Neural-Symbolic Learning and Reasoning
AAAI Technical Report WS-12-11

Efficiently Merging Symbolic Rules into Integrated Rules

Jim Prentzas[a], Ioannis Hatzilygeroudis[b]

[a] Democritus University of Thrace, School of Education Sciences
Department of Education Sciences in Pre-School Age
68100 Nea Chili, Alexandroupolis, Greece
dprentza@psed.duth.gr

[b] University of Patras, School of Engineering
Department of Computer Engineering & Informatics
26500 Patras, Greece
ihatz@ceid.upatras.gr

Abstract

Neurules are a type of neuro-symbolic rules integrating neurocomputing and production rules. Each neurule is represented as an adaline unit. Neurules exhibit characteristics such as modularity, naturalness and ability to perform interactive and integrated inferences. One way of producing a neurule base is through conversion of an existing symbolic rule base yielding an equivalent but more compact rule base. The conversion process merges symbolic rules having the same conclusion into one or more neurules. Due to the inability of the adaline unit to handle inseparability, more than one neurule for each conclusion may be produced. In this paper, we define criteria concerning the ability or inability to convert a rule set into a single neurule. Definition of criteria determining whether a set of symbolic rules can (or cannot) be converted into a single, equivalent but more compact rule is of general representational interest. With application of such criteria, the conversion process of symbolic rules into neurules becomes more time- and space-efficient by omitting useless trainings. Experimental results are promising.

Introduction

There have been efforts combining neural and symbolic approaches (Garcez, Broda and Gabbay 2002). Neural networks and symbolic rules have complementary advantages and disadvantages (Hatzilygeroudis and Prentzas 2004) and their combination constitutes a popular research trend. Neurules (Hatzilygeroudis and Prentzas 2010, Prentzas and Hatzilygeroudis 2011) are a type of integrated rules combining symbolic rules (of propositional type) and neurocomputing (adaline approach). In contrast to other approaches, neurules give pre-eminence to the symbolic part of the integration. Therefore, they retain the naturalness and modularity of symbolic rules in a large degree. Neurules can be produced either from symbolic rules or from empirical data (Hatzilygeroudis and Prentzas 2000, 2001). Also a neurule-based system possesses an interactive inference mechanism (Hatzilygeroudis and Prentzas 2010).

A neurule base may be produced from a symbolic rule base by applying a conversion process (Hatzilygeroudis and Prentzas 2000). Conversion does not involve refinement of the symbolic rule base, but creates an equivalent knowledge base. This means that the conclusions drawn from the neurule base are the same as those drawn from the symbolic rule base, given the same inputs. Each produced neurule usually merges two or more symbolic rules with the same conclusion. Therefore, the size of the produced neurule base is less than that of the symbolic rule base as far as both the number of rules and the number of conditions are concerned. This results in improvements to the efficiency of the inferences from the neurule base, compared to those from the symbolic rule base as shown in (Hatzilygeroudis and Prentzas 2000). The conversion process tries to merge all symbolic rules having the same conclusion into a single neurule. However, this is not always possible, due to the inability of the adaline unit to handle inseparability, and thus more than one neurule for each conclusion may be produced.

In this paper, we define criteria concerning the ability or inability to convert a rule set to a single neurule which is an equivalent but more compact rule-based approach. With application of such criteria, the conversion process of symbolic rules to neurules becomes more efficient by

avoiding trainings not directly producing neurules. The definition of such criteria is of general representational interest. Various approaches have been presented concerning the reduction of the size of rule bases and/or the production of compact rule bases. Such approaches have been presented in the context of rule extraction from neural networks (Chorowski and Zurada 2011), learning rules from datasets (Riid and Rustern 2011) and evolutionary computing (Shi, Shi and Gao 2009). Our approach lies in a neuro-symbolic context that provides integrated inference, involves available symbolic rules elicited from experts or produced from datasets and reduces rule base size through conversion to an equivalent and more compact formalism by merging symbolic rules.

This paper is organized as follows. First, main aspects concerning neurules are outlined. The following section discusses criteria for efficiently merging symbolic rules into neurules. Experimental results are then presented. Finally, it concludes.

Neurules: Syntax and Semantics

Neurules are a kind of integrated rules. The form of a neurule is depicted in Figure 1a. Each condition C_i is assigned a number sf_i, called its *significance factor*. Moreover, each rule itself is assigned a number sf_0, called its bias factor. Internally, each neurule is considered as an adaline unit (Figure1b). The inputs C_i (i=1,...,n) of the unit are the conditions of the rule. The weights of the unit are the significance factors of the neurule and its bias is the bias factor of the neurule. Each input takes one of the following discrete values: [1(true), -1(false), 0(unknown)].

The output D, which represents the conclusion of the rule, is calculated via the standard formulas:

$$D = f(a), a = sf_0 + \sum_{i=1}^{n} sf_i C_i \qquad (1)$$

$$f(a) = \begin{cases} 1, & if \ \ a \geq 0 \\ -1, & if \ \ a < 0 \end{cases} \qquad (2)$$

where ***a*** is the *activation value* and *f(x)* the *activation function*, which is a threshold function. Hence, the output can take one of two values ('-1', '1') representing failure and success of the rule respectively. The significance factor of a condition represents the significance (weight) of the condition in drawing the conclusion. The LMS learning algorithm is used to compute the values of the significance factors as well as the bias factor of a neurule. Examples of neurules are shown in Tables 2 and 4.

The general syntax of a neurule (in a BNF notation, where '< >' denotes non-terminal symbols) is:

<rule>::= (<bias-factor>) **if** <conditions> **then** <conclusion>
<conditions>::= <condition> | <condition>,<conditions>
<condition>::= <variable> <l-predicate> <value> (<significance-factor>)
<conclusion>::= <variable> <r-predicate> <value> .

where <variable> denotes a *variable*, that is a symbol representing a concept in the domain, e.g. 'sex', 'pain' etc in a medical domain, and <l-predicate> denotes a symbolic or a numeric predicate. The symbolic predicates are {is, isnot}, whereas the numeric predicates are {<, >, =}. <r-predicate> can only be a symbolic predicate. <value> denotes a value; it can be a symbol (e.g. "male", "night-pain") or a number (e.g "5"). <bias-factor> and <significance-factor> are (real) numbers

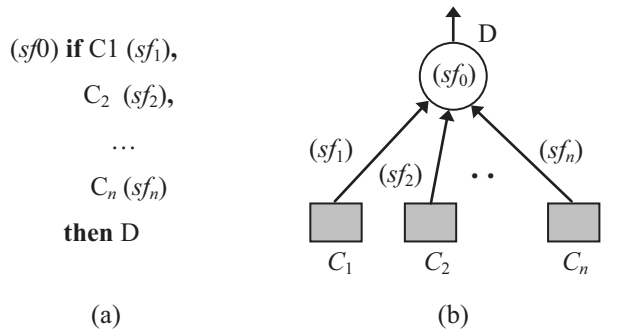

Figure 1. (a) Form of a neurule (b) corresponding adaline unit

Merging Symbolic Rules into Neurules

One way of producing a neurule base (NRB) is by conversion from a (propositional type) symbolic rule base (SRB). A symbolic rule consists of a conjunction of conditions and a conclusion. Examples of symbolic rules are shown in Tables 1 and 3 where "," (as already mentioned) denotes conjunction. Existing SRBs (of propositional type) can be easily transformed into an SRB of the above syntax and then converted to an NRB. An SRB may be the result of direct knowledge elicitation from experts or the product of an automated knowledge acquisition method. In this way, existing SRBs can be exploited for the production of neurules.

The conversion of an SRB to an NRB is achieved by applying the conversion algorithm presented in (Hatzilygeroudis and Prentzas 2000). Application of the conversion algorithm does not result in a refinement of the converted SRB. It creates an equivalent knowledge base (NRB) whose size is less than that of SRB. The conversion algorithm tries to merge all symbolic rules having the same conclusion into one neurule. However, this is not always possible, due to non-linearity problems, as explained later in this section. In any case, each produced neurule usually is the result of merging two or more symbolic rules.

We introduce the following definitions:

- Two conditions C_i and C_k are *related* if they contain the same variable.
- *conds(R_i)* denotes the set of conditions of rule R_i.
- A set of symbolic rules MS={R_1, R_2,…, R_m}, m≥1 is called a *merger set* if all rules contain the same conclusion.
- A *non-merging rule* is a symbolic rule with a unique conclusion in the SRB.
- The *closeness* between two symbolic rules (*R-closeness*) is defined as the number of their common conditions (i.e. conditions having the same variable, predicate and value).
- A *least closeness pair* (LCP) of rules in a merger set is a of pair rules that have the minimum R-closeness.

The conversion algorithm is outlined as follows:

1. Group symbolic rules into (initial) merger sets.
2. For each merger set,
2.1 Construct a merger
2.2 Produce a training set for the merger
2.3 Train the merger individually
2.4 If training is successful, produce the corresponding neurule.
2.5 Otherwise, split the merger set into two disjoint subsets and execute recursively Steps 2.1-2.5 for each subset.

The initial merger sets contain all rules of the SRB having the same conclusion. A merger is a neurule having as conditions all the conditions of the symbolic rules in the corresponding merger set without duplications and significance factors as well as bias factor set to a proper initial value. For each merger, a training set is extracted from the truth table of the combined logical function of the rules in the set (the disjunction of the conjunctions of the conditions of each rule) eliminating unacceptable training patterns since certain conditions cannot be simultaneously true or false (Hatzilygeroudis and Prentzas 2000).

Each merger is individually trained using the standard LMS algorithm. Training of a merger may not be always successful meaning that it cannot always find a set of significance and bias factors that classify correctly all of the training patterns. This is so, if the patterns of the training set are inseparable (as in the case of the patterns corresponding to the XOR function). When training fails, the merger set is split into disjoint subsets producing more than one neurule having the same conclusion.

Splitting a merger (sub)set is guided by an LCP (chosen based on a strategy) of the merger set. Two merger subsets are created each containing as its initial element one of the rules of the LCP, called its pivot. Each of the other rules in the set is distributed between the two subsets based on their closeness to their pivots. That is, each subset contains rules, which are closer to its pivot. If training fails, for a merger of a merger subset, the corresponding subset is further split into two other subsets, based on one of its LCPs. This continues, until training succeeds or the merger subset contains only one rule. This kind of splitting stems from the observation that separable sets have rules with larger average closeness than inseparable ones.

Table 1. A set of symbolic rules

R_1	***R_3***
if patient is human0-20,	**if** patient is human21-35,
fever is high,	fever is medium,
pain is night	pain is continuous
then disease is inflammation	then disease is inflammation
R_2	***R_4***
if patient is human0-20,	**if** patient is human36-55,
fever is no-fever,	fever is high,
ant-reaction is medium,	pain is night
pain is night	**then** disease is inflammation
then disease is inflammation	

Table 2. Neurules produced from the merger set in Table 1

NR_1-R_2-R_4
(-5.6) **if** fever is high (8.7),
pain is night (8.6),
patient is human0-20 (8.2),
patient is human36-55 (5.1),
fever is no-fever (1.5),
ant-reaction is medium (1.3)
then disease is inflammation
NR_3
(-2.0) **if** pain is continuous (1.1),
fever is medium (0.8),
patient is human21-35 (0.8)
then disease is inflammation

As an example, we use the merger set shown in Table 1 that consists of four symbolic rules {R_1, R_2, R_3, R_4} taken from a medical diagnosis rule base. The merger of this merger set contains the nine distinct conditions of the four rules. The training set of the merger is extracted from the truth table of the combined logical function of the rules of the merger set: $F = (C_1 \wedge C_2 \wedge C_3) \vee (C_1 \wedge C_4 \wedge C_5 \wedge C_3) \vee (C_6 \wedge C_7 \wedge C_8) \vee (C_2 \wedge C_3 \wedge C_9)$, where C_1≡patient is human0-20, C_2≡fever is high, C_3≡pain is night, C_4≡fever is no-fever, C_5≡ant-reaction is medium, C_6≡patient is human21-35, C_7≡fever is medium, C_8≡pain is continuous, C_9≡patient is human36-55. The training patterns of the training set are inseparable and the initial merger set is split in two subsets: MS_1={R_1, R_2, R_4} and MS_2={R_3}. The LCP that guides splitting is (R_1, R_3). Training of the merger of MS_1 is successful and neurule NR_1-R_2-R_4 is produced. Rule R_3 is converted to a neurule (i.e. NR_3). So, from the initial merger set of four symbolic rules, two neurules are produced. Table 2 depicts the produced neurules.

Mergability Criteria

An aspect of interest in the above process concerns introduction of criteria concerning the 'mergability' of a merger set that is, determining whether a merger set can (or cannot) be converted (or merged) into a single neurule without using training. By determining in advance whether a merger set cannot be converted into a single neurule, training of the corresponding merger can be omitted. In such cases, splitting could be directly performed, without training the mergers. So, the time required to convert a symbolic rule base into a neurule base would decrease, since certain trainings would be omitted.

In the aforementioned example (rules of Table 1), three trainings concerning the mergers of merger (sub)sets $\{R_1, R_2, R_3, R_4\}$, $\{R_1, R_2, R_4\}$ and $\{R_3\}$ were performed. Two of the trainings, those corresponding to merger subsets $\{R_1, R_2, R_4\}$ and $\{R_3\}$, were successful and resulted in the production of neurules. By avoiding the training corresponding to merger set $\{R_1, R_2, R_3, R_4\}$ and simply splitting the set to subsets $\{R_1, R_2, R_4\}$ and $\{R_3\}$, conversion would have taken less time.

Besides conversion time gains, determining whether a merger set can be converted into a single neurule is of general interest from a representational point of view. More specifically, it would be interesting to determine criteria of whether a set of symbolic rules can (or cannot) be converted to a single, equivalent but more efficient neuro-symbolic rule. This is attempted in the following.

A merger corresponding to a merger set containing a single symbolic rule can be successfully trained since its training set corresponds to a conjunction and is separable. So, the interest goes to merger sets containing at least two symbolic rules. It should be mentioned that the rules in a merger set may contain an unequal number of conditions. We define criteria guided from experimental results. The criteria are based on R-closeness of rule pairs in a merger set. In certain criteria, we distinguish between merger sets with rules that contain related conditions and merger sets with rules that do not contain related conditions. Some criteria apply to both types of merger sets. Examples regarding specific merger sets are also given for the defined criteria. The defined criteria involve merger sets with rules containing at least two conditions.

It should be mentioned that in order to be able to merge a set of rules into a single neurule, corresponding rules should have certain common conditions. More specifically, a pair of symbolic rules without any common condition cannot be merged into a single neurule. Therefore, merger sets containing one or more pairs of such rules cannot be converted into a single neurule. So, we introduce the following criterion:

Mergability criterion 1. *A merger set* $MS=\{R_1, R_2,...,R_m\}$, $m\geq 2$, $|conds(R_i)|\geq 2$ $\forall R_i \in MS$, *cannot be converted to a single neurule if* $\exists (R_i, R_k)$, $R_i, R_k \in MS$ *with* $R\text{-closeness}(R_i, R_k)=0$, $1\leq i\leq m$, $1\leq k\leq m$, $i\neq k$.

Criterion 1 is satisfied by the merger set $\{R_1, R_2, R_3, R_4\}$ of the rules in Table 1, given that $R\text{-closeness}(R_1,R_3)=0$, $R\text{-closeness}(R_2,R_3)=0$ and $R\text{-closeness}(R_3,R_4)=0$. Therefore, this merger set cannot be converted into a single neurule.

According to criterion 1, a requirement that a merger set $MS=\{R_1, R_2,...,R_m\}$, $m\geq 2$ should satisfy so that it could be converted into a single neurule is the satisfaction of the condition: $R\text{-closeness}(R_i, R_k)>0$ $\forall$ $i\neq k$, $1\leq i\leq m$, $1\leq k\leq m$. In order to identify specific positive values for R-closeness of rule pairs in a merger set that might have an effect on the mergability of a merger set, we conducted a number of experiments. We started with merger sets containing only two rules and then investigated merger sets with at least three rules.

We noticed that any merger set $MS=\{R_1, R_2\}$ with only two rules can be converted into a single neurule if $R\text{-closeness}(R_1, R_2)=\min(|conds(R_1)|, |conds(R_2)|) - 1$. We also noticed that any merger set $MS=\{R_1, R_2,...,R_m\}$, $m\geq 2$ can be converted into a single neurule if the merger set of each pair $\{R_i, R_k\}$ can be converted to a single neurule, $\forall$ $i\neq k$, $1\leq i\leq m$, $1\leq k\leq m$. As explained in the following, this condition is less strict for merger sets with rules that have related conditions. However, merger sets whose rules do not have any related conditions can be converted into a single neurule only if the merger set of each pair of rules can itself be converted into a single neurule. Therefore, we introduce the following criterion:

Mergability criterion 2. *A merger set* $MS=\{R_1, R_2,...,R_m\}$, $m\geq 2$, $|conds(R_i)|\geq 2$ $\forall R_i \in MS$, whose rules do not have any related conditions *can be converted into a single neurule only if* $R\text{-closeness}(R_i, R_k) = \min(|conds(R_i)|, |conds(R_k)|) - 1$, $\forall i\neq k$, $1\leq i\leq m$, $1\leq k\leq m$.

For instance, this second criterion is satisfied by the merger set $\{R_1, R_2, R_3, R_4\}$ of rules in Table 3. So, the merger set can be converted into a single neurule (shown in Table 4). Notice that those four symbolic rules do not have related conditions.

As mentioned, a merger set $MS=\{R_1, R_2,...,R_m\}$, $m\geq 2$ whose rules have related conditions can be converted into a single neurule even if $R\text{-closeness}(R_i,R_k) < \min(|conds(R_i)|, |conds(R_k)|) - 1$ for some rules R_i, R_k, $i\neq k$, $1\leq i\leq m$, $1\leq k\leq m$. This is so because certain training patterns are excluded from the merger's training set as invalid.

So, criterion 2 for mergers with rules having related conditions becomes as follows:

Mergability criterion 2A. *A merger set* $MS=\{R_1, R_2,...,R_m\}$, $m\geq 2$, $|conds(R_i)|\geq 2$ $\forall R_i \in MS$, *containing rules having related conditions can be converted into a single neurule if* $R\text{-closeness}(R_i,R_k) = \min(|conds(R_i)|, |conds(R_k)|) - 1$, $\forall i\neq k$, $1\leq i\leq m$, $1\leq k\leq m$.

Furthermore, experiments showed that a merger set containing rules with related conditions, with most pairs of

rules (R_m, R_j) having R-closeness(R_m, R_j)=min(|conds(R_m)|, |conds(R_j)|) – 1 and with certain pairs of rules (R_i, R_k) having R-closeness(R_i, R_k)=min(|conds(R_i)|, |conds(R_k)|) – 2 can be converted into a single neurule. So, we introduce the following two criteria.

Table 3. A set of rules that do not contain related conditions

R_1	***R_3***
if var1 is A1,	**if** var1 is A1,
var3 is C1,	var3 is C1,
var5 is F3,	var5 is F3,
var7 is H2	var7 is H2,
var6 is G1	var8 is I1
var9 is J1	**then** output is D
then output is D	
R_2	***R_4***
if var1 is A1,	**if** var1 is A1,
var3 is C1,	var2 is B1,
var4 is E3,	var3 is C1
var7 is H2	**then** output is D
then output is D	

Table 4. Neurule produced from the merger set in Table 3

NR_1-R_2-R_3-R_4
(-92.3) **if** var3 is C1 (58.8),
var1 is A1 (55.6)
var2 is B1 (37.5)
var7 is H2 (22.5)
var4 is E3 (12.4),
var5 is F3 (8.3),
var8 is I1 (4.4),
var6 is G1 (1.5),
var9 if J1 (1.0)
then output is D

Mergability criterion 3. *A merger set MS=$\{R_1, R_2, ..., R_m\}$, $m \geq 2$, $|conds(R_i)| \geq 3$ $\forall R_i \in MS$, containing rules having related conditions cannot be converted to a single neurule if the following is satisfied:*
$\exists (R_i, R_k)$, $R_i, R_k \in MS$ with R-closeness$(R_i, R_k) <$ $min(|conds(R_i)|, |conds(R_k)|) - 2$, $1 \leq i \leq m$, $1 \leq k \leq m$, $i \neq k$.

Mergability criterion 4. *A merger set MS=$\{R_1, R_2, ..., R_m\}$, $m \geq 2$, $|conds(R_i)| \geq 3$ $\forall R_i \in MS$, containing rules having related conditions, cannot be converted into a single neurule if the following are satisfied:*
i) $\forall R_i, R_k \in MS$, R-closeness$(R_i, R_k) \geq min(|conds(R_i)|, |conds(R_k)|) - 2$, $1 \leq i \leq m$, $1 \leq k \leq m$, $i \neq k$ and
ii) $|S_{P1}| < |S_{P2}|$, where $S_{P1} = \{(R_i, R_k)$: $R_i, R_k \in MS$, R-closeness$(R_i, R_k) = min(|conds(R_i)|, |conds(R_k)|) - 1$, $1 \leq i \leq m$, $1 \leq k \leq m$, $i \neq k\}$ and $S_{P2} = \{(R_i, R_k)$: $R_i, R_k \in MS$, R-closeness$= min(|conds(R_i)|, |conds(R_k)|) - 2$, $1 \leq i \leq m$, $1 \leq k \leq m$, $i \neq k\}$.

It should be mentioned that in case of merger sets with rules having related conditions, the first criterion is subsumed by the third criterion. In the following, we give examples for the third and fourth criterion.

The third criterion is satisfied by the merger set $\{R_1, R_2, R_3, R_4\}$ of rules in Table 1. So, this merger set cannot be converted into a single neurule.

Furthermore, by the merger subset $\{R_1, R_2, R_4\}$ neither third nor fourth criterion is satisfied. More specifically, condition (i) of the fourth criterion is satisfied. However, $S_{P1} = \{(R_1, R_2), (R_1, R_4)\}$ and $S_{P2} = \{(R_2, R_4)\}$, so $|S_{P1}| > |S_{P2}|$, since $|S_{P1}|=2$ and $|S_{P2}|=1$. Thus, mergability criteria cannot give indications that the merger set cannot be converted into a single neurule. Indeed, after training, neurule NR_1-R_2-R_4 (shown in Table 2) is produced.

Conversion Process Improvement

By checking the satisfaction of the mergability criteria, the conversion algorithm is improved, given that certain (unnecessary) training effort may be omitted. This is based on the indications about merger sets provided by the mergability criteria, which are the following:

(a) A merger set can be converted into a single neurule (satisfaction of the second criterion).
(b) A merger set cannot be converted into a single neurule (satisfaction of the third or fourth criterion for merger sets which contain rules having related conditions and all rules have at least three conditions and; second criterion is not satisfied for merger sets which contain rules not having related conditions; satisfaction of first criterion for merger sets containing certain rules with two conditions).
(c) There is no indication that a merger set which contains rules having related conditions cannot be converted into a single neurule (first, third and fourth criterion are not satisfied).

So, the conversion process has been revised to take into account the indications provided by the mergability criteria. This is done by embedding the following steps:

1. If the second criterion is satisfied by the merger set, then the merger set can be converted to a single neurule.
2. Else if the rules in the merger set have no related conditions, then the merger set cannot be converted to a single neurule.
3. Else if the some rules in the merger set have two conditions and the first criterion is satisfied, then the merger set cannot be converted to a single neurule.
4. Else if the rules in the merger set have related conditions and all rules have at least three conditions and the third criterion is satisfied, then the merger set cannot be converted to a single neurule.
5. Else if the rules in the merger set have related conditions and all rules have at least three conditions and the fourth

criterion is satisfied, then the merger set cannot be converted to a single neurule.

6. Else if none of the above is satisfied, then there is no indication from the mergability criteria that the specific merger set cannot be converted to a single neurule.

Experimental Results

Experiments were conducted to demonstrate the improvement in the conversion process by applying the defined criteria. More specifically, the criteria were applied to indicate whether a merger set cannot be converted to a single neurule and omit training effort. The conversion process was applied to two symbolic rule bases concerning medical diagnosis. The rules in the ensuing merger sets had related conditions. We applied the first, third and fourth criteria (in this order) to save useless trainings. We could have avoided applying the first criterion, given that it is subsumed by the third one, but we wanted to record the percentage of merger sets satisfying it.

Table 5. Rule base and conversion characteristics

Rule Base and Conversion Characteristics	***RB1***	***RB2***
Number of symbolic rules in SRB	68	130
Number of neurules in equivalent NRB	39	86
Number of non-merging symbolic rules	18	37
Number of initial merger sets (includes number of non-merging symbolic rules)	28	59
Number of total merger trainings (excluding sets with a single rule)	24	51
Number of merger sets having a single rule (excluding non-merging rules)	4	26
Number of merger trainings that would have failed (included in total merger trainings)	7	28
Number of merger trainings that would have failed (as indicated by criteria)	7 (4/3/0)	28 (1/20/6)

Table 5 summarizes characteristics of the rule bases and the conversion process. The table depicts separately the number of non-merging symbolic rules and the number of merger sets that (after splitting) have a single rule. The number of produced neurules is the sum of the number of non-merging rules, the number of merger sets having a single rule and the number of total merger trainings subtracting the number of merger trainings that would have failed (not producing neurules). Entry "7 (4/3/0)" in the last row means on the one hand that in total the criteria indicate that seven trainings would have failed and on the other hand, four, three and zero of those trainings were indicated by the first, third and fourth criterion respectively. By applying the defined criteria, the merger trainings that would have failed are indicated and thus omitted. By applying the criteria, roughly 29% (7/24) and 55% (27/49) of trainings respectively are omitted (not taking into account merger sets having a single rule). All of the three criteria are useful. Runtime savings were roughly 90% (with upper limit of training epochs set to 300).

Conclusion

In this paper, we define criteria determining whether a set of symbolic rules having the same conclusion can (or cannot) be merged into a single neurule, a type of integrated rule. The conversion process of symbolic rules to neurules involves certain trainings not directly resulting in the production of neurules. A practical aspect of the criteria is to identify and omit such trainings.

Our future research work involve additional experiments concerning other available symbolic rule bases from other domains. Also, further investigation for more criteria is on the agenda. Finally, we intend to test alternative conversion algorithms based on the defined criteria.

References

Chorowski, J., and Zurada, J.M. 2011. Extracting Rules from Neural Networks as Decision Diagrams. *IEEE Transactions on Neural Networks* 22:2435–2446.

d'Avila Garcez, A.S., Broda, K., and Gabbay, D.M. 2002. *Neural-Symbolic Learning Systems: Foundations and Applications.* In: Perspectives in Neural Computing. Berlin Heidelberg: Springer-Verlag.

Hatzilygeroudis, I., and Prentzas, J. 2000. Neurules: Improving the Performance of Symbolic Rules. *International Journal on Artificial Intelligence Tools* 9:113–130.

Hatzilygeroudis, I., and Prentzas, J. 2001. Constructing Modular Hybrid Knowledge Bases for Expert Systems. *International Journal on Artificial Intelligence Tools* 10:87–105.

Hatzilygeroudis, I., and Prentzas, J. 2004. Neuro-Symbolic Approaches for Knowledge Representation in Expert Systems. *International Journal on Hybrid Intelligent Systems* 1:111–126.

Hatzilygeroudis, I., and Prentzas, J. 2010. Integrated Rule-Based Learning and Inference. *IEEE Transactions on Knowledge and Data Engineering* 22:1549–1562.

Prentzas, J., and Hatzilygeroudis, I. 2011. Neurules – A Type of Neuro-Symbolic Rules: An Overview. In Hatzilygeroudis, I., and Prentzas, J. (eds.), *Combinations of Intelligent Methods and Applications,* Smart Innovation, Systems and Technologies, 8, 145–165. Berlin Heidelberg: Springer-Verlag.

Riid, A., and Rustern, E. 2011. An Integrated Approach for the Identification of Compact, Interpretable and Accurate Fuzzy Rule-Based Classifiers from Data. In *Proceedings of the 15th International Conference on Intelligent Engineering Systems,* 101–107. IEEE.

Shi, L., Shi, Y. and Gao, Y. 2009. Clustering with XCS and Agglomerative Rule Merging. In *Lecture Notes in Computer Science, 5788,* 242–250. Berlin Heidelberg: Springer-Verlag.