文章编号：1000-3673（2010）04-0069-07　　中图分类号：TM 715　　文献标志码：A　　学科代码：470·4054

改进的 TSK 型动态模糊神经网络在短期负荷预测中的应用

杜鹃
（南洋理工大学，新加坡 南洋道 50 号 639798）

An Improved TSK-Type Dynamic Fuzzy Neural Network Approach for Short-Term Load Forecasting

DU Juan
(Nanyang Technological University, 50 Nanyang Avenue, Singapore 639798)

ABSTRACT: In this paper, an improved TSK-Type fuzzy neural network (FNN) is proposed for short-term load forecasting. The FNN is based on ellipsoidal basis function neurons consisting of a center vector and a width vector, and has the following features: structure identification and parameters estimation are performed automatically and simultaneously without partitioning the input space and selecting initial parameters in advance; fuzzy rules can be recruited or deleted dynamically during the learning process, and can be generated quickly without resorting to iteration algorithms. This growing and pruning fuzzy neural network (GPFNN) is simple and effective. It can not only reduce the complexity of the network but also accelerate the learning speed. The GPFNN is tested on the actual electric load data from EUNITE competition data. Results show that it provides the superior accuracy when applying in the short-term load forecasting.

KEY WORDS: dynamic fuzzy neural network; short-term load forecasting; ellipsoidal basis function; fuzzy rules; EUNITE competition data

摘要：将改进的 TSK 型模糊神经网络(fuzzy neural network，FNN)应用于短期负荷预测。该 FNN 由椭圆基函数构成神经元的中心和宽度参数，并且具有以下特征：网络结构和参数可自动并同时进行调整，不需提前分割输入空间，也不需提前选择网络初始参数；模糊规则在学习过程中可动态增删，不需采用迭代算法即可快速生成。这种模糊规则可动态增删的模糊神经网络(growing and pruning fuzzy neural network，GPFNN)简单有效，可以降低网络的复杂性，加快网络的学习速度。使用 EUNITE 竞赛数据作测试数据对上述 GPFNN 方法进行测试，结果表明采用该方法进行短期负荷预测时可获得较高的准确率。

关键词：动态模糊神经网络；短期负荷预测；椭圆基函数；模糊规则；EUNITE 竞赛数据

0　Introduction

Short-term electric load forecasting has become increasingly important and plays a crucial role in power systems[1-7]. Accurate short-term load forecasting can remarkably increase the operational efficiency of different areas of power systems, such as unit commitment, annual hydro-thermal maintenance scheduling, hydro-thermal coordination, demand side management, interchange evaluation, security assessment and others. Therefore, improvements in the accuracy of short-term load forecasting can result in significant financial savings for utilities and co-generators. Various forecasting techniques have been proposed in the last few decades. In the past, most algorithms applied in the load forecast problem relied on the statistical analysis, such as auto-regression[8] and time-series methods[9-10]. However, they are basically linear methods, and the load series they try to explain are known to be distinctly nonlinear functions of the exogenous variables[11-12]. The auto-regression model is not efficient in modeling the loads of weekends, holidays, and seasonal change periods. The time-series model must assume that the load is a stationary time series and has normal distribution characteristics. When the historical load data does not support this condition, the accuracy of the forecast is decreased. And the model becomes more complicated if other factors that influence the load are considered.

Recently, the artificial intelligence techniques has been carried out to handle load forecast problems. Several research groups have studied the application of artificial neural networks (ANN) in load forecasting[12-13]. ANN has the ability to learn and construct a complex nonlinear mapping through a set of input/output examples. It estimates a function without requiring a mathematical description of how the output functionally depends on the input. Fuzzy inference system (FIS) is an artificial intelligence technique for load forecasting[14]. FIS can offer a very powerful framework for approximate reasoning, as it attempts to simulate the human reasoning process at a cognitive level. Therefore, the artificial intelligence techniques are extremely suitable for load forecasting due to their capabilities in responding nonlinearity and general approximating. However, they still have some shortcomings to be resolved, such as low training speed, weaker searching capability for the overall optimal solution and difficulty in selecting the parameters and structure of the system.

An improved method was proposed to use the fuzzy neural network (FNN) to adjust the network structure and parameters. FNN has the advantages of both neural networks (such as learning abilities, optimization abilities, and connectionist structures) and FIS (such as human-like IF-THEN rules thinking and ease of incorporating expert knowledge)[15]. The fuzzy neural system produces appropriate improvement for load forecasting[12, 16-17]. In this paper, a growing and pruning Takago-Sugeno-Kang (TSK) type fuzzy neural network (GPFNN) for the short-term load forecasting is proposed. It is based on an ellipsoidal basis function (EBF) neural network, which is functionally equivalent to the TSK model based fuzzy system. When the forecasting model is developed with the actual electric load data, the proposed method provides the excellent forecasting accuracy.

This paper is organized as follows. In Section 1, the learning algorithm of GPFNN is introduced. Section 2 describes the load which is used for training the GPFNN method and the task of short-term load forecasting. In addition, results and observation of different models are showed in Section 2. Finally, Section 3 presents the conclusion.

1 Structure and algorithm of GPFNN

1.1 Structure of GPFNN

GPFNN is a multi-input multi-output system which has r inputs and s outputs. With the structure based on the EBF neural network, it is functionally equivalent to the TSK model based fuzzy system. The functions of various nodes in each of the four layers of GPFNN are depicted in Fig. 1.

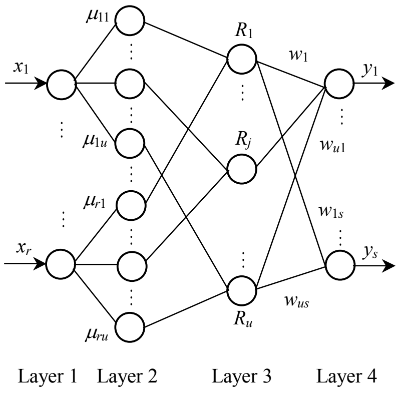


图 1 GPFNN 的结构
Fig. 1 Structure of the GPFNN

Each node in Layer 1 represents an input linguistic variable. And each node in Layer 2 represents a membership function, which is of the following form of a Gaussian function:

$$\mu_{ij}(x_i) = \exp[-(x_i - c_{ij})^2 /(2\sigma_{ij}^2)] \tag{1}$$

where i=1~r, j=1~u; μ_{ij} is the j th membership function of the i th input variable x_i; c_{ij} is the center of the j th Gaussian membership function of x_i; σ_{ij} is the width of the j th Gaussian membership function of x_i.

Each node in Layer 3 represents a possible IF-part for fuzzy rules. For the j th node, its output is given by

$$\phi_j(x_1, x_2, \cdots, x_r) = \exp[-\sum_{i=1}^{r}(x_i - c_{ij})^2 /(2\sigma_{ij}^2)], \quad j = 1 \sim u \tag{2}$$

The firing strength of each rule in Equation (2) can be regarded as a function of regularized Mahalanobis distance (M-distance). It is given by

$$\phi_j = \exp[-m^2(j)] \tag{3}$$

where

$$m(j)=\sqrt{(\boldsymbol{X}-\boldsymbol{C}_j)^{\mathrm{T}}\boldsymbol{A}(\boldsymbol{X}-\boldsymbol{C}_j)} \tag{4}$$

is the M-distance; $\boldsymbol{X}=[x_1,x_2,\cdots,x_r]^{\mathrm{T}}\in\mathbf{R}^r$; and $\boldsymbol{A}$ is defined by

$$\boldsymbol{A}=\begin{pmatrix} \frac{1}{2\sigma_{1j}^2} & 0 & \cdots & 0 \\ 0 & \frac{1}{2\sigma_{2j}^2} & \cdots & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & \cdots & \frac{1}{2\sigma_{rj}^2} \end{pmatrix},\quad j=1\sim u \tag{5}$$

Each node in Layer 4 represents the output variable as a weighted summation of incoming signals and is given by

$$y(x_1,x_2,\cdots,x_r)=\sum_{j=1}^{u}\boldsymbol{\omega}_j\phi_j \tag{6}$$

where y is the value of an output variable and $\boldsymbol{\omega}_j$ is the THEN-part (consequent parameters) or connection weight of the j th rule. For the TSK model, the consequents are the polynomials in the input variables and are given by

$$\omega_{ij}=\alpha_{0j}+\alpha_{1j}x_1+\cdots+\alpha_{rj}x_r,\quad j=1\sim u \tag{7}$$

Suppose that n fuzzy rules are generated for u observations. Equation (6) can be rewritten as follows in a more compact form

$$\boldsymbol{Y}=\boldsymbol{W\phi} \tag{8}$$

1.2 Learning algorithm of GPFNN

1.2.1 Adding a fuzzy rule

The learning algorithm of GPFNN can automatically add and remove fuzzy rules. When no new fuzzy rules are generated, only parameters are adjusted. GPFNN begins with no fuzzy rules. Two criteria are used in order to generate a new rule, namely system errors and ε-Completeness.

Output error of GPFNN with respective to the training data is an important factor in determining whether or not a new rule should be recruited. It can be described as follows:

For each observation $(\boldsymbol{X}_k,\boldsymbol{t}_k),k=1\sim n$ (where n is the number of total training sets; $\boldsymbol{X}_k$ is the input variable; $\boldsymbol{t}_k$ is the k th desired output), compute the overall GPFNN output $\boldsymbol{y}_k$ of the existing structure using Equations (6). Define the system error as

$$\|\boldsymbol{e}_k\|=\|\boldsymbol{t}_k-\boldsymbol{y}_k\| \tag{9}$$

If

$$\|\boldsymbol{e}_k\|>k_{\mathrm{e}} \tag{10}$$

a new rule should be considered. Here, k_{e} is a predefined threshold that decays during the learning process.

The second criterion is ε-Completeness of fuzzy rules. When an observation $(\boldsymbol{X}_k,\boldsymbol{t}_k),k=1\sim n$ enters the system, we calculate the M-distance between $\boldsymbol{X}_k$ and the center $\boldsymbol{C}_j(j=1\sim n)$ of the existing EBF neurons according to Equation (4) and (5). We find

$$J=\arg\min_{1\le j\le u}(m_k(j)) \tag{11}$$

If

$$m_{k,\min}=m_k(j)>k_{\mathrm{d}} \tag{12}$$

where k_{d} is a prespecified threshold that decays during the learning process, this implies that the existing system does not satisfied ε-Completeness and a new rule should be considered.

1.2.2 Parameter adjustment

A new Gaussian membership function is allocated whose width and center are set as follows:

$$c_{i(u+1)}=x_i^k \tag{13}$$

$$\sigma_i=\frac{\max\{|c_i-c_{i-1}|,|c_i-c_{i+1}|\}}{\sqrt{\ln(\sqrt{1/\varepsilon})}},\quad i=1\sim m \tag{14}$$

where c_{i-1} and c_{i+1} are the two centers of neighboring membership functions of i th membership function.

The width of Gaussian membership function is significant for its generalization. If the width is less than the distance between adjacent inputs, the membership function cannot cover the state space of inputs. If the width is too large, the output of system may severely deviate from the normal values. For the above reasons, the width must be carefully selected.

Suppose that u fuzzy rules are generated according to the two criteria stated above for n observation with r number of input variables. According to Equation (6), the following equation can be obtained:

$$\boldsymbol{W\phi}=\boldsymbol{Y} \tag{15}$$

where $\boldsymbol{W}\in\mathbf{R}^{u(r+1)}$; $\boldsymbol{\phi}\in\mathbf{R}^{u(r+1)\times n}$; $\boldsymbol{Y}\in\mathbf{R}^n$.

Assume that the desired output is $\boldsymbol{T}=[t_1,t_2,\cdots,t_n]\in\mathbf{R}^n$. The problem of determining the optimal parameters $\boldsymbol{W}^*$ can be formulated as a linear problem of minimizing $\|\boldsymbol{W\phi}-\boldsymbol{T}\|_2$, and $\boldsymbol{W}^*$ is determined by the pseudo-inverse technique as follows

$$W^* = T(\phi^{\mathrm{T}}\phi)^{-1}\phi^{\mathrm{T}} \tag{16}$$

1.2.3 Pruning a fuzzy rule

Hassibi and Storck developed the OBS method[18] in 1993, which can simultaneously cut connections between neurons and adjust weights according to the gradient of the error function. The OBS method identifies and deletes the weight that has the smallest effect on the neural network performance, and unlike other pruning methods, OBS also adjusts the remaining weights during the process of deleting a weight. In this paper, a modified OBS pruning method is utilized by correlating Hessian matrix with each hidden unit but not each weight. It relies on the sensitivity analysis of the weight parameter. The main idea is that the weight is critical, the slight change of the weight corresponds to a large change of the cost function.

Suppose that u fuzzy rules are generated according to the two criteria stated above for n observation with r number of input variables. According to Equation (6), the outputs of N nodes of the system can be given by

$$d(k) = \sum_{i=1}^{u} \boldsymbol{h}_i(k)\boldsymbol{\theta}_i(k) + \varepsilon(k) \tag{17}$$

The cost function of the system using a Taylor series is as follows

$$\Delta \boldsymbol{B} = \boldsymbol{B}(\boldsymbol{w}+\Delta\boldsymbol{w}) - \boldsymbol{B}(\boldsymbol{w}) = \left(\frac{\partial \boldsymbol{B}}{\partial \boldsymbol{w}}\right)^{\mathrm{T}} \Delta\boldsymbol{w} + \frac{1}{2}\Delta\boldsymbol{w}^{\mathrm{T}}\boldsymbol{H}\Delta\boldsymbol{w} + O(\|\Delta\boldsymbol{w}\|^3) \tag{18}$$

where $\Delta\boldsymbol{w}$ is the increase of weight; $\boldsymbol{H}$ is the Hessian matrix, its equation is as follows

$$\boldsymbol{H} \equiv \frac{\partial^2 \boldsymbol{B}}{\partial^2 \boldsymbol{w}} \tag{19}$$

In the learning process, when the network reaches a local minimum, the following equation can be obtained

$$\left(\frac{\partial \boldsymbol{B}}{\partial \boldsymbol{w}}\right)^{\mathrm{T}} \Delta\boldsymbol{w} = 0 \tag{20}$$

and all higher order terms can be ignored. Then Equation (18) can be simply approximated as

$$\Delta \boldsymbol{B} \approx \frac{1}{2}\Delta\boldsymbol{w}^{\mathrm{T}}\boldsymbol{H}\Delta\boldsymbol{w} \tag{21}$$

From Equation (17), the cost function is defined as the squared error as follows

$$\boldsymbol{B}(k) = \frac{1}{2}\sum_{i=1}^{n}[\boldsymbol{d}(i) - \boldsymbol{p}(i)^{\mathrm{T}}\boldsymbol{\theta}]^2 \tag{22}$$

where $\boldsymbol{p}(i)^{\mathrm{T}} = \boldsymbol{h}_i$; $\boldsymbol{\theta} = \boldsymbol{w}$. The Hessian matrix can be written in the following form

$$\frac{\partial^2 \boldsymbol{B}}{\partial^2 \boldsymbol{w}} = \boldsymbol{H} = \sum_{i=1}^{n} \boldsymbol{p}(i)\boldsymbol{p}^{\mathrm{T}}(i) \tag{23}$$

Because the dimension of Hessian matrix are equal to the number of hidden units in the network, the following equation exists

$$\bar{w}_i = \frac{\sum_{j=1}^{m} w_{ij}}{m} \tag{24}$$

where m is the number of weights which connect with the i th hidden unit. Reference [19] demonstrates that the higher this sum is, the more important that unit is to the network. The saliency of the i th unit is as follows

$$S_i = \frac{\bar{w}_i^2}{2[\boldsymbol{H}^{-1}]_{i,i}} \tag{25}$$

We can see from Equation (25) that the smaller the value of the saliency is, the less important the neuron is. And if

$$S_i < S_{\exp} \tag{26}$$

the i th neuron should be deleted. Here, $S_{\exp}$ is a predefined threshold.

The flowchart of the learning algorithm is illustrated in Fig. 2.

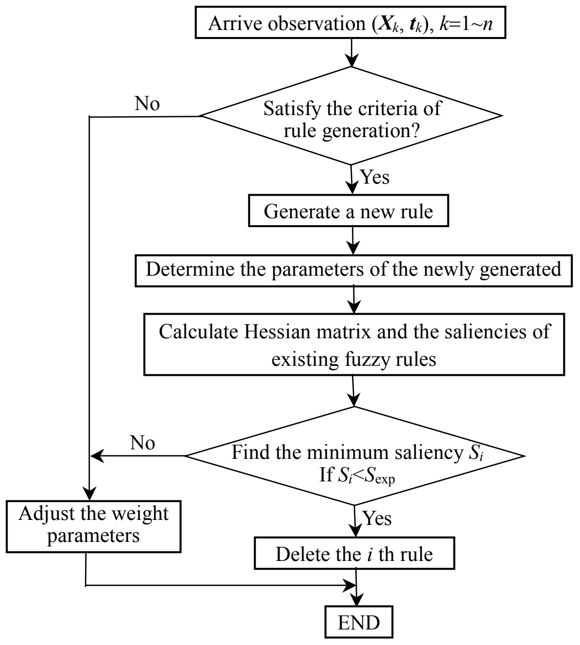

Arrive observation (X_k, t_k), k=1~n
Satisfy the criteria of rule generation?
No
Yes
Generate a new rule
Determine the parameters of the newly generated
Calculate Hessian matrix and the saliencies of existing fuzzy rules
Find the minimum saliency S_i If S_i<$S_{\exp}$
No
Yes
Adjust the weight parameters
Delete the i th rule
END

图 2 GPFNN 算法的学习流程

Fig. 2 Learning flowchart of GPFNN

2 Short-term load forecasting

2.1 Data and task description

To search for the proper parameters, we need to assess the performance of models. To do this, we usually divide the training data into two sets. One is used to train a model, while the other, called the validation set, is used for evaluating the model. According to the performance of models on the validation set, we try to infer the proper values of system parameters.

In this study, to evaluate the performance of the proposed load forecasting scheme, the GPFNN method was tested with data obtained from the well-known EUNITE competition data, to predict the maximum daily values of electric loads. The load data includes electric load demand recorded every half hour from January 1998 to January 1999. The training data used in this study covers the load data from January 1998 to December 1998. The task is to supply the prediction of maximum daily values of electric loads for each day of January 1999.

The assessment of the prediction performance of different models was performed by quantifying the prediction obtained from an independent data set. The mean percentage error (MPE), mean absolute percentage error (MAPE) and maximal error (ME) are used to evaluate the performance of the forecasting model. MPE, MAPE and ME are defined as follows

$$\sigma_{\mathrm{MPE},i}=\frac{L_{\mathrm{actual},i}-L_{\mathrm{predicted},i}}{L_{\mathrm{actual},i}}\times 100 \tag{27}$$

$$\sigma_{\mathrm{MAPE},i}=\frac{1}{N}\sum_{i=1}^{N}\frac{\left|L_{\mathrm{actual},i}-L_{\mathrm{predicted},i}\right|}{L_{\mathrm{actual},i}}\times 100 \tag{28}$$

$$\sigma_{\mathrm{ME},i}=\max\left|L_{\mathrm{actual},i}-L_{\mathrm{predicted},i}\right| \tag{29}$$

where $L_{\mathrm{actual},i}$ and $L_{\mathrm{predicted},i}$ are the real and the forecast data of maximum daily electric load on the i th day of the year 1999 respectively. N is the number of days in January 1999.

2.2 Simulation result

In this study, to forecast the peak load L_i of the i th day, the training input data include seven input variables: L_{i-1}, L_{i-2}, L_{i-3}, L_{i-4}, L_{i-5}, L_{i-6}, L_{i-7}. Therefore, the proposed GPFNN forecasting model consists of the seven-dimensional input and one-dimensional output. Before training the GPFNN model, there are some parameters, which influence the performance of the model, need to be chosen. In this paper, the following training parameters are chosen: $k_{\mathrm{e}}=0.995\,9$; $k_{\mathrm{d}}=1.024\,9$; $S_{\exp}=0.000\,003$.

Tab. 1 shows the comparison between actual load and forecasting load of each day in January 1999. Fig. 3 shows the GPFNN forecasting result. From Tab. 1, we can see that GPFNN has good performance in peak load forecasting.

Fig. 4 shows the comparison of the actual load, GPFNN forecasting load and radial basis function (RBF) forecasting load. The RBF neural network used in this paper is based on gradient descent learning

表 1 GPFNN 模型的预测值与实际值的比较

Tab. 1 Comparison between GPFNN forecasting load and actual load

Day	Actual Load/MW	Forecasting Load/MW	σ_{MPE}/%
1999-01-01	751	725.35	3.42
1999-01-02	703	727.98	−3.56
1999-01-03	677	702.38	−3.75
1999-01-04	718	723.47	-0.76
1999-01-05	738	732.83	0.70
1999-01-06	709	728.54	−2.76
1999-01-07	745	710.42	4.64
1999-01-08	749	752.90	−0.52
1999-01-09	734	728.14	0.79
1999-01-10	679	697.19	−2.68
1999-01-11	748	733.95	1.88
1999-01-12	739	751.28	−1.66
1999-01-13	756	726.82	3.86
1999-01-14	763	762.58	0.06
1999-01-15	752	765.16	−1.75
1999-01-16	738	739.04	−0.14
1999-01-17	699	697.82	0.17
1999-01-18	782	769.66	1.58
1999-01-19	782	773.63	1.070
1999-01-20	792	789.47	0.32
1999-01-21	801	789.39	1.45
1999-01-22	781	784.84	−0.49
1999-01-23	731	748.39	−2.38
1999-01-24	708	701.70	0.89
1999-01-25	789	790.33	−0.17
1999-01-26	798	786.03	1.50
1999-01-27	791	801.86	−1.37
1999-01-28	776	788.41	−1.60
1999-01-29	792	774.81	2.17
1999-01-30	763	757.83	0.68
1999-01-31	743	738.20	0.65

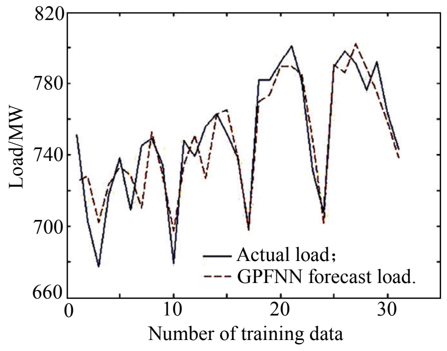


图 3　GPFNN 模型的预测曲线与实际曲线的比较

Fig. 3　Comparison between GPFNN forecasting curve and actual curve

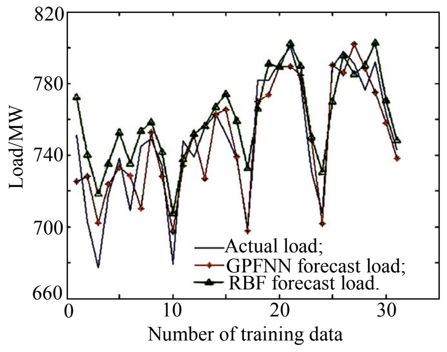


图 4　各种方法的预测值与实际值的比较

Fig. 4　Comparison of different forecasting loads and actual load

method. The number of hidden neurons is chosen as 10 in advance. The same training set and validation set are utilized for the RBF neural network. In Tab. 2, the ME and MAPE comparison are made between the RBF method and the GPFNN method. From Tab. 2, we can see that the GPFNN method has better performance in load forecasting.

表 2　GPFNN 模型与 RBF 模型预测结果的比较

Tab. 2　Result comparison between GPFNN model and RBF model

Model	σ_{ME}/MW	σ_{MPE}/%
RBF	82.6641	4.09
GPFNN	34.5820	1.59

3　Conclusion

In the proposed GPFNN method, the determination of the fuzzy rules and the adjustment of the premise and consequent parameters in fuzzy rules can be achieved simultaneously. In determination of the fuzzy rules, the rules can be added and removed automatically without predefining. In this paper, the GPFNN method is applied in the load forecasting, and is tested on the actual electric load data from EUNITE competition data. Results show the GPFNN method has better performance in short-term load forecasting.

Reference

[1]　唐杰明，刘俊勇，杨可，等．基于灰色模型和最小二乘支持向量机的电力短期负荷组合预测[J]．电网技术，2009，33(3)：63-68.
Tang Jieming，Liu Junyong，Yang Ke，et al．Short-term load combination forecasting by grey model and least square support vector machine[J]. Power System Technology，2009，33(3)：63-68(in Chinese).

[2]　李妮，江岳春，黄珊，等．基于累积式自回归动平均传递函数模型的短期负荷预测[J]．电网技术，2009，33(8)：93-97.
Li Ni，Jiang Yuechun，Huang Shan，et al. Short-term load forecasting based on ARIMA transfer function model[J]．Power System Technology，2009，33(8)：93-97(in Chinese).

[3]　刘旭，罗滇生，姚建刚，等．基于负荷分解和实时气象因素的短期负荷预测[J]．电网技术，2009，33(12)：94-100.
Liu Xu，Luo Diansheng，Yao Jiangang，et al．Short-term load forecasting based on load decomposition and hourly weather factors[J]．Power System Technology，2009，33(12)：94-100(in Chinese).

[4]　刘晓军，颜艳，张镝，等．基于量子神经网络的电力系统短期负荷预测[J]．电网技术，2009，33(S1)：181-184.
Liu Xiaojun，Yan Yan，Zhang Di，et al．Forecasting method for short-term load of power system based on quantum neural network[J]．Power System Technology，2009，33(S1)：181-184(in Chinese).

[5]　黎灿兵，刘梅，单业才，等．基于解耦机制的小地区短期负荷预测方法[J]．电网技术，2008，32(5)：87-90.
Li Canbing，Liu Mei，Shan Yecai，et al．Short-term load forecasting method of small region based on decoupling mechanism[J]．Power System Technology，2008，32(5)：87-90(in Chinese).

[6]　王德意，杨卓，杨国清．基于负荷混沌特性和最小二乘支持向量机的短期负荷预测[J]．电网技术，2008，32(7)：66-70.
Wang Deyi，Yang Zhuo，Yang Guoqing．Short-term load forecasting based on chaotic characteristic of loads and least squares support vector machines[J]. Power System Technology，2008，32(7)：66-70(in Chinese).

[7]　陈昊．基于不对称自回归条件异方差模型的短期负荷预测[J]．电网技术，2008，32(15)：84-89.
Chen Hao．Short-term load forecasting based on asymmetric autoregressive conditional heteroscedasticity models[J]．Power System Technology，2008，32(15)：84-89(in Chinese).

[8]　Vemuri S，Huang W L，Nelson D L．On-line algorithms for forecasting hourly loads of an electrical utility[J]．IEEE Trans on Power Apparatus and Systems，1981，100(8)：3775-3784.

[9]　Moghram I，Ruhman S．Analysis and evaluation five of load forecasting techniques[J]．IEEE Trans on Power Systems，1989，14(4)：1484-1491.

[10]　Hagan M T，Behr S M．The time series approach to short term load forecasting[J]．IEEE Trans on Power Systems，1987，2(3)：832-837.

[11]　Hippert H S，Pedreira C E，Souza R C. Neural networks for short-term

load forecasting: a review and evaluation[J]．IEEE Trans on Power Systems，2001，6(1)：44-55．

[12] Campbell P R J ． A hybrid modelling technique for load forecasting[C]．IEEE 2007 Electrical Power Conference，Montreal，Quebec，Canada，2007．

[13] Saini L M，Soni M K．Artificial neural network-based peak load forecasting using conjugate gradient methods[J]．IEEE Trans on Power Systems，2002，17(3)：907-912．

[14] Mori H，Kobayashi H．Optimal fuzzy inference for short-term load forecasting[J]．IEEE Trans on Power Systems，1996，11(1)：390-396．

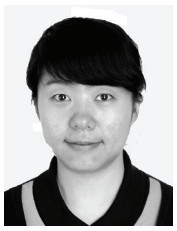
[15] Meneganti M，Saviello F，Tagliaferri R．Fuzzy neural networks for classification and detection of anomalies[J]．IEEE Trans on Neural Networks，1998(9)：848-861．

[16] Papadakis S E，Theocharis J B，Kiartzis S J，et al．A novel approach to short-term load forecasting using fuzzy neural networks[J]．IEEE Trans on Power Systems，1998，13(2)：480-492．

[17] Bakirtzis A G，Theocharis J B，Kiartzis S J，et al．Short term load forecasting using fuzzy neural networks[J]．IEEE Trans on Power Systems，1995，10(3)：1518-1524．

[18] Hassibi B，Stork D G．Second-order derivatives for network pruning: optimal brain surgeon[C]．Advances in Neural Information Processing，Los Altos，Morgan Kaufman CA，1993．

[19] Messer K，Kittler J．Choosing an optimal neural network size to aid search through a large image database[C]．Proceedings of the British Machine Vision Conference BMVC98，Southampton，United Kingdom，1998．


Du Juan

Received date: 2009-11-05.

Biographies:

Du Juan (1980—), female, a Ph.D student of the School of Electrical and Electronic Engineering, Nanyang Technological University. Her research interest focus on application of intelligent computation in power systems.

(编辑　董佳馨)