

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/4152128>

Single exponential smoothing method and neural network in one method for time series prediction

Conference Paper · February 2004

DOI: 10.1109/ICCIS.2004.1460680 · Source: IEEE Xplore

CITATIONS

10

READS

262

3 authors:



Dimche Risteski

Max Planck Institute for Innovation and Competition

6 PUBLICATIONS 78 CITATIONS

[SEE PROFILE](#)



Andrea Kulakov

Ss. Cyril and Methodius University in Skopje Macedonia

87 PUBLICATIONS 819 CITATIONS

[SEE PROFILE](#)



Danco Davcev

Ss. Cyril and Methodius University in Skopje

208 PUBLICATIONS 1,316 CITATIONS

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:



Multimedia content delivery from cloud to mobile devices [View project](#)



MODELS FOR PERCEPTION OF QUALITY [View project](#)

SINGLE EXPONENTIAL SMOOTHING METHOD AND NEURAL NETWORK IN ONE METHOD FOR TIME SERIES PREDICTION

Dimce Risteski

Computer Science Department
Faculty of Electrical Engineering
Skopje, Macedonia
rdimce@yahoo.com

Andrea Kulakov

Computer Science Department
Faculty of Electrical Engineering
Skopje, Macedonia
kulak@etf.ukim.edu.mk

Danco Davcev

Computer Science Department
Faculty of Electrical Engineering
Skopje, Macedonia
ctfdav@etf.ukim.edu.mk

Abstract— the purpose of this paper is to present a new method that combines statistical techniques and neural networks in one method for the better time series prediction. In this paper we presented single exponential smoothing method (statistical technique) merged with feed forward back propagation neural network in one method named as Smart Single Exponential Smoothing Method (SSESM). The basic idea of the new method is to learn from the mistakes. More specifically, our neural network learns from the mistakes made by the statistical techniques. The mistakes are made by the smoothing parameter, which is constant. In our method, the smoothing parameter is a variable. It is changed according to the prediction of the neural network. Experimental results show that the prediction with a variable smoothing parameter is better than with a constant smoothing parameter.

I. INTRODUCTION

To stay competitive, companies must plan their activities into the future. This planning is vital for the success on the market. Companies collect information about their activities as sales, expenses, profit and many other aspects of their operations. In the human nature is to use past to predict the future. To do successful planning companies must predict the future. Time series data is an ordered sequence of values of a variable at equally spaced time intervals. The time series prediction goal is to observe or model the existing data series to enable future unknown data values to be forecasted accurately. Examples of data series include financial data series (stocks, indices, rates, etc.), physically observed data series (sunspots, weather, etc.), and mathematical data series (Fibonacci sequence, integrals of differential equations, etc.). The phrase "time series" generically refers to any data series, whether or not the data are dependent on a certain time increment. To do time series prediction we can use many different forecasting techniques, which are based on historical time series data and the trends the data reveal. In modern research, different methods taken from a variety of fields are employed for this task [1].

Neural Networks (NN) has been widely used as time series forecasters: most often these are feed-forward networks [2]. Typical examples of use of neural networks in financial

forecasting are explained in many papers, for instance in stock forecasting [7], bankruptcy prediction for credit risk [5] and some guidelines for financial forecasting [6]. Proposals for better time series forecasting with NN, are given in many research papers published on various international journals and conferences [3, 4, 11]. Reference [13] gives guidelines and limitations in time series forecasting with feed-forward neural networks. The conclusion from that paper is that feed-forward neural networks are extremely sensitive to architecture and parameter choice. The future of neural networks time series forecasting seems to be in more complex network types that merge other technologies with neural networks. The example of that kind of time series prediction is given in [4] where the authors use smoothing unit based on the wavelet multiresolution analysis to reduce the influence of noise in time series prediction with neural networks. They merge two different technologies to make a better time series prediction.

The other techniques that are used for the time series prediction are statistical methods or techniques [11]. The most common used method for forecast involving time series are exponential smoothing [9]. Exponential smoothing is a statistical family of methods or techniques. The basic method of this family is a Single Exponential Smoothing Method (SESM). Single exponential smoothing method is very popular statistical method for the prediction of time-series. The basic idea of our method named as Smart Single Exponential Smoothing Method (SSESM) was to learn from the error that the SESM produces and to predict that error. The error is made by smoothing parameter. If we choose the right smoothing parameter then we can make a better prediction. SESM uses smoothing parameter with a constant value. In our method the better prediction is provided by using the neural network which learns when to increase or decrease the smoothing parameter from the train data. The smoothing parameter is changed according to the prediction of the neural network. The comparison of predictions of our method and SESM shows that our method makes better predictions. In the second section of this paper we present in more details the single exponential smoothing method, while the feed-forward neural networks with back propagation training is described in section 3. In section 4 we present our method, while in section 5 the evaluation results are given. The last section concludes the paper.

II. SINGLE EXPONENTIAL SMOOTHING METHOD

Exponential Smoothing is a very popular statistical family of techniques and scheme to produce a smoothed time series. This family of statistical techniques is the most common techniques for forecasts involving time series. Exponential smoothing assigns exponentially decreasing weights as the observation get older. In other words, recent observations have relatively higher weights in forecasting than the older observations. In exponential smoothing there is one or more smoothing parameters to be determined (or estimated) and these choices determine the weights assigned to the observations. The single exponential smoothing method is an exponential smoothing method with one smoothing parameter. The general formula for the calculations of the prediction of single exponential smoothing method is

$$\Phi_t = (\alpha)(K_{t-1}) + (1-\alpha)(\Phi_{t-1}), \quad (1)$$

where Φ_0 presents starting point value that usually is average of the data over some period, Φ_t presents forecast for time period ($t=1,2,\dots$), K_0 presents previous time period's data value, K_t presents data value for time period ($t=1,2,\dots$), while α presents smoothing parameter which is a constant between 0 and 1.

Equation (1) is a basic equation of exponential smoothing. One drawback of the method is that we have to use a great deal of our data to calculate a starting point value (Φ_0) for the method. This value is usually the mean or average of the data over some period. A smoothing parameter denoted by α must be chosen to jump-start the forecast calculations. In the process of forecasting smoothing parameter is a constant. The smoothing parameter must be between 0 and 1. The smaller parameter, the higher weight will be given to (initially) starting average (Φ_0).

III. FEED-FORWARD BACK PROPAGATION NEURAL NETWORK

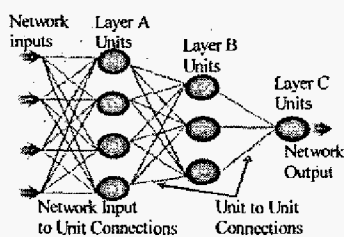


Figure 1. Feed-Forward Neural Network

Fig.1 depicts an example feed-forward neural network. A neural network (NN) can have any number of layers, units per layer, network inputs, and network outputs. This network has four units in the first layer (layer A) and three units in the second layer (layer B), which are called hidden layers. This network has one unit in the third layer (layer C), which is called the output layer. Finally, this network has four network inputs and one network output. All data propagate along the connections in the direction from the network inputs to the network outputs, hence the term feed-forward. The learning algorithm for this network is back propagation.

To make meaningful forecasts, the neural network has to be trained on an appropriate data series. Examples in the form of <input, output> pairs are extracted from the data series, where input and output are vectors equal in size to the number of network inputs and outputs, respectively. Then, for every example, back propagation training consists of three steps:

1. Present an example's input vector to the network inputs and run the network: compute activation functions sequentially forward from the first hidden layer to the output layer (Fig.1 from layer A to layer C).
2. Compute the difference between the desired output for that example, output, and the actual network output (output of unit(s) in the output layer). Propagate the error sequentially backward from the output layer to the first hidden layer (Fig.1 from layer C to layer A).
3. For every connection, change the weight modifying that connection in proportion to the error.

When these three steps have been performed for every example from the data series, one epoch has occurred. Training usually lasts thousands of epochs; possibly until a predetermined maximum number of epochs (epochs limit) are reached or the network output error (error limit) falls below an acceptable threshold. Training can be time-consuming, depending on the network size, number of examples, epochs limit, and error limit.

Over fitting is another major concept in the design of a NN. Neural network that produces high forecasting error on unforeseen inputs, but low error on training inputs, is said to have over fitting the train data. When there is no enough data available to train the NN and the structure of NN is too complex, the NN tends to memorize the data rather than to generalize from it. Keeping the NN small is one way to avoid over fitting. In order to train NN better, all the data available should be used. To test the NN model, we partition the data into three parts. The first two parts are used to train (and validate) the NN while the third part of data is used to test the model. The NN is trained (with back propagation) directly on the training set, its generalization ability is monitored on the validation set, and its ability to forecast is measured on the test set. The general partition rule for training, validation and testing set is 70%, 20% and 10% respectively.

To control over fitting, the following procedure to train the network we used:

1. After every n epoch, we sum the total squared error for all examples from the training set.
2. Also, sum the total squared error for all examples from the validation set.
3. Stop training when the trend in the error from step 1 is downward and the trend in validation error from step 2 is upward.

When consistently the error in step 2 increases, while the error in step 1 decreases, this indicates the network has over-learned or over fitted the data and training should stop.

IV. SMART SINGLE EXPONENTIAL SMOOTHING METHOD

Smart Single Exponential Smoothing Method (SSESM) is based on the single exponential smoothing method (SESM) and feed forward back propagation neural network (NN). The idea for this new method was to learn from the mistakes that SESM makes. The mistakes in the SESM are made by the smoothing parameter, which is constant. The difference between the right value for the smoothing constant for one time interval and the best smoothing constant of the SESM, represents a factor that we call change factor (γ). This factor has two states. One state means, increase the smoothing constant, while the other means decrease the smoothing constant. In other words, the mistakes that SESM made are incorporated in the change factor (γ). NN learns from the time series of the change factor or NN learns from the mistakes that SESM makes. NN then predicts the next change factor. The trust in the NN prediction represents a factor that we call trust factor (τ). Trust factor is a constant that we define. The smoothing parameter for the SSESM depends on the change factor and trust factor. In one case, the smoothing parameter is equal to the smoothing constant of SESM increased by the trust factor, in another case it is equal to smoothing constant of SESM decreased by the trust factor. The procedure for making prediction with SSESM consists of two major steps. The first step is preparation step, and the second step is prediction step. The SSESM schema is show on Fig.2.

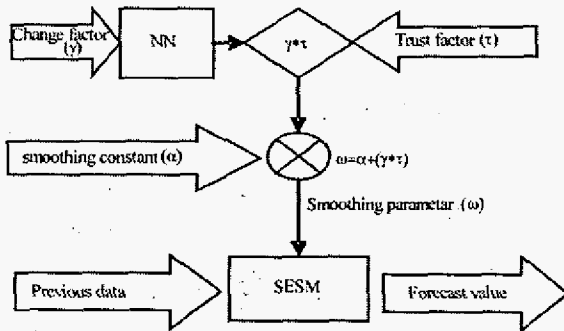


Figure 2. Smart Singel Exponential Smoothing Method Shema

A. Preparation step

In the preparation step we train neural network to predict change factor. First we determined the best smoothing constant for SESM on data series by the several measures that are commonly used to determine the performance of forecasting methods. After we make prediction with SESM, we calculate the best smoothing constant for every time interval (α_t) in time series by (2) which is derived from (1)

$$(\alpha_t) = (K_{t-1} - \Phi_{t-1}) / (\Phi_t - \Phi_{t-1}). \quad (2)$$

Then for every time interval we calculate the difference between best smoothing constant for that time interval and the

smoothing constant of SESM. If the defERENCE is positive we set (γ) to 1 else we set (γ) to -1. Now we have another time series for a change factor (γ).The next step is training of feed forward back propagation neural network for iteratively prediction for one step ahead. Fig.3 shows examples of γ pairs that are used for training NN for iteratively prediction for one step ahead.

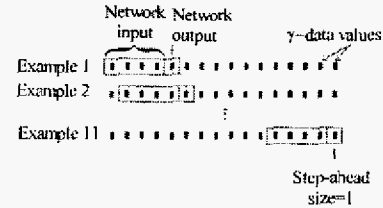


Figure 3. Examples of γ pairs for NN training

B. Prediction step

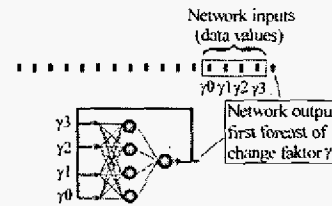


Figure 4. Iteratively prediction for one spep ahead

In the first step NN predict the next change factor (γ) Fig.4. After that the smoothing parameter (ω) for SSESM is calculate by (3)

$$\omega = \alpha + (\tau * \gamma). \quad (3)$$

In (3) the smoothing parameter α is the best smoothing constant for SESM that we found in the preparation step, τ is the trust factor and γ is the prediction of NN. In the next step, the forecast value is calculated by (1) where $\alpha = \omega$. For another prediction, we repeat the previous steps.

V. EVALUATION

There are several measures that are commonly used to determine the performance of forecasting methods. These measures are based on the forecast errors, or the differences between the actual and forecast errors. The first of these measures is called cumulative sum of the forecast errors (CFE). To get the CFE, we add all the errors, keeping in mind that some are positive and some are negative. The closer the CFE is to zero, the better our forecasting method is performing. A positive value for the CFE means that our forecast have been too low on average; a negative CFE means that the forecast have been too high. If we don't care for the positive and negative signs, we may take the absolute value of each of the forecast errors. If we sum these absolute values and divide by the total number we have, we obtain what is

called mean absolute deviation (MAD). It would be nice if this value would also be close to zero. The TS, or tracking signal, is the ratio of the CFE to the MAD. It is desirable to have the TS between about -1.5 and +1.5, although it may range a little lower or higher (say from -3.0 to +3.0) depending on the judgment of the data miner. The mean square error (MSE) measures the prediction accuracy in a similar way to MAD. It averages the sizes of prediction errors avoiding the cancelling of positive and negative terms. The MSE, instead of using the absolute value of the prediction errors, uses their square value. The advantage of using the square value of the prediction errors is that it gives more weight to large prediction errors than MAD. Since the forecast errors should usually be normally distributed, there is a rule of thumb that the ratio of the MAD to the standard deviation (ST_DEV) is about 0.8. This allows a quick check to see if the forecasting technique is accurate. We'll apply all these measures to our two data examples [12] and for two different forecasting methods we examined.

A. Experimental results from first data series

The first data series is Monthly basic iron production in Australia that contains 473 data values. In the first experiment we chose starting point to be the average of first 300 data values of data series. Then we chose the best smoothing parameter for the single exponential smoothing method with $\alpha=0.1$. In the next step we calculate the change factor which is the data source for NN training. After that we use general partition rule for training, validation and testing set is 70%, 20% and 10% respectively to train and validate the NN prediction. The architecture of NN is 8 neurons in the input layer, 6 neurons in the hidden layer and one neuron to the output layer. The best parameters for NN are: learning rate (0.3), epoch count (75) where we avoid the over fitting (according to the recommendations given in [13]). The results from the experimentation are shown in Table I.

We can see the results for single exponential smoothing method with $\alpha=0.1$ and smart single exponential smoothing with smoothing parameter $\alpha=0.1$ and $\tau=0.9$. The result of the smart exponential smoothing method is the best if the trust factor is equal to 0.9. Table I summarizes the results from the experimentations with first data series. In Table I we can see that CFE from smart single exponential smoothing with and $\tau=0.9$ show 58 % better result than CFE from single exponential smoothing with smoothing parameter equal to 0.1. The standard deviation of SSESM with $\tau=0.9$ show 39% better result than SESM. The MAD from SSESM with $\tau=0.9$ is 43% better than MAD from SESM. The MSE from SSESM with $\tau=0.9$ show 65% better result than MSE from SESM. The tracking signal from SSESM with $\tau=0.9$ is a better from tracing signal from SESM. On Fig. 5 we can see the prediction that SESM and SSESM make on data series. The Black line is real data time series, the blue line is the result from the single exponential smoothing method with smoothing constant $\alpha=0.1$ and the red line is the result from the smart single exponential smoothing with smoothing constant $\alpha=0.1$ and trust factor $\tau=0.9$. We can see that SSESM gives much better prediction than SESM.

TABLE I. RESULTS FROM THE EXPERIMENTATION WITH FIRST DATA SERIES

Measures	Method's	
	SESM with $\alpha=0.1$	SSESM with $\alpha=0.1$ and $\tau=0.9$
CFE	-76	-32
ST_DEV	28	17
MAD	23	13
MSE	765	270
TS	-3.2	-2.4

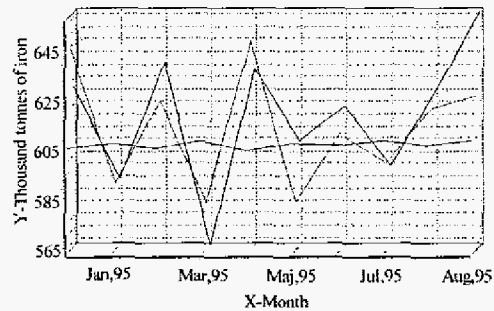


Figure 5. Prediction by SESM and SSESM on first data series

B. Experimental results from second data series

The second data series is Monthly Australian wine sales that contain 187 data values. In this experiment we chose starting point to be the average of the first 37 data values. The best smoothing parameter from the single exponential smoothing is $\alpha=0.9$. The architecture of NN is 8 neurons in the input layer, 6 neurons in the hidden layer and one neuron to the output layer. The best parameters for NN are: learning rate (0.2), epoch count (50) where we avoid the over fitting. The best result for smart single exponential smoothing prediction is show when we use a trust factor equal to $\tau=0.5$. The results are summarized in Table II.

In Table II we can see that CFE from smart single exponential smoothing with trust factor $\tau=0.5$ is much closer to 0 than CFE from single exponential smoothing with smoothing parameter equal to $\alpha=0.9$. The standard deviation of SSESM with $\tau=0.5$ show 17% better result than SESM. The MAD from SSESM with $\tau=0.5$ is 28% better than MAD from SESM. The MSE from SSESM with $\tau=0.9$ show 34% better result than MSE from SESM. The tracking signal from SSESM with $\tau=0.9$ is a better from tracing signal from SESM. On Fig.6 is shown the predictions by SESM and SSESM on second data series. The black line is real data time series, the blue line is the result from the single exponential smoothing method with smoothing constant $\alpha=0.9$ and the red line is the result from the smart single exponential smoothing with smoothing constant $\alpha=0.9$ and the trust factor $\tau=0.5$. We can see that the red line makes a better prediction than the blue line or that smart single exponential smoothing

makes better prediction than single exponential smoothing. A number of other simulations were performed on several data series given in [12]. The results of simulation confirmed the superiority of our method as shown on the two examples given in the paper.

TABLE II. RESULTS FROM THE EXPERIMENTATION WITH SECOND DATA SERIES

Measures	Method's	
	SESM with $\alpha=0.9$	SSESM with $\alpha=0.9$ and $\tau=0.5$
CFE	-1367	65
ST_DEV	520	430
MAD	413	307
MSE	260847	173998
TS	-3.3	0.2

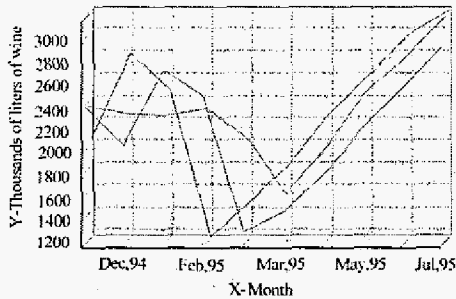


Figure 6. Prediction by SESM and SSESM on second data series

VI. CONCLUSION

This new method is a method that merges two different kind of forecasting techniques. The statistical techniques-Single exponential smoothing method and artificial intelligence techniques-feed forward neural networks. The influence of neural network to single exponential smoothing is in changes of the smoothing parameter. The neural network learns from history data when to increase or decrease the

smoothing parameter. The experimental results show that if the neural network produce right predictions than the predictions with the new method are more accurate. This means that the predictions of the new method depend on how good we predict with neural network. The results from the new Smart Single Exponential Smoothing Method show much better results than forecasting with Single Exponential Smoothing Method. In our future work, we plan to investigate the application of NN for better time series prediction of Double Exponential Smoothing Method.

REFERENCES

- [1] Weigend A. and Gershenfeld N.A, "Time Series Prediction: Forecasting the Future and understanding the past," *International Journal of forecasting* 10,1994,161-163
- [2] Michal Rosen-Zvi, Ido Kanter and Wolfgang Kinzel, "Time series prediction by feedforward neural networks - is it difficult?," *Journal of Physics A: Mathematical and General* 36, 2003, 4543-4550
- [3] R.J.Frank, N.Davey, S.P.Hunt, "Time series prediction and neural networks," in Proc. of the Fifth International Conference on Engineering Applications of Neural Networks Warsaw, Poland September 1999
- [4] Uros Lotric, Andrej Dobnikar, "Wavelet Based Smoothing in Time Series Prediction with neural networks," in Proc. of the Artificial Neural Nets and Genetic Algorithms Conference, Prague, Czech Republic, 2001, 43-46.
- [5] Amir F.Atiya, A Survey and New Results "Bankruptcy Prediction for Credit Risk Using Neural Networks," *IEEE Transactions on Neural Networks*, 12(4), 2001, 929-935.
- [6] JingTao YAO , ChewLim TAN, "Guidelines for financial forecasting with neural networks," in Proc. of the International Conference on Neural Information Processing, Shanghai, China, 2001, 757-761.
- [7] Hiroshi Takaho, Takayuki Arai, Tsuyoshi Otake and Mamoru Tanaka, "Prediction of the next stock price using Neural Networks for data mining," in Proc of the International Symposium on Nonlinear Theory and its Applications, Hian, China ,2002,163-166.
- [8] Michael Small and C.K.Tse December 2002 "Minimum description length neural network for time series prediction," *Physical Review E*, vol. 66, 066701-1-12, December 2002.
- [9] Robert P.Trueblood and John N. Lovett, JR, *Data Mining and Statistical Analysis using SQL* (Springer, Apress , 2001).
- [10] Trevor Hastie, Robert Tibshirani, Jerome Friedman , *The Elements of Statistical Learning* (Springer, Springer-Verlag, 2001).
- [11] *NIST/SEMATECH e-Handbook of Statistical Methods* , <http://www.itl.nist.gov/div898/handbook/>.
- [12] Time Series Data Library <http://www.personal.buseco.monash.edu.au/~hyndman/TSDL/>.
- [13] Eric A. Plummer, "Time series forecasting with feed-forward neural networks: Guidelines and Limitations," http://www.karlbranting.net/papers/plummer/Paper_7_12_00.doc.