World Applied Sciences Journal 7 (Special Issue of Computer & IT): 54-58, 2009
ISSN 1818-4952

Rule Extraction from Dynamic Cell Structure Neural Networks by Modified LREX Algorithm-A Comparative Study

[1]Mansour Sheikhan and [2]Amir Khalili

[1]Department of Electrical Engineering, Islamic Azad University, South Tehran Branch, Tehran, Iran
[2]Department of Computer Engineering, Islamic Azad University, South Tehran Branch, Tehran, Iran

Abstract: The semantic of neural networks is not explicit and they are considered as black box systems. There are many researches investigating the area of rule extraction by neural networks. In this paper, the eclectic approach of rule extraction from a dynamic cell structure (DCS) neural network is investigated. To do this, a modified version of LERX algorithm is used for rule generation. Empirical results show that the DCS performs better than other self-organizing maps, e.g. Kohonen and competitive neural networks in generation of effective rules. Accuracy of classification, separability of Voronoi regions and mean squared error (MSE) in regions are the measures in which DCS performs better than other networks.

Key words: Rule extraction . neural networks . dynamic cell structure

INTRODUCTION

In machine learning and data mining research, rule learning has become an increasingly important topic. Although artificial neural networks (ANNs) received the attention of research groups, because of their adaptivity and ability to learn, lack of explanation capability is an important drawback of them [1]. In other words, the semantic of a neural network (NN) is not explicit and it is considered as a black box system [2, 3].

Knowledge embedded within ANNs is distributed over the activations and connections of neurons [4] and is not transparent to users [5]. There are many researches investigating the area of rule extraction by ANNs [6-15].

The researches on rule extraction can be classified into three approaches: decompositional, pedagogical and eclectic. Analyzing the activation and weights of the hidden layers of ANN is performed in decompositional approach [16-19]. The pedagogical approach treats the ANN as a black box and extract rules by only looking at the input and output activations [1, 20]. The number of these rules and their forms do not directly correspond to the number of weights or the architecture of ANN [21]. Finally, the eclectic approach, which is based on two former approaches, is characterized by any use of knowledge concerning the internal architecture and/or weight vectors in a trained ANN to complement a symbolic learning algorithm [22]. In this paper, the third approach is investigated in which rules are extracted from a dynamic cell structure (DCS) neural network.

Most of the techniques developed thus far for rule extraction are very NN-specific. Two specific rule extraction techniques seemed closely related to this work. One technique, RULEX, was applied first to a constrained multilayer perceptron (MLP) [1] and then to a local-cluster NN [23]. Another technique, LREX, is used to extract rules from radial basis function (RBF) neural network [24, 25]. In this paper, a modified version of LREX is used for rule extraction from the DCS.

The rest of this paper is organized as follows. In Section 2, the foundation of DCS is described. The rule extraction algorithm is introduced in Section 3. Empirical results and conclusions are also drawn in Section 4 and Section 5, respectively.

DCS NEURAL NETWORK

The DCS neural network is known as a member of self-organizing maps (SOMs). This neural network, which is implemented in the GEN1 system by National Aeronautics Space Administration (NASA), was originally developed by Bruske and Sommer [26] and was a derivative of Fritzke's work [27] combined with competitive Hebbian learning by Martinez [28].

The DCS is designed as a topology representing network, whose role is to learn the topology of an input space with perfect preservation. This network learns the

Corresponding Author: Dr. Mansour Sheikhan, P.O. Box 11365/4435,
Post-Graduate Center, South Tehran Branch, Islamic Azad University, Iran

function that describes a map of the input space, represented as Voronoi regions.

The neurons within the NN represent the reference vector (centroid) for each of the Voronoi regions. The connection between the neurons, c_{ij}, is then part of the Delaunay triangulation connecting neighboring Voronoi regions through their reference vectors. This reference vector is known as the "best matching unit" (BMU). Given an input, X, the BMU is the neuron whose weights, W, are closest to X. Along with the BMU, the "second BMU" (SBU) is found to maintain the Delaunay triangulation and to adjust nearby neurons within the BMU neighborhood (NBR), defined as the neurons connected to the BMU through the triangulation.

The DCS neural networks consist of two learning rules, Hebbian and Kohonen. Hebbian learning updates c_{ij} between neurons i and j to reflect the topology of the input space:

$$C_{ij}(t+1)=\begin{cases}1 & ;(i=\text{BMU}) \wedge (j=\text{SBU})\\ 0 & ;(i=\text{BMU}) \wedge (C_{ij}<\theta) \wedge (j\in \text{NBR}\setminus\{\text{SBU}\})\\ \alpha C_{ij}(t) & ;(i=\text{BMU}) \wedge (C_{ij}\geq\theta) \wedge (j\in \text{NBR}\setminus\text{SBU}\})\\ C_{ij}(t) & ;(i,\ j\neq BMU)\end{cases} \quad (1)$$

The forgetting constant, α, is included to produce a weakening between i and j, if they are not currently the closest to the stimulus and θ is the edge threshold, a minimum acceptable connection strength in order for the connection to be considered valid. Kohonen learning is used to adjust the weight vectors, W, of the neurons:

$$\Delta\vec{w}_i=\begin{cases}e_{BMU}(\vec{X}-\vec{w}(t)) & ;\ (i=\text{BMU})\\ e_{NBR}(\vec{X}-\vec{w}(t)) & ;\ (i\in\text{NBR})\\ 0 & ;\ (i\neq\text{BMU}) \wedge (i\notin\text{NBR})\end{cases} \quad (2)$$

where e_{BMU} is the BMU weight adjustment parameter and e_{NBR} is the weight adjustment applied to the neighborhood of the BMU. These two learning rules allow the DCS to change its structure. The ability to add new neurons into the network, as it grows, gives the DCS the potential to evolve into many different configurations.

RULE EXTRACTION ALGORITHM

As mentioned in the introduction, a modification of the LERX algorithm by McGarry *et al*. [24, 25] is used in this work for extracting rules from the DCS. This algorithm was originally used to extract rules from RBF neural network [29].

Input:
Weights of the DCS (centers of Voronoi regions)
Best matching unit for each input
Output:
One rule for each cell of the DCS
Procedure:
Train DCS on the data set
Record BMU for each input
Collect all inputs with common BMU to form cell
For each weight (w_i)
  For each independent variable
    $x_{lower} = \min\{x \mid x \text{ has BMU} = w_i\}$
    $x_{upper} = \max\{x \mid x \text{ has BMU} = w_i\}$
Build rule by:
  Independent variable in $[x_{lower}, x_{upper}]$
  Join antecedent statements with AND
  Dependent variable = category
    OR
  Dependent variable in $[y_{lower}, y_{upper}]$
  Join conclusion statements with AND
Write Rule

Fig. 1: Rule extraction algorithm

After training the network, the weights of DCS are used as inputs to the algorithm. The BMU corresponding to each data point is recorded during training and is used as an input to the algorithm, too. The training data is divided into regions based on the BMU. Then for each region, x_{lower} is the smallest value of the independent variable that has a particular BMU and x_{upper} is the largest value of that independent variable that has the same BMU. These two numbers form bounds for the intervals in the antecedent statement (e.g. ("variable>= x_{lower}" AND "variable<= x_{upper}")). An interval is determined for each of the independent variables and the statements are connected by "AND" to form the full antecedent. The algorithm of rule extraction is shown in Fig. 1.

EMPIRICAL RESULTS

In this study, the classification of Iris data [30] by the DCS is investigated as a benchmark application. The performance of DCS in terms of "accuracy of responses", "number of regions and number of data points in each region", "seperability of regions" and "mean squared error (MSE)" is compared with the performance of two other networks in the family of SOM neural networks. These networks are competitive NN (C-NN) and Kohonen NN (K-NN).

80% of Iris data (120 samples) are used for training the networks. Test dataset includes 20% of Iris data (30 samples). The accuracy of classification is evaluated by comparing the output of neural network module and rule-extraction module (Fig. 2).

The agreement of responses for three mentioned NNs with rule-extraction module is reported in Table 1.

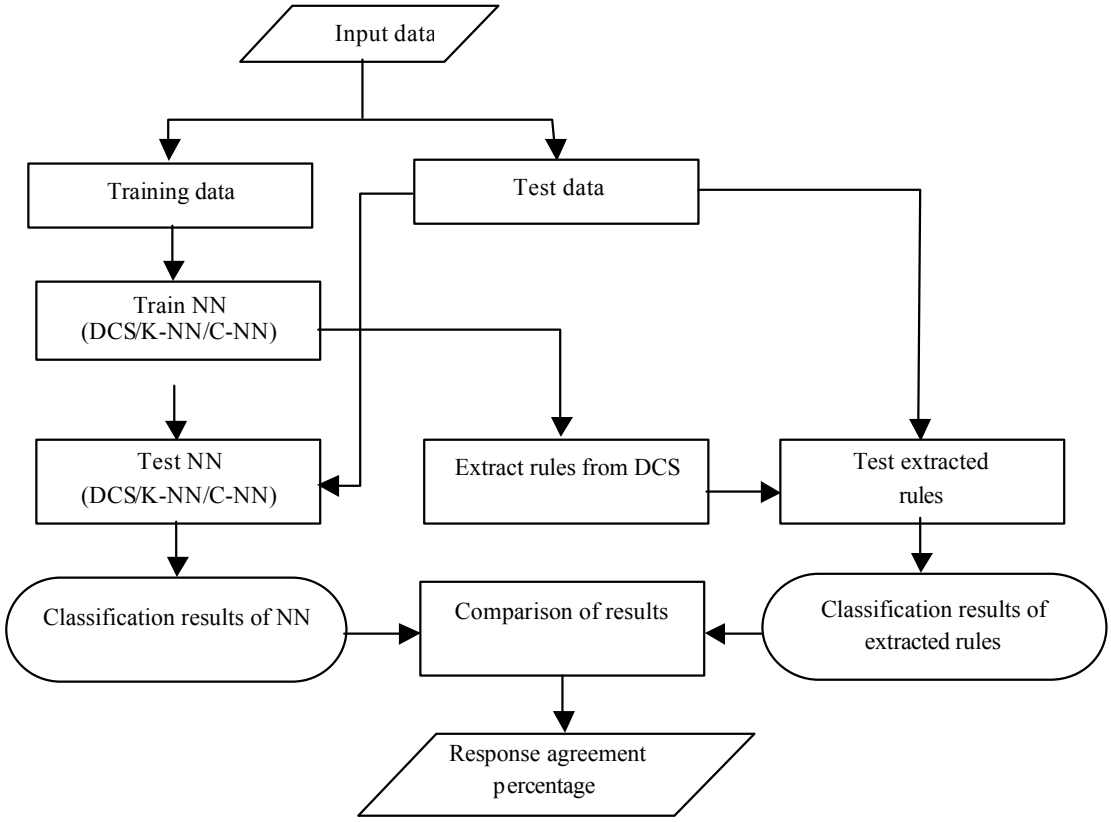

Fig. 2: Performance comparison procedure of NN and rule extraction modules

Table 1: Response agreement of ANN module with rule-extraction module

ANN	Agreement (%)
DCS	93.33
K-NN	85.33
C-NN	70.06

Table 2: Distribution of input data in each region-DCS neural network

Region	Distribution (%)
1	1.33
2	33.33
3	0.67
4	6.67
5	10.67
6	0.67
7	46.67

Table 3: Distribution of input data in each region-Kohonen neural network

Region	Distribution (%)
1	16.00
2	17.33
3	4.00
4	0.67
5	8.00
6	7.33
7	7.33
8	4.00
9	8.67
10	4.00
11	6.67
12	6.00
13	2.67
14	7.33

The "number of BMUs (or regions)" and "number of input data in each region" are important indicators that show the effectiveness of algorithms. In this study, a lower threshold is assumed that shows the validity of a region and its extracted rules. In our work, this threshold is set 1% of the dataset size.

The distributions of data in each region, for each of three mentioned NNs, are reported in Table 2-4, respectively. By comparing the results, we conclude that C-NN presents better uniformity in distribution. So, C-NN is more successful than DCS and K-NN from this viewpoint.

Table 4: Distribution of input data in each region-Competitive neural network

Region	Distribution (%)
1	13.33
2	11.33
3	4.00
4	14.00
5	16.67
6	16.67
7	19.33
8	4.67

Table 5: Overlap between Voronoi regions in DCS, K-NN and C-NN

ANN	Overlapped regions	Overlap percentage
DCS	1,3	22
K-NN	3,5	27
	3,6	15
	5,9	33
	5,10	24
	7,11	24
	9,13	14
	12,14	23
C-NN	1,2	31
	1,6	30
	4,7	24

Table 6: Regional MSE of DCS, K-NN and C-NN

ANN	Regional MSE
DCS	4.0065
K-NN	5.9236
C-NN	10.5233

The "separability of regions" is another measure in rule extraction efficiency. In other words, the minimum overlap between regions is desirable. The percents of overlap between regions are reported for each of three mentioned NNs in Table 5.

As shown in Table 5, the DCS has the minimum overlap between regions and performs the best in this measure.

If the data points in each region have the minimum distance with BMU, then the classification will be more accurate. To evaluate the effectiveness of ANNs from this viewpoint, MSE is calculated for each network using Equation (3):

$$MSE = \frac{\sum_{k=1}^{M}\sum_{i=1}^{N} d^2(m_{ki}, BMU_k)}{M} \quad (3)$$

in which k and i are the indices of region and data point, respectively. M and N represent the number of regions and the number of data points in a region, respectively. It is noticeable that the DCS has the lowest MSE and performs better than K-NN and C-NN (Table 6).

CONCLUSIONS

In this paper, the performance of rule extracting module from a dynamic cell structure neural network was investigated in data classification applications and compared with competitive and Kohonen neural networks. In this way, a modified version of LERX algorithm was used for rule extraction from the DCS. To evaluate the effectiveness of generated rules, a procedure was introduced in Fig. 2 and the experiments showed that the classification accuracy of DCS is higher than two other networks. Although the distribution of data points in Voronoi regions was more uniform in competitive network, DCS presented more regional seperability and MSE. So, the DCS performed better than Kohonen and competitive neural networks in rule extraction.

REFERENCES

1. Andrews, R., J. Diederich and A.B. Tickle, 1995. A Survey and Critique of Techniques for Extracting Rules from Trained Artificial Neural Networks. Knowledge-Based Systems, 8: 373-389.
2. Behloul, F., B.P.F. Lelieveldt, A. Boudraa and J.H.C. Reiber, 2002. Optimal Design of Radial Basis Function Neural Networks for Fuzzy-Rule Extraction in High Dimensional Data. Pattern Recognition, 35: 659-675.
3. Mantas, C.J., J.M. Puche and J.M. Mantas, 2006. Extraction of Similarity Based Fuzzy Rules from Artificial Neural Networks. International Journal of Approximate Reasoning, 43: 202-221.
4. Bologna, G., 2004. Is the Worth Generating Rules from Neural Network Ensembles? Journal of Applied Logic, 2: 325-348.
5. Huang, S.H. and H. Xing, 2002. Extract Intelligeble and Concise Fuzzy Rules from Neural Networks. Fuzzy Sets and Systems, 132: 233-243.
6. Towell, G. and J. Shavlik, 1993. The Extraction of Refined Rules from Knowledge Based Neural Networks. Machine Learning, 13: 71-101.
7. Omlin, C.W. and C.L. Giles, 1996. Extraction of Rules from Discrete-Time Recurrent Neural Networks. Neural Networks, 9: 41-52.
8. Leng, G., T.M. McGinnity and G. Prasad, 2005. An Approach for On-Line Extraction of Fuzzy Rules Using a Self-Organising Fuzzy Neural Network. Fuzzy Sets and Systems, 150: 211-243.

9. Hruschka, E.R. and N.F.F. Ebecken, 2006. Extracting Rules from Multilayer Perceptrons in Classification Problems: A Clustering-Based Approach. Neurocomputing, 70: 384-397.
10. Odajima, K., Y. Hayashi, G. Tianxia and R. Setiono, 2008. Greedy Rule Generation from Discrete Data and Its Use in Neural Network Rule Extraction. Neural Networks, 21: 1020-1028.
11. Zárate, L.E., S.M. Dias and M.A.J. Song, 2008. FCANN: A New Approach for Extraction and Representation of Knowledge from ANN Trained via Formal Concept Analysis. Neurocomputing, 71: 2670-2684.
12. Setiono, R., B. Baesens and C. Mues, 2009. A Note on Knowledge Discovery Using Neural Networks and Its Application to Credit Card Screening. European Journal of Operational Research, 192: 326-332.
13. Kolman, E. and M. Margaliot, 2009. Extracting Symbolic Knowledge from Recurrent Neural Networks-A Fuzzy Logic Approach. Fuzzy Sets and Systems, 160: 145-161.
14. Kahramanli, H. and N. Allahverdi, 2009. Rule Extraction from Trained Adaptive Neural Networks Using Artificial Immune Systems. Expert Systems with Applications, 36: 1513-1522.
15. Yu, S., X. Guo, K. Zhu and J. Du, 2009. A Neuro-Fuzzy-GA-BP Method of Seismic Reservoir Fuzzy Rules Extraction. Expert Systems with Applications. In Press (Available Online 5 July 2009).
16. Hayashi, Y., 1991. A Neural Expert System with Automated Extraction of Fuzzy if-then Rules. In: Advances in Neural Information Processing Systems, 3: 1263-1268.
17. Giles, C.L. and C.W. Omlin, 1993. Extraction, Insertion and Refinement of Symbolic Rules in Dynamically Driven Recurrent Networks. Connection Science, 5: 307-328.
18. Fu, L.M., 1994. Rule Generation from Neural Networks. IEEE Transactions on System, Man and Cybernetics, 28: 1114-1124.
19. Optiz, D.W. and J.W. Shavlik, 1995. Dynamically Adding Symbolically Meaningful Nodes to Knowledge-Based Neural Networks. Knowledge-Based Systems, 8: 301-311.
20. Saito, K. and P. Nakano, 1988. Medical Diagnosis Expert System Based on PDP Model. In the Proceedings of the International Conference on Neural Networks, 1: 255-262.
21. Saad, E.W. and D.C.H. Wunsch, 2007. Neural Network Explanation Using Inversion. Neural Networks, 20: 78-93.
22. Keedwell, E., A. Narayanan and D. Savic, 2000. Creating Rules from Trained Neural Networks Using Genetic Algorithms. International Journal of Computers, Systeming Signals, 1: 30-42.
23. Andrews, R. and S Geva, 2002. Rule Extraction from Local Cluster Neural Nets. Neurocomputing, 47: 1-20.
24. McGarry, K., J. Tait, S. Wermter and J. McIntyre, 1999. Rule-Extraction from Radial Basis Function Networks. In the Proceedings of the International Conference on Artificial Neural Networks, 1: 613-618.
25. McGarry, K., S. Wermter and J. McIntyre, 2001. The Extraction and Comparison of Knowledge from Local Function Networks. International Journal of Computational Intelligence and Applications, 1: 369-382.
26. Bruske, J. and G. Sommer, 1994. Dynamic Cell Structures. In the Proceedings of Neural Information Processing Systems, pp: 497-504.
27. Fritzke, B., 1994. Growing Cell-Structures-A Self-Organizing Network for Unsupervised and Supervised Learning. Neural Networks, 7: 1441-1460.
28. Martinez, T.M., 1993. Competitive Hebbian Learning Rule Forms Perfectly Topology Preserving Maps. In the Proceedings of the International Conference on Artificial Neural Networks, pp: 427-434.
29. Darrah, M., B. Taylor and M. Webb, 2005. A Geometric Rule Extraction Approach Used for Verification and Validation of a Safety Critical Application. In the Proceedings of 18th Annual Florida Artificial Intelligence Research Society Conference, 3: 624-627.
30. http://en.wikipedia.org/wiki/Iris_flower_data_set# References.