Extracting Rules from RBF Neural Networks Using Reduced Data Attribute Sets

Xiuju Fu and *Lipo Wang**

School of Electrical and Electronic Engineering
Nanyang Technological University
Block S2, Nanyang Avenue
Singapore 639798
Email: {p146793114,elpwang}@ntu.edu.sg
http://www.ntu.edu.sg/home/elpwang
*Corresponding author

Abstract

Extracting concise rules from neural networks is a challenging task, especially when the dimensionality of inputs is high. Removing the redundant or irrelevant attributes is the process of reducing the dimensionality of data, increasing classifier efficiency, and maintaining high classification accuracy, which may lead to more compact rules representing the concept of data. In this paper, a separability-correlation measure (SCM) is proposed to rank attribute importance. An RBF classifier is used to evaluate the best subset of attributes. Based on the retained subset of attributes used as the inputs to the RBF neural network, rules are extracted. Simulations show that our method leads to more compact rules.

1 Introduction

Neural networks are frequently used in data mining. The task of extracting concise, i.e., a small number of, rules from a trained neural network is usually challenging due to the complicated architecture of a neural network [1][16].

An RBF network is generally much easier to train than Multi-layer perceptron (MLP). In recent years, rule extraction from RBF neural networks has been carried out by some researchers. In [4][6], redundant inputs are removed before extracting rules. Huber [8] selects rules according to importance. However, the accuracy is reduced with pruning. McGarry [11][12][13] extracts rules from RBF neural networks by considering the parameters of Gaussian kernel functions and weights which connect hidden units to the output layer. However, when the number of rules is small, the accuracy is low. When the accuracy is acceptable, the number of rules becomes large.

Data dimensionality reduction (DDR) is an important task in data processing, which helps us clean data and save cost for further processes. In DDR, irrelevant attributes are found and then removed from the original attribute set to reduce the number of attributes while maintaining the concept of data. DDR has become an important aspect of pattern recognition, since operators and automated controllers are able to make better use of lower-dimensional data compared to higher-dimensional ones. Reduced data dimensionality leads to less complicated network structure and thus increased efficiency in processing data.

In order to obtain more compact rules, on the one hand, data dimensionality reduction (DDR) is needed before inputting data to neural networks. We propose a rule extraction algorithm based on RBF neural networks. The simple architecture and the local approximation in kernel functions of RBF neural networks make extracting rules from RBF networks more desirable than from other neural networks. On the other hand, reducing the number of hidden units of RBF neural networks should be carried out before extracting rules.

For reducing the dimensionality of data, we rank the attribute importance first. In this paper, a separability-correlation measure (SCM) is proposed for determining the importance of the original attributes. The SCM is composed of the intraclass distance to interclass distance ratio and an attribute-class correlation measure. The magnitude SCM corresponding to a certain attribute gives the importance of the attribute. An RBF classifier is used to remove unimportant attributes using the ranking result from SCM. Large overlaps between clusters of the same class are allowed, which leads to a less number of hidden units. Then rule extraction is carried out based on the results of DDR.

We present the SCM measure for ranking the importance of attributes in Section 2. In Section 3, we introduce how to construct an RBF neural network classifier efficiently by allowing for large overlaps between clusters of the same class. Section 4 shows our rule extraction algorithm. Experimental results are shown in Section 5. Finally, we conclude the paper in Section 6.

2 Separability-Correlation Measure

Class separability and the correlation between attributes and class labels are used to measure the importance of each attribute. The probability of correct classification is large, when the distances between different classes are large. Therefore, the subset of features which can maximize the separability between classes is a desirable objective of feature selection. Class separability may be measured by the intraclass distance S_w and the interclass distance S_b [5]. The greater S_b is and the smaller S_w is, the better the separability of the data set is. The ratio of S_w and S_b is calculated and is used to measure the separability of the classes: the smaller the ratio, the better the separability. If omitting attribute k_1 from the data set leads to less class separability, i.e., a greater S_w/S_b, compared to the case where attribute k_2 is removed, attribute k_1 is more important for classification of the data set than attribute k_2, and vice versa. Hence the importance of the attributes can be ranked by computing the intraclass-to-interclass distance ratio with each attribute omitted in turn.

In addition, we propose to use the correlation C_k between the changes in attributes and their corresponding changes in class labels as another indication of importance of attribute k in classifying the patterns.

Hence we propose a separability-correlation measure (SCM) R_k as the sum of the class separability measure S_{wk}/S_{bk} and the correlation measure C_k (k refers to the k-th attribute), where S_{wk} and S_{bk} are intraclass and interclass distances calculated with the k-th attribute omitted from each pattern, respectively.

The importance level of attributes is ranked through the value of R_k. The greater the magnitude of R_k, the more important the k-th attribute.

3 A Simplified RBF Classifier

RBF neural networks accomplish non-linear mapping from attributes (inputs) to class labels (outputs). If there are M classes in the dataset, the non-linear mapping is as follows:

$$y_m(\mathbf{X}) = \sum_{j=1}^{K} w_{mj} \o_j(\mathbf{X}) + w_{m0} b_m \quad . \qquad (1)$$

Here $\mathbf{X}$ is the input pattern vector (n-dimension). $m = 1, 2, ..., M$. K is the number of hidden units, M is the number of output. w_{mj} is the weight connecting the j-th hidden unit to the m-th output node. b_m is the bias, w_{m0} is the weight connecting the bias and the m-th output node. $\o_j(\mathbf{X})$ is the activation function of the j-th hidden unit:

$$\o_j(\mathbf{X}) = \mathrm{e}^{\frac{||\mathbf{X}-\mathbf{C_j}||^2}{2\sigma_\mathbf{j}^2}} \quad , \qquad (2)$$

where $\mathbf{C_j}$ and σ_j are the center and the width for the j-th hidden unit, respectively, and are adjusted during learning.

In the RBF neural network, the output of a hidden unit is based on the distance between the input vector and the center vector of the hidden unit. The weights connecting the hidden layer and the output layer, can be calculated by the linear least square (LLS) method [2], which is fast and free of local minima, in contrast to the multilayer perceptron neural network.

There are two kinds of overlaps. One is the overlap between clusters of different classes. In [14], it is shown that overlapped receptive fields of different clusters can improve the performance of the RBF classifier in rejecting noise when tackling with noisy data. In [15] and [10], overlapping Gaussian kernel functions are created to map out the territory of each cluster with a less number of Gaussians. Small overlaps between the Gaussians for different classes are measured by the ratio between the number of the in-class patterns and the number of the out-class patterns in this cluster. A pre-defined value θ is set, i.e., the ratio should be not less than θ to guarantee a small classification error rate for the training data set (this is θ-criterion). When small overlaps among clusters for *different classes* are permitted, small overlaps among clusters for the *same class* may exist.

The other kind of overlaps is the overlap between clusters of the same class. In this paper, by allowing for large overlaps between clusters for *the same class*, we can further reduce the number of clusters substantially. This will lead to a simplified RBF neural network.

Through the algorithm described in Section 2, the importance level of each attribute is obtained. The subset of attributes which includes k most important attributes is input to the RBF classifier, and for $k = 1, 2, ..., N$, classification error rate is calculated for each k. For small k, classification error decreases as k increases until all important attributes are included. If for some k_1, the classification rate for $k = k_1 + 1$ is greater than that for $k = k_1$, then attributes $(k_1 + 1)$, $(k_1 + 2), ..., N$ are considered irrelevant.

4 Rule Extraction

We explain the concept of data by rules extracted from RBF classifiers. By training the RBF neural network, the concept of data is memorized in the construction of the RBF classifier. Since each hidden unit of the RBF neural network is responsive to a subset of patterns (instances), the weights connecting the hidden unit with output units can reflect for which output the hidden unit serves. Our rule extraction algorithm is directly based on the widths, centers of Gaussian kernel functions, and weights connecting hidden units and the output layer. Obtaining the important attributes by the method stated in our previous section, the important attributes are used as the input of RBF neural networks.

First, we determine the corresponding output unit which each hidden unit serves for through simplifying the weights between hidden units and output units. The maximum value of each row of the weight matrix W is converted into 1, and others are converted to be 0. Thus, the new weight matrix W_1 reflects the corresponding output which each hidden unit mainly serves.

The suitable interval of attributes composes of the premise parts of extracted rules. Let us assume that the number of attributes is N. The upper limit $Upper(j, i)$ and the lower limit $Lower(j, i)$ of the jth attribute in the ith rule are initialized as:

$$Upper(j, i) = \mu_i + \rho_{i,j} \quad , \tag{3}$$

$$Lower(j, i) = \mu_i - \rho_{i,j} \quad , \tag{4}$$

$$\rho_{i,j} = \eta_j * \overline{x}_{ik_i} * \sigma_i \quad . \tag{5}$$

where μ_i is the jth item of the center of the ith kernel function, initially, $\eta_j = 1$, σ_i is the width of the ith kernel function. $\overline{x}_{ik_i}$ is the corresponding element in W_1. $\rho_{i,j}$ will be adjusted according to our iteration steps as follows.

η_j is modified according to:

$$\eta_j = \eta_j + Sign * 0.025. \tag{6}$$

$Sign$ has two value $+1$ and -1, which is determined by the trend of change in the rule accuracy when adjusting. The initial $Sign$ is $+1$. If the rule accuracy becomes lower with the adjusted η, $Sign$ is changed to -1. Otherwise, $Sign$ remains the same. The stop-criterion for the iteration is a predefined rule error rate. When adjusting the intervals to obtain high accuracy, the validation set is used for determining the direction of adjusting, which can help the extraction rules not too fit to the training data set, and obtain good results in the testing data set.

5 Experimental Results

Iris data set is used for testing our method. There are 4 attributes in Iris data set. The data set is divided into 3 parts, i.e., training, validation, and test sets. In 150 patterns of Iris data set, 90 patterns are for training, 30 for validation, 30 for testing. We set $\alpha = 0.1$ and $\theta = 7$ in our experiments. We set $\alpha_1 = 0.8$ and we set the maximum epoch number of searching one cluster is R_Count=20. The experiment is repeated 5 times with different initial conditions and the average results are recorded.

Our simulations show that by allowing for large overlaps between clusters of the same class, the number of hidden units of the RBF classifier is decreased from 5.2 to 4 on average. The classification error rate just increase a little bit. The importance ranking order of the attributes according to our SCM is: 4, 3, 1, 2 for Iris data set. The classification error rates of the RBF classifier for different subsets of attributes in the order of importance are calculated. As the number of attributes used increases the validation error first decreases, reaches minimum when attributes 4 and 3 are used, and then increases. Hence in the Iris data set, attributes 1 and 2 are considered to be unimportant for classification and are then removed. This decreases the classification error rate from 0.0467 to 0.0333, the number of inputs from 4 to 2, and the number of hidden units of the RBF neural network from 4 to 3.

The attributes left after our DDR procedure are used as inputs to an RBF neural network. After removing unimportant attributes, the accuracy of the RBF classifier becomes better. The architecture of the RBF neural network is simplified by DDR, and at

Table 1: Comparisons with other rule extraction algorithms

comparison	rule number	average number of premises	rule accuracy
McGarry1 [11]	53	4	100%
McGarry2 [12]	3	4	40%
McGarry3 [13]	3	4	85%
Halgamuge [6]	5-6	-	-
Huber [8]	8	-	-
Our method	3	2	93.3%

"-" represents that the value is not available

the same time, DDR lead to more compact rules, i.e., a less number of premises in each rule.

By the rule-extraction algorithm described in Section 3, the rule set for Iris is obtained. There are 3 rules in the rule set. There are two premises corresponding to the two attributes left in each rule. The accuracy of obtained rules is 93.3%.

In Table 1, we compare our rule extraction algorithm with other rule extraction algorithms based on RBF neural networks. It shows that our method obtain better performance with a less number of premises, a less number of rules, and higher accuracy.

On the other hand, many rule-extraction methods have been explored based on the MLP. Desirable results have been obtained both in accuracy and numbers of rules (e.g., [3][7][9]). Compared with the rule extraction techniques using MLP, the accuracy of the rules extracted from the RBF neural networks is lower, however, the training of the RBF neural network can escape from local minima, which is very important for large data sets. The architecture of the RBF neural network is simpler and the training time is usually shorter in comparison with the MLP.

6 Conclusion

In this paper, we propose a novel attribute importance ranking method called SCM in order to reduce the computational burden for selecting a suitable attribute subset. Irrelevant attributes are deleted from the original attribute set, which lead to lower dimensional inputs to the RBF classifier according to the ranking results from SCM. Experimental results show that the method proposed is effective in reducing the size of data sets and reducing the structural complexity of the RBF neural network. A simplified RBF classifier is constructed by allowing for large overlaps between clusters of the same class. Then the results obtained are used extracting concise rules. More concise rules are extracted, i.e., the number of rules is decreased, the premises of rules are simplified and the accuracy is desirable, after removing the unimportant attributes in data and simplifying the architecture of RBF neural networks.

References

[1] R. Andrews, J. Diederich, and A. B. Ticle, "Survey and critique of techniques for extracting rules from trained artificial neural networks", *Knowledge-Based Syst.* vol. 8, no. 6, pp. 373-389, 1995.

[2] C. M. Bishop, *Neural network for pattern recognition*, Oxford University Press Inc., New York, 1995.

[3] G. Bologna and C. Pellegrini, "Constraining the MLP power of expression to facilitate symbolic rule extraction", *Proc. IEEE World Congress on Computational Intelligence*, vol. 1, pp. 146-151, 1998.

[4] T. Brotherton, G. Chadderdon, and P. Grabill, "Automated rule extraction for engine vibration analysis", *Proc. 1999 IEEE Aerospace Conference*, vol. 3, pp. 29-38, 1999.

[5] P. A. Devijver, and J. Kittler, *Pattern recognition: a statistical approach*, Prentice-Hall International, Inc. London, 1982.

[6] S. K. Halgamuge, W. Poechmueller, A. Pfeffermann, P. Schweikert, and M. Glesner, "A new method for generating fuzzy classification systems using RBF neurons with extended RCE learning Neural Networks", *Proc. IEEE World Congress on Computational Intelligence*, vol. 3, pp. 1589-1594, 1994.

[7] E. R. Hruschka and N. F. F. Ebecken, "Rule extraction from neural networks: modified RX algorithm", *Proc. International Joint Conference on Neural Networks*, Vol. 4, pp. 2504-2508, 1999.

[8] K.-P. Huber and M. R. Berthold, "Building precise classifiers with automatic rule extraction", *Proc. IEEE International Conference on Neural Networks*, vol. 3, pp. 1263-1268, 1995.

[9] H. Ishibuchi and T. Murata, "Multi-objective genetic local search for minimizing the number of

fuzzy rules for pattern classification problems", *IEEE World Congress on Computational Intelligence*, vol. 2, pp. 1100-1105, 1998.

[10] T. KaylaTni, and S. Dasgupta, "A new method for initializing radial basis function classifiers systems", IEEE International Conference on Man, and Cybernetics, Vol.3, pp.2584-2587, 1994.

[11] K. J. McGarry, S. Wermter, and J. MacIntyre, "Knowledge extraction from radial basis function networks and multilayer perceptrons", *Proc. International Joint Conference on Neural Networks*, vol. 4, pp. 2494-2497, 1999.

[12] K. J. McGarry, J. Tait, S. Wermter, and J. MacIntyre, "Rule-extraction from radial basis function networks", *Proc. Ninth International Conference on Artificial Neural Networks*, vol. 2, pp. 613-618, 1999.

[13] K. J. McGarry and J. MacIntyre, "Knowledge extraction and insertion from radial basis function networks", *IEE Colloquium on Applied Statistical Pattern Recognition (Ref. No. 1999/063)*, pp. 15/1-15/6, 1999.

[14] P. Maffezzoni, and P. Gubian, "Approximate radial basis function neural networks(RBFNN) to learn smooth relations from noisy data", Proceedings of the 37th Midwest Symposium on Circuits and Systems, Vol.1, pp.553 -556, 1994.

[15] A. Roy, S. Govil, and R. Miranda, "An algorithm to generate radial basis function (RBF)-like nets for classification problems", Neural networks, Vol.8, No.2, pp.179-201, 1995.

[16] H. Tsukimoto, "Extracting rules from trained neural networks", *IEEE Transactions on Neural Networks*, Vol. 11, Issue: 2, Page(s): 377 -389, March 2000.