Journal of Applied Sciences

ISSN 1812-5654

ANSInet
an open access publisher
http://ansinet.com

Journal of Applied Sciences 11 (15): 2855-2860, 2011
ISSN 1812-5654 / DOI: 10.3923/jas.2011.2855.2860
© 2011 Asian Network for Scientific Information

Explaining Results of Artificial Neural Networks

D. Yedjour, H. Yedjour and A. Benyettou
S.IM.PA Laboratory, Department of Computer Science,
Faculty of Science, University of Science and Technology of Oran,
Mohammed Boudiaf USTO, P.O. Box 1505, El Menaour, Oran, Algeria

Abstract: Neural networks are very efficient in solving various problems but they have no ability of explaining their answers and presenting gathered knowledge in a comprehensible way. Two main approaches are used, namely the pedagogical one that treats a network as a black box and the local one that examines its structure. Because searching rules is similar to NP-hard problem it justifies an application of evolutionary algorithm to the rule extraction. Pedagogical approaches such as GA are insensitive to the number of units of neural networks as they see them as "black boxes" interested only their inputs and their outputs. In the study we describe new rule extraction method based on evolutionary algorithm called GenRGA. It uses logical rules and is composed of three (03) main parts: genetic module, neural networks module and rules simplification module. GenRGA is tested in experimental studies using different benchmark data sets from UCI repository. Comparisons with other methods show that the extracted rules are accurate and highly comprehensible.

Key words: Symbolic interpretation of artificial neural networks, extracting comprehensible rules, symbolic knowledge, quine-mccluskey algorithm

INTRODUCTION

Artificial Neural Networks (ANNs) are increasingly used in problem domains involving classification. They are adept at finding commonalities in a set of seemingly unrelated data and for this reason are used in a growing number of classification tasks. The, one of their major drawbacks is that their knowledge is not easily interpretable by a human, the main challenge to the use of supervised neural networks in data mining applications is to get explicit knowledge from these models. Generally speaking, there are two types of approaches to extracting rules from multi-layer networks: local and global. In the global methods (Markowska-Kaczmar, 2008; Markowska-Kaczmar and Mularczyk, 2006; Thrun, 1995), the neural network is treated as a black box, a set of global rules characterize the output classes directly in terms of the inputs (these), the decompositionnel or local approaches, methods go into details of the neural networks structure, describing each neuron separately in terms of rules, followed by concatenation of them (Kamruzzaman and Monirul Islam, 2010; Taha and Ghosh, 1999; Andrews and Geva, 1994; Towell and Shavlik, 1993; McMillan *et al*., 1991). Extracting rules from complex ANNs may therefore be intractable (Zhou *et al*., 2003). In this study we propose a novel, evolutionary global approach which integrates traditional ANNs with genetic algorithms for extracting simple, intelligible and useful rules from trained ANNs. The genetic algorithm uses chromosomes which can be mapped directly onto intelligible rules (phenotypes). In brief, the study proposes the use of a genetic algorithm to search the best rules for classification which represents the knowledge in the trained neural networks.

GenRGA SYSTEM

GenRGA (generation of rules by genetic algorithm) system is composed of three (03) main parts: genetic module, neural networks module and rules simplification module. The first stage of the explanation is to create a list of initial rules, The initial population of rules is chosen randomly from the truth table of n variables, where n is the number of attributes. An initial population consists of different types of genotypes, each one of them codifying the rules. This population is passed to neural network module for evaluation in order to determine the fitness of each individual. Afterwards, this group is made to evolve repeatedly by means of different genetic operators (selection, crossover, mutation) until a determined

Corresponding Author: Dounia Yedjour, S.IM.PA Laboratory, Department of Computer Science, Faculty of Science,
University of Science and Technology of Oran, Mohammed Boudiaf USTO,
P.O. Box 1505, El Menaour, Oran, Algeria

termination criteria is satisfied (for example, a sufficiently good individual is obtained, or that a predetermined maximum number of generations is reached). The obtained rules are simplified by module of simplifying the rules based on the exact method of QuineMccluskey algorithm. The general idea of has been presented in Fig. 1.

Genetic module: The extraction algorithm is based on the Pitt approach which means that every individual encodes a entire sets of rules.

Individual representation: The chromosome consists of a vector of rules. So, the length of the vector is equal to the number of rules (Fig. 2). The length of the chromosome is initially fixed and can be changed by user. The premises can take different values as follows:

- -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

Genetic operators: Two genetic operators are used: crossover and mutations. Crossover: it generates new chromosomes by exchanging the whole sets of membership functions for a randomly selected attribute of the parent chromosomes. The mutation consists just of randomly choosing a character in a string and change it . Mutation is more complicated, since it must allow modifying the contents of a given set by adding or removing premises (attributes) of rules and exclude them from the solution. The next generation is created from the current population by using crossover and mutation. The individuals from the current generation are chosen on the basis of a roulette wheel.

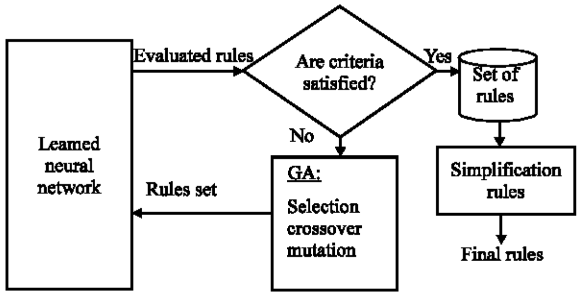

Fig. 1: GenRGA architecture

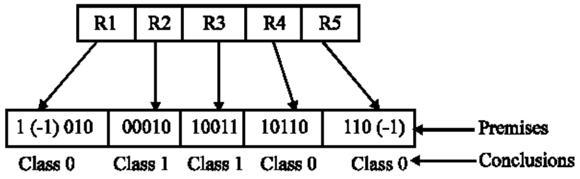

Fig. 2: Chromosome representation in GenRGA

Fitness function: 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. Comprehensibility is calculated for each rule as follows: (Markowska-Kaczmar and Chumieja, 2004):

$$\text{Compr} = \frac{\text{No.of the input attributes} - \text{No.of the premisses in the rule}}{\text{No.of the input attributes} - 1} \quad (1)$$

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.

```
AG Program
1. Select an initial population
2. Send population (pop) to ANN
3. Rank population according to fitness
4. Randomly select mating pairs from population according to fitness
5. Breed from mates with crossover and mutation
6. Goto (2)
    The algorithm is repeated until:
```

- a solution is found that satisfies a target (fidelity >μ_0 and compr> μ_1)
- a fixed number of generations has been produced
- the highest ranking solutions reach a plateau and there is no further improvement with repeated iteration
- $\mu 0$ and $\mu 1$ are predefined thresholds

Neural networks module: Neural networks are regarded commonly as black boxes but can be used to provide simple and accurate sets of logical rules. The neural network we used is the multilayer perceptron, During phase of the evaluation of rules, the attributes whose value is equal to -1 (not involved in the rule) are omitted and therefore are not included by retro-propagation algorithm. We changed the standard backprobagation 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. 3). So, we gain in computational time.

Simplification rules: The rules obtained by the genetic module are simplified by the method of Quine-Mc-cluskey

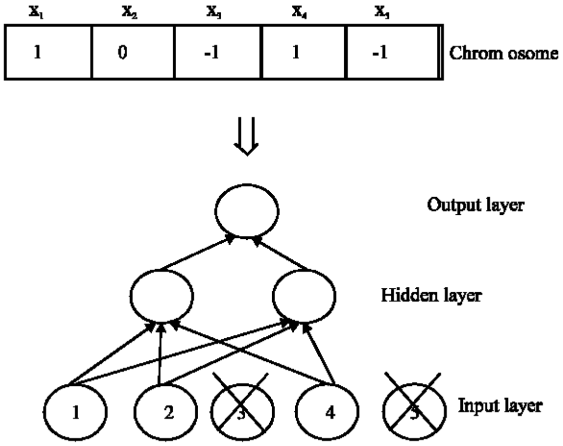

Fig. 3: A modified backpropagation algorithm

(McCluskey and Schorr, 1956) which reduces the number of rules and premises in the rule and removes rules that do not add new information (Yedjour *et al*., 2010).

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 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 (The detail of the QM algorithm can be found by McCluskey and Schorr (1956).

Table 1: Modified QMC algorithm (stage1)

For each class
Group all rules which have the same number of the absence attributes
Each group represents one iteration of the first stage of the QMC see in the section A.
(i.e., group which have *nb* absence attributes represents the iteration number *nb*)
Start with the first iteration (iter)
Group minterms by the number of 1's
Applying xy + xy' = x
Resulting combinations are add to the following iteration (iter+1)
Jump to 4 until all possible combinations have been generated

Table 2: Algorithm of optimisation of rules

- For each rule R_{Qi}, calculate the accuracy and comprehensibility using formulas (2),
- Insert the best rule (better accuracy) in the final rules set, if several rules exist, choose the rule which has a best comprehensibility,
- If all samples are covered by the final rules set then stop,
- Otherwise choose rules as economically as possible to cover the remaining samples and insert them in the final rules

Table 3: 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 modified Quine-Mccluskey 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 (Table 1). This reduces the runtime complexity of the algorithm.

The second stage of QMC algorithm is modified as follows: each minterms is replaced by training samples and prime implicants by rules generated by above algorithm (Table 3). The idea underlying this step is to identify dominant rules. Table 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. Table 3 shows the correspondence with QMC algorithm and rule simplification module.

EXPERIMENTS AND RESULTS

This section evaluates GenRGA in the context of 03 problem domains: breast cancer, Iris and diabetes databases. The results are 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 4 shows the results for the data sets.

Pima indian diabetes database[17]: This data set was obtained from the UCI Repository of Machine Learning Databases. The data set was selected from a larger data set held by the National Institutes of Diabetes and Digestive and Kidney Diseases. All patients in this database are Pima-Indian women at least 21 years old and living near Phoenix, Arizona, USA. The binary response variable takes the values '0' or '1', where '1' means a positive test for diabetes and '0' is a negative test for diabetes. There are 268 (34.9%) cases in class '1' and 500 (65.1%) cases in class '0'. There are eight clinical findings: 1. Number of times pregnant 2. Plasma glucose concentration a 2 hours in an oral glucose tolerance test 3. Diastolic blood pressure (mm Hg) 4. Triceps skin fold thickness (mm) 5. 2-Hour serum insulin (mu U/ml) 6. Body mass index 7. Diabetes pedigree function 8. Age (years).

If the original attribute values are not binary, then a binarization process (2) will be applied to non-binary data (Taha and Ghosh, 1999):

$$y_i = \begin{cases} 1\text{if } x_i \geq u_i \\ 0 \text{ sin on} \end{cases} \qquad (2)$$

where, x_i is the value of the attribute X_i, ui is the average value of X_i and yi is the corresponding binary value.

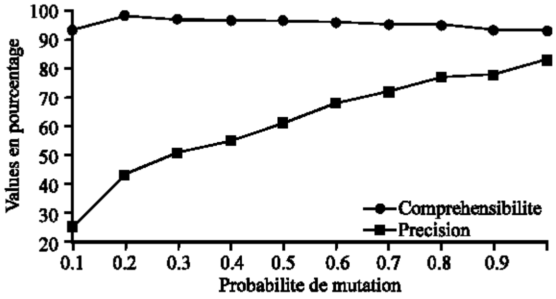

Fig. 4: Influence of parameter on the accuracy and comprehensibility of rules

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, Fig. 4 shows the average results obtained by fifteen 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.

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% comprehensibiliy = 100%
if (v(8) Normal Nucleoli <2.77) then benin
if (v(6) Bare Nuclei <3.45) then benin
if (v(8) Normal Nucleoli = 2.77) then malignant
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

Table 4: Performance comparison between Nn and Rule sets

Data sets	Attributes number	Attributes type	Classes number	Samples number	Rules number	Accuracy%	Comprehensibility%
Breast-cancer	9	Discrete	2	683	5	99.13	100
Iris	4	Continue	3	150	5	95,33	92,5
Diabete	8	continue	2	768	6	91,99	78,5

Table 5: Comparison of GenRGA with Gex and full-Re

Breast cancer database (50% Training 50% test)	Full-Re	Gex	Mc-Rulgen
Training (accuracy%)	96.77	92.03	98.24
Test (%)	95.61	91.38	100

Table 6: Comparison of GenRGA with Gex and Santos methods

Iris(70% Training 30% test)	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 set rule	No. of rules	Maximum No. of premises per rule
GenRGA	100	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
GEX	99,7	30	-

Table 8: Performance comparison of GenRGA with other algorithms for diabetes data

Diabetes	REANN	NN RULES	C4.5	NN-C4.5	OC1	CART	GenRGA
No. of rules	2	4	-	-	-	-	6
accuracy	76.56	76.32	70.9	76.4	72.4	72.4	91.99

Diabete database
Accuracy = 91,99% comprehensibility = 78,5%
if A2< 120.9 and A8 < 33.2 then testPositif
if A2< 120.9 and A4 > 20.5 and A7< 0.5 then testPositif
if A3 < 69.1 and A8 < 33.2 then testPositif
if A1 < 3.8 and A4 < 20.5 then testPositif
if A2> 120.9 and A6 > 32.0 then testNegatif
if A8 > 33.2 and A4 > 20.5 then testNegatif

Table 4 shows number of extracted rules and rules accuracy for three benchmark problems. In most of the cases GenRGA produces fewer rules with better accuracy. The aim of experiments was to test the quality of the rule extraction made by GenRGA for data sets with different types of attributes.

COMPARING WITH RELATED WORK

Table 5 and 6 show the performance of the extracted rules on the test and the training database for different techniques. The distribution of samples in each base plays an important role on the performance (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, C4.5rules and GEX). The rules extracted by GenRGA are more comprehensible than those extracted by the other techniques. The results presented in this paper indicate that GenRGA is able to extract a set of rules of better performance from trained networks.

Table 8 compares GenRGA results of diabetes data with those produced by REANN, NN RULES, C4.5, NN-C4.5, OC1 and CART algorithms. GenRGA achieved 91,% accuracy although REANN was closest second with 76.56% accuracy. Due to the high noise level, the diabetes problem is one of the most challenging problems in our experiments. GenRGA has outperformed all other algorithms.

CONCLUSION

A new global method of rule extraction from neural networks called GenRGA, has been presented in the study. The purpose of GenRGA is to produce a set of rules that describe a network's performance with the highest fidelity possible, taking into account its ability to generalize, in a comprehensible way. The approach combines both metaheuristics (genetic algorithms) and exact algorithms (Quine-Mc-cluskey) to extract the binary rules of the form if-then). The computational results have shown that the performance of the rules extracted by GenRGA, is very high. With the rules extracted by the method introduced here, ANNs should no longer be regarded as black boxes. 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. In the near future, we want applied GenRGA to continuous data.

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.

Kamruzzaman, S.M. and Md. Monirul Islam, 2010. Extraction of symbolic rules from artificial neural networks. World Academy Sci. Engin. Technol., 10: 271-277.

Markowska-Kaczmar, U. and K. Mularczyk, 2006. GA-Based Pareto Optimization of Studies in Computational Intelligence. Springer, Berlin Heidelberg, New York.

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.

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.

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.

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.

Yedjour, D., H. Yedjour and A. Benyettou, 2010. Combining quine mc-cluskey and genetic algorithms for extracting rules from trained neural networks. Asian J. Applied Sci., 4: 72-80.

Zhou, Z.H., Y. Jiang and S.F. Chen, 2003. Extracting symbolic rules from trained neural network ensembles. AI Commun., 16: 3-15.