

ISSN 1996-3343

Asian Journal of
Applied
Sciences

Combining Quine Mc-Cluskey and Genetic Algorithms for Extracting Rules from Trained Neural Networks

D. Yedjour, H. Yedjour and A. Benyettou

S.I.M.P.A Laboratory, Department of Computer Science, Faculty of Science, University of Science and Technology of Oran, Mohamed Boudiaf USTO, P.O. Box 1505, El-Menaour, Oran, Algeria

Corresponding Author: Dounia Yedjour, S.I.M.P.A Laboratory, Department of Computer Science, Faculty of Science, University of Science and Technology of Oran, Mohamed Boudiaf USTO, P.O. Box 1505, El-Menaour, Oran, Algeria

ABSTRACT

Genetic algorithms are very efficient in the problems of exploration and seem to be able to find the single optimal solution in a huge space of possible solutions. However, they are ineffective when it comes to finding the exact value of the optimum in this space. This is precisely what the exact optimization algorithms perform best. It is therefore normal to think of combining exact and genetic algorithm to find the exact value of the optimum. Mc-RULEGEN is a new system to extract symbolic rules from a trained neural network, based on two approaches genetic and exact. MC-RULEGEN consists of three major components: A multi-layer perceptron neural component, a genetic component, a simplification rules component based on Quine McCluskey algorithm. Our method is tested on breast cancer and iris databases, the computational results have shown that the performances of the rules extracted by MC-RULEGEN are very high.

Key words: Neural networks, genetic algorithms, extracting rules, symbolic knowledge, Quine-McCluskey algorithm

INTRODUCTION

Neural networks demonstrate powerful problem solving ability, however it is well-known that neural networks usually represent their knowledge in the form of numeric weights and interconnecting which is not comprehensible for the user (Santos *et al.*, 2000). To overcome this restriction, various algorithms have been proposed to extract a meaningful description of the underlying 'black box' models. These algorithms' dual goal is to mimic the behavior of the black box as closely as possible while at the same time they have to ensure that the extracted description is maximally comprehensible (Huysmans *et al.*, 2006). Moreover, the extraction of accurate, comprehensible symbolic knowledge from networks is important if the results of neural learning are used in related problems (Towell and Shavlik, 1993). A system combining connectionist and symbolic reasoning models is known as an intelligent hybrid system (Fu, 1993; Craven and Shavlik, 1995). There are two popular approaches of extracting rules from multilayer networks, in one, we search for each node (hidden or output) the conditions that cause them to be active a set of rules is extracted to describe unit in terms of the units that have weighted connections to it, this method (called local or decompositional) uses the weight and threshold of nodes, examples of these systems are RuleNet (McMillan *et al.*, 1991), RULEX (Andrews and Geva, 1994), Subset (Towell, 1991), Full-RE (Taha and Ghosh, 1999) and Extract (Yedjour and Benyettou, 2003). In

the other method, a neural network is treated as a black box because rules are extracted regardless to the type or the structure of neural network. This involves exploring a space of candidate rules and testing individual candidates against the network to see if they are valid rules. This class is called global or pedagogical approaches. VIA (Validity Internal analysis) (Thrun, 1995), GEX (crisp Rule Extraction) and REX (fuzzy Rule Extraction) (Markowska-Kaczmar, 2008; Markowska-Kaczmar and Chumieja, 2004), MulGEX (Markowska-Kaczmar and Mularczyk, 2006), BIO-RE (Binary Input-Output Rule Extraction) (Taha and Ghosh, 1999) are examples of this approach. Metaheuristics algorithms are widely recognized as one of the most practical approaches for combinatorial optimization problems, this approach includes, among others, genetic algorithms (Mutsunori and Toshihide, 2000). Genetic algorithms are very efficient in the problems of exploration and seem to be able to find the single optimal solution in a huge space of possible solutions. However, they are ineffective when it comes to finding the exact value of the optimum in this space. This is precisely what the exact optimization algorithms perform best. It is therefore normal to think of combining exact and genetic algorithm to find the exact value of the optimum. We can easily apply at the end of the genetic algorithm the exact algorithm on the best element found. In this paper we describe a new method for combining genetic algorithm and exact algorithm of Quine Mc-cluskey for extracting binary if-then rules from artificial neural networks. Our system is tested on the database of breast cancer made at the University of California and on the iris database.

MC-RULEGEN SYSTEM

Mc-RULEGEN is a new system to extract symbolic rules from a trained neural network, based on two approaches genetic and exact. As shown in Fig. 1, the MC-RULEGEN consists of three major components: a Multi-Layer Perceptron neural component, a genetic component, a simplification rules component.

The training samples of binary inputs/outputs can be described through a suitable Boolean function f_{class} , which can be always written in another form, i.e., a logical sum of and Eq. 1.

$$f_{class} = \sum_{j=1}^{jclass} exp_j \quad \text{where, } exp_j = \prod_{i=1}^{imax} x_i \tag{1}$$

where, jclass is the number of the training samples satisfying fclass, j is the sample index, xi is the attribute value i of the sample j ($x_i = 1$ or 0), imax is the number of attributes and i is the attribute's index.

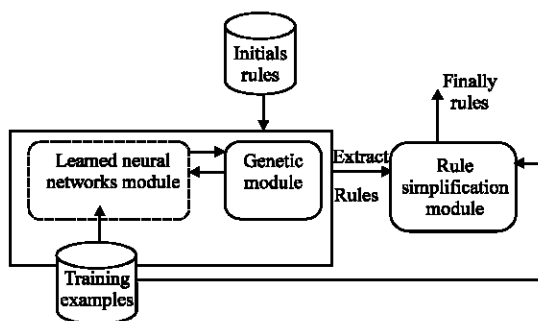


Fig. 1: MC-RULEGEN architecture

Each of these and operations (\exp_j) is called minterms in *Boolean algebra* and can be viewed as rule (R_j), where the attribute x_i represent the premisses and desired output represent the conclusion of rule. In genetic algorithm, rules must be represented by chains on which we can apply the genetic operators crossover and mutation. We decided to encode the rules as a string of integers. Each integer represents the value of an attribute. the chromosome codes a set of rules (Fig. 2).

Two genetic operators are used: crossover and mutations. the mutation consists just of randomly choosing a character in a string and change it. Optimal chromosomes, are allowed to breed and mix their datasets by genetic operators producing a new generation that will be even better. In this study, two fitness measures are used: fidelity and comprehensibility. Fidelity is calculated for each individual passed in the neural network for classification, the percentage of the good answers is the value of fidelity associated to the individual (Fig. 3). Comprehensibility is calculated for each rule as follows: (Markowska-Kaczmar and chumieja, 2004)

$$\text{Compr} = \frac{\text{No. of the input attributes} - \text{No. of the premisses in the rule}}{\text{No. of the input attributes} - 1} \tag{2}$$

If the value *compr* is close to 1, we say this rule has a high comprehensibility. In each generation, the fitness of every individual in the population is evaluated and the best individual moves to the next generation. After many generations this process should tend to produce individuals with increasing fitness.

We changed the standard backpropagation algorithm (developed by Rumelhart Hinton, Williams (Rumelhart *et al.*, 1986), so that inactive attributes (which are not involved in the rule, value =-1) are omitted in the calculation (Fig. 4). So, we gain in computational time.

MC CLUSKEY METHOD FOR SIMPLIFYING RULES

Theoretical background of Quine-Mccluskey algorithm: Quine-McCluskey Algorithm (QMC) is a method used for minimization of Boolean functions. This involves two main stage: (1) Find all

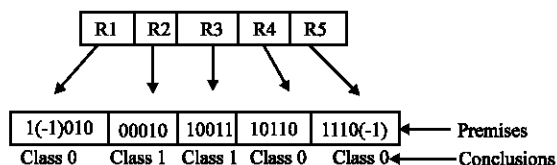


Fig. 2: Chromosome representation in MC-RULEGEN-1 means that the attribute is not involved in the rule 0 means that the attribute is written not (x) in the rule generated 1 means that the attribute is written as (x) in the rule generated

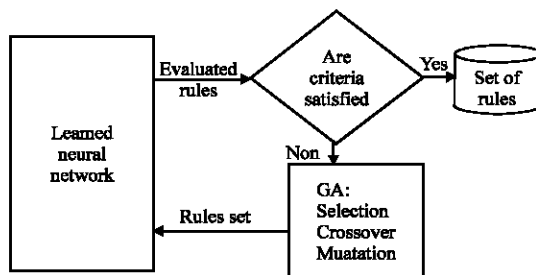


Fig. 3: Genetic module for generating rules

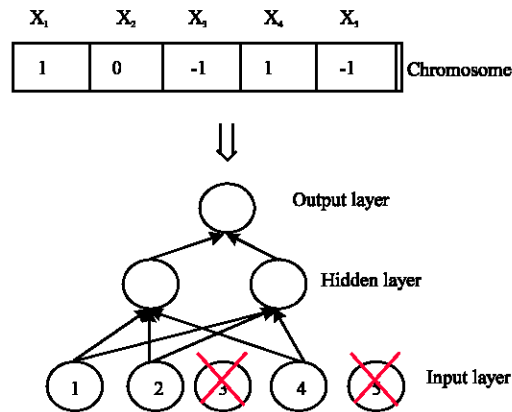


Fig. 4: A modified backpropagation algorithm

the prime implicants: The first step is to express the function as a sum of minterms, to group minterms by the number of 1's, to represent each minterm by its minterm number in binary form, finally Eliminate as many attributes as possible from each term by applying $xy + xy' = x$. The resulting terms are called prime implicants. if G is number of group then this stage ends after $(G-1)$ iterations. (2) Find all essential prime implicants: the table with the minterms in the top row and the reduced terms (the prime implicants) in the left column was built. The first step is to scan the columns for minterms that are covered by only a single unique prime implicant. The column is called distinguished column and the row in which the x occurs is called essential row. The second step is to determine the distinguished columns and essential rows (if any exists). The third step is to draw a line through each column which contains a x in any of the essential rows, since inclusion of the essential rows in the solution will guarantee that these columns contain at least one x . for the rest of the minterms that are yet to have a x , choose PIs as economically as possible to cover the remaining minterms. Finally, write out the final solutions as a set of PIs (McCluskey and Schorr, 1956).

The Modified Quine-Mc-Cluskey algorithm: The computational and spatial complexity of QMC algorithm is exponential and can quickly become unreasonable for complex formulas involving many variables. It is guaranteed to find a solution with the minimum number of terms. That's why it's it is preferable to used as a post-processing of the solution generated by the genetic algorithm. A set of rules generated by the genetic algorithm could be expressed as combinations (or strings) of ones and zeros and -1 of the corresponding inputs, an attribute in true-form is denoted by 1, in inverted-form by 0 and the absence attribute is represented by (-1). Unlike the method of QMC which works with minterms, our method can start by any term (Algorithm 1). This reduces the runtime complexity of the algorithm.

Algorithm 1: Modified QMC algorithm (stage 1)

For each class

1. Group all rules which have the same number of the absence attributes
2. Each group represents one iteration of the first stage of the QMC see in earlier (i.e., group which have rb absence attributes represents the iteration number rb)
3. Start with the first iteration (iter)

Algorithm 1: Continued

-
4. Group minterms by the numbers of 1's
 5. Applying $xy - xy' = x$
 6. Resulting combinations are add to the following iteration (iter-1)
 7. Jump to 4 until all possible combinations have been generated
-

Table 1: Relationship with algorithm QMC and rule simplification module

QMC algorithm	Rule simplification module
Variables	Attributes / premises
Function	Class / conclusion / output
Minterms	Training/test samples
Prime implicant	Simplified rules by stage 1 of modified QMC algorithm
Essential prime implicants	Final rules set obtained by stage 2 of modified QMC algorithm
-shows variable(s) dropped when adjacency is used	1 shows attribute(e) not involved in the rule

The second stage of QMC algorithm is modified as follows: each minterms is replaced by training samples and prime implicants by rules generated by earlier algorithm (Table 1). The idea underlying this step is to identify dominant rules. The Algorithm 2 shows how to obtain the final set of rules as small as possible in the whole population, rules covers the most of the training samples is chosen. The Table 1 shows the correspondence with QMC algorithm and rule simplification module.

Algorithm 2: Algorithm of optimisation of rules

-
1. for each rule R_{qi} , calculate the accuracy an comprehensibility using Eq. (2)
 2. Insert the best rule (better accuracy) in the final rules set,
 - if several rules exist, choose the rules which has a best Comprehensibility,
 3. If all samples are covered by the final rules set then stop,
 4. Otherwise choose rules as economically as possible to cover the remaining samples and insert them in the rules.
-

RESULTS

The were evaluated McRulegen in the context of 02 problem domains: breast cancer and Iris databases The results may be compared with those obtained by other methods.

Wisconsin breast cancer (Uci repository): This database contains 699 samples belonging to 02 classes (458 are benign and 241 are malignant). Each pattern is described by nine attributes; each one takes an ordinal integer value from 1 to 10. Half of each class is randomly selected for training and the remaining set is used for testing.

Iris basis (Uci repository): It consists of 150 iris divided in similar proportions in 03 classes: setosa, versicolor and virginica, each flower is described by 4 quantitative variables measuring the length and width of the sepal and the length and width of its petals, Table 2 shows the results for the data sets.

If the original attribute values are not binary, then a binarization Eq. 3 will be applied to non-binary data (Taha and Ghosh, 1999).

The parameters of the genetic algorithm play an important role on the network's convergence. Several experiments are conducted to achieve the optimum values. For example, for each value of the probability of mutation, we calculate the accuracy and comprehensibility of the rules extracted,

Table 2: Results of the network training

Data set	Attributes No.	Attributes type	Classes No.	Samples No.	Rules No.	Accuracy (%)	Comprehensibility (%)
Breast-cancer	9	Discrete	2	683	5	99.13	100.0
Iris	4	Continue	3	150	5	95.33	92.5

Table 3: Comparison of the classification by neural networks and the rules set

	Benin	Malignant	Accuracy test (%)	No. attributes
NN (Examples) MC-RULEGEN	229	120	98	9
(rules)	2	3	100	5

Table 4: Performance comparison between NN and rule sets

	Neural network		Extracted rules	
	Ratio	%	Ratio	%
Training	336/341	98.53	335/341	98.24
Testing	342/349	98.00	349/349	100.00
Overall	678/690	98.26	684/690	99.13

the average results obtained by 15 experiences after 500 generations. The following parameters are obtained in the same way: probability of mutation = 0.2, the probability of crossover = 0.8, population size = 30, the number of individuals = 15 rules.

$$y_i = \begin{cases} 1 & \text{if } x_i \geq u_i \\ 0 & \text{sin on} \end{cases} \quad (3)$$

where, x_i is the value of the attribute X_i , u_i is the average value of X_i and y_i is the corresponding binary value.

The initial population is created on the basis of the table of truth, containing all possible combinations of inputs. The method used in selection is the roulette wheel. The next generation is created from the current population by using crossover and mutation. For mutation, the gene has a high chance (2 / 3) to switch from the value 1 or 0 value to -1, it can also switch from -1 to 1 or 0 with a probability of 1/3. The chromosomes with a higher fitness value survive and participate in the creation of new populations. The population continuously evolves toward better fitness and the algorithm converges to the best chromosome after several generations. The modified algorithm of QMC allows a horizontal simplification (fewer rules) and vertical simplification (reduction of

attributes in the resulting rule), so the comprehensibility grows significantly. The finally phase of modified QMC, eliminate the useless rules and keeps only those rules that cover the majority of data sets.

The final set of rules is given by:

Breast cancer database:

- Accuracy = 99.13% comprehensibility = 100%
- If (v(8) Normal Nucleoli <2.77) then benign
- If (v(6) Bare Nuclei <3.45) then benign
- If (v(8) Normal Nucleoli =2.77) then malignant

Table 5: Comparison of Mc-RuleGen with GEX and full-RE

Breast cancer database	Full-RE	GEX	Mc-Rulgen
Training (accuracy %)	96.77	92.03	98.24
Test (%)	95.61	91.38	100.00

Table 6: Comparison of Mc-RuleGen with GEX and santos methods

Iris	GEX	Santos's method	Mc-Rulgen
Test (accuracy %)	92.48	93.33	95.33

Table 7: Correct classification (%) of the rule sets extracted by different techniques

Breast cancer database	Accuracy rate of the rule set	No. of rules	Maximum No. of premises per rule
MC-RULEGEN	100.00	05	01
Bio-Re	96.63	10+default rule	04
Partial-Re	96.49	09	3
Full-RE	96.19	05	2
NeuroRule	97.21	3+ default rule	04
C4.5rules	97.21	7	04

+Add a default rule to a rule sets

- If (v(6) Bare Nuclei =3.45) then malignant
- If (v(4) Marginal Adhesion = 3) then malignant

Iris database:

- Accuracy = 95.33% comprehensibility = 92.5%
- if (v(4) petal width <1.2) then setosa
- if (v(1) sepal length ≥ 5.84) and (v(2) sepal width <3.05) then virginica
- If (v(2) sepal width ≥ 3.05) and (v(3) petal length ≥3.76) then virginica
- If (v(4) petal width ≥1.2) then versicolor
- If (v(1) sepal length <5.84) and (v(2) sepal width <3.05) then versicolor

Compared to networks that use all of the attributes, the results shown in Table 3 indicate that the rules extracted by Mc-Rulgen uses a subset of attributes.

The accuracy is improved from 98 to 100% using only 03 attributes and only 05 rules.

Table 4 provides an overall comparison between the sets of extracted rules and their corresponding trained networks for the breast cancer problem.

DISCUSSION

Table 1 shows the performance of the extracted rules on the test and the training database, for different techniques (Table 5). The distribution of samples in each base plays an important role on the performance achieved by GEX (Table 6), it also significantly increases performance if the training method is the method of cross-validation (Markowska-Kaczmar and Chumieja, 2004).

Table 7 compares the classification rates obtained using the rules extracted by four techniques (Bio-Re, Partial-Re, Full-Re, Neurorule, C4.5 rules and GEX). The rules extracted by Mc-Rulegen are more comprehensible than those extracted by the other techniques. The results presented in this study indicate that MC-RULEGEN is able to extract a set of rules of better performance from trained networks.

CONCLUSION

This study presents a new approach of extracting rules from trained neural networks. The approach combines both metaheuristics (genetic algorithms) and exact algorithms (Quine-Mc-Cluskey) to extract the binary rules of the form if-then. During phase of the evaluation of rules by neural network, some attributes are omitted and therefore are not included in the calculation by the retropropagation. The rules obtained by the genetic module are simplified by the modified method of Quine-Mc-cluskey, which reduces the number of rules and premises in the rule and removes rules that do not add new information. At the end of this phase, the comprehensibility has grown significantly. The computational results have shown that the performance of the rules extracted by MC-RULEGEN is very high. An interesting extension could be the generalization of the MC-RULEGEN method for non-binary data, finally the current paper may lead to future work in data mining in general and in rule extraction, in particular these rules may be used as an initial theory for a similar problem.

ACKNOWLEDGMENTS

The authors would like to thank to UCI Machine Learning Repository for providing the data sets used to train and test the system.

REFERENCES

- Andrews, R. and S. Geva, 1994. Rule extraction from a constrained error back-propagation MLP. Proceedings of the 6th Australian Conference on Neural Networks, (ACCN'94), Brisbane Queensland, pp: 9-12.
- Craven, M.W. and J.W. Shavlik, 1995. Extracting comprehensible concept representations from trained neural networks. Proceedings of the Working Notes on the IJCAI'95 Workshop on Comprehensibility in Machine Learning, (WNIWCML'95), Montreal, Canada, pp: 61-75.
- Fu, L.M., 1993. Knowledge-based connectionism for revising domain theories. IEEE Trans. Syst. Man Cybernetics, 23: 173-182.
- Huysmans, J., B. Baesens and J. Vanthienen, 2006. Using rule extraction to improve the comprehensibility of predictive models. K.U. Leuven KBI, Res., 0612, 2006. <http://ideas.repec.org/p/ner/leuven/urnhdl123456789-121060.html>.
- Markowska-Kaczmar, U. and M. Chumieja, 2004. Discovering the mysteries of neural networks. Int. J. Hybrid Intell. Syst., 1: 153-163.
- Markowska-Kaczmar, U., 2008. Evolutionary approaches to rule extraction from neural networks. Studies Computational Intelligence, 82: 117-209.
- Markowska-Kaczmar, U., Mularczyk, K., 2006. GA-Based Pareto Optimization of Studies in Computational Intelligence. Springer, Berlin Heidelberg, New York.
- McCluskey, E.J. and H. Schorr, 1956. Minimization of boolean functions. Bell Syst. Tech. J., 35: 1417-1444.
- McMillan, C., M.C. Mozer and P. Smolensky, 1991. The connectionist scientist game: Rule extraction and refinement in a neural network. Proceedings of the 13th Annual Conference of the Cognitive Science Society, May 1991, Hillsdale, New Jersey, Erlbaum, pp: 424-430.
- Mutsunori, Y. and I. Toshihide, 2000. On metaheuristic algorithms for combinatorial optimization problems. IEICE Trans. Inform. Syst. Pt.1 (Japanese Edition), J83-D-1: 3-25.

- Rumelhart, D.E., G.E. Hinton and J.L. McClelland, 1986. A general framework for parallel distributed processing. *Parallel Distributed Process. Explorat. Microstructure Cognition*, 1: 45-76.
- Santos, R.T., J.C. Nievola and A.A. Freitas, 2000. Extracting comprehensible rules from neural networks via genetic algorithms. *Proceedings of the 2000 IEEE Symposium on Combination of Evolutionary Algorithm and Neural Network, (ISCEANN'00), Curitiba-PR, Brazil*, pp:130-139.
- Taha, T.A. and J. Ghosh, 1999. Symbolic interpretation of artificial neural networks. *IEEE Trans. Knowl. Data Eng.*, 11: 448-463.
- Thrun, S.B., 1995. *Advances in Neural Information Processing Systems*. MIT, SanMateo, CA, California,.
- Towell, G.G. and J.W. Shavlik, 1993. Extracting refined rules from knowledge-based neural networks. *Machine Learning*, 13: 71-101.
- Towell, G.G., 1991. *Symbolic knowledge and neural networks: Insertion, refinement and extraction*. Ph.D. Thesis, Computer Sciences Department, University of Wisconsin, Madison.
- Yedjour, D. and A. Benyettou, 2003. *Knowledge-Based Neural Networks for Speech Recognition*. University of Science and Technology Department of Informatics, SIMPA-lab, Oran, Algiers.