

Kernel Width Optimization for Faulty RBF Neural Networks with Multi-node Open Fault

Hong-Jiang Wang · Chi-Sing Leung · Pui-Fai Sum ·
Gang Wei

© Springer Science+Business Media, LLC. 2010

Abstract Many researches have been devoted to select the kernel parameters, including the centers, kernel width and weights, for fault-free radial basis function (RBF) neural networks. However, most are concerned with the centers and weights identification, and fewer focus on the kernel width selection. Moreover, to our knowledge, almost no literature has proposed the effective and applied method to select the optimal kernel width for faulty RBF neural networks. As is known that the node faults inevitably take place in real applications, which results in a great many of faulty networks, it will take a lot of time to calculate the mean prediction error (MPE) for the traditional method, i.e., the test set method. Thus, the letter derives a formula to estimate the MPE of each candidate width value and then use it to select the optimal one with the lowest MPE value for faulty RBF neural networks with multi-node open fault. Simulation results show that the chosen optimal kernel width by our proposed MPE formula is very close to the actual one by the conventional method. Moreover, our proposed MPE formula outperforms other selection methods used for fault-free neural networks.

Keywords Faulty neural networks · Multi-node open fault · Radial basis function · Kernel width · Mean prediction error

H.-J. Wang (✉)
South China Normal University, Tianhe, Guangzhou, China
e-mail: eehjwang@hotmail.com

H.-J. Wang · C.-S. Leung
City University of Hong Kong, Kowloon, Hong Kong

P.-F. Sum
National Chung Hsing University, Taichung, Taiwan

H.-J. Wang · G. Wei
South China University of Technology, Tianhe, Guangzhou, China

1 Introduction

RBF neural networks have been widely researched and applied in many fields, such as function approximation, time series prediction and classification and so on [1]. To gain the best learning performance, it is necessary to select the proper parameters for RBF neural networks, i.e., center, kernel width and weights [2].

So far, many literatures have been devoted to determine the centers and weights [3–5], but few only focus on the kernel width selection, especially for faulty RBF neural networks. The existing kernel width optimization methods are closely relative to the centers identification process. The most simple method to determine the optimal kernel width is the definition, i.e., proportional to the maximum distance between the centers [6], which is only fit for the uniform distribution of the centers. In addition, Moody and Darken [7] select the kernel width for each RBF node using the r -nearest neighbors heuristic, those process identifying each kernel width is so complex that it is hardly applied, especially when the size of neural networks is large. Moreover, some stochastic algorithms have been proposed to evaluate the optimal kernel width in the past years as well, e.g., the EM algorithms [8]. However, stochastic algorithms require a large number of samples, which does not well work in the situations of the sparse sample space.

Noted that the samples are very valuable so that a small number of samples can be gotten and used to train and test the networks in the real application. Thus, an exhaustive search by computing the prediction error for the candidate kernel width values is the best method in order to optimize the RBF kernel width value. Furthermore, to improve the search speed, some rapid search algorithms have also been proposed, which take the tradeoff between the accuracy and complexity [9].

However, all the existing methods of the kernel width optimization are concerning fault-free RBF neural networks. As is known that the neural node faults unavoidably occur in the real application, especially in VLSI [10,11]. The multi-node open fault is one of the most popular fault models, where several hidden nodes and their associated weights in a neural network are out of work at the same time [12,13]. Due to many potential faulty models in the multi-node open fault situations, it will take a lot of time for the conventional selection method, i.e., the test set method, to calculate the prediction error of each faulty model and then take the mean value [14]. Thus, it is necessary to design a formula to rapidly evaluate the mean prediction error (MPE) value. Moody has suggested that the mean prediction error can be expressed as the sum of the training error and other terms and used to select the architecture for neural networks [15]. The notion has been used for studying the generalization ability of fault-free feedforward neural networks. However, to our knowledge, no literature on the kernel width optimization for faulty neural networks with multi-node open fault has been published. Therefore, in the letter, we follow Moody's thinking and derive a formula to evaluate the mean prediction error for the faulty RBF neural networks and then use it to select the optimal kernel width.

In addition, due to no intrinsic fault tolerant for neural networks, some extra methods must be employed to improve the generalization performance [12]. Regularization methods have been recognized as one of the most effective approaches to improve the fault tolerant [3,16,17]. Leung and Sum have proposed one regularizer [18], which has the better ability to tolerate the multi-node open fault than other regularizers, e.g., weight decay methods. Therefore, we will derive the MPE formula for faulty RBF neural networks with the Leung's regularizer in the multi-node open fault situations and then use it to choose the optimal kernel width, i.e., estimating the mean prediction error values for all the candidate kernel width and then taking the kernel width with the minimal mean prediction error value as the optimal one.

In the rest of the letter, we first will introduce the background acknowledge, including data model, RBF network, multi-node open fault model and Leung’s regularizer. Then, we derive the MPE formula based on the Leung’s regularizer for faulty RBF neural networks with multi-node open fault. At last, some simulations have been made to verify our theoretical results.

2 Background

2.1 Data Model

Throughout the paper, we consider the following data model,

$$\mathcal{D} = \left\{ (x_i, y_i) : x_i \in \mathcal{R}^K, y_i \in \mathcal{R}, i = 1, 2, \dots, N. \right\} \tag{1}$$

where x_i and y_i are the input and output samples of an unknown system $f(\cdot)$, respectively, i.e.,

$$y_i = f(x_i) + e_i \tag{2}$$

where e_i ’s are the random noise with zero-mean and variance σ_e^2 .

2.2 RBF Model

In the RBF approach, the mapping $f(\cdot)$ is approximated by an RBF network, given by

$$f(\mathbf{x}) \approx \Phi^T(\mathbf{x}) \mathbf{w} \tag{3}$$

where $\mathbf{x} \in \mathcal{R}^K$ is the input, $\mathbf{w} = [w_1, w_2, \dots, w_M]^T$ is the RBF weight vector, $\Phi(\mathbf{x}) = [\phi_1(\mathbf{x}), \phi_2(\mathbf{x}), \dots, \phi_M(\mathbf{x})]^T$ is the RBF kernel function vector and the j th kernel function $\phi_j(\mathbf{x})$ is given by

$$\phi_j(\mathbf{x}) = \exp\left(-\frac{\|\mathbf{x} - c_j\|^2}{\sigma_j}\right) \tag{4}$$

where c_j ’s are the centers, σ_j ’s (> 0) are the width of the RBF kernel function, $\|\cdot\|$ is the Euclidean matrix norm. For simplicity, we assume that all the kernel width values in the RBF functions are identical, i.e., $\sigma_j = \sigma$, in the letter.

2.3 Multi-node Open Fault

The multi-node open fault is an important fault model where some hidden nodes are disconnected to the output layer, which can be described by a weight multiplicative model, given by

$$\tilde{\mathbf{w}} = \mathbf{b} \otimes \mathbf{w} \tag{5}$$

where \otimes is the element-wise multiplication operator, $\mathbf{b} = [b_1, \dots, b_M]^T$.

The fault factor b_i denotes whether the i th node operates properly or not. That is to say, when $b_i = 1$, the i th node operates properly and when $b_i = 0$, it is out of work.

For a fault-free network, the training error is

$$\mathcal{E}_{train}(\mathbf{w}) = \frac{1}{N} \sum_{i=1}^N \left(y_i - \Phi^T(x_i) \mathbf{w} \right)^2 \tag{6}$$

For a faulty network, the training error is

$$\mathcal{E}_{train}(\mathbf{w}, \mathbf{b}) = \frac{1}{N} \sum_{i=1}^N \left(y_i - \Phi^T(x_i) \tilde{\mathbf{w}} \right)^2 \tag{7}$$

2.4 Leung’s Regularizer

To improve the fault tolerance, some especial cares must be taken for faulty neural networks [19,20]. Regularization methods have been recognized as one of the most simple and effective fault tolerant techniques. Leung and Sum [18] have proposed a regularizer to tolerate the multi-node open fault based on the Kullback-Leibler divergence, i.e.,

$$p\mathbf{w}^T \mathbf{R} \mathbf{w} \tag{8}$$

where $p(> 0)$ is the regularization parameter and the node fault rate as well, $\mathbf{R} = \mathbf{G} - \mathbf{H}_\phi$ is the regularization matrix, $\mathbf{H}_\phi = \frac{1}{N} \sum_{i=1}^N \Phi(x_i) \Phi^T(x_i)$, $\mathbf{G} = \text{Diag}(\mathbf{H}_\phi)$. Experiments have shown that the proposed regularizer has the better ability to tolerate the multi-node open fault comparing with other conventional approaches, e.g., weight decay based regularizer. Therefore, in the letter, we derive the MPE formula based on the Leung’s regularizer for faulty RBF neural networks with multi-node open fault. The objective function is

$$\mathcal{J}(\mathbf{w}, p) = \frac{1}{N} \sum_{i=1}^N (y_i - f(x_i))^2 + p\mathbf{w}^T \mathbf{R} \mathbf{w} \tag{9}$$

and the optimal weight vector is

$$\hat{\mathbf{w}} = (\mathbf{H}_\phi + p\mathbf{R})^{-1} \frac{1}{N} \sum_{i=1}^N \Phi(x_i) y_i \tag{10}$$

3 MPE Formula

According to [15], the mean testing error for RBF neural networks can be expressed as the sum of the training error and other additive terms. Therefore, The MPE formula for faulty neural networks can be expressed as

$$\bar{\mathcal{E}}_{test}(\hat{\mathbf{w}}, \mathbf{b}) = \bar{\mathcal{E}}_{train}(\hat{\mathbf{w}}, \mathbf{b}) + 2S_e \frac{M_{eff}}{N} \tag{11}$$

where $\bar{\mathcal{E}}_{train}(\hat{\mathbf{w}}, \mathbf{b})$ is the mean training error for faulty networks, S_e is the measured noise variance and M_{eff} is the Moody’s effective number of parameters in the nonlinear model, which can be defined as

$$M_{eff} = \frac{N}{2} \sum_{i=1}^N \sum_{\alpha=1}^M \sum_{\beta=1}^M T_{i\alpha} U_{\alpha\beta}^{-1} T_{\beta i} \tag{12}$$

Combining the Eqs. 9 and 6, we can get

$$T_{i\alpha} = \frac{\partial^2 \mathcal{E}_{train}(\hat{\mathbf{w}})}{\partial y_i \partial w_\alpha} = -\frac{2}{N} \phi_\alpha(x_i) \tag{13}$$

$$T_{\beta i} = \frac{\partial^2 \mathcal{E}_{train}(\hat{\mathbf{w}})}{\partial w_\beta \partial y_i} = -\frac{2}{N} \phi_\beta(x_i) \tag{14}$$

$$U_{\alpha\beta} = \frac{\partial^2 \mathcal{J}(\hat{\mathbf{w}}, p)}{\partial w_\alpha \partial w_\beta} = 2(\mathbf{H}_\phi + p\mathbf{R}) \tag{15}$$

Therefore, we have

$$M_{eff} = \mathbf{TR}(\mathbf{H}_\phi(\mathbf{H}_\phi + p\mathbf{R})^{-1}) \tag{16}$$

where **TR** denotes the trace operator.

In addition, the averaged train error for faulty neural networks $\bar{\mathcal{E}}_{train}(\hat{\mathbf{w}}, \mathbf{b})$ can be expressed as

$$\begin{aligned} \bar{\mathcal{E}}_{train}(\hat{\mathbf{w}}, \mathbf{b}) &= \left\langle \frac{1}{N} \sum_{i=1}^N (y_i - \Phi^T(x_i) \hat{\mathbf{w}})^2 \right\rangle \\ &= \left\langle \frac{1}{N} \sum_{i=1}^N (y_i - \Phi^T(x_i) (\mathbf{b} \otimes \hat{\mathbf{w}}))^2 \right\rangle \\ &= \frac{1}{N} \sum_{i=1}^N y_i^2 + (1-p) \hat{\mathbf{w}}^T (\mathbf{H}_\phi + p\mathbf{R}) \hat{\mathbf{w}} \\ &\quad - 2(1-p) \cdot \frac{1}{N} \sum_{i=1}^N (y_i \cdot \Phi^T(x_i) \hat{\mathbf{w}}) \end{aligned} \tag{17}$$

where $\forall i, j, \langle b_i b_j \rangle = 1 - p$ and $\langle b_i b_j \rangle = \begin{cases} 1 - p, & i = j, \\ (1 - p)^2, & i \neq j. \end{cases}$

Moreover, the training error in the Eq. 6 for fault-free neural networks $\mathcal{E}_{train}(\hat{\mathbf{w}})$ can be re-expressed as

$$\mathcal{E}_{train}(\hat{\mathbf{w}}) = \frac{1}{N} \sum_{i=1}^N y_i^2 - 2 \cdot \frac{1}{N} \sum_{i=1}^N (y_i \cdot \Phi^T(x_i) \hat{\mathbf{w}}) + \hat{\mathbf{w}}^T \mathbf{H}_\phi \hat{\mathbf{w}} \tag{18}$$

Generally, Eq. 18 can be used to evaluate the prediction error for the fault-free neural networks [21].

Subtracting (18) from (17), we can get

$$\bar{\mathcal{E}}_{train}(\hat{\mathbf{w}}, \mathbf{b}) = (1-p) \mathcal{E}_{train}(\hat{\mathbf{w}}) + p \cdot \frac{1}{N} \sum_{i=1}^N y_i^2 + (p-p^2) \hat{\mathbf{w}}^T \mathbf{R} \hat{\mathbf{w}} \tag{19}$$

Thereby, according to the Eqs. 16 and 19, the Eq. 11 can be transformed as

$$\begin{aligned} MPE = \bar{\mathcal{E}}_{test}(\hat{\mathbf{w}}, \mathbf{b}) &= (1-p) \mathcal{E}_{train}(\hat{\mathbf{w}}) + p \cdot \frac{1}{N} \sum_{i=1}^N y_i^2 \\ &\quad + (p-p^2) \hat{\mathbf{w}}^T \mathbf{R} \hat{\mathbf{w}} + 2 \frac{S_e}{N} \mathbf{TR}(\mathbf{H}_\phi(\mathbf{H}_\phi + p\mathbf{R})^{-1}) \end{aligned} \tag{20}$$

In addition, according to the Fedorov's method [22], the measured noise variance can be given by

$$S_e = \frac{1}{N - M_{eff}} \sum_{i=1}^N \left(y_i - \Phi^T(x_i) \hat{\mathbf{w}} \right)^2 \quad (21)$$

From the mean prediction error expression for faulty RBF neural networks, i.e., the Eq. 20, we can see that our proposed MPE formula can be comprised of the training error for fault-free neural networks and other terms.

4 Simulation Results

To verify our theoretical results, we test two problems, i.e., regression and time series prediction. The candidate kernel width values are set as $\sigma = [10^{-2}, 10^{-1.95}, \dots, 10^1]$. Moreover, to calculate the actual testing error for all the simulations, 10,000 faulty networks are randomly generated.

4.1 Sinc Function Approximation

The sinc function approximation example is a common benchmark regression problem [23], which can be expressed as

$$y = \text{sinc}(x) + e \quad (22)$$

where e is a mean zero Gaussian signal noise with variance $\sigma_e^2 = 0.01$. We produce 200 samples for the training dataset and 1,000 samples for the testing dataset. Figure 1 has given out the training data for the sinc function. The optimal weight vector can be obtained using

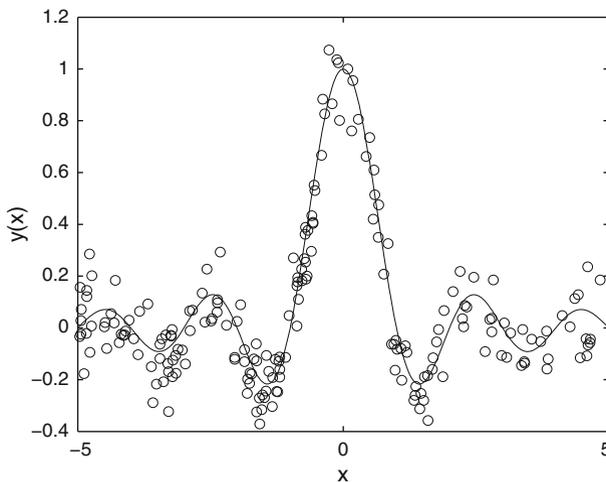


Fig. 1 Training data for the sinc function

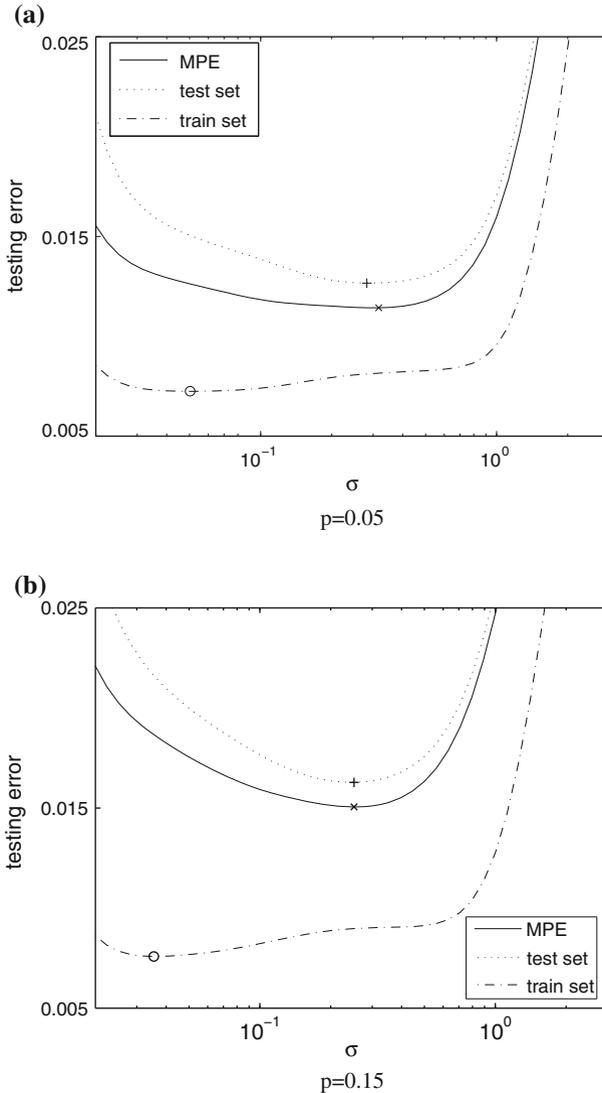


Fig. 2 The relation between the testing error and the candidate kernel width values with three measured methods, i.e., MPE formula, test set method and train set method, for sinc function approximation example with the different fault rate p . The signs, 'x', '+' and 'o', denote the lowest point on the curve of MPE formula, test set method and train set method, respectively

the 200 training dataset according to the Eq. 10 and the 37 RBF's centers are selected as $\{-4.5, -4.25, \dots, 4.25, 4.5\}$.

Figure 2 has shown testing error of the different candidate kernel width values for sinc function approximation example with three kinds of selection methods, i.e., MPE formula, test set method and train set method. The test set method means to use the testing dataset to calculate the prediction error (testing error) for each faulty network and then take the mean

Table 1 Comparison of different kernel width optimization methods, i.e., test set method [14], our MPE formula, MSE formula [9] and formula $d_{max}/\sqrt{2M}$ [6]

σ	SINC		MGTS	
	$p = 0.05$	$p = 0.15$	$p = 0.05$	$p = 0.15$
test set (actual)	0.281838	0.262125	0.501187	0.501062
MPE (ours)	0.316228	0.261532	0.501187	0.501062
MSE (fault-free)	0.050119	0.035323	0.1	0.055102
$d_{max}/\sqrt{2M}$		0.029062		0.317265

value, i.e., the actual mean prediction error. The results for all the candidate kernel width are shown with the dotted line in the figure. And the train set method means to use the mean square error (MSE) formula, i.e., the Eq. 18, to calculate the prediction error, which can be taken as the kernel width selection method for fault-free RBF neural networks [9], those results are denoted by the dash-dot line. The MPE formula means to use our proposed Eq. 20 to evaluate the prediction error, as shown with the solid line. In addition, for verifying our theoretical results, we have marked the optimal kernel width values with the denotations ‘+’, ‘o’ and ‘x’ in the curves for three methods, respectively.

The comparison of the testing error for the MPE formula and test set method shows the good agreement in choosing the optimal kernel width. For example, when the fault rate is equal to 0.05, i.e., Fig. 2a, the selected kernel width by the MPE formula is equal to about 0.316228, and the selected one by the test set method is about 0.281838. The distance between two testing error values is about 0.03439, i.e., 12.20%. Moreover, we repeat the simulation for 50 times and find that the range of selected kernel width values by our proposed MPE formula is around 0.251189–0.316228, which are very close to the selected result by the test set method. In addition, in the figure, the selected width value by the train set method (MSE formula) has been given, e.g., about 0.050119 when $p = 0.05$. The distance away from the actual optimal kernel width value is about 0.231719, i.e., 82.22%. The Fig. 2b has shown the selection results for three methods when $p = 0.15$. From the figure, we can obtain the same results with the Fig. 2a.

Therefore, from the simulation results for the sinc function approximation example, one can see that the kernel width selected by our proposed formula is very close to the optimal kernel width by the test set method. Noted that for the test set method, the computing time is relative with the number of faulty neural network models. Assuming the R faulty neural network models, e.g., $R = 10,000$ in our simulations, the total computing time for the test set method is RT , where T is the computing time for one neural network. However, the computing time for MPE formula is about T . It is evident that the computing time for MPE formula is shorter than that of the test set method.

In addition, Table 1 shows the comparison results between the different selection method of the optimal kernel width value, i.e., test set method [14], our MPE formula, MSE formula [9] and formula $d_{max}/\sqrt{2M}$ [6], where d_{max} is the maximum distance between any pair of centers, for faulty RBF neural networks with multi-node open fault. Comparing with the selected results by other formulae [6,9] used for fault-free neural networks, our proposed formula outperforms others. That is to say, our proposed method selects the optimal kernel width faster than the test set method and more accurately than the other formulae method for faulty RBF neural networks.

4.2 Mackey Glass Time Series

The Mackey Glass time series (MGTS) prediction example is a complex noisy time series prediction problem [24], those common expression is

$$y(t) = \frac{ay(t - \tau)}{b + y^h(t - \tau)} - cy(t) + e(t) \tag{23}$$

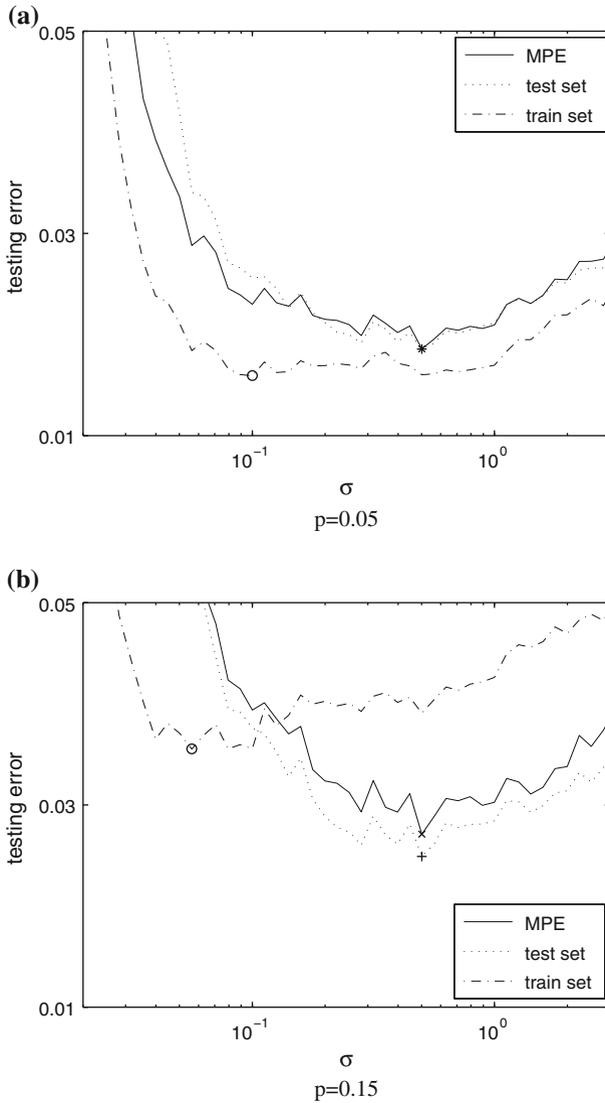


Fig. 3 The relation between the testing error and the candidate kernel width values with three measured methods, i.e., MPE formula, test set method and train set method, for MGTS prediction example with the different fault rate p . The signs, 'x', '+' and 'o', denote the lowest point on the curves of MPE formula, test set method and train set method, respectively

In the simulation, we set the parameters a, b, c, h as $a = 0.2, b = 1, c = 0.9, h = 10$. $e(t)$ is a mean zero Gaussian random variable whose variance is equal to 0.01. Thousand samples are generated, and 500 ones are used for training and the rest used for testing. The RBF model is used to predict $y(t)$ based on the past observations, $\{y(t-4), y(t-3), y(t-2), y(t-1)\}$. The prediction is given by

$$\hat{y}(t) = \hat{f}(x(t), \hat{\mathbf{w}}) = \sum_{j=1}^M w_j \phi_j(x(t)) \quad (24)$$

where $x(t) = [y(t-4), y(t-3), y(t-2), y(t-1)]^T$. Meanwhile, the optimal weight vector $\hat{\mathbf{w}}$ is obtained using the Eq. 10 based on the 500 training dataset.

Moreover, we employ the Chen's LROLS method [25] to select the important RBF kernel centers. According to selection results, the center number is set as 40.

Figure 3 has shown testing error of the different candidate kernel width values for MGTS prediction example with three selection methods, i.e., MPE formula, test set method and train set method (MSE formula). The solid line, dotted line and dash-dot line denote the results of MPE formula, the test set method and the train set method, respectively. Except for the train set method, the other two methods are exclusively used for the faulty neural networks. From the figure, we can see that the optimal kernel width selected by MPE formula is very close to that by test set method. For instance, when the fault rate is equal to 0.05, i.e., Fig. 3a, the selected width value by MPE formula is 0.501187, which is the same with that by the test set method, but the selected width value by train set method is 0.1. Therefore, the selected kernel width by our proposed formula is much closer to the actual optimal one than that by the train set method. Moreover, we repeat the same simulation process for 50 times and obtain the range of the selected results for the MPE formula is around 0.446684–0.562341. It is very obvious that the selected optimal kernel width values by the MPE formula are very close to that by the test set method. In addition, Fig. 3b has also shown the selection results for three methods when $p = 0.15$. From three curves, we can conclude the same conclusions with that of Fig. 3a. That is to say, the optimal kernel width value selected by our proposed MPE formula is closer to that by the test set method than that by the other formulae method [6, 9] used for fault-free neural networks in the multi-node open fault situations.

In summary, we can arrive at a conclusion that our proposed MPE formula outperforms the train set method and can be used for quickly selecting the optimal kernel width instead of the test set method.

5 Conclusion

In the real world, the samples are so valuable that the search process on the whole candidate kernel width is the only best optimal method. However, in multi-node open fault situations, where many probable faulty network models exist, it will take a great deal of time to calculate the mean prediction error of the whole faulty networks for each candidate kernel width. Hence, deriving a formula to evaluating the mean prediction error value can contribute to shorten the search process.

In the letter, we follow the Moody's thinking to derive a formula, i.e., MPE formula, based on the Leung's regularizer and then use it for selecting the optimal kernel width value for faulty RBF neural networks in the multi-node open fault situations. The theoretical and experimental results have demonstrated that our proposed MPE formula can be used to obtain

the more accurately optimal kernel width value than the conventional rapid method, i.e., the train set method, and to select the optimal kernel width value instead of the test set method.

Acknowledgements The work is supported by the Hong Kong Special Administrative Region RGC Ear-marked Grants (No. CityU 115606) and the Nature Scientific Youth Fund of South China University of Technology and the Nature Scientific Fund of Guangdong province, China (No. 07006488).

References

1. Park J, Sandberg I (1993) Approximation and radial-basis function networks. *Neural Comput* 5:305–316
2. Orr MJL (1996) Introduction to radial basis function networks. Technical reports. www.anc.ed.ac.uk/~mjo/papers/intro.ps
3. Bishop CM (1995) *Neural networks for pattern recognition*. Oxford University Press, Oxford
4. Chen S, Cowan CFN, Grant PM (1991) Orthogonal least squares learning algorithm for radial basis function networks. *IEEE Trans Neural Netw* 2(2):302–309
5. Masashi S, Hidemitsu O (2001) Subspace information criterion for model selection. *Neural Comput* 13(8):1863–1889
6. Haykin S (1999) *Neural networks a comprehensive foundation*, 2nd edn. Prentice-Hall Inc., Upper Saddle
7. Moody J, Darken CJ (1989) Fast learning in networks of locally-tuned processing units. *Neural Comput* 1: 281–294
8. Lázaro M, Santamaría I, Pantaleón C (2003) A new EM-based training algorithm for RBF networks. *Neural Netw* 16:69–77
9. Benoudjit N, Verleysen M (2003) On the kernel widths in radial-basis function networks. *Neural Process Lett* 18(2):139–154
10. Bolt GR (1991) Fault models for artificial neural networks, pp. 1371–1378
11. Murray AF, Edwards PJ (1994) Enhanced MLP performance and fault tolerance resulting from synaptic weight noise during training. *IEEE Trans Neural Netw* 5:792–802
12. Zhou Z, Chen S (2003) Evolving fault-tolerant neural networks. *Neural Comput Appl* 11:156–160
13. Bernier JL et al (2003) Assessing the noise immunity and generalization of radial basis function networks. *Neural Process Lett* 18(1):35–48
14. Moody J (1994) Prediction risk and architecture selection for neural networks. From statistic to neural networks: theory and pattern recognition application. NATO ASI Series F, Springer-Verlag, New York
15. Moody J (1991) Note on generalization, regularization and architecture selection in nonlinear learning systems. In: Juang BH, Kung SY, CA Kamm (eds) *Neural networks for signal processing*. IEEE Press, Piscataway, pp. 1–10
16. Leung CS, Young G, Sum J, Kan W (1999) On the regularization of forgetting recursive least square. *IEEE Trans Neural Netw* 10:1482–1486
17. Leung CS, Tsoi AC, Chan LW (2001) Two regularizers for recursive least squared algorithms in feedforward multilayered neural networks. *IEEE Trans Neural Netw* 12(6):1314–1332
18. Leung CS, Sum J (2008) A fault-tolerant regularizer for RBF networks. *IEEE Trans Neural Netw* 19(3):493–507
19. Murray AF, Edwards PJ (1994) Enhanced MLP performance and fault tolerance resulting from synaptic weight noise during training. *IEEE Trans Neural Netw* 5:792–802
20. Phatak D, Koren I (1995) Complete and partial fault tolerance of feedforward neural nets. *IEEE Trans Neural Netw* 6:446–456
21. Sum J, Leung CS, Ho K (2009) On Objective function, regularizer and prediction error of a learning algorithm for dealing with multiplicative weight noise. *IEEE Trans Neural Netw* 20(1):24–138
22. Fedorov VV (1972) *Theory of optimal experiments*. Academic Press, New York
23. Chen S, Hong X, Harris CJ, Sharkey PM (2009) Sparse modeling using orthogonal forward regression with press statistic and regularization. *IEEE Trans Syst Man Cybern B* 34:898–911
24. Mackey M, Glass L (1977) Oscillation and chaos in physiological control systems. *Science* 197(4300):287–289
25. Chen S (2006) Local regularization assisted orthogonal least squares regression. *Neurocomputing* 69(4C6):559–585