ESANN 2016 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 27-29 April 2016, i6doc.com publ., ISBN 978-287587027-8.
Available from http://www.i6doc.com/en/.

The WiSARD Classifier

Massimo De Gregorio[1] and Maurizio Giordano[2]

1 - Istituto di Scienze Applicate e Sistemi Intelligenti "E. Caianiello" (ISASI – CNR)
2 - Istituto di Calcolo e Reti ad Alte Prestazioni (ICAR – CNR)

Abstract. WiSARD is a weightless neural model which essentially uses look up tables to store the function computed by each neuron rather than storing it in weights of neuron connections. Although WiSARD was originally conceived as a pattern recognition device mainly focusing on image processing, in this work we show how it is possible to build a multi–class classifier method in Machine Learning (ML) domain based on WiSARD that shows equivalent performances to ML state–of-the–art methods.

1 Introduction

Mimicking biological neurons by focusing on the excitatory/inhibitory decoding performed by the dendritic trees is a different and attractive alternative to the integrate–and–fire McCullogh–Pitts neuron stylisation [1]. In such alternative analogy, neurons can be seen as a set of RAM nodes addressed by Boolean inputs and producing Boolean outputs. The shortening of the semantic gap between the synaptic–centric model introduced by the McCullogh–Pitts neuron and the dominating, binary digital, computational environment, is among the interesting benefits of the weightless neural approach.

WiSARD [2] is a RAM–based neural network model developed by Igor Aleksander at Brunel University in the 1984. RAM–based neural networks essentially use look up tables to store the function computed by each neuron, and hence are easily implemented in digital hardware and have efficient training algorithms.

Although WiSARD was designed as a pattern recognition device mainly focusing on image processing domain, in this paper we show how it is possible to build a multiclass classification method in ML domain based on WiSARD computing model. As far as we know this is the first proposal of a general–purpose ML method that uses WiSARD as base technique for learning/classification.

The second contribution of this work is to show how the proposed WiSARD–based method for ML, that we called *WiSC* (***WiS***ARD ***C***lassifier), has, in the average, performances comparable to those of the most state–of–the–art ML methods found in literature. This statement is validated by the statistical analysis carried out on a set of experiments consisting in running a set of ML classifiers, including *WiSC*, on datasets publicly available on the KEEL archive.

2 WiSARD in numeric and symbolic domain

WiSARD (Wilkes, Stonham and Aleksander Recognition Device) was the first artificial neural network machine to be patented and produced commercially [2]. WiSARD is composed of a set of classifiers, called *discriminators*, each

ESANN 2016 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 27-29 April 2016, i6doc.com publ., ISBN 978-287587027-8. Available from http://www.i6doc.com/en/.

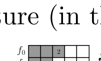
one assigned to learn binary patterns belonging to a particular class. Each discriminator consists of a set of RAM neurons. Each RAM has a number of input entries given by the binary address formed by its corresponding input subpattern. In training mode, an addressed pattern is stored in a RAM position as an integer value different from zero; non–addressed entries remain zero. In classification mode, each discriminator outputs the number of addressed RAM positions, for which the address was energized in training mode. Given a binary pattern of size S, the so–called *retina*, it can be classified by a set of WiSARD discriminators, each one having m RAMs with 2^n cells such that $S = m \times n$. For a more accurate description of WiSARD and other WNN please refer to [3][4][5].

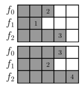
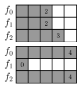
Being WiSARD a pattern recognition device, it mainly accepts black and white images as input. With *ad hoc* data transformation, WiSARD can be also successfully used as multiclass classifier in ML domain. Indeed, if we consider numeric data domains in which each datum can be represented by a vector of features (attributes), we can adopt the well–known LibSVM [6] or CSV format to represent numeric data such that each datum (sample) of a training set can be represented in the form: $s = \langle c_i,\ f_0 : v_0,\ \ldots,\ f_j : v_j \rangle$; where c_i is the class identifier (a string or a number) the datum belongs to, f_j and v_j are respectively a feature identifier (a string or a number), and its value (a real number, an integer or a nominal) inside the feature vector representing the datum.

In order to feed WiSARD with such data, they need to be converted to binary patterns. First of all, feature values v in the numeric range $[v_{min}, v_{max}]$ have to be discretized and scaled to integers $\underline{v}$ in the interval $[0, n]$. Doing so, any real number $v \in [v_{min}, v_{max}]$ will be represented by the non–negative integer:

$$\underline{v} = \left\lceil \frac{(v - v_{min}) \times n}{v_{max} - v_{min}} \right\rceil . \tag{1}$$

Thus, under the transformation of Equation 1, the dataset sample format becomes: $sample = \langle c_i,\ f_0 : \underline{v}_0,\ \ldots,\ f_j : \underline{v}_j \rangle$; where $\underline{v}_0, \underline{v}_1, \ldots, \underline{v}_j$ are non–negative integers in the range $[0, n]$. For example, let us consider a training set of class c, called TS_c, composed of 4 samples ($|TS_c|$=4). The scaled and discretized feature (in the range $[0, 4]$) will make the new samples of TS_c looking as:

 $s_0 = \langle 1,\ f_0 : 2,\ f_1 : 1,\ f_2 : 3 \rangle$,  $s_1 = \langle 1,\ f_0 : 2,\ f_1 : 2,\ f_2 : 3 \rangle$,

 $s_2 = \langle 1,\ f_0 : 3,\ f_1 : 2,\ f_2 : 4 \rangle$,  $s_3 = \langle 1,\ f_0 : 4,\ f_1 : 0,\ f_2 : 4 \rangle$.

Samples can be represented by binary patterns through the *thermometer encoding*, that guarantees close values of $\underline{v}_j$ will correspond to binary patterns with small Hamming distance (see left pictures of samples). With the transformation of Equation 1, numeric datasets can be now used both for training and classification in WiSARD systems. With *ad hoc* transformations [7], WiSARD can also treat symbolic data, like nominal (also called *categorical*) and ordinal datatypes.

The ML method proposed in this work, called *WiSC*, exploits learning/classification capabilities of WiSARD with the support of the above data transformations for numeric/symbolic data processing. *WiSC* was developed as part of the `sklearn` library[1] and it is compliant to its programming interface.

[1] `http://scikit-learn.org`

ESANN 2016 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 27-29 April 2016, i6doc.com publ., ISBN 978-287587027-8.
Available from http://www.i6doc.com/en/.

(a)

Acronym	Name
	Artificial Neural Networks
MLP	Multi Layer Perceptron
WiSC	WiSARD Classifier
	Base Learners
LDA	Linear Discriminant Analysis
LR	Logistic Regression
kNN	*k*–Nearest Neighbors
SVC	SVM with RBF kernel
	Ensemble Learners
GTB	Gradient Tree Boosting
RF	Random Forrest
ERT	Extremely Randomized Trees

(b)

Name	Features			Instances	Classes
	Real	Int.	Nom.		
ecoli	7	–	–	336	8
pima	8	–	–	268	2
segment	19	–	–	2310	7
wine	13	–	–	178	3
dermatology	–	34	–	358	6
penbased	–	16	–	10992	10
vehicle	–	18	–	846	4
wisconsin	–	9	–	683	2
breast	–	–	9	263	2
splice	–	–	60	3175	3
heart	1	12	–	270	2
ionosphere	32	1	–	351	2
vowel	10	3	–	990	11
german	–	7	13	1000	2
lymphography	–	3	15	148	4
automobile	15	–	10	150	6
australian	3	5	6	690	2
crx	3	3	9	653	2

Table 1: Methods (a) and datasets (b) used in *WiSC* performance evaluation

3 Related Works

The generality of WNN systems allowed them to be used in past years also as multi–discriminator classifiers in several ML domains, ranging from text cathegorisation [8] to HIV–1 subtypes classification [9], from language POS–tagging [7] to WiFi signal pattern recognition [10]. All the cited works are specific applications of WNNs to a problem domain, and the proposed WNN systems are tuned and configured to optimally behave in the target ML domain.

On the contrary, *WiSC* is a domain–independent classification method for ML, in the sense that it can apply to every problem with no *ad hoc* configuration. *WiSC* is configured by two parameters: 1) bit–addressing resolution (b), and 2) data scaling factor (n). By running under the default configuration (b=16, n=1024) *WiSC* can perform as classifier for any dataset, provided that data are available in the LibSVM or CSV format.

Apart from *MLP*, the other algorithms considered in this work (see Table 1a) do not fall in the category of neural network algorithms. Since those algorithms exploit approaches different from neural models (statistical, clustering, decision trees, ensembling, etc.), the comparisons between *WiSC* and those classifiers could be made only in terms of performance. In what follows, we provide a set of ML methods with their categorization and references in the ML literatures.

The Logistic Regression [11] (*LR*) falls in the category of generalized linear models. The Support Vector Classifier [12] (*SVC*) is the only considered method based on SVM and it is applied with a non–linear kernel (RBF). Methods like Random Forests [13] (*RF*), Extremely Randomized Trees [14] (*ERT*), and Gradient Tree Boosting [15] (*GTB*) all fall in the category of decision–tree learning algorithms. *GTB* is a boosting methods that combine decisions–tree classifiers. The list of reference methods ends up with Linear Discriminant Analysis (*LDA*) and k–Nearest Neighbors (*kNN*) that are unique representatives of the corresponding learning method categories: Linear and Quadratic Discriminant Analysis [16] and Nearest Neighbor [17] algorithms.

To better refine the categorization of methods, *LDA*, *LR*, and *kNN* are base learning methods while the rest are ensemble methods [18]: in particular, *RF* and *ERT* are bagging meta–estimators based on decisions–tree classifiers.

ESANN 2016 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 27-29 April 2016, i6doc.com publ., ISBN 978-287587027-8.
Available from http://www.i6doc.com/en/.

	WiSC	*MLP*	*RF*	*ERT*	*GTB*	*LDA*	*kNN*	*LR*	*SVC*
australian	0.85015	0.86261	0.84638	0.84493	0.86435	0.86203	0.65275	0.85623	*0.54667*
automobile	0.81000	0.61750	0.80125	0.84125	0.88125	0.74750	0.39125	0.70625	*0.28375*
breast	0.71407	0.74222	0.70667	*0.69926*	0.71185	0.74963	0.73704	0.74815	0.75926
crx	0.85000	0.86970	0.85818	0.84909	0.87061	0.86576	0.65758	0.87242	*0.54818*
dermatology	0.97889	0.96611	0.96000	0.97222	0.95167	0.96167	*0.86944*	0.97333	0.92444
ecoli	0.85471	0.71000	0.82412	0.83529	0.81176	0.77706	0.82941	0.79941	*0.43177*
german	0.74420	0.76320	0.73600	0.72840	0.76360	0.77100	*0.69200*	0.76920	0.71980
heart	0.83259	0.84074	0.79111	0.79852	0.79852	0.83926	0.68444	0.83852	*0.54444*
ionosphere	0.92889	0.85833	0.93278	0.93778	0.94556	0.87111	*0.85222*	0.87889	0.93778
lymphography	0.81733	0.80533	0.79067	0.80800	0.82000	0.81333	*0.75333*	0.79200	0.77600
penbased	0.99213	0.92131	0.98731	0.99062	0.98967	0.87518	0.99318	0.92696	*0.10364*
pima	0.76390	0.75429	0.73377	0.75740	0.75740	0.72467	0.77792	*0.64208*	*0.64208*
segment	0.97628	0.88329	0.97584	0.97584	0.98260	0.91671	0.94442	0.92840	*0.62719*
splice	0.94094	0.84233	0.93981	0.93340	0.97120	0.84516	*0.78157*	0.84837	0.91516
vehicle	0.75294	0.63365	0.75294	0.61365	*0.46588*	0.71271	0.74777	0.72988	0.72988
vowel	0.98808	*0.45172*	0.92283	0.95919	0.89475	0.61232	0.93374	0.53434	0.86970
wine	0.99000	0.97000	0.97444	0.96889	0.96111	0.99222	0.70556	0.95667	*0.42667*
wisconsin	0.97217	0.96522	0.96696	0.96783	0.96783	*0.96029*	0.97275	0.96435	0.96087

Table 2: Average accuracy on 50 repetitions of a 10–fold cross–validation

Method	**Friedman**	**Aligned Friedman**	**Quade**
WiSC	2.917 (1)	53.806 (1)	2.664 (1)
GTB	3.694 (2)	60.861 (2)	3.661 (2)
ERT	4.527 (3)	65.972 (4)	4.140 (3)
RF	4.889 (4)	64.944 (3)	4.474 (4)
LDA	5.056 (5)	85.000 (5)	5.205 (5)
LR	5.111 (6)	83.389 (6)	5.374 (6)
MLP	5.556 (7)	95.167 (8)	5.842 (8)
kNN	5.944 (8)	102.611 (9)	5.713 (7)
SVC	7.1154 (9)	86.8846 (7)	7.927 (9)
Distribution	χ^2	χ^2	F-distribution
Degrees of fredom	8	8	with 8 and 136
Statistic	30.748	15.468	5.196
p-value	1.558×10^{-4}	0.050	1.128×10^{-5}

Table 3: Multiple comparison of method performances by nonparametric tests

4 Performance Evaluation Through Statistical Analysis

In order to evaluate the performance of *WiSC* we choose to test it on a set of N (N=18) classification problems listed on Table 1b. All problems were selected from a list of 76 standard classification datasets available on the KEEL Archive.[2]

The aim of the experimental study is to compare accuracy of *WiSC* with that of other classification methods (see Table 1a) available in the `sklearn` library. We measured classification accuracy of each method on the N chosen datasets, that is the average of accuracies over 50 repetitions of a ten–fold cross validation on each dataset. Random splits of each dataset were prepared to form 50 pairs of train and test sets. All methods worked on the same 50 pairs of sample sets.

In all experiments no features selection was carried out, as well as the default configuration was used for all methods independently on the target problem domain, size and feature type combination. This fair assumption is due, one side, to the difficult and time–consuming task of finding the optimal configuration for each method when applied to a specific problem, and, on the other side, to the intention of testing each method "as it is" regardless of the specific problem. The average accuracy over 50 experiments for each method running on each dataset is reported in Table 2. The best average accuracies across all methods are underlined, while the worst performances are in italic.

Methods performances are evaluated by multiple comparison nonparametric tests: the Friedman test [19], the Aligned Friedman test [20], and the Quade

[2]Available at `http://sci2s.ugr.es/keel`

ESANN 2016 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 27-29 April 2016, i6doc.com publ., ISBN 978-287587027-8.
Available from http://www.i6doc.com/en/.

WiSC versus	*GTB*	*ERT*	*RF*	*LDA*	*LR*	*MLP*	*kNN*	*SVC*	$\alpha^\dagger$
unadjusted p	$3.94{\cdot}10^{-1}$	$7.76{\cdot}10^{-2}$	$3.07{\cdot}10^{-2}$	$1.91{\cdot}10^{-2}$	$1.62{\cdot}10^{-2}$	$3.84{\cdot}10^{-3}$	$9.11{\cdot}10^{-3}$	$1.53{\cdot}10^{-6}$	0.05
z	$8.52{\cdot}10^{-1}$	1.76	2.16	2.34	2.40	2.89	3.32	4.81	
p_{Holm}	$4.\overline{54}{\cdot}10^{-3}$	$2.\overline{27}{\cdot}10^{-3}$	$2.0{\cdot}10^{-3}$	$1.92{\cdot}10^{-3}$	$1.79{\cdot}10^{-3}$	$1.56{\cdot}10^{-3}$	$1.47{\cdot}10^{-3}$	$1.3\overline{8}{\cdot}10^{-3}$	$1.\overline{51}{\cdot}10^{-3}$
$p_{Shaffer}$	$4.\overline{54}{\cdot}10^{-3}$	$2.\overline{27}{\cdot}10^{-3}$	$2.0{\cdot}10^{-3}$	$1.92{\cdot}10^{-3}$	$1.79{\cdot}10^{-3}$	$1.79{\cdot}10^{-3}$	$1.79{\cdot}10^{-3}$	$1.3\overline{8}{\cdot}10^{-3}$	$1.79{\cdot}10^{-3}$

Table 4: Pairwise comparisons between *WiSC* and the rest of methods

test [21]. Before starting the statistical analysis we define the null hypothesis:

$$\boxed{H_0 = \textit{accuracy distributions over N datasets of all methods are the same.}}$$

In Table 3 methods' rankings according to the three statistic tests are reported. Methods are ordered according to the Friedman test ranking. In each column the rank is reported with the relative position. As one can notice, *WiSC* method ranks first in all tests. By comparing the p–values of the three statistics with the significance level α (0.05) we can assert that H_0 is rejected by all tests. Then, we can proceed with post–hoc tests to carry out all possible pairwise comparisons of methods ($N{\times}N$ comparison). In particular, we test the set of hypotheses:

$$\boxed{H_{X,Y} = \textit{accuracy distributions over N datasets of method X and Y are the same.}}$$

Two classic procedures used for the purpose are the Holm [22] and the Shaffer [23] tests. These tests adjust the significant level α (0.05) to a new reference value $\alpha^\dagger$. Both tests are used to compute the p–values of each pairwise comparison of methods. In Table 4 comparison results of *WiSC* with the rest of algorithms are reported: gray cells represent rejected hypotheses resulting from the comparison of the p–value, as computed by the test, with respect to the corresponding $\alpha^\dagger$.

The comparison analysis of Table 4 proves that both Holm and Shaffer test reject the hypotheses $H_{WiSC,kNN}$ and $H_{WiSC,SVC}$, while the Shaffer test rejects even $H_{WiSC,LR}$ and $H_{WiSC,MLP}$. Therefore, by considering the null hypotheses which are not rejected by both tests, we can statistically assert likely the equivalence of *WiSC* to methods in the set $\{GTB, ERT, RF, LDA\}$ in term of accuracy performance over the chose N datasets. By considering the magnitude of the significant value (p–value), we deduce that *WiSC* is "more significantly" equivalent, in terms of performance, to *GTB*. This result is even more relevant if we consider that *GTB* is an *ensemble learning* method, while *WiSC* is a *base learning* method. The statistical analysis assigns the best ranking to *WiSC*, in terms of average accuracy on the chosen datasets (see Table 3), as well as it proves that *WiSC* outperforms other base learners (the set $\{LR, kNN, SVC, LDA\}$).

5 Concluding Remarks

In this work a WiSARD–based classifier for ML has been proposed, namely *WiSC*. When tested on a large dataset archive, *WiSC* proved to be equivalent to some of the most performant ensemble learning techniques in the ML state–of–the–art, such as Gradient Tree Boosting, Radom Forrest, Extra Randomized Trees. Although *WiSC* performs better than weighted ANNs counterpart methods, it still have processing times greater than the equivalent methods. Just to

ESANN 2016 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 27-29 April 2016, i6doc.com publ., ISBN 978-287587027-8. Available from http://www.i6doc.com/en/.

evaluate an order of magnitude of timing, *WiSC* runs three times slower than *GTB* and five times slower than *ERT* and *RF*. We will further investigate optimization techniques to improve *WiSC* performance, both in terms of RAM neuron memorization strategy as well as data input encoding.

References

[1] W. McCulloch and W. Pitts. A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, 7:115–133, 1943.

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

[3] Aleksander I. Morton H. *An introduction to neural computing*. Chapman & Hall, 1990.

[4] T. B. Ludermir, A. C. Carvalho, A. P. Braga, and M. C. P. Souto. Weightless neural models: a review of current and past works. *Neural Computing Surveys*, 2:41–61, 1999.

[5] I. Aleksander, M. De Gregorio, F. M. G. França, P. M. V. Lima, and H. Morton. A brief introduction to weightless neural systems. In 17^{st} *ESANN*, pages 299–305, 2009.

[6] Chih–Chung Chang and Chih–Jen Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011.

[7] Hugo C. C. Carneiro, Felipe M. G. França, and Priscila M. V. Lima. WANN-TAGGER - a weightless artificial neural network tagger for the portuguese language:. In *IJCCI (ICFC-ICNC)*, pages 330–335. SciTePress - Science and Technology Publications, 2010.

[8] C. Badue, F. Pedroni, and A. Souza. Multi-label text categorization using vg-ram weightless neural networks. In *Neural Networks, 2008. SBRN '08.*, pages 105–110, Oct 2008.

[9] C. Souza, F. Nobre, P.M.V. Lima, R. Silva, R. Brindeiro, and F.M.G. França. Recognition of hiv-1 subtypes and antiretroviral drug resistance using weightless neural networks. In *ESANN'12*, pages 429–434, 2012.

[10] D.O. Cardoso, J. Gama, M. De Gregorio, and M. Giordano F.M.G. França. Wips: the wisard indoor positioning system. In *ESANN'12*, pages 521–526, 2012.

[11] Alan Agresti. *Categorical Data Analysis*. Wiley-Interscience, 2002.

[12] C. Cortes and V. Vapnik. Support-vector networks. *Mach. Lear.*, 20(3):273–297, 1995.

[13] Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

[14] P. Geurts, D. Ernst, and L. Wehenkel. Extremely randomized trees. *Mach. Learn.*, 63(1):3–42, 2006.

[15] Jerome H. Friedman. Greedy function approximation: A gradient boosting machine. *The Annals of Statistics*, 29(5):pp. 1189–1232, 2001.

[16] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. 2009.

[17] T. Darrell G. Shakhnarovich and P. Indyk. *Nearest-Neighbor Methods in Learning and Vision*. MIT Press, 2006.

[18] Zhi-Hua Zhou. *Ensemble Methods: Foundations and Algorithms*. Chapman & Hall/CRC, 1^{st} edition, 2012.

[19] M. Friedman. The use of ranks to avoid the assumption of normality implicit in the analysis of variance. *J. of the American Statistical Ass.*, 32(200):pp. 675–701, 1937.

[20] Jr. Hodges, J.L. and E.L. Lehmann. Rank methods for combination of independent experiments in analysis of variance. In Javier Rojo, editor, *Selected Works of E. L. Lehmann*, Selected Works in Probability and Statistics, pages 403–418. Springer, 2012.

[21] Dana Quade. Using weighted rankings in the analysis of complete blocks with additive block effects. *Journal of the American Statistical Association*, 74(367):pp. 680–683, 1979.

[22] S. Holm. A simple sequentially rejective multiple test procedure. *Scandinavian Journal of Statistics*, 6:65–70, 1979.

[23] Juliet Popper Shaffer. Modified sequentially rejective multiple test procedures. *Journal of the American Statistical Association*, 81(395):826–831, 1986.