requires the stage of selection of model, which is an important phase in the design of a neural network.

There are various kinds and architectures of neural networks depending fundamentally on the manner how they learn. In this work, the multi-layer perception approach is used. Many researchers have focused on learning neural networks problems and several algorithms have been developed. Higher accuracy and faster convergence are a crucial issue in choosing the appropriate training algorithm.

The most popular technique for training multilayered is known as the back propagation (BP) algorithm (Narendra and Parthasarathy, 1991). The use of this algorithm is not always successful due to its sensitivity to learning parameters, initial state and perturbation (Yu et al., 2002). There has been much work on the convergence of (BP) algorithm by using the gradient method (Gori and Maggini, 1996). Also, different versions of (BP) learning algorithms have been proposed, such as on-line algorithm for dealing with varying inputs (Zhao, 1996) and the Levenberg-Marquardt-algorithm (Hagan and Menhaj, 1994).

In this paper, training algorithm with one hidden layer have been presented. According to various studies (Homik, 1991; Narendra and Parthasarathy, 1991), a single layer of hidden neurons is used to resolve most problems and is made the approximation of some functions thanks to its universal approximation capability.

Consider that the system model is describes by the following recurrent equation:

$$y(k+1) = F[y(k), y(k-1), \dots, y(k-n+1), u(k), u(k-1), \dots, u(k-m+1)] \quad (1)$$

Where u and y represent respectively the input and the output vector of the network, m is the order of u , F is assumed unknown nonlinear function.

$[y(k), y(k-1), \dots, y(k-n+1), u(k), u(k-1), \dots, u(k-m+1)]$ represents the input vector.

The neural model can be used therefore to provide the estimated output $y_m(k+1)$ of the process at time $k+1$ based on the input values and output values at time k .

Each neuron is connected to all of the next layer by connections whose weights are arbitrary real numbers. A neuron in the hidden layer or in the output layer combines its inputs into single value, that which it transforms afterwards to produce the output This transformation is called the activation function.

For multi-layer networks, the sigmoid activation function is defined as: $(x) = \frac{1}{1+e^{-x}}$.

3.2 Direct Neural Network « DNM »

A simple (DNM), with a single output may be described in Fig. 1.

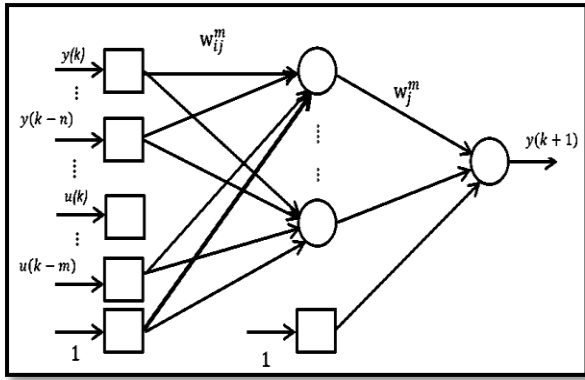


Fig 1: Structure of a direct neural model

In this work we mean by the name the direct neural model the global neural model. To obtain the global model we use one neural network this is called the mono-network neural modeling.

This model is easily synthesized. For other more complex case, another structure modeling can be considered; it will be detailed in section 4.

3.2.1 Learning

Artificial neural networks (ANN) can be considered as a black box containing learned and memorized information. But at startup this box is empty, it contains neither information nor knowledge, learning is necessary. The learning process is a phase during which the behavior of the network is modified until the desired behavior is achieved. Learning of "ANN" consists of adapting its different parameters (weight) according to an iterative algorithm of adjustment or adaptation. The learning process takes into consideration all the examples provided at the entrance. Learning algorithms yield better results for multiple and varied input examples (Solla, 1989).

The learning procedure requires:

- A training set consisting of N examples, each one consisting of a vector applied to the input network and a vector of desired values of outputs. This training set must be rich enough that it covers the maximum possible area of operation for the network.
- Defining a cost function which measures the difference between network outputs and desired outputs present in the training set.
- An algorithm for minimizing the cost function.

$$E_r = \frac{1}{2} [y_p(k+1) - y_m(k+1)] \quad (2)$$

$y_p(k+1)$ and $y_m(k+1)$ respectively represent the output of the process and that of the model at k+1.

Most "neural network" learning algorithms are optimization algorithms: they seek to minimize, by nonlinear optimization methods, a cost function that is a measure of the gap between the network responses and the desired responses. This optimization is done iteratively, by modifying the weights as a function of the gradient of the cost function: the gradient is estimated by a method specific to neural networks, the so-called backpropagation method. The backpropagation algorithm is one of the most used supervised algorithms for learning multilayer perceptions (MLP). Generally, the learning is done over a relatively long period, and includes four stages of calculation:

Generally, learning requires a relatively long period, and includes four stages of calculation:

1. Initialization of synaptic weights of the network.
2. Presentation of the input vector and propagation of states.
3. Calculation of the error at the output of the network.
4. Calculation of the vector of correction.

For this type of network, we use the back-propagation algorithm as a learning algorithm.

3.2.2 The back propagation algorithm

In an "MLP", for each example presented to the network, an estimated output is calculated by propagating the calculation from one layer to another to the output layer, then an error will be computed and then backpropagated in the network to end. adjust each weight. The same procedure is repeated for all learning examples. This process is repeated until the outputs of the network are sufficiently close to the desired outputs.

Denote that $y(y_1, \dots, y_m)$ is the vector of desired outputs and the sigmoid function f , its derivative is: $f(x) = f(x)(1 - f(x))$.

The input and output of a neuron v are respectively denoted by I_v and O_v .

$$\begin{cases} I_v = \sum_j W_{jv} O_j \\ O_v = f(I_v) \end{cases} \quad (3)$$

The error corresponding of an example p is given by:

$$E_p(w) = \sum_{i=1}^m \frac{1}{2} (S_i^p - Y_i^p)^2 \quad (4)$$

The back-propagation algorithm is an approximation of the gradient method. It is defined by:

$$W_{uv}(t+1) = W_{uv}(t) - \varepsilon(t) \frac{\partial E^p(w)}{\partial W_{uv}} \quad (5)$$

t denotes the numbers of iterations, $\varepsilon(t)$ is the step of gradient. This equation requires the calculation of W_{uv} .

Suppose that $d_v = \frac{\partial E^p}{\partial I_v}$, so we have:

$$W_{uv}(t+1) = W_{uv}(t) - \varepsilon(t) d_v O_u \quad (6)$$

This learning algorithm depends on several factors (Rivals, 1995):

The initialization of the network parameters: the choice of the initial values of the weights and the iteration step must be done in a way that ensures a fast and stable convergence.

The complexity of the learning base and the order of presentation of the examples.

The structure of the "Neural Networks" considered must be chosen in an appropriate way, especially the number of neurons of the hidden layer which must be optimal.

The main goal of learning is to make a network capable of generalizing. It is dangerous to continue indefinitely the learning phase without control (Solla, 1989). The learning capacity of a neural network is so strong that after a certain number of iterations, the synaptic weights are able to predict almost the data without error. At this stage, the neural network is no longer mistaken in its predictions, but there is a great risk that these predictions are just as good as the data on which learning was based. This is called learning by heart or

over-learning. This phenomenon is then prevented by adding a phase to the modeling procedure called the generalization or validation phase.

3.2.3 Generalization

The concept of generalization for a neural network is used to accurately measure model performance for a given problem once learning is complete. The generalization is manifested by an input basis, unknown to the network, given to the model to test its ability to generalize; if error is minimal, we say that the process has learned otherwise, so we speak of over learning. It represents one of the features of neural networks because they are able to generalize from a test.

This phase is influenced mainly by three factors: the number and quality of the learning examples, the complexity of the learning algorithm used and the size of the network. (Narendra and Parthasarathy, 1991).

4. THE PROPOSED APPROACH

Artificial intelligence forecasting techniques have been receiving much attention lately in order to solve problems that are hardly solved by the use of traditional methods. They have been cited to have the ability to learn like humans, by accumulating knowledge through repetitive learning activities (Efendigil, 2008). For that, the main objective of this paper is to propose new forecasting techniques via artificial approaches to manage demand in fluctuating environment. In this work, a modular neural networks technique is presented for customer demands in a multi-level supply chain structure.

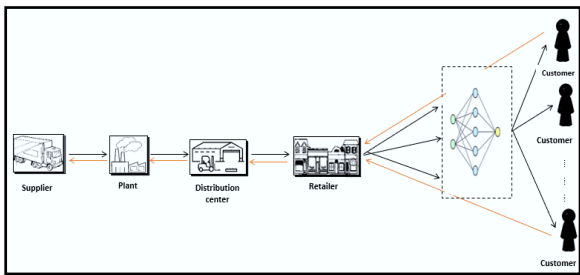


Fig 2: A simple supply chain structure

A typical SC includes different levels of enterprises as shown in Fig. 2. The first level organization is retailer where products are presented and sold to customers; the second level organization is distribution center or warehouse where products are delivered from plans to retailer; the third level organization is a plant where products are produces according to determined production and inventory scheduling; the fourth level organization is a supplier where raw materials are transported to plants to be produced. Although SCs have many different aspects, we focused on customer side related to demand forecasting. Fig.2 shows the supply and demand chain structure between the supply chain entities. The examined side of numerical example is also pointed out in Fig.2.

Modeling a nonlinear system must take into account its complexity. Indeed, this complexity can make it difficult or even impossible to represent a complex system by a single model. So, we assume a number of assumptions to reduce complexity while guaranteeing a certain degree of realism and efficiency of modeling. The approach studied in this work proposes a method for multi-network neural modeling. It requires more than a neural network to obtain the fused neural

model. The fused model of a system is an interpolation of local models according to their degree of validity associated with these models.

We will present first local models then fused model (The multi-network model).

4.1 local models

When we represent a nonlinear system as a multi-model, the problem of modeling nonlinear systems is reduced to modeling sub-systems defined by local models (Gasso, 2000).

These models are complementary; each one has an appropriate area of operation and may have areas of overlap.

The determination of local models can be made as follows: We decompose the domain of the training set given to diverse restricted sets that represent a small part of the learning basis overall. So, we obtain sub-models called local models. Its training basis depends on the operation area used.

The input vector of the neural network used for local modeling is only a part of that used for global modeling.

$$u(k) \in D_u = \prod_i^m D_u^i$$

With: $u(k)$: the command input of the global model, D_u : the domain of the command input of the global model and D_u^i : the domain of the command input of each local model.

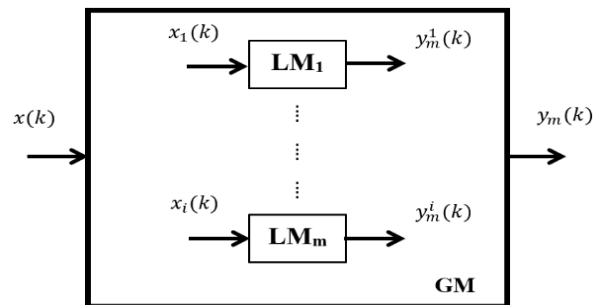


Fig 3: Structure of local models

$x(k)$ and $x_i(k)$ represent respectively the input of the global model and that of the local model.

$y_m(k)$ and $y_m^i(k)$ represent respectively the estimated output of the global model and that of the local model.

The global structure of this multi-network may undergo a learning phase of the complete structure. Another type of learning is to train each local network itself. This second type is the approach used in this work.

The learning procedure of local neural model is the same as that of the learning process of the global neural network. It will be used locally. The criterion to minimize is defined by (2).

Since the number of hidden neurons depends on the data basis presented to the network and the basis of the local model is less rich compared to the global model, the number must be less than the global model. The generalization of such models respects the operating area of the local model itself.

4.2 Fused Neural Model: Multi-Network Neural Model

A multi-network neural model is a model whose architecture meets all local models of the process through an appropriate fusion method. The fusion occurs at the level of outputs generated from each local model.

The learning of this model reduces to the learning of local modes.

There are several methods to determine the fused model. The fusion method adopted in this paper is based on the implementation of fuzzy techniques in which membership functions are related to validity functions associated to the considered model.

The calculation of the output of the fused model is based on a fuzzy decision mechanism which requires a group of fuzzy rules.

We are based on the division's area of command variable in zones; each zone is characterized by a membership function.

The fuzzy rules are as follows:

If $u(k)$ is A_i then $y_m^i(k+1) = MN_i[y(k), \dots, y(k-n+1), u(k), \dots, u(k-m+1)]$

The A_i are fuzzy sub-sets characterized by membership functions of Gaussian type centered in c_i .

The output of the fused model takes the following form:

$$y(k+1) = \frac{\sum_i \mu_i y_i(k+1)}{\sum_i \mu_i} \quad (7)$$

With:

$$\mu_i = \frac{e^{-(u-c_i)^2}}{2\sigma^2} \quad (8)$$

σ represents the opening of the Gaussian and we assume that $\sum_i \mu_i = 1$.

Assume that the system model is described by the equation (1).

Denote that $e_i(k)$ the input vector of neural network i .

The following figure represents the general structure of the multi-model neural network.

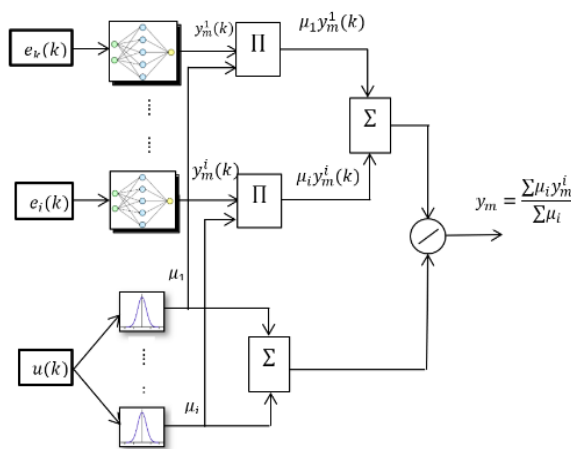


Fig 4: Structure of a multi-network neural network

This multi-network model will be compared to the mono-network model.

5. SIMULATION RESULTS

In this section, we present the simulation results. The application of the modeling approach is showed. We use this modeling approach for the neural identification of a Dynamic Supply Chain Demand. The neural modeling was done through a neural network with one hidden layer and one output. The activation function used in the example is the sigmoid function and the network generates the output $y_m(k+1)$.

5.1 Non-Linear System of Supply Chain management

The most significant factor that impact the demand are determined in the light of documents of Aburto and Weber (2007) according to experts from the retailer. We denote that the demand forecasting system for this study constructed by unit sales price (input) which is a competitive factor affecting the customer behaviors, especially for independent retailer. It is processed as quantitative information. Furthermore, demand quantity (output) which is the quantity from customers to retailer, it is processed as quantitative information.

We suppose that the data were obtained from each retailer via the dynamic mathematical model described by the following equation:

$$y(k+1) = u(k) + \frac{y(k)}{1+y^2(k-1)+u(k-1)} \quad (9)$$

$u(k)$ and $y(k)$ respectively represent the input signal applied to the process and the output of the system at time k for 600 examples. The input variable varies in $[0,200]$

$x(k) = [u(k), u(k-1), y(k), y(k-1)]$: regression vector.

These variables have undergone a phase of normalization where they can cover the domain of variation of the sigmoid function.

5.1.1 Mono-Network Neural modeling: Global Neural Network Model

The input $u(k)$ is set on a working horizon of 600 examples. This horizon is composed of three zones of operation, each one is set to 200 points. Several tests were made to determine the learning rate and the number of neurons in the hidden layer. The best results have been obtained for ϵ equal to 0.2 and a number of neurons in the hidden layer equal to 4. These simulation parameters and the convergence time. The back-propagation algorithm, has been used to identify the model.

To determine the generalization capacity of the developed neural model, we apply to the model a new input and we test its response to this input which is not found in the training set. This phase is ensured by three bases of generalization over a horizon of 150 examples each one: the first one is relative to the first zone of operation, the second is relative to the second zone of operation and the third is relative to the third area of operation.

The architecture of the neural network generating the database is illustrated in Figure (5).

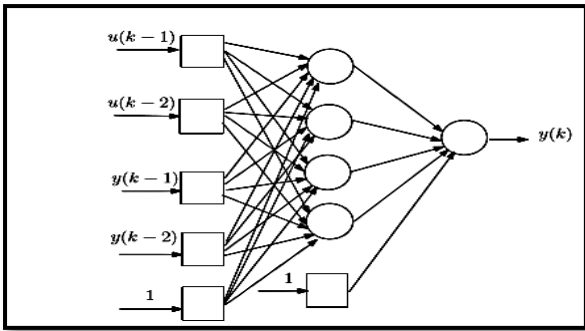


Fig 5: The architecture of the neural network generating the database.

The evolution of the output model with that of the system at the generalization of the three bases of examples is shown in the figure below:

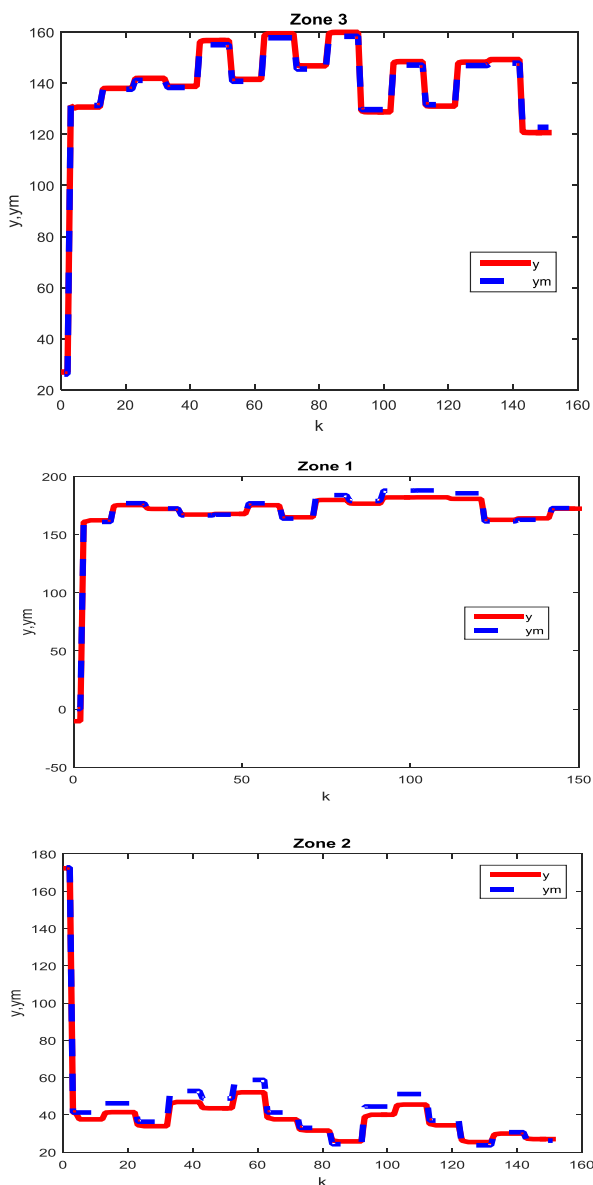


Fig 6: Evolution of the output of the global system with that the global models in the three bases of examples at the generalization phase

For the first area, the generalization is not perfect. However, for the other two areas the generalization is good. This shows that the developed model is a good model only in certain areas of operation.

5.1.2 Multi-Network Neural Model:

In this part, the learning of the multi network model is done; ie, calculate the validity degree μ_i of each examples then adjust the connection weights of all networks at the same time using an iterative process (the learning) until an error value is reached.

The input vector covers three operating zones, each one have 200 examples.

The fuzzy rules take the form:

- **Locals Models**

Considered that the input sequence $u(k)$ is divided into three zone of operations; each one has been set in a horizon of 200 points:

Zones 1: $u(k)$ varies around the value 180 (of the order of 10%).

Zones 2: $u(k)$ varies around the value 40 (of the order of 10%).

Zones 3: $u(k)$ varies around the value 120 (of the order of 10%).

From these learning subsets the definition of three subsystems is provided. Each subsystem was studied in its area of operation. Each learning subset was used for the synthesis of a neural model which is valid locally in the area covering its learning basis.

These local models will be tested using a generalization phase.

A good set of local models will give a good fused model.

To develop the three local models, it was assumed that:

- The output is defined by the model described by (9).
- The number of hidden neurons equal to 4 and a step iteration ϵ equal to 0.2. These values are determined with tacking into account a good approximation and a good generalization.
- The learning algorithm used is the back-propagation algorithm.

The results of generalization of the local models are given by this figure:

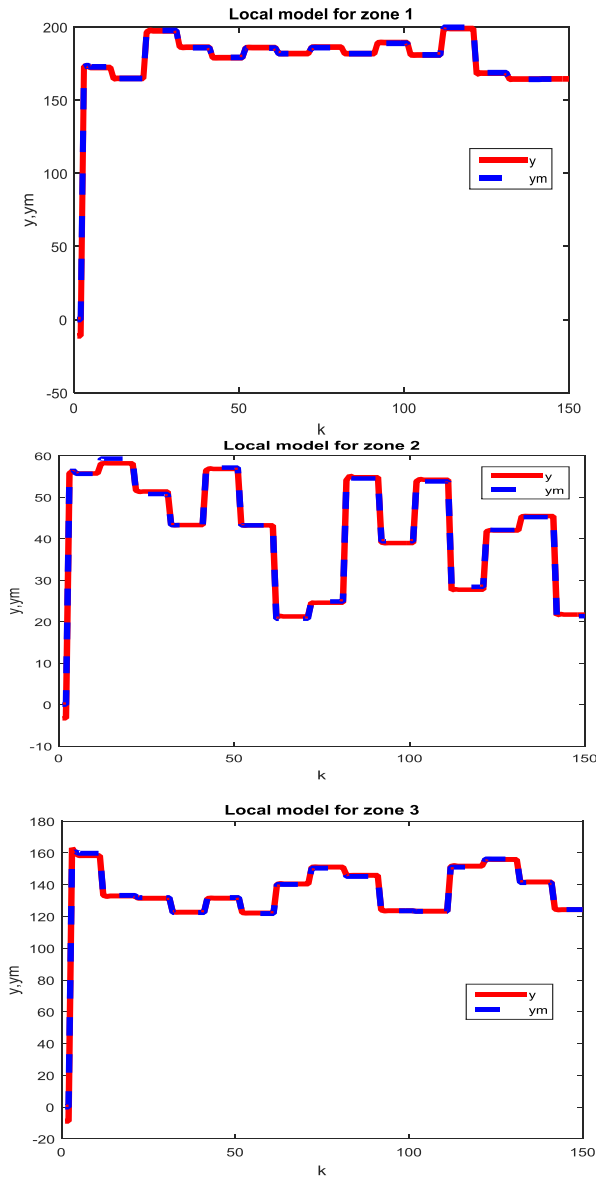


Fig 7: Evolution of the three local models at the generalization phase

Note that the local model obtained have good generalization ability.

▪ **Multi-Network Neural Model**

The conditions of implementation are:

The input of neural networks $u(k)$ covers all areas of operation command signals of the subsystems. The basis of examples is defined on a working horizon of 600 examples.

The membership function adopted in this example is the Gaussian function.

- The fuzzy rules have the following forms :

If $u(k)$ is A_1 then $y_1(k + 1) = MN_1[y(k-1), u(k-1)]$

If $u(k)$ is A_2 then $y_2(k + 1) = MN_2[y(k-1), u(k-1)]$

If $u(k)$ is A_3 then $y_3(k + 1) = MN_3[y(k-1), u(k-1)]$

- The fuzzy subsets A_i are characterized by Gaussians centered respectively at 180, 40 and 120.

- The output of the fused model interpolates all the outputs of local models, it takes the following form:

$$y(k + 1) = \frac{\sum_{i=1}^3 \mu_i y_i(k+1)}{\sum_i \mu_i} \quad (10)$$

With:

$$\mu_i = \frac{e^{-(u-c_i)^2}}{2\sigma^2} \quad (11)$$

σ is equal to 30 and we assume that $\sum_{i=1}^3 \mu_i = 1$.

This type of modeling provided good results.

Comparison between the Global Model and the fused Model.

The fused model gives similar results to those given by the global model.

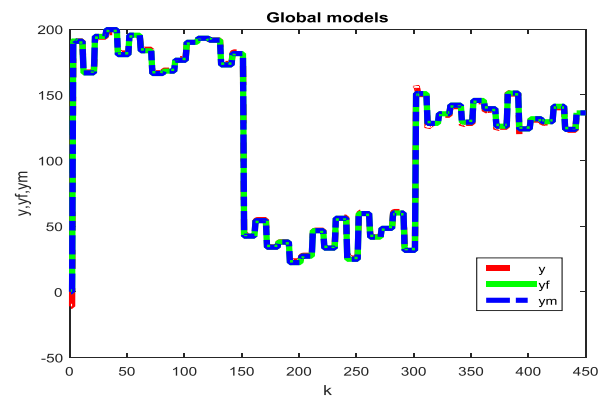


Fig 8: Evolution of the output of the system with that of the fused model and with examples the global model

The fused model was obtained by learning each local models and fusing them so it takes three neural networks, while the global model was obtained by learning a single model so it requires a single network neurons.

The generalization of the global model is acceptable but not really perfect. For each of the local models, generalization is perfect in all areas of operation. So, the decomposition of the field operating areas results local models which have a simple structure, a time of learning reduced and a good performance.

The following table shows the error, the test error, the learning time and the hidden neurons number of local models and of the global model at learning phase while assuming that the number of iterations was set at 1500000 iterations.

Table 1. Results of different models

Models	Learning error	Test error	Learning time (s)	Hidden neurons number
Global Model	0.0232	0.0248	1631.781532	6
First local model	0.0006	0.00065	304.711083	4
Second local model	0.0023	0.0041	296.605346	4
Third local mode	0.0005	0.00058	332.930301	4

Note that the amount of learning error of all local models which represents the learning time of the fused model is lower compared to the global model.

Therefore, the multi-network neural modeling is better than the mono-network neural modeling because win in learning time and provide a model that perfectly reproduces the dynamics of the system despite all approximations used is the main objective.

6. CONCLUSION

This work develops a forecasting mechanism based on artificial neural networks techniques to manage the demand forecasting. At the first step the ability of neural networks to model varied dynamic structures is shown. In return, limit its application if taking into account its many parameters and difficult to solve as the number of neurons in hidden layers, the learning rate and the number of iterations.

In a second step, demonstrate the effectiveness of the approach of multi-model neural network that includes neural networks with fuzzy approach while overcoming the major drawbacks of neural networks compared to the mono-network modeling approach.

7. REFERENCES

- [1] Abid, S., Chtourou, M., & Djemel, M. (2014). Statistical and incremental methods for neural models selection. *International Journal of Artificial Intelligence and Soft Computing*, 4(1), 41.
- [2] Aburto, L., & Weber, R. (2007). Improved supply chain management based on hybrid demand forecasts. *Applied Soft Computing*, 7(1), 136–144.
- [3] agan, M.T. and Menhaj, M.B. (1994) Training Feed forward Techniques with the Marquardt Algorithm. *IEEE Transactions on Neural Networks*, 5, 989-993.
- [4] Bousqaoui, H., Slimani, I., Achchab.S. (2018). Improving Coordination in Supply Chain Using Artificial Neural Networks and Multi-agent Approach. *Proceedings of the 6th International Conference on Engineering Optimization*, pp. 1353–1359, 2019.
- [5] Chawla, A., Singh, A., Lamba, A., Gangwani, N., & Soni, U. (2018). Demand Forecasting Using Artificial Neural Networks—A Case Study of American Retail Corporation. *Applications of Artificial Intelligence Techniques in Engineering*, 79–89. doi:10.1007/978-981-13-1822-1_8.
- [6] Chen, F., Ryan, J.K., and Simchi-Levi, D. (2000). The impact of exponential smoothing forecasts on the bullwhip effect. *Naval Research Logistics*, 47(4), 269–286.
- [7] Chen, F., Drezner, Z., Ryan, J.K., and Simchi-Levi, D. (2000). Quantifying the Bullwhip Effect in a Simple Supply Chain: The Impact of Forecasting, Lead Times, and Information. *Management Science*, 46(3), 436–443.
- [8] Chopra, S., Meindl, P. (2001). *Supply Chain Management: Strategy, Planning and Operation*. NJ: Prentice Hall, Inc., 2001, 457 pages.
- [9] Efendigil, T., Önüt, S., & Kahraman, C. (2009). A decision support system for demand forecasting with artificial neural networks and neuro-fuzzy models: A comparative analysis. *Expert Systems with Applications*, 36(3), 6697–6707.
- [10] Frasconi, P., Gori, M., Maggini, M., & Soda, G. (1996). Representation of finite state automata in Recurrent Radial Basis Function networks. *Machine Learning*, 23(1), 5–32.
- [11] Garetti, M., & Taisch, M. (1999). Neural networks in production planning and control. *Production Planning & Control*, 10(4), 324–339.
- [12] Hornik, K. (1991). Approximation capabilities of multilayer feedforward networks. *Neural Networks*, 4(2), 251–257.
- [13] K. Gasso. « Identification des systemes dynamiques non-lineaires : approche multi-modele ». These de doctorat de l'INPL, 2000.
- [14] Lee, H. L, Padmanabhan, V., and Whang, S. (1997). The Bullwhip Effect in Supply Chains. *Sloan Management Review*, 38, 93-102.
- [15] Lu, Lauren Xiaoyan and Swaminathan, Jayashankar M., *Supply Chain Management* (2015). *International Encyclopedia of Social and Behavioral Sciences*, 2nd edition, edited by James Wright. Elsevier, March 2015.
- [16] Leung, H. C. (n.d.). Neural networks in supply chain management. *Proceedings for Operating Research and the Management Sciences*.
- [17] Narendra, K. S., & Parthasarathy, K. (1991). Gradient methods for the optimization of dynamical systems containing neural networks. *IEEE Transactions on Neural Networks*, 2(2), 252–262.
- [18] Rivals I. Modélisation et commande de processus par réseaux de neurones; application au pilotage d'un véhicule autonome, Thèse de Doctorat de l'Université Paris 6 (1995).
- [19] S, B., J, B., & Hassani, H, E. (2013). Causal Method and Time Series Forecasting model based on Artificial Neural Network. *International Journal of Computer Applications*, 75(7), 37–42.
- [20] Solla S.A., « Learning and Generalization in Layered Neural Networks: the Contiguity Problem» In L. Personnas and G. Drefus (eds), *Neural Networks: from Models and Applications*. Paris. I.D.S.E.T, 1989.
- [21] Xinghuo Yu, Efe, M. O., & Kaynak, O. (2002). A general backpropagation algorithm for feedforward neural networks learning. *IEEE Transactions on Neural Networks*, 13(1), 251–254.
- [22] Zhao, X., Xie, J., & Leung, J. (2002). The impact of forecasting model selection on the value of information sharing in a supply chain. *European Journal of Operational Research*, 142(2), 321–344.
- [23] Zhao, Y. (1996). On-line neural network learning algorithm with exponential convergence rate. *Electronics Letters*, 32(15), 1381