Non-negative Radial Basis Function Neural Network in Polynomial Feature Space

Huiyang Wang\(^1\), Yang Zhao\(^{2, a}\), Jihong Pei\(^1\), Dehuai Zeng\(^1\) and Meijuan Liu\(^3\)

\(^1\)College of Mechatronics and Control Engineering, Shenzhen University, 518060 Shenzhen, Guangdong, P.R.China
\(^2\)College of Information Engineering, Shenzhen University, 518060 Shenzhen, Guangdong, P.R.China
\(^3\)College of Design and Art, University of South China, 421001 Hengyang, Hunan, P.R.China
\(^a\)Corresponding author: zhaoyangmaths@163.com

Abstract. Radial basis function neural network (RBFNN) is an effective nonlinear learning model, which has a strong nonlinear fitting capability. The hidden neurons and the weight play important roles in the neural network. In the existing researches, the hidden neurons are computed in the form of a rigorous linear combination. And the weight is hard to be solved. To address these issues, in this paper a novel neural network named non-negative radial basis function neural network (NRBFNN) is proposed. The main thought of non-negative matrix factorization (NMF) is to be used to train the parameters of RBFNN. According to the structure of the neural network, the label information of the samples is decomposed into the weight matrix and the mapped feature by activation functions in polynomial feature space. And the proposed method is able to obtain the weight matrix and the hidden neurons implied in the activation functions iteratively. Furthermore, the proposed NRBFNN is able to improve the representability of the hidden neurons and the iterative formulas of the weight can ensure the solvability and interpretability of it. ORL, Yale and Caltech 101 face databases are selected for evaluations. Experimental results show that the proposed algorithm outperforms several related algorithms.

1. Introduction

Face recognition technology is an active research area spanning several disciplines such as image processing, pattern recognition, computer vision and neural networks[1]. Recent decades, it has attracted more and more attention of relevant researchers all over the world, and it has made a great progress. Radial basis function neural network (RBFNN) has been successfully used for face recognition for its excellent approximation ability. Face recognition algorithm is based on RBFNN[2], whose hidden layer of neural network can transform data into high-dimensional space and make it linearly separable in high dimensional space. And the clustering center of RBF neural network is obtained by K-means clustering algorithm[3]. The clustering results of K-means clustering algorithm usually converge to the local extreme points, and the clustering results are strongly dependent on the first local extreme points encountered at the beginning of clustering. The difference of initial cluster center selection is likely to result in different clustering results. And the weight vector of RBF neural network is obtained by using least square method[4], which leads to a larger error with the complex
data structures. The solving process involves matrix inversion, which will increase computational costs. Specifically, when the dimension of the samples is larger than the number of samples, the solution cannot be worked out by least square method.

In recent years, many researchers propose improvements for RBFNN\[5-8\]. In the existing methods, hidden neurons are computed through a rigorously linear combination of local samples. It may impact the localization characteristics of the hidden neurons. Centering to the above problems, the proposed paper proposes a non-negative radial basis function neural network (NRBFNN) with optimal hidden neurons.

In the proposed neural network, the hidden neurons and the weight value of NRBFNN are trained with the help of the idea of NMF. NMF is an efficient feature extraction approach\[9,10\]. It can represent the sample in terms of the product of base images and the feature coefficient. And the method is able to guarantee the non-negativity of the decomposition result in the decomposition process. Specifically, instead of rigorously requiring the hidden neurons being a linear combination of the partial samples, the proposed method obtains the optimal hidden neurons iteratively, so the hidden neurons can be chosen in a larger region. And the chosen hidden neurons can be more accurate than the existing methods. Moreover, the weight value of NRBFNN is also obtained by working iteratively. And when the stop standard of iteration is reached, the proposed method can always get the weight values. After that, the proposed NRBFNN is applied to the face recognition. Meanwhile the proposed algorithm can provide an effective approach to learn the parameters of RBFNN.

Overall, the algorithm proposed in this paper has made two contributions:

1) Enhancing the accuracy of hidden neurons. The hidden neurons of the proposed NRBFNN are obtained by training iteratively, which do not depend on the first local extreme points and can be chosen at any point in space. The proposed method provides a new way for enhancing the accuracy of the hidden neurons in neural network.

2) Improving the calculability and interpretability of weight. The algorithm proposed in this paper can ensure that the weights matrix is definitely non-negative, and their values are between 0 and 1, and the sum of all weights is 1, which corresponding to the meaning of probability. And the weight of NRBFNN is computed iteratively without matrix inverse. The proposed algorithm can always obtain the value of weight.

In order to verify the recognition performance of the proposed algorithm, experiments are conducted on ORL, Yale and Caltech face database. NMF, back propagation neural network (BPNN) and RBFNN are conducted as comparative experiments. The experimental results show that the proposed algorithm has a certain improvement in recognition rate and good robustness compared with other algorithms. It is feasible and effective.

2. Radial basis function neural network

In this part, we will briefly review radial basis function neural network. RBF neural network, an artificial neural network, can be consider as a mapping: \( R' \rightarrow R' \) [2], which embraces three different layers: an input layer, a hidden layer and an output layer.

Input layer: training and testing samples.

Hidden layer: the number of hidden layer nodes depends on the requirement. Radial basic function, typically Gaussian function, as the activation function of hidden layer to transform the input information into space mapping.

Output layer: respond to input mode. The action function of the output layer neurons is a linear function. And take the weighed sum of the output information of the hidden layer as the output of the whole neural network.

The output of each RBF unit can be calculated as follow:

\[
R_i(X) = \exp\left[-\|X - C_i\|^2 / \sigma_i^2\right],
\]

where \( X \in R^{m\times r} \) is the input of RBF neural network, \( C_i \in R^{m\times r} \) is the prototype of the input vector, \( \| \cdot \| \) indicates the Euclidean norm on the input space, and \( \sigma_i \) is the width of the \( i \) th RBF unit.
3. Proposed non-negative radial basis function neural network (NRBFNN)

In this part, we will introduce the architecture and the learning method of NRBFNN. And the feature extraction of NRBFNN will be introduced at last.

3.1 Architecture of NRBFNN

The NRBFNN embraces three different layers: input layer, hidden layer and output layer, as shown in Figure 1.

![Figure 1. The structure of NRBFNN](image)

Input layer: the training samples $X = [x_1, x_2, \ldots, x_n]$.

Hidden layer: this layer consists of $k$ hidden neurons, whose activation function is radial basis functions $\phi(\cdot)$. By training the neural network, we can obtain the weight matrix $W$, the feature matrix $H = [h_1, h_2, \ldots, h_k]$, and the central matrix $U = [u_1, u_2, \ldots, u_k]$ of each neurons, where $u_j$ is the $j$th hidden neuron and $h_{ji} = \phi(\|x_i - u_j\|)$ is the $j$th element of $h_i$, $j = 1, 2, \ldots, k$, $i = 1, 2, \ldots, n$. The output of the hidden layer is the feature matrix $H$, which is computed by mapping the input samples with RBF $\phi(\cdot)$.

Output layer: the output of NRBFNN is the label of training and testing samples. It is calculated by the product of the feature matrix $H$ and the weight matrix $W$. The dimension of the output is the same as the number of class.

From the architecture of the proposed NRBFNN, the relationship among the feature of input sample $x_i$, the weight matrix $W$ and the label vector $d_i$ can be rewritten as $d_i = Wh_i$, where $d_i$ is the label information of $x_i$. For the $i$th sample: $h_i = [\phi(x_i), \phi_j(x_i), \ldots, \phi_l(x_i)]^T$. Hence, for all of the training samples:

$$D = WH,$$

which satisfies with $W \geq 0, H \geq 0$, and $W \in R^{k \times k}$, where $k$ is the number of hidden layer of neural network, $D = [d_1, d_2, \ldots, d_n] \in R^{n \times n}$ is the label matrix of samples and $H = [h_1, h_2, \ldots, h_k]$.

In order to be convenient for derivative operation, the training samples $x_i$ and the initialization of the hidden neurons $u_j$ are normalized. The distance between a sample $x_i$ and a hidden neuron $u_j$ can be rewritten in the following form:

$$\|x_i - u_j\| = (x_i^T u_j - x_i^T u_j - 2x_i^T u_j)^2 = (2 - 2x_i^T u_j)^2.$$  

In this case, the distance between the input sample $x_i$ and hidden neurons $u_j$ depends only on the value of $x_i^T u_j$. In this paper, the radial basis functions is defined as:

$$\phi(\|x_i - u_j\|) = (x_i^T u)^r.$$  

The form of RBF in this paper is the same as polynomial kernel function. It can be considered that the input samples are mapping into the polynomial feature space. And the feature of the input samples in the polynomial feature space is $h_{ji} = \phi(\|x_i - u_j\|) = (x_i^T u_j)^r$. Then $H = (U^T X)^r$, where $(\cdot)^r$ denotes the $r$th power of the matrix elements. Establishing the objective function of NRBFNN:
3.2 The learning method of the NRBFNN

The algorithm proposed in this paper needs to solve the following optimization problems:

\[
\begin{aligned}
\min_{W,U} & J(W,U) \\
\text{s.t.} & \ W \geq 0, \ U \geq 0, \ u_i^T u_i = 1, \ i = 1, 2, \ldots, k.
\end{aligned}
\]  

(6)

However, it is very hard to solve Eq.(13) directly. In order to simplify the solution process, Eq.(13) can be transformed into two sub-problems, and then solved by gradient descent method to obtain \( W \) and \( U \).

\[
\begin{aligned}
\min_{U} & J_w(U) \\
\text{s.t.} & \ U \geq 0, \ u_i^T u_i = 1, \ i = 1, 2, \ldots, k
\end{aligned}
\]  

(7)

and

\[
\min_{W} J_v(W) \\
\text{s.t.} \ W \geq 0
\]  

(8)

where \( J_w(U) \) and \( J_v(W) \) are the sub-problems when \( W \) and \( U \) keep static respectively.

First, we will solve the sub-problem (7) by gradient descent method to obtain the iterative formula of \( H \), and then we can obtain the central matrix \( U \) of each neuron hidden in \( H \).

According to gradient descent method:

\[
(U^{(t+1)}) = U^{(t)} - \rho(U^{(t)}) \otimes \nabla J_w(U^{(t)}),
\]  

(9)

where \( \rho(U^{(t)}) \) is step matrix, \( \nabla J_w(U^{(t)}) \) is the gradient of \( J_w(U^{(t)}) \) about \( U \), which can be calculated as fellow:

\[
\nabla J_w(U) = X(H^T \otimes (H^T A^T)) - X(H^T \otimes B^T),
\]  

(10)

where \( A = W^T W \), \( B = W^T D \) and \( H^* = r(U^T X)^{t-1} \). Substituting Eq.(10) for Eq.(9):

\[
U^{(t+1)} = U^{(t)} - \rho(U^{(t)}) \otimes (XH^T \otimes (H^T A^T)) + \rho(U^{(t)}) \otimes (XH^T \otimes B^T)
\]  

(11)

In order to keep \( U^{(t+1)} \) non-negative, step matrix must satisfy with the equation as follow:

\[
\rho(U^{(t)}) = U^{(t)} \otimes (X(H^T \otimes (H^T A^T))).
\]  

(12)

And we can drawn the following theorem:

**Theorem 2.1** When \( W \) keep static, the iterative formula of sub-problem (7) about objective function \( J(W,U) \) keep non-increasing, the iterative formula can be calculated as fellow:

\[
U^{(t+1)} = U^{(t)} \otimes (X(H^{(t+1)} \otimes B^{(t+1)})) \otimes (X(H^{(t+1)} \otimes H^{(t+1)} A^{(t+1)})),
\]  

(13)

where \( H = (U^T X)^n \) and \( H^* = r(U^T X)^{t-1} \). Apparently, instead of rigorously requiring the hidden neurons being a linear combination of the partial samples, the proposed method obtains the optimal hidden neurons iteratively.

Second, for sub-problems (8), we apply the same method to optimize it. And then we can drawn the following theorem:

**Theorem 2.2** When \( H \) is fixed, the iterative formula of sub-problem (8) about objective function \( J(W,U) \) keep non-increasing, the iterative formula can be calculated as fellow:

\[
W^{(t+1)} = W^{(t)} \otimes (DH^{(t)} \otimes (W^{(t)} H^{(t)} H^{(t)})).
\]  

(14)

For a testing sample set \( Y = \{ y_1, y_2, \ldots, y_l \} \), where \( l \) is the number of testing samples. And the label of testing sample can be calculated as fellow:

\[
D_y = WH_y = W(U^T Y)^\gamma.
\]  

(15)
Finally, by comparing $D_y$ with the label of test samples, the accuracy of face recognition can be calculated.

4. Experimental result

This part will verify the performance of the proposed algorithm through ORL, Yale face database and Caltech 101 face databases. The algorithms used for comparison are NMF, BP neural network and RBF neural network face recognition algorithm. The experimental results are obtained by 10 consecutive experiments under the same conditions, and then the accuracy can be obtained by calculating the average values of 10 consecutive experiments. There are 400 images of 40 individuals in ORL face database and 165 images of 15 individuals in Yale face database. The Caltech 101 face databases includes 342 images of 19 individuals. The experimental results are showed in table 1-3 and figure 2-4.

The experimental results show that the proposed algorithm achieves excellent results on 3 face databases. On ORL face databases, when the number of training samples is 2, the recognition rate of NRBFNN is 70.13%, which is slightly lower than NMF. However, the recognition rate increases with the increasing of training samples. When it is 9, the recognition rate of NRBFNN can reach 98.5%, which is higher than other 3 algorithms. NRBFNN performs well on Yale and Caltech 101 face databases too. On the whole, the algorithm proposed in this paper achieves excellent results in face recognition. It has good robustness and broad application prospects.

Figure 2. The accuracy on ORL face databases

| TN | NMF  | RBFNN | BPNN | NRBFNN |
|----|------|-------|------|--------|
| 2  | 77.22| 53.90 | 59.94| 70.13  |
| 3  | 80.00| 62.90 | 68.79| 81.32  |
| 4  | 79.17| 80.00 | 79.92| 85.13  |
| 5  | 84.50| 87.20 | 85.50| 89.45  |
| 6  | 87.50| **94.25** | 91.44| 91.81  |
| 7  | 89.10| 93.40 | 93.42| **95.00** |
| 8  | 92.50| 96.00 | 92.50| **96.38** |
| 9  | 95.00| 94.75 | 92.50| **98.50** |
Figure 3. The accuracy on Caltech 101 face databases

Table 2. The accuracy (%) on Caltech 101 face databases

| TN  | NMF  | RBFNN | BPNN  | NRBFNN |
|-----|------|-------|-------|--------|
| 3   | 62.7 | 56.56 | 35.16 | 73.83  |
| 6   | 71.80| 74.47 | 58.20 | 83.20  |
| 9   | 75.91| 73.86 | 68.19 | 87.00  |
| 12  | 76.79| 78.60 | 76.84 | 82.37  |
| 15  | 79.82| **85.26** | 84.21 | 84.04  |

Figure 4. The accuracy on Yale face databases

Table 3. The accuracy (%) on Yale face databases

| T N  | NMF  | RBFNN | BPNN  | NRBFNN |
|------|------|-------|-------|--------|
| 2    | 61.33| 56.30 | 33.78 | 67.33  |
| 3    | 67.08| 67.82 | 39.17 | 79.00  |
| 4    | **76.76** | 74.57 | 55.24 | 75.62  |
| 5    | 75.00| **82.67** | 63.11 | 82.22  |
| 6    | **84.67** | 83.87 | 70.13 | 82.26  |
| 7    | **85.83** | 83.33 | 83.00 | 85.17  |
| 8    | 84.22| 83.56 | 87.56 | **90.22** |
| 9    | 87.33| 79.33 | 88.00 | **90.33** |
| 10   | 93.33| 86.67 | 93.33 | **98.00** |
5. Conclusion
In the proposed NRBFNN, the label information of the samples is decomposed into the weight matrix and the mapped feature by activation functions in polynomial feature space. And the hidden neurons and the weight value of NRBFNN are obtained with the help of the thought of NMF, so the hidden neurons can be chosen iteratively in a larger region and the chosen hidden neurons are more accurate than the existing methods. The weight of NRBFNN is obtained by working iteratively, so the value of weight can also be computed. In conclusion, the proposed NRBFNN provides a new way for enhancing the accuracy of the hidden neurons in neural network and improves the calculability and interpretability of weight. And the experimental results show that the proposed algorithm is effective in face recognition and has great robustness and broad application prospects.

Acknowledgments
This paper is partially supported by the National Natural Science Foundation of China (Grant No.61331021), Shenzhen Science and Technology Projection (JCYJ20170818143547435, JCYJ2017081810000628, JCYJ20170302142239165, JCYJ20170302142203605) and the Postgraduate Innovation Development Fund Project of Shenzhen University(PIDFP-ZR2018005).

References
[1] R. Chellappa, C.L. Wilson, S. Sirohey, Human and machine recognition of faces: a survey. Proceeding of the IEEE, 8, 705-741(1995)
[2] J.E. Meng, S. Wu, J. Lu, H.L. Toh, Face recognition with radial basis function neural networks. IEEE Trans Neural Netw. 13, 697-710(2002)
[3] M.C. Su, C.H. Chou, A Modified Version of the K-Means Algorithm with a Distance Based on Cluster Symmetry. IEEE Transactions on Pattern Analysis and Machine Intelligence. 23, 674-680(2001)
[4] C.M. Bishop, Neural networks for pattern recognition. Agricultural Engineering International the Cigir Journal of Scientific Research & Development Manuscript Pm. 12, 1235-1242(1995)
[5] W. Huang, S.K. Oh, W. Pedrycz, Design of hybrid radial basis function neural networks realized with the aid of hybridization of fuzzy clustering method (fcm) and polynomial neural networks (pnn). Neural Networks, 60, 166-181(2014)
[6] P.S. Xiang, Survey of clustering algorithm. Journal of Southwest University for Nationalities. 37, 112-114 (2011)
[7] L.I. Hui, Y. Wang, X. Yang, R. Jia, X. Luo, Vibration fault diagnosis for hydropower generating unit based on multiwavelet and pso-rbf neural network. Journal of Northwest A & F University, 45, 227-234(2017)
[8] A. Dey, J.K. Sing, Face recognition by fuzzy generalized 2DFLD method and RBF neural network classifier. Computational Intelligence: Theories, Applications and Future Directions, 1-6(2016)
[9] D.D. Lee, H.S. Seung, Algorithms for Non-negative Matrix Factorization. In Proc. NIPS, 13, 556-562(2000)
[10] D.D. Lee, H.S. Seung, Learning the parts of objects by non-negative matrix factorization. Nature, 401, 788-791(1999)