ESANN'2003 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 23-25 April 2003, d-side publi., ISBN 2-930307-03-X, pp. 149-154

Extraction of Fuzzy Rules from Trained Neural Network Using Evolutionary Algorithm*

Urszula Markowska-Kaczmar, Wojciech Trelak

Wrocław University of Technology, Poland
kaczmar@ci.pwr.wroc.pl, trelak@ci.pwr.wroc.pl

Abstract. This paper presents our approach to the rule extraction problem from trained neural network. A method called REX is briefly described. REX acquires a set of fuzzy rules using an evolutionary algorithm. Evolutionary algorithm searches not only fuzzy rules, but also a description of fuzzy sets. The way of coding and evaluation process of an individual is presented. The method was tested using the following benchmark data sets: IRIS, WINE and Wisconsin Breast Cancer Diagnosis. On the basis of the experimental studies shown in this paper, we can conclude that rules obtained by REX can be easily understood by human – they include small number of premises, and their fidelity is very high. Obtained results are compared to other rule extraction methods.

1. Introduction

In the last decade a big interest in searching methods of knowledge acquisition from neural network (NN) has been observed. Most methods of knowledge extraction are in the form of the set of the prepositional crisp rules (KT [5], Subset, M of N [12], BRE [10], VIA [11]; a detailed survey can be found in [1] and [3]).
Recently, there is a great interest in fuzzy techniques. Several methods of the fuzzy rule extraction from NN [3, 7, 8] and from examples has been proposed [13]. However, some of them assume that the fuzzy sets description is given, eventually allowing a slight tuning of already existing fuzzy sets. The method presented in this paper deal with fuzzy rules and descriptions of fuzzy sets, and is used to extract the knowledge from the trained neural network. It uses an evolutionary algorithm to reach the goal.

2. The main idea of the fuzzy Rule Extraction Method (REX)

The goal of our method is to obtain a set of fuzzy rules, which describes an internal decision of NN executing a classification task. A typical classification problem solved by NN can be described as follows. A set of examples or patterns is represented by a vector of attributes $X = [x_1, \dots x_k]$. Each attribute is associated with one of the NN input. Each pattern is classified by NN into one of the mutually exclusive class from a set of classes $C = \{C_1, \dots, C_N\}$. The class is shown by the NN output vector

* Work partially supported by the Polish Committee for Scientific Research, grant 4T11E02323.

$y = [y_1,..,y_k]$, where only one y_j is equal to 1, and it corresponds to class C_j. Other y_i are equal to 0. Each rule has a form given by (1):

$$\text{IF } x_1 \text{ is } Z_{1i} \text{ AND} \ldots \text{ AND } x_k \text{ is } Z_{kj} \text{ THEN } y_1 y_2 \ldots y_n \qquad (1)$$

where each premise x_k *is* Z_{ki} states that attribute x_k belongs to the fuzzy set Z_{ki}. Symbol x_i corresponds to the input of NN, as well. The conclusion part matches answer of the NN. There are no special limitations on a type of NN, although experiments described later have been made with MLP [4].

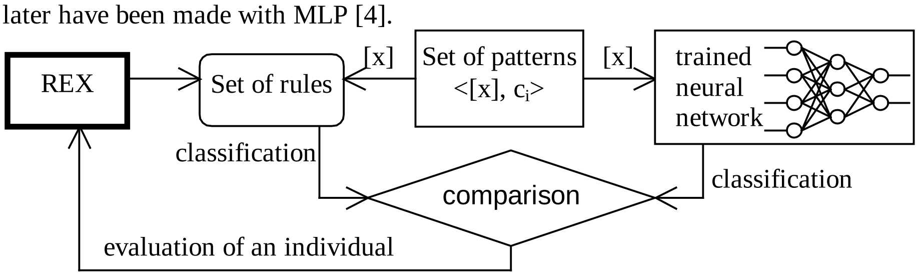

Fig. 1. Schema of the data flow in REX system

In our approach EA seeks (fig. 1) a complete set of fuzzy rules describing classes recognized by the NN. EA optimises the fuzzy sets related to them, as well. After each generation rules are evaluated. To perform this task, input patterns [x] are sequentially passed to the set of rules created in a given generation. One looks for fired rules for each pattern. Concerning the fact that conclusions are not fuzzy, the following reasoning procedure takes place during evaluation. First, activations of all fired rules are calculated. For active premises of the given rule, membership functions are calculated, according to the current pattern. Then, the minimal value is selected (T-norm operator) and the result becomes the activation of the given rule. Finally, the rule with the highest value of activation is selected. The result of the reasoning is the conclusion of the selected rule.
The set of extracted rules is assessed on the basis of fidelity between the answer of NN and the conclusion part of the fired rule for a given input pattern and the patterns covering. It means that each individual is evaluated on the phenotype level.

3. Description of REX method

REX is an evolutionary algorithm, where an individual represents a set of fuzzy rules and fuzzy sets related to them. The length of a chromosome is constant, and each rule, premise and fuzzy set can be marked as active or inactive.

3.1 An individual and the initial population

As stated before, an individual consists of the collection of fuzzy rules and the collection of fuzzy sets grouped according to input variables. The chromosome (fig. 3) is divided into two parts containing genes coding rules and genes coding fuzzy sets. REX uses triangular fuzzy sets (fig. 3). Each fuzzy set is encoded as one real number d_i representing central point, and it begins at the point where the previous active fuzzy set has its central point and ends at the point where the next active fuzzy set has its central point. Before each gene corresponding to the fuzzy set or to the rule stands a flag *F*, which informs if it is active or not. This way of coding leads to simpler rules. Fuzzy sets related to one input variable form a group of fuzzy sets. In each group of fuzzy sets must exist at least one active fuzzy set.

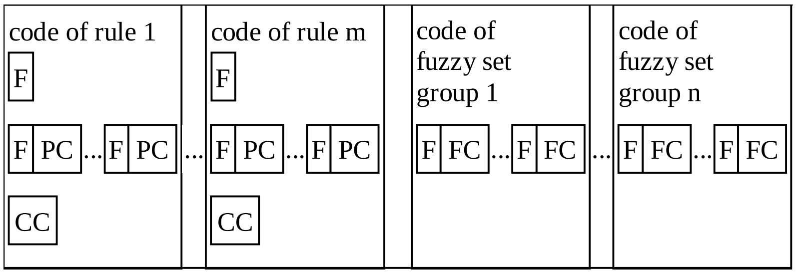

Fig 2. A chromosome of an example individual; *F* – activity bit, PC – code of premise, CC – code of conclusion, FC – code of fuzzy set

Each premise in the rule describes a condition for a given input variable. Premises are coded as an integer number related to an index of fuzzy set in a given group of fuzzy sets. The conclusion of the rule is binary coded.

The first population is selected at random. After selecting a random individual, it has to be repaired. Fuzzy sets in one group have to be ordered from the smallest to the biggest value of the central point. Premises in rules have to point to active fuzzy sets. There must be at least one active premise in each rule, at least one active fuzzy set in a group of fuzzy sets, and at least *c* active rules, where *c* is the number of possible classes of solutions.

3.2 Evaluation of the individual

To evaluate an individual in REX, a fuzzy reasoning must be performed. Its result has to be compared with that of the NN (fig. 1), giving the following metrics:

- *corr* –the number of correctly classified patterns, that is those patterns that fired at least one rule, and the result of reasoning (classification) was equal to the output of the neural network; this parameter determines the fidelity of the rule set;
- *incorr* – the number of incorrectly classified patterns, that is those patterns, for which the reasoning gave different result from the result of the neural network;
- *unclass* – the number of patterns that were not classified, that is all those patterns, for which none of rules fired, and it was impossible to perform the reasoning.

There are also metrics describing complexity of rules and fuzzy sets: *prem* – total number of active premises in active rules, *fsets* – total number of active fuzzy sets in an individual. They all were used to create the following evaluation function:

$$f(i) = \beta \cdot corr_i \cdot \mathrm{k}(incorr_i) + \chi \cdot corr_i \cdot \mathrm{k}(unclass_i) + \delta \cdot \mathrm{k}(prem_i) + \varepsilon \cdot \mathrm{k}(fsets_i) \quad (2)$$

where: i – index of the individual, β, χ, δ, ε – coefficients; k(x) is a function equal to 2 when x = 0, and equal to 1/x when x ≠ 0.

ESANN'2003 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 23-25 April 2003, d-side publi., ISBN 2-930307-03-X, pp. 149-154

3.3 The evolutionary operators and termination condition

There are several mutation operators in REX algorithm. Mutation of the central point of a fuzzy set (FC, fig. 2) is based on adding or subtracting a random floating-point number. Mutation of an integer value (PC) relies on adding a random integer number modulo allowed range. Mutation of a bit (F, bits from CC) is simply its negation.
The crossover operator is uniform one. During the crossover, rules and fuzzy set groups can be exchanged between two individuals.
The algorithm terminates when one of the following conditions occurs: maximum number of steps elapsed, there is no progress for certain number of steps, or when the evaluation function for the best individual reaches certain value.

4. Experimental studies

The tests have been performed on the following test sets taken from UCI Repository: IRIS, WINE and Wisconsin Breast Cancer Diagnosis. Obtained results are shown in tables 1 to 3, and they are average values from 5 runs. Columns named "%" contain fidelity for REX or classification ratio for other methods. Columns signed "NN %" inform about the classification ratio of the NN. We compare our results to Full-RE [9], which is a crisp rule extraction method from NN, and other methods of extraction of fuzzy rules from NN and extraction of fuzzy rules on the base of the set of examples only, trying to find similarities between them.
The assumed parameter values of REX were experimentally determined and they were set to: size of a population = 20, mutation probability = 0.01, crossover probability = 0.6, maximum number of rules in an individual = 20.

method	training	testing	Sum	%	NN %
Full-RE	83/89	60/61	146/150	97.33	97,33
REX	88/89	58/61	146/150	97.33	98,67

Tab. 1. Fidelity of training and testing sets for REX and Full-RE

Next, tests were performed with different test methods on test sets mentioned before. First, IRIS data was divided into two partitions – 89 patterns in learning partition and 61 patterns in testing one. Results show that REX and Full-RE [9] give comparable results in terms of fidelity (tab. 1). One can notice that rules extracted by REX are very similar to those extracted by Full-RE, taking into account the number of rules and the number of premises. Also, boundary values for each attribute in rules extracted by Full-RE are very close to centres and widths of fuzzy sets in premises of rules extracted by REX (fig. 3.).
Another test was performed on IRIS data, but using different testing schema – N-fold cross validation. 150 patterns were divided into 10 partitions. The results of these tests were compared to MDTF [6], which is a method of extraction of fuzzy rules from patterns (tab. 2a). The results of these two methods are relatively the same. However, we have to keep in mind that REX mimics the NN classification, and is expressed by fidelity. The results for MDTF are in terms of classification ratio (number of patterns covered by set of rules extracted only on the basis of the training set). The average

ESANN'2003 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 23-25 April 2003, d-side publi., ISBN 2-930307-03-X, pp. 149-154

number of rules obtained by REX method was also less than by method presented in [8], where after tuning the final number of fuzzy rules was 6.

Full-RE

```
IF x2 ≤ 2.1
       THEN Setosa
IF x2 ≤ 5.1 AND x3 ≤ 1.7
       THEN Versicolor
IF x2 ≥ 4.8
       THEN Virginica
```

REX

```
IF x2 is PL4
       THEN 100   // Setosa
IF x2 is PL14
       THEN 001  // Virginica
IF x2 is PL5 AND x3 is PW8
THEN 010 // Versicolor
```

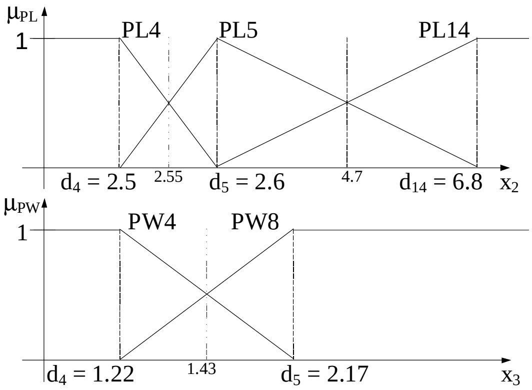

Fig. 3. Rules extracted by Full-RE compared to REX

a.

Method	no. of rules	%	NN %
MDTF	5	97.33	—
REX	3.8	98.67	96.66

b.

method	no. of rules	%	NN %
SLAVE2	5.2	96.76	—
REX	4	90.98	99.40

Tab. 2. The comparison of REX results with other methods. a) Classification ratio and number of rules for MDTF algorithm and fidelity for REX (IRIS) b) Classification ratio and number of rules for SLAVE2 algorithm and fidelity for REX (WINE)

In order to test the scalability of REX we tested it on a data set with a larger number of input parameters. Data set WINE contains 178 patterns with 13 continuous variables (tab. 2b). In comparison to SLAVE2 algorithm [2], which extracts fuzzy rules from examples, we obtained in average a smaller number of rules. However, fidelity of REX was considerably smaller than classification ratio of SLAVE2. Similar situation was noticed for the method presented in [7], where classification rate was 100%, yet the number of fuzzy rules was 17.

The last data set used was Wisconsin Breast Cancer Diagnosis. It consists of 569 patterns, and each pattern has 30 continuous input variables and two possible output values. We obtained in average 3 rules with fidelity 95.97% (with classification ratio of the NN 95.16%). The results suggest that REX can extract a small number of rules from large and complicated (large number of input variables) data sets.

5. Conclusions

In this paper the method of extraction of fuzzy rules from neural network is presented. The intention of use of REX is not to replace NN – much faster and more accurate than reasoning based on the set of rules extracted from it. On the contrary, REX can be used in hybrid systems, in explanatory mechanisms.

This approach focuses not only on extraction of fuzzy rules, but provides the simultaneous tuning of fuzzy sets during searching fuzzy rules. It also extracts rule sets in average with a small number of rules from NN for presented data sets. The rules contain a small number of premises, too. Tests show that REX can be found as a method for rule extraction from domains with a large number of continuous attributes. However, in some cases (see comparison REX to SLAVE2) the method can generate rules that cover smaller domain area than compared method, but the results from SLAVE2 and REX cannot be compared to the full extent.

References

1. Andrews, R., Survey and critique of techniques for extracting rules from trained artificial neural networks. Knowledge–Based Systems, Volume: 8, Issue: 6, December, (1995) 373–389.
2. Castillo L., González A., Pérez R., Including a simplicity criterion in the selection of the best rule in a genetic algorithm, Fuzzy Sets and Systems, 120 (2001), 309-321.
3. Duch W, Adamczak R, Grąbczewski K., A new methodology of extraction, optimization and application of crisp and fuzzy logical rules, IEEE Transactions on Neural Networks 12 (2001), 277-306.
4. Freeman, J., Skapura, D., Neural Networks, Algorithms, Applications and Programming Techniques. Addison–Wesley Publishing Company, Inc. (1992).
5. Fu L., M., Rule Generation from neural network IEEE. Transactions on Systems, Man and Cybernetics vol. 24, No. 8. August (1994) 1114–1124.
6. Hong T.-P., Chen J.-B., Processing individual fuzzy attributes for fuzzy rule induction, Fuzzy Sets and Systems, 112 (2000), 127-140.
7. Ishibushi H., Nii M., Tanaka K., Linguistic rule extraction from neural network for high-dimensional classification problems, Complexity International vol. 6, 1999.
8. Matthews Ch., Jagielska I., Fuzzy rule extraction from trained multilayer neural network, IEEE ICNN'95 Int. Conf. on Neural Networks, Perth, November 1995.
9. Taha I., Ghosh J., Symbolic interpretation of artificial neural networks, Tech. Report TR-97-01-106, the Computer and Vision Research Center, University of Texas, Austin, 1996.
10. Tickle, A., Orlowski, M., Diedrich, J., A methodology for extracting rules from trained neural networks. Rules and Networks: Proceedings of the Rule Extraction From Trained Artificial Neural Networks Workshop, Queensland University of Technology, Neurocomputing Research Center QUT NRC, April (1996) 90–102.
11. Thrun, S., Extracting rules from artificial neural networks with distributed representation. Proceedings of the Eleventh International Conference on Machine Learning, New Brunswick, NJ. Morgan Kaufman.
12. Towell, G. G., Shavlik, J., Extracting refined rules from knowledge based neural networks. Machine Learning vol. 131 (1993) 71–101.
13. Yuan Y., Zhuang H., A genetic algorithm for generating fuzzy classification rules, Fuzzy Sets and Systems 84 (1996), 1–19.