ICIC Express Letters ICIC International ©2017 ISSN 1881-803X
Volume **11**, Number **3**, March **2017** pp. **735–742**

POWER LOAD FORECASTING USING ELMAN NEURAL NETWORK BASED ON EXTENDED KALMAN FILTER

Ming Sun and Shaojie Cui

College of Computer and Control Engineering
Qiqihar University
No. 42, Wenhua Street, Qiqihar 161006, P. R. China
snogisunming@163.com

Received August 2016; accepted November 2016

Abstract. *In this paper, a new method is proposed to forecast the short-term power load. In this method, an Elman neural network with multiple inputs and one output is trained by using extended Kalman filter (EKF). The detailed formulations of the method are provided in this paper. The method used in this paper combines the advantages of Elman neural network and EKF, and can directly reflect the dynamic characteristics of the system of power load, and can obtain the optimal estimation with better stability. Simulation results indicate that, compared to the gradient descent method, the method proposed in this paper can achieve better forecasting accuracy, and is more suitable for forecasting of the short-term power load.*
Keywords: Elman neural network, Extended Kalman filter, Forecasting, Short-term power load

1. **Introduction.** With the rapid development of productivity, the imbalance of supply and demand of power resource leads to a higher requirement for the prediction accuracy of power load. Hence, it is important for the electric power company to perform the prediction of power load. Currently, there are many methods to predict the power load, such as fuzzy forecasting method and artificial neural network [1].

Because of the nonlinear and dynamic characteristics of power load, a neural network that can directly reflect those characteristics of power load should be used to forecast power load. Elman neural network introduces a special context layer into multi-layer feed-forward BP neural network to have a memory ability, thereby reflecting the nonlinear and dynamic characteristics of a system. Hence, Elman neural network can be used for the prediction of power load.

It has been proven that the gradient descent based Elman neural network can obtain the satisfactory result as applied into the one-order system. However, it cannot obtain the satisfactory result as applied into the higher order system [2], because the gradient descent based Elman neural network can be easily trapped in local minima [3].

Extended Kalman filter (EKF) is a kind of nonlinear parameter-estimation method on the basis of the linear Kalman filter. Specifically, EKF transforms a nonlinear function into Taylor Series surrounding a value, and linearizes the function by cancelling the second order and the higher orders of Taylor Series. Recent studies prove that the computational cost can be reduced efficiently by using EKF to train the radial basis function (RBF) neural network [4] and the back propagation (BP) neural network [5]. However, BP neural network and RBF neural network are not suitable for modeling dynamic models because they do not have the dynamic characteristics themselves.

Compared with BP neural network and RBF neural network, Elman neural network can better reflect the dynamic characteristic of a system, because of the memory ability provided by the special context layer. Considering the advantages of EKF and Elman neural network, this paper proposes to apply EKF to training Elman neural network, and uses the trained model to forecast the short-term power load. The model of the proposed EKF-aided Elman neural network (EKF-Elman) can directly reflect the dynamic characteristic of the system, and can be used for forecasting with higher accuracy and stability. The paper is organized as follows. In Sections 2 and 3, Elman neural network and EKF are briefly introduced. And the detailed formulations of the proposed EKF-Elman are provided in Section 4. In Section 5, two cases are used to demonstrate the performance of the proposed model. And the simulation results indicate that the model of the proposed EFK-Elman is better to forecast the short-term power load. Conclusions are given in Section 6.

2. **Elman Neural Network.** Elman neural network consists of the input layer, the hidden layer, the context layer, and the output layer [6,7]. The context layer is considered as a one-step delay factor, mainly used to memorize the output value of the hidden layer in the previous moment. The structure of Elman neural network can be plotted in Figure 1.

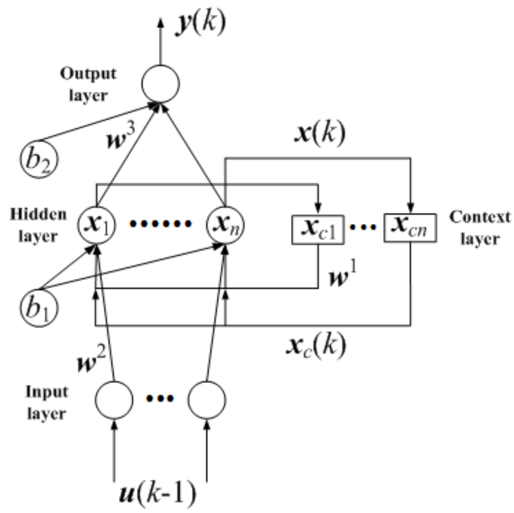

FIGURE 1. Structure of Elman neural network

As can be seen from Figure 1, the memorized output value of the hidden layer in the previous moment is acted as a next feedback input of the hidden layer by the context layer, which can make the neural network have an ability to dynamically reflect the historical data. This means that the relationship between the current input and the previous input for the neural network is considered and processed more reasonably by Elman neural network than BP neural network and RBF neural network.

The model of Elman neural network shown in Figure 1 can be expressed as follows.

$$y(k) = g\left(\left(\boldsymbol{w}^3\right)^{\mathrm{T}} \boldsymbol{x}(k)\right) \tag{1}$$

$$\boldsymbol{x}(k) = f\left(\boldsymbol{w}^1 \boldsymbol{x}_c(k) + \boldsymbol{w}^2 \boldsymbol{u}(k-1)\right) \tag{2}$$

$$\boldsymbol{x}_c(k) = \boldsymbol{x}(k-1) \tag{3}$$

where $\boldsymbol{x}_c$ is the n-dimension feedback state vector, $\boldsymbol{u}$ the r-dimension vector in the input layer, $\boldsymbol{x}$ the n-dimension output vector in the hidden layer, y the 1-dimension vector in the output layer, $\boldsymbol{w}^1$ the connection weight from the neuron of the context layer to the neuron of the hidden layer, $\boldsymbol{w}^2$ the connection weight from the neuron of the input layer to the neuron of the hidden layer, and $\boldsymbol{w}^3$ the connection weight from the neuron of the hidden layer to the neuron of the output layer. $g(\cdot)$ is the transfer function of the neuron in the output layer, which is the linear combination of neuron outputs in the hidden layer;

$f(\cdot)$ is the transfer function of the neuron in the hidden layer, represented by Sigmoid function. In the training process of Elman neural network, the sum of the squared errors is used as the objective function, shown as follows.

$$E(w) = \sum_{i=1}^{M} \left(y^i(w) - \tilde{y}^i(w)\right)^2 \tag{4}$$

where $y^i(w)$ is the actual output while $\tilde{y}^i\,(w)$ is the target output of the ith training sample.

3. **EKF.** Consider the following nonlinear system:

$$\boldsymbol{X}_k = f\left(\boldsymbol{X}_{k-1}, k-1\right) + \boldsymbol{W}_{k-1} \tag{5}$$

$$\boldsymbol{Z}_k = h(\boldsymbol{X}_k, k) + \boldsymbol{V}_k \tag{6}$$

$$\begin{cases} E\left(\boldsymbol{W}_k\right) = 0,\ E\left(\boldsymbol{W}_k \boldsymbol{W}_j^{\mathrm{T}}\right) = \boldsymbol{Q}_k \delta_{kj} \\ E\left(\boldsymbol{V}_k\right) = 0,\ E\left(\boldsymbol{V}_k \boldsymbol{V}_j^{\mathrm{T}}\right) = \boldsymbol{R}_k \delta_{kj} \\ E\left(\boldsymbol{W}_k \boldsymbol{V}_j^{\mathrm{T}}\right) = 0 \end{cases} \tag{7}$$

where $\boldsymbol{W}_k$ and $\boldsymbol{V}_k$ are the white noise sequences, independent from each other, and $\boldsymbol{Q}_k$ and $\boldsymbol{R}_k$ are diagonal matrices.

The recursive equations of EKF are shown as follows.

(1) The time update equations:

$$\hat{\boldsymbol{X}}_{k+1,k} = f\left(\hat{\boldsymbol{X}}_k, k\right) \tag{8}$$

$$\boldsymbol{P}_{k+1,k} = \boldsymbol{\Phi}_{k+1,k} \boldsymbol{P}_k \boldsymbol{\Phi}_{k+1,k}^{\mathrm{T}} + \boldsymbol{Q}_k \tag{9}$$

(2) The measurement update equations:

$$\boldsymbol{K}_k = \boldsymbol{P}_{k,k-1} \boldsymbol{H}_k^{\mathrm{T}} \left[\boldsymbol{H}_k \boldsymbol{P}_{k,k-1} \boldsymbol{H}_k^{\mathrm{T}} + \boldsymbol{R}_k\right]^{-1} \tag{10}$$

$$\hat{\boldsymbol{X}}_k = \hat{\boldsymbol{X}}_{k,k-1} + \boldsymbol{K}_k \left[\boldsymbol{Z}_k - \hat{\boldsymbol{Z}}_{k,k-1}\right] \tag{11}$$

$$\boldsymbol{P}_k = \left[\boldsymbol{I} - \boldsymbol{K}_k \boldsymbol{H}_k\right] \boldsymbol{P}_{k,k-1} \tag{12}$$

$$\hat{\boldsymbol{Z}}_{k,k-1} = h\left[\hat{\boldsymbol{X}}_{k,k-1}, k-1\right] \tag{13}$$

$$\boldsymbol{\Phi}_{k+1,k} = \left.\frac{\partial f\left[\hat{\boldsymbol{X}}_k, k\right]}{\partial \hat{\boldsymbol{X}}_k}\right|_{x_k = \hat{x}_k} \tag{14}$$

$$\boldsymbol{H}_k = \left.\frac{\partial h}{\partial \boldsymbol{X}_k}\right|_{x_k = \hat{x}_{k,k-1}} \tag{15}$$

4. **Formulation of the Proposed EKF-Elman.** Elman neural network is a typical recurrent neural network which can be capable of reflecting the dynamic characteristic of a system due to the memorized inner state of the neural network. Elman neural network has the mechanism of dynamic feedback, which can work well at the smaller number of the input, reducing the number of neurons in the input layer [8]. The gradient descent method has been proven to be able to train Elman neural network. However, the gradient descent method needs a high computational cost, limiting the use of the gradient descent method in the application.

This paper proposes to use EKF as a learning algorithm to train Elman neural network. Compared to the gradient descent method, the proposed EKF-Elman considered in this paper achieves the better forecasting accuracy while the lower computational cost. Elman neural network can be described by the following nonlinear discrete-time system:

$$\boldsymbol{\theta}(k+1) = \boldsymbol{\theta}(k+1) + \boldsymbol{\omega}(k) \tag{16}$$

$$y(k) = h(\boldsymbol{\theta}(k)) + v(k) \tag{17}$$

where $\boldsymbol{\theta}(k)$ is the state vector in the time k, $\boldsymbol{\omega}(k)$ the process noise, $y(k)$ the measurement vector, and $v(k)$ the measurement noise. It is assumed that the process noise and the measurement noise are independent Gaussian noises, and that the initial state $\boldsymbol{\theta}(0)$ is also independent from both the process noise and the measurement noise. The recursive formulations of the proposed EKF-Elman can be expressed as follows:

$$\boldsymbol{P}(k\,|k-1) = \boldsymbol{P}(k-1) + \boldsymbol{Q}(k-1) \tag{18}$$

$$\boldsymbol{H}(k) = \left.\frac{\partial h[\boldsymbol{\theta}(k)]}{\partial \boldsymbol{\theta}(k)}\right|_{\boldsymbol{\theta}(k)=\widehat{\boldsymbol{\theta}}(k-1)} \tag{19}$$

$$\boldsymbol{K}(k) = \boldsymbol{P}(k|k-1)[\boldsymbol{H}(k)]^{\mathrm{T}} \left\{\boldsymbol{H}(k)\boldsymbol{P}(k|k-1)[\boldsymbol{H}(k)]^{\mathrm{T}} + \boldsymbol{R}(k)\right\}^{-1} \tag{20}$$

$$\widehat{\boldsymbol{\theta}}\,(k) = \widehat{\boldsymbol{\theta}}(k-1) + K(k)\left\{y(k) - h\left[\widehat{\boldsymbol{\theta}}(k-1)\right]\right\} \tag{21}$$

$$\boldsymbol{P}(k) = [\boldsymbol{I} - \boldsymbol{K}(k)\boldsymbol{H}(k)]\boldsymbol{P}(k|k-1) \tag{22}$$

Consider Figure 1 with r nodes in the input layer, n nodes in the hidden layer, n feedback nodes from the context layer, and one node in the output layer. The target output of Elman neural network is represented by y, and the actual output of Elman neural network is represented by $h\left(\widehat{\boldsymbol{\theta}}\,(k)\right)$.

$$y = \begin{bmatrix} w_{10}^3 & w_{12}^3 & \cdots & w_{1n}^3 \end{bmatrix} \boldsymbol{x}^{\mathrm{T}} = \begin{bmatrix} w_{10}^3 & w_{12}^3 & \cdots & w_{1n}^3 \end{bmatrix} \begin{bmatrix} x_0 & x_1 & \cdots & x_n \end{bmatrix}^{\mathrm{T}} \tag{23}$$

where y and $\boldsymbol{x}$ represent the 1-dimension output vector of the output layer and n-dimension output vector of the hidden layer, and w_{1j}^3 represents the connection weight from the jth node in the hidden layer to the node in the output layer. For convenience,

$$\begin{bmatrix} w_{11}^1 & w_{12}^1 & \cdots & w_{1n}^1 \\ w_{21}^1 & w_{22}^1 & \cdots & w_{2n}^1 \\ \vdots & \vdots & \vdots & \vdots \\ w_{n1}^1 & w_{n2}^1 & \cdots & w_{nn}^1 \end{bmatrix} = \begin{bmatrix} (\boldsymbol{w}_1^1)^{\mathrm{T}} \\ (\boldsymbol{w}_2^1)^{\mathrm{T}} \\ \vdots \\ (\boldsymbol{w}_n^1)^{\mathrm{T}} \end{bmatrix} = \boldsymbol{w}^1 \tag{24}$$

$$\begin{bmatrix} w_{10}^2 & w_{11}^2 & \cdots & w_{1r}^2 \\ w_{20}^2 & w_{21}^2 & \cdots & w_{2r}^2 \\ \vdots & \vdots & \vdots & \vdots \\ w_{n0}^2 & w_{n1}^2 & \cdots & w_{nr}^2 \end{bmatrix} = \begin{bmatrix} (\boldsymbol{w}_1^2)^{\mathrm{T}} \\ (\boldsymbol{w}_2^2)^{\mathrm{T}} \\ \vdots \\ (\boldsymbol{w}_n^2)^{\mathrm{T}} \end{bmatrix} = \boldsymbol{w}^2 \tag{25}$$

$$\begin{bmatrix} w_{10}^3 & w_{12}^3 & \cdots & w_{1n}^3 \end{bmatrix}^{\mathrm{T}} = \boldsymbol{w}^3 \tag{26}$$

where w_{ij}^2 is the connection weight from the jth node in the input layer to the ith node in the hidden layer, and w_{ij}^1 is the connection weight from the jth node in the context layer to the ith node in the input layer.

For the M input samples $\{u^m, y^m\}$ $(m = 1, \ldots, M)$,

$$h\left(\widehat{\boldsymbol{\theta}}\,(k)\right) = \begin{bmatrix} \hat{y}^1 & \hat{y}^2 & \cdots & \hat{y}^M \end{bmatrix} \tag{27}$$

$$\begin{bmatrix} \hat{y}^1 & \hat{y}^2 & \cdots & \hat{y}^M \end{bmatrix} = \left(\boldsymbol{w}^3\right)^{\mathrm{T}} \begin{bmatrix} x_0^1 & x_0^2 & \cdots & x_0^M \\ x_1^1 & x_1^2 & \cdots & x_1^M \\ \vdots & \vdots & \vdots & \vdots \\ x_n^1 & x_n^2 & \cdots & x_n^M \end{bmatrix} \tag{28}$$

where $x_0^m = 1$ $(m = 1, \ldots, M)$, $w_{10}^3 = b_2$, $w_{i0}^2 = b_1$ $(i = 1, \ldots, n)$.

$$x_i^m(k) = f\left[\sum_{j=0}^{r} w_{ij}^2 u_j^m(k-1) + \sum_{l=1}^{n} w_{il}^1 x_{cl}^m(k)\right] \ (i = 1, \ldots, n) \tag{29}$$

$$x_c^m(k) = x^m(k-1) \tag{30}$$

$$x_i^m(k) = f\left[\sum_{j=0}^{r} w_{ij}^2 u_j^m(k-1) + \sum_{l=1}^{n} w_{il}^1 x_l^m(k-1)\right] \ (i = 1, \ldots, n) \tag{31}$$

$$\begin{aligned}\hat{y}^m(k) &= w_{10}^3 + w_{11}^3 f\left[\sum_{j=0}^{r} w_{1j}^2 u_j^m(k-1) + \sum_{l=1}^{n} w_{1l}^1 x_l^m(k-1)\right] \\ &\quad + w_{12}^3 f\left[\sum_{j=0}^{r} w_{2j}^2 u_j^m(k-1) + \sum_{l=1}^{n} w_{2l}^1 x_l^m(k-1)\right] \\ &\quad + \cdots + w_{1n}^3 f\left[\sum_{j=0}^{r} w_{nj}^2 u_j^m(k-1) + \sum_{l=1}^{n} w_{nl}^1 x_l^m(k-1)\right] \\ &= w_{10}^3 + \sum_{i=1}^{n} w_{1i}^3 f\left[\sum_{j=0}^{r} w_{1j}^2 u_j^m(k-1) + \sum_{l=1}^{n} w_{il}^1 x_l^m(k-1)\right]\end{aligned} \tag{32}$$

The system state vector $\widehat{\boldsymbol{\theta}}(k)$ is composed by weights and thresholds, in total $n(1 + r + n) + (n + 1)$ parameters of Elman neural network.

$$\widehat{\boldsymbol{\theta}}(k) = \left[\ (\boldsymbol{w}_1^1(k))^{\mathrm{T}},\ \ldots,\ (\boldsymbol{w}_n^1(k))^{\mathrm{T}},\ (\boldsymbol{w}_1^2(k))^{\mathrm{T}},\ \ldots,\ (\boldsymbol{w}_n^2(k))^{\mathrm{T}},\ (\boldsymbol{w}^3(k))^{\mathrm{T}}\ \right] \tag{33}$$

$$\boldsymbol{H}(k) = \left.\frac{\partial h[\boldsymbol{\theta}(k)]}{\partial \boldsymbol{\theta}(k)}\right|_{\widehat{\boldsymbol{\theta}}(k) = \boldsymbol{\theta}(k-1)} = \left[\ \boldsymbol{H}_1(k) \quad \boldsymbol{H}_2(k) \quad \boldsymbol{H}_3(k)\ \right]^{\mathrm{T}} \tag{34}$$

$$\boldsymbol{H}_1(k) = \left[\ \boldsymbol{H}_{11}(k) \quad \boldsymbol{H}_{12}(k) \quad \cdots \quad \boldsymbol{H}_{1n}(k)\ \right]^{\mathrm{T}} \tag{35}$$

$$\begin{aligned}\boldsymbol{H}_{1a}[b,c](k) &= w_{1a}^3 \cdot \lambda \cdot x_b^c(k-1) f\left[\sum_{j=0}^{r} w_{aj}^2 u_j^c(k-1) + \sum_{l=1}^{n} w_{al}^1 x_l^c(k-1)\right] \\ &\quad \cdot \left\{1 - f\left[\sum_{j=0}^{r} w_{aj}^2 u_j^c(k-1) + \sum_{l=1}^{n} w_{al}^1 x_l^c(k-1)\right]\right\}\end{aligned} \tag{36}$$

$(a = 1, \ldots, n;\ b = 1, \ldots, n;\ c = 1, \ldots, M)$

$$\boldsymbol{H}_2(k) = \left[\ \boldsymbol{H}_{21}(k) \quad \boldsymbol{H}_{22}(k) \quad \cdots \quad \boldsymbol{H}_{2n}(k)\ \right]^{\mathrm{T}} \tag{37}$$

$$\begin{aligned}\boldsymbol{H}_{2a}[b,c](k) &= w_{1a}^3 \cdot \lambda \cdot u_b^c(k-1) f\left[\sum_{j=0}^{r} w_{aj}^2 u_j^c(k-1) + \sum_{l=1}^{n} w_{al}^1 x_l^c(k-1)\right] \\ &\quad \cdot \left\{1 - f\left[\sum_{j=0}^{r} w_{aj}^2 u_j^c(k-1) + \sum_{l=1}^{n} w_{al}^1 x_l^c(k-1)\right]\right\}\end{aligned} \tag{38}$$

$(a = 1, \ldots, n;\ b = 0, \ldots, r;\ c = 1, \ldots, M)$

$$\boldsymbol{H}_3(k) = \begin{bmatrix} x_0^1(k) & x_0^2(k) & \cdots & x_0^M(k) \\ x_1^1(k) & x_1^2(k) & \cdots & x_1^M(k) \\ \vdots & \vdots & \vdots & \vdots \\ x_n^1(k) & x_n^2(k) & \cdots & x_n^M(k) \end{bmatrix} \tag{39}$$

5. **Simulation.** In this section, the proposed EKF-Elman is initialized and used to forecast the short-term power load. Two cases are used to demonstrate the performance of the proposed EKF-Elman.

(1) Case 1

This case is from [6]. As can be seen from [6], for the power load normalization data, the gradient descent method can achieve the best performance as the number of the neurons in the hidden layer equals 11. In the following, the proposed EKF-Elman is used to predict the power load normalization data and compare with the gradient descent based Elman neural network (GD-Elman). It has been proven from the theory that Elman neural network can realize the approximation for any nonlinear maps. Hence, this paper adopts the following network structure: 4 nodes in the input layer used to represent thresholds and actual power load, 11 nodes in the hidden layer determined by try and error, 1 node in the output layer used to stand for the forecasted power load. And the parameters are set as follows: $n = 11$, $b_1 = 0.96$, $b_2 = 5$, $\lambda = 1$, $P_0 = 40$, $Q_0 = 40$, $R_0 = 40$. The results forecasted by the proposed EKF-Elman and the GD-Elman are summarized in Table 1.

Table 1. Results obtained by EKF-Elman and GD-Elman for Case 1

O'clock	Actual power load	Algorithms	Forecasting results	Errors	Relative errors (%)
9	0.1432	GD-Elman	0.1648	0.0216	15.0645
		EKF-Elman	0.1432	-8.0050×10^{-4}	0.0590
10	0.5845	GD-Elman	0.6366	0.0521	8.9157
		EKF-Elman	0.5405	-0.0440	7.5297
11	0.7942	GD-Elman	0.7900	-0.0042	0.5335
		EKF-Elman	0.7939	-2.5521×10^{-4}	0.0321
Relative errors = |actual − forecasting|/actual × 100%					

Results in Table 1 indicate that the proposed EKF-Elman is more accurate in the prediction of the power load data than the gradient descent based Elman neural network.

(2) Case 2

The power load data from Nov. 24, 2014 to Nov. 30, 2014 are used to test the prediction performance of the gradient descent based Elman neural network and the proposed EKF-Elman. For the better use of the power load data, every 3-day power load data are used as the input vector while the 4th-day power load data are used as the target vector. Hence, 4 groups of sample data can be used to train the neural network. In the following, the 7th-day power load data are used as the test sample to verify whether the power load data can be predicted with accuracy. In order to improve the accuracy of the prediction, the parameter u is used to modify the prediction covariance matrix as

$$\boldsymbol{P}(k|k-1) = \left[\boldsymbol{I} - \boldsymbol{P}(k-1)\left(\boldsymbol{P}(k-1) + u^{-1}\boldsymbol{I}\right)\right]\boldsymbol{P}(k-1)$$

In this paper, the gradient descent based Elman neural network (GD-Elman), EKF-aided RBF neural network (EKF-RBF), and the proposed EKF-Elman are used to predict the power load data and make comparisons. The prediction results obtained by the above three methods are shown in Table 2.

In order to better show the prediction performance of the three methods, the error and the relative error between the prediction value and the true value are listed in Table 3. In the simulations, parameters are set as $n = 7$, $b_1 = 1$, $b_2 = 0.729$, $\lambda = 4.3$, $u = 0.101$, $P_0 = 40$, $Q_0 = 40$, $R_0 = 40$. To guarantee the input with larger values to be mapped by the transfer function with larger gradient, this paper uses the following (40) to normalize the input data while uses the following (41) to recover the data outputted in the output

Table 2. Results obtained by GD-Elman, EKF-RBF, and EKF-Elman for Case 2

O'clock	Actual power load	Algorithms	Forecasting results
9	975.73	GD-Elman	965.6329
		EKF-RBF	974.2866
		EKF-Elman	974.2612
10	975.94	GD-Elman	945.6105
		EKF-RBF	973.9631
		EKF-Elman	976.6603
11	973.4	GD-Elman	947.4539
		EKF-RBF	974.2615
		EKF-Elman	974.2612

Table 3. Errors and relative errors obtained by GD-Elman, EKF-RBF, and EKF-Elman for Case 2

O'clock	Algorithm	Errors	Relative errors
9	GD-Elman	−10.0971	1.0348
	EKF-RBF	−1.4434	0.14793
	EKF-Elman	−1.4688	0.15053
10	GD-Elman	−30.3295	3.1077
	EKF-RBF	−1.9769	0.20256
	EKF-Elman	0.72026	0.073802
11	GD-Elman	−25.9461	2.6655
	EKF-RBF	0.8615	0.088506
	EKF-Elman	0.8612	0.088477

layer.

$$y = \frac{x - x_{\min}}{x_{\max} - x_{\min}} \tag{40}$$

$$x = (1 - y)\, x_{\min} + y x_{\max} \tag{41}$$

where $x_{\max}$, $x_{\min}$ represent the maximum and the minimum in the sample set.

As can be seen from Table 2 and Table 3, both the EKF-RBF neural network and the proposed EKF-Elman neural network have much higher accuracy than the gradient descent based Elman neural network. On the one hand, the EKF-RBF neural network is similar to the proposed EKF-Elman neural network in the prediction accuracy at 9 o'clock and 11 o'clock. However, at 10 o'clock, the proposed EKF-Elman is superior to the EKF-RBF in the relative error. On the other hand, the average relative error obtained by the proposed EKF-Elman is the smallest among the three methods. Hence, the proposed EKF-Elman can obtain the higher accuracy than the EKF-RBF in the prediction of the power load data. The main reason that the proposed EKF-Elman is superior to both the gradient descent based neural network and the EKF-RBF neural network can be explained as follows. The proposed EKF-Elman combines the advantages of both EKF and Elman neural network. For one thing, EKF can make reliable estimation on the states of the nonlinear system as the statistic characteristics are well known. For another, Elman neural network has an inner feedback loop which can make the neural network have the ability to reflect the dynamic characteristics of a nonlinear system.

6. **Conclusions.** The proposed EKF-Elman neural network can combine the advantages of both EKF and Elman neural network and is more valid than both the gradient descent based Elman neural network and the EKF-aided RBF neural network in the prediction of the power load data. The simulations in the two cases indicate the superiority of the

proposed EKF-Elman neural network to both the gradient descent based Elman neural network and the EKF-RBF neural network.

Acknowledgment. This work is partially supported by the Young Backbone Teachers' Program of Heilongjiang Teaching Department (1254G065).

REFERENCES

[1] N. Liao, Z. Hu, Y. Ma and W. Lu, Review of the short-term load forecasting methods of electric power system, *Power System Protection and Control*, vol.39, no.1, pp.147-152, 2011.
[2] Z. Wei, Y. Deng and Z. Yang, Wind farm noise forecasting based on Elman neural network, *Water Resource and Power*, vol.33, no.5, pp.203-206, 2015.
[3] X. Dang, Z. Hao and J. Men, Traffic prediction of new chaotic elman neural network, *Computer Engineering*, vol.37, no.3, pp.172-174, 2015.
[4] S. Dan, Training radial basis neural networks with the extended Kalman filter, *Neurocomputing*, vol.48, pp.455-475, 2002.
[5] Y. Li, C. Wu, X. Ma, Z. Huang and H. Zhang, A recognition model for lane change intention based on neural network with EKF algorithm, *Journal of Wuhan University of Technology (Transportation Science & Engineering)*, vol.37, no.4, pp.843-847, 2013.
[6] J. Qian, C. Lv, W. Zhao and Q. Pan, Comparison of application on Elman and BP neural networks in discriminating water bursting source of coal mine, *Systems Engineering – Theory & Practice*, vol.30, no.1, pp.145-150, 2010.
[7] M. Malleswaran, V. Vaidehi and N. Sivasankari, A novel approach to the integration of GPS and INS using recurrent neural networks with evolutionary optimization techniques, *Aerospace Science and Technology*, vol.32, no.1, pp.169-179, 2014.
[8] X. Li, H. Chen, D. Li and J. Chen, Study on combat effectiveness prediction model using Elman feedback network, *Journal of System Simulation*, vol.27, no.1, pp.43-49, 2015.