Journal of Southeast University (English Edition) Vol. 27, No. 2, pp. 159 – 163 June 2011 ISSN 1003—7985

# Prediction of coal ash fusion temperature using constructive-pruning hybrid method for RBF networks

Ding Weiming[1] Wu Xiaoli[1] Wei Haikun[2]

([1] School of Energy and Environment, Southeast University, Nanjing 210096, China)
([2] School of Automation, Southeast University, Nanjing 210096, China)

**Abstract:** A constructive-pruning hybrid method (CPHM) for radial basis function (RBF) networks is proposed to improve the prediction accuracy of ash fusion temperatures (AFT). The CPHM incorporates the advantages of the construction algorithm and the pruning algorithm of neural networks, and the training process of the CPHM is divided into two stages: rough tuning and fine tuning. In rough tuning, new hidden units are added to the current network until some performance index is satisfied. In fine tuning, the network structure and the model parameters are further adjusted. And, based on components of coal ash, a model using the CPHM is established to predict the AFT. The results show that the CPHM prediction model is characterized by its high precision, compact network structure, as well as strong generalization ability and robustness.

**Key words:** radial basis function (RBF) networks; function approximation; ash fusion temperature
**doi:** 10.3969/j.issn.1003 – 7985.2011.02.009

The AFT of coal directly relates to the thermal efficiency of steam boilers in coal-fired power stations[1]. Romero et al.[2–4] reported the fusibility of the ash as a function of the content of the eight principal oxides frequently found in coal ash, i.e., $SiO_2$, $Al_2O_3$, $Fe_2O_3$, CaO, MgO, $TiO_2$, $K_2O$ and $Na_2O$. And until now, many relational expressions about the AFT have been developed with a number of parameters which involve one or more chemical constituents of the coal ash. The regression formula is one of the widely used methods while there are less mature schemes. However, in many cases, more optimal results are obtained by neural networks[5].

Amoudi et al.[6–7] reported comparisons of back propagation (BP) and RBF neural networks and proved that RBF networks perform better than MLP networks using the BP algorithm in model building, prediction accuracy and network simplicity.

However, practical model performance of RBF networks heavily depends on the network structure. In general, minimal networks that satisfy training accuracy are preferred to ensure the generalization performance. In this paper, we propose a CPHM for designing RBF nets and apply it to establish a prediction model of the AFT. The CPHM combines the advantages of ROLS[8], RAN[9], and regularization[10], so the number and position of unit centers of the RBF net are both adapted in learning. The learning process of the CPHM can be divided into two stages: rough tuning and fine tuning. In rough tuning, hidden units are added to the current network using RAN until some performance index is met. In fine tuning, the fat of the network is pruned using regularization. In experiments, we show that RBF networks obtained with the CPHM not only generalize well, but also maintain satisfactory performance even when the training set is changed. And the prediction model of the AFT works well.

**Received** 2010-10-12.

**Biography:** Ding Weiming (1959—), male, associate professor, wmding@seu.edu.cn.

**Foundation items:** The National Natural Science Foundation of China (No. 60875035), the Natural Science Foundation of Jiangsu Province (No. BK2008294), the National High Technology Research and Development Program of China (863 Program) (No. 2006AA05A107).
**Citation:** Ding Weiming, Wu Xiaoli, Wei Haikun. Prediction of coal ash fusion temperature using constructive-pruning hybrid method for RBF networks[J]. Journal of Southeast University (English Edition), 2011, 27(2): 159 – 163. [doi: 10.3969/j.issn.1003 – 7985.2011.02.009]

## 1 Structure Definition

Considering an RBF net with $N$ inputs, $M$ hidden units, one output, and Gaussian radial basis functions (see Fig 1), the overall input-output transfer function of the network can be defined as

$$f(\boldsymbol{x}) = \sum_{i=1}^{M} w_i \varphi_{c_i}(\boldsymbol{x}) + b \tag{1}$$

where $\boldsymbol{x} \in \mathbf{R}^n$ and $f(\boldsymbol{x}) \in \mathbf{R}$ are the input and output of the network, $w_i$ denotes the weight from the $i$-th hidden unit to the output node, and $b$ is the bias term. $\varphi_{c_i}(\boldsymbol{x}) = e^{-\|\boldsymbol{x}-\boldsymbol{c}_i\|/r_i}$ is the response of hidden units, where $c_i \in \mathbf{R}^n$ is unit centers, $r_i$ is the spread of the $i$-th radial basis function.

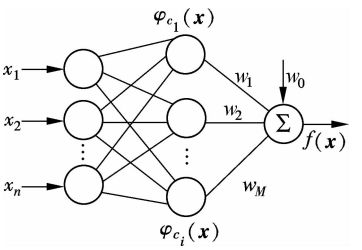


**Fig. 1** Structure of RBF net

The $N$ observations of the net are $\boldsymbol{T} = \{(\boldsymbol{X}_i, y_i) \mid i = 1, 2, \ldots, N\}$. We use $\boldsymbol{X} = [\boldsymbol{X}_1, \boldsymbol{X}_2, \ldots, \boldsymbol{X}_N]$ to denote observation input matrix, and the teacher signal is $\boldsymbol{Y} = [y_1, y_2, \ldots, y_N]^{\mathrm{T}}$. Thus, the output of the RBF net is

$$f(\boldsymbol{X}) = \boldsymbol{P}_M \boldsymbol{W} + \boldsymbol{B} = \sum_{i=1}^{M} w_i \boldsymbol{p}_i + \boldsymbol{B} \tag{2}$$

where $\boldsymbol{W} = [w_1, w_2, \ldots, w_M]^{\mathrm{T}}$ is the output weight vector, $\boldsymbol{P}_M = [\boldsymbol{p}_1, \boldsymbol{p}_2, \ldots, \boldsymbol{p}_M]$, $\boldsymbol{p}_i = [\Phi_{c_i}(X_1), \Phi_{c_i}(X_2), \ldots, \Phi_{c_i}(X_N)]^{\mathrm{T}}$; and $\boldsymbol{B}$ is the bias vecor. Generally, there is a deviation $\boldsymbol{e}$ between $\boldsymbol{f}(\boldsymbol{X})$ and $\boldsymbol{Y}$,

$$\boldsymbol{Y} = f(\boldsymbol{X}) + \boldsymbol{e} = \boldsymbol{P}_M \boldsymbol{W} + \boldsymbol{B} + \boldsymbol{e} = \tilde{\boldsymbol{P}}_M \tilde{\boldsymbol{W}} + \boldsymbol{e} \tag{3}$$

where $\tilde{\boldsymbol{P}}_M = [\boldsymbol{p}_1, \boldsymbol{p}_2, \ldots, \boldsymbol{p}_M, \boldsymbol{1}_N]$, $\boldsymbol{1}_N$ is an $N$-dimensional column vector with all its elements equal to 1, and $\tilde{\boldsymbol{W}} = [w_1, w_2, \ldots, w_M, b]^{\mathrm{T}}$.

## 2 Constructive-Pruning Hybrid Method

The training process of the CPHM is divided into two stages: rough tuning and fine tuning. In rough tuning, we add new hidden units to the current network. In fine tuning, we further use regularization to adjust the network structure and model parameters including unit number, unit centers, and output weights.

### 2.1 Rough tuning

The crux of rough tuning is how to select a new hidden unit and when to terminate this process.

If the current RBF net with $M$ hidden units cannot approximate the target function well, we add new hidden units to the RBF net. Consider adding one unit each time. Since we have no prior knowledge about the target function, we select the center of the new hidden unit among sample inputs, and then adjust it to an appropriate position.

From Eq. (3), if we consider $\boldsymbol{Y}$ as a vector with dimension $N$, then $\boldsymbol{Y}$ is a linear combination of $\boldsymbol{p}_i$'s and $\boldsymbol{1}_N$. Then $\{\boldsymbol{p}_1, \boldsymbol{p}_2, \ldots, \boldsymbol{p}_M, \boldsymbol{1}_N\}$ forms a basis in an $N$-dimensional vector space.

The cost function with Gaussian regularization is

$$E = (\boldsymbol{Y} - \tilde{\boldsymbol{P}}_M \tilde{\boldsymbol{W}})^{\mathrm{T}} (\boldsymbol{Y} - \tilde{\boldsymbol{P}}_M \tilde{\boldsymbol{W}}) + \lambda \tilde{\boldsymbol{W}}^{\mathrm{T}} \tilde{\boldsymbol{W}} \tag{4}$$

where $\lambda$ is the regular. Let $\partial E / \partial \tilde{W} = 0$, we obtain the minimum value of Eq. (4),

$$E_M = \boldsymbol{Y}^{\mathrm{T}} [\boldsymbol{I}_N - \tilde{\boldsymbol{P}}_M (\tilde{\boldsymbol{P}}_M^{\mathrm{T}} \tilde{\boldsymbol{P}}_M + \lambda \boldsymbol{I}_M)^{-1} \tilde{\boldsymbol{P}}_M^{\mathrm{T}}] \boldsymbol{Y} \tag{5}$$

Suppose that the newly selected unit center is $\boldsymbol{X}_i$. Let $\boldsymbol{s}_i = [\Phi_{c_i}(X_1), \Phi_{c_i}(X_2), \ldots, \Phi_{c_i}(X_N)]^{\mathrm{T}}$, $\boldsymbol{S} = [\boldsymbol{s}_1, \boldsymbol{s}_2, \ldots, \boldsymbol{s}_N]$, and $\tilde{\boldsymbol{P}}_{M+1} = [\tilde{\boldsymbol{P}}_M \quad \boldsymbol{s}_i]$. Thus, we should select $\boldsymbol{s}_j$ that satisfies

$$E_{M+1}(\boldsymbol{s}_j) = \min\{E_{M+1}(\boldsymbol{s}_i), i = 1, 2, \ldots, N\} \tag{6}$$

The first unit center is selected as the sample input $\boldsymbol{X}_i$ such that the relative $\boldsymbol{s}_j$ has the maximum projection on $\boldsymbol{Y}$. That is

$$E_1(\boldsymbol{X}_i) = \max\{\boldsymbol{Y}^{\mathrm{T}} \boldsymbol{s}_j, j = 1, 2, \ldots, N\} \tag{7}$$

As we know, the condition number of a matrix indicates its illness, so we can use the condition number of $\tilde{\boldsymbol{P}}_{M+1}^{\mathrm{T}} \tilde{\boldsymbol{P}}_{M+1}$ to decide when to stop rough tuning. Concretely, when the following equation is satisfied, we stop rough tuning.

$$C(\tilde{\boldsymbol{P}}_{M+1}^{\mathrm{T}} \tilde{\boldsymbol{P}}_{M+1}) > C_{\max} \tag{8}$$

where $C(\boldsymbol{A}) = \| \boldsymbol{A} \| \| \boldsymbol{A}^{-1} \|$ is the condition number of matrix $\boldsymbol{A}$, $\| \boldsymbol{A} \|$ is the Frobenius distance. $C_{\max}$ should be selected *a priori*. Empirically, it should not be greater than $10^6$, because most simulation software such as Matlab will give warning if $C_{\max}$ is greater than $10^6$.

### 2.2 Fine tuning

Fine tuning includes the tuning of unit centers and the output weights, and the pruning of redundant units.

Considering that the RBF net has a "local" property, we use those observations whose inputs are near the new unit center to tune its position. If $\boldsymbol{c}_i$ is the new unit center and $r_i$ is its spread, then the teacher samples are

$$A_i = \{(\boldsymbol{X}_j, y_j) \mid \| \boldsymbol{X}_j - \boldsymbol{c}_i \| < \kappa r_i \qquad j = 1, 2, \ldots, N\} \tag{9}$$

where $\kappa$ denotes the overlap coefficient. The greater $\kappa$ is, the more teacher samples are selected.

We use the least mean square algorithm to adjust the unit center, and then the modification to $\boldsymbol{c}_i$ is

$$\Delta \boldsymbol{c}_i(\boldsymbol{X}_j, y_j) = 4 \frac{\eta}{r_i} (\boldsymbol{X}_j - \boldsymbol{c}_i) \phi_{c_i}(\boldsymbol{X}_j) (y_j - f(\boldsymbol{X}_j)) w_i \tag{10}$$

where $\eta$ is the learning rate.

Each time we adjust the unit centers, and we should also adjust the output weights and bias. In fact, when all the unit centers are fixed, we can obtain the optimal output weights by minimizing Eq. (4). So we obtain

$$[w_1, w_2, \ldots, w_M, b]^{\mathrm{T}} = \tilde{\boldsymbol{W}}_M \tag{11}$$

In the CPHM, the cost function has a pruning property, some redundant output weights may decay to zeros during fine tuning. Thus, the corresponding hidden units can be deleted. For convenience, if the output weight $w_i$ meets the following condition the corresponding hidden unit can be pruned.

$$\mathrm{abs}(w_i) < w_{\min} \tag{12}$$

where $w_{\min}$ is the threshold weight. Besides, we use the rules proposed by Weigend[11] to adjust $\lambda$ by introducing the following errors and checking their relationships during learning.

## 3 Test of CPHM on Function Approximation Problem

Considering the approximation of the following polynomial function,

$$y(x) = \sin(3(x + 0.8)^2)$$

where $x \in \mathbf{R}$. The training set is generated as follows: The size is $N = 100$, with sample inputs $x_i$ uniformly generated from $[1, 1]$ and sample outputs $y(x_i) + e_i$. Here $e_i$ is a Gaussian noise with mean 0, variance 0.5.

Fig. 2(a) shows a typical training set, and Figs. 2(b), (c) and (d) show the results fitted by RAN, ROLS and CPHM, respectively. In Fig. 2, RAN generates 6 hidden units, with $\delta_{\max} = 1.6$, $\delta_{\min} = 0.3$, $\gamma = 0.977$, $\kappa = 0.87$, $\eta = 0.05$, and the learning epoch is 400; ROLS generates 25 hidden units, with parameters $\gamma = 0.2$ and $\lambda = 0.006$.

Comparatively, the CPHM generates 17 hidden units with the following parameters: $\gamma = 0.2$, $\kappa = 1$, $\lambda = 0.006$, $\Delta\lambda = 0.003$, $\mu = 0.95$, $\rho = 0.95$, $\eta = 10^{-4}$, $C_{\max} = 10^6$ and $\omega_{\min} = 0.1$. The target error is set to zero, and the learning epoch is 200.

Fig. 3 shows the history of training error, regular $\lambda$ and the number of hidden units in the CPHM. In rough tuning,

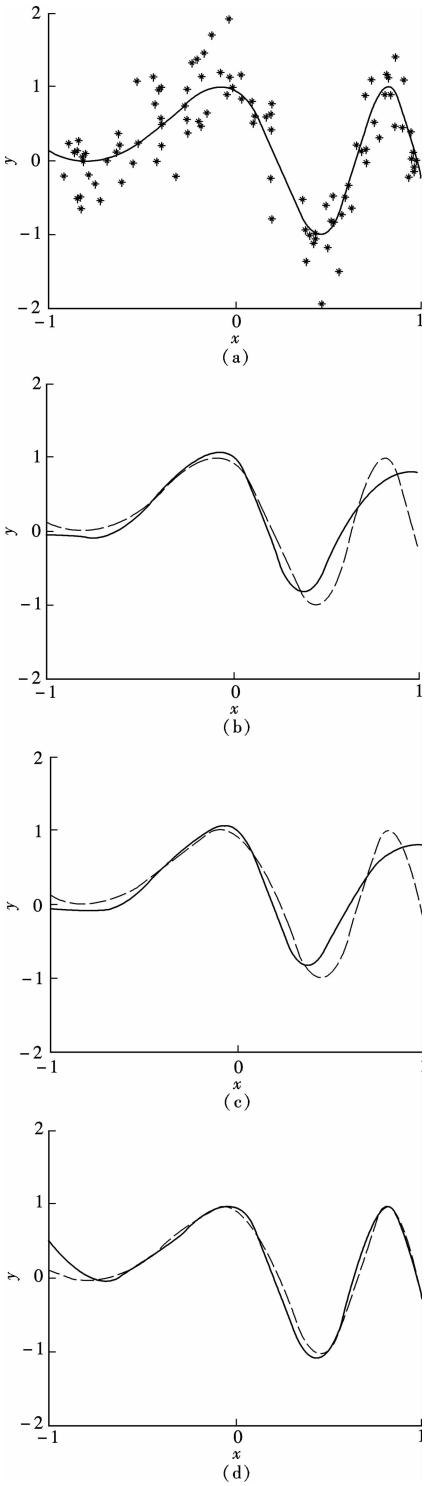


**Fig. 2** Training set and fitting curves of RAN, ROLS, and CPHM. (a) Target function and one example of the training sets; (b) Fitted by RAN; (c) Fitted by ROLS; (d) Fitted by CPHM

training error decreases rapidly with the increase in hidden units. When the number of hidden units reaches 24, the network reaches its condition number threshold, and rough tuning stops, and fine tuning begins. During fine tuning, the training error fluctuates slightly, but the hidden unit number decreases as $\lambda$ increases. At the 71th epoch, the hidden unit number is 18. After that, this number remains unchanged although $\lambda$ increases. This indicates that the fine tuning is able to adjust weights and simplify the network without increasing the training error.

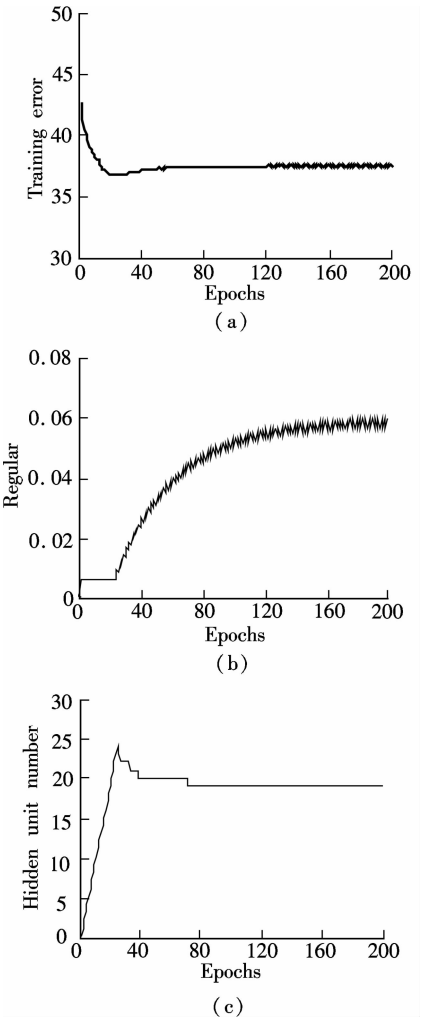


**Fig. 3** History of training error, $\lambda$ and number of hidden units. (a) Training error; (b) $\lambda$; (c) Hidden unit number

To further compare the performance of the three methods, we give plots of data error vs. fitting error for 100 training sets(see Fig. 4). Data error is defined as $\sqrt{\frac{1}{N}\sum_{1}^{N} e_i^2}$, and fitting error is $\sqrt{\frac{1}{100}\sum_{1}^{100}(G(\tilde{x}_i) - f(\tilde{x}_i))^2}$, which can be denoted as a generalization error of the RBF net. In Fig. 4, each data set corresponds to a point, and each data error is near 0.5, which is the variance of the noise.

Tab. 1 shows major performance of the RBF nets obtained by three methods on 100 and 40 different data sets. All the parameters are well chosen to obtain optimal performance.

From Fig. 4 and Tab. 1, we can find that although networks obtained by RAN have the compact structure, they do

**Tab. 1** Comparison of RBF nets obtained by three methods

| Items | 100 data sets | | | 40 data sets | |
|---|---|---|---|---|---|
| | RAN | ROLS | CPHM | ROLS | CPHM |
| Averaged hidden units | 8.35 | 21.6 | 16.1 | 27.5 | 18.6 |
| Variance of hidden units | 1.037 | 5.471 | 8.628 | 16.47 | 18.64 |
| Averaged data error | 0.496 | 0.499 | 0.501 | 0.496 | 0.489 |
| Averaged fit error | 0.265 | 0.172 | 0.157 | 0.327 | 0.231 |
| Variance of fit error | 0.008 | 0.001 | 0.001 | 0.016 | 0.003 |

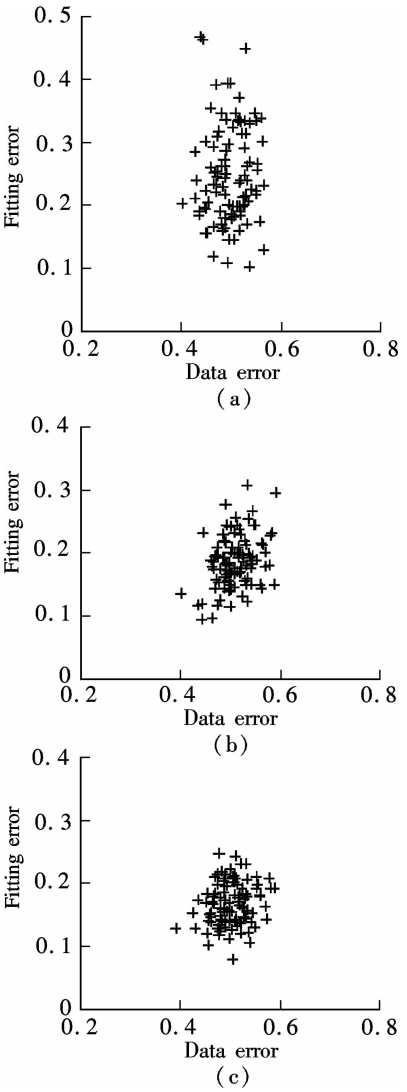


**Fig. 4** Data-fitting error plot of three methods when using 100 training samples. (a) RAN; (b) ROLS; (c) CPHM

not generalize so well as networks obtained by the CPHM and ROLS. In our simulation with RAN, we try to decrease $\delta_{\min}$ and $\gamma$, but the results become worse with dramatically increased hidden units and fitting error. Comparatively, RBF nets designed with the CPHM have both compact structure and low testing error, indicating that fine tuning of the CPHM not only has the simplified network structure, but also increases its generalization ability. When we use 100 samples in each data set, the testing error of the RBF nets obtained by ROLS is comparable to that of the CPHM, indicating that rough tuning of the CPHM is efficient. The variance of the fitting error of the CPHM is smaller than that of ROLS, indicating that fine tuning can increase generalization ability of RBF nets and robustness on different data sets.

In the above simulation, we use 100 samples in each data set. In this case, the bias between the original unit centers and their optimal values after fine tuning is small, so the fine tuning does not improve generalization so much. In the next experiment, we use 40 samples in each data set, and the results are shown in Tab. 1. It is obvious that the testing error of the CPHM is much lower than that of ROLS, which indicates that the CPHM performs better than ROLS when the size of the training set is small. The fitting error of the CPHM remains at 0.003, indicating that the CPHM maintains good robustness even when the data set is changed.

## 4 Prediction of Fusion Temperature of Coal Ash

A RBF net with eight inputs and one output is used to predict the fusion temperature of mixed coal, the eight inputs are the ingredients of the oxides $SiO_2$, $Al_2O_3$, $Fe_2O_3$, CaO, MgO, $TiO_2$, $K_2O$ and $Na_2O$, and the output is the fusion temperature. 155 samples are used for training, and 50 other samples are used for testing. All the samples are obtained by using chemical analysis in the field, and all the samples are normalized in the range of [0, 1].

The CPHM and RAN are used to design RBF nets. In the CPHM, the epoch is 145, spread constant $\gamma = 0.88$, $\lambda = 0.006$, $C_{\max} = 10^6$, $\Delta\lambda = 0.0002$, $\mu = 0.95$, the target error is set to zero, the learning rate $\eta = 10^{-8}$, and the threshold weight for pruning is $\omega_{\min} = 0.1$. In RAN, $\delta_{\max} = 2.6$, $\delta_{\min} = 0.8$, learning rate $\eta = 0.01$, $e_{\min} = 60$, and the epoch is 165.

Tab. 2 shows the results of two methods. Fig. 5 plots predicted fusion temperature vs. measured fusion temperature on testing samples and training samples. In Fig. 5, "+" is for testing samples, "∗" is for training samples, and the diagonal line indicates that the measured fusion temperature

**Tab. 2** Comparison of RBF nets obtained by CPHM and RAN methods

| Method | Training error/$10^5$ | Testing error/$10^5$ | Hidden units |
|---|---|---|---|
| RAN | 5.541 | 1.748 | 10 |
| CPHM | 4.604 | 1.436 | 4 |

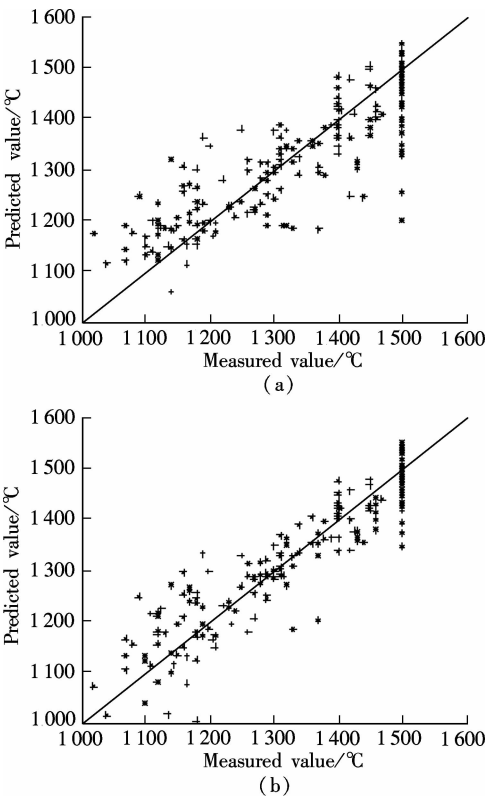


**Fig. 5** Test fusion temperature of RBF nets obtained by CPHM and RAN. (a) RAN; (b) CPHM

is equal to the predicted fusion temperature.

Compared with RAN, we can find from Tab. 2 and Fig. 5 that the RBF net obtained with the CPHM has not only small training and testing errors, but also a more compact network structure.

## 5 Conclusion

In this paper, an RBF model is developed for the prediction of the ash fusion temperature from ash composition using the constructive-pruning hybrid method. And the trained model can always achieve much better results than traditional RAN and ROLS nets. With the CPHM, we adjust both the network structure parameters and the model parameters in learning. Tests show that networks designed with the CPHM have both compact structure and better generalization ability. And, the CPHM method maintains good performance even when training set is changed. That is to say, designed networks have a low variance of fitting error with different training data sets.

## References

[1] van Dyk J C, Keyser M J, van Zyl J W. Suitability of feedstocks for the Sasol-Lurgi fixed bed dry bottom gasification process [C]//*Gasification Technologies Conference*. San Francisco, USA, 2001: 1 - 11.

[2] Romero C E, Saro R D, Craparo J, et al. Laser-induced breakdown spectroscopy for coal characterization and assessing slagging propensity [J]. *Energy Fuels*, 2010, **24**(1): 510 - 517.

[3] Song W J, Tang L H, Zhu X D, et al. Effect of coal ash composition on ash fusion temperatures [J]. *Energy Fuels*, 2010, **24**(1): 182 - 189.

[4] Dunnu G, Maier J, Scheffknecht G. Ash fusibility and compositional data of solid recovered fuels[J]. *Fuel*, 2010, **89**(7): 1534 - 1540.

[5] Kalogirou S A. Artificial intelligence for the modeling and control of combustion processes[J]. *Progress in Energy and Combustion Seience*, 2003, **29**(6): 515 - 566.

[6] Amoudi A, Zhang L. Application of radial basis function networks for solar array modeling and maximum power-point prediction[J]. *IEE Proceedings*: *Generation, Transmission and Distribution*, 2000, **147**(5): 310 - 316.

[7] Chandok J S, Kar I N, Tuli Suneet. Estimation of furnace exit gas temperature(FEGT) using optimized radial basis and back-propagation neural networks [J]. *Energy Conversion and Management*, 2008, **49**(8): 1989 - 1998.

[8] Orr M. Regularization in the selection of radial basis function centers[J]. *Neural Computation*, 1995, **7**(3): 606 - 623.

[9] Plat J. A resource-allocating network for function interpolation[J]. *Neural Computation*, 1991, **3**(2): 213 - 225.

[10] Chen S. Local regularization assisted orthogonal least squares regression[J]. *Neurocomputing*, 2006, **69**(4/5/6): 559 - 585.

[11] Weigend A, Rumelhart D, Huberman B. Generalization by weight-elimination with application to forecasting[C]//*Advances in Neural Information Processing Systems*. Cambridge, MA, USA: The MIT Press, 1991, **3**: 875 - 882.

# 基于构造-剪枝混合优化 RBF 网络的煤灰熔点预测方法

丁维明[1] 吴小丽[1] 魏海坤[2]

([1] 东南大学能源与环境学院,南京 210096)

([2] 东南大学自动化学院,南京 210096)

**摘要**:为提高煤灰熔点的预测精度,提出了一种基于构造-剪枝混合优化 RBF 网络的煤灰熔点预测方法. 该方法融合了神经网络构造算法和剪枝算法的优点,分为"粗调"和"精调"2 个阶段. 粗调阶段动态增加隐节点数目直至满足相应的停止准则;精调阶段对粗调得到的 RBF 网络的结构和参数作进一步调整. 基于煤灰的化学组成成分建立相应的构造-剪枝混合优化 RBF 网络预测煤灰熔点. 预测结果表明:所建模型在具有较高精度的同时,具有较小的结构、较好的泛化能力和较强的鲁棒性.

**关键词**:RBF 网络;函数逼近;煤灰熔点

**中图分类号**:TQ520.62