07 - 10 September 2009

PROCEEDINGS

54. IWK

Internationales Wissenschaftliches Kolloquium
International Scientific Colloquium

Information Technology and Electrical Engineering - Devices and Systems, Materials and Technologies for the Future

Faculty of Electrical Engineering and Information Technology

TECHNISCHE UNIVERSITÄT
ILMENAU

Impressum

Herausgeber: Der Rektor der Technischen Universität Ilmenau
Univ.-Prof. Dr. rer. nat. habil. Dr. h. c. Prof. h. c.
Peter Scharff

Redaktion: Referat Marketing
Andrea Schneider

Fakultät für Elektrotechnik und Informationstechnik
Univ.-Prof. Dr.-Ing. Frank Berger

Redaktionsschluss: 17. August 2009

Technische Realisierung (USB-Flash-Ausgabe):
Institut für Medientechnik an der TU Ilmenau
Dipl.-Ing. Christian Weigel
Dipl.-Ing. Helge Drumm

Technische Realisierung (Online-Ausgabe):
Universitätsbibliothek Ilmenau
ilmedia
Postfach 10 05 65
98684 Ilmenau

Verlag: isle
Verlag ISLE, Betriebsstätte des ISLE e.V.
Werner-von-Siemens-Str. 16
98693 Ilmenau

ISBN (USB-Flash-Ausgabe): 978-3-938843-45-1
ISBN (Druckausgabe der Kurzfassungen): 978-3-938843-44-4

Startseite / Index:
http://www.db-thueringen.de/servlets/DocumentServlet?id=14089

NEURO-FUZZY ELMAN NETWORK FOR SHORT-TERM ELECTRIC LOAD FORECASTING

Ye. Bodyanskiy[1] / P. Otto[2] / S. Popov[1] / T. Rybalchenko[1]

[1]Kharkiv National University of Radio Electronics

[2]TU Ilmenau

ABSTRACT

The problem of short-term electric load forecasting (STLF) is considered. A modified architecture of Elman-type recurrent neural network is proposed. It utilizes a special fuzzification layer to deal with quantitative as well as ordinal and nominal data. The second hidden layer of the network consists of standard Rosenblatt-type neurons with sigmoidal activation functions. The context layer is formed by delay units and feeds the output signals of the second hidden layer back to its inputs. The output layer contains a single neuron with sigmoidal activation function. Several modifications of a learning algorithm for this architecture are derived based on Widrow-Hoff, Levenberg-Marquardt, and Chan-Fallside procedures. The proposed approach was tested on historical electric load data from several energy systems and showed promising results with respect to forecasting accuracy and speed of learning in comparison to feedforward neural and neuro-fuzzy systems.

Index Terms - Short-term electric load forecasting, recurrent neural networks, Elman network, learning

1. INTRODUCTION

In the context of the global economic crisis, the task of accurate electric load forecasting becomes even more important for power systems operation planning, power distribution, and operational control of the energy sector of the economy as a whole. Inaccurate electric load forecasting leads to significant economic losses. For example, in [1, 2] it is indicated that for the UK economy over the period of 1985-2000 improvement of the forecasting accuracy by 1% could lead to additional profits for energy companies of about 100 million pounds per year. This is because overestimation of future load leads to unnecessary fuel overexpenditure, and its underestimation – to the quality of power supply decrease. Both situations incur economic losses. In this regard, increasing the accuracy of electric load forecasting is indeed an urgent problem that requires efficient methods for its solution.

Among the multitude of approaches used in this task, it is possible to mark out [3] traditional methods of time series analysis, i.e. Box-Jenkins approach, Kalman filtering, adaptive systems theory, regression, correlation and spectral algorithms and as the most effective – methods of computational intelligence and, above all, artificial neural networks (ANNs). The success of ANNs in this problem is due to the nonlinear nature of the processes under investigation, high level of uncertainty (structural and parametric) about their properties, their stochastic and chaotic nature. All this hinders the effective use of traditional methods of statistical analysis and adaptive forecasting.

At present, it can be noted a large number of successful examples of ANNs use in the task of electric load forecasting in different countries [4-11]. The vast majority of cases is based on the “workhorse of neural networks” [3] – multilayer perceptron with all its modifications, united by a common feedfoward architecture. All these ANNs in terms of the random processes theory are “non-linear autoregressive models with exogenous inputs“ (NARX-models), which is a special case of more general structures containing moving average components, and known as NARMAX-models with greater flexibility and potentially higher accuracy. Theoretically, predictive NARMAX-model can be based on the conventional multilayer perceptron with global feedback [12], but training of such a system is characterized by a low speed of convergence, the need for training by epochs, inability to work with nonstationary signals. An alternative to feedforward neural networks in the tasks of forecasting could be recurrent neural networks [13], incorporating into their architecture both global and local (layer-level) feedback connections and trained by specialized procedures. There is a number of known examples of the use of recurrent ANNs in electric load forecasting tasks in Brazil, South Africa, Japan, the United States, Taiwan, the Czech Republic [3, 12, 14-17] with a sufficiently high accuracy, however, we must note quite regular nature of signals, describing the electric load in these countries.

Electric load time series analysis for Ukraine has shown a high level of nonstationarity, the presence of sharp outliers and drops, irregular trends, etc. Therefore well-known recurrent ANN architectures (Elman, Jordan, Williams-Zipser, Volterra, etc.) in their “pure” form cannot be applied and require modification. In this paper, we attempt to use the modified Elman recurrent network [18] for solving the electric load forecasting problem in Ukraine at the level of regional power grids.

2. ARCHITECTURE OF THE FORECASTING RECURRENT NEURAL NETWORK

Architecture of the modified Elman recurrent neural network is shown in Fig. 1. Its use implies that the power consumption process can be described by the output signal of a nonlinear dynamic system disturbed by many factors (weather, time, etc.), including the previous system’s states. In addition to traditional hidden and output layers of ANN, Elman proposed to introduce an additional layer of feedback connections, called the context or the states layer. This layer receives signals from the output of the hidden layer and sends them to the previous layer through the delay elements z^{-1}, thus preserving the processed information from previous cycles within the network. Our modification concerns the first and the output layers of the recurrent neural network.

The role of the architectural “building blocks” here play the standard neurons (elementary Rosenblatt perceptrons) with sigmoid-type activation functions, delay elements z^{-1} and fuzzification blocks intended to convert the ordinal and nominal input variables characterizing the influence of the environment into a quantitative form. Thus, the modified ANN has an additional (first) hidden layer, which fully coincides with the first layer of the forecasting NARX neuro-fuzzy system proposed in [19]. In addition, instead of an adaptive linear combiner in the output layer, we employ a non-linear neuron with a sigmoidal activation function, which improves the extrapolation properties of the network.

Output signals of the first hidden layer of delays and fuzzification in the form of $(n\times 1)$-vector $o^{[1]}(k)=(o_1^{[1]}(k),\ldots,o_n^{[1]}(k))^T$ with components describing current electric load $y(k)$, its past values $y(k-1),\ldots,y(k-d)$, time and weather characteristics converted into a quantitative form by fuzzification blocks, are fed to the second hidden layer. It is formed by n_2 identical neurons with sigmoidal activation functions $\psi_j^{[2]}, j=1,2,\ldots,n_2$ and $n_2(1+n+n_2)$ tuned synaptic weights $w_{ji}^{[2]}$. It can be seen from the shown architecture that the output signal of the j-th neuron of the second hidden layer can be presented as

$$\begin{cases} u_j^{[2]}(k)=w_{j0}^{[2]}+\sum_{i=1}^{n} w_{ji}^{[2]}o_i^{[1]}(k)+\sum_{i=1}^{n_2} w_{ji}^{C}o_i^{[2]}(k-1), \\ o_j^{[2]}(k)=\psi_j^{[2]}(u_j^{[2]}(k)), j=1,2,\ldots,n_2 \end{cases} \quad (1)$$

(here $k=0,1,2,\ldots$ – current discrete time, $w_{j0}^{[2]}$ – bias of the j-th neuron of the second hidden layer). The output of the whole layer can be presented in a vector-matrix notation

$$o^{[2]}(k)=\Psi^{[2]}(W^{[2]}x^{[2]}(k)+W^{C}o^{[2]}(k-1)), \quad (2)$$

where $o^{[2]}(k)-(n_2\times 1)$-vector signal, which then is sent to the output layer in the form $x^{[3]}(k)=(1,o^{[2]T}(k))^T$, $\Psi^{[2]}=diag\left\{\psi_j^{[2]}\right\}-(n_2\times n_2)$-diagonal activation functions matrix, $W^{[2]}-(n_2\times(n+1))$-matrix of the tuned synaptic weights of the second hidden layer, $x^{[2]}(k)=(1,o^{[1]T}(k))^T-(n+1)\times 1$-vector of input signals of the second hidden layer coming from the first hidden layer, $W^{C}-(n_2\times n_2)$-matrix of the tuned synaptic weights of the context layer.

The context layer is formed by n_2 delay elements z^{-1}. The delayed signals $o_j^{[2]}(k-1)$ are again fed via the synaptic weights w_{ji}^{C} to the neurons of the second hidden layer. Combining all input signals of the second hidden layer in the aggregate vector $\tilde{x}^{[2]}(k)=(1,o^{[1]T}(k),o^{[2]T}(k-1))^T$ of dimension $(1+n+n_2)\times 1$, we can write the transform realized by the second hidden layer and the context layer together in the form

$$o^{[2]}(k)=\Psi^{[2]}(W^{[2]C}\tilde{x}^{[2]}(k)), \quad (3)$$

where the matrix of tuned synaptic weights $W^{[2]C}$ has the dimension $n_2\times(1+n+n_2)$.

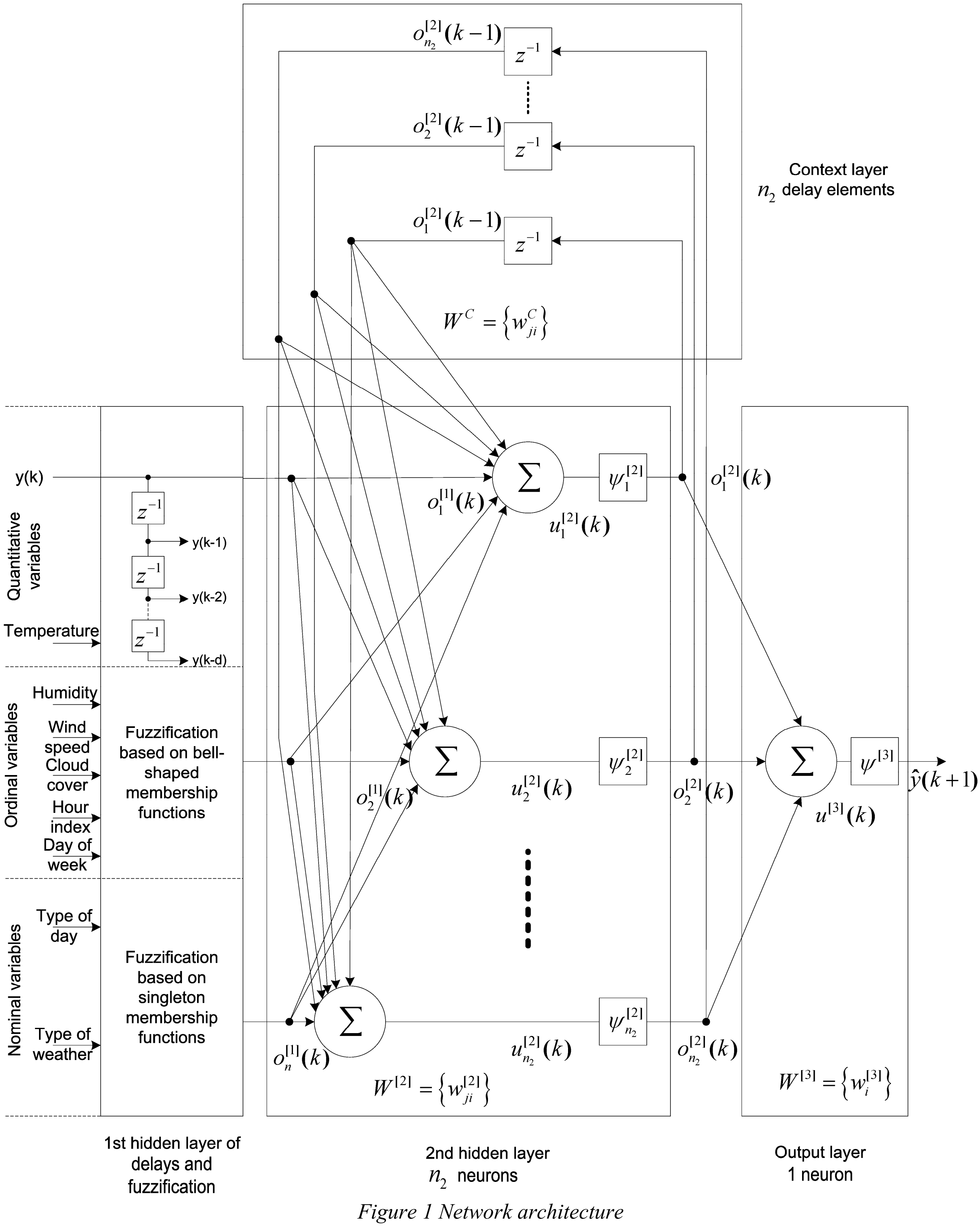

Figure 1 Network architecture

The output layer of the modified recurrent neural network is formed by a single neuron that produces the scalar forecast

$$\hat{y}(k+1) = \psi^{[3]}(u^{[3]}(k)) = \\ = \psi^{[3]}(\sum_{i=0}^{n_2} w_i^{[3]} x_i^{[3]}(k)) = \psi^{[3]}(w^{[3]T} x^{[3]}(k)), \quad (4)$$

where $w^{[3]} - (n_2+1)\times 1$-vector of tuned synaptic weights of the output layer.

In a general form, the transformation performed by the proposed modified Elman network can be expressed in the following form

$$\begin{cases} o^{[2]}(k) = \Psi^{[2]}(W^{[2]}x^{[2]}(k) + W^C o^{[2]}(k-1)) = \\ \qquad = \Psi^{[3]}(W^{[2]C}\tilde{x}^{[2]}(k)), \\ \hat{y}(k+1) = \psi^{[3]}(w^{[3]T}x^{[3]}(k)). \end{cases} \tag{5}$$

3. LEARNING OF THE FORECASTING RECURRENT NEURAL NETWORK

Learning of the proposed network will be performed as a step-wise minimization of the standard local quadratic criterion

$$\begin{aligned} E(k+1) &= \frac{1}{2}e^2(k+1) = \frac{1}{2}(y(k+1) - \hat{y}(k+1))^2 = \\ &= \frac{1}{2}(y(k+1) - \psi^{[3]}(u^{[3]}(k)))^2 = \\ &= \frac{1}{2}(y(k+1) - \psi^{[3]}(\sum_{i=0}^{n_2} w_i^{[3]}x_i^{[3]}(k)))^2, \end{aligned} \tag{6}$$

where

$$\psi^{[s]}(u^{[s]}(k)) = \frac{1}{1+e^{-\gamma u^{[s]}(k)}}, \tag{7}$$

$\gamma > 0$ is a parameter defining the shape of the activation function, which also can be tuned; $s = 2,3$ – layer index.

The learning process is based on the backpropagation concept and starts by tuning the synaptic weights of the output neuron. For the current time instant k, for which values of $y(k), e(k)$ are available, a gradient descent procedure that minimizes (6) can be written in the form

$$\begin{aligned} & w_i^{[3]}(k+1) = w_i^{[3]}(k) - \\ & -\eta^{[3]}(k)\frac{\partial E(k)}{\partial e(k)} \bullet \frac{\partial e(k)}{\partial u^{[3]}(k)} \bullet \frac{\partial u^{[3]}(k)}{\partial w_i^{[3]}(k)} = \\ & = w_i^{[3]}(k) + \eta^{[3]}(k)e(k)\frac{\partial \psi^{[3]}(u^{[3]}(k))}{\partial u^{[3]}(k)}x_i^{[3]}(k) = \\ & = w_i^{[3]}(k) + \eta^{[3]}(k)\delta^{[3]}(k)x_i^{[3]}(k), i = 0,1,\ldots,n_2, \end{aligned} \tag{8}$$

where $\eta^{[3]}(k)$ is a search step parameter that is usually chosen empirically, $\delta^{[3]}(k) = e(k)\frac{\partial \psi^{[3]}(u^{[3]}(k))}{\partial u^{[3]}(k)} = \frac{\partial E(k)}{\partial u^{[3]}(k)}$ – local error (δ-error) of the output layer.

Algorithm (8) can be rewritten in a compact vector form

$$\begin{aligned} & w_i^{[3]}(k+1) = w_i^{[3]}(k) + \eta^{[3]}(k)\delta^{[3]}(k)x^{[3]}(k) = \\ & = w^{[3]}(k) + \eta^{[3]}(k)\delta^{[3]}(k)\nabla_{w^{[3]}}u^{[3]}(k), \end{aligned} \tag{9}$$

and considering (7) –

$$\begin{aligned} & w^{[3]}(k+1) = w^{[3]}(k) + \\ & +\eta^{[3]}(k)\gamma e(k)\hat{y}(k)(1-\hat{y}(k))x^{[3]}(k). \end{aligned} \tag{10}$$

Properly choosing the value of the search step parameter $\eta^{[3]}(k)$, it is possible to increase the learning speed. Consider a one-step modification of the Levenberg-Marquardt learning procedure [20]

$$\begin{aligned} & w^{[3]}(k+1) = w^{[3]}(k) + (\nabla_{w^{[3]}}u^{[3]}(k)\nabla_{w^{[3]}}^T u^{[3]}(k) + \\ & +\rho I_{n_2+1})^{-1}\delta^{[3]}(k)\nabla_{w^{[3]}}u^{[3]}(k) = \\ & = w^{[3]}(k) + (x^{[3]}(k)x^{[3]T}(k) + \rho I_{n_2+1})^{-1}\delta^{[3]}(k)x^{[3]}(k) \end{aligned} \tag{11}$$

(here $\rho > 0$ is a regularization term, I_{n_2+1} – $((n_2+1)\times(n_2+1))$ identity matrix). Using simple transformations based on Sherman-Morrison formula or Moore-Penrose pseudo-inverse, (11) can be rewritten in a simple form

$$w^{[3]}(k+1) = w^{[3]}(k) + \frac{\delta^{[3]}(k)x^{[3]}(k)}{\rho + \left\|x^{[3]}(k)\right\|^2}, \tag{12}$$

which structurally coincides with an additive form of Kaczmarz identification procedure [21] and with Widrow-Hoff learning algorithm when $\rho = 0$.

Analysis of (10) and (12) shows that the learning process severely decelerates on the “tails” of sigmoidal functions where their derivatives are close to zero. In this case, it is reasonable to use regularized procedures, e.g. of Chan-Fallside type [22]

$$\begin{aligned} & w^{[3]}(k+1) = w^{[3]}(k) + \\ & +\eta^{[3]}\delta^{[3]}(k)x^{[3]}(k) + \rho\Delta w^{[3]}(k-1), \end{aligned} \tag{13}$$

(here $\eta^{[3]}$ – $\text{const} > 0$; $1 > \rho > 0$; $\Delta w^{[3]}(k-1) = w^{[3]}(k) - w^{[3]}(k-1)$), that successfully passes plateaus of the target function, which are caused by “tails” of sigmoidal functions.

To accelerate convergence of (13), it is possible to make its hybrid with (12) in the following form [23]

$$\begin{aligned} & w^{[3]}(k+1) = w^{[3]}(k) + \\ & + \frac{\eta^{[3]}\delta^{[3]}(k)x^{[3]}(k) + \rho\Delta w^{[3]}(k-1)}{\left\|x^{[3]}(k)\right\|^2}. \end{aligned} \tag{14}$$

The algorithm of simultaneous tuning of synaptic weights of the second hidden layer and the context layer can be expressed in the form

$$\begin{aligned} & w_{ji}^{[2]C}(k+1) = w_{ji}^{[2]C}(k) - \eta^{[2]C}(k)\frac{\partial E(k)}{\partial w_{ji}^{[2]C}(k)} = \\ & = w_{ji}^{[2]C}(k) - \eta^{[2]C}(k)\frac{\partial E(k)}{\partial u_j^{[2]}(k)} \bullet \frac{\partial u_j^{[2]}(k)}{\partial w_{ji}^{[2]C}(k)} = \\ & = w_{ji}^{[2]C}(k) + \eta^{[2]C}(k)\delta_j^{[2]C}(k)\tilde{x}_i^{[2]}(k), \end{aligned} \tag{15}$$

where $\delta_j^{[2]C}(k) = \frac{\partial E(k)}{\partial u_j^{[2]}(k)}, j = 1,2,\ldots,n_2$; $i = 0,1,\ldots,n+n_2$.

If we write the local error of these layers as

$$\delta_j^{[2]C}(k) = \frac{\partial E(k)}{\partial u_j^{[2]}(k)} = \frac{\partial E(k)}{\partial o_j^{[2]}(k)} \bullet \frac{\partial o_j^{[2]}(k)}{\partial u_j^{[2]}(k)} \tag{16}$$

and take into account that

$$o_j^{[2]}(k) = \psi_j^{[2]}(u_j^{[2]}(k)), \tag{17}$$

it is easy to obtain

$$\delta_j^{[2]C}(k) = \frac{\partial E(k)}{\partial o_j^{[2]}(k)} \bullet \frac{\partial \psi_j^{[2]}(u_j^{[2]}(k))}{\partial u_j^{[2]}(k)}. \tag{18}$$

Then substituting $\dfrac{\partial E(k)}{\partial o_j^{[2]}(k)}$ in the form

$$\begin{aligned} &\frac{\partial E(k)}{\partial o_j^{[2]}(k)} = \frac{\partial E(k)}{\partial u^{[3]}(k)} \bullet \frac{\partial u^{[3]}(k)}{\partial o_j^{[2]}(k)} = \\ &= \frac{\partial E(k)}{\partial u^{[3]}(k)} \bullet \frac{\partial}{\partial o_j^{[2]}(k)} \sum_{p=0}^{n_2} w_p^{[3]}(k) x_p^{[3]}(k) = \\ &= \delta^{[3]}(k) w_j^{[3]}(k), \end{aligned} \tag{19}$$

we can rewrite (18) as

$$\delta_j^{[2]C}(k) = \frac{\partial \psi_j^{[2]}(u_j^{[2]}(k))}{\partial u_j^{[2]}(k)} \delta^{[3]}(k) w_j^{[3]}(k). \tag{20}$$

Then it follows

$$\begin{aligned} &w_{ji}^{[2]C}(k+1) = w_{ji}^{[2]C}(k) + \\ &+\eta^{[2]C}(k)\tilde{x}_i^{[2]}(k) \frac{\partial \psi_j^{[2]}(u_j^{[2]}(k))}{\partial u_j^{[2]}(k)} \delta^{[3]}(k) w_j^{[3]}(k) = \\ &= w_{ji}^{[2]C}(k) + \eta^{[2]C}(k)\tilde{x}_i^{[2]}(k)\delta_j^{[2]}(k), \end{aligned} \tag{21}$$

where $\delta_j^{[2]C}(k) = \dfrac{\partial \psi_j^{[2]}(u_j^{[2]}(k))}{\partial u_j^{[2]}(k)} \delta^{[3]}(k) w_j^{[3]}(k).$

Thus the process of tuning of the neurons of the second hidden layer with additional inputs from the context layer in a general form can be written in the form

$$w_j^{[2]C}(k+1) = w_j^{[2]C}(k) + \eta^{[2]C}(k)\delta_j^{[2]C}(k)\tilde{x}^{[2]}(k), \tag{22}$$

and taking into account (14) –

$$\begin{aligned} &w_j^{[2]C}(k+1) = w_j^{[2]C}(k) + \\ &+ \frac{\eta^{[2]C}\delta_j^{[2]C}(k)\tilde{x}^{[2]}(k) + \rho \Delta w_j^{[2]C}(k-1)}{\left\| \tilde{x}^{[2]}(k) \right\|^2}. \end{aligned} \tag{23}$$

In the process of learning of the modified recurrent neural network, first it is necessary to compute local errors $\delta^{[3]}(k), \delta_j^{[2]C}(k)$ sequentially, and search step parameters $\eta^{[3]}(k), \eta^{[2]C}(k)$, and then proceed to synaptic weights tuning.

4. CONCLUSIONS

We proposed a modified architecture of recurrent forecasting neural network and its learning algorithms that can be used for short-term electric load forecasting. The first hidden layer of delays and fuzzification provides a unified way to deal with quantitative, ordinal, and nominal variables. This is important for maximum utilization of all available information about the electric load and its influencing factors.

We have tested the proposed network in the task of short-term electric load forecasting for several regional power systems of Ukraine. It performed better in terms of learning speed than specialized feedforward neuro-fuzzy systems having the same first hidden layer of delays and fuzzification. This is due to smaller number of parameters needed for a recurrent network to capture long-term dependencies in data. It also performed better in terms of forecasting accuracy than traditional Elman recurrent network because of the ability to better utilize the information given in ordinal and nominal measurement scales.

Thus, the simplicity and high processing rate of the proposed network provide advantages over traditional approaches, which are currently employed for the solution of the short-term electric load forecasting problem.

5. REFERENCES

[1] D.W. Bunn, "Forecasting loads and prices in competitive power markets," Proc. IEEE, 88, pp. 163-169, 2000.

[2] Bunn, D.W., and E.D. Farmer, eds., Comparative Models for Electrical Load Forecasting, John Wiley&Sons, New York, 1985.

[3] H.S. Hippert, C.E. Pedreira, and R.C. Souza, "Neural networks for short-term load forecasting: a review and evaluation," IEEE Trans. Power Systems, 16(1), pp. 44-55, 2001.

[4] R.O. Tkachenko, and O.M. Pavlyuk, "Electric load forecasting for the L'viv region using artificial neural networks," Visnyk NU «L'vivs'ka politehnika». Computer engineering and information technology, 450, pp. 76-80, 2002. (In Ukrainian)

[5] V.L. Prikhno, and P.A. Chernenko, "Models, methods, and software for analysis and forecasting of electric load, and for operative control of power systems," Trans. IED NASU, 3(18), IED NASU, Kyiv, pp. 26-33, 2007. (In Russian)

[6] Yu.N. Bardachev, O.V. Grinavtsev, V.I. Litvinenko, and A.A. Fefelov, "Synthesis and analysis of fuzzy neural networks operation using immune algorithms in the electric load forecasting problem," Modelyuvannya ta keruvannya stanom ekologo-ekonomichnyh system regionu, 3, pp. 47-68, 2006. (In Russian)

[7] J.T. Connor, "A Robust Neural Network Filter for Electricity Demand Prediction," Journal of Forecasting, 15(6), pp. 437-458, 1996.
[8] S.H. Ling, F.H.F. Leung, H.K. Lam, and P.K.S. Tam, "Short-term electric load forecasting based on a neural fuzzy network," IEEE Trans. Industrial Electronics, 50(6), pp. 1305-1316, 2003.
[9] A. Piras, A. Germond, B. Buchenel, K. Imhof, and Y. Jaccard, "Heterogeneous artificial neural network for short term electrical load forecasting," IEEE Trans. Power Systems, 11(2), pp. 397-402, 1996.
[10] S. Tzafestas, and E. Tzafestas, "Computational intelligence techniques for short-term electric load forecasting," Journal of Intelligent and Robotic Systems, 31, pp. 7-68, 2001.
[11] J.-L. Yuan, and T.L. Fine, Forecasting demand for electric power, in S.J. Hanson, J.D. Cowan, and C.L. Giles, eds., Advances in Neural Information Processing Systems, Morgan Kauffman, San Mateo, CA, 1993, pp. 739-746.
[12] J. Vermaak, and E.C. Both, "Recurrent Neural Networks for Short-Term Load Forecasting," IEEE Trans. Power Systems, 13(1), pp. 126-132, 1998.
[13] Mandic, D.P., and J.A. Chambers, Recurrent Neural Networks for Prediction, John Wiley&Sons, Chichester, 2001.
[14] H. Mori, and K. Hidenori, "Optimal fuzzy inference for short-term load forecasting," IEEE Trans. Power Systems, 11(1), pp. 390-396, 1996.
[15] A. Khotanzad, R. Afkhami-Rohani, T.L. Lu, A. Abaye, M. Davis, and D. Maratukulam, "ANNSTLF – A Neural-Network-Based Electric Load Forecasting System," IEEE Trans. Neural Networks, 8(4), pp. 835-846, 1997.
[16] P.-F. Pai, and W.C. Hong, "Forecasting regional electricity load based on recurrent support vector machines with genetic algorithms," Electric Power Systems Research, 74, pp. 417-425, 2005.
[17] M.R. Khan, and A. Abraham, "A Hybrid Fuzzy-Neural Network for Modelling Short-Term Demand Forecasting in Czech Republic," Second International Workshop on Intelligent Systems Design and Applications, pp. 187-194, 2002.
[18] J.L. Elman, "Finding structure in time," Cognitive Science, 14, pp. 179-211, 1990.
[19] Ye. Bodyanskiy, S. Popov, and T. Rybalchenko, Multilayer neuro-fuzzy network for short term electric load forecasting, Lecture Notes in Computer Science, Springer-Verlag, Berlin, Heidelberg, 2008, pp. 339-348.
[20] Shepherd, A.J., Second-Order Methods for Neural Networks (Fast and Reliable Training Methods for Multi-Layer Perceptrons), Springer-Verlag, London, 1997.
[21] Raybman, N.S., and V.M. Chadeev, Synthesis of models of the production processes, Energy, Moscow, 1975. (In Russian)
[22] Cichocki, A., and R. Unbehauen, Neural Networks for Optimization and Signal Processing, Teubner, Stuttgart, 1993.
[23] Ye.V. Bodyanskiy, and Ye.A. Viktorov, "Cascade orthogonal neural network based on double ortho-neurons and its learning algorithm in information processing problems," Proc. XV Int. Conf. on Automatic Control "Avtomatika – 2008", pp. 70-73, 2008. (In Russian)