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 other training methods of RBF neural networks on a wide series of classification and regression problems from the relevant literature and the results are reported.

Keywords: RBF networks, classification, regression, genetic algorithms.

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 $\overrightarrow{x}$ 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 $\overrightarrow{x}$ from some other vector $\overrightarrow{c}$, 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. Also, recently the RBF networks have been used in more difficult problems such as authentication assurance of meat products [20], trajectory tracking for electro-hydraulic servo systems, identification of geographical origin for foods [22], prediction of solution gas-oil ratio of crude oil systems [23], prediction of occurrences of haloketones in tap water [24] , health monitoring [25] 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 [26, 27], methods that improve the initialization of the RBF parameters [28, 29, 30], methods that alter the architecture of the network [31, 32, 33], methods aimed to locate the best set of the RBF parameters with global optimization techniques[34, 35, 36] 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 :

$$\min\left(E\left(y\right)\right) = \sum_{i=1}^{m}\left(y\left(x_i\right) - t_i\right)^2 \tag{3}$$

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 k-means [38] 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.

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

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 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 k-means 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 genetic algorithm used here is based on the algorithm denoted as GA (c_{r1}, l) in the paper of Kaelo and Ali[39] with a modified stopping rule as proposed in [37]. The basic steps of the main algorithm are given below:

1. **Initialization Step**
    - (a) **Read** the train set with m patterns of d dimension.

Algorithm 1 The k-means 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.
```

(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.

2. **Evaluation Step**

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

3. **Genetic operations step**
   During this step three genetic operations are performed: selection, crossover and mutation.

   (a) **Selection procedure.** Firstly the chromosomes are sorted with relevance to their corresponding fitness value. The best $(1 - p_s) \times N_c$ are transferred without change to the next generation and the remain ones are substituted by offsprings created through the crossover procedure. In the crossover procedure the mating parent are selected using tournament selection for every parent. The tournament selection has as follows:

   i. Select as set of $T > 2$ chromosomes from the population

   ii. Return the chromosome with the best fitness value in this subset.

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

$$\begin{aligned} \tilde{z_i} &= a_i z_i + (1 - a_i) w_i \\ \tilde{w_i} &= a_i w_i + (1 - a_i) z_i \end{aligned} \tag{5}$$

   with a_i a random number and $a_i \in [-0.5, 1.5]$ [39]. This crossover scheme will be able to better explore the search space of the train error.

   (c) **Mutation procedure** : For every element of each chromosome create a random number $r \in [0,1]$. If $r \leq p_m$ then change randomly this element. The mutation is performed in a way similar to other approaches of genetic algorithms[37] and it is described in subsection 2.5.

   (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:**
   $$\Phi^T \left(T - \Phi W^T\right) = 0 \tag{6}$$
   $$W^T = \left(\Phi^T \Phi\right)^{-1} \Phi^T T = \Phi^{\dagger} T \tag{7}$$
   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.

2.5 Mutation procedure

Let $w = (w_1, w_2, ..., w_n)$ be the chromosome to be mutated. The proposed mutation procedure modifies w_i with probability p_m and the resulting element w_i' is given by

$$w_i' = \begin{cases} w_i + \Delta\left(\text{iter}, R_i - w_i\right), \text{ if} & t > \frac{1}{2}0 \\ w_i - \Delta\left(\text{iter}, w_i - L_i\right), \text{ otherwise} & \end{cases} \tag{10}$$

where t is a random number with $t \in [0, 1]$. The function $\Delta(\text{iter}, y)$ is given by:

$$\Delta(\text{iter}, y) = y\left(1 - r^{\left(1 - \frac{\text{iter}}{\text{ITERMAX}}\right)^b}\right) \tag{11}$$

Table 1: Experimental parameters.

PARAMETER	VALUE
k	10
N_c	200
p_s	0.90
p_m	0.05
F	3.0
i_{max}	200

where r is a random number in $[0,1]$ and b controls the change of element w_i. In the proposed algorithm the value $b=5$ was used.

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 [40] 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 [42] for parallelization and the Ubuntu Linux operating system. The parameters for the genetic algorithm are displayed in Table 1. The parameters of the method were chosen so that there is a balance between speed and efficiency of the method.

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`[41].

The following classification datasets were used:

1. **Alcohol** dataset which is related to alcohol consumption [43].
2. **Appendictis** dataset, proposed in [44].
3. **Australian** dataset [45], which refers to credit card applications.
4. **Balance** dataset [46], which is used to predict psychological states.
5. **Cleveland** dataset, a dataset used to detect heart disease used in various papers[47, 48].
6. **Dermatology** dataset [49], which is used for differential diagnosis of erythemato-squamous diseases.
7. **Glass** dataset. The dataset contains glass component analysis and it has been used in a variety of papers [50, 51].
8. **Hayes roth** dataset [52].
9. **Heart** dataset [53], used to detect heart disease.
10. **HouseVotes** dataset [54], which is about votes in the U.S. House of Representatives Congressmen.
11. **Ionosphere** dataset. The ionosphere dataset contains data from the Johns Hopkins Ionosphere database and it has been studied in a bunch of papers [55, 56].
12. **Liverdisorder** dataset [57], used for detect liver disorders in peoples using blood analysis.
13. **Mammographic** dataset [58], used to identify the severity of a mammographic mass lesion.
14. **Parkinsons** dataset. This dataset is composed of a range of biomedical voice measurements from 31 people, 23 with Parkinson's disease (PD)[59].
15. **Pima** dataset [60], used to detect the presence of diabetes.
16. **Popfailures** dataset [61], that is related to climate model simulation crashes of simulation crashes.
17. **Regions2** dataset, is created from liver biopsy images of patients with hepatitis C [62].
18. **Ring** dataset [63]. It is an 20 dimensional problem with two classes. Each class is drawn from a multivariate normal distribution.
19. **Saheart** dataset [64], used to detect heart disease.

20. **Segment** dataset [65]. This database contains patterns from a database of 7 outdoor images (classes).

21. **Sonar** dataset [66]. The task is discriminate between sonar signals bounced off a metal cylinder.

22. **Spiral** dataset: The spiral artificial dataset contains 1000 two-dimensional examples that belong to two classes.

23. **Tae** dataset [67], which concerns evaluations of teaching performance.

24. **Thyroid** dataset, which concerns Thyroid disease records[79].

25. **Wdbc** dataset [68], which contains data for breast tumors.

26. **Wine** dataset, used to detect through chemical analysis determine the origin of wines and is been used in various research papers [69, 70].

27. **Eeg** datasets, the eeg dataset in [72] is utilized. 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 [71], where the task is classify animals in seven predefined classes.

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

1. **Abalone** dataset [73]. 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 [74].

3. **Anacalt** dataset [75]. This contains information about the decisions taken by a supreme court.

4. **BK** dataset [81], used to estimates the points 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, used to measure the concrete compressive strength [76].

7. **Housing** dataset. This dataset was taken from the StatLib library which is maintained at Carnegie Mellon University and it is described in [77].

8. **Laser** dataset. Dataset used in laser experiments. It has been obtained from the The Santa Fe Time Series Competition Data repository.

9. **MB** dataset. This dataset is available from Smoothing Methods in Statistics [78].
10. **NT** dataset. This dataset contains data from [80] 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. It has been obtained from the Bilkent University Function Approximation Repository.

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, GRBF denotes the method proposed in [82] 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 k-means 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 other methods in almost all datasets. In the proposed method the appropriate initialization interval is found for the parameters of RBF using k-means. 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. Also, in order to study the effectiveness of the selection of parameter F an additional experiment were conducted, where the best fitness of the genetic algorithm is plotted for the WINE problem. The outcome of this experiment is graphically outlined in Figure 2. The graph shows that the behavior of the proposed method does not change significantly for different values of the parameter F. Also the plot for the WINE dataset of best, worst and average fitness for $F = 3$ is shown in Figure 3. An additional experiment was performed to evaluate the effect of the parameter change k on the results. In Figure 4 the plot for different values of k for the Housing dataset is outlined and in Figure 5 the same experiment is shown for the Z_F_S classification dataset. Of course from the value $k = 4$ onwards the error falls but not at the same rate. The proposed value $k = 10$ was used in all data sets in order to have a balance between the speed and the efficiency of the method. Finally, classification performance was evaluated based on two evaluation metrics: precision and recall for some datasets as show in Table 4.In these results, the precision of the proposed method showed the best classification performance under different datasets. We found the same trends in recall metric in the two of three datasets (spiral and EEG datasets).

Table 2: Classification error for different datasets.

DATASET	KRBF	GRBF	PROPOSED
Alcohol	46.63%	52.30%	21.86%
Appendicitis	12.23%	16.83%	16.03%
Australian	34.89%	41.79%	22.97%
Balance	33.42%	38.02%	12.88%
Cleveland	67.10%	67.47%	51.75%
Dermatology	62.34%	61.46%	37.37%
Glass	50.16%	61.30%	49.16%
Hayes Roth	64.36%	63.46%	35.26%
Heart	31.20%	28.44%	17.80%
HouseVotes	6.13%	11.99%	3.67%
Ionosphere	16.22%	19.83%	10.33%
Liverdisorder	30.84%	36.97%	28.73%
Mammographic	21.38%	30.41%	17.25%
Parkinsons	17.42%	33.81%	17.37%
Pima	25.78%	27.83%	24.00%
Popfailures	7.04%	7.08%	5.44%
Regions2	38.29%	39.98%	25.81%
Ring	21.65%	50.36%	2.09%
Saheart	32.19%	33.90%	29.38%
Segment	59.68%	54.25%	39.44%
Sonar	27.85%	34.20%	19.62%
Spiral	44.87%	50.02%	18.98%
Tae	60.07%	61.78%	52.44%
Thyroid	10.52%	8.53%	7.12%
Wdbc	7.27%	8.82%	5.29%
Wine	31.41%	31.47%	8.67%
Z_F_S	13.16%	23.37%	4.21%
ZO_NF_S	9.02%	22.18%	4.17%
ZONF_S	4.03%	17.41%	2.18%
ZOO	21.93%	33.50%	9.00%

Table 3: Regression error for different datasets.

DATASET	KRBF	GRBF	PROPOSED
ABALONE	2559.48	4161.66	1960.22
AIRFOIL	5.49	18.15	0.58
ANACALT	11.628	5.58	0.003
BK	0.17	0.21	0.23
BL	0.05	0.019	0.0009
CONCRETE	1.15	1.50	0.52
HOUSING	2884.09	4784.50	693.22
LASER	2.35	6.94	1.04
MB	11.33	2.44	0.63
NT	72.14	0.22	0.09
QUAKE	15.36	171.43	7.86

Figure 2: Plot of best fitness for the WINE problem for different values of parameter F.

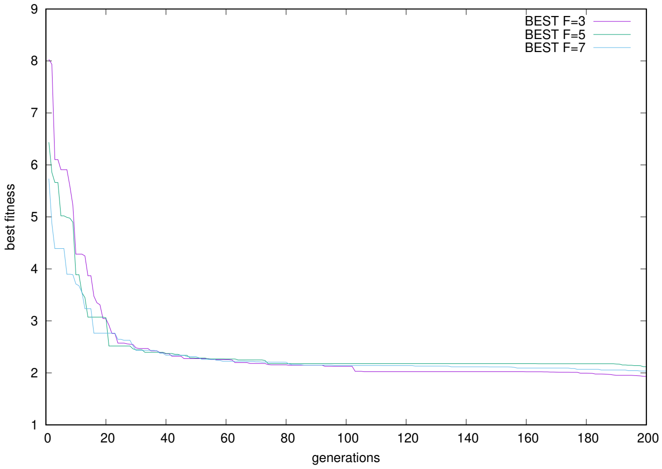


Table 4: Comparison of precision and recall between the traditional RBF and the proposed method for some datasets.

DATASET	PRECISION KRBF	RECALL KRBF	PRECISION PROPOSED	RECALL PROPO
HOUSEVOTES	90.61%	95.60%	96.44%	94.25%
SPIRAL	55.53%	55.93%	82.93%	84.66%
ZONF_S	92.76%	87.23%	97.27%	94.55%

Figure 3: Plot of best, worst and average fitness for the Wine dataset and $F = 3$.

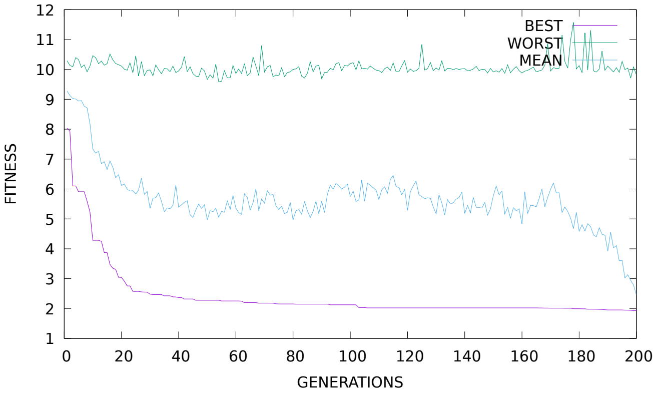


Figure 4: Experiments with different values of k for the Housing dataset.

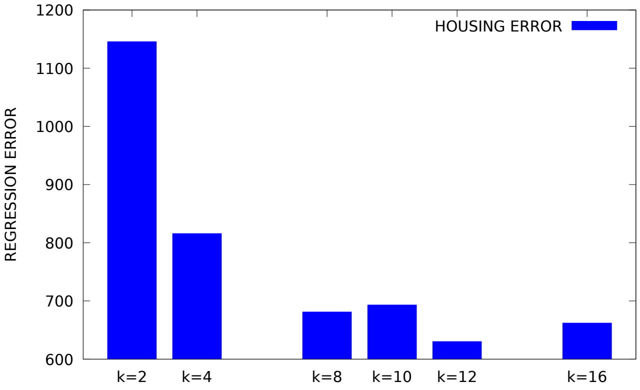
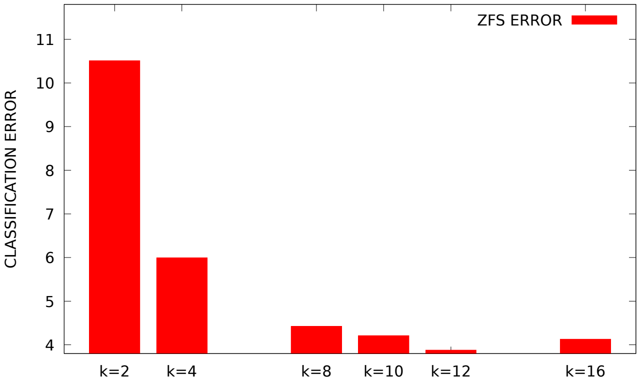


Figure 5: Classification error of Z_F_S dataset for different values of k.



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 k-means clustering phase of the method.

2. Dynamic selection of K in k-means 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.

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] H. Jiang, Y. Yang, M. Shi, Chemometrics in Tandem with Hyperspectral Imaging for Detecting Authentication of Raw and Cooked Mutton Rolls, Foods **10**, 2021.

[21] H. Feng, Q. Song, S. Ma, W. Ma, C. Yin, D., H. Yu, A new adaptive sliding mode controller based on the RBF neural network for an electro-hydraulic servo system, ISA Transactions, 2022.

[22] X. Li, H. Jiang, X. Jiang, M. Shi, Identification of Geographical Origin of Chinese Chestnuts Using Hyperspectral Imaging with 1D-CNN Algorithm, Agriculture **11**, 2021.

[23] A.H. Fath, F. Madanifar, M. Abbasi, Implementation of multilayer perceptron (MLP) and radial basis function (RBF) neural networks to predict solution gas-oil ratio of crude oil systems, Petroleum **6**, pp. 80-91, 2020.

[24] Y. Deng, X. Zhou, J. Shen, G. Xiao, H. Hong, H. Lin, F. Wu, B.Q. Liao, New methods based on back propagation (BP) and radial basis function (RBF) artificial neural networks (ANNs) for predicting the occurrence of haloketones in tap water, Science of The Total Environment **772**, 2021.

[25] H.R. Aqdam, M.M. Ettefagh, R. Hassannejad, Health monitoring of mooring lines in floating structures using artificial neural networks, Ocean Engineering **164**, Pages 284-297,2018.

[26] 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.

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

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

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

[30] 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.

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

[32] 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.

[33] 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.

[34] 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.

[35] 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.

[36] 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.

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

[38] 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.

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

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

[41] 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.

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

[43] 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.

[44] 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.

[45] J.R. Quinlan, Simplifying Decision Trees. International Journal of Man-Machine Studies **27**, pp. 221-234, 1987.

[46] T. Shultz, D. Mareschal, W. Schmidt, Modeling Cognitive Development on Balance Scale Phenomena, Machine Learning **16**, pp. 59-88, 1994.

[47] Z.H. Zhou,Y. Jiang, NeC4.5: neural ensemble based C4.5," in IEEE Transactions on Knowledge and Data Engineering **16**, pp. 770-773, 2004.

[48] R. Setiono , W.K. Leow, FERNN: An Algorithm for Fast Extraction of Rules from Neural Networks, Applied Intelligence **12**, pp. 15-25, 2000.

[49] G. Demiroz, H.A. Govenir, N. Ilter, Learning Differential Diagnosis of Eryhemato-Squamous Diseases using Voting Feature Intervals, Artificial Intelligence in Medicine. **13**, pp. 147–165, 1998.

[50] G. Valentini, F. Masulli, NEURObjects: an object-oriented library for neural network development, Neurocomputing **48**, pp. 623-646, 2002.

[51] T. Denoeux, A neural network classifier based on Dempster-Shafer theory, IEEE Transactions on Systems, Man and Cybernetics - Part A: Systems and Humans **30**, pp. 131-150, 2000.

[52] 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.

[53] I. Kononenko, E. Šimec, M. Robnik-Šikonja, Overcoming the Myopia of Inductive Learning Algorithms with RELIEFF, Applied Intelligence **7**, pp. 39–55, 1997.

[54] R.M. French, N. Chater, Using noise to compute error surfaces in connectionist networks: a novel means of reducing catastrophic forgetting, Neural Comput. **14**, pp. 1755-1769, 2002.

[55] J.G. Dy , C.E. Brodley, Feature Selection for Unsupervised Learning, The Journal of Machine Learning Research **5**, pp 845–889, 2004.

[56] S. J. Perantonis, V. Virvilis, Input Feature Extraction for Multilayered Perceptrons Using Supervised Principal Component Analysis, Neural Processing Letters **10**, pp 243–252, 1999.

[57] J. Garcke, M. Griebel, Classification with sparse grids using simplicial basis functions, Intell. Data Anal. **6**, pp. 483-502, 2002.

[58] M. Elter, R. Schulz-Wendtland, T. Wittenberg, The prediction of breast cancer biopsy outcomes using two CAD approaches that both emphasize an intelligible decision process, Medical Physics **34**, pp. 4164-4172, 2007

[59] M.A. Little, P.E. McSharry, E.J. Hunter, J. Spielman, L.O. Ramig, Suitability of dysphonia measurements for telemonitoring of Parkinson's disease, IEEE Trans Biomed Eng. **56**, 2009.

[60] J.W. Smith, J.E. Everhart, W.C. Dickson, W.C. Knowler, R.S. Johannes, Using the ADAP learning algorithm to forecast the onset of diabetes mellitus, In: Proceedings of the Symposium on Computer Applications and Medical Care IEEE Computer Society Press, pp.261-265, 1988.

[61] D.D. Lucas, R. Klein, J. Tannahill, D. Ivanova, S. Brandon, D. Domyancic, Y. Zhang, Failure analysis of parameter-induced simulation crashes in climate models, Geoscientific Model Development **6**, pp. 1157-1171, 2013.

[62] N. Giannakeas, M.G. Tsipouras, A.T. Tzallas, K. Kyriakidi, Z.E. Tsianou, P. Manousou, A. Hall, E.C. Karvounis, V. Tsianos, E. Tsianos, A clustering based method for collagen proportional area extraction in liver biopsy images, In: Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, , pp. 3097-3100, 2015.

[63] B.L. Bias, variance and arcing classifiers. Tec. Report 460, Statistics department. University of california. April 1996.

[64] T. Hastie, R. Tibshirani, Non-parametric logistic and proportional odds regression, JRSS-C (Applied Statistics) **36**, pp. 260–276, 1987.

[65] M. Dash, H. Liu, P. Scheuermann, K. L. Tan, Fast hierarchical clustering and its validation, Data & Knowledge Engineering **44**, pp 109–138, 2003.

[66] R.P. Gorman, T.J. Sejnowski, Analysis of Hidden Units in a Layered Network Trained to Classify Sonar Targets, Neural Networks **1**, pp. 75-89, 1988.

[67] T.S. Lim, W.Y. Loh, A Comparison of Prediction Accuracy, Complexity, and Training Time of Thirty-Three Old and New Classification Algorithms, Machine Learning **40**, pp. 203-228, 2000.

[68] W.H. Wolberg, O.L. Mangasarian, Multisurface method of pattern separation for medical diagnosis applied to breast cytology, Proc Natl Acad Sci U S A. **87**, pp. 9193–9196, 1990.

[69] M. Raymer, T.E. Doom, L.A. Kuhn, W.F. Punch, Knowledge discovery in medical and biological datasets using a hybrid Bayes classifier/evolutionary algorithm. IEEE transactions on systems, man, and cybernetics. Part B, Cybernetics : a publication of the IEEE Systems, Man, and Cybernetics Society, **33** , pp. 802-813, 2003.

[70] P. Zhong, M. Fukushima, Regularized nonsmooth Newton method for multi-class support vector machines, Optimization Methods and Software **22**, pp. 225-236, 2007.

[71] M. Koivisto, K. Sood, Exact Bayesian Structure Discovery in Bayesian Networks, The Journal of Machine Learning Research **5**, pp. 549–573, 2004.

[72] 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.

[73] W. J Nash, T.L. Sellers, S.R. Talbot, A.J. Cawthor, W.B. Ford, The Population Biology of Abalone (_Haliotis_ species) in Tasmania. I. Blacklip Abalone (_H. rubra_) from the North Coast and Islands of Bass Strait, Sea Fisheries Division, Technical Report No. 48 (ISSN 1034-3288), 1994.

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

[75] Jeffrey S. Simonoff, Analyzing Categorical Data, Springer-Verlag, New York, 2003.

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

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

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

[79] 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.

[80] 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

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

[82] S. Ding, L. Xu, C. Su et al, An optimizing method of RBF neural network based on genetic algorithm. Neural Comput & Applic 21, pp. 333–336, 2012.