Research Journal of Applied Sciences, Engineering and Technology 12(11): 1146-1151, 2016
DOI:10.19026/rjaset.12.2856
ISSN: 2040-7459; e-ISSN: 2040-7467

Submitted: December 2, 2015 Accepted: March 1, 2016 Published: June 05, 2016

Research Article

Automated Architecture Selection for Radial Basis Function Neural Networks

[1]Tiny du Toit, [2]Nawaf Hamadneh, [3]Saratha Sathasivam and [4]Waqar Khan
[1]School of Computer, Statistical and Mathematical Sciences, Potchefstroom Campus, North-West University, Private Bag X6001, Potchefstroom, 2531, South Africa
[2]College of Science and Theoretical Studies, Saudi Electronic University, Riyadh 13316, Saudi Arabia
[3]School of Mathematical Sciences, Universiti Sains Malaysia, 11800 USM, Penang, Malaysia
[4]Department of Mechanical and Industrial Engineering, College of Engineering, Majmaah University, Majmaah 11952, Saudi Arabia

Abstract: A new model selection algorithm is established to determine the best number of hidden neurons for radial basis function neural networks. We used a Bayesian information-based criterion to select the best number of hidden neurons. The new algorithm grows the number of hidden neurons while the Bayesian information-based criterion is used for improvement. The optimal parameter values of a current neural network are used in the subsequent architecture. The computational results are compared with the trial-and-error approach through publicly available data sets. It is found that the new algorithm is suitable to improve the performance of the neural networks automatically. The root mean square error function is used to measure out-of-sample performance.

Keywords: Model selection, radial basis neural network, schwarz information criterion

INTRODUCTION

Radial Basis Function Neural Networks (RBFNNs) (Broomhead and Lowe, 1988; Lee *et al*., 2011; Zhao *et al*., 2014; Wang *et al*., 2016) have elicited a lot of attention in the past. Apart from training faster than Multilayer Perceptron (MLP) networks, it provides a unifying link between classification, function approximation, regularization, density estimation and noisy interpolation (Ghodsi and Schuurmans, 2003). Training an RBFNN normally takes place in two steps (Wang and Zhu, 2000; Lampariello and Sciandrone, 2001; Hamadneh *et al*., 2012). Firstly, it determines the hidden parameters corresponding to the radial basis function. Secondly, it determines the output weights. Compared to MLP networks, one advantage of RBFNNs is that suitable parameters for the hidden layer can be found without performing non-linear optimization of the network parameters. Still, a critical issue remains for RBFNNs in the selection of the appropriate number of hidden neurons (basis functions). Complexity and generalization ability of the RBFNN are controlled by the number of hidden neurons. Too few hidden neurons results in poor prediction (inadequate generalization) on new data as the model has limited flexibility. Conversely, an RBFNN with too many hidden neurons also provides insufficient generalization as it is too flexible and fits the training data noise. A low variance, high bias estimator is obtained by a small number of hidden neurons whereas a high variance, low bias estimator is obtained by a large number of hidden neurons. To determine the best generalization performance, a trade-off between the conflicting requirements of reducing variance and simultaneously reducing bias must be accomplished. This compromise emphasizes the importance of optimizing the complexity of the RBFNN model to obtain the best generalization.

The N2C2S algorithm is used to select automatically the most suitable MLP networks architecture with a single hidden layer for classification (Setiono, 2001). Cross-validation or hold-out samples are utilized to determine when to stop adding hidden units to the network. This algorithm can be adapted to determine the best RBFNN topology. Many techniques have been proposed to determine automatically the RBFNN model complexity (Bors and Gabbouj, 1994; Leonardis and Bischof, 1998; Ghodsi and Schuurmans, 2003; Peng *et al*., 2006, 2007). In this study, we proposed a new algorithm to select automatically the most suitable RBFNN topology for classification. This algorithm is called the best number of hidden neurons method in RBFNN or BH-RBFNN. We utilized the SAS® commercial software to obtain our computational results. To select the best number of hidden neurons, we used an information-based model

Corresponding Author: Nawaf Hamadneh, College of Science and Theoretical Studies, Saudi Electronic University, Riyadh 13316, Saudi Arabia

selection criterion. The Schwarz Bayesian Criterion (SBC) in-sample model selection criterion is employed to determine the appropriate model complexity. In this study, we determined the parameters' values of the neural networks by using the Levenberg-Marquardt Algorithm (LMA) (Moré, 1978; Ye, 2003; Heertjes and Verstappen, 2014). In addition, the Root Mean Squared Error (RMSE) and classification accuracy indicate the how good the model will generalize to new unseen data. The global values' solutions, obtained by LMA, are based on the initial parameters (Moré, 1978; Gow and Manning, 1999).

The main objective is to select the best number of hidden neurons in RBFNNs. In the next two Sections, RBFNNs and model selection criteria are discussed respectively. After that, we proposed a new method to select the best RBFNN architecture. The experimental results obtained by applying the new technique to ten data sets are presented. Finally, the conclusions are presented in the last section.

RADIAL BASIS FUNCTION NEURAL NETWRK

RBFNNs have typically three layers, an input layer, hidden layer with a non-linear activation function and a linear output layer (Lowe, 1989; Moody and Darken, 1989; Lee *et al*., 2011; Hamadneh *et al*., 2012). An activation function in the hidden layer is a radial basis function determined by the distance between the input vector and a prototype vector. Originally, RBF models became popular for performing exact interpolation of a set of data points in high-dimensional space (Ghodsi and Schuurmans, 2003; Tilahun *et al*., 2013). The initial technique yielded an interpolating function that passed through every data point. Suppose there is a mapping from a *d*-dimensional input space to a one-dimensional target space *y*, with *N* input vectors x_i in the data set and with corresponding targets y_i, $i = 1, ..., N$. An exact interpolation is obtained by creating a set of *N* basis functions (one for each data set) and then determining the linear combination weights of the basic functions. For exact interpolation, the radial basis functions are non-linear functions, $\varphi(\|X - c_i\|)$, of the input vectors *X* with c_i the centre of the function. A linear combination of these basis functions can be denoted as:

$$\sum_{i=1}^{N} w_i \varphi(\|X - c_i\|).$$

The exact interpolation procedure can be modified in a number of ways to obtain an RBFNN. First, the number of data points, *N*, is usually much greater than the number of radial basis functions, *M*. Second, rather than obtaining the centres of the radial basis functions using the input data vectors, they can be determined during the training process. Third, each radial basis function can have its own width parameter, σ_j, which is also determined by the training process. When these changes are applied to the original (exact) interpolation formula, the following form of the RBFNN mapping is obtained:

$$y_k(X) = \sum_{j=1}^{M} w_{kj} \varphi_j(X) + w_{k0}$$

where, the biases w_{k0} can be incorporated into the final summation by including an extra basis function φ_0 with activation fixed at 1, M is the number of hidden neurons and X is an input value.

Of all the radial basis function forms, considered in previous research on RBF models, the most common form is the Gaussian function:

$$\varphi_j(X) = \exp\left(-\frac{\|X - c_j\|^2}{2\sigma_j^2}\right)$$

where, X is the *d*-dimensional input vector with elements x_i and c_j and σ_j are the centre and width respectively of basis function φ_j.

In practice, training an RBFNN is performed in two steps. The first step determines the basis function parameters c_j and σ_j based on the *X*-values of the training samples. Then, while the basic functions are kept fixed, the second-layer weights w_i are estimated. In this study, we determined the parameters' values by using the Levenberg-Marquardt algorithm (Moré, 1978; Ye, 2003).

For the first phase, the generalized Lloyd algorithm or Kohonen's self-organizing maps can be exploited. Consider the RBFNN mapping for the second phase. When the bias parameters are absorbed into the weights, this can be written in matrix notation as $Y = W \varphi$ where Y is the matrix of output values and $W = (w_{kj})$ is a matrix of the weights in the second layer to be estimated. This formulation is a classical least squares estimation problem. For $\|Y - W\varphi\|^2$ to be minimized, a necessary condition is that W must satisfy $W = (\varphi^T \varphi)^{-1} \varphi^T Y$.

In the next section the model selection criterion which forms the foundation of the new BH-RBFNN algorithm is discussed.

MODEL SELECTION CRITERION

Information-based model selection criteria such as the Schwarz Information Criterion or Schwarz Bayesian criterion (SBC), which penalize large models that often tend to over-fit, are the most widely used in-sample model selection criteria (Akaike, 1974; Schwarz, 1978; Cong and Brady, 2012). In the BH-RBFNN algorithm, the SBC is applied for selecting the most appropriate RBFNN model. Use of this metric is based on the assumption that the true RBFNN model is included in the set of candidate RBFNN models and as a result consists of a finite number of parameters (Du Toit, 2006). By making this assumption, the goal of model selection becomes that of correctly identifying the true model from a list of candidate models generated by the BH-RBFNN algorithm. Furthermore, it is assumed that all variables can be measured and the list of important variables is known. This is clearly the case for the ten publicly available data sets used in this study. The Gaussian error model formulation of the SBC is employed (Burnham and Anderson, 2002):

$$SBC = n\ \ln(MSE) + K\ \ln(n)$$

where,
n = The number of data points
K = The total number of free parameters which are the centres, widths and output weights

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(\hat{Y}_i - Y_i)^2$$

where,
$\hat{Y}_i$ = The predicted target value
Y_i = The target value

In the computational results, the RMSE and predictive accuracy on the testing data sets have also been included for reference. The first term in SBC measures the goodness-of-fit of the model to the data and tends to become less as more parameters are added to the model, while the second term increases as more parameters are added to the model. The latter term sets a penalty for model over-fitting. The optimal model is selected when SBC is minimized. As a result, SBC is a reasonable criterion for RBFNN model selection, which balances model fitting and model parsimony.

A NEW METHOD FOR RBFNN ARCHITECTURE SELECTION

In this section, we have established a new method for selecting the best number of the hidden neurons in RBFNNs, called BH-RBFNNs. The algorithm simulates a trial-and-error approach to determine the best architecture. The algorithm commences with a simple RBFNN model. Ineachiteration, the best parameters found in the previous iteration are utilized as starting parameters. The model is then grown until the SBC value does not further improve. In the latter case, the same RBFNN architecture is trained from random starting parameters and only if the SBC value does not again improve the algorithm terminates. Suppose there are S data samples. Partition the data set randomly into two disjoint subsets: a training set and a testing data set. The training data is used for model selection and to improve the performance of RBFNNs by determining the best set of parameters. The testing data set is used to reveal the generalization capability of the neural network. The steps be as follows:

Step 1: Let $\mathcal{N}_1$ be an RBFNN with N input neurons, M output neurons and H hidden neurons.

Step 2: Initialize the parameters of $\mathcal{N}_1$ randomly and train $\mathcal{N}_1$ to minimize the RMSE error function. Let the SBC model selection criterion value of the trained RBFNN be $\text{SBC}_{\mathcal{N}1}$.

Step 3: Let $\mathcal{N}_2$ be a network with N input neurons, M output neurons and $H+h$ hidden neurons.

Step 4: Set the parameters to and from the first H hidden neurons of $\mathcal{N}_2$ to the optimal parameters of $\mathcal{N}_1$ and set the remaining parameters of $\mathcal{N}_2$ randomly. Train $\mathcal{N}_2$ to minimize the RMSE error function and let the SBC model selection criterion value of the trained RBFNN be $\text{SBC}_{\mathcal{N}2}$.

Step 5: If $\text{SBC}_{\mathcal{N}2} < \text{SBC}_{\mathcal{N}1}$ then

a) Set $H{:} = H+h$.
b) Let $\mathcal{N}_1{:} = \mathcal{N}_2$ and $\text{SBC}_{\mathcal{N}1}{:} = \text{SBC}_{\mathcal{N}2}$.
c) If $H < MaxH$ then return to Step iii.
Else:
1. Train a new RBFNN, $\mathcal{N}_3$, with $H+h$ hidden neurons and all parameters initially assigned random values to minimize the RMSE error function. Let the SBC model selection criterion value of the trained RBFNN be $\text{SBC}_{\mathcal{N}3}$.
2. If $\text{SBC}_{\mathcal{N}3} < \text{SBC}_{\mathcal{N}1}$ then
a) Set $H{:} = H+h$.
b) Let $\mathcal{N}_1{:} = \mathcal{N}_3$ and $\text{SBC}_{\mathcal{N}1}{:} = \text{SBC}_{\mathcal{N}3}$.
c) If $H<\text{Max}H$ then return to Step iii.

Step 6: Output $\mathcal{N}_1$ as the final constructed RBFNN.

EXPARIMANTAL RESULTS

For the experimental results, the starting number of hidden neurons was 1. The growing network was expanded by one neuron at a time (h=1). A good starting value as one hidden neuron is the simplest neural network we can use as the starting point since we do not know how many neurons will be needed. The maximum number of hidden neurons allowed was 10 ($maxH$). Parameter optimization was performed by the Levenberg-Marquardt algorithm (Moré, 1978). Training was terminated if the relative decrease in RMSE error function values after two consecutive iterations was less than 10^{-4} or the maximum number of function evaluations was reached. The latter was set to 10000. In this study the Kyphosis data set available in http://www.jhsph.edu/ and nine data sets from the UCI Machine Learning Repository (Bache and Lichman, 2013) were used in Table 1. Note that the missing values were ignored. Table 2 presents results for a trial-and-error with 10-fold cross-validation model selection approach (grid search) by varying the number of hidden neurons (basis functions) between 1 and 10. In addition, the RMSE values and classification accuracies obtained on the test sets are displayed.

Table 1: Data sets used in the experiments

			Attributes			RBFNN	
Data Set	Size	Missing values (%)	Continuous	Binary	Nominal	Inputs	Outputs (binary)
Kyphosis	83	0	3	0	0	3	1
Breast cancer	699	2.3	9	0	0	9	1
Iris	150	0	4	0	0	4	3
Thyroid	215	0	5	0	0	5	3
Contraceptive method choice	1473	0	2	3	4	9	3
Vertebral column	310	0	6	0	0	6	3
Indian liver patient	583	0	9	1	0	10	1
Fertility	100	0	1	6	2	9	1
Seeds	210	0	7	0	0	7	3
MAGIC Gamma telescope	19020	0	10	0	0	10	1

Table 2: Results obtained using trial-and error approach

Data Set	Hidden neurons	SBC	RMSE	Accuracy (%)
Kyphosis	1	-140.990	0.359	81.803
	9	-167.943	2.736	77.432
Breast cancer	1	-2170.773	0.166	96.301
Iris	4	-1428.518	0.164	96.245
	10	-1776.928	0.646	94.462
Thyroid	3	-2035.104	0.161	96.681
	10	-2696.808	1.350	96.225
Contraceptive method choice	3	-6540.808	0.429	55.316
	10	-5906.410	0.427	55.489
Vertebral column	2	-2038.107	0.283	84.744
	3	-2017.947	0.279	84.774
Indian liver patient	1	-839.434	0.426	72.398
	2	-776.302	0.425	73.175
Fertility	1	-166.227	0.315	85.900
Seeds	5	-1767.129	0.190	95.330
	8	-1848.087	0.437	93.427
MAGIC Gamma telescope	7	-36816.548	0.334	85.727
	8	-36893.294	0.416	85.851

Table 3: Results obtained using BH-RBFNN approach

Data Set	Hidden neurons	SBC	RMSE	Accuracy (%)
Kyphosis	1.5±0.85	-141.3400	0.705	79.365
Breast cancer	3.9±2.56	-2299.112	0.443	96.804
Iris	7.2±2.15	-1625.348	0.211	95.357
Thyroid	6.9±3.07	-2348.396	0.233	96.142
Contraceptive method choice	2.3±0.67	-6478.357	0.433	53.619
Vertebral column	2.7±0.95	-2023.838	0.285	85.002
Indian liver patient	1.2±0.42	-830.2830	0.429	70.779
Fertility	1.1±0.32	-172.9950	0.347	85.118
Seeds	4.3±1.70	-1754.214	0.441	95.073
MAGIC Gamma telescope	6.3±2.54	-37450.890	0.330	86.304

For most of the data sets the best number of hidden neurons (indicated by bold values) suggested by the SBC and RMSE differ. In Table 2 this can be observed by two rows for the specific data set. The RMSE and classification accuracy indicate the how good the model will generalize to new unseen data. It is well known that in-sample model selection does not always agree with out-of-sample model selection as can be seen from all the data sets except the Breast Cancer and Fertility data sets (Kingston *et al*., 2005). For these two data sets there is only one row in Table 2 as the SBC and RMSE both suggest the same best number of neurons. Since there is no general consensus which criterion is better it is difficult to decide which model to choose. The BH-RBFNN algorithm produced the 10-fold cross-validation results shown in Table 3. For each Table 3: Results obtained using BH-RBFNN approach fold the best number of hidden neurons is identified by the SBC value. At this critical number of hidden neurons the algorithm could not improve further by adding another hidden node. In Table 3 the average values over the 10 folds are shown. The±in the second column is a way to express the statistical standard deviation. For example, ±0.85 it basically means the standard deviation is in the range (average-0.85, average+0.85). The 1.5 hidden neurons is the average of the 10-fold cross validation experiments. Each experiment had an integer number of hidden neurons, but 1.5 is the average of them. Note that, in BH-RBFNN algorithm, training the neural networks by LMA is to minimize the RMSE and the SBC is to select the best model. As mentioned above, the main objective of MLA is determined the best local minimum value in terms of RMSE, which is not necessarily being the global minimum value. LMA converges to the best local minimum value only if the initial parameters are already somewhat close to that solution. Accordingly, The optimal solution to RBFNNs in terms of RMSE, which determined by MLA, is based on the initial parameters. In trial-and-error method, we used randomly initial parameters values of each time. While, in BH-RBFNN algorithm the initial parameters were fixed and only the new parameters we determine randomly. For that, there is more chance for trial-and-error method to determine the best solutions. Nevertheless, the BH-RBFNN technique produced better results than the trial-and-error approach for the Breast Cancer, Fertility and MAGIC Gamma Telescope data sets (Table 2 and 3). For the latter data set, the RBFNN model is also more parsimonious than the models obtained by the trial-and-error approach in terms of SBC and the speed. So, BH-RBFNN algorithm is quicker to run than the trial and error method.

CONCLUSION

In this study we have established a new approach to determine the best RBFNN topology for a specific pattern classification problem. This constructive approach minimizes the SBC information-based criterion to identify the optimal complexity and consequently generalization capability of the model. As a result, the difficulty of choosing the number of hidden neurons before network training begins is removed. Although there are several network construction algorithms in the literature, the BH-RBFNN is simple to implement, provides good results and is based on a proven topology selection strategy. Furthermore, it relieves the data modeller from making difficult decisions regarding the complexity of the RBFNN architecture. Compared to a trial-and-error procedure, the algorithm may select a more parsimonious topology and speed which is less prone to over-fitting. However one can show analytically that SBC generally gives preference to simpler models because it penalizes complex models more harshly.

REFERENCES

Akaike, H., 1974. A new look at the statistical model identification. IEEE T. Automat. Contr., 19(6): 716-723.

Bache, K. and M. Lichman, 2013. UCI machine learning repository. School of Information and Computer Science, University of California, Irvine, Retrieved from: http://archive.ics.uci.edu/ml.

Bors, A.G. and M. Gabbouj, 1994. Minimal topology for a radial basis functions neural network for pattern classification. Digit. Signal Process, 4(3): 173-188.

Broomhead, D.S. and D. Lowe, 1988. Radial basis functions, multi-variable functional interpolation and adaptive networks. Technical Report, DTIC Document.

Burnham, K.P. and D.R. Anderson, 2002. Model Selection and Multimodel Inference: A Practical Information-theoretic Approach. 2nd Edn., Springer, New York.

Cong, R.G. and M. Brady, 2012. The interdependence between rainfall and temperature: Copula analyses. Sci. World J., 2012: 11.

Du Toit, J.V., 2006. Automated construction of generalized additive neural networks for predictive data mining. Ph.D. Thesis, North-West University, Potchefstroom Campus.

Ghodsi, A. and D. Schuurmans, 2003. Automatic basis selection techniques for RBF networks. Neural Networks, 16(5-6): 809-816.

Gow, J.A. and C.D. Manning, 1999. Development of a photovoltaic array model for use in power-electronics simulation studies. IEE P-Elect. Pow. Appl., 146(2): 193-200.

Hamadneh, N., S. Sathasivam, S.L. Tilahun and O.H. Choon, 2012. Learning logic programming in radial basis function network via genetic algorithm. J. Appl. Sci., 12(9): 840-847.

Heertjes, M. and R. Verstappen, 2014. Self-tuning in integral sliding mode control with a Levenberg–Marquardt algorithm. Mechatronics, 24(4): 385-393.

Kingston, G.B., H.R. Maier and M.F. Lambert, 2005. A bayesian approach to artificial neural network model selection. Proceeding of the International Congress on Modelling and Simulation (MODSIM, 05). Melbourne, Vic., pp: 1853-1859.

Lampariello, F. and M. Sciandrone, 2001. Efficient training of RBF neural networks for pattern recognition. IEEE T. Neural Networ., 12(5): 1235-1242.

Lee, T.Y., S.A. Chen, H.Y. Hung and Y.Y. Ou, 2011. Incorporating distant sequence features and radial basis function networks to identify ubiquitin conjugation sites. PloS One, 6(3): e17331.

Leonardis, A. and H. Bischof, 1998. An efficient MDL-based construction of RBF networks. Neural Networks, 11(5): 963-973.

Lowe, D., 1989. Adaptive radial basis function nonlinearities, and the problem of generalisation. Proceeding of the 1st IEE International Conference on Artificial Neural Networks. London, pp: 171-175.

Moody, J. and C.J. Darken, 1989. Fast learning in networks of locally-tuned processing units. Neural Comput., 1(2): 281-294.

Moré, J.J., 1978. The Levenberg-Marquardt Algorithm: Implementation and Theory. In: Watson, G.A. (Ed.), Numerical Analysis. Springer, Berlin, Heidelberg, 630: 105-116.

Peng, J.X., K. Li and D.S. Huang, 2006. A hybrid forward algorithm for RBF neural network construction. IEEE T. Neural Networ., 17(6): 1439-1451.

Peng, J.X., K. Li and G.W. Irwin, 2007. A novel continuous forward algorithm for RBF neural modelling. IEEE T. Automat. Contr., 52(1): 117-122.

Schwarz, G., 1978. Estimating the dimension of a model. Ann. Stat., 6(2): 461-464.

Setiono, R., 2001. Feedforward neural network construction using cross validation. Neural Comput., 13(12): 2865-2877.

Tilahun, N.H.S.L., S. Sathasivam and O.H. Choon, 2013. Prey-predator algorithm as a new optimization technique using in radial basis function neural networks. Res. J. Appl. Sci., 8(7): 383-387.

Wang, T., H. Gao and J. Qiu, 2016. A combined adaptive neural network and nonlinear model predictive control for multirate networked industrial process control. IEEE T. Neural Networ. Lear. Syst., 27(2): 416-425.

Wang, Z.O. and T. Zhu, 2000. An efficient learning algorithm for improving generalization performance of radial basis function neural networks. Neural Networks, 13(4-5): 545-553.

Ye, N., 2003. The Handbook of Data Mining. Lawrence Erlbaum Associates, Publishers, Mahwah, NJ.

Zhao, Z.Q., X.D. Wu, C.Y. Lu, H. Glotin and J. Gao, 2014. Optimizing widths with PSO for center selection of Gaussian radial basis function networks. Sci. China Inform. Sci., 57(5): 1-17.