ESANN 2017 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 26-28 April 2017, i6doc.com publ., ISBN 978-287587039-1. Available from http://www.i6doc.com/en/.

ELM vs. WiSARD: a performance comparison

Luiz Fernando R. Oliveira and Felipe M. G. França *

Systems Engineering and Computer Science Program, COPPE
Universidade Federal do Rio de Janeiro, RJ, Brazil

Abstract. The extreme learning machine (ELM) is known for being a fast learning neural model. This work presents a performance comparison between ELM and the WiSARD weightless neural network model, regarding training and testing times, and classification accuracy as well. The two models were implemented in the same programming language and experiments were carried out on the same hardware environment. By using a group of datasets from the public repositories UCI and Statlog, experimental results shows that the WiSARD presented training times approximately one order of magnitude smaller than ELM, while classification accuracy varied according the number of classes involved. However, while WiSARD's architecture setups were not exhaustively searched, architecture setups for ELM were kept the same as the ones found in the literature as the best for each given dataset.

1 Introduction

There has been great interest on the improvement of more recent and sophisticated techniques, regarding both learning theory and implementation aspects. A typical example is Deep Learning, which has gathered large attention due to GPU computing advances, in order to deal with the reduction of it's huge training time. However, training time is still an issue when trying to apply machine learning into online situations. Two neural models oriented to high performance training are **Wi**lkes, **S**tonhan and **A**leksander **R**ecognition **D**evice (WiSARD) and the extreme learning machine (ELM). The first is a Weightless neural network model that uses RAM-based neurons and a set of discriminators with pseudo-random input mapping, while the second is a single layer feedfoward network in which the hidden layer do not need to be iteractively adjusted.

These two models share interesting properties like low training time and ease of implementation, while having some particularities, such as WiSARD being able to be implemented directly in Boolean logic hardware and ELM in a single-step iteration. Motivated by these properties, this work proposes a comparative study of these two techniques regarding training time, testing time and average classification accuracy when applied to several datasets having different characteristics.

*The authors would like to thank CAPES - Ministry of Education, Brazil, CNPq - Ministry of Science and Technology, Brazil - grant number 307437/2013-2, FINEP - Ministry of Science and Technology, Brazil.

2 Extreme Learning Machines

Classical neural network models, such as multi-layer perceptrons, are usually formed by an input layer, a set of hidden layers and an output layer. The hidden layers contains weights that need to be optimized by an algorithm like backpropagation. ELMs intend to be fast trainable by using just a single hidden layer with random nodes that do not need to be tuned. This is done by generating a random feature mapping matrix $H_{N \times L}$, where N is the number of data samples and L is the number of neurons. H is defined as:

$$H = \begin{bmatrix} G(a_1, b_1, x_1) & \cdots & G(a_L, b_L, x_1) \\ \vdots & \ddots & \vdots \\ G(a_1, b_1, x_N) & \cdots & G(a_L, b_L, x_N) \end{bmatrix},$$

where $a_i \in R^d$ and $b_i \in R^+$ are random parameters, x_i is a sample of dataset $X = (x_i, t_i)|x_i \in R^d, t_i \in R^m, i = 1, ..., N$, where m is the number of attributes involved, and $G(a_i, b_i, x)$ is the output function of the ith hidden node.

As presented in [4], the basic ELM algorithm consists in randomly generating the hidden nodes parameters (a_i, b_i), create the hidden output matrix H and calculate the output weight vector β by solving $\beta = H^\dagger T$, where $H^\dagger$ is the *Moore-Penrose generalized inverse.* Usually, the orthogonal projection method is preferable to generate this inverse, and is defined as $H^\dagger = (H^T H)^{-1} H^T$ if $H^T H$ is nonsingular or $H^\dagger = H^T (H H^T)^{-1}$ if $H H^T$ is nonsingular.

It is also suggest to add a positive value $1/C$ to the diagonal of the matrix that is being inverted in order to improve the generalization performance, which results in the modification of the algorithm so that $\beta = (\frac{I}{C} H^T H)^{-1} H^T T$ or $\beta = H^T (\frac{I}{C} H H^T)^{-1} T$

3 The WiSARD weightless neural network

Proposed by Wilkes, Stonhan and Aleksander in 1984 [8], WiSARD is a weightless neural network model which aims at recognizing patterns represented as binary data. It's a multidiscriminator model where each discriminator is in charge of recognizing a specific class of patterns. A discriminator is formed by a set of m RAM-like structures based on a pseudo-random mapping of the input data field [3] [7].

Figure 1 shows how a discriminator is trained. Initially, all RAM structures stores 0 in all of its it's addressable contents. Given a binary input, m groups of n bits are randomly defined. The combination of the values in each element creates a memory address, and the value 1 is then stored at the address of the corresponding RAM. [6]

Figure 2 shows how a pattern is classified. A pattern is presented to all discriminators and, for each one, the addresses bits are selected the same way as the training by using the discriminator pseudo-random mapping. Each RAM

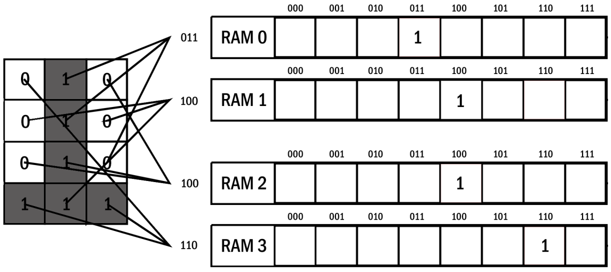

Fig. 1: Training a WiSARD discriminator

is verified using the created addresses and, if the value stored if different from 0, the RAM is said to be "activated".

At the end of the verification, each discriminator returns a value r, related to it's activation response. The discriminator with highest response implies in the classification value of the presented pattern. Figure 2 also show that the discriminator related to class **T** presents the highest response, meaning that the input pattern was classified as being a **T**.

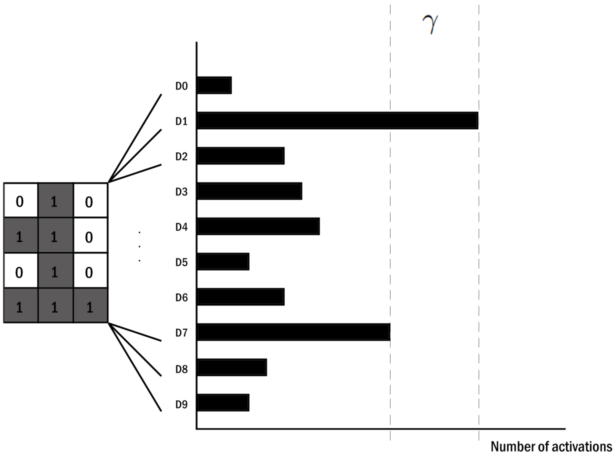

Fig. 2: Classifying a pattern using WiSARD.

4 Experimental Framework

4.1 Datasets

To compare both models, a subset of the datasets used in [5] was selected. Most of these datasets were obtained from the UCI public repository [1] and present very different characteristics, regarding number of samples, number of attributes, as well as type of attributes. A total of 14 datasets were used, six of them being binary classification datasets, while eight other are multiclass problems. Some of the datasets were already divided into training and testing sets, while others consist on a single data file. For the latest, two different methodologies were used. The first is a 10-fold cross validation and the second is the same used in [5]: 2/3 of data for training and 1/3 for testing in combination with a random permutation of data. Following the setup defined in [5], data was normalized between [-1, 1] when exercising the ELM model.

4.2 Architectural parameters

For the ELM model, it was used the *sigmoid additive node*, while the number of neurons and the regularization factor C were the ones specified in [5] that optimizes the model stabilization. These parameters were obtained by the author by exploring the combination space. As such exploration is yet to be done in WiSARD, it was decided that the binarization of the data values would be done by using a 20 bit thermometer representation for each feature of the sample. Thus, the configuration would result in a $m \times 20$ retina, and a tuple of size $n = 20$ was used in all of the experiments.

4.3 Environment

Both models were implemented using the Python 2.7 language on a Intel Core i3 2.7GHz CPU with 16GB of RAM memory and Ubuntu Linux 14.04 operating system.

5 Results and Discussion

Table 1 shows the experimental results for both binary and multiclass datasets. The table shows the test accuracy using both methodologies, as well as training and testing time performance of both models. All the results were obtained through the mean value after 20 independent runs. The standard deviation of both accuracies were significantly small, varying within the range (0.009, 0.054).

By observing the testing time of WiSARD on some of the datasets, it is possible to observe that some of them are far slower than the one of ELM. Two factors can be pointed out: (i) the first is related to the number of discriminators; the greater the number of classes a dataset has, greater number of comparisons the WiSARD needs to perform; (ii) the second, which also relates to the first, is related to the network's architecture; If the input field is big and tuples are

Table 1: Binary (1-6) and multiclass (7-14) datasets; Accuracy 1 (10-fold cross validation); Accuracy 2 ([5]); best values in boldface.

Dataset	**Model**	**Accuracy 1**	**Accuracy 2**	**Training**	**Testing**
1-Adult	ELM	**0.811**	**0,802**	8,160 s	**1,542** s
	WiSARD	0.802	0,773	**0,329** s	170,911 s
2-Australian	ELM	**0.778**	**0,739**	0,817 s	**0,079** s
	WiSARD	0.715	0,687	**0,246** s	1,259 s
3-Banana	ELM	**0.876**	**0,877**	0,682 s	1,743 s
	WiSARD	0.856	0,848	**0,029** s	**1,070** s
4-Diabetes	ELM	**0.778**	**0,765**	0,888 s	**0,088** s
	WiSARD	0.677	0,706	**0,366** s	0,574 s
5-Liver	ELM	**0.687**	**0,722**	0,849 s	**0,042** s
	WiSARD	0.554	0,570	**0,125** s	0,354 s
6-Mushroom	ELM	0.463	**0,859**	1,379 s	**2,105** s
	WiSARD	**0.855**	0,850	**0,298** s	10,013 s
7-Ecoli	ELM	0.766	**0,876**	0,781 s	**0,040** s
	WiSARD	**0.805**	0,811	**0,143** s	0,755 s
8-Glass	ELM	0.805	**0,899**	0,717 s	**0,027** s
	WiSARD	**0.854**	0,875	**0,116** s	0,590 s
9-Iris	ELM	**0.966**	**0,974**	0,618 s	**0,018** s
	WiSARD	0.943	0,935	**0,038** s	0,070 s
10-Letter	ELM	0.574	**0,932**	6,205 s	**1,009** s
	WiSARD	**0.819**	0,808	**1,339** s	18,959 s
11-Satimage	ELM	0.573	**0,882**	2,567 s	**0,660** s
	WiSARD	**0.854**	0,848	**0,950** s	2,640 s
12-Segment	ELM	**0.938**	**0,962**	1,346 s	**0,275** s
	WiSARD	0.923	0,933	**0,284** s	0,651 s
13-Vehicle	ELM	0.683	**0,828**	0,792 s	**0,102** s
	WiSARD	**0.696**	0,674	**0,634** s	0,480 s
14-Wine	ELM	0.901	**0,980**	0,791 s	**0,021** s
	WiSARD	**0.972**	0,952	**0,128** s	0,210 s

small, more RAM structures will be created, raising even more the number of comparisons.

6 Conclusion

This work presented a comparison between two classification models that are characterized by fast training. Interestingly, ELM turned out to provide a higher accuracy on all of the datasets using the second methodology, which used the parameters presented in [5], although WiSARD could perform in a similar way on the majority of them. Meanwhile, the 10-fold crossvalidation presented a variation on accuracies, which reinforces the argument that the best architectural parameters of the models can drastically affect the accuracy performance. Nevertheless, as stated before, WiSARD's best configurations are yet to be explored.

Regarding training time, the WiSARD presented better training performance in all datasets. As for testing time, it should be stated that the WiSARD configuration implies directly on this component of the model. Although WiSARD presented a similar classification time in the majority of the datasets, there were a small group it was far slower than ELM. It is important to notice that, for bigger datasets, ELM can present a memory usage issue. It is stated in [5] that for the four largest datasets, a different computer with higher amount of RAM memory was used. But WiSARD has presented no issues regarding memory usage. Thus, it turns to be necessary to add this comparison in a future work. Finally, although ELM can also be applied to regression problems, it's yet to be discussed the applications of weightless neural network models in this kind of domain, which is another motivation for future studies.

References

[1] C. L. Blake and C. J. Merz, "UCI Repository of Machine Learning Databases," Dept. Inf. Comput. Sci., Univ. California, Irvine, CA, 1998.

[2] 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." ESANN (2013).

[3] F. M. G. França, M. De Gregorio, P. M. V. Lima, W. R. Oliveira Jr, "Advances in Weightless Neural Systems". Proc. of ESANN 2014. Brussels: i6doc.com, 2014. p. 497-504.

[4] G-B Huang, H. Wang, Y. Lan, Extreme learning machines: a survey. Int J Mach Learn Cybern 2(2):107–122, 2011

[5] G-B Huang, H. Zhou, X. Ding, and R. Zhang, "Extreme Learning Machine for Regression and Multiclass Classification". Trans. Sys. Man Cyber. Part B 42, 2 (April 2012), 513-529.

[6] H. L. França, J. C. P. Silva, M. De Gregorio, O. Lengerke, M. S. Dutra, F. M. G. França, "Movement Persuit Control of an Offshore Automated Platform via a RAM-based Neural Network". In: 11th. Int. Conf. Control, Automation, Robotics and Vision, 2010, Singapore.

[7] I. Aleksander, M. De Gregorio, F. M. G. França, P. M. V. Lima, H. Morton, "A Brief introduction to Weightless Neural Systems". Proc. of ESANN 2009. Evere, Belgium: d-side, 2009. p. 299-305.

[8] I. Aleksander, W. Thomas, and P. Bowden, "WiSARD: a radical step forward in image recognition," Sensor review, vol. 4, no. 3, pp. 120– 124, 1984.

[9] K. M. Khaki, "Weigthless Neural Networls for Face and Pattern Recognition: an Evaluation using open-source databases". PhD thesis, Brunel University, 2013.