11th World Congress on Structural and Multidisciplinary Optimisation
07th -12th, June 2015, Sydney Australia

A RBF Neural Network Modeling Method based on Sensitivity Analysis and Pareto Law

Zongyu Wu, Yong Chen, Wen Yao, Xiaoqian Chen

National University of Defense Technology, Changsha, China, vjzwzy@outlook.com

1. Abstract
Radial basis function neural network (RBFNN) has been widely used in nonlinear function approximation. In this paper, two limits of RBFNN have been handled which are network complexity and large-scale calculation respectively. Firstly, network complexity, which results from problems of numerous width parameters optimization, is solved by a method of space decomposition based on sensitivity analysis. If a dimension is more sensitive to approximation error, the design space along this dimension is decomposed into several subspaces and the width parameter in each subspace is regarded as an independent variable and optimized respectively. In this way, the number of width parameters to be optimized can be reduced while the flexibility of parameter settings is maintained, so that the approximation accuracy and modeling efficiency can be balanced. Secondly, large-scale calculations, which come from leave-one-out method for cross validation error estimation, are improved by adopting the Pareto law in economic science. According to the Pareto law, referred as "majority is decided by the minority", we propose to choose only those sample points, which play dominant roles on the global errors, as cross validation points. Then large-scale calculations can be greatly reduced as the cross validation need not be conducted at those samples which have minor effects on the global errors. Combining the space decomposition and leave-one-out methods, the improved RBFNN modeling method based on sensitivity analysis and Pareto principle can effectively reduce the calculation costs and improves the accurate. Finally, several mathematical examples are tested to verify the efficacy of this method.
2. Keywords: RBFNN, sensitivity analysis, Pareto law

3. Introduction
With the development of computer technology, numerical simulation has been playing an increasingly important role in modern engineering design, but the huge amount of calculation is always a barrier. In recent years, metamodel, also called response surface method, is found to be a valuable tool in simulation areas. A metamodel is an approximation to system response constructed from its value at a limited number of selected input values [1]. Because of its simplicity and accurate results, it is used to instead of complex and computationally expensive systems.
The commonly used metamodels include Kriging, polynomial response surface model, support vector machine, radial basis function, neural network model et al. As the combination of radial basis function and neural network, radial basis function neural network is not only simple but also accurate. It is one of the most suitable approximation methods in approximating high nonlinear systems [2]. Especially in modeling the deterministic computer experiment response data which are identical each time the simulation is repeated, the interpolation modeling feature of RBFNN is very feasible and applicable. In the classical interpolation form of RBFNN, the RBF centers of neurons in network hidden layer are the training sample data. The accuracy of the model directly depends on the selected basis functions and the training samples [3].
With an unknown system, an efficient method to improve the approximation quality is to increase the number of sample data, which is also the neuron number in hidden layer of the network. If the number is big, the network would be extremely complicated and the calculation efficiency would be influenced. Besides, every sample data needs to run the high fidelity model to obtain the system response. If the sample set is too large, the calculation burden may be unacceptable. In recent years, the commonly used method to improve the approximation quality is shape parameter optimization to instead of increasing the number of sample data. The present research on the shape parameter optimization mainly includes two categories:
a) The shape parameters of all the RBF neurons are fixed to the same value with optimization methods.
b) The shape parameters of all the RBF neurons are designed respectively with optimization methods.
In the first category, the shape parameters of all the RBF neurons are set to be the same. It is much easier to design and optimize this single parameter, but this simplification may greatly limit the capability of RBFNN in complex highly nonlinear approximation problems. The second category designs and optimizes the shape parameter of each neuron respectively. Theoretically this method can find the best approximation model, but the giant calculation cost of each neuron optimization respectively may be unacceptable.
In this paper, two limits of RBFNN have been handled which are network complexity and large-scale calculation respectively. Actually, network complexity is also the problem of large-scale calculation. Firstly, we propose a

new method of space decomposition based on sensitivity analysis to solve the problem of network complexity, which results from problems of numerous width parameters optimization. In this way, the number of width parameters to be optimized can be reduced while the flexibility of parameter settings is maintained, so that the approximation accuracy and modeling efficiency can be balanced. Secondly, we improved the traditional leave-one-out method for cross validation error estimation by adopting the Pareto law, to solve the problem of large-scale calculations. In this way, large-scale calculations can be greatly reduced. Combining the space decomposition and leave-one-out methods, the improved RBFNN modeling method based on sensitivity analysis and Pareto principle can effectively reduce the calculation costs and improves the accurate. Finally, several mathematical examples are taken to verify the proposed method, and the results are discussed.

4. RBFNN based on sensitivity analysis and Pareto law

4.1. RBF neural network

Radial basis function neural network is essentially an interpolation method. In classical RBFNN model, the approximation response is defined as a linear combination of radial functions. The approximation response expression is as follows,

$$\hat{y} = \sum_{i=1}^{Ns} w_i \phi_i(\|x - x_i\|) \tag{1}$$

Ns is the number of hidden layer neurons, which is also the number of sample points. x_i is the sample point, and w_i is the output layer weight. ϕ_i is the radial basis function of the Euclidean distance $\|x - x_i\|$. $\hat{y}$ is the approximation response at the unknown point x.

The output weighs constitute a vector $W=[w_1,w_2,\cdots,w_{Ns}]^{\mathrm{T}}$, and the radial basis functions constitute a vector $\varphi(x) = [\phi_1(\|x - x_1\|), \phi_2(\|x - x_2\|), \quad , \phi_{Ns}(\|x - x_{Ns}\|)]^{\mathrm{T}}$. Eq.(1) can be written as

$$\hat{y} = W^{\mathrm{T}} \cdot \varphi(x) \tag{2}$$

The weight vector W can be obtained from

$$W = \boldsymbol{\Phi}^{-1} Y \tag{3}$$

The accurate responses of sample points constitute the vector $Y = [y_1, y_2, \quad , y_{Ns}]^{\mathrm{T}}$. The expression of matrix $\boldsymbol{\Phi}$ can be written as

$$\boldsymbol{\Phi} = \begin{bmatrix} \phi_1(x_1, x_1) & \phi_2(x_1, x_2) & \phi_{Ns}(x_1, x_{Ns}) \\ \phi_1(x_2, x_1) & \phi_2(x_2, x_2) & \phi_{Ns}(x_2, x_{Ns}) \\ & & \\ \phi_1(x_{Ns}, x_1) & \phi_2(x_{Ns}, x_2) & \phi_{Ns}(x_{Ns}, x_{Ns}) \end{bmatrix} \tag{4}$$

The radial basis function $\phi_i(\cdot)$ has many forms such as the Gaussian function, the thin-plate-spline function, the multi-quadric function, etc.[4, 5]. As the theoretical investigation and practical results suggest that multi-quadric function has a high rate of convergence [6, 7], in this paper we choose multi-quadric function as basis function. We use r_i substituting the Euclidean distance $\|x - x_i\|$, then the radial basis function can be written as

$$\phi_i = \sqrt{r_i^2 + c_i^2} \tag{5}$$

c_i is constant number of each neuron. Because the shape characteristic of basis function is decided by c_i, we call c_i the shape parameter.

4.2. Space decomposition base on sensitivity analysis

To obtain the high approximation accuracy, we usually choose the second category of shape parameter optimization stated in section 3, but the expensive computational cost is a barrier. To improve the computational efficiency in solving the shape parameter optimization problem, a method of space decomposition is proposed based on sensitivity analysis. We decompose the whole design space into several sub-spaces, and the shape parameter in each subspace is regarded as an independent variable and optimized respectively. In this way, the number of shape parameters to be optimized can be reduced. Therefore, the large-scale optimization problem is decomposed into several sub-problems, each of which has less optimization variables and smaller matrix to

manipulate.
For a *n*-dimensional design space, a simple method to decompose the space is: if each dimension is divided into *m* parts equally, we can obtain m^n sub-spaces. This method is simple and easy to operate, but it may encounter a problem, dimension disaster. For a 20-dimensional design space, we divide each dimension into 2 parts, then the design space is decomposed into 2^{20} sub-spaces. The big number of sub-spaces makes it difficult to complete the computation.
Sensitivity analysis is the study of how the change in the output of a mathematical model or system can be apportioned to different sources of change in its inputs. It is an important method to solve the complex problem in MDO. In this section, a practical approach of space decomposition is proposed based on sensitivity analysis. The basic idea of this method includes three parts:
a) Analyze the sensitivity of each dimension respectively, and arrange these dimensions according to the impact on system in an order from largest to smallest.
b) If a dimension is more sensitive to approximation error, we insist that change in this dimension has larger influence to system. Select *k* dimensions which have largest impact on the system, and divide these dimensions into *m* parts respectively. Then the design space is decomposed into m^k sub-spaces and the shape parameter in each subspace is regarded as an independent variable and optimized respectively.
c) If a dimension is less sensitive to approximation error, we insist that change in this dimension has little influence to system. Therefore, measures are not taken to this dimension.
For a 20-dimensional design space, we divide each dimension into 2 parts, then the design space is decomposed into 220 sub-spaces. The big number of sub-spaces makes it difficult to complete the computation.

4.3. Leave-one-out method based on Pareto law
Leave-one-out method is a cross validation method of estimating the approximation model prediction error. The basic ideas of this method are as follows:
a) For a sample set $X=[x_1,x_2,\cdots,x_{Ns}]^{\mathrm{T}}$, leave out one sample point $x_i(1 \le i \le Ns)$, and construct metamodel based on the rest of the sample points $X_{-i}=[x_1,x_2,\cdots,x_{i-1},x_{i+1},\cdots,x_{Ns}]^{\mathrm{T}}$.
b) Use this metamodel to predict the response on the leave-out point $\hat{y}_i$.
c) Calculate the difference e_{-i} between the prediction $\hat{y}_i$ and the accurate response y_i.
Generally speaking, we use RMSE (root of mean square errors) to estimate the global error of metamodel. The RMSE is calculated by the following equation:

$$\mathrm{RMSE} = \sqrt{\left(\sum_{i=1}^{Ns}(y_i - \hat{y}_i)^2\right) \Big/ Ns} \tag{6}$$

Where *Ns* is the number of sample points, $\hat{y}_i$ and y_i are the predicted response and accurate response. The smaller the value of RMSE, the better accurate the metamodel will be.
By optimizing Eq.(6), the smallest value of RMSE can be found. But the drawback of this optimization based on leave-one-out is large-scale calculations. It will take much time to build the leave one out metamodel for cross validation of every point in the sample set, especially for RBFNN which needs large matrix calculation. To save cross validation time and calculation cost, we improve the leave-one-out method by Pareto law, which only use part of the sample data as the key points to conduct cross validation and build the global error prediction metamodel.
The Pareto law (also known as the law of vital few) states that, for many events, roughly 80% of the effects come from 20% of the causes. Therefore, many businesses have an easy access to dramatic improvements by focusing on the most effective areas and eliminating, ignoring the rest, as appropriate.
According to the Pareto law, referred as "majority is decided by the minority", we propose to choose only those sample points, which play dominant roles on the global errors, as cross validation points. Then large-scale calculations can be greatly reduced as the cross validation need not be conducted at those samples which have minor effects on the global errors.
The leave-one-out method is improved by the Pareto law, and the main idea of this method is as follows:
a) Calculate the square errors $e_i^2(1 \le i \le Ns)$ based on the traditional leave-one-out method by the following equation,

$$e_i^2 = \left|y_i - \hat{y}_i\right|^2 \tag{7}$$

b) Rearrange the sample set $X=[x_1,x_2,\cdots,x_{Ns}]^{\mathrm{T}}$, according to the value of $e_i^2(1 \le i \le Ns)$ in an order from largest to smallest. Then, we obtain the new sample set $X' = [x_1', x_2', \quad, x_{Ns}']^{\mathrm{T}}$.

c) It is obvious that, those points with larger square error have larger influence on the global error. Choose *k* points with larger square error as Pareto points from the new sample set X' by the following program:

$$
\begin{aligned}
&E' = 0;\\
&m = 0;\\
&for\ i = 1 : Ns\\
&\quad E' = E' + e_i'^2;\\
&\quad m = m + 1;\\
&\quad if\ (E' \geq \mu E)\\
&\quad\quad break;\\
&\quad end\\
&end\\
&return(m);
\end{aligned}
\tag{8}
$$

Where $\mu(0 < \mu \leq 1)$ is a constant, we call it Pareto coefficient. We set it to 0.8.

d) After determining the Pareto points, the RMSE is calculated by the following equation:

$$\text{RMSE} = \sqrt{\left(\sum_{i=1}^{k} (y_i' - \hat{y}_i')^2\right) \Big/ k} \tag{9}$$

Where $\hat{y}_i'$ and y_i' are the predicted value and accurate value on the point x_i'.

4.4. The method of SAPRBFNN

Combining the space decomposition and leave-one-out methods, the RBFNN modeling method based on sensitivity analysis and Pareto principle (SAPRBFNN) is proposed. The flowchart of SAPRBFNN is shown in Fig.1.

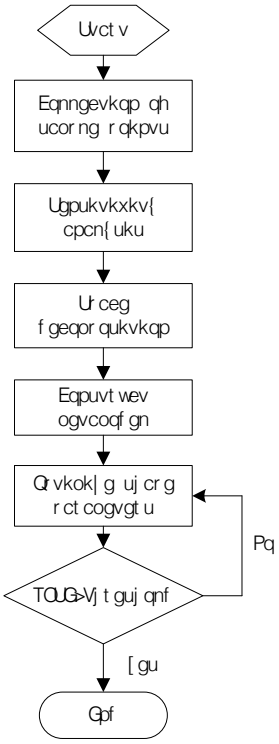


Figure 1: SAPRBFNN modeling flowchart

The proposed method mainly includes the following parts:

a) Collection of sample points. In this paper, an optimum Latin hypercube design method is utilized to construct the uniformly distributed sample points filling the design space.

b) Sensitivity analysis. Analyze the sensitivity of each dimension respectively, and arrange these dimensions according to the impact on system in an order from largest to smallest.
c) Space decomposition. Select k dimensions which have largest impact on the system, and divide these dimensions into m parts respectively. Then the design space is decomposed into m^k sub-spaces and the shape parameter in each subspace is regarded as an independent variable and optimized respectively.
d) Construct metamodel in each sub-space respectively.
e) Optimize shape parameters based on Pareto law. Determine the Pareto points of each sub-space, and then calculate RMSE by Eq.(9). The shape parameter in each subspace is regarded as an independent variable and optimized respectively.
f) Judge the convergence of the optimization. When RMSE reaches the required accuracy threshold, the optimization process terminates.

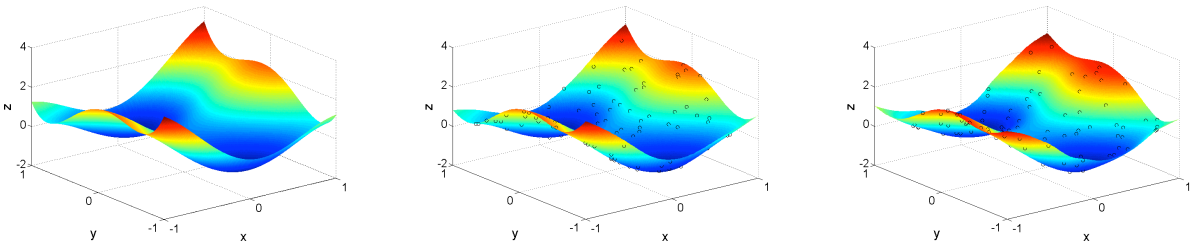
5. Tests of SAPRBFNN

Now we will test SAPRBFNN using Camelback function and a 7-dimensional function. All the two functions are constructed through optimum Latin hypercube design sampling strategy.

5.1. Camelback function

$$f(x_1,x_2)=(4-2.1x_1^2+x_1^4/3)x_1^2+x_1x_2+(4x_2^2-4)x_2^2\ ,\quad x_1\in[-1,1],x_2\in[-1,1] \tag{10}$$

The sample set is composed of 100 uniformly distributed points in the design space designed by optimum Latin hypercube design method. Comparing the traditional RBFNN and SAPRBFNN, and the results are shown in Fig. 2 and Table 1.



(a)Accurate function (b)RBFNN modeling result (c)SAPRBFNN modeling result

Figure 2: Test result of Camelback function

Table 1: Test result of Camelback function

Method	Shape parameter		Percentage of validation points	RMSE
RBFNN	———		———	0.0451
SAPRBFNN	Subspace 1	0.3189	17%	0.0460
	Subspace 2	0.1046	8%	
	Subspace 3	0.3305	11%	
	Subspace 4	0.0098	16%	

5.2. 7-dimensional function

$$\begin{aligned} f(x)=&0.7854x_1x_2^2(3.3333x_3^2+14.9334x_3-43.0934)-1.5079x_1(x_6^2+x_7^2)+\\ &7.477(x_6^3+x_7^3)+0.7854(x_4x_6^2+x_5x_7^2)\quad ,\ 1\le x_i\le 5 \end{aligned} \tag{11}$$

The sample set is composed of 300 uniformly distributed points in the design space designed by optimum Latin hypercube design method. Firstly, analyzing the sensitivity of this function. After the normalization process, the results of sensitivity analysis is depicted in Fig. 3. It is obvious that x_2 and x_3 are more sensitive than other variables. Therefore, these two dimensions is divided into 2 parts respectively. Then the design space is decomposed into 4 sub-spaces. Comparing the traditional RBFNN and SAPRBFNN, and the results are shown in Table 2.

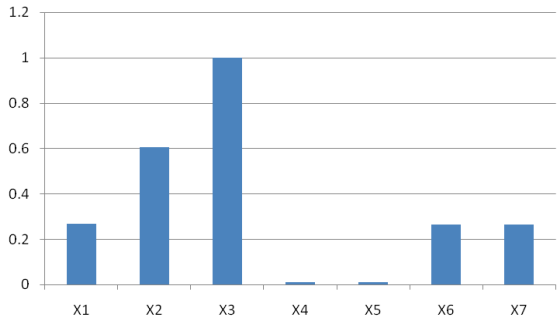


Figure 3: Sensitivity analysis

Table 2: Test result of 7-dimensional function

Method	Shape parameter		Percentage of validation points	RMSE
RBFNN	———		———	93.7760
SAPRBFNN	Subspace 1	0.2886	18%	97.2409
	Subspace 2	0.5836	20%	
	Subspace 3	1.0060	19%	
	Subspace 4	0.0198	19%	

6. Discussions and conclusions

From Camelback function, the number of shape parameters is decreased from 100 to 4, and the percentage of validation points (Pareto points) in all sample points is less than 20%. As the Fig. 3 shows, the accuracy remains almost the same with the traditional RBFNN. The RMSE test also supports the judgment. For 7-dimensional function, we can draw the same conclusion.

In this paper, two limits of RBFNN have been handled which are network complexity and large-scale calculation. Firstly, a method of space decomposition based on sensitivity analysis is proposed to solve network complexity, which results from problems of numerous width parameters optimization. In this way, the number of width parameters to be optimized can be reduced while the flexibility of parameter settings is maintained, so that the approximation accuracy and modeling efficiency can be balanced. Secondly, the leave-one-out is improved by Pareto law to solve the problem of large-scale calculations, which come from cross validation error estimation. Then large-scale calculations can be greatly reduced as the cross validation need not be conducted at those samples which have minor effects on the global errors. The results of the benchmark functions show that, the proposed SAPRBFNN which combining the space decomposition based on sensitivity analysis and leave-one-out based on Pareto law can effectively reduce the calculation costs. Most importantly, the accuracy remains almost the same with the traditional RBFNN. Further research of this method should be conducted in high dimensional problems and engineering application problems.

7. Acknowledgements

This work is supported by the National Natural Science Foundation of China (No. 11404405)

8. References

[1] X. Wei, Y.Z. Wu and L.P. Chen, A new sequential optimal sampling method for radial basis functions, *Applied Mathematics and Computation*, 218, 9635-9646, 2012.

[2] J. Park and W. Sandberg, Universal approximation using radial basis function networks, *Neural Computation*, 3, 246-257, 1991.

[3] H.B. Fang and H.F. Mark, Metamodeling with radial basis functions, *The 46th AIAA/ASME/ASCE/ AHS/ASC Structures, Structural Dynamics and Materials Conference*, Austin, Texas, AIAA2005-2059, 2005.

[4] M.J.D. Powell, Radial basis functions for multivariable interpolation: A review, Algorithms for Approximation, Clarendon Press, New York, 143-167, 1987.

[5] M.J.D. Powell, Radial basis function approximations to polynomials, *12th Biennial Numerical Analysis Conference*, Dundee, 223-241, 1987.

[6] L.P. Wang, X.Y. Wang and H.R. Zhan, The error analysis of element-free method based on radial basis function used in mechanical problems, *Mechanics in Engineering*, 35 (2), 67-72, 2013.

[7] V. Bayona, M. Moscoso and M. Kindelan, Optimal variable shape parameter for multiquadric based RBF-FD method, *Journal of Computational Physics*, 231(6), 2466-2481, 2012.