A two phase evolutionary method to train RBF networks

Ioannis G. Tsoulos, Alexandros Tzallas, Evangelos Karvounis

Department of Informatics and Telecommunications, University of Ioannina, Greece

Abstract

This article proposes a two phase hybrid method to train RBF neural networks for classification and regression problems. During the first phase a range for the critical parameters of the RBF network is estimated and in the second phase a genetic algorithm is incorporated to locate the best RBF neural network for the underlying problem. The method is compared against the traditional training method of RBF neural networks on a wide series of classification and regression problems from the relevant literature and the results are reported.

1 Introduction

In machine learning appear many practical problems such as classification and regression problems. A good programming tool that can be used to tackle this problem is Radial Basis Function (RBF) networks[1]. These networks typically are expressed as a function:

$$y(x) = \sum_{i=1}^{k} w_i \phi \left(\| x - c_i \| \right) \tag{1}$$

where x is the input pattern, the vector $\overrightarrow{w}$ is called the weight vector and $y(x)$ is the predicted value of the network. RBF networks are feedforward neural networks[2] with three computational layers:

1. The input layer, where the problem is presented in the form of patterns to the neural network
2. The processing layer, where a computation is performed using the Gaussian processing units $\phi(x)$. These units can have many forms in the relevant literature but the most used form is the Gaussian function expressed as:
$$\phi(x) = \exp\left(-\frac{(x-c)^2}{\sigma^2} \right) \tag{2}$$

The value $\phi(x)$ depends only on the distance of vector x from some other vector c, which typically is called centroid.

3. The output layer where the output of every function $\phi(x)$ is multiplied by a corresponding weight value w_i.

RBF networks have been used in many classification and regression problems from the areas of physics [3, 4, 5, 6], medicine [7, 8, 9], solution of differential equations [10, 11], chemistry [12, 13, 14], economics [15, 16, 17], digital communications [18, 19] etc. Because of the extensive use of RBF networks, many methods have been proposed in the recent literature to enhance them. There are methods that parallelize the RBF networks [20, 21], methods that improve the initialization of the RBF parameters [22, 23, 24], methods that alter the architecture of the network [25, 26, 27], methods aimed to locate the best set of the RBF parameters with global optimization techniques[28, 29, 30] etc. This article transforms the problem of RBF training into an optimization problem and applies a modified genetic algorithm technique to solve it. The global optimization problem is defined as :

$$\min\left(E\left(y\right)\right) = \sum_{i=}^{m} \left(y\left(x_i\right) - t_i\right)^2 \tag{3}$$

where m is the total number of input patterns and t_i is the output for pattern x_i. The suggested approach has two phases: firstly reasonable bounds for the RBF parameters are estimated using the Kmeans [32] algorithm and in the second phase the modified algorithm is used to solve the problem of equation 3 inside the bounds located in the first phase.

The rest of this paper is organized as follows: in section 2 the proposed method is described, in section 3 the conducted experiments are listed and the proposed method is compared against the traditional training of RBF networks and finally in section 4 some conclusions are derived.

2 Method description

The proposed method can be divided into two main phases: during the first phase an approximation for the bound of RBF parameters is made using the K-Means algorithm in the second phase the optimization problem is solved using a modified genetic algorithm. These phases are outlined in detail in the following subsections.

2.1 Bound location phase

The proposed genetic algorithm has chromosomes with dimension $(d+1) \times k$, where d is the dimension of the input problem, i.e. the dimension of the vector $\overrightarrow{x_i}$ in equation 3 and k is the total number of processing units of the RBF network. The layout of each chromosome is presented in Figure 1. Every center $\overrightarrow{c_i}$ in the

Figure 1: The layout of the chromosomes in the proposed genetic algorithm.

c_{11}	c_{12}	...	c_{1d}	σ_1	c_{21}	c_{22}	...	c_{2d}	σ_2	...	c_{k1}	c_{k2}	...	c_{kd}	σ_k

equation 1 is a vector of dimension d and also an additional parameter is reserved for the parameter σ of every $\phi(x)$ function. The centroids and the corresponding variances are estimated using the Kmeans algorithm that described in algorithm 1. The value σ_i for every $\phi_i(x)$ is calculated as:

$$\sigma_i = \sum_{j=1}^{d} s_{ij}^2 \tag{4}$$

After the estimation of c_i and σ_i the vectors $\overrightarrow{L}$, $\overrightarrow{R}$ with dimension $(d+1)\times k$ are constructed. These vectors will serve as the bounds for the chromosomes of the genetic population. These vector are constructed using the following procedure:

1. **Set** m=0
2. **Set** $F > 1$
3. **For** $i = 1..k$ **do**
   (a) **For** $j = 1..d$ **do**
      i. **Set** L_m=$-F \times c_{ij}$, R_m=$F \times c_{ij}$
      ii. **Set** $m = m + 1$
   (b) **EndFor**
   (c) **Set** $L_m = -F \times \sigma_i$, $R_m = F \times \sigma_i$
   (d) **Set** $m = m + 1$
4. **EndFor**

2.2 Main algorithm

The basic steps of the main algorithm are given below:

1. **Initialization Step**
   (a) **Read** the train set with m of d dimension.
   (b) **Set** k the number of nodes for the RBF network.
   (c) **Estimate** the vectors $\overrightarrow{L}$, $\overrightarrow{R}$ using the procedure of subsection 2.1.
   (d) **Initialize** a genetic population of N_C random chromosomes inside $[L, R]$.
   (e) **Set** the selection rate $P_s \in [0, 1]$, the mutation rate $P_M \in [0, 1]$, $iter = 0$, and i_{max} the maximum number of generations.

Algorithm 1 The KMeans algorithm.

1. **Repeat**
   (a) $S_j = \{\},\ j = 1..k$
   (b) **For** every sample x_i **Do**
      i. **Set** $j^* = \min_{i=1}^{k} \{D(x_i, c_j)\}$, where j^* is the nearest center for sample x_i.
      ii. **Set** $S_{j^*} = S_{j^*} \cup \{x_i\}$.
   (c) **EndFor**
   (d) **For** every center c_j **Do**
      i. **Set** M_j=number of elements in S_j
      ii. **Update** c_j

$$c_j = \frac{1}{M_j} \sum_{i=1}^{M_j} x_i$$

   (e) **EndFor**
2. **Calculate** the corresponding variances

$$s_j^2 = \frac{\sum_{i=1}^{M_j} (x_i - c_j)^2}{M_j}$$

3. **Terminate** when c_j no longer change.

2. **Evaluation Step**

   For every chromosome g **calculate** using the procedure defined in subsection 2.3 the fitness f_g.

3. **Genetic step**

   (a) **Select** $P_s \times N_c$ parents from the population using tournament selection, i.e. create subgroups of $T > 2$ chromosomes and select the one with the best fitness value as the parent.

   (b) **Crossover**: For every pair (x, y) of selected parents create two new offsprings $\tilde{x}$ and $\tilde{y}$:

   $$\begin{aligned} \tilde{x_i} &= a_i x_i + (1 - a_i)\, y_i \\ \tilde{y_i} &= a_i y_i + (1 - a_i)\, x_i \end{aligned} \tag{5}$$

   with a_i a random number and $a_i \in [-0.5, 1.5]$ [33].

   (c) **Mutation**: For every element of each chromosome create a random number $r \in [0, 1]$. If $r \leq P_m$ then change randomly this element.

   (d) **Replace** the $P_s \times N_c$ worst chromosomes in the population with the generated offsprings.

4. **Termination Check Step**

   (a) **Set** $iter = iter + 1$

   (b) **Terminate** if the termination criteria of subsection 2.4 are satisfied, **else Goto** Evaluation Step.

2.3 Fitness evaluation

In this step a valid RBF network $y(x) = \sum_{i=1}^{k} w_i \phi\left(\|x - c_i\|\right)$, is created using the chromosome g and subsequently is trained using the typical training procedure for RBF networks. The main steps to calculate the fitness f_g of a chromosome g are the following:

1. **Decode** the chromosome g to the parts (centers and variances) of the RBF network as defined by the layout of Figure 1.

2. **Calculate** the output vectors $w_1, w_w, \ldots, w_k$ by solving the an induced system of equations:

   (a) **Set** $W = w_{kj}$ the matrix of k weights, $\Phi = \phi_j\left(x_i\right)$ and $T = \{t_i\}$.

   (b) **Solve:**

   $$\Phi^T \left(T - \Phi W^T\right) = 0 \tag{6}$$

   givivg**:**

   $$W^T = \left(\Phi^T \Phi\right)^{-1} \Phi^T T = \Phi^{\dagger} T \tag{7}$$

   The matrix $\Phi^{\dagger} = \left(\Phi^T \Phi\right)^{-1} \Phi^T$ is the pseudo-inverse of Φ, with

   $$\Phi^{\dagger} \Phi = I \tag{8}$$

3. **Set** $f_g = \sum_{i=}^{m} \left(y\left(x_i\right) - t_i\right)^2$

2.4 Stopping rule

Define as g_{best} the best chromosome in the population and define as $\sigma^{(\text{iter})}$ the variance of best fitness $f\left(g_{\text{best}}\right)$ at generation iter. If fitness $f\left(g_{\text{best}}\right)$ has not improved for a number of generations, then probably the algorithm should terminate. Hence, the termination rule is defined as:

$$\text{iter} \geq i_{\max} \text{ OR } \sigma^{(\text{iter})} \leq \frac{\sigma^{(\text{klast})}}{2} \tag{9}$$

where klast is the last generation where a new minimum was found.

3 Experiments

In order to evaluate the performance of the proposed method, comparative experiments were performed on a series of well - known classification and regression datasets from the relevant literature.

3.1 Experimental setup

The RBF network was coded in ANSI C++, using the Armadillo library [34] and the optimization was performed using the Genetic optimization method of the optimization package OPTIMUS, that is freely available from `https://github.com/itsoulos/OPTIMUS/`. Also, to have more reliability in the results the common used method of 10 - fold cross validation was used, which means that the the original data was randomly partitioned into 10 equal sized subsamples. Subsequently, 10 independent experiments were conducted: in each experiments one subsample is used as the testing data and all the others as the training data. The average error on the test data is the total test error. All the experiments were executed 30 times with different initialization for the random generator each time. The random generator used was the function drand48() of C programming language. The execution environment was an Intel Xeon E5-2630 multi core machine using the OpenMP library [36] for parallelization and the Ubuntu Linux operating system. The parameters for the genetic algorithm are displayed in Table 1.

3.2 Experimental datasets

The classification problems used for the experiments were found in most cased in two internet databases:

1. UCI dataset repository, `https://archive.ics.uci.edu/ml/index.php`
2. Keel repository, `https://sci2s.ugr.es/keel/datasets.php`[35].

Table 1: Experimental parameters.

PARAMETER	VALUE
k	10
N_c	200
P_s	0.10
P_m	0.05
F	3.0
i_{max}	200

The following classification datasets were used:

1. **Alcohol** dataset, a dataset about Alcohol consumption [37].
2. **Appendictis** dataset, proposed in [38].
3. **Australian** dataset, the dataset is related to credit card applications.
4. **Balance** dataset, which models psychological experimental results.
5. **Cleveland** dataset. A dataset obtained from the V.A. Medical Center, Long Beach and Cleveland Clinic Foundation aimed to detect the presence of heart disease in patients.
6. **Dermatology** dataset, which is used for differential diagnosis of erythemato-squamous diseases.
7. **Glass** dataset. The dataset contains glass component analysis for glass pieces that belong to 6 classes.
8. **Hayes roth** dataset. This dataset[39] contains **5** numeric-valued attributes and 132 patterns.
9. **Heart** dataset, which is used to identify the absence or presence of heart disease.
10. **HouseVotes** dataset. This data set includes votes for each of the U.S. House of Representatives Congressmen on the 16 key votes.
11. **Ionosphere** dataset. The ionosphere dataset contains data from the Johns Hopkins Ionosphere database.
12. **Liverdisorder**: This dataset contains blood analysis data from people with liver disorders.
13. **Mammographic** dataset. This dataset be used to identify the severity (benign or malignant) of a mammographic mass lesion from BI-RADS attributes and the patient's age. It contains 830 patterns of 5 features each.

14. **Parkinsons** dataset. This dataset is composed of a range of biomedical voice measurements from 31 people, 23 with Parkinson's disease (PD)[40].

15. **Pima** dataset. The Pima Indians Diabetes dataset contains 768 examples of 8 attributes each that are classified into two categories: healthy and diabetic.

16. **Popfailures** dataset. This dataset contains records of simulation crashes encountered during climate model uncertainty quantification (UQ) ensembles.

17. **Regions2** dataset. It is created from liver biopsy images of patients with hepatitis C [41]. From each region in the acquired images, 18 shape-based and color-based features were extracted, while it was also annotated form medical experts. The resulting dataset includes 600 samples belonging into 6 classes.

18. **Ring** dataset. It is an 20 dimensional problem with two classes. Each class is drawn from a multivariate normal distribution.

19. **Saheart** dataset. The dataset is about to categorize persons if have a coronary heart disease. The dataset contains 462 patterns with 9 features each.

20. **Segment** dataset. This database contains patterns from a database of 7 outdoor images (classes). The dataset contains 2310 patterns with 19 features each.

21. **Sonar** dataset. This data set contains signals obtained from a variety of different aspect angles, spanning 90 degrees for mines and 180 degrees for rocks. Each pattern is a set of 60 numbers in the range 0.0 to 1.0, where each number represents the energy within a particular frequency band, integrated over a certain period of time. The output attribute contains the letter R if the object is a rock and M if it is a mine (metal cylinder).

22. **Spiral** dataset: The spiral artificial dataset contains 1000 two-dimensional examples that belong to two classes (500 examples each). The number of the features is 2. The data in the first class are created using the following formula: $x_1 = 0.5t\cos(0.08t)\,,\ x_2 = 0.5t\cos\left(0.08t + \frac{\pi}{2}\right)$ and the second class data using**:** $x_1 = 0.5t\cos(0.08t + \pi)\,,\ x_2 = 0.5t\cos\left(0.08t + \frac{3\pi}{2}\right)$

23. **Tae** dataset. The data consist of evaluations of teaching performance over three regular semesters and two summer semesters of 151 teaching assistant (TA) assignments at the Statistics Department of the University of Wisconsin-Madison.

24. **Thyroid** dataset, which concerns Thyroid disease records[47].

25. **Wdbc** dataset. The Wisconsin diagnostic breast cancer dataset contains data for breast tumors.

26. **Wine** dataset. The wine recognition dataset contains data from wine chemical analysis.

27. **Eeg** datasets. As an real word example, consider an EEG dataset described in [42] is used here. The dataset consists of five sets (denoted as Z, O, N, F and S) each containing 100 single-channel EEG segments each having 23.6 sec duration. With different combinations of these sets the produced datasets are Z_F_S, ZO_NF_S, ZONF_S.

28. **Zoo** dataset. A simple database where the task is to classify animals in seven predefined classes and most of the attributes are boolean-valued.

The regression datasets are in most cases available from the Statlib URL `ftp://lib.stat.cmu.edu/datasets/index.html`:

1. **Abalone** dataset. This data set can be used to obtain a model to predict the age of abalone from physical measurements.

2. **Airfoil** dataset, which is used by the NASA for a series of aerodynamic and acoustic tests [43].

3. **Anacalt** dataset. This contains information about the decisions taken by a supreme court.

4. **BK** dataset. This dataset comes from Smoothing Methods in Statistics [49] and is used to estimate the points scored per minute in a basketball game.

5. **BL** dataset: This dataset can be downloaded from StatLib. It contains data from an experiment on the affects of machine adjustments on the time to count bolts.

6. **Concrete** dataset. This dataset is taken from civil engineering[44].

7. **Housing** dataset. This dataset was taken from the StatLib library which is maintained at Carnegie Mellon University and it is described in [45].

8. **Laser** dataset. Dataset used in laser experiments.

9. **MB** dataset. This dataset is available from Smoothing Methods in Statistics [46] and it includes 61 patterns.

10. **NT** dataset. This dataset contains data from [48] that examined whether the true mean body temperature is 98.6 F.

11. **Quake** dataset. The objective here is to approximate the strength of a earthquake.

3.3 Experimental results

The results for the classification datasets are listed in Table 2 and for the regression datasets the results are reported in Table 3. For the first case the average classification error is reported and for the case of regression datasets the total test error is reported. The column KRBF denotes the classic RBF training method and the column PROPOSED denotes the proposed method. The KBF simply consists of two phases: in the first phase centers and variances are estimated through the Kmeans algorithm and in the second phase a system of equations is solved to obtain the weights w_i of the RBF network.

From the experimental results it is clear that the proposed method is significantly superior to traditional technique in almost all datasets. In the proposed method the appropriate initialization interval is found for the parameters of RBF using Kmeans. A parallel genetic algorithm was then applied to this previous value range creating a variety of neural networks. This combination of techniques obviously has very good results as it combines a very efficient clustering method and an excellent optimization method that is ideally parallelized. Of course, the new method requires much more execution time, due to the presence of the genetic algorithm but the parallel execution of the software drastically reduces this time.

4 Conclusions

A two phase method was proposed in this article to train RBF neural networks for classification and regression problems. Firstly, a common used clustering method was used to estimate an interval for the critical parameters of the neural network. Subsequently, a parallel genetic algorithm was incorporated to locate the best RBF network with good generalization capabilities. The used software was coded using ANSI C++ and open source libraries such as the Armadillo library and the OpenMP library for parallelization. Future research may include

1. Use parallel methods for the Kmeans clustering phase of the method.
2. Dynamic selection of K in Kmeans algorithm.
3. More advanced stopping rules for the genetic algorithm.
4. Replace the Genetic algorithm with other optimization methods such as Particle Swarm Optimization, Ant Colony Optimization etc.

Acknowledgments

The experiments of this research work was performed at the high performance computing system established at Knowledge and Intelligent Computing Laboratory, Dept of Informatics and Telecommunications, University of Ioannina, acquired with the project "Educational Laboratory equipment of TEI of Epirus" with MIS 5007094 funded by the Operational Programme "Epirus" 2014-2020, by ERDF and national finds.

Table 2: Experimental results for classification problems.

DATASET	KRBF	PROPOSED
Alcohol	46.63%	21.86%
Appendicitis	12.23%	16.03%
Australian	34.89%	22.97%
Balance	33.42%	12.88%
Cleveland	67.10%	51.75%
Dermatology	62.34%	37.37%
Glass	50.16%	49.16%
Hayes Roth	64.36%	35.26%
Heart	31.20%	17.80%
HouseVotes	6.13%	3.67%
Ionosphere	16.22%	10.33%
Liverdisorder	30.84%	28.73%
Mammographic	21.38%	17.25%
Parkinsons	17.42%	17.37%
Pima	25.78%	24.00%
Popfailures	7.04%	5.44%
Regions2	38.29%	25.81%
Ring	21.65%	2.09%
Saheart	32.19%	29.38%
Segment	59.68%	39.44%
Sonar	27.85%	19.62%
Spiral	44.87%	18.98%
Tae	60.07%	52.44%
Thyroid	10.52%	7.12%
Wdbc	7.27%	5.29%
Wine	31.41%	8.67%
Z_F_S	13.16%	4.21%
ZO_NF_S	9.02%	4.17%
ZONF_S	4.03%	2.18%
ZOO	21.93%	9.00%

Table 3: Experiments for regression datasets.

DATASET	KMEANS RBF	PROPOSED
ABALONE	2559.48	1960.22
AIRFOIL	5.49	0.58
ANACALT	11.628	0.003
BK	0.17	0.23
BL	0.05	0.0009
CONCRETE	1.15	0.52
HOUSING	2884.09	693.22
LASER	2.35	1.04
MB	11.33	0.63
NT	72.14	0.09
QUAKE	15.36	7.86

Compliance with Ethical Standards

All authors declare that they have no conflict of interest.

References

[1] J. Park and I. W. Sandberg, Universal Approximation Using Radial-Basis-Function Networks, Neural Computation 3, pp. 246-257, 1991.

[2] C.M. Bishop, Neural networks and their applications, Review of Scientific Instruments **65**, pp. 1803-1832,1994.

[3] P. Teng, Machine-learning quantum mechanics: Solving quantum mechanics problems using radial basis function networks, Phys. Rev. E **98**, 033305, 2018.

[4] R. Jovanović, A. Sretenovic, Ensemble of radial basis neural networks with K-means clustering for heating energy consumption prediction, FME Transactions **45**, pp. 51-57, 2017.

[5] A. Alexandridis, E. Chondrodima, E. Efthimiou, G. Papadakis, F. Vallianatos and D. Triantis, Large Earthquake Occurrence Estimation Based on Radial Basis Function Neural Networks,IEEE Transactions on Geoscience and Remote Sensing **52**, pp. 5443-5453, 2014.

[6] V.I. Gorbachenko, M.V. Zhukov, Solving boundary value problems of mathematical physics using radial basis function networks, Comput. Math. and Math. Phys. **57**, pp. 145–155, 2017.

[7] Y.P. Wang, J.W. Dang, Q. Li and S. Li, Multimodal medical image fusion using fuzzy radial basis function neural networks, in 2007 International

Conference on Wavelet Analysis and Pattern Recognition, pp. 778-782, 2017.

[8] S, Mehrabi, M. Maghsoudloo, H. Arabalibeik, R. Noormand and Y. Nozari, Congestive heart failure, Chronic obstructive pulmonary disease, Clinical decision support system, Multilayer perceptron neural network and radial basis function neural network, Expert Systems with Applications **36**, pp. 6956-6959, 2009.

[9] M. Veezhinathan, S. Ramakrishnan, Detection of Obstructive Respiratory Abnormality Using Flow–Volume Spirometry and Radial Basis Function Neural Networks, J Med Syst **31**, pp. 461, 2007.

[10] Nam Mai-Duy, Thanh Tran-Cong, Numerical solution of differential equations using multiquadric radial basis function networks, Neural Networks 14, pp. 185-199, 2001.

[11] N. Mai-Duy, Solving high order ordinary differential equations with radial basis function networks. Int. J. Numer. Meth. Engng. **62**, pp. 824-852, 2005.

[12] Chuanhao Wan and Peter de B. Harrington, Self-Configuring Radial Basis Function Neural Networks for Chemical Pattern Recognition, J. Chem. Inf. Comput. Sci. **39**, 1049–1056, 1999.

[13] Xiaojun Yao, Xiaoyun Zhang, Ruisheng Zhang, Mancang Liu, Zhide Hu, Botao Fan, Prediction of enthalpy of alkanes by the use of radial basis function neural networks, Computers & Chemistry **25**, pp. 475-482, 2001.

[14] A. Shahsavand, A. Ahmadpour, Application of optimal RBF neural networks for optimization and characterization of porous materials, Computers & Chemical Engineering 29, pp. 2134-2143, 2005.

[15] J. A. Momoh and S. S. Reddy, Combined Economic and Emission Dispatch using Radial Basis Function, 2014 IEEE PES General Meeting | Conference & Exposition, National Harbor, MD, 2014, pp. 1-5, doi: 10.1109/PESGM.2014.6939506.

[16] Jau-Jia Guo and P. B. Luh, Selecting input factors for clusters of Gaussian radial basis function networks to improve market clearing price prediction, IEEE Transactions on Power Systems **18**, pp. 665-672, 2003.

[17] L. Falat, Z. Stanikova, M. Durisova, B. Holkova, T. Potkanova, Application of Neural Network Models in Modelling Economic Time Series with Non-constant Volatility, Procedia Economics and Finance **34**, pp. 600-607, 2015.

[18] C. Laoudias, P. Kemppi and C. G. Panayiotou, Localization Using Radial Basis Function Networks and Signal Strength Fingerprints in WLAN, GLOBECOM 2009 - 2009 IEEE Global Telecommunications Conference, Honolulu, HI, 2009, pp. 1-6, 2009.

[19] S. Chen, B. Mulgrew and P. M. Grant, A clustering technique for digital communications channel equalization using radial basis function networks, IEEE Transactions on Neural Networks **4**, pp. 570-590, 1993.

[20] M.G. Arenas, E. Parras-Gutiérrez, V.M. Rivas, P.A. Castillo, M.J. Del Jesus, J.J. Merelo. Parallelizing the Design of Radial Basis Function Neural Networks by Means of Evolutionary Meta-algorithms. In: Cabestany J., Sandoval F., Prieto A., Corchado J.M. (eds) Bio-Inspired Systems: Computational and Ambient Intelligence. IWANN 2009. Lecture Notes in Computer Science, vol 5517. Springer, Berlin, Heidelberg, 2009.

[21] A. Brandstetter and A. Artusi, "Radial Basis Function Networks GPU-Based Implementation," in IEEE Transactions on Neural Networks **19**, pp. 2150-2154, 2008.

[22] L.I. Kuncheva, Initializing of an RBF network by a genetic algorithm, Neurocomputing **14**, pp. 273-288, 1997.

[23] M. Kubat, Decision trees can initialize radial-basis function networks, IEEE Transactions on Neural Networks. **9**, pp. 813-821, 1998.

[24] D. G. B. Franco and M. T. A. Steiner, New Strategies for Initialization and Training of Radial Basis Function Neural Networks, IEEE Latin America Transactions. **15**, pp. 1182-1188, 2017.

[25] E. Ricci, R. Perfetti, Improved pruning strategy for radial basis function networks with dynamic decay adjustment, Neurocomputing **69**, pp. 1728-1732, 2006.

[26] M. Bortman and M. Aladjem, A Growing and Pruning Method for Radial Basis Function Networks, IEEE Transactions on Neural Networks **20**, pp. 1039-1045, 2009.

[27] Guang-Bin Huang, P. Saratchandran and N. Sundararajan, A generalized growing and pruning RBF (GGAP-RBF) neural network for function approximation, IEEE Transactions on Neural Networks **16**, pp. 57-67, 2005.

[28] J. Y. Chen, Z. Qin and J. Jia, A PSO-Based Subtractive Clustering Technique for Designing RBF Neural Networks, 2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence), Hong Kong, pp. 2047-2052, 2008.

[29] A. Esmaeili and N. Mozayani, Adjusting the parameters of radial basis function networks using Particle Swarm Optimization, 2009 IEEE International Conference on Computational Intelligence for Measurement Systems and Applications, Hong Kong, pp. 179-181, 2009.

[30] B. O'Hora, J. Perera and A. Brabazon, Designing Radial Basis Function Networks for Classification Using Differential Evolution, The 2006 IEEE International Joint Conference on Neural Network Proceedings, Vancouver, BC, pp. 2932-2937, 2006.

[31] I.G. Tsoulos, Modifications of real code genetic algorithm for global optimization, Applied Mathematics and Computation **203**, pp. 598-607, 2008.

[32] MacQueen, J.: Some methods for classification and analysis of multivariate observations, in: Proceedings of the fifth Berkeley symposium on mathematical statistics and probability, Vol. 1, No. 14, pp. 281-297, 1967.

[33] P. Kaelo, M.M. Ali, Integrated crossover rules in real coded genetic algorithms, European Journal of Operational Research **176**, pp. 60-76, 2007.

[34] C. Sanderson, R. Curtin, Armadillo: a template-based C++ library for linear algebra, Journal of Open Source Software **1**, pp. 26, 2016.

[35] J. Alcalá-Fdez, A. Fernandez, J. Luengo, J. Derrac, S. García, L. Sánchez, F. Herrera. KEEL Data-Mining Software Tool: Data Set Repository, Integration of Algorithms and Experimental Analysis Framework. Journal of Multiple-Valued Logic and Soft Computing 17, pp. 255-287, 2011.

[36] R. Chandra, L. Dagum, D. Kohr, D. Maydan,J. McDonald and R. Menon, Parallel Programming in OpenMP, Morgan Kaufmann Publishers Inc., 2001.

[37] Tzimourta, Katerina D. and Tsoulos, Ioannis and Bilero, Thanasis and Tzallas, Alexandros T. and Tsipouras, Markos G. and Giannakeas, Nikolaos, Direct Assessment of Alcohol Consumption in Mental State Using Brain Computer Interfaces and Grammatical Evolution, Inventions **3**, pp. 1-12, 2018.

[38] Weiss, Sholom M. and Kulikowski, Casimir A., Computer Systems That Learn: Classification and Prediction Methods from Statistics, Neural Nets, Machine Learning, and Expert Systems, Morgan Kaufmann Publishers Inc, 1991.

[39] B. Hayes-Roth, B., F. Hayes-Roth. Concept learning and the recognition and classification of exemplars. Journal of Verbal Learning and Verbal Behavior **16**, pp. 321-338, 1977.

[40] Little MA, McSharry PE, Hunter EJ, Spielman J, Ramig LO. Suitability of dysphonia measurements for telemonitoring of Parkinson's disease. IEEE Trans Biomed Eng. 2009;56(4):1015. doi:10.1109/TBME.2008.2005954

[41] Giannakeas, N., Tsipouras, M.G., Tzallas, A.T., Kyriakidi, K., Tsianou, Z.E., Manousou, P., Hall, A., Karvounis, E.C., Tsianos, V., Tsianos, E. A clustering based method for collagen proportional area extraction in liver biopsy images (2015) Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS, 2015-November, art. no. 7319047, pp. 3097-3100.

[42] R.G. Andrzejak, K. Lehnertz, F. Mormann, C. Rieke, P. David, and C. E. Elger, Indications of nonlinear deterministic and finite-dimensional structures in time series of brain electrical activity: Dependence on recording region and brain state, Phys. Rev. E **64**, pp. 1-8, 2001.

[43] T.F. Brooks, D.S. Pope, and A.M. Marcolini. Airfoil self-noise and prediction. Technical report, NASA RP-1218, July 1989.

[44] I.Cheng Yeh, Modeling of strength of high performance concrete using artificial neural networks, Cement and Concrete Research. **28**, pp. 1797-1808, 1998.

[45] D. Harrison and D.L. Rubinfeld, Hedonic prices and the demand for clean ai, J. Environ. Economics & Management **5**, pp. 81-102, 1978.

[46] J.S. Simonoff, Smooting Methods in Statistics, Springer - Verlag, 1996.

[47] Quinlan,J.R., Compton,P.J., Horn,K.A., and Lazurus,L. (1986). Inductive knowledge acquisition: A case study. In Proceedings of the Second Australian Conference on Applications of Expert Systems. Sydney, Australia.

[48] Mackowiak, P.A., Wasserman, S.S., Levine, M.M., 1992. A critical appraisal of 98.6 degrees f, the upper limit of the normal body temperature, and other legacies of Carl Reinhold August Wunderlich. J. Amer. Med. Assoc. 268, 1578–1580

[49] J.S. Simonoff, Smooting Methods in Statistics, Springer - Verlag, 1996.