Postprint: Application of RBFNN Based on Binary Search Density Peak Algorithm in Monthly Precipitation Forecasting

Authors: Jiang Linli, Wu Jiansheng, Ding Lixin

Date: 2018-05-20T00:00:00+00:00

Abstract

To address the limitation that the network structure and initial data centers of Radial Basis Function (RBF) networks are difficult to determine objectively, the Binary Search Density Peak Clustering Algorithm (TSDPCA) is adopted to determine the data center values and the number of data clusters, which serve as the initial parameters and the number of hidden layer nodes for the RBF neural network. The gradient descent method is then employed to optimize the RBFNN structure and its various parameters to establish a forecasting model, which is applied to monthly precipitation forecasting in Guangxi to validate the model' s effectiveness. The results indicate that, compared with the K-RBFNN and OLS-RBFNN models, the TSDPCA-RBFNN model achieves a 10%~35% reduction in the average relative error of forecasting, demonstrating better forecasting performance.

Full Text

Preamble

Application of RBFNN Based on TSDPCA in Monthly Precipitation Forecasting

Jiang Linli[1],[2], Wu Jiansheng[1], Ding Lixin[2]

[1]School of Mathematics & Computer Science, Guangxi Science & Technology Normal University, Laibin Guangxi 546199, China
[2]State Key Laboratory of Software Engineering, Wuhan University, Wuhan 430072, China

Abstract: To address the difficulty in objectively determining the structure and initial data centers of Radial Basis Function Neural Networks (RBFNN), this

paper proposes using the Two-Search Density Peak Clustering Algorithm (TSDPCA) to identify data center values and cluster counts as initial parameters for RBFNN and as the number of hidden layer nodes. The gradient descent method then optimizes the RBFNN structure and parameters to establish a forecasting model, which is applied to monthly precipitation forecasting in Guangxi to validate its effectiveness. Results demonstrate that compared with K-RBFNN and OLS-RBFNN models, the TSDPCA-RBFNN model reduces the mean relative error by 10%-35%, exhibiting superior forecasting performance.

Keywords: dichotomy; density peak; RBFNN; rainfall forecasting; gradient descent method

0 Introduction

Long-term precipitation forecasting, which predicts rainfall amounts for a month or season, holds significant importance for flood control, reservoir operation, and agricultural irrigation. In agrometeorology particularly, highly accurate long-term precipitation forecasts can generate substantial economic benefits. However, due to Guangxi' s uniquely complex terrain and variable climate, linear statistical prediction methods—commonly used in long-term weather forecasting —exhibit limitations when handling complex nonlinear rainfall problems. Consequently, investigating new techniques for meteorological long-term precipitation forecasting to improve practical forecast accuracy represents a critical research endeavor.

In recent years, neural network methods have demonstrated superior forecasting capabilities in meteorological prediction, providing novel techniques for climate analysis and forecasting in atmospheric science. Among these, Back Propagation (BP) neural networks and Radial Basis Function Neural Networks (RBFNN) are most widely applied. BP algorithms, the most common neural network training method, rely heavily on initial weight selection. Since real-world problems involve extremely complex high-dimensional surfaces with multiple local extrema, BP networks suffer from slow convergence and susceptibility to local optima. RBFNN, a three-layer feedforward neural network with simple structure, fast computation, and strong nonlinear mapping capabilities, is better suited for nonlinear time series prediction problems. Nevertheless, RBFNN performance is primarily constrained by its topological structure—specifically, the difficulty in determining three key parameters: hidden layer basis function centers, number of hidden nodes, and spread width. To date, no satisfactory method exists for objectively determining these parameters.

Many researchers have applied clustering concepts to optimize RBFNN structure design, proposing topology optimization methods based on K-nearest neighbors, K-means clustering, K-centroid clustering, Orthogonal Least Squares (OLS), and Self-Organizing Map (SOM) clustering, achieving notable results. In June 2014, Rodriguez et al. introduced a novel clustering method based on

mutual distances, utilizing Euclidean distance to compute inter-point distances and constructing decision graphs through local density and minimum distance to higher-density points. Operators then select points with high values for both attributes as cluster centers. This approach offers simplicity, computational efficiency, and the ability to identify clusters of arbitrary shapes. However, for high-dimensional complex data, Euclidean distance metrics exhibit robustness issues and ambiguous physical interpretation, leading to clustering instability.

To address these limitations, this paper proposes the Two-Search Density Peak Clustering Algorithm (TSDPCA) to automatically and rapidly obtain cluster centers and counts for optimizing RBFNN structure. First, cosine similarity measures inter-sample distances, emphasizing relative dimensional differences while overcoming Euclidean distance limitations and offering translation and rotation invariance. Second, based on a user-defined threshold, the algorithm computes local density ρ_i and corresponding distance δ_i, sequentially selecting the two largest points as cluster centers. Samples are assigned to corresponding clusters based on minimum distance to the nearest higher-density neighbor. The second cluster repeats this bisection process until inter-cluster variation becomes minimal, at which point clustering terminates, preserving center values and cluster counts. Finally, TSDPCA and gradient descent hybrid optimization establishes the forecasting model, applied to monthly precipitation prediction in Guilin, Guangxi, demonstrating rapid network convergence and high forecasting accuracy.

1.1 RBFNN Structure

RBFNN is a three-layer feedforward neural network comprising an input layer, hidden layer, and output layer. Its network topology is shown in [Figure 1: see original paper]. The fundamental principle employs Radial Basis Functions as hidden layer unit "bases," transforming low-dimensional input data into high-dimensional space automatically. RBFNN activation functions use Gaussian functions, expressed mathematically as:

$$\phi_j(X) = \exp\left(-\frac{\|X - C_j\|^2}{2\sigma^2}\right), \quad j = 1, 2, \ldots, N$$

where X is input data, $\|\cdot\|$ denotes Euclidean norm, C_j is the Gaussian function center, and σ is the Gaussian function variance.

For input sample X_i, the expected RBFNN output is:

$$Y_i' = F(X_i) = \sum_{k=1}^{K} \omega_{kj} \exp\left(-\frac{\|X_i - C_k\|^2}{2\sigma_k^2}\right)$$

where ω_{ij} represents connection weights from hidden layer to output layer.

1.2 RBFNN Gradient Descent Learning

Gradient descent optimizes hidden node data centers, spread constants, and hidden-to-output layer weights by minimizing the objective function. The network learning objective function is:

$$E = \frac{1}{2}\sum_{i=1}^{N} e_i^2$$

where $e_i = Y_i - Y_i' = Y_i - F(X_i)$ is the error between actual value Y_i and expected output, and β is a forgetting factor.

The iterative learning methods for centers C_i, widths γ_i, and weights ω_i are:

$$C_i(t+1) = C_i(t) + \Delta C_i = C_i(t) + \gamma_i \cdot \omega_i \cdot e_i \cdot \phi_i' \cdot (X_i - C_i)$$

$$\gamma_i(t+1) = \gamma_i(t) + \Delta\gamma_i = \gamma_i(t) + \gamma_i \cdot \omega_i \cdot e_i \cdot \phi_i' \cdot \|X_i - C_i\|^2$$

$$\omega_i(t+1) = \omega_i(t) + \Delta\omega_i = \omega_i(t) + \gamma_i \cdot e_i \cdot \phi_i$$

2 TSDPCA-Based RBFNN Model

2.1 Two-Search Density Peak Clustering Algorithm

Literature [19] proposed a fast clustering algorithm based on density peaks and distances, where the distance sample dataset D comprises all points and pairwise distances forming a three-dimensional matrix. Using Euclidean distance to compute inter-point distances primarily reflects absolute numerical differences rather than dimensional variations. This paper employs cosine similarity to compute inter-point distances, which is insensitive to absolute values and distinguishes differences from directional or relative dimensional perspectives. For an $n \times d$ sample dataset $X = \{X_1, X_2, \ldots, X_n\}$, cosine similarity constructs the three-dimensional distance matrix D^* as:

$$D^* = \begin{bmatrix} 0 & d_{12} & d_{13} & \cdots & d_{1n} \\ d_{21} & 0 & d_{23} & \cdots & d_{2n} \\ d_{31} & d_{32} & 0 & \cdots & d_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ d_{n1} & d_{n2} & d_{n3} & \cdots & 0 \end{bmatrix}$$

Cosine similarity is mathematically expressed as:

$$d_{ij} = 1 - \frac{\sum_{a=1}^{d} X_{ia} X_{ja}}{\sqrt{\sum_{a=1}^{d} X_{ia}^2} \sqrt{\sum_{a=1}^{d} X_{ja}^2}}$$

The clustering algorithm operates on two premises: (1) cluster centers have higher local density than surrounding samples, and (2) cluster centers have relatively large distances to points with higher local density. This method can detect clusters of arbitrary shapes. Based on density and distance, the algorithm constructs decision distribution graphs, selecting density peaks in the upper-right region—points clearly distant from most samples—as initial cluster centers. Samples are assigned to corresponding clusters based on minimum distance to the nearest higher-density neighbor. Although simple and effective, this method requires manual cluster center selection, which may yield inconsistent results for the same dataset and can incorrectly split one class into multiple clusters.

To avoid redundant cluster centers, this paper comprehensively considers both ρ_i and δ_i values, proposing to terminate density peak searching when inter-cluster variation falls below 0.01. The parameters and clustering steps are as follows:

Local density is defined as:

$$\rho_i = \sum_{j=1}^{n} \chi(d_{ij} - d_c)$$

where $\chi(x) = 1$ if $x < 0$ and 0 otherwise. This counts samples within distance d_c of point i, where d_c is the cutoff distance. The principle for selecting d_c ensures each data point has approximately 1%-3% of the total points as "neighbors." This paper sets d_c as the distance at the 2% percentile when sorting all pairwise distances from smallest to largest.

Distance to higher-density points is defined as:

$$\delta_i = \min_{j:\rho_j > \rho_i} (d_{ij})$$

For the point with maximum density, $\delta_i = \max_j(d_{ij})$. For standardization, δ_i values are normalized to [0,1].

Inter-cluster variation is:

$$\lambda_i = \rho_i \cdot \delta_i$$

Larger λ_i values indicate more likely cluster centers. After sorting λ_i values, the two largest samples are selected as cluster centers. Samples are assigned to two clusters based on minimum distance to the nearest higher-density neighbor. The first cluster's center and data are preserved, while the second cluster repeats the

bisection process until the termination condition is met, ultimately obtaining cluster centers C_k and count k.

The algorithm steps are:

Input: Experimental sample dataset $X = \{X_1, X_2, \dots, X_n\}$
Output: Cluster center values $C = \{C_1, C_2, \dots, C_k\}$, where k is cluster count.

a) Compute pairwise distances d_{ij} using Equation (8) and construct three-dimensional matrix D^* from Equation (9).

b) Compute local density ρ_i and minimum distance δ_i to higher-density points for each sample using Equations (10) and (11).

c) Calculate λ_i values, sort descending, and select the two largest samples as cluster centers. Assign samples to corresponding clusters based on nearest higher-density minimum distance. Preserve the first cluster' s center and position.

d) Compute inter-cluster error $|\lambda_{ij} - \lambda_i|$.

e) Repeat steps b)-d) on the remaining cluster until satisfying the termination condition $|\lambda_{ij} - \lambda_i| \leq 0.01$.

f) Obtain cluster centers C_k and count k, extracting corresponding samples from X using center positions as indices.

The resulting cluster centers and count serve as RBFNN basis function centers and hidden node numbers.

2.2 TSDPCA-Based RBFNN Prediction Model Establishment

RBFNN model parameters include:
- Input neuron count: 10 precipitation factors as input nodes
- Hidden neuron count: Determined through numerical experiments and empirical formulas; this paper uses the cluster count obtained above
- Spread constant: Computed as minimum distance between hidden node data centers:

$$\sigma_k = \min_{j \neq k} \|C_j - C_k\|$$

The TSDPCA-RBFNN model establishment comprises four steps:

a) Process precipitation factors using Mean Generating Function (MGF) and Singular Spectrum Analysis (SSA), then apply Principal Component Analysis (PCA) to extract 10 comprehensive precipitation forecasting factors. Normalize to [0,1] as RBFNN training input matrix.

b) Compute inter-sample distances using cosine similarity and construct three-dimensional matrix D^*.

c) Apply TSDPCA clustering to obtain data center values and counts.

d) Compute hidden layer spread width using Equation (14), then establish forecasting model using gradient descent optimized Gaussian RBF.

The TSDPCA-RBFNN model development process is illustrated in [Figure 2: see original paper].

3 Experiments and Analysis

3.1 Study Area and Data

Using monthly real-time rainfall data (1949-2008) from ten Guilin meteorological stations, we establish a TSDPCA-RBNN prediction model. The dataset comprises 720 training samples with 360 rainfall influence factors. Data from 1949-2005 (684 months) serve as training set, while 2006-2008 (36 months) constitute the test set.

3.2 Data Preprocessing

The 360 precipitation factors are reconstructed from original precipitation sequences using Singular Spectrum Analysis (SSA) and extended via Mean Generating Function (MGF). PCA extracts the top 10 strongest comprehensive precipitation factors (80% variance) as RBFNN input matrix vectors. Since RBFNN requires input/output data in [0,1], the processed precipitation samples undergo normalization.

3.3 Parameter Settings

Experiments run on Windows 7 with MATLAB 7. The initial "neighbor" count d_c is set to 2% (values between 1%-3% yield similar results). TSDPCA automatically obtains 14 data centers. RBFNN training parameters: hidden node center learning coefficient lrCent=0.01; spread constant learning coefficient lrSP=0.01; output weight learning coefficient lrW=0.01; hidden node count equals cluster center count; target error=0.9; maximum training epochs=5000.

3.4 Results Analysis

[Figure 3: see original paper] shows monthly average precipitation in Guilin from 1949-2008, revealing maximum average rainfall of approximately 250mm and minimum of 50mm. Monthly precipitation 250mm indicates catastrophic flooding, while 50mm indicates drought.

[Figure 4: see original paper]-[Figure 6: see original paper] compare monthly precipitation forecasts (2006-2008) from three models against observed data. All

three models generally follow observed trends, but TSDPCA-RBFNN shows the best performance with minimal deviation and strong consistency.

To evaluate performance comprehensively, we compare TSDPCA-RBFNN against K-means RBFNN (K-RBFNN) and Orthogonal Least Squares RBFNN (OLS-RBFNN) using five evaluation metrics from . All RBFNN models use gradient descent for structure optimization.

[Figure 7: see original paper] shows TSDPCA classification results sorted by λ_i, demonstrating that cluster count stabilizes at 14, indicating this as an appropriate cluster number. Using minimal inter-cluster variation as termination criterion enables automatic cluster count and center acquisition.

Performance statistics in reveal TSDPCA-RBFNN achieves the smallest mean relative error, RMSE, and MAE across 2006-2008 test data. Compared to K-RBFNN, mean relative error decreases by ~10%; compared to OLS-RBFNN, it decreases by 35%. This substantial improvement confirms TSDPCA-RBFNN' s enhanced prediction accuracy.

Convergence speed comparison shows TSDPCA-RBFNN training time of 18.6345 seconds with 10 convergence epochs; OLS-RBFNN requires 61.6801 seconds and 180 epochs; K-RBFNN needs 44.6128 seconds and 362 epochs. Thus, TSDPCA-RBFNN converges significantly faster.

For disaster prediction, monthly rainfall 250mm indicates catastrophic flooding. [Figure 4: see original paper] shows actual flooding in April 2006 and drought in January, October, November, and December; TSDPCA-RBFNN predictions match exactly. [Figure 5: see original paper] indicates predicted flooding in February, April, and May 2007, with drought in August and September; actual flooding occurred in April, drought in August and September. Detailed comparison reveals February and May actual rainfall exceeded 200mm, indicating heavy precipitation, so predictions closely match reality. [Figure 6: see original paper] shows 2008 predicted flooding matches actual April flooding, with drought predictions also closely aligned. Across all three years, TSDPCA-RBFNN demonstrates higher accuracy and stability, providing valuable reference for flood/drought disaster prevention and runoff forecasting.

4 Conclusion

Precipitation exhibits strong nonlinear characteristics due to multiple influencing factors. RBFNN' s robust nonlinear modeling capability, parameter efficiency, and self-learning adaptability make it suitable for meteorological factor modeling and forecasting. However, practical applications typically rely on manual trial-and-error for structure design and parameter tuning, requiring time-consuming experimental exploration to identify appropriate network architectures.

This paper proposes TSDPCA to automatically, rapidly, and accurately obtain cluster centers and counts, overcoming RBFNN limitations. Combined with gra-

dient descent for structure optimization, experimental results yield the following conclusions:

a) **Convergence Speed:** Compared with traditional K-means RBFNN and OLS-RBFNN, TSDPCA-RBFNN offers advantages in determining cluster count and basis function centers, with significantly improved convergence speed and prediction accuracy.

b) **Forecasting Accuracy:** For Guangxi monthly precipitation forecasting, the improved RBFNN substantially enhances mean relative error accuracy. Predictions for 2006-2008 flood and drought events closely match actual observations, providing a new reference method for long-term precipitation forecasting with strong nonlinear uncertainties.

c) **Future Work:** This study employs cosine similarity instead of Euclidean distance to compute inter-sample distances for monthly precipitation modeling. Further research could analyze data similarity more deeply and integrate intelligent optimization algorithms to improve hidden node count determination. Additionally, the λ threshold determination method requires further investigation.

References

[1] Jin L, Luo Y, Lin Z. Comparison of long-term forecasting of june-august rainfall over changjiang-huaihe valley [J]. Advances in Atmospheric Sciences, 1997, 14(1): 88-95.

[2] Jiang Linli, Wu Jiansheng. Hybrid evolutionary algorithms for artificial neural network training in rainfall forecasting [C]//Proc of International Conference on Advances in Neural Networks. 2013: 191-195.

[3] Wu J, Long J, Liu M. Evolving RBF neural networks for rainfall prediction using hybrid particle swarm optimization and genetic algorithm [J]. Neurocomputing, 2015, 148: 136-142.

[4] Hu J, Zhou Y, Jin J. Application of BP neural network flood forecasting model in flood forecasting system [J]. Hydrology, 2015, 35(1): 20-25.

[5] Min J, Sun J, Liu H, et al. An improved BP algorithm and its application in precipitation forecasting [J]. Journal of Applied Meteorological Science, 2010, 21(1): 55-62.

[6] Yaseen Z M, El-Shafie A, Afan H A, et al. RBFNN versus FFNN for daily river flow forecasting at Johor River, Malaysia [J]. Neural Computing and Applications, 2016, 27(6): 1-10.

[7] Cao C, Wang S, Tang H. Rainfall forecasting using RBFNN optimized by FKCN [J]. Journal of Hefei University of Technology, 2000.

[8] He X, Guan H, Qin J. A hybrid wavelet neural network model with mutual information and particle swarm optimization for forecasting monthly rainfall [J].

Journal of Hydrology, 2015, 527(17): 88-100.

[9] Jiang Linli. Precipitation forecasting model based on hybrid optimized RBF neural network ensemble [J]. Journal of Liuzhou Teachers College, 2012, 27(2): 113-119.

[10] Long W, Liang X, Long Z, et al. RBF neural network time series prediction based on hybrid evolutionary algorithm [J]. Control and Decision, 2012, 27(8): 1265-1268.

[11] Xie J, Gao R. K-medoids clustering algorithm optimized by Num-nearest neighbor variance [J]. Application Research of Computers, 2015, 32(1): 30-34.

[12] Shen Y, Yang C, Zhang Q, et al. Precipitation sequence prediction in Chizhou based on RBF neural network [J]. Journal of Anhui Agricultural University, 2012, 39(3): 451-455.

[13] Liu B, Xiao C, Liang X. Application of SOM-RBF neural network model in groundwater level prediction [J]. Journal of Jilin University: Earth Science Edition, 2015, 45(1): 225-231.

[14] Yi Y, Lu W, Zhang Y, et al. Research on surrogate model of groundwater numerical simulation based on radial basis function neural network [J]. Research of Soil and Water Conservation, 2012, 19(4): 265-269.

[15] Zhou W, Shi Y. Optimization algorithm for selecting clustering centers based on density K-means [J]. Application Research of Computers, 2012, 29(5): 1726-1728.

[16] Rodriguez A, Laio A. Clustering by fast search and find of density peaks [J]. Science, 2014, 344(6191): 1492-1496.

[17] Xie J, Qu Y. K-medoids clustering algorithm with density peak optimized initial centers [J]. Computer Science and Exploration, 2016(2): 1-16.

[18] Zhu Y, Feng W, Guo J, et al. An improved K-centroid clustering algorithm and its application in thunderstorm clustering [J]. Journal of Wuhan University: Science Edition, 2015, 61(5): 497-502.

[19] Guo H, Xing Z, Fu Q, et al. RBF network based on density parameter K-means algorithm and its application in precipitation prediction [J]. Research of Soil and Water Conservation, 2014, 21(6): 299-303.

[20] Wu J. Prediction of rainfall time series using modular RBF neural network model coupled with SSA and PLS [C]//Proc of Asian Conference on Intelligent Information and Database Systems. 2012: 509-518.

Note: Figure translations are in progress. See original paper for figures.