ESANN 2014 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 23-25 April 2014, i6doc.com publ., ISBN 978-287419095-7.
Available from http://www.i6doc.com/fr/livre/?GCOI=28001100432440.

Extracting rules from DRASiW's "mental images"

Paulo V. S. Coutinho, Hugo C. C. Carneiro, Danilo S. Carvalho, Felipe M. G. França *

Universidade Federal do Rio de Janeiro - PESC/COPPE
Rio de Janeiro, Brazil

Abstract. DRASiW is an extension of the WiSARD weightless neural model that provides the ability of producing examples/prototypes, called "mental images", from learnt categories. This work introduces a novel way of performing rule extraction by applying the WiSARD/DRASiW RAM-based neural model upon a well-known machine learning benchmark. A functional exploration is offered in order to demonstrate how the new rule extraction mechanism behaves under different system configurations. Experimental results suggest that the rules conformance to data increases proportionally to the corresponding classifier accuracy. Furthermore, comparison with C4.5 decision tree algorithm shows that the DRASiW-based technique produces more compact sets of rules.

1 Introduction

DRASiW [1] [2] is an extension to the WiSARD weightless neural model capable of producing approximated examples of learnt categories, the so-called "mental images". This work introduces a novel way of performing rule extraction (production rules) from "mental images" produced by a WiSARD/DRASiW multidiscriminator trained with the *Iris* machine learning dataset [3, 4]. In order to demonstrate how the proposed rule extraction mechanism works, the system is stressed under different configurations. The remainder of the text is organised in the following manner. Background knowledge on WiSARD, DRASiW and bleaching is briefly reviewed in Section 2. The explanation of how rules can be extracted by the use of DRASiW is presented in Section 3. Section 4 provides experiments with a well-known benchmark and a detailed analysis in order to demonstrate the capabilities of DRASiW. Finally, in Section 5 some concluding points are drawn and some further research challenges are presented.

2 WiSARD and Bleaching

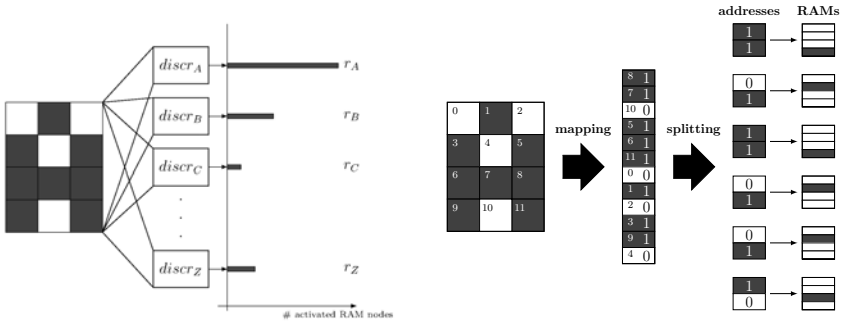
WiSARD (**Wi**lkie, **S**tonham and **A**leksander's **R**ecognition **D**evice) [5] is both a weightless neural network and an n-tuple classifier, i.e., its input is an array of bits. The basic structure of its architecture is a RAM (random access memory) discriminator. Each of these structures is assigned to a particular category, therefore a WiSARD network possesses as many discriminators as the number

*This work was partially supported by Inovax, and CAPES, CNPq and FAPERJ Brazilian research agencies.

ESANN 2014 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 23-25 April 2014, i6doc.com publ., ISBN 978-287419095-7.
Available from http://www.i6doc.com/fr/livre/?GCOI=28001100432440.

of categories it should be able to distinguish. WiSARD can be thus called a multidiscriminator architecture. Figure 1a depicts this architecture.

The discriminators are composed by RAM nodes, which store the information learnt by the network. Every node is addressed by an array of bits which is formed by concatenating particular bits, shuffled from the network input according to a pseudo-random mapping. This procedure is demonstrated in Figure 1b. The generalising capability of WiSARD highly depends on the size of the inputs of the RAM nodes in its discriminators. The smaller their input size more generalising is the network.



(a) Multidiscriminator architecture (b) Pseudomapping procedure

Fig. 1: WiSARD architecture

WiSARD training procedure consists in initially setting every memory position to "0", and then, for every input pattern (input, category) (i, c) presented to the network, set to "1" the position addressed by this pattern. The classification procedure consists of retrieving the information stored in the memory positions by addressing every RAM node of every discriminator. The nodes then output the information stored in their addressed position and, finally, the discriminators output a similarity measure whose value is the sum of the responses of their RAM nodes, i.e., the number of addressed positions which were accessed during the training phase. The chosen category is the one assigned to the discriminator with the highest response.

WiSARD is an overtraining-prone n-tuple classifier, since if a large amount of patterns is trained, especially noisy ones, many memory positions are set to "1" and then, the network gets "saturated", i.e., ties would happen often given that, upon testing a target pattern, the vast majority of RAM nodes would output "1" and thus two or more discriminators would output the highest possible response. DRASiW is an extension to WiSARD that takes profit from the way WiSARD is trained. Instead of having memory positions simply set to "1" if accessed under training ("0" otherwise), every memory position stores the amount of times it was accessed during the training phase. Thus it allows one to check what a

ESANN 2014 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 23-25 April 2014, i6doc.com publ., ISBN 978-287419095-7.
Available from http://www.i6doc.com/fr/livre/?GCOI=28001100432440.

WiSARD discriminator has learnt by reconstructing an image from the different memory positions writing frequencies, the so called "mental images".

A tie-breaking technique, called "bleaching", was derived from the idea of DRASiW's "mental images". This technique uses a threshold b, initially set to 0, which is used during the classification phase: each RAM node, instead of output "1" if it were accessed and "0" otherwise, it would output "1" if it were accessed more than b times and "0" otherwise. Furthermore, if a classification procedure ends in a tie, the value of b should be incremented by a constant k; the procedure should be done again until the tie is broken [6].

3 Rule extraction

By using the DRASiW extension, WiSARD is capable of storing integer numbers in its memory positions instead of only "0" and "1". This way, it is possible to compare *informally equivalent* memory positions. Two memory positions p_1 and p_2 are informally said to be *equivalent* if they have the same input address and if their RAM nodes r_1 and r_2 belong to different discriminators but their input address lines (**IAL**s – number of RAM address input bits) are composed from the same indices of the network input array. For example, the input address marked position of the first RAM node (from top to bottom) of Figure 1b is "11" and its IALs are composed by bits 8 and 7 of the network input. For another discriminator containing a RAM node whose IALs would be composed by the aforementioned bits, the position of this RAM which were addressed by "11" would be an equivalent memory position to the one initially mentioned.

Equivalent memory positions can indicate if there are features that can be used to distinguish between categories, because they are frequent or in some cases exclusive to a category. Such features appear as high values on certain RAM positions of the discriminator assigned to the category for which the feature is frequent or exclusive, i.e., it was accessed several times during the training phase, and as low values on the equivalent memory positions on the remaining discriminators. Mapping back the memory positions to the network input array reveals which data is related for these features, although some features may also be a consequence of noise in the database. Algorithm 1 is used for the attribute selection (lines 6-14) and rule extraction (lines 15-19).

This procedure differs from C4.5 algorithm [7], a more traditional approach to rule induction, which collects the rules first and then applies pruning to avoid overfitting. By using DRASiW, the pruning is done before, using the bleaching method. This simplifies the following task, as there are less noise to induce from.

4 Experiments

4.1 Dataset and preprocessing

The dataset used to demonstrate the ability to extract rules from a WiSARD network, by using the DRASiW extension, is the well-known *Iris* benchmark [3, 4]. It contains 3 categories (*I. setosa*, *I. versicolor* and *I. virginica*). The base

ESANN 2014 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 23-25 April 2014, i6doc.com publ., ISBN 978-287419095-7.
Available from http://www.i6doc.com/fr/livre/?GCOI=28001100432440.

Algorithm 1 Rule extraction algorithm.

```
1: uncovered_classes ← all_classes
2: uncovered_ial_input ← get_ial_combinations(ial, ram_inputs)
3: rules_root ← new tree()
4: rule ← @rules_root
5: while ( uncovered_classes not empty) do
6:     if ( uncovered_ial_input.end() ) then
7:         if ( no new rule generated in this epoch ) then break end if
8:         (ial, input) ← uncovered_ial_input.first()
9:     else
10:        (ial, input) ← uncovered_ial_input.next()
11:    end if
12:    equiv_positions ← get_equiv_positions(ial, input)
13:    new_equiv_positions ← remove_positions_if_class_not_in_list(equiv_positions, uncovered_classes)
14:    if ( get_entropy(new_equiv_positions) < threshold ) then
15:        for ( i from 1 to max_position.ial.length ) do
16:            rule_i ← new rule(get_attribute(ial_i))
17:            rule_i.separating_value ← get_separating_value(ial_i)
18:            rule.inequality ← ( ">" if input_i = 1 else "≤" )
19:            rule.and(rule_i)
20:        end for
21:        max_position ← get_position_by_value(max(value in equiv_positions))
22:        rule.child_yes ← get_class(get_discriminator(max_position))
23:        uncovered_classes.remove(get_class(get_discriminator(max_position)))
24:        uncovered_ial_input.remove(ial, input)
25:        if ( uncovered_classes.length = 1 ) then
26:            rule.child_no ← uncovered_classes.first()
27:            uncovered_classes.remove(uncovered_classes.first())
28:        else
29:            rule.child_no ← new rule()
30:            rule ← @rule.child_no
31:        end if
32:    end if
33: end while
```

has 50 observations of each category, totalling 150 observations. A 10-fold cross-validation procedure was done in the search of the best network configuration.

Each observation of the dataset is composed of four arguments (real numbers): sepal length, sepal width, petal length and petal width. As these parameters are not Boolean, an encoding is needed in order to transform each of these into a bit array. An unary coding, also known as thermometer code, is used to encode these arguments. This coding consists in selecting both the minimum and the maximum value of a parameter and splitting this interval into k blocks of equal size, k being a value to be calibrated.

4.2 Network configuration

In order to be able to extract rules from a WiSARD network, it must be capable of classifying correctly and, therefore, have a high mean accuracy. Thus, some WiSARD configurations are tested using a hill climbing methodology. The parameters to be determined are the "number of bits per feature" and the "size of the RAM input". After the tests being performed, two candidates for best configuration were found, as can be seen in Table 1: (i) 5 bits per feature and RAMs with 1 bit in their input, which proved to be the most accurate network, and (ii) 3 bits per feature and RAMs with 1 bit in their input, which was almost

ESANN 2014 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 23-25 April 2014, i6doc.com publ., ISBN 978-287419095-7.
Available from http://www.i6doc.com/fr/livre/?GCOI=28001100432440.

as accurate as the previous configuration, but with a quite simpler architecture. Thus, the latter configuration was the one chosen for this work.

Table 1: Accuracy rate with distinct configurations (optimal one marked in **bold**)

# bits per feature	#IALs	Mean accuracy	Standard deviation
1	1	0.340	0.128
2	1	0.713	0.055
3	**1**	**0.960**	**0.047**
4	1	0.867	0.077
5	1	0.973	0.034
6	1	0.960	0.084
3	2	0.720	0.177
3	3	0.713	0.218

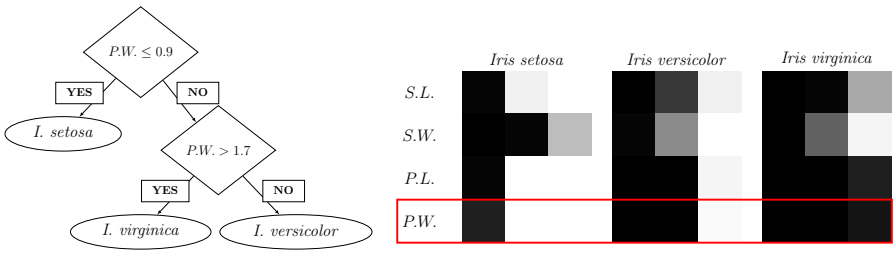
4.3 Rules extraction

To prove the correlation between the WiSARD network accuracy in the given machine learning task and the quality of the rules that can be inferred from the information stored in its RAM nodes, the rules extracted this way are applied to the whole dataset to verify how well they represent the data.

Table 2: Arguments and their extreme values in the optimal configuration

Argument	Minimum	Maximum	Step
Sepal length (*S.L.*)	4.3	7.9	1.20
Sepal width (*S.W.*)	2.0	4.4	0.80
Petal length (*P.L.*)	1.0	6.9	1.97
Petal width (*P.W.*)	0.1	2.5	0.80

The optimal configuration produced a minimal set of rules that are depicted as a tree in Figure 2a. This set of rules correctly represented 96% of the database. This setup uses RAMs with only one IAL and 3 bits per feature, which lead to a thermometer encoding according to the values of Table 2. For the sake of comparison and for a more illustrative depiction of how the tree works, the "mental images" of all *Iris* species are shown in Figure 2b.



(a) Rule tree extracted (b) "Mental images"

Fig. 2: Tree generated by DRASiW-based model and its "mental images"

At last, the rule extraction procedure is compared with a C4.5 decision tree algorithm [7], which produced the rules depicted in Figure 3. Both models

ESANN 2014 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 23-25 April 2014, i6doc.com publ., ISBN 978-287419095-7.
Available from http://www.i6doc.com/fr/livre/?GCOI=28001100432440.

performed equivalently, correctly classifying 96% of the dataset. Nonetheless, the rules produced by the DRASiW-based technique proved to be more compact.

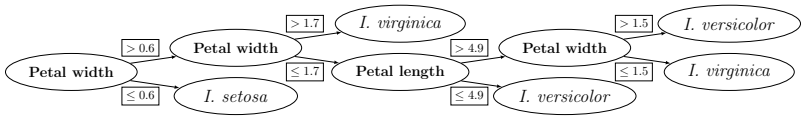


Fig. 3: Tree generated by C4.5 algorithm

5 Conclusions

It is possible to extract rules from a WiSARD neural network by means of DRASiW's "mental images". Experimental analysis showed that a simple network can derive a concise set of production rules. Moreover, these rules tend to be more compact than the ones obtained with the traditional C4.5 algorithm for decision trees. Finally, the use of a RAM-based model seems to be adequate in many other interesting scenarios, especially the ones involving limited computing resources, given its implementation easiness.

The use of the rule extraction procedure in problems which already use both DRASiW and the bleaching technique is left for further research. This kind of exploration is planned for problems in which some rules are already known [8] and in those that there are only a small set of known rules [9]. Furthermore, the analysis of the RAM-node content and the rules induced from it could help on the discovering of how encoding affects learning and how to optimize it.

References

[1] M. De Gregorio. On the reversibility of multi-discriminator systems. Technical Report 125/97, Istituto di Cibernetica–CNR, 1997.

[2] C. M. Soares, C. L. F. da Silva, M. De Gregorio, and F. M. G. França. Uma implementação em software do classificador WISARD (In Portuguese). In *V Simpósio Brasileiro de Redes Neurais*, volume 2, page 225–229, Belo Horizonte, MG, Brazil, December 1998.

[3] K. Bache and M. Lichman. The *Iris* Dataset. UCI Machine Learning Repository, 2013.

[4] R. A. Fisher. The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7:179–188, 1936.

[5] I. Aleksander, W. V. Thomas, and P. A. Bowden. WISARD: A radical step foward in image recognition. *Sensor Review*, 4:120–124, July 1984.

[6] D. S. Carvalho, H. C. C. Carneiro, F. M. G. França, and P. M. V. Lima. B-bleaching: Agile overtraining avoidance in the wisard weightless neural classifier. In *Proceedings of ESANN 2013*. i6doc.com, April 2013.

[7] J. R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, 1 edition, 1992.

[8] H. C. C. Carneiro, F. M. G. França, and P. M. V. Lima. WANN-Tagger: A weightless artificial neural network tagger for the Portuguese language. In *Proceedings of ICFC-ICNC 2010*, 2010.

[9] C. R. Souza, F. F. Nobre, P. M. V. Lima, R. M. Silva, R. M. Brindeiro, and F. M. G. França. Recognition of HIV-1 subtypes and antiretroviral drug resistance using weightless neural networks. In *Proceedings of ESANN 2012*, page 429–434. i6doc.com, April 2012.