A Dimension Reduction Method Based on Adaptive RBF Neural Network in Supervised Machine Learning

YAN Taishan*, WEN Yiting

School of Information science and Engineering, Hunan Institute of Science and Technology, Yueyang, Hunan, 414000, China

*Corresponding author’s e-mail: yantaishan163@163.com

Abstract. For the purpose of high dimensional data reduction in supervised machine learning, an adaptive RBF (Radial Basis Function) neural network algorithm was proposed. According to the relationship between the total error of this training and last training, the learning rate and momentum factor in the algorithm were adjusted dynamically and adaptively. So, the learning direction of RBF neural network was enhanced and the learning speed was improved. This algorithm was used to reduce the dimension of high dimensional data. The data set auto-mpg in UCI machine learning database was taken as an example to carry out dimension reduction experiments. The experimental results show that the learning performance of RBF neural network was improved and its general error was reduced after reducing the dimensionality of high dimensional data set by this algorithm. This reduction method is of great significance to solve the "dimensionality disaster" and "cost disaster" in machine learning.

1. Introduction

Machine learning is a subject that researches how to use computers to simulate human learning activities. It has been applied in many fields successfully. Nowadays, human society has entered big data era. Most of the data are featured by high dimension, large scale and complex structure. In machine learning, when the dimensions of data set is too high, the number of samples needed to analyze high dimensional vectors increases exponentially with the dimension, thus lead to a sharp increase in the number of training samples. Meanwhile, the high computational complexity will be produced and the computational cost will increase. This is the famous "dimension disaster" and "cost disaster" [1].

Therefore, in modern machine learning, it is necessary to reduce the dimension of high-dimensional data. In the aspect of dimension reduction methods, scholars have proposed a series of dimension reduction algorithms, including classical algorithm[2-7], kernel method[8-9] and manifold learning algorithm[10-14]. However, with the increasing complexity and diversity of data, the limitations of these algorithms become increasingly obvious, which are mainly reflected in the following aspects: the linear nature of the classical algorithms makes them unable to deal with complex data. In the kernel method, the selection of kernel and kernel parameters is difficult to grasp. There are many problems in manifold learning algorithm, such as low computational efficiency and failure to generalize to out-of-sample.

Neural network is a distributed information processing system established by simulating the structure and function of human brain. Compared with other methods of dimension reduction, neural network has huge advantages, which is attributed to its characteristics of high nonlinearity, complex structure, self-learning and self-adaptation. It has strong ability to extract high-quality and low
dimensional features, especially in the face of complex real data, the representation ability and solving ability of neural network are particularly outstanding. Radial basis function (RBF) neural network is a novel and effective feed-forward neural network with strong nonlinear mapping ability. Since there is a complex nonlinear relationship between input and output of data sets in machine learning, it not only has sufficient theory basis but also has superiority to realize dimension reduction of high-dimensional data set by RBF neural network.

2. Basic theory
In supervised machine learning, there is a certain correspondence between each input component of data set and output result, which is often a complex nonlinear relationship. We call this relationship the importance of input dimension for output result. In the process of data set processing, how to judge the importance of each input component for output result is a problem to be solved firstly. Based on the nonlinear function approximation capability of RBF neural network, it can be used to simulate the importance of each input component for output result. According to the differential calculus principle, the influence of each input component with little changes on output result can be calculated by RBF neural network.

We introduce the concept importance to measure the influence of each input component on output result. The definition of importance is defined as: Let the input vector of RBF neural network be: 
\[ X_k = (x_1, x_2, \cdots , x_m), k = 1, 2, \cdots, p, \]
where, \( p \) is the number of input modes, \( m \) is the dimension number of the input vector. The corresponding output mode vector is 
\[ Y_k = (y_1, y_2, \cdots , y_n), \]
where \( n \) is the dimension number of the output vector. Suppose \( Y_i \) is the output of \( Y \) when input component \( i \) changes slightly with \( \Delta x_i \). The following formula
\[ I_i = \frac{\partial Y}{\partial x_i} = \lim_{\Delta x_i \to 0} \frac{\Delta Y}{\Delta x_i} \approx \frac{\|Y_i - Y\|}{\Delta x_i} \]
is called the importance of \( x_i \) for \( Y \).

In this way, according to the importance of each input component for output, the influence on output of each input component can be analyzed. If there are some dimensions in the input vector of the data set, they have little influence on output result, that is to say their importance is very low, so these dimensions belong to unimportant dimensions. The unimportant dimension is defined as: Among all the input components of data set, the component with the worst influence on output result is called unimportant dimension. The importance of unimportant dimension is:
\[ I_a \leq I_i \]
(2)

\( I_i \) is the importance threshold given as needed.

The input dimension that becomes the unimportant dimension most frequently in the given experiments will become redundant dimension.

3. An adaptive RBF neural network algorithm
RBF neural network uses Radial Basis Function (RBF) as the "basis" of network hidden layer units to form hidden layer space. Hidden layer transforms the input vector, that is to say transforms the input data of low dimensional mode into the higher-dimensional space which makes the linear inseparable problem in low dimensional space linearly separable in high dimensional space. The training of RBF neural network usually adopts gradient descent method, and the center, width and weight parameters are adjusted to the optimal value by learning. The iterative calculation is:
\[ w_{kj}(t) = w_{kj}(t-1) - \eta \frac{\partial E}{\partial w_{kj}(t-1)} + \alpha [w_{kj}(t-1) - w_{kj}(t-2)] \]
(3)
\[ c_{ij}(t) = c_{ij}(t-1) - \eta \frac{\partial E}{\partial c_{ij}(t-1)} + \alpha [c_{ij}(t-1) - c_{ij}(t-2)] \]  
(4)

\[ d_{ij}(t) = d_{ij}(t-1) - \eta \frac{\partial E}{\partial d_{ij}(t-1)} + \alpha [d_{ij}(t-1) - d_{ij}(t-2)] \]  
(5)

Where, \( w_{ij}(t) \) is the adjust weight between output neuron \( k \) and hidden layer neuron \( j \) in iterative computation \( t \); \( c_{ji}(t) \) is the center component in iterative computation \( t \) that hidden layer neuron \( j \) for input neuron \( i \); \( d_{ji}(t) \) is the corresponding width of center \( c_{ji}(t) \); \( \eta \) is the learning factor; \( E \) is the evaluation function of RBF neural network and the expression is:

\[ E = \frac{1}{2} \sum_{i=1}^{N} \sum_{k=1}^{q} (y_{ik} - O_{ik})^2 \]  
(6)

RBF neural network has strong ability of global optimization and generalization, so it has been widely used in many fields. However, in practical application, it is found that the effectiveness of RBF neural network learning algorithm depends on the selection of learning rate and momentum factor to a certain extent. Since the learning rate and momentum factors in the basic RBF neural network algorithm are fixed and invariant, which have certain influence on the convergence rate, the algorithm needs a long training time for some complex problems.

In order to accelerate the convergence speed of the algorithm, we make adaptive adjustment to the learning rate \( \eta \) and momentum factor \( \alpha \), and design an adaptive RBF neural network algorithm. The basic idea of this algorithm is: first, set an initial learning rate and momentum factor. In the learning process, according to the relationship between the total error \( E(t+1) \) obtained after each training and the total error \( E(t) \) obtained after the last training, the learning rate and momentum factor are dynamically adjusted. If the error increases after a iteration, it means that the adjustment direction is invalid and the learning rate and momentum factor should be appropriately reduced to re-calculate the next iteration in the original direction. If the error decreases after a iteration, it indicates that the adjustment direction is effective. The learning rate and momentum factor should be appropriately increased to continue the next iteration. The specific adjustment methods of learning rate and momentum factor are:

\[ \eta(t+1) = \begin{cases} 1.1\eta(t), & E(t+1) < 0.95E(t) \\ 0.9\eta(t), & E(t+1) > 1.05E(t) \\ \eta(t), & others \end{cases} \]  
(7)

\[ \alpha(t+1) = \begin{cases} 1.1\alpha(t), & E(t+1) < 0.95E(t) \\ 0.9\alpha(t), & E(t+1) > 1.05E(t) \\ \alpha(t), & others \end{cases} \]  
(8)

where, \( t = 0, 1, 2, \ldots, T - 1 \), \( T \) is the maximum number of learning set.

The flow of adaptive RBF neural network learning algorithm is shown in Fig.1.

### 4. Dimension reduction method based on adaptive RBF neural network

The steps of dimension reduction with adaptive RBF neural network can be described as follows:

1. **Normalization of data.** Min_max normalization method is used to prevent the platform phenomenon in the process of neural network training. Data is normalized between 0 and 1 according to the following method:
   
   \[
   \text{maxval}[i] = \text{the maximum value of component } i \\
   \text{minval}[i] = \text{the smallest value of component } i \\
   \text{range}[i] = \text{maxval}[i] - \text{minval}[i] \\
   \text{for ( sample p ) } \\
   \text{for ( component i ) }
   \]

2. **Dimension reduction method based on adaptive RBF neural network.**

   The steps of dimension reduction with adaptive RBF neural network can be described as follows:

   1. **Normalization of data.** Min_max normalization method is used to prevent the platform phenomenon in the process of neural network training. Data is normalized between 0 and 1 according to the following method:

   \[
   \text{maxval}[i] = \text{the maximum value of component } i \\
   \text{minval}[i] = \text{the smallest value of component } i \\
   \text{range}[i] = \text{maxval}[i] - \text{minval}[i] \\
   \text{for ( sample p ) } \\
   \text{for ( component i ) }
   \]

3. **Dimension reduction method based on adaptive RBF neural network.**

   The steps of dimension reduction with adaptive RBF neural network can be described as follows:

   1. **Normalization of data.** Min_max normalization method is used to prevent the platform phenomenon in the process of neural network training. Data is normalized between 0 and 1 according to the following method:

   \[
   \text{maxval}[i] = \text{the maximum value of component } i \\
   \text{minval}[i] = \text{the smallest value of component } i \\
   \text{range}[i] = \text{maxval}[i] - \text{minval}[i] \\
   \text{for ( sample p ) } \\
   \text{for ( component i ) }
   \]

4. **Dimension reduction method based on adaptive RBF neural network.**

   The steps of dimension reduction with adaptive RBF neural network can be described as follows:

   1. **Normalization of data.** Min_max normalization method is used to prevent the platform phenomenon in the process of neural network training. Data is normalized between 0 and 1 according to the following method:

   \[
   \text{maxval}[i] = \text{the maximum value of component } i \\
   \text{minval}[i] = \text{the smallest value of component } i \\
   \text{range}[i] = \text{maxval}[i] - \text{minval}[i] \\
   \text{for ( sample p ) } \\
   \text{for ( component i ) }
   \]
input[p][i]= (input[p][i]– minval[i])/range[i]

(2) The normalized data set is used to train the adaptive RBF neural network.

(3) Calculate the importance of each input components. Add small changes Δx_i to each input component x_i of the sample, and get the network’s output Y^k_i, where, k = 1, 2, ..., p , i = 1, 2, ..., m , p is the number of input modes, m is the dimension number of the input vector. Calculate the importance \( \frac{\partial Y^k_i}{\partial x_i} \) of each input component for all samples respectively. Then, calculate the mean value of the importance of each input component:

\[
I_{i, \text{ave}} = \frac{1}{p} \sum_{j=1}^{p} \frac{\partial Y^k_i}{\partial x_i} \tag{9}
\]

(4) The mean value of the importance of each input component \( I_{i, \text{ave}} \) is compared with the threshold \( I_i \). The input components which are below or equal to the threshold will be considered unimportant dimensions.

(5) The increment of each input dimension is changed, their importance and its mean value is recalculated, the unimportant dimension judged, and the experiment is repeated n times.

(6) Redundant dimension judgement. If input dimension j becoming the unimportant dimension is the most frequent in n experiments, then input dimension j becomes the redundant dimension and should be eliminated.

**Fig.1** The flow of adaptive RBF neural network learning algorithm

5. **Experimental analysis**

The auto-mpg data set in UCI machine learning database (some data are shown in Table 1) is used to verify the validity of the dimension reduction method proposed in this paper. The RBF neural network was trained by the data set before and after dimension reduction respectively. The general errors of the trained neural network before and after data processing were compared.

The data set has seven input components, they are: x1(cylinders), x2(displacement), x3(horsepower), x4(weight), x5(acceleration), x6(model year), x7(origin). These input components are
used to predict the mpg (miles per gallon) of city. The data set has a total of 406 samples. 330 examples were used as training sample set, and the other were used as test sample set.

Table 1 Part of data set auto-mpg

| No | x1  | x2  | x3  | x4  | x5  | x6  | x7  | y   |
|----|-----|-----|-----|-----|-----|-----|-----|-----|
| 1  | 8   | 307.0 | 130.0 | 3504 | 12.0 | 70  | 1   | 18.0 |
| 2  | 8   | 440.0 | 215.0 | 4312 | 8.5  | 70  | 1   | 14.0 |
| 3  | 4   | 97.00 | 46.00 | 1835 | 20.5 | 70  | 2   | 26.0 |
| 4  | 6   | 250.0 | 100.0 | 3282 | 15.0 | 71  | 1   | 19.0 |
| 5  | 4   | 79.00 | 70.00 | 2074 | 19.5 | 71  | 2   | 30.0 |
| 6  | 8   | 350.0 | 155.0 | 4502 | 13.5 | 72  | 1   | 13.0 |
| 7  | 6   | 250.0 | 88.00 | 3021 | 16.5 | 73  | 1   | 18.0 |
| 8  | 4   | 97.00 | 88.00 | 2279 | 19.0 | 73  | 3   | 20.0 |
| 9  | 8   | 400.0 | 170.0 | 4668 | 11.5 | 75  | 1   | 16.0 |
| 10 | 4   | 116.0 | 81.00 | 2220 | 16.9 | 76  | 2   | 25.0 |
| 11 | 8   | 350.0 | 105.0 | 3725 | 19.0 | 81  | 1   | 26.6 |
| 12 | 4   | 91.00 | 67.00 | 1965 | 15.7 | 82  | 3   | 32.0 |

The training sample set is firstly used to train the RBF neural network, and the significance of each input dimension is calculated by the method proposed in this paper. Table 2 shows the results of 6 experiments, in which each row data are the importance values of each input dimension when each increment is taken.

The attribute with the smallest significance is taken as the unimportant dimension every time. The sixth dimension is the unimportant dimension in all experiments. So, this dimension will be deleted to achieve dimension reduction. RBF network is trained by the data set before and after dimension reduction processing. The smaller the general error is, the higher the learning performance will be. The general error curves before and after dimension reduction is shown in Fig.2 and Fig.3 respectively.

Table 2 The importance of input component

| \( \Delta \alpha_i \) | x1  | x2  | x3  | x4  | x5  | x6  | x7  |
|---------------------|-----|-----|-----|-----|-----|-----|-----|
| 0.1                 | 5.1262 | 5.1262 | 5.1262 | 5.1217 | 5.1225 | 5.0360 | 5.1262 |
| 0.2                 | 4.6685 | 4.6680 | 4.6665 | 4.6678 | 4.6634 | 4.66154 | 4.6680 |
| 0.3                 | 3.7580 | 3.7572 | 3.7580 | 3.7580 | 3.7580 | 3.7585 | 3.7580 |
| 0.4                 | 3.0875 | 3.0822 | 3.0875 | 3.0875 | 3.0875 | 3.0026 | 3.0875 |
| 0.5                 | 2.5568 | 2.5563 | 2.5568 | 2.5568 | 2.5568 | 2.5515 | 2.5568 |
| 0.6                 | 2.1076 | 2.1055 | 2.1076 | 2.1076 | 2.1076 | 2.1028 | 2.1076 |

Fig.2 General error curve before dimension reduction

Fig.3 General error curve after dimension reduction

We can draw the following conclusions from the above experimental results: Firstly, the significance defined in this paper can reflect the close relationship between each input component and...
the output, and can evaluate the importance of each input dimension for the solving problem. Secondly, after dimension reduction processing by the method proposed in this paper, the machine learning performance can be improved significantly and its general error can be reduced greatly.

6. Conclusion
Data dimension pre-processing is very important for supervised machine learning, which is the premise of efficient learning. In this paper, an adaptive RBF neural network learning algorithm is proposed for data dimension. It took advantage of the highly nonlinear function approximation ability of RBF neural network in data dimension reduction. This method is suitable for supervised machine learning, in which the nonlinear relationship between experimental data and for solving problems is complex and conventional mathematical statistics dimension reduction method is difficult to achieve satisfied effect. Experimental results show that the proposed method can improve the machine learning performance. So, it is a novel method with good application value to reduce dimension by RBF neural network.

Acknowledgments
This work was financially supported by the Natural Science Foundation of Hunan Province under Grant No.2017JJ2107 and the Science and Technology Program of Hunan Province under Grant No. 2016TP1021.

References
[1] Domingos P. A few useful things to know about machine learning. Communications of the Acm, 2012, 55(10):78-87
[2] Ruan Yue, Chen Hanwu, Liu Zhihao. Quantum principal component analysis algorithm. Chinese Journal of Computers, 2014, 37(3):666-676
[3] Si Jiarong, Zhou Shuisheng, Zhen Xiuyun. Multilinear robust principal component analysis. Chinese JOURNAL OF ELECTRONICS, 2014, 42(8):666-676
[4] Cui Zhen, Shan Shiguan, Chen Xilin. Structured sparse linear discriminant analysis. Journal of Computer Research and Development, 2014, 51(10):2295-2301
[5] Chen Sibao, Chen Daoran, Luo Bin. Two-dimensional linear discriminant analysis based on L1-norm. Journal of electronics and information technology, 2015, 37(6):1372-1377
[6] Zhou Hangxing, Chen Songchan. Ordinal discriminant canonical correlation analysis. Journal of software, 2014, 25(9):2018-2025
[7] Zu Chen, Zhang Daoqiang. Sparse keep canonical correlation analysis with sample deletion. Pattern recognition and artificial intelligence, 2014, 27(2):179-186
[8] K.R. Muller, S. Mika, et al. An Introduction to Kernel-based Learning algorithms. IEEE Trans. on Neural Networks, 2001, 12(2):181-202
[9] Wang Hongqiao, Cai Yanning, Sun Fuchuan. Adaptive sequence learning and application of multi-scale kernel method [J]. Pattern recognition and artificial intelligence, 2011, 24(1):72-81
[10] Belkin M, Niyogi P. Laplacian Eigenmaps for Dimensionality Reduction and Data Representation. Neural Computation, 2003, 15(15):1373-1396.
[11] Hinton G, Roweis S. Stochastic Neighbor Embedding. Advances in Neural Information Processing Systems, 2010, 41(4):835-840.
[12] Huang Yunjuan, Li Fanchang. Equal spectral manifold learning algorithm. Journal of software, 2013, 24(11):2656-2666
[13] Shao Chao, Wan Chunhong. Supervised multimanifold learning algorithm based on isometric mapping. Pattern recognition and artificial intelligence, 2014, 27(2):111-119
[14] Yuan Min, Cheng Lei, Zhu Rangang. A new supervised manifold learning algorithm based on MMC and LSE. Journal of automation, 2013, 39(12):2077-2089