A New Heuristic for Radius Estimation in RBF Neural Networks

Lucas Pimenta de Souza, Paulo V. C. Batista, Petrônio C. L. Silva

[1]Grupo de Pesquisa em Ciência de Dados e Inteligência Computacional - $\{ci\partial ic\}$
Instituto Federal do Norte de Minas Gerais (IFNMG) - Campus Januária
Fazenda São Geraldo, S/N Km 06 - 39480-000 - Januária /MG
lpds1@aluno.ifnmg.edu.br, paulo.batista@ifnmg.edu.br, petronio.candido@ifnmg.edu.br

Abstract. *RBF-based neural networks models (RBFNN) are classic machine learning methods which contains a layer of Radial Basis Functions (RBF) which act as the feature extractor for the final layer, which performs the pattern recognition. The estimation of RBF radius during the model training is one of the most crucial tasks for this method and directly affects its generalization power and accuracy. This works presents a new heuristic for radius estimation and employ computational experiments to assess its performance against other approaches using 14 classification problems. The proposed method is shown competitive performance, winning the other approaches in 9 ou of 14 problems.*

Resumo. *Redes Neurais baseadas em Funções de Base Radial (RBFNN) são métodos clássicos do aprendizado de máquina que contêm uma camada de Funções de Base Radial (RBF) que atuam como extrator de características para a camada final, que executa o reconhecimento de padrões. A estimação do raio das RBFs é uma das atividades mais cruciais do treinamento de modelos RBFNN e afeta diretamente o seu poder de generalização e acurácia. Neste trabalho é apresentado uma nova heurística para estimação do raio e experimentos computacionais são empregados para medir sua eficácia comparada à outras abordagens usando 14 problemas de classificação. A método proposta mostrou uma eficácia competitiva, vencendo os demais métodos em 9 dos 14 problemas.*

1. Introduction

The Radial Basis Functions Neural Networks (RBFNN) are a widespread shallow architecture of neural network that gather some key desirable properties as simplicity, flexibility and explainability. The core idea of RBFNN is the employment of Radial Basis Functions (RBF) in their hidden layer, where each neuron represents a center, i. e., an interest point in feature space and its neighborhood, and the RBF measures the distance of an input data x_i with these centers. Then, each neuron (with its center and width) represent the features of one or more patterns, and the distance of the input data from these centers means the adherence or not to those patterns. Finally, the output layer is responsible to combine the distances from the centers and recognize the patterns. RBF Networks are universal aproximators and can handle linear and nonlinear problems properly.

The first RBFNN models were proposed more than 30 years ago in [Rumelhart 1986], but they still being actively employed in the industry and academia, for instance in Nonlinear Control ([Wang et al. 2021],[Kaaniche et al. 2021]), Sensor Validation ([Alves et al. 2021]), Renewable Energy ([Barreto et al. 2021]), among others applications.

Besides all those recent advances, the vanilla RBFNN still is employed in several industrial and academic problems, being this one of the most classic and used models of Machine Learning. There are still several enhancement opportunities for RBFNN, including the optimization of the number of neurons of the hidden layer (the number of centers), the chosing of optimal points for the centers and, mainly, the radius of influence of each center, also known as widths. Many times these parameters are chosen empirically.

This last parameter - the influence radius or width - determine the amount of overlaps between radial functions, as well as the generalizability of the network. The objective of this work is to present the new method Maximum and Minimum Distances and Number of Centers (MMDNC) for calculating the radius of the function radial, applying it to supervised learning problems, more specifically of classification, whose goal is to predict qualitative target variables from a set of independent variables, previously observed. The results obtained by the proposed method were also compared with some traditional methods, as the proposed by [Haykin 1994], described in Equation 2, where σ is the fixed value of the radii, d_{max} is the maximum distance between the centers and m is the number of centers.

The remaining of the paper is organized as follows. In Section 2 is presented the background concepts abount RBFs and RBFNNs, and also reviewed recent advances in the field. In Section 3 a new heuristic method for definition of RBF widths is proposed, comparing with previous methods. In Section 4 computational experiments are employed in order to assess the reliability of the proposed heuristic. Finally, in Section 5, the conclusions are presented, as the remarks of future works.

2. Background

According to [Buhmann 2003], Radial Basis Functions (RBF) are functions $\varphi : \mathbb{R}^p \to \mathbb{R}$ which depends only on the distance $\|x - \mu\|$ from a fixed point μ (the center or origin point) to another given point x, both $\mu, x \in \mathbb{R}^p$ where p is the dimensionality of the input data, and the radius (or width) σ which determines the influence maximum extent of the distance $\|x - \mu\|$. Hereafter, the distance calculation and the parameters μ_j and σ of each RBF φ_j will be implicit in the notation, such that $\varphi_j(x) \equiv \varphi_j(\|x - \mu_j\|, \sigma)$. In the Table 1, some common RBF functions employed in RBFNNs models are listed and the relationships between μ and σ becomes more clear in each case.

Name	$\varphi(x)$
Logistic	$1 + \exp\left(\frac{\|x-\mu_j\|}{\sigma^2}\right)$
Gaussian	$\exp\left(\frac{-\|x-\mu_j\|^2}{2\sigma^2}\right)$
Multiquadratic	$\sqrt{1 + (\sigma\|x - \mu_j\|)^2}$
Inverse Multiquadratic	$\frac{1}{\sqrt{(\sigma\|x-\mu_j\|)^2}}$

Table 1. Most common Radial Basis Functions

RBF Neural Networks (RBFNN) were first proposed in [Rumelhart 1986] and are neural networks with a three-layered architecture (input, hidden and output), as shown in Figure 1, where the RBFs are employed as the activation functions in the hidden layer. According to [Haykin 1994], RBF networks has three types of parameters: the centers $\mu_j \in \mathbb{R}^d$, the variances (or widths) $\sigma_j \in \mathbb{R}^+$ for each center μ_j and weights $w_j \in \mathbb{R}$ of the output layer, for $j = 0..K$ and $K \in \mathbb{N}^+$ the number of centers.

Then, given a training dataset $\mathcal{D} = \{x_i, y_i\}$, where $x_i \in \mathbb{R}^p$ are the p independent attributes, $y_i \in \mathbb{R}^q$ are the q dependent attributes (or labels), for $i = 1..n$ and n the number of instances in $\mathcal{D}$, the RBF network input layer is represented by the q inputs in x_i, the hidden layer contains K neurons with RBF activation functions parametrized by μ_j and σ_j, for $j = 1..K$, and the output layer contains q neurons which are linear combinations of the hidden layer outputs weighted by w_j, such as each output $\hat{y}_z$ is defined in Equation 1, for $z = 1..q$.

$$\hat{y}_z = w_{z,0} + \sum_{j=1}^{K} w_{z,j}\varphi_j(x_i) \tag{1}$$

Is worth noting that each RBF function has two main parameters: the center μ_j, which is different in each φ_j and the radius σ, which may be different for each φ_j or unique and shared among all of them. The radius σ governs the width of influence of each RBF and have direct impact in the generalization and specialization capabilities of the RBFNN. Together with the chosing of K and the centers μ_j, the chosing of optimal values of σ is determinant to avoid the underfit or overfit of RBFNN models.

It is also notable that the hidden layer acts as a feature extraction layer, creating an embedding space to be later used by the output layer. Also, the radial functions are used to transform non-linearly separable problems into linearly separable ones, whose patterns are recognizable by each linear output neuron.

There exist many approaches for training RBFNNs, separated mainly by how the hidden and output layers are trained. The most common approach is train these layers separately in the two phase method listed below:

1. *Hidden Layer Training*
   In this phase the appropriate number of centers K is chosen, then the centers μ_i and the widths σ_i are determined. Some methods employed include random sampling,
2. *Output Layer Training*:
   In this phase the weights of the output layer are trained using the outputs of the hidden layer for

In the literature there are also methods that train both layers together, as using Backpropagation algorithm or Genetic Algorithms, for instance. However, the two-phase approach has some key advantages as granting the semantic explainability of the centers μ_j and radius σ, which may exploit expert knowledge e the employment of computationally cheap methods (as the OLS) in the training of the output layer.

Recent RBFNN research also proposed new improved architectures, as new multilayered architectures [Bodyanskiy et al. 2020], self-adaptative and multi-hierarchical ar-

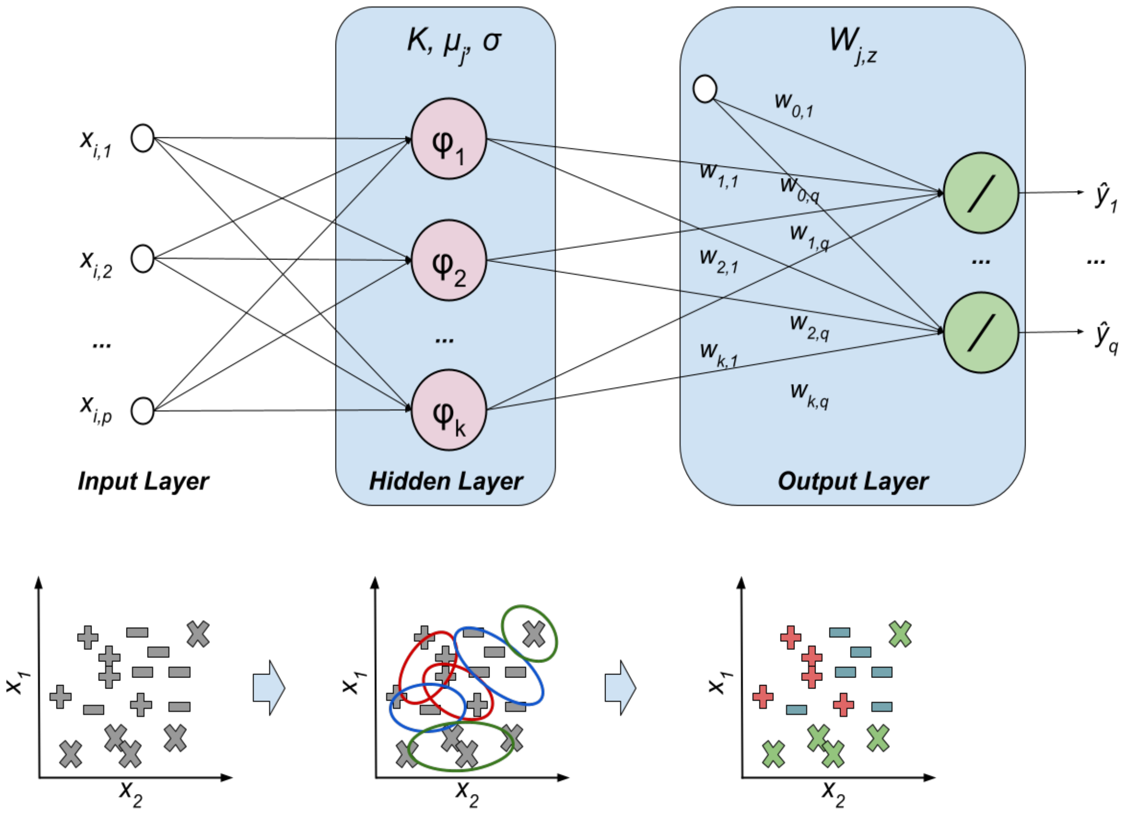

Figure 1. RBFNN Architecture and its nonlinear capabilities

chitectures [Zhao-zhao et al. 2020] and multicolumn networks [Han et al. 2021]. Ensembles of RBFNNs were subject of investigation in [Wu et al. 2021]. Deep Learning extensions for RBFNN where proposed in [Zadeh et al. 2018] and [Burruss et al. 2021].

2.1. RBF Radius Determination Methodologies

The initialization of centers and radius in RBFNNs still the target of investigation, for instance [Dong et al. 2011], and many improvements were proposed in recent years.

The classic approach for estimating the ideal radius σ were proposed in [Haykin 1994] and presented in Equation 2, where d_{max} represents the maximum distance between the centers.

$$\sigma = \frac{d_{max}}{\sqrt{2K}} \tag{2}$$

In [Benoudjit et al. 2002] a simple heuristic is proposed based in the standard deviation of the clusters represented by each center multiplied by an scalar s determined by the least validation error, according to Equation 3.

$$\sigma'_j = q\sigma_j \tag{3}$$

An iterative approach is proposed in [Verleysen and Hlavackova 1994], where the width σ is adapted during the learning of the centers μ, according to Equation 4 where β is the learning rate.

$$\sigma_{t+1} = (1 - \beta_t)\sigma_t + \beta_t 2||x_i - \mu_t|| \tag{4}$$

Most recently, [BATISTA et al. 2019] proposed the MMRBF (Minimum Mean RBF) heuristic based in the average of the minimum distance between the centers and the averages of the distances, according to Equation 5, where $d(j, i) = \|\mu_j - \mu_i\|$ represents the average distances between the center μ_j and all the other centers μ_i, for $i \neq j$.

$$\sigma_j = \frac{1}{2}\left[\left(\frac{1}{k}\sum_{i=1}^{k} d(j,i)\right) + \min(d(j,i))\right] \tag{5}$$

In the next section a new method for radius estimation is proposed, as well as the complete details of the RBFNN method.

3. Proposed Methodology

In this section we present the method Maximum and Minimum Distances and Number of Centers (MMDNC), a RBFNN method that employs a two phase training approach, with gaussian RBFs in the hidden layer. The hidden layer training procedure starts with the definition of K hyperparameter, which is strongly dependent of the application instance and empirically defined by the user.

The centers μ_j are defined using the K-means algorithm. K-means is a unsupervised clustering technique based in the Expectation-Maximization optimization algorithm, where in its initial phase randomly selects K input vectors to be the μ_j centers. Then an iterative process assigns the μ_j center for each instance $(x_i, y_i) \in \mathcal{D}$ which minimize the Euclidean Distance $\|(x_i, y_i) - \mu_j\|x$ and later update each μ_j to be the average of all instances assigned to it. This process is repeated until there is convergence of the algorithm, according to [Braga et al. 2000]. This process assign values for the K centers μ_j, remaining in this phase the determination of the radius σ_j for all centers.

The proposed radius (variance) calculation is described in Equation 6 where d_j^{max} and d_j^{min} are respectively the maximum and the minimum distance between the center μ_j and all the other centers, such that $d_j^{max} = \max \|\mu_j - \mu_k\|$, and $d_j^{min} = \min \|\mu_j - \mu_k\|$, $k = 1..K, k \neq j$. The intuition of the method is to find the average distance between the centers and increase these mean values by the square value of K, which means the increment of uncertainty due the number of centers. This intuition helps also in the model's interpretability, once is clear that the radius is a function of average distance, unlike the methods reviewed in Section 2.1.

$$\sigma_j = \frac{d_j^{max} - d_j^{min}}{2} + \sqrt{K} \tag{6}$$

The output layer training employs the Ordinary Least Squares algorithm to calculate the vector of weights using the training data $\mathcal{D}$, through the multiplication between the inverse matrix of the results of the hidden layer H and the matrix of the true output values Y.

Given the matrix $X \in \mathbb{R}^{nxp}$ of independent attributes, the column vectors $Y_z \in \mathbb{R}^{nx1}$ for each one of the q dependent attributes, the matrix $H \in \mathbb{R}^{nxK}$ contains the outputs of the hidden layer such as $h_{i,j} = \varphi_j(x_i)$. For each output y_z, $z = 1..q$, the weights vector $W_z \in \mathbb{R}^{Kx1}$ is calculated by multiplying Y_z by the inverse of H, as defined in Equation 7.

$$W_z = H^{-1} \cdot Y_z \quad \forall z = 1..q \tag{7}$$

The presented approach introduces a new simple method for estimating the radius σ that improves the generalization capability of the vanilla RBFNNs, reducing the overfit and keeps the method simple and explainable. In the next section the proposed method is assessed and compared with traditional methodologies of radius estimation.

4. Computational Experiments

The aim of this section is to assess the effectiveness of the MMDNC method, measuring its performance against 14 open benchmark datasets, and comparing the results with two other classic methods. In order to grant the transparency and reproducibility of this research the source code of the proposed method and experiments can be accessed at the address `https://bit.ly/source_rbf_cbic_2021`[1]. The methods were developed using the Python 3 language. For commodity, all datasets files employed in these research are also stored and available at the address `https://github.com/LucasPepper/rbf_sbic`.

The chosen competing methods are the traditional RBFNN defined in [Haykin 1994], which employs the radius estimation defined by Equation 2, hereafter named only by RBF, and the method proposed by [BATISTA et al. 2019] which employs the radius estimation defined by Equation 5, hereafter named by MMRBF.

The benchmark datasets were collected from the UCI Machine Learning Repository[2], published in [Dua and Graff 2017], with the exception of the Appendicitis dataset which was taken from the KEEL Dataset Repository[3], published in [Hou 2018]. Both repositories are open and well known in the literature.

It is noticed that both neural networks achieved similar accuracies with respect to bases 3 and 4, with percentage differences less than 3 points between them. For cases 1 and 2, the network proposal achieved considerably better results than the traditional one. It is reinforced that they were used the same parameters for both networks, changing only the method of calculation of rays.

These results can be explained through the influence of the radius on the superposition of radial functions of different classes, which can lead to ambiguity or erroneous results. This fact reinforces the importance of careful selection of the radius in RBF networks, as well as other hyperparameters, such as the number of clusters and method of definition of centers.

[1] The source codes are hosted in Google Colab online framework

[2] `https://archive.ics.uci.edu/ml/index.php`

[3] `https://data.mendeley.com/datasets/py4hhv3rb8/1`

Dataset	**RBF**	**MMRBF**	**MMDNC**
Append	0.8600	0.89	**0.9**
Austr	0.6753	0.6812	**0.6957**
Bank	0.9956	0.9971	**0.9985**
Blood	0.7392	**0.7432**	0.7405
Bupa	**0.7471**	0.7441	0.7412
Cancer	0.9750	0.9735	**0.9824**
Diabetes	0.7211	0.7276	**0.7316**
Fertil	**0.9000**	**0.9000**	**0.9000**
Haberm	0.7633	**0.7800**	0.7733
hHeart	**0.7704**	0.7630	0.7593
ilpd	**0.7018**	**0.7018**	**0.7018**
Ionosp	0.8229	0.8943	**0.9114**
Parkins	**0.8579**	0.8474	**0.8579**
sonar	0.6600	**0.8800**	0.8750

Table 2. Average accuracy by dataset and method

5. Conclusion

This work introduced the Maximum and Minimum Distances and Number of Centers (MMDNC) method for RBFNN networks and evaluated its performance for 14 classification problems, being applied in databases known in the literature. For the training of networks, the k-means clustering technique and the application of the radial Gaussian function, with the subsequent calculation of the vector of weights, to determine the resulting classes in the output layer. Different methods of calculating the radii of radial functions were also compared, which proved to be relevant for the achieving high accuracy, which is intrinsically related to the choice of architectures and appropriate hyperparameters.

In relation to future works, it is suggested to evaluate the proposed method for regression problems and its robustness for datasets with imbalanced classes. Deep RBFNN models and its hyperparameter optimization are also considered for future research.

References

Alves, M. A. D., Pinto, J. O. P., Galotto, L., Kimpara, M. L. M., Garcia, R. C., Godoy, R. B., Teixeira, H. C. G., and Campos, M. C. M. (2021). A modified algorithm for training and optimize rbf neural networks applied to sensor measurements validation. *IEEE Sensors Journal*.

Barreto, G. A., Brasil, I. S., and Souza, L. G. M. (2021). Revisiting the modeling of wind turbine power curves using neural networks and fuzzy models: an application-oriented evaluation. *Energy Systems*, pages 1–28.

BATISTA, P. V., BRAGA, A. D. P., and TORRES, L. C. B. (2019). Cálculo do raio em redes rbf baseado na distância média entre os centros. In *Congresso Brasileiro de Automática-CBA*.

Benoudjit, N., Archambeau, C., Lendasse, A., Lee, J. A., Verleysen, M., et al. (2002). Width optimization of the gaussian kernels in radial basis function networks. In *ESANN*, volume 2, pages 425–432.

Bodyanskiy, Y., Pirus, A., and Deineko, A. (2020). Multilayer radial-basis function network and its learning. In *2020 IEEE 15th International Conference on Computer Sciences and Information Technologies (CSIT)*, volume 1, pages 92–95. IEEE.

Braga, A. d. P., Carvalho, A., and Ludermir, T. B. (2000). *Redes neurais artificiais: teoria e aplicações*. Livros Técnicos e Científicos.

Buhmann, M. D. (2003). *Radial basis functions: theory and implementations*, volume 12. Cambridge university press.

Burruss, M., Ramakrishna, S., and Dubey, A. (2021). Deep-rbf networks for anomaly detection in automotive cyber-physical systems. *arXiv preprint arXiv:2103.14172*.

Dong, C.-R., Chan, P. P. K., Ng, W. W. Y., and Yeung, D. S. (2011). A survey of the initialization of centers and widths in radial basis function network for classification. In *2011 International Conference on Machine Learning and Cybernetics*, volume 3, pages 1082–1087.

Dua, D. and Graff, C. (2017). UCI machine learning repository.

Han, Z., Qian, X., Huang, H., and Huang, T. (2021). Efficient design of multicolumn rbf networks. *Neurocomputing*, 450:253–263.

Haykin, S. (1994). Neural networks: A comprehensive foundation: Macmillan college publishing company. *New York*.

Hou, Y. (2018). KEEL dataset repository. ZSCC: NoCitationData[s0] Type: dataset.

Kaaniche, K., Rashid, N., Miraoui, I., Mekki, H., and El-Hamrawy, O. I. (2021). Mobile robot control based on 2d visual servoing: A new approach combining neural network with variable structure and flatness theory. *IEEE Access*, 9:83688–83694.

Rumelhart, D. E. (1986). Learning internal represectations by error propagation. *Parallel distributed processing*, 1(8).

Verleysen, M. and Hlavackova, K. (1994). An optimized rbf network for approximation of functions. In *ESANN'1994 proceedings - European Symposium on Artificial Neural Networks*.

Wang, C., Cui, L., Liang, M., Li, J., and Wang, Y. (2021). Adaptive neural network control for a class of fractional-order nonstrict-feedback nonlinear systems with full-state constraints and input saturation. *IEEE Transactions on Neural Networks and Learning Systems*, pages 1–13.

Wu, H., Zhao, Y.-P., Yang, T.-L., and Tan, H.-J. (2021). An ensemble radius basis function network based on dynamic time warping for real-time monitoring of supersonic inlet flow patterns. *Aerospace Science and Technology*, 111:106551.

Zadeh, P. H., Hosseini, R., and Sra, S. (2018). Deep-rbf networks revisited: Robust classification with rejection. *arXiv preprint arXiv:1812.03190*.

Zhao-zhao, Z., Qiu-wan, W., and Ying-qin, Z. (2020). A self-adaptive multi-hierarchical modular neural network for complex problems. In *International Conference on Verification and Evaluation of Computer and Communication Systems*, pages 244–256. Springer.