Local Weighted Approach to Time Series Forecasting

Ventsislav Nikolov

Abstract: *In this paper an approach is proposed for associating priorities to the data according to their actuality and using of local neural network forecasting method. For this purpose, modified learning rules are derived that lead to shifting of the prototype vectors in the self-organizing map and weighted training in the multilayer perceptron. This method is generally applicable to the forecasting of long and complex time series.*

Key words: *Time Series Forecasting, Local Weighted Approach, Neural Networks.*

INTRODUCTION

In [5, 6] an approach is proposed for clustering proceding the forecasting. This leads to using of a set of local models which are considered together as a global composite model. This approach is generally introduced because of the need to process long time series for which the alternative global model becomes too complex. The composite model works adequately if every local model is activated at least one. If the data items in every local data region are too few then a simplified local linear model could be used. Furthermore, if the corresponding local model is not activated at least once during the recursive forecasting, then it is out of the general global knowledge that in turn leads to non-adequacy. Here a self-organizing map is used for clustering and, assuming the time series is long and complex, the local models are multilayer perceptrons trained by back-propagation algorithm [5].

The approach proposed here is based on both using of a composite model and priorities for the training patterns. A priority is considered as how many times the training pattern is presented to the neural network inputs. Usually the priority of every training pattern is unique, but this is not obligatory. The priorities are the same in both clustering and training of the local models. Moreover, the priorities may not only be integer but also real numbers. Thus, if there is an associated priority 1.5 to a given training pattern then this means that the weight of the pattern is one and a half compared to the case when priorities are not used. Combining both local and weighted approaches leads to a composed model that implements the local weighted approach.

WEIGHTED APPROACH

Using of weights for the training patterns is needed in order to take into account of the data actuality. The higher priority for the more actual data means that the last data is more important. If for example the whole data are historical observations of a given process for ten years, then the data over the last several years would be more interesting to analyzers. Similarly, the more actual data may be more important for the respective mathematical model. The priorities of the training patterns would be defined by using of typical function shapes, for example similar to the membership functions in the fuzzy theory or they could be defined by an expert. The creating of training patterns by the sliding window technique and three exemplary priority functions are graphically shown in fig.1 where higher priorities are associated to the observations closer to the current moment. After that the training patterns are considered together with their priorities and respective modifications of the training rules for the self-organizing map and multilayer perceptron should be used.

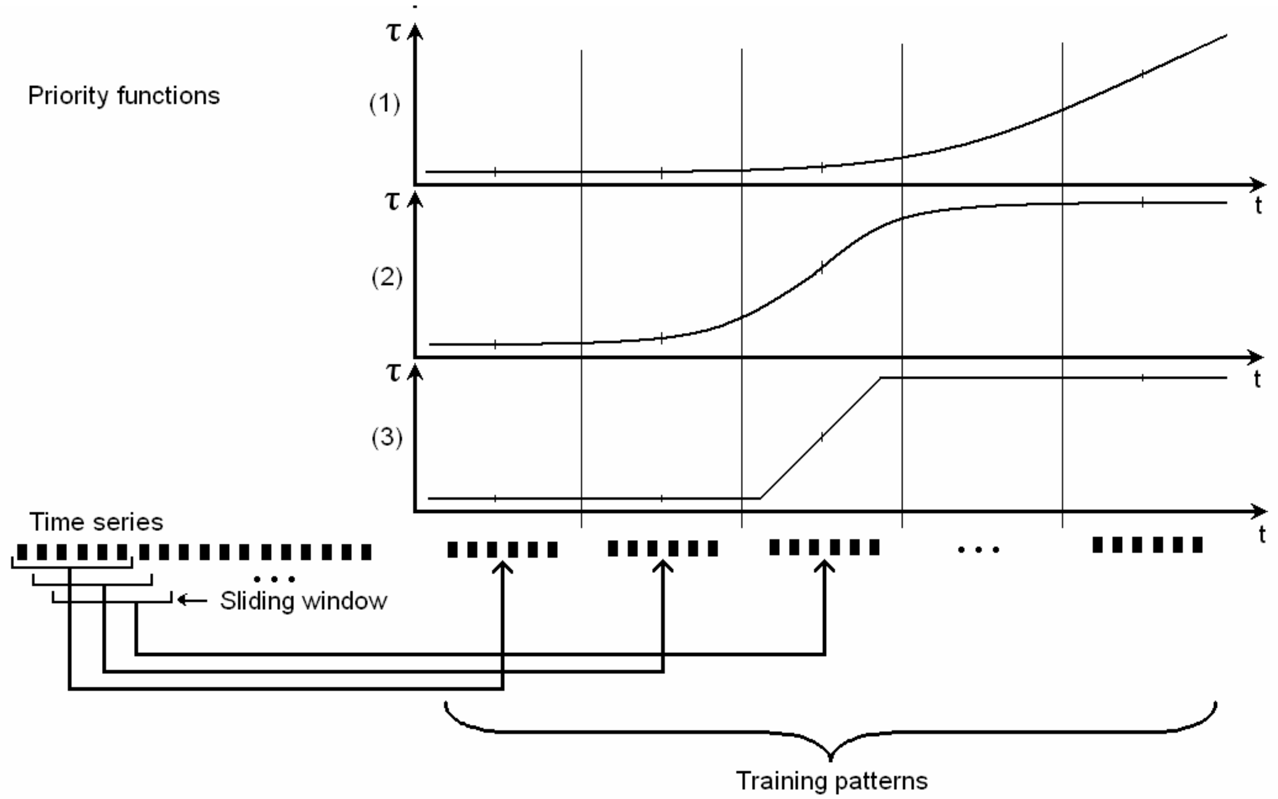

Fig.1 Three possible shapes of the priority functions

Modification of the Kohonen training rule

To derive the modified training rule the following form of the Kohonen rule [1] is used:

$$m_i(t+1) = m_i(t) + \alpha(t)h_{ci}(t)[x(t) - m_i(t)] \quad (1)$$

where $m_i(t+i)$ is the i-th weight of the connection between input and output units;
$\alpha(t)$ is the learning rate;
$h_{ci}(t)$ is the neighborhood function;
x(t) is the input value.

that may also be written as:

$$m_i(t+1) = m_i(t)[1 - \alpha(t)h_{ci}(t)] + \alpha(t)h_{ci}(t)x(t) \quad (2)$$

and after successive substitution of $m_i(t+j)$ in $m_i(t+j+1)$, j=1,2,…, τ, … the following general equation is derived:

$$m_i(t+\tau) = [m_i(t) - x(t)][1 - \alpha(t)h_{ci}(t)]^{\tau} + x(t) \quad (3)$$

where τ is the priority.

This equation could also be written as [4]:

$$m_i(t+\tau) = m_i(t) + \{1 - [1 - \alpha(t)h_{ci}(t)]^{\tau}\}[x(t) - m_i(t)] \qquad (4)$$

Modification of the generalized delta rule of the multilayer perceptron

In this training rule the output error is usually calculated as:

$$\varepsilon_k = \frac{1}{2}(y_k - o_k)^2 \qquad (5)$$

where y_k is the target output of neuron k;

$$o_k = g\left(\sum_{j=1}^{B} f\left(\sum_{i=1}^{A} x_i w_{ji}\right) w_{kj}\right) \qquad (6)$$

is the calculated output of neuron k and g and f are the activation functions of the output and hidden layer respectively.

As it is well-known the main idea of the algorithm is the output error to be minimized by the gradient descending technique. In order to modify hidden-output layer connections the following delta term is calculated:

$$\delta_k = \frac{\partial \varepsilon_k}{\partial w_{kj}} = \frac{\partial \varepsilon_k}{\partial f}\frac{\partial f}{\partial w_{kj}} \qquad (7)$$

Giving an account of the priority τ of the m-th training pattern the error function becomes:

$$\varepsilon_{km} = \frac{1}{2}(\tau_m y_k - \tau_m o_k)^2 = \tau_m \frac{1}{2}(y_k - o_k)^2 \qquad (8)$$

It is assumed that the priority is determined beforehand and it is regarded as a constant. Thus, using the rule for differentiation of a product of a function and constant the error becomes:

$$\frac{\partial \varepsilon_{km}}{\partial w_{kj}} = \frac{\partial\left(\tau_m \frac{1}{2}(y_k - o_k)^2\right)}{\partial w_{kj}} = \tau_m \frac{\partial\left(\frac{1}{2}(y_k - o_k)^2\right)}{\partial w_{kj}} \qquad (9)$$

Then the modification of the connection weight is computed as:

$$\delta_{km} = \tau_m \frac{\partial \varepsilon_k}{\partial f}\frac{\partial f}{\partial w_{kj}} \qquad (10)$$

$$\Delta w_{kj} = \eta \delta_{km} h_j = \eta \tau_m \delta_k h_j \qquad (11)$$

where h_j is the calculated output of neuron j in the hidden layer.

The last equation is the modified training rule giving an account of the training patterns priorities. Some known modifications of the back-propagation training could also be applied as using of momentum factor:

$$\Delta w_{jk}(t+1) = \eta\tau_m\delta_k h_j + \mu\Delta w_{jk}(t) \tag{12}$$

Similarly, the weights between the input and hidden layers are computed.

LOCAL APPROACH

The local approach for time series forecasting is investigated in some previous works [3, 5, 6]. When priorities are associated with the training patterns the prototypes of the clusters are shifted toward the patterns with higher priorities. The only stage in which the prototypes are used is when the sub-series in the last window of the initial time series is classified to any cluster in order to activate the respective local model – step 1 and 2 in fig. 2. As a result of prototypes shifting the sub-series in the last window may be classified in a different cluster compared to the case when non-weighted approach is used. Thus, the forecast values will be generated by a local model trained with more actual data and using the same priorities as those used in the clustering stage.

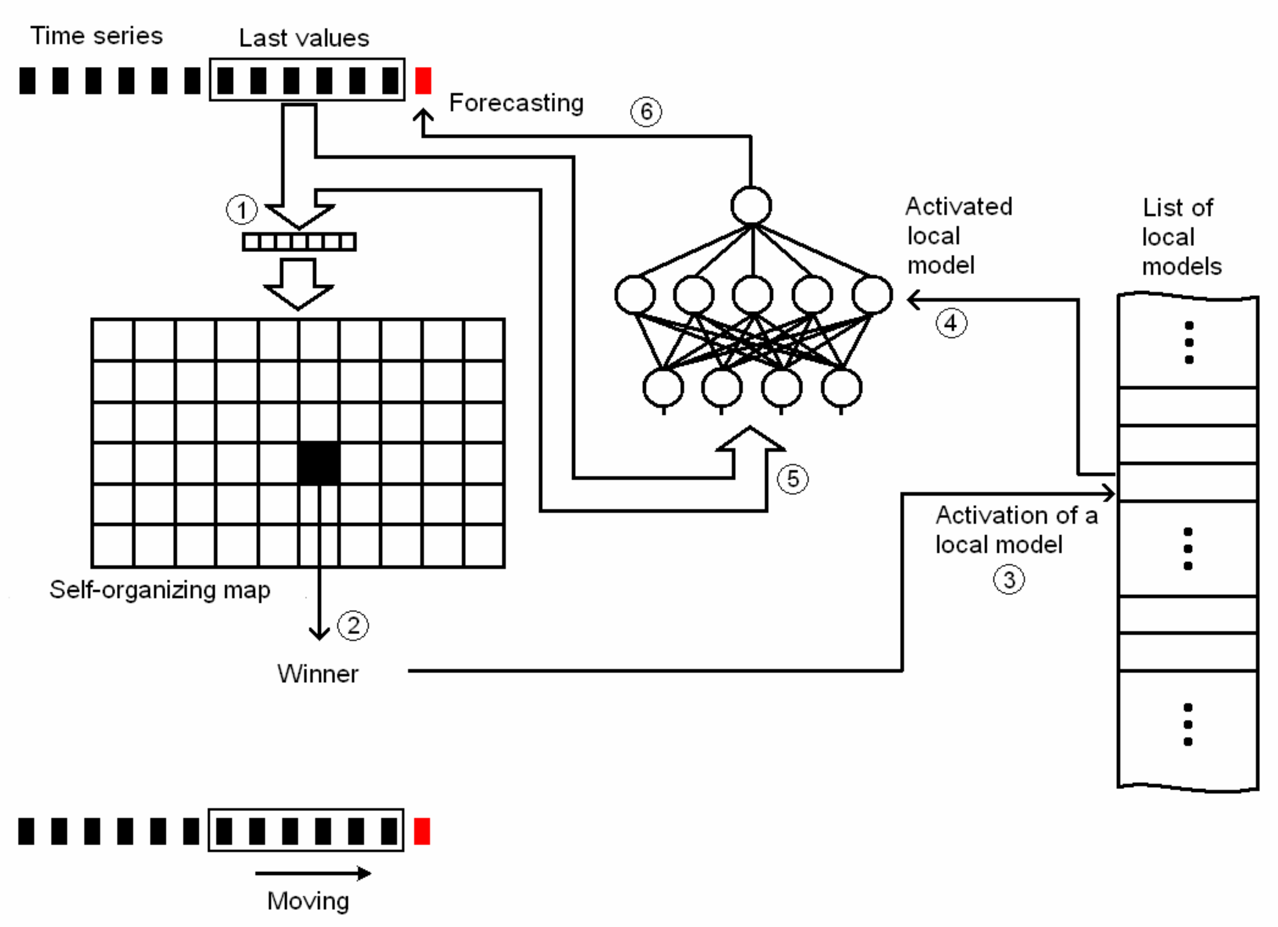

Fig.2 Local forecasting approach

RESULTS

In order to compare the results from the local weighted approach with those from the global and local non-weighted approaches the following priority function is used:

$$\tau = 3(b + \frac{t - s}{c - s}) \qquad (13)$$

where b is taken to be 1;

The function is graphically shown in fig. 3 [2]. sx and cx are 10% and 90% of the time series length respectively.

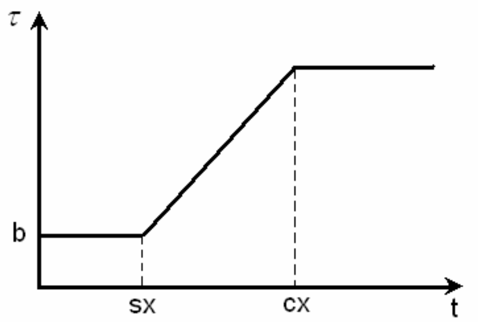

Fig. 3 Priority function used in the experimental analysis

The training of the self-organizing map and the multilayer perceptron is performed using the modified learning rules (3) and (12). The number of clusters varied from 2 to 10 for the different time series. The horizon for all time series is 30 values selected to be test data excluded from the training patterns. The results are summarized in table 1 where the time series are descending ordered according to their length. In the last tree columns the order of the approaches according to their accuracy is shown.

Table 1 Comparison between global, local non-weighted and local weighted approaches

Time series	Length	Approach		
		Global	Local	Local weighted
Radioactivity	4300	3	2	**1**
IBM shares	3000	3	2	**1**
Sun spots	3000	3	2	**1**
EUR-USD	2300	3	**1**	2
Sinusoid	900	2	3	**1**
TWI rate	550	2	3	**1**
Star	500	**1**	3	2
Electricity	400	**1**	2	3
Index S&P	320	2	3	**1**
Index M1	304	2	**1**	3
Goods import	300	**1**	3	2
CO2	300	2	3	**1**
Pressure	270	**1**	3	2
Shares	250	3	2	**1**

CONCLUSIONS

The results demonstrate that the local weighted approach is better compared to the global and local approaches in long time series forecasting. Generally, time series with more than 1000 values are better forecasted with the local approach. The local and local weighted approaches are also suitable to forecast time series in which behavior changing is observed and in this sense these approaches exploit some similar ideas as the SETAR method [7].

There are also some difficulties using the above-mentioned approaches. First of all in order to cluster data beforehand additional information is needed for the forecasting horizon. If only a few forecasted values are needed, then the clustering in many clusters may cause misuse of the whole available data because there will be local models never activated. Second, if the forecasting of the composite model is not accurate enough, then the non-adequate local model is identified with difficulty. There are also some other disadvantages generally for technical reasons that can be seen in [5]. Nevertheless, in the most cases for long and complex time series the local weighted approach shows better results.

REFERENCES

[1] Kohonen, T. Self-Organizing Maps. Springer, ISBN: 354062017-6. 2001

[2] Zhang, G. P. Neural Networks in Business Forecasting. Idea Group Publishing, ISBN: 1591401771, 2004

[3] Pavlidis, N. G., V. P. Plagianakos, D. K. Tasoulis, M. N. Vrahatis. Financial Forecasting through Unsupervised Clustering and Neural Networks. Operational Research: An International Journal, Vol. 6, No. 2, 2006, pp. 103-127

[4] Antonov, A., V. Nikolov. 2009. Decision Making System for Clustering of Spread Curves. International conference Automatics and Informatics'09 Bulgaria, Sofia 29.09 - 4.14.2009, V-9 – V-12.

[5] Nikolov, V. Optimizations in Time Series Clustering and Prediction. International Conference on Computer Systems and Technologies CompSysTech’10, Sofia, Bulgaria, 2010, pp. 528-533.

[6] Sanchez-Marono, N., O. Fontela-Romero, A. Alonso-Betanzos, B. Guijarro-Berdinas. Self-organizing maps and functional networks for local dynamic modeling. ESANN'2003 proceedings - European Symposium on Artificial Neural Networks, Bruges (Belgium), 23-25 April 2003, d-side publi., ISBN 2-930307-03-X, pp. 39-44

[7] Fu, Q., H. Fu, Y. Sun. Self-Exciting Threshold Auto-Regressive Model (SETAR) to Forecast the Well Irrigation Rice Water Requirement. Nature and Science, Vol. 2 no. 1, 2004, pp. 36-43.

ABOUT THE AUTHOR

Ventsislav Nikolov
Senior Software Developer
Eurorisk Systems Ltd.
31, General Kiselov Str., 9002 Varna, Bulgaria
E-mail: vnikolov at euroriskssystems dot com