Knowledge Discovery From a Simplified RBF Neural Network

Xiuju Fu and *Lipo Wang*[*]

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

Abstract

Neural networks are usually considered as powerful tools for extracting rules from a set of data. In this paper, a novel rule-extraction algorithm based on the radial basis function (RBF) neural network classifier is proposed for representing the hidden concept of numerical data. The premises of each rule consist of the intervals of inputs. The premises are adjusted in order to obtain high accuracy in the extracted rules. Simulations using Iris data set demonstrate that our approach leads to more accurate and compact rules compared to other methods for extracting rules from RBF neural networks. Comparisons between our algorithm with rule-extraction algorithms based on MLP and decision trees are made.

1 INTRODUCTION

As an important aspect of data mining, extracting rules to represent the concept of data have been explored widely. The extracted rules help to not only describe the information of data but also predict future trends or build classification systems [4]. Many rule extraction techniques have been developed based on neural networks [11][19][21]. Despite the merits of neural networks for various applications, there still exist difficulties for determining the training parameters and efficient network architecture. 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.

Many rule-extraction algorithms are proposed based on multi-layer perceptron (MLP) neural networks. Currently there are three main schemes for concise rule extraction. The first is to preprocess data [3][6][25] before presenting to a neural network. In most cases, redundant or irrelative attributes exist in data sets and preprocessing aims at removing them before training a neural network. The second scheme is to prune redundant hidden neurons or weights [1][4][5][14][15][19][20][21][22][23] during training of a neural network. In the third scheme, rule selection is carried out after training a neural network and forming the initial rules. Genetic algorithm is often involved [11][12] to select concise rules without reducing accuracy of the rules.

RBF neural networks have been implemented in various pattern recognition problems, such as speech recognition, signal recognition, and function approximation [1][13][17][18]. Its kernel function is responsive to only a subset of patterns within the receptive field of the kernel function, which makes extracting rules from the RBF neural network easier compared to other types of neural networks. Real world applications of the RBF neural network in knowledge discovery have been reported.

The architecture of an RBF neural network is simple, and it can represent the global and local concept of data at the same time. Some research work has been carried out in extracting rules based on RBF neural networks. In [3][8], redundant inputs are removed before rule extraction. Huber [10] selects rules according to importance; however, the accuracy is reduced with pruning. McGarry [14][15] [16] 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.

In this paper, a novel method is proposed to extract rules from the RBF neural network. First, the architecture of the RBF classifier is simplified through a modification in training the RBF neural network (Section 2). The weights connecting hidden units with output units are simplified (pruned) subsequently. Then the interval for each input in the premise of each rule is adjusted in order to obtain a high rule accuracy (Section 3). We show that our method leads to a compact rule set with desirable accuracy (Section 4).

2 A SIMPLIFIED RBF CLASSIFIER

Typically, an RBF neural network has three layers, i.e., the input layer, the hidden layer with Gaussian activation functions, and the output layer. The activation of a hidden unit is determined by 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 determined by the linear least square (LLS) method [2][24], which is fast and free of local minima, in contrast to the multilayer perceptron neural network.

Besides the centers, widths and the weights connecting hidden nodes and the output nodes, the number of hidden units is an important parameter for determining an RBF neural network. Both the dimensionality and the distribution of the input patterns affect the number of the hidden units. It is desirable for an RBF classifier to have a small number of hidden units, and at the same time, a low classification error rate. We now discuss the effect of overlapped receptive fields of Gaussian kernel functions of the RBF neural network on the number of its hidden unit.

There are two kinds of overlaps, one is the overlap between different classes, the other is the overlap between clusters of the same class. Overlapped receptive fields of different clusters can improve the performance of the RBF classifier in rejecting noise when tackling with noisy data

[13]. However, when tackling with noise free data, the overlap between different classes will lead to lower accuracy in classification. The degree of overlaps between different classes is measured by the parameter θ, i.e., θ is the ratio between in-class and out-class patterns in a cluster. The other overlap is the overlap between clusters with the same class label. The overlap between clusters with the same class label can decrease the number of hidden units and reduce the effect of noise too.

By allowing for large overlaps between clusters for the same class, we can further reduce the number of clusters substantially. This will lead to more efficient construction of RBF networks, i.e., the number of hidden units will be reduced. The experimental results will be shown in Section 4.

3 A METHOD FOR EXTRACTING RULES

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.

First, we determine the corresponding output unit which each hidden unit serves for by simplifying the weights between hidden units and output units. Assume there are m hidden units and n output units. Consider the weight matrix , which is a $m * n$ array. The maximum value of each row j (j=1, ... m) of matrix W is selected, and other points in the row are set to be zero. Thus, the new W reflects the corresponding output which each hidden unit mainly serves for.

In the symbolic rules which represent the concept of data, the IF part of one rule is composed of the suitable interval of each attribute. Assume that the number of attributes is N. Then the procedure to obtain the If part of symbolic rules is as follows.

$center(j, i)$ is the jth item of the center of the ith kernel function, $W_K(i)$ is the width of the ith kernel function, and $W_I(i)$ is the width of the ith interval. RunM is the Maximum cycle number of the implementation. ErrorP is the pre-defined rule extraction error. ErrorR is the error rate of rules extracted.

```
DO{
1. Initialize:
set L=0, n=0, $\eta_i$=1, ErrorR=1, Error=ErrorR, Run=0, and Sign=1.
2. Calculate width:
set L=L+1, i=L, $W_I(i) = \eta_i * W_K(i)$.
3. Obtain the interval:
set n=n+1, j=n, and
$Upper(j, i) = center(j, i) + W_I(i)$;
$Lower(j, i) = center(j, i) - W_I(i)$;
4. If j<R, go to 3, else if i<m, n=0, go to 2.
5. Check the rule extraction error:
obtain the error of rule extraction, ErrorR, by using the obtained intervals.
```

6. Update parameters:

η_i=η_i+Sign*0.025, Run=Run+1, if Run >2 and if ErrorR>Error, Sign=-Sign, Error=ErrorR;

} While Run < RunM and ErrorR> ErrorP

Compared with the technique proposed by McGarry [14][15][16], we can obtain a higher accuracy with concise rules. In [14][16], the input intervals in rules are expressed in the following equations:

$$X_{upper} = \mu_i + \sigma_i - S \quad , \tag{1}$$

$$X_{lower} = \mu_i - \sigma_i + S \quad , \tag{2}$$

where S is feature "steepness", which was discovered empirically to be about 0.6 by McGarry. Obviously the empirical parameter may not be suitable to all data sets. The experimental results are shown in the next section.

The ith symbolic rule is written as follows:

```
IF the 1th input is within the interval (Lower(1,i),Upper(1,i))
   and the 2th input is within the interval (Lower(2,i),Upper(2,i))
   and
   .
   .
   .
   and the Nth input is within the interval (Lower(N,i),Upper(N,i))
THEN the class label is $k_i$.
```

4 EXPERIMENTS

Iris and Thyroid data sets are standard examples used for testing classification methods. They can be accessed from the UCI Repository of Machine Learning Databases. There are 4 attributes and 3 classes in Iris data set. The four input attributes of Iris data are: sepal length, sepal width, petal length, and petal width. The 3 classes are Setoda, Versicolor, and Virginica. There are 5 attributes and 3 classes in Thyroid data set. Each data set is divided into 3 parts, i.e., training, validation, and test sets. 150 patterns of Iris data set is divided into 50 patterns for each set. There are 215 patterns in Thyroid data set: 115 patterns are for training, 50 patterns for validation and 50 patterns for testing. We set $\alpha = 0.1$ and $\theta = 7$ in our experiments. The smallest number of hidden units in constructing an RBF neural network classifier is 3 for Iris data set. For Thyroid, at least 7 hidden units are needed.

When we use the modification in which large overlaps among clusters of the same class are permitted, the number of hidden units is decreased and the classification error rate is maintained or even decreased. For example, for Iris, the average classification error rate for the testing set is unchanged compared with the result without the modification, but the number of hidden units is reduced from 4.6 to 3.4 on average. For Thyroid, the error rate in classification even decreases after applying the modification, and the number of hidden units is reduced from 14.4 to 8 on average. All the results are averaged over 5 experiments.

After the learning procedure and using our proposed rule extraction method, we obtain 3 symbolic rules for Iris data set. The accuracy of the symbolic rules that we obtain through the proposed method is 90% for Iris data set. For Thyroid data set, 7 rules are obtained, with 5 conditions in each rule, and the accuracy is 80%.

We now compare our results with other rule-extraction results based on RBF neural networks. In [8], 5 or 6 rules are needed to represent the concept of Iris data (the accuracy is not available). Huber [10] extracted 8 rules to represent Iris data set (the accuracy is not available). In order to get a small rule base, unimportant rules are pruned according ranking [10]. However, the accuracy of rules is reduced [10] at the same time. McGarry [14][15][16] extracted rules from RBF neural networks directly from the parameters of Gaussian kernel functions and weights. In [14], the accuracy reaches 100%, but the number of rules is large (for the Iris data set, 53 rules are needed). In [15] and [16], the number of rules for the Iris data set is small, i.e., 3, but the accuracy of the extracted rules is only 40% and around 80%, respectively. For Thyroid data set, we obtain 8 symbolic rules, and there are 5 conditions in each rules. The accuracy of extracted rules is 80%. The results of extracted rules for Thyroid data set using other methods are not available.

Much work has been carried out in extracting rules using the MLP. Good results have been obtained both in accuracy and numbers of rules (e.g., [9][11]). 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.

Decision tree, a non-neural network based approach, is popular in the rule extraction field. Decision-tree-building algorithms can represent rules themselves, in contrast to neural networks. However, the accuracy of prediction of decision-tree-building algorithms is usually lower than neural networks, and it is difficult for them to tackle with dynamic data. In [26], it indicates that the accuracy for Iris data set based on a decision-tree-building algorithm is around 97%, the number of rules is not available. In [7], the number of rules for Iris is 7. There are records showing that the rule extraction results based on decision tree for some data sets are better than neural networks.

5 CONCLUSIONS

In this paper, we proposed a useful modification to train the RBF network by allowing for large overlaps among clusters of the same class, which reduces the number of hidden units while maintaining the classification performance. Rule extraction is carried out from the simplified RBF classifier in order to explain and represent the concept of data in a concise way. The weights between the hidden layer and the output layer are simplified first. Then the interval for each input as the condition part of one rule is determined by iteration steps. Experimental results show that our rule extraction technique is simple to implement, and concise rules with high accuracy are obtained through the simplified RBF neural network classifiers.

References

[1] S. Adachi, T. Miyoshi and H. Ichigashi, "Extraction of underlying rules by projection pursuit learning networks with M-apoptosis structural learning algorithm", *Proc. IEEE Inter-*

national Joint on Computational Intelligence, vol. 1, pp. 651-656, 1998.

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

[3] 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.

[4] B. Chakraborty and G. Chakraborty, "Rule extraction from structured neural network for pattern classification", *Proc. IEEE International Conference on Systems, Man, and Cybernetics*, vol. 4, pp. 869-874, 1999.

[5] T. D. Gedeon, "Indicators of hidden neuron functionality: the weight matrix versus neuron behaviour", *Proc. Second New Zealand International Two-Stream Conference on Artificial Neural Networks and Expert Systems*, pp. 26-29, 1995.

[6] S. Geva, M. T. Wong and M. Orlowski, "Rule extraction from trained artificial neural network with functional dependency preprocessing ", *Proc. First International Conference on Knowledge-Based Intelligent Electronic Systems*, vol. 2, pp. 559-564, 1997.

[7] A. Gupta, Park Sang, S. M. Lam, "Generalized Analytic Rule Extraction for feedforward neural networks", *IEEE Transactions on Knowledge and Data Engineering*, vol. 11, No. 6, pp. 985 -991, Nov.-Dec. 1999.

[8] 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.

[9] 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.

[10] 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.

[11] H. Ishibuchi, M. Nii, and T. Murata, "Linguistic rule extraction from neural networks and genetic-algorithm-based rule selection", *Proc. International Conference on Neural Networks*, vol. 4, pp. 2390-2395, 1997.

[12] H. Ishibuchi and T. Murata, "Multi-objective genetic local search for minimizing the number of fuzzy rules for pattern classification problems", *Fuzzy Systems Proceedings, 1998. IEEE World Congress on Computational Intelligence*, vol. 2, pp. 1100-1105, 1998.

[13] 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.

[14] 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.

[15] 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.

[16] 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.

[17] 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.

[18] A. Roy, S. Govil, and R. Miranda, “A neural-network learning theory and a polynomial time RBF algorithm”, *IEEE Transactions on neural network*, vol. 8, no. 6, pp. 1301-1313, November, 1997.

[19] T. Saito and Y. Takefuji, “Logical rule extraction from data by maximum neural networks Intelligent Processing and Manufacturing of Materials”, *Proceedings of the Second International Conference on Intelligent Processing and Manufacturing of Materials*, vol. 2, pp. 723-728, 1999.

[20] R. Setiono and H. Liu, “Symbolic representation of neural networks”, Computer, Vol. 29, pp. 71-77, March. 1996.

[21] R. Setiono, ”Extracting rules from neural networks by pruning and hidden-unit splitting”, *Neural Computation*, vol. 9, no.1, pp. 205-225, Jan. 1997.

[22] R. Setiono and W. K. Leow, “Generating rules from trained network using fast pruning Neural Networks”, *Proc. International Joint Conference on Neural Networks*, vol. 6, pp. 4095-4098, 1999.

[23] R. Setiono, “Extracting M -of-N rules from trained neural networks”, *IEEE Transactions on Neural Networks*, vol. 11, no. 2, pp. 512-519, March, 2000.

[24] N. Sundararajan , P. Saratchandran, and Y. W. Lu , *Radial basis function neural networks with sequential learning: MRAN and its applications*, Singapore , River Edge, N.J.: World Scientific, 1999.

[25] H. Tsukimoto, “Extracting rules from trained neural networks”, *IEEE Transactions on Neural Networks*, vol. 11, no. 2, pp. 377-389, March, 2000.

[26] M. Umano, T. Okada, I. Hatono, and H. Tamura, “Extraction of quantified fuzzy rules from numerical data”, *The Ninth IEEE International Conference on Fuzzy Systems*, vol. 2, pp. 1062 -1067, 2000.