Study of Applications of Radial Basis Function Network in Forecasting

Akhilesh Kumar Yadu[1,*], Gyanesh Shrivastava[2,*]

[1]MATS University, Pandari, Raipur, Chhattisgarh 492004, India
[2]School of Information Technology, MATS University, Pandari, Raipur, Chhattisgarh 492004, India

*Corresponding authors: E-mail: yadu.akhilesh4@gmail.com; gyaneshnkshrivastava@gmail.com

DOI: 10.5185/amlett.2021.051628

Forecasting is a broad research domain for multidisciplinary researchers. Forecasting climatic conditions is a challenging task that is carried out by scientists around the globe. The researchers for Forecasting of the various climatic phenomena have used multiple architectures of Artificial Neural Networks (ANN). Such tasks demand huge analysis on recent and past results to give appropriate with precise results of the forecast. The RBF network has a single hidden layer, simpler structure as well as a much faster training process; therefore, it is a popular alternative to the other ANN architectures. In this paper, we are proposing a Radial Basis Function Network (RBFN) as a machine learning tool for making forecasting. In a variety of real-time packages, that includes the prediction of weather, load forecasting, forecasting approximately a variety of traveller and in various programs RBFN applied.

Introduction

A "Radial Basis Function Network" (RBFN) is a selected form of Artificial Neural Network. In general, the RBFN consists of three layers: first, an input layer, then an activation function called the hidden layer for non-linear RBF, and a linear output layer. RBFN's convergence of optimization objective is much faster because it has only one hidden layer. Despite having one hidden layer, it proved that RBFN is suitable for universal approximations. It has many applications like function approximation, interpolation, time series prediction, and classification. In comparison to MLP, the RBFN approach is more intuitive.

We required specifying the activation function of the hidden unit to apply and to use an RBFN. Finding the weights of RBFN is referred to as network training. If we have at hand a group of input-output pairs, known as a training set, we optimize the network standards on the way to match the network out-puts to the given in-puts. The RBF network may be used later in training for data where the underlying data is much like the training set.

We know that human society is affected by Climate and weather in all possible ways. Production of Crop in agriculture, the most significant factor for water resources i.e., Rain, water elements, and the extent of these elements increases or decreases because of change in the climate. The impact of frost on the quality of crops and growth is a principal perspective to total harvest failure. One such domain is Meteorology, where data mining can give better productivity to its analysts by changing their unmanageable, voluminous, and prone to ignorance data into usable bits of information [**16**].

Weather Forecasting started with early human advancements and depended on recurring meteorological and astronomical events. These days, the predictions of weather are made by gathering information regarding the present status of the environment and utilizing a logical framework to anticipate how the environment will develop. The unpredictable behavior of the environment, the enormous computational power is required to solve all the climate-representing equations, means that forecasts are less reliable and too expensive as the sequence of predictions improve, which takes us to a situation where we need to think about better weather forecast approaches that can be more efficient or potentially more affordable [**20,25**].

Accurate prediction of weather has been one of the most important issues in weather research; because early warning of intense weather, made possible through prompt and exact forecasting can assist prevent fatality and destruction caused by natural disasters. In the most recent years, analysts have started to examine the capability of ANN as a simulation tool for the behavior of weather, but it needs a huge analysis of past and recent results to give accurate forecasted results. In this regard, a popular alternative to the ANN architecture is RBFN, which is a much faster training process [**9,17**].

Recently, the idea of coupling various prototypes has fascinated additional attention in hydrologic estimating. They can be extensively classified into group models and particular (or hybrid) models. The fundamental thought at the back of the ensemble prototype is to make a number of dissimilar or equal models for a similar procedure and to combine them [**3**,**23**]. Likewise, it has been applied in various regions such as identification, modeling, prediction, and optimization. Weather forecasting is accomplished by the use of AI techniques and their tools [**11**,**12**].

In the study of RBFN, Almita, [**1**,**10**], have found that the proposed adaptive RBFNN model increases the reliability of network systems. Forecasting looks like to be a simple process but is not. It is based on lots of analysis on current and past results that can give accurate forecasted results in a specified time [**2**,**4**]. The RBFN has the property of value decomposition for solving for the weights of the network [**5**]. Radial Basis Function can deal with other meteorological data such as temperature forecasting, heavy rain, daily rain, etc. [**6-8**]. In another study and research, Praynlin and Latha, [**13**], Santhanam and Subhajini, [**15**], Saba *et. al.*, [**14**], have compared the RBFN and BPN network model and found that the RBFN is a suitable function for forecasting.

Architecture of RBFN

The RBF network is an "Artificial Neural Network" that consists of an input layer, a hidden layer, and an output layer [**18**]. The second layer, known as "the Hidden RBF layer," consists of hidden neurons, and a Gaussian function is the activation function of these neurons. A signal corresponding to an input vector is generated by the input layer as well as a hidden layer of the RBF network and a response is generated corresponding to that signal network [**19**].

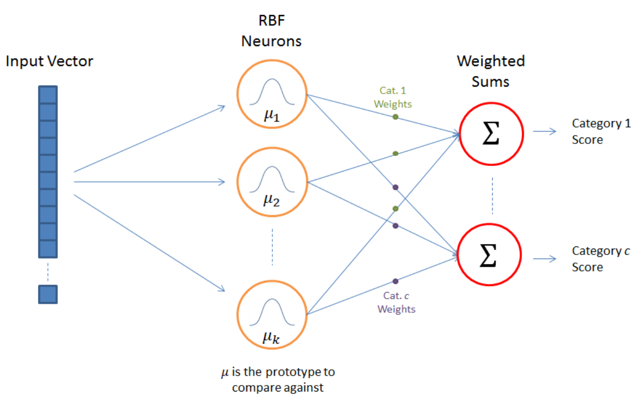

Fig. 1. Typical architecture of an RBF Network.
(Image source: http://chrisjmccormick.files.wordpress.com/2013/08/architecture_simple2.png)

The typical architecture of an RBF Network has been presented in the above-shown example. In the above RBF Network, we can see that it is composed of an input vector including with RBF neuron layer and with one node per category which is generally called an output layer. The above-shown figure input vector has the n-dimensional vectors that offer to categorize [**21**].

An input vector

Input vectors are the multi-dimensional vectors that attempt to systematize, and complete input vectors are shown towards every RBF neuron.

An RBFN

Each RBF neuron stores a "prototype" vector, which is only one in the training set of the vectors. The prototype of every RBF neuron has been compared with the input vector and the out-put a value in the range from 0 to 1 which calculates the similarity. The RBF Neuron produces 1 output when the given input is similar to the prototype. The result moves exponentially towards 0, only if the distance grows between the input and prototype. As explained and shown in the "network architecture diagram", the response of the RBFN neuron shape is similar to the bell curve [**22**].

The response value of the neuron is also termed as its "value of activation".

A model vector is generally known as the Neuron Centre because it is a value in the middle of a bell curve.

An output node

An output of a network is comprised of the group of nodes, each of the ranges that we are attempting to classify. Every out-put node enumerates such a score for an adjuvant classification. Generally, with the highest score, a categorization decision is prepared by assigning inputs to the category. For the computation of the result, the weighted amount of activation values has been taken from each RBFN neuron. By weighted amount, it implies that an out-put node adjuvant the weight value with every of the RBFN neurons and increases the neuron's actuation with present weight prior to count it to the overall reaction [**24**].

Every out-put node has its own arrangement of weights, on the grounds that the out-put node computes the score for the odd categories. The out-put node, which belongs to the RBF neurons, will typically give a "positive weight", and the "negative weight" to others.

Activation function of RBF neuron

There is a computed scale of equality amongst input and its model vector of each RBF neuron. Input vector gives the result closer to 1 when the input is more similar to the prototype [**26**,**27**]. For the similarity function, there are various possible options; However, the most famous is depends on "the Gaussian". The calculation of an equation to the Gaussian with 1-dimensional in-put is given as follows:

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

where x is input, μ is the standard deviation of mean and sigma. This produces the familiar bell curve shown below, which focuses on the mean:

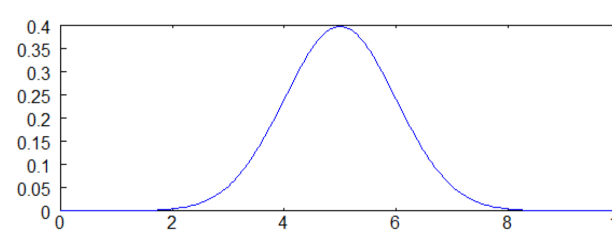

Fig. 2. Bell curve.
(Image source: http://chrisjmccormick.files.wordpress.com/2013/08/architecture_simple2.png).

The first transformation is that we have eliminated the outer coefficient $\frac{1}{\sigma\sqrt{2\pi}}$. The height of the Gaussian is controlled by this term normally. Here, although, it is unnecessary with the applied weights by the out-put nodes. For the period of training, the out-put nodes will be trained the right coefficient or weight to enforce the "neuron's reaction".

The subsequent transformation is that we have changed an internal coefficient $\frac{1}{2\sigma^2}$, with a solitary parameter 'beta'. Bell curve's width is controlled by the beta coefficient. Thereafter, in this reference, we couldn't care less about the estimation of sigma, we simply supervise that there are few coefficients, which controls bell curve's width. So, the equations are simplified via changing the term by unique variable.

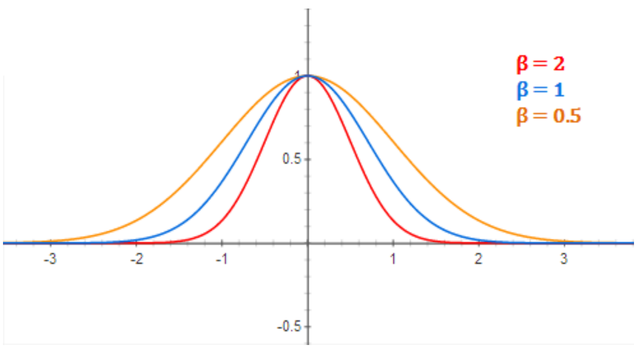

Fig. 3. RBF Activation of the Neuron for various beta values.
(Image source: http://chrisjmccormick.files.wordpress.com/2013/08/architecture_simple2.png).

In addition, each RBF neuron produces its greatest response when input is the same as the vector of prototype. This makes it possible to take it as 'a degree of similarity', and summing the effects of all RBF neurons.

The response falls off exponentially as we circulate out from the 'prototype vector'. "Recall from the instance of the RBFN structure that only the "output node of each category takes different phrases to the weighted sum of each community RBF neuron as well as any community neuron can have an effect on the class set. The 'exponential decrease' of the activation function, however, implies that the neurons whose 'prototypes' are a long way from the entry vector will truly contribute only slightly to the outcome.

Structure and algorithms of RBF

Because of the broad variety of adaptability and learning potential, ANN has been extensively used inside the nonlinear system's prediction.

The feed-forward neural community is the maximum extensively used ANN. At gift, the BPN network to get extra programs within the prediction of the earthquake, but the convergence of the BPN community gaining knowledge of the system is intently associated with the initial fee. The RBF network is an excellent performance of the feed-ahead synthetic neural community.

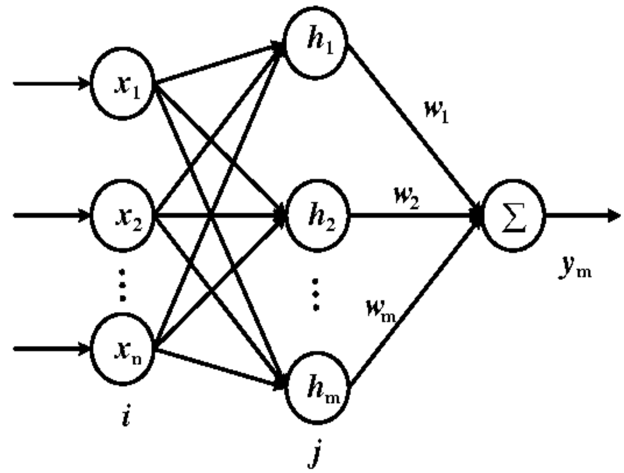

Fig. 4. RBF Neural network structure.
(Image source: Speed Sensorless Control of PMSM using Model Reference Adaptive System and RBFN).

A 3-layer feed-forward community is the 'single hidden layer' is characteristic of the radial basis neural network. Mapping from input to output is nonlinear while maps from hidden layer field to output space are linear, so it can significantly improve velocity sensitivity and stay away from local minima.

In the composition of the RBF network, X = [x1, x2, …, xn] T is the vector for input. Two inputs are available, which are q-axis current deviation rate of charge Δe.

As the membership function, the Gaussian function is introduced as follows:

$$h_j = \exp\left(\frac{\|X - C_j\|^2}{2b_j^2}\right), j = 1, 2, \dots, m$$

where the B = [b1, b2, …, bm] T and Cj = [cj1, cj2, …, cjm] T, respectively, is the standard deviation and mean.

The input layer to hidden layer weight value is 1.0, and the weight vector from hidden layer to output layer is

$$W = [w_1, w_2, \dots, w_m]^T$$

The RBF network output is

$$Y_m(k) = w_1h_1 + w_2h_2 + \dots + w_mh_m$$

The energy function is defined as

$$E(k) = \frac{1}{2}[i_q(k) - \hat{\imath}_q(k)]^2 = \frac{1}{2}e(k)^2$$

And the learning algorithms about the weight value, standard deviation and mean are as follows:

$$w_j(k) = w_j(k\text{-}1) + \eta\,[i_q(k) - \hat{\imath}_q(k)]\,h_j + \alpha[w_j(k\text{-}1) - w_j(k\text{-}2)]$$

$$\Delta b_j - [i_q(k) - \hat{\imath}_q(k)]\,w_jh_j\frac{\|X - C_j\|^2}{b_j^3}$$

$$b_j(k) = b_j(k\text{-}1) + \eta\Delta b_j + \alpha[b_j(k\text{-}1) - b_j(k\text{-}2)]$$

$$\Delta c_{ji} = [i_q(k) - \hat{\imath}_q(k)]\,w_jh_j\frac{x_i - c_{ji}}{b_j^2}$$

$$c_{ji}(k) = c_{ji}(k\text{-}1) + \eta\Delta c_{ji} + \alpha[c_{ji}(k\text{-}1) - c_{ji}(k\text{-}2)]$$

where η is the learning rate and α is the momentum factor.

Training of network

The elements that should be decided for the development of an RBF network are:

- Choosing a suitable value for the center

  There are a variety of techniques, which may be applied for identical, which includes:
  - o Randomly selected from data set
  - o K means algorithm
  - o OLS Algorithm
- Deciding the role of activation function to be enforced in the hidden layer (Linear, Gaussian)
- Deciding the role of activation function to be enforced in the hidden layer (Linear, Gaussian)
- Adjusting the output weights (Gradient Descent, Least Square, etc.) where
- Deciding the value of bias/spread

There is a need to carry out normalization as well. It can be possible to do this by using

$$\bar{Y} = \frac{Y - Y_{min}}{Y_{max} - Y_{min}}$$

where Y is the actual value of the data sample
Y_{max} takes the value larger than the foresting year
Y_{min} takes the value that is minimum from the sample of data

Conclusion

The RBF network has a much quicker training system. We have included a survey of the RBF community in this paper. A variety of factors of the RBF community were defined, with emphasis located on RBF community getting to know and network shape optimization. The topics of RBF networks with normalized networks, RBF networks in 'dynamic systems modeling' and 'complex RBF networks' are defined in order to managing nonlinear complex-valued indicators.

We find that the outcomes produced by RBF are not ideal before normalization. We can not only improve the precision of the RBF neural network but also diminish time intricacy by using the normalization method. We can eliminate unessential features and decrease network complexity. Thus, it becomes easier to train the RBF network more accomplished and easily in the classification of forecasting datasets with the best results. Like there is a lot of research about RBF network methods, researchers are interested in making use of neural networks for other purposes also. Therefore, there is a lot of research work there with neural networks as future work.

Keywords

Forecast, ANN, RBFN.

Received: 17 May 2020
Revised: 7 January 2021
Accepted: 10 January 2021

References

1. Almaita, E.; *IOSR Journal of Computer Engineering*, **2017**, *19*, 33.
2. Arora, Y.; Singhal, A.; Bansal, A.; *International Journal of Computer Applications*, **2014**, *94*, 17.
3. Ayodele, A. P.; Precious, E. E.; *Asian Journal of Research in Computer Science*, **2019**, *3*, 1.
4. Biswas, R. K.; Application of Radial Basis Function Network (RBFN) for Water Levels Prediction in the Surma River of Bangladesh; 3rd International Conference on Water & Flood Management, **2011**.
5. Chen, S.; Cowan, C. F. N.; Grant, P. M.; *IEEE Transactions on Neural Networks*, **1991**, *2*, 302.
6. Feghi, I. E.; Zubi, Z. S.; Abozgaya, A.; *Recent Advances in Image, Audio and Signal Processing*, **2014**, 99-104.
7. Gao, W.; Guo, Z.; *Journal of Networks*, **2013**, *8*, 213.
8. Hota, H. S.; Handa, R.; Shrivas, A. K.; *International Journal of Computational Intelligence Research*, **2017**, *13*, 1145.
9. Krasnopolsky, V. M.; Rabinovits, M. S. F.; Tolman, H. L.; Belochitski, A. A.; Neural Network approach for robust and fast calculation of physical processes in numerical environmental models: Compound parameterization with a quality control of large errors; *Science Direct*, **2008**, 535-543.
10. Kumar, S.; Indian. A.; Khan, Z.; *International Journal of Soft Computing and Engineering*, **2013**, *3*, 195.
11. Mazima, J. K.; Johnson, A.; Manasseh, E.; Kaijage, S.; *International Journal of Artificial Intelligence and Applications*, **2016**, *7*, 27.
12. Pallavi; Singh, G.; *International Journal of Innovative Research in Computer and Communication Engineering*, **2016**, *4*, 2901.
13. Praynlin, E.; Latha, P.; *International Journal of Compute, Electrical, Automation, Control and Information Engineering*, **2014**, *8*, 258.
14. Saba, T.; Rehman, A.; AlGhamdi, J. S.; Weather forecasting based on hybrid neural model; *Springer*, **2017**, 3869-3874.
15. Santhanam, T.; Subhajini, A. C.; *Journal of Computer Science*, **2011**, *7*, 962.
16. Sawale, G. J.; Gupta, S. R.; *International Journal of Computer Science and Applications*, **2013**, *6*, 383.
17. Shafie, A. H. E.; Shafie, A. E.; Mazoghi, H. G. E.; Shehata, A.; Taha, M. R.; *International Journal of the Physics Sciences*, **2011**, *6*, 1306.
18. Sossa, H.; Cortes, G.; Guevara, E.; New Radial Basis Function Neural Network Architecture for Pattern Classification: First Results; Springer International Publishing Switzerland, **2014**, 706-713.
19. Thandar, A. M.; Khaing, M. K.; *International Journal of Computer Applications*, **2012**, *54*, 20.
20. Tiwari, M. K.; Chatterjee, C.; *Journal of Hydroinformatics*, **2011**, 500-519.
21. Uykan, Z.; Ertugrul, M.; Koivo, H. N.; *IEEE Transactions on Neural Networks*, **2000**, *11*, 851.
22. Vivekanandan, N.; *International Journal Advanced Networking and Applications*, **2014**, *5*, 1974.
23. Wu, C. L.; Chau, K. W.; and Fan, C.; *Journal of Hydrology*, **2010**, *389*, 146.
24. Wu, Y.; Wang, H.; Zhang, B.; Du, L. K.; Using Radial Basis Function Networks for Function Approximation and Classification; *International Scholarly Research Network*, **2012**, 1-34.
25. Zaytar, M. A.; Amrani, C. E.; *International Journal of Computer Applications*, **2016**, *143*, 7.
26. Zhang, J.; Baqai, W.; Knoll, A.; A Comparative Study of B-Spline Fuzzy Controller and RBFN; Proceedings of the Fourth European Workshop on Fuzzy Decision Analysis and Recognition Technology, **1999**, 1-9.
27. Zheng, Z. W.; Chen, Y. Y.; Zhou, X. W.; Huo, M. M.; Zhao, B.; Guo, M. Y.; *International Journal of Smart Grid and Clean Energy*, **2012**, 192.