02- Days, International Conference on Recent Trends in Engineering Science, Humanities and Management
Sri S Ramasamy Naidu Memorial College, Sattur, Tamil Nadu, India (RTESHM-17)
02nd-03rd February 2017, www.conferenceworld.in ISBN: 978-93-86171-18-4

MODIFIED BACKPROPAGATION AND RULE EXTRACTION FROM THE HIDDEN LAYER FOR THE CLASSIFICATION PROBLEM

R.Sripathy[1],B.Srinivasa Ragavan[2]

[1]Asst. Professor in MCA, [2]Asst. Professor in Computer Science,
Sri SRNM College,Sattur, Tamilnadu,(India)

ABSTRACT

Rule Extraction is the process of obtaining rules from the output of the output layer by applying normal Backpropagation algorithm. By this way of extracting rules, it consumes lot of time for training the network. Instead of using the normal Backpropagation in training, the proposed system uses modified Backpropagation algorithm in which the new error term is added in cost function of the normal Backpropagation algorithm. And also the rules are extracted based on the results of the hidden layer outputs. This type of extraction of rules gives efficiency in rule extraction algorithm and also consumes less time for training the network.

KeyWords : Modified Backpropagation; Rule Extraction.

I. INTRODUCTION

Backpropagation method is commonly used for the classification problem. A simple network consists of an input layer, output layer and hidden layer. The network internal representation is difficult for a researcher to understand what the trained network learned. This system is used to extract the human readable rules from the trained network so the researcher must be confident about its classifications.

There are three main approaches that have been taken in past work on rule extraction from neural networks. They are pedagogical, decompositional and eclectic . In pedagogical method learning activities are constructed so that network discover and build knowledge for them and develop largely on their own an understanding of concepts, principles and relationships. They often do this by wrestling with questions, and/or solving problems by exploring the environment, and/or physically manipulating objects. Decompositional method first extract the rules that explains the hidden unit activation and outputs and then extract the rules based on the output and these rules are combined to form input-output rules. Eclectic method is derived from the above two methods. The number of rules generated based on the number of hidden neurons.

Rules are extracted by examining the discretized activation values of the hidden unit. The algorithm first discretizes the activation values of hidden nodes into a manageable number of discrete values without sacrificing the classification accuracy of the network. A small set of the discrete activation values make it possible to determine both the dependency among the output values and the hidden node values and the

ISBN: 978-93-86171-18-4

dependency among hidden node activation values and input values. From the dependencies rules can be generated.

The aim of this work is to search for simple rules with high predictive accuracy. The basic idea of the proposed algorithm is this: using first order information in the data to determine shortest sufficient conditions in a pattern (i.e. the rule under consideration) that can differentiate the pattern from patterns of other classes and prune redundant rules. The sole use of first order information avoids the combinatorial complexity in computation, although it is well that using higher order information may provide better results.

II. RELATED WORK

There are a number of approaches that have been proposed in the past for extracting the rules from neural network. Andrews, Tickle and Diederich , 2001 have surveyed that, without some form of explanation capability, the full potential of trained artificial neural networks (ANNs) may not be realised. This survey gives an overview of techniques developed to redress this situation. Specifically, the survey focuses on mechanisms, procedures, and algorithms designed to insert knowledge into ANNs (knowledge initialisation), extract rules from trained ANNs (rule extraction), and utilize ANNs to refine existing rule bases (rule refinement). The survey also introduces a new taxonomy for classifying the various techniques, discusses their modus *operandi*, and delineates criteria for evaluating their efficacy. Bader, Holldobler and Mayer-Eichberger, 2007 have presented a new decompositional approach for the extraction of propositional rules from feed-forward neural networks of binary threshold units. After decomposing the network into single units, it show how to extract rules describing a unit's behavior. This is done using a suitable search tree which allows the pruning of the search space. Furthermore, it present some experimental results, showing a good average runtime behavior of the approach. Huysmans , Setiono , Baesens and Vanthienen ,2008 have shown that artificial neural networks and support vector machines often have superior performance when compared to more traditional machine learning techniques. The main resistance against these newer techniques is based on their lack of interpretability: it is difficult for the human analyst to understand the reasoning behind these models' decisions. Various rule extraction (RE) techniques have been proposed to overcome this opacity restriction. These techniques are able to represent the behavior of the complex model with a set of easily understandable rules. However, most of the existing RE techniques can only be applied under limited circumstances, e.g., they assume that all inputs are categorical or can only be applied if the black-box model is a neural network. The main advantage of Minerva is its ability to extract a set of rules from any type of black-box model.

III. ALGORITHM

The classification system uses single hidden layer Neural Network for training. In this system x_i^p the i^{th} input unit of the p^{th} pattern, w_{ji} be the weight connecting i^{th} input unit to the j^{th} hidden unit, v_{kj} be the weight connecting the j^{th} hidden unit to the k^{th} output unit, the system uses the activation function $\frac{1}{1+e^{-x}}$, t_k^p is the target output for the k^{th} output unit of the p^{th} pattern. Let N be the number of input patterns. Output of the j^{th} hidden unit for the p^{th} input pattern computed using

02nd-03rd February 2017, www.conferenceworld.in ISBN: 978-93-86171-18-4

$$a_{H_j}^p = \sigma(\varepsilon_{i\epsilon\, input}\ w_{ji}\, x_i^p) \quad \text{---------- (1)}$$

Where σ is the activation function. The output of the k^{th} output layer unit for p^{th} pattern is

$$a_{Ok}^p = \sigma(\varepsilon_{j\epsilon hidden}\, v_{kj} a_{Hj}^p) \quad \text{---------- (2)}$$

The error function of the network is computed using the formula

$$E_1 = \frac{1}{2}\varepsilon_{p=1}^{N}\varepsilon_{k\epsilon output}(t_k^p - a_{Ok}^p)^2 \quad \text{--------- (3)}$$

The new error function proposed by Q.Huynh and A. Reggia in their modified Backpropagation algorithm is considered for new weight updation. They have used a penalty term E_2 to decrease the hidden unit activation vector which is defined as

$$E_2 = -\frac{1}{2}\varepsilon_{p=1}^{N}\varepsilon_{q=1}^{N}\varepsilon_{k\epsilon hidden}(a_{Hk}^p - a_{Hk}^q)^2 \quad \text{---------- (4)}$$

A new cost function is

$$E = \alpha\ E_1 + \beta\ E_2$$

Where $\alpha, \beta > 0, \alpha + \beta = 1.$

Weight updation formula is

$$\frac{\partial E}{\partial w_{ji}} = \alpha\left(\frac{\partial\ E_1}{\partial w_{ji}}\right) + \beta\left(\frac{\partial\ E_2}{\partial w_{ji}}\right)$$

$$\frac{\partial E}{\partial v_{kj}} = \alpha\left(\frac{\partial E_1}{\partial v_{kj}}\right) + \beta\left(\frac{\partial E_2}{\partial v_{kj}}\right)$$

The partial differentiation of the penalty term E_2 is computed by

$$\frac{\partial E_2^p}{\partial w_{ji}} = -N\left(a_{Hj}^p - a_{Hj}\right)a_{Hj}^p(1 - a_{Hj}^p)x_i^p$$

Training Algorithm

1) Enter an input pattern for training.
2) Compute output of all hidden units using the formula (1).
3) Compute output for all output layer units using the formula (2).
4) Compute error of each output neuron.
5) Find new weight updation for all weight connections using formula (4).
6) Repeat the steps 1 to 5 for all input patterns.

ISBN: 978-93-86171-18-4

7) Compute error of the network.
8) Steps 1 to 7 to be repeated until desired accuracy is obtained.

The above training algorithm is executed to classify the problem.

Rule Extraction Algorithm:

1) Apply a pattern on the trained network.
2) Compute output of the hidden neurons for the trained network and find corresponding resultant class.
3) Set these values as L_j, j=1,2,….,number of hidden neurons, for the resultant class
4) Apply next pattern on the trained network.
5) Compute output of the hidden neuron for the trained network and find the corresponding resultant class
6) Corresponding to the resultant class if already L_j is found and U_j is not found then set this as U_j otherwise set this as L_j for the class.
7) If both L_j and U_j are already found for the resultant class then based on threshold the new h_j to be used in the range.
8) If new h_j is within the existing L_j and U_j then based on threshold the new h_j to be used in the range. If new h_j is within the existing L_j and U_j then we can ignore the case. If $h_j < L_j$ by the threshold then L_j is replaced with the new h_j. If $h_j > U_j$ by the threshold then U_j is replaced with the new h_j.
9) Repeat the steps 4 to 8 for all input patterns.

Rule

If $h_j \in [L_{jc} U_{jc}]$ for all j then the pattern is belonging to class "c".

On the trained work the pattern to be classified is to be sent and all h_j values to be computed then apply the rule to identify the class for the entered input pattern.

IV. RESULT AND DISCUSSION

The extraction of rule from output layer is a tedious process and also a time consuming one. By using the proposed system save the training time by adding new cost function to the normal backpropagation and also extracting the rule based on the hidden layer. The proposed work has been carried out on a Matlab with the Statlog (Heart) data set contains 13 attributes used to predict the presence or absence of heart disease. It consists of 270 instance. It does not consist of missing values. Attributes types are Real, Ordered, Binary, Nominal. The attributes contains information like age, sex, chest pain type (4 values), resting blood pressure, serum cholestoral in mg/dl, fasting blood sugar > 120 mg/dl, resting electrocardiographic results (values 0,1,2), maximum heart rate achieved, exercise induced angina, oldpeak = ST depression induced by exercise relative to rest, the slope of the peak exercise ST segment number of major vessels (0-3) colored by flourosopy, thal: 3 = normal; 6 = fixed defect; 7 = reversable defect. It consists of 2 classes (1, 0) and (0, 1). (1, 0) represents the Absence of the heart disease and (0, 1) represents the Presence of the heart disease. The datasets are input to the training algorithm. Among 270, 220 patterns are considered for training and 50 patterns are considered for

ISBN: 978-93-86171-18-4

testing. Training consisted of 400 epochs, and each input vector was presented to the network once during each epoch. Training automatically stops when the full number of epochs has occurred. The problem is simulated for 20 different times. The results obtained are shown in the Table I.

Results for the Heart Dataset:

Table I: Execution samples for Heart Dataset with the Learning Rate 0.0009.

Trail	Epoch	Time(in Seconds)	Accuracy (%)
1	5830	2.90	98
2	9666	4.90	92
3	8302	4.27	90
4	1859	1.01	88
5	2350	1.35	96
6	2057	1.16	92
7	3807	2.13	90
8	8107	4.30	89
9	3095	1.63	90
10	3161	1.64	100
11	2559	1.34	82
12	1851	0.99	93
13	2577	1.45	82
14	2986	1.47	86
15	5410	2.71	100
16	2470	1.39	99
17	9066	4.61	70
18	3595	1.85	100
19	7254	3.99	100
20	3999	2.11	98

For 20 Execution with learning rate of 0.0009

Average Number of Epochs : 4500

Average Time(s) : 2.361

Average Accuracy(%) : 91.75

The simulations are presented with a graph (MSE Vs Number of Epochs) for the Heart Data with learning rate of 0.0009 is shown in the figure 1. Table II shows the results obtained for a trial.

Table II: Results for single execution

Epoch	MSE
500	0.250
1000	0.201
1500	0.190
2000	0.167
2500	0.138
3000	0.062
3500	0.045
4000	0.039
4500	0.033
5000	0.021
5500	0.02
6000	0.017
6500	0.015

ISBN: 978-93-86171-18-4

7000	0.012
7500	0.011
8000	0.010
8500	0.010

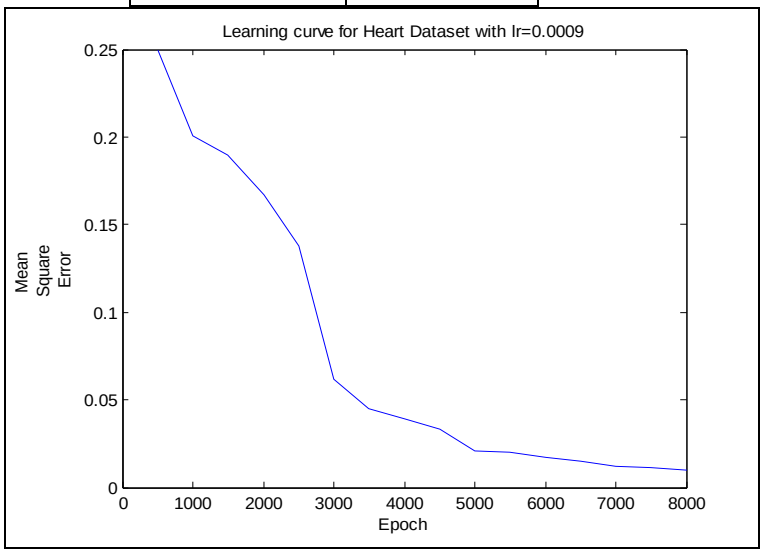

Figure 1 Learning curve for Heart dataset with lr=0.0009

	Number Of Rules			Rule Accuracy	
Dataset	E1	E1+E2	Reduced	E1	E1+E2
Statlog (heart data set)	70.12	14.30	80%	85.05%	85.19%
Waveform database generator	90.17	51.37	43%	51.55%	51.40%
Dermatology Database.	38.34	32.02	16%	89.29%	89.29%
Wine recognition Data	90.21	41.85	78%	89.33%	89.33%

V. CONCLUSION

The proposed system gives efficient way to extract rules based on the hidden layer for classification. The number of rules extracted depends on the number of hidden layers presented in the network. In this work, rule extraction from feedforward network is improved by adding additional terms to the cost function. This new term encourages the formation of more separable internal representation at the hidden layer. Rule sets extracted from networks trained with new error term are smaller than the regular error backpropagation.

ISBN: 978-93-86171-18-4

REFERENCES

[1] R. Andrews , A. Tickle and J. Diederich "Clinical Applications of Artificial Neural Networks", Cambridge Univ.Press, (2001) 256 - 297.

[2] S. Bader ,S. Holldobler and V.Mayer-Eichberger,"Extracting propositional rules from feed-forward neural networks a new decompositional approach", Proc. 3rd Int Workshop Neural-Symbolic Learn. Reason, 20071- 6.

[3] W. Duch , R. Adamczak and K. Grabczewski "A new methodology of extraction, optimization and application of crisp and fuzzy logical rules", IEEE Trans. Neural Netw*ork*, 12 (2001) 277 -306.

[4] R. Nayak "Generating rules with predicates, terms and variables from the pruned neural networks", Neural Network, 22 (2009) 405 - 414

[5] T. Etchells and P. Lisboa "Orthogonal search-based rule extraction (OSRE) for trained neural networks: A practical and efficient approach", IEEE Trans. Neural Network, 17 (2006) 374 -384.

[6] R. Setiono , B. Baesens and C. Mues "Recursive neural network rule extraction for data with mixed attributes", IEEE Trans. Neural Network, 19 (2008) 299 - 307

[7] J. Huysmans , R. Setiono , B. Baesens and J. Vanthienen "Minerva: Sequential covering for rule extraction", IEEE Trans. Syst., Man, Cybernetics, *Part B*: Cybernetics, 38 (2008) 299 - 309.