

Sales Prediction through Neural Networks for a Small Dataset

Rosa María Cantón Croda, Damián Emilio Gibaja Romero, Santiago Omar Caballero Morales *

Deanship of Graduate Studies in Engineering and Businesses, Universidad Popular Autónoma del Estado de Puebla, Puebla (Mexico)

Received 5 November 2017 | Accepted 17 February 2018 | Published 13 April 2018



ABSTRACT

Sales forecasting allows firms to plan their production outputs, which contributes to optimizing firms' inventory management via a cost reduction. However, not all firms have the same capacity to store all the necessary information through time. So, time-series with a short length are common within industries, and problems arise due to small time series does not fully capture sales' behavior. In this paper, we show the applicability of neural networks in a case where a company reports a short time-series given the changes in its warehouse structure. Given the neural networks independence from statistical assumptions, we use a multilayer-perceptron to get the sales forecasting of this enterprise. We find that learning rates variations do not significantly increase the computing time, and the validation fails with an error minor to five percent.

KEYWORDS

Artificial Intelligence, Artificial Neural Networks, Forecasting.

DOI: 10.9781/ijimai.2018.04.003

I. INTRODUCTION

SALES forecasting has positive implications within enterprises. It can improve the planning of production processes as well as inventory management practices [1]. Specifically, the fast growth of enterprises and their incursion into new markets make inventory optimization necessary. Even more, because sales have a close relationship with the clients' demands which depend on factors that enterprises cannot control, knowing the customers' behavior may reduce the enterprises' operational costs [2].

It is important to recall that predicting future demand is a complex problem since it relies on determining a concrete definition of what demand is [3]. Note that order times, preferences and previous sales affect the current demand that a company must satisfy or supply. Thus, forecasting accuracy depends on the information gathered by the enterprise [4]. In this sense, the literature recognizes that previous sales are information that companies register to be used as starting points. Even more, sales do not only summarize the sold quantity, but they also include information that reflects changes in consumer's behavior concerning the selling prices. Hence, sales represent a good proxy to understand the demand trends that a company faces [5].

Regarding the quantity of data required to estimate demand trends, there is an active discussion in the literature focused on determining the optimal size of a dataset for demand forecasting [6]. In general, research studies on this topic agree that dataset size depends on the prediction model [7]. For example, ARIMA MODELS need $p + q + P + Q + d + mD$ observations, where p is the number of time

lags, q represents the model's order, m is the total number of periods at each season, and d is the differencing degree. Also, this model pretends to capture seasonal effects, a reason why it includes terms related to autoregressive P , moving Q and differencing averages D . On the other hand, literature has suggested the use of artificial neural networks (ANNs) as robust techniques to analyze small datasets due to their performance being independent of the total number of input data [8]. Recently, it is common to find applications of ANNs to solve forecasting problems in medicine, where small datasets prevail since patients and diseases' evolution change regularly [9]. This last factor has a direct impact on patients' medical visits, which tend to no show at them because they "feel better". A prediction of such behavior is useful to reduce costs and improve medical services efficiency [10].

Given the favorable applications of ANNs to analyze non-linear behavior in datasets [1, 9] our work was focused on the application of this technique to predict the sales of a chemical company located in Mexico. For this case, the dataset consisted of 12 monthly sales data as the company reports its sales on a monthly basis. 70.0% of this data was considered for training and 30.0% for initial validation of the ANN which was designed for this work.

This ANN consisted of a multi-layer perceptron with a sigmoid activation function. Implementation of the ANN was performed using programming with the software R v.3.2.1. We find that a small dataset does not significantly increase the number of iteration, even when we consider the maximum possible learning rate. However, as it is expected by the classical time series literature, the prediction presents a higher error, which we demonstrate in the validation phase.

The paper is organized as follows. Section II presents a brief literature review of the sales forecasting literature, and we discuss the viability and applications of ANNs. Then, Section III describes the designing of the ANN with a general overview of how these techniques work. Section IV presents the experiments associated

* Corresponding author.

E-mail addresses: rosamaria.canton@upaep.mx (R. M. Cantón Croda), damianemilio.gibaja@upaep.mx (D. E. Gibaja Romero), santiagoomar.caballero@upaep.mx (S. O. Caballero Morales).

with the ANN, highlighting interesting insights about the training, validation and prediction processes. To motivate the use of ANN, even in small databases, we compare the ANN performance with the one of using the average moving technique. Finally, Section V presents our conclusions.

II. LITERATURE REVIEW

Given that time series are central in determining predictions, there is a considerable literature that analyzes their characteristics and the methodologies to find the desired results [11]. This paper does not pretend to be an exhaustive exposition of the time-series forecasting literature. However, we present some of the most significant contributions in the application of time-series for demand forecasting.

A. Sales Forecasting and Industry

Industries, in general, concern about demand forecasting since their benefits depend on consumer's behavior. Hence, forecasting studies contribute to diminishing uncertainty about future sales, which implies a better production planning and inventory cost reduction [12].

Although future sales have a high degree of uncertainty, there exist different approaches to approximate the consumer's behavior in future periods. The literature on this problem considers different approaches and methodologies, which rely on the analysis of time series that summarizes how the demand evolves through time together with its related variables [3].

It is important to recall that demand and sales have a close relationship since the last summarizes the consumers' behavior. Thus, sales work as a good indicator of consumers' preferences even though they are subjective [13]. For example, demand forecasting uses variables such as sales, expenses, and benefits [14]. However, there exist other studies where demand is predicted or forecasted through different variables that depend on the market where companies and products stand. For example, Rodriguez et al. [18] built an index for water demand that summarized water consumption, land use, average incomes, and population. In the case of electricity demand, it is common to consider income, prices of related fuels, community, and technology [15]. All the previous applications used time-series to determine future events.

B. Classical Time-Series Forecasting Literature

Typically, Time-Series Forecasting relies on the application of two methods: Holt-Winters and Box-Jenkins [16]. In the case of time-series with linear behavior, the previous techniques generate a near-exact approximation of future demands. Thus, we can find applications of these methods in forecasting exchange rate [17], demand flowmeters [14], aircraft failure rates [18], general prices [19], and transport demand [20] among others.

It is important to mention that not all time-series present a linear behavior. In a non-linear situation, the Holt-Winters and Box Jenkins methods fail to provide a good forecasting of future events. Specifically, linear models cannot capture and analyze amplitude dependence, volatility clustering, and asymmetry, which prevail in finance time-series [21]. In these cases, it is appealing to assume different states [22], or considering situations with the minimum number of statistical assumptions [23].

The analysis of non-linear time-series requires the construction of models suitable for each phenomenon, which pretend to capture stochastic processes. Hence, Campbell, Lo and McKinley [24] showed that it is possible to classify most of the non-linear time series according to their deviations from their mean, the non-linear moving average model [25], or their variance (for example, the ARCH model and their variations like GARCH and EGARCH [26].

C. Neural Networks and Sales Forecasting

Classical non-linear time series models have a significant dependence on statistical assumptions, which diminish their capacity to predict a behavior that was not previously registered. To deal with such problem, Artificial Neural Networks (ANNs) appear as a suitable alternative. These techniques try to mimic neural cells behavior, which makes them non-linear models by nature [27]. Because ANNs are data-driven, they have a self-adaptive feature. This adaptation capacity makes them suitable to analyze problems where data is not complete, or when its documentation was not appropriately gathered (i.e., data with noise [28]).

Given that ANNs allow the approximation of almost all non-linear continuous function, they are suitable for sales forecasting [29]. However, in the classical literature, there is not a unique approach to sales forecasting because ANNs may have different structures according to the problem. For example, electricity demand forecasting considers a hybrid structure since data may present seasonal effects. An et al. [15] propose a two-step architecture, where seasonal effects are reduced in the first step through an empirical mode decomposition; and later, a multi-output feed-forward ANN is used to get the demand forecast for the following H periods given the demand of the previous T periods. Unlike electricity demand, water consumption follows an ANN with stochastic connections since it noisy data prevail in such situations [14]. In both cases, ANNs' architecture requires a long time-series.

Graphically, an ANN is a graph with neurons (nodes) and their synapses (edges). These models may learn and identify consumers' behavior patterns according to the input data and the architecture of the ANN. The structure of an ANN summarizes the interconnection between neurons, which are organized in layers [30]. The most uncomplicated architecture considers the input layer, the output layer, and the hidden layer, whose objective is the identification and communication of pattern to get the demand forecasting. Fig. 1 shows an example of an ANN with three layers (input, hidden and output layer).

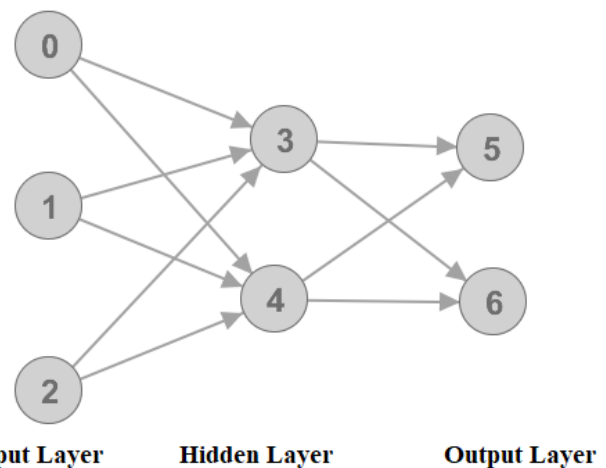


Fig. 1. ANN with only one hidden layer.

The hidden layer serves to adjust the importance of the input data in the determination of future behavior. Therefore, it is possible to add as many hidden layers as required [27]. Although no theoretical tool indicates the optimal number of hidden layers, experiments demonstrate that more than one hidden layer does not significantly improve the prediction's quality. Also, there is a positive relationship between the number of hidden layers and the computing time. Thus, one hidden layer is enough to get a proper prediction, mainly when the database is small [30].

III. DESIGNING A NEURAL NETWORK FOR FORECASTING

We analyze the case of a chemical company located in Mexico. This firm owns nine central warehouses across the Mexican territory which are used to fulfill the requisitions of its clients. Because these distribution centers may generate inventory costs to the detriment of company's benefits, the company is interested if sales justify the construction of additional warehouses. This motivates the prediction of future sales through ANNs.

A. Data Selection

The chemical company builds central warehouses at different locations, which imply the absence of a long time-series that includes all central warehouses sales. Thus, we considered a time series from 2015 to 2016 that consists of the sales of the current distribution centers.

B. Data Collection

The chemical enterprise provided a database with the sales that each of its central warehouses must comply. This dataset included information related to a product, client and transportation costs. However, due to confidentiality restrictions, we considered only sales information of twelve months from 2015 to 2016 for the time series that shows the company's sales for one year.

C. Choosing the Neural Network

Forecasting literature that applies ANNs agrees in the use of the multi-layer perceptron (MLP). This ANN is characterized by a feedforward architecture where unidirectional weighted edges connect neurons. Since we have a total of 12 data, we considered an MLP with three neurons in its hidden layer. Thus, the MLP mathematical expression is

$$y_t = \alpha_0 + \sum_{j=1}^3 \alpha_j g(\beta_{0j} + \sum_{i=1}^{12} \beta_{ij} y_{t-i}) + \epsilon_t, \forall t,$$

where α_j and β_{ij} are the connection weights, for all $j = 0, 1, 2, 3$ and $i = 0, 1, 2, \dots, 12$. Also, g represents the activation function that defines the output of a neuron. Given the size of the dataset, we considered a hidden layer with three neurons.

There are different activation functions, and their use depends on the problem's features. Such functions may produce an activation within a discrete range, like a set $\{0,1\}$ that represents "yes" or "no," or a continuous range like \mathbb{R} . In the case of forecasting, Thomonopoulos [2] suggested that continuous ranges are more desirable because they do not restrict neurons behavior to a finite number of results. Also, the sigmoid function was suggested as beneficial for forecasting because it allows any real number as an input, and does not present convergence problems since it is bounded between -1.0 and 1.0. Thus, in this paper, we consider the sigmoid function as the activation function

$$\text{sig}(t) = \frac{1}{1+e^{-t}}$$

For our sales prediction problem, we compute y_t which is the company's sales during period t . To generate the prediction, the perceptron that we consider uses the sales from the three previous periods. In other words, we "learn" from months one, two and three to "predict" the fourth month. Then, the ANN computes the weights that minimize the error, and such values are saved. In the next iteration, the ANN "learns" from month two, three and four to "predict" the fifth-month sale. So, we proceed in the following way

ANN_0: Input (1,2,3) → output (4)

ANN_1: Input (2,3,4) → output (5)

ANN_2: Input (3,4,5) → output (6)

ANN_3: Input (4,5,6) → output (7)

ANN_4: Input (5,6,7) → output (8)

ANN_5: Input (6,7,8) → output (9)

We recall that the previous architecture produces a single output, and weights are actualized every time a fourth value is predicted using the three previous weights. That is to say, we choose a recurrent architecture for the neural network [30].

IV. NEURAL NETWORK IMPLEMENTATION

There is much mathematical software that is capable of programming and implementing ANNs. We choose the software R to implement the ANN given the possibility to change the ANN parameters.

Now, the construction of an ANN requires the weights' computation, and their validation as well. To do this, we divide the database into two sets: one for training (or learning) and one for validation. By following the guidelines of [31, 32], we determined the size of both sets: the training set was considered with nine data, and the validation set was considered with three data. This corresponds to the 75.0% and 25.0% of the entire database respectively.

A. Training

The self-adaptation property allows ANNs to adjust future behavior given previous behavior. Thus, in the training phase, the ANN learned the time series behavior through computing how high the connections between neurons must be. Below, we present the code in R language.

DEFINING THE NEURAL NETWORK AS A FUNCTION

```
neuron = function (normaldemands, learn, lA) {
  ## VARIABLES INITIALIZATION
  weights = c (runif (length (normaldemands), min = 0, max = 1))
  weights0 = weights; weightsE = 0; sigmoid = 0;
  error1 = 2; itera = 1; ErrorI = 0
  while (error1 >= 0.001) {
    x = 0; weightsE = weights
    for (i in 1: length (normaldemands)) {
      x = x + c (weights [i] * normaldemands [i])
    } ## end for

    sigmoid = (1 / (1 + exp (-x))) ## Sigmoid function
    error1 = (learn - sigmoid) * sigmoid * (1-sigmoid)

    ## NEW WEIGHTS COMPUTATION
    for (i in 1: length (normaldemands)) {
      weights [i] = weights [i] + lA * error1 *
        normaldemands[i]
    } ## end for
    ErrorI [itera] = error1;
    iterate = iterate + 1;
  } ## end while

  output = list (weights0 = weights0, weightsE =
    weightsE, sigmoid = sigmoid, error = error1, ErrorI =
    ErrorI, learn = learn)
  return (output)
} ## end function
```

About the previous code, it is important to remark the following:

- First, we know that the sigmoid function range is the interval $[0,1]$. Thus, this function outputs values between zero and one. Since the time series registers the total sales, we need to normalize the values of our database. So, let x_t be the sales of period t , and y_t the normalized sales for period t . We use the following transformation

$$x_t = \frac{1}{z_{max}} z_t,$$

where $z_{max} = \max\{z_1, z_2, \dots, z_{12}\}$ and the values of the time series are: 2975760.227, 3451339.887, 3567417.723, 3560103.948, 3933955.267, 3573328.611, 4032969.915, 3788327.879, 4350361.992, 4597373.737, 3567670.195, 4047731.206 and $z_{max} = 4597373.737$.

- Second, the training part is a supervised learning process. This means that the ANN adjusts the weights between the neurons to learn the behavior or pattern of the time series. Then, the ANN computes the weights that minimize the following error

$$error = (y_t - x_t)^2$$

In this exercise, we tolerate an error less than 0.001.

- Third, programming the ANN in R allows determining the learning rate. This indicates how fast the ANN adapts to new behavior presented in the time series. Given that we have a small dataset, we compared different learning rates to assess the ANN's performance.

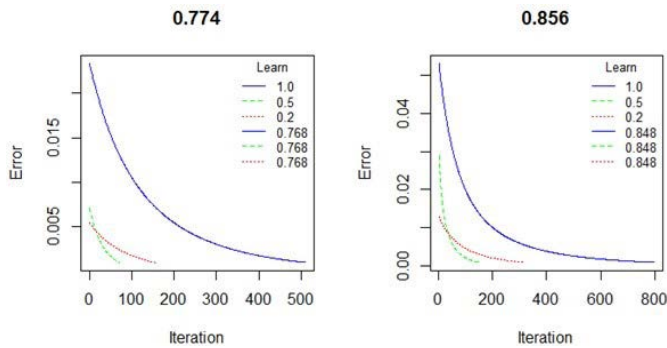


Fig. 2. Comparison between learning rates and iterations.

Fig. 2 shows the relationship between the error and the number of iterations when we consider the following learning rates: 0.2, 0.5 and 1.0. In this situation, the ANN tries to learn the real values 0.774 and 0.856. Note that the highest learning rate implies a complete understanding of the time series behavior, but it requires the most significant number of iterations. However, despite the database's size, the number of iterations is not unusual in the literature [27].

In general, the previous behavior is replicated by the ANN when we change the real values. For example, Fig. 3 shows the relationship between the three learning rates when the ANN learns the largest possible sale, which is equal to 1.0. Although the number of iteration increases while the ANN tries to reduce the error, this number does not excessively increase.

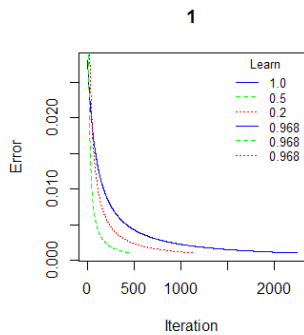


Fig. 3. Learning the largest possible sale.

Fig. 2 and 3 illustrate how fast the ANN reduces the approximation error when learning rate is switched. In all cases, the small dataset

does not induce a time convergence problem, the number of iterations is relatively small. Also, as the literature suggests, when we impose a learning rate equal to 1.0, the number of iterations increases. It is interesting to note that a learning rate equal to 0.5 is faster than the learning rate equal to 0.2. However, the computation time is not an additional restriction for our work, R spends 27.82 seconds to compute all weights.

Given the limited information, we consider appropriate to choose a learning rate equal to 0.2. This assumption comes from the literature because an ANN based on small dataset requires slackness in their learning process [8].

B. Validation

For the validation process, we use the nine weights produced by the algorithm during the previous step. Remember that the fifth iteration summarizes the knowledge required by the ANN to forecast. Then, we have that

- Input (7,8,9) → ANN_5 → Output (10)
- Input (8,9,10) → ANN_5 → Output (11)
- Input (9,10,11) → ANN_5 → Output (12)

Table I presents the weight for each value in the training set and the number of iterations that R took to compute them.

TABLE I. WEIGHTS FROM THE SUPERVISED PHASE

MONTH	SALES (z_t)	WEIGHT	ITERATIONS
1	2975760.227	0.9544753	250
2	3451339.887	0.9833160	294
3	3567417.723	0.3501016	230
4	3560103.948	0.4076781	390
5	3933955.267	0.4222303	280
6	3573328.611	1.2014761	627
7	4032969.915	1.0023636	1090
8	3788327.879	0.3538968	172
9	4350361.992	0.4986251	274

To proceed with the validation phase, we use the sales for months 10, 11 and 12. By the ANN architecture, we follow an iterative process where the weights of the three previous periods were used. In this case, the process was unsupervised, i.e., we let the ANN adjust the weights in a free manner to get the last three known values. Table II illustrates the weights that we use in the validation process.

TABLE II. VALIDATION PROCESS

MONTH	WEIGHTS		
10	1.6621	0.9737	1.2102
11	0.1779	0.7778	0.3281
12	0.4470	1.0622	0.5489

During the validation process, we find that the ANN does not predict the time series with a one hundred percent of accuracy. However, the error is not higher than the 10%, as Table III shows with the comparison between the real values and the ones produced by the ANN.

TABLE III. VALIDATION PROCESS

REAL SALES	TEST SALES	ERROR
1.00	.967	0.043
.776	.770	0.01
.880	.871	0.0103

C. Predictions

Although the ANN does not entirely replicate the time series behavior, this is due to the database size. Consequently, we use the

weights in Table II to get the predictions for the next three months. Then, the ANN architecture assumes that the first projection, sales of month 13, is real to compute the prediction of month 14. We continue iteratively to get the prediction for month 15. Then, the algorithm does the following computations

- Input (10,11,12) → ANN_5 → Output (13)
- Input (11,12, Output (13)) → ANN_5 → Output (14)
- Input (12, Output (13), Output (14)) → ANN_5 → Output (15)

Following the previous process, Table IV shows the weights that the ANN requires to compute the sales prediction for the next three months.

TABLE IV. PREDICTION WEIGHTS

MONTH	WEIGHTS		
13	0.4470030	1.0622665	0.5489295
14	1.0914885	0.5261332	0.2219956
15	0.5159739	0.9045625	0.5206516

Using the information in Table IV, we proceed to compute the sales that the chemical company may face in the first three months of the next year. In Table V we present these results in the normalized form and their corresponding transformation.

TABLE V. MAIN RESULTS

MONTH	NORMALIZED SALES (x_t)	SALES (z_t)
13	0.8525044	3919281.34
14	0.8448968	3884306.36
15	0.8375715	3850629.22

Finally, Fig. 4 shows the relation between the three phases: training, validation, and prediction. In this graph, we compare the prediction values with the real values, and we can see the trend for future sales. As we mentioned before, during the supervised phase, the ANN fully captures the previous behavior of the time series, while in the validation phase it presents problems to predict with a 100% of accuracy the other values. However, the distance between both lines is not significant (less than 5%).

Also, it is interesting to note that the following next three months present a decreasing behavior maybe which does not match with the behavior in the corresponding months in the database. This last point may be related to the fact that we consider the 13th month as real information to get the months 14 and 15.

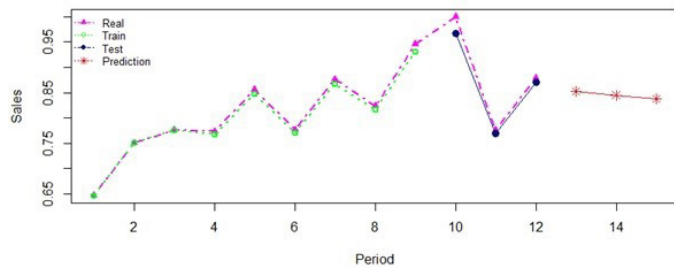


Fig. 4. Comparison between real and predicted time series with normalized data.

D. Comparison with Moving Average

As we mentioned before, moving average models (ARMA, ARIMA, ...) are commonly used to analyze non-linear series [10]. In this section, we compare the sales forecasting that results from the application of a simple moving average, with the ANN's results. Since we have a small database, we forecast a month by using the following simple moving average (SMA) model

$$y_t = \frac{x_{t-1} + x_{t-2} + x_{t-3}}{3} \text{ for all } t \geq 4.$$

In other words, this model approximates the sales of month t using the mean of the three previous month sales.

Table VI shows the sales forecasting when we use the SMA technique and the approximation error for each month. Fig. 5 illustrates the original time series with the moving average results.

TABLE VI. MOVING AVERAGE RESULTS

MONTH	SALES (z_t)	MOVING AVERAGE FORECAST (y_t)	ERROR
1	2975760.227		
2	3451339.887		
3	3567417.723		
4	3560103.948	3331505.95	0.064211
5	3933955.267	3526287.19	0.103628
6	3573328.611	3687158.98	-0.031855
7	4032969.915	3689129.28	0.085257
8	3788327.879	3846751.26	-0.015421
9	4350361.992	3798208.8	0.126921
10		4057219.93	
11		4069344.94	
12		4350361.99	

Considering the sales of the validation set (months 7, 8 and 9), we note that the SMA prediction reports an error greater than 5% percent for months 7 and 9 (see Table VI). In comparison, for the same set of months, the ANN provides predictions with an error less than 5%, see Table III. Since the sales of the validation are known, we can conclude that the ANN's predictions are more accurate than the ones from the SMA.

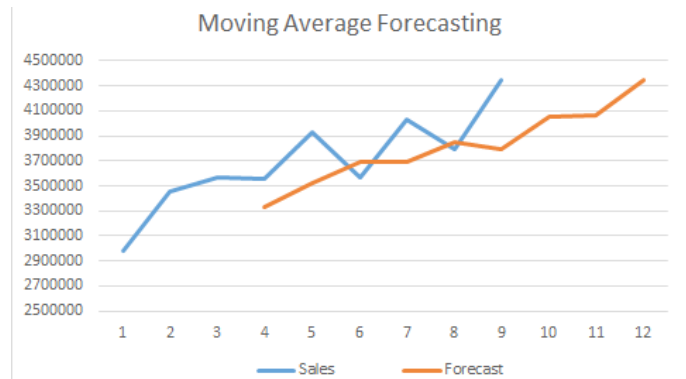


Fig. 5. Original sales (blue) versus average moving predictions (orange).

V. CONCLUSION

The present paper shows the flexibility of neural networks to deal with sales forecasting in the presence of a small database. As the literature suggests, the ANN does not fully capture the time series behavior as we show during the validation phase. However, it is worth noting that the prediction error is not greater than 5%, which is accurate in sales forecasting instances [31].

Even more, in our case of study, the small dataset does not increase the time convergence even when we impose a high learning rate. Also, the ANN has a better performance than the SMA concerning error.

Specifically, for months 7 and 9, whose sales are known, the SMA reports an error greater than 5% while the ANN reports an error less than 5%. The previous facts support the use of this technique in a situation when time series are small, or the database does not present all the possible information.

Therefore, ANN may be useful in a situation where time series present a non-linear behavior and do not show an explicit statistic behavior, as it is the situation in time series with a short length. However, in the present experiment, we observe the significant dependence that new predictions have when the first prediction is considered real information. In future works, we address this problem from theoretical and empirical perspectives.

REFERENCES

- [1] R. Carbonneau, K. Laframboise, and R. Vahidov, "Application of machine learning techniques for supply chain demand forecasting." *European Journal of Operational Research*, vol. 184, n° 3, pp. 1140-1154, 2008.
- [2] N. Thomopoulos, "Demand Forecasting for Inventory Control. In: Demand Forecasting for Inventory Control." Springer, Cham, 2015.
- [3] J. R. M Hosking, "Demand Forecasting Problems in Production Planning." *Planning Production and Inventories in the Extended Enterprise. International Series in Operations Research & Management Science*, vol. 151, 2011.
- [4] P. Yenradeea, A. Pinnoi, and A. Charoenthavornying, "Demand forecasting and production planning for highly seasonal demand situations: a case study of a pressure container factory," *Science Asia*, vol. 27, pp. 271-278, 2001.
- [5] R. Skyrius, G. Kazakevičienė, and V. Bujauskas, "From management information systems to business intelligence: The development of management information needs," *International Journal of Interactive Multimedia and Artificial Intelligence*, vol. 2, n°3, 2013.
- [6] R. J. Hyndman, and A. V. Kostenko, "Minimum sample size requirements for seasonal forecasting models," *Foresight*, vol. 6, pp. 12-15, 2007.
- [7] P. B. Prakash, V. Ramya, and M. Yugandhar, "A Statistical GA Based Demand Forecasting Model for Automotive Batteries Manufacturing Company," *i-Manager's Journal on Mechanical Engineering*, vol.6, n° 3, pp-18, 2016.
- [8] S. Ingrassia, and I. Morlini, "Neural network modeling for small datasets," *Technometrics*, vol.47, n° 3, pp. 297-311, 2005.
- [9] A. Pasini, "Artificial neural networks for small dataset analysis." *Journal of thoracic disease*, vol. 7, n° 5, pp- 953, 2015.
- [10] C. Elvira, A. Ochoa, J. C. González and F. Mochón, "Machine-Learning-Basec no Show Prediction in Outpatient Visits," *International Journal of Interactive Multimedia and Artificial Intelligence*, vol. 4, n°7, 29-34, 2017.
- [11] C. Chatfield, "Time-series forecasting." CRC Press, 2000.
- [12] L. Šečkute and A. Pabedinskaite, "Application of forecasting methods in business," *Journal of Business Economics and Management*, vol. 4, n° 2, pp. 144-157, 2003.
- [13] L. Kurzak, "Importance of Forecasting in Enterprise Management." *Advanced Logistic systems*, vol. 6, n° 1, pp.173-182, 2012.
- [14] H. Rodriguez, V. Puig, J. J. Flores, and R. Lopez, "Combined holt-winters and GA trained ANN approach for sensor validation and reconstruction: Application to water demand flowmeters." 2016 3rd Conference on Control and Fault-Tolerant Systems (SysTol), on, pp. 202-207, 2016.
- [15] N. An, W. Zhao, J. Wang, D. Shang, and E. Zhao, "Using multi-output feedforward neural network with empirical mode decomposition based signal filtering for electricity demand forecasting." *Energy*, vol. 49, pp. 279-288, 2013.
- [16] L. Escobar, J. Valdes, and S. Zapata, "Redes Neuronales Artificiales en predicción de Series de Tiempo. Una aplicación a la Industria," *Working Paper Series Universidad de Palermo*, 2009.
- [17] E. Valakevicius, and M. Brazenas, "Application of the Seasonal Holt-Winters Model to Study Exchange Rate Volatility." *Engineering Economics*, vol. 26, n° 4, 384-390, 2015.
- [18] Y. M. Yang, H. Yu, and Z. Sun, "Aircraft failure rate forecasting method based on Holt-Winters seasonal model," 2017 IEEE 2nd International Conference on Cloud Computing and Big Data Analysis (ICCCBDA), pp. 520-524, 2017.
- [19] A. Singh, and G. C. Mishra, "Application of Box-Jenkins method and Artificial Neural Network procedure for time series forecasting of prices," *Statistics in Transition new series*, vol. 1, n° 16, pp. 83-96, 2015.
- [20] S. Anvari, S. Tuna, M. Canci, and M. Turkey, "Automated Box-Jenkins forecasting tool with an application for passenger demand in urban rail systems," *Journal of Advanced Transportation*, vol.50, n°1, pp. 25-49, 2016.
- [21] C. M. Kuan, "Lecture on the Markov switching model. Institute of Economics Academia Sinica," pp. 1-30, 2002.
- [22] N. Davies, and J. Petruccielli, "Detecting Non-Linearity in Time Series," *Journal of the Royal Statistical Society. Series D (The Statistician)*, vol. 35, n° 2, pp. 271-280, 1986.
- [23] M. Troffaes, J. Gledhill, D. Škulj, and S. Blake, "Using imprecise continuous-time Markov chains for assessing the reliability of power networks with common cause failure and non-immediate repair," SIPTA working papers, 2015.
- [24] J. Y. Campbell, A. W. C. Lo, and A. C. MacKinlay, "The econometrics of financial markets," Princeton University Press, 1997.
- [25] P. M. Robinson, "The estimation of a nonlinear moving average model," *Stochastic processes and their applications*, vol. 5, n°1, pp. 81-90, 1977.
- [26] D. B. Nelson, "ARCH models as diffusion approximations," *Journal of econometrics*, vol. 45, n° 1-2, pp. 7-38, 1990.
- [27] R. Adhikari, and R. K. Agrawal, "An introductory study on time series modeling and forecasting," arXiv preprint arXiv:1302.6613, 2013.
- [28] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feed-forward networks are universal approximators", *Neural Networks*, vol. 2, pp. 359-366, 1989.
- [29] M. Garetti, and M. Taisch, "Neural networks in production planning and control," *Production Planning & Control*, vol. 10, n° 4, pp. 324-339, 1999.
- [30] I. A. Basheer, and M. Hajmeer, "Artificial neural networks: fundamentals, computing, design, and application," *Journal of microbiological methods*, vol. 43, n° 1, pp. 3-31, 2000.
- [31] A. S. Ahmad, M. Y. Hassan, M. P. Abdullah, H. A. Rahman, F. Hussin, H. Abdullah, and R. Saidur, "A review on applications of ANN and SVM for building electrical energy consumption forecasting," *Renewable and Sustainable Energy Reviews*, vol. 33, pp. 102-109, 2014.
- [32] K. Haddouch, K. Elmoutaoukil, and M. Ettaouil, "Solving the Weighted Constraint Satisfaction Problems Via the Neural Network Approach." *International Journal of Interactive Multimedia and Artificial Intelligence*, 4(1), 56-60, 2016.



Rosa María Cantón Croda

Ph.D. in Computer Science from the Tecnológico de Monterrey, Ciudad de México, Master in Information Technologies from the Tecnológico de Monterrey, Veracruz, was graduated in Computer Science Administrator from the Tecnológico de Monterrey, Monterrey. She was Manager of Systems in transport and construction companies. More than twenty years of experience in the academic administration of Tecnológico de Monterrey and currently Dean of postgraduate studies in Engineering and Business at UPAEP. She has written some articles for national and international congresses. She was certified in PMBook, Positive Psychology by Tecmilenio and in Project-Based Learning by Aalborg University in Denmark. Currently, her research interests are Big Data and Business Intelligence.



Damián Emilio Gibaja Romero

Ph.D. in Economics from El Colegio de México, Ciudad de México, Master in Economics from El Colegio de México, México. He has written some articles for national and international congresses. He did research visits at the Paris School of Economics and the University of Glasgow. Currently, his research interests are Game Theory and Mathematical Economics.



Santiago Omar Caballero Morales

Santiago Omar Caballero Morales is a professor-researcher in the Department of Logistics and Supply Chain Management at Universidad Popular Autónoma del Estado de Puebla (UPAEP) in Mexico. In 2009, he received a Ph.D. in Computer Science from the University of East Anglia in the United Kingdom. Since 2011 he has been a member of the National System of Researchers (SNI) in Mexico.

His research interests are quality control, operations research, combinatorial optimization, pattern recognition, analysis and simulation of manufacturing processes, and human-robot interaction.