Multi-Label Text Categorization using VG-RAM Weightless Neural Networks

Claudine Badue, Felipe Pedroni, and Alberto F. De Souza

Departamento de Informática, Universidade Federal do Espírito Santo, Brazil
claudine@lcad.inf.ufes.br, fpedroni@lcad.inf.ufes.br, alberto@lcad.inf.ufes.br

Abstract

*In automated multi-label text categorization, an automatic categorization system should output a category set, whose size is unknown a priori, for each document under analysis. Many machine learning techniques have been used for building such automatic text categorization systems. In this paper, we examine Virtual Generalizing Random Access Memory Weightless Neural Networks (*VG-RAM WNN*), an effective machine learning technique which offers simple implementation and fast training and test, as a tool for building automatic multi-label text categorization systems. We evaluate the performance of* VG-RAM WNN *on the categorization of Web pages, and compare our results with that of the multi-label lazy learning approach* ML-KNN*, the boosting-style algorithm* BOOSTEXTER*, the multi-label decision tree* ADTBOOST.MH*, and the multi-label kernel method* RANK-SVM*. Our experimental comparative analysis shows that, on average,* VG-RAM WNN *either outperforms the other mentioned techniques or show similar categorization performance.*

1. Introduction

Automatic text categorization is still a very challenging computational problem to the information retrieval communities both in academic and industrial contexts. Most works on text categorization in the literature are focused on single-label text categorization problems, in which exactly one category must be assigned to each document [12]. However, in real-world problems, multi-label categorization, in which any number of categories may be assigned to the same document, is frequently necessary [11, 5, 7, 6, 3, 13, 14].

From a theoretical point of view, the single-label case is more general than the multi-label case, since an algorithm for single-label categorization can also be used for multi-label categorization: one needs only to transform the multi-label categorization problem into n independent single-label problems, where n is number of possible categories, or labels [12]. However, this equivalence only holds if the n categories are stochastically independent, that is, the association of a category c_i to a document is independent of the association of another category, c_j, to the same document. However, this frequently is not the case. Fortunately, several approaches specially designed for multi-label categorization have been proposed, such as decision trees [5, 6], kernel methods [7, 3] or neural networks [10, 13], and many of them specifically for multi-label text categorization [7, 11, 6, 10, 13, 14].

In this paper, we present an experimental evaluation of the performance of virtual generalizing random access memory weightless neural networks (VG-RAM WNN [8]) on multi-label text categorization. VG-RAM WNN is an effective machine learning technique which offers simple implementation, and fast training and test [2]. We evaluate the performance of VG-RAM WNN on a real-world multi-label problem: the categorization of Web pages. Web page categorization is used by several Web search companies, such as Google and Yahoo, for helping users navigate the Internet, and has significant economic value. We analyze the performance of VG-RAM WNN using four multi-label categorization metrics: *hamming loss*, *one error*, *coverage*, and *average precision* [11]. We also compare the VG-RAM WNN performance, according to these metrics, with that of the multi-label lazy learning technique ML-KNN [14], the boosting-style algorithm BOOSTEXTER [11], the multi-label decision tree ADTBOOST.MH [6], and the multi-label kernel method RANK-SVM [7]. Our results show that, on average, VG-RAM WNN either outperforms these techniques or show similar categorization performance.

This paper is organized as follows. Section 2 introduces the multi-label text categorization problem and the metrics used to evaluate the performance of the multi-label categorizers examined. Section 3 briefly introduces VG-RAM WNN and describes how we have used it for multi-label text categorization. Section 4 presents our experimental methodology and analyzes our experimental results. Our conclusions and directions for future work follow in Section 5.

2. Multi-Label Text Categorization

Text categorization may be defined as the task of assigning categories (or labels), from a predefined set of categories, to documents [12]. In multi-label text categorization, one or more categories may be assigned to a document.

Let $\mathcal{D}$ be the domain of documents, $\mathcal{C} = \{c_1, \ldots, c_{|\mathcal{C}|}\}$ a set of pre-defined categories, and $\Omega = \{d_1, \ldots, d_{|\Omega|}\}$ an initial corpus of documents previously categorized manually by a domain expert into subsets of categories of $\mathcal{C}$. In multi-label learning, the training(-and-validation) set $TV = \{d_1, \ldots, d_{|TV|}\}$ is composed of a number documents, each associated with a subset of categories of $\mathcal{C}$. TV is used to train and validate (actually, to tune eventual parameters of) a categorization system that associates the appropriate combination of categories to the characteristics of each document in the TV. The test set $Te = \{d_{|TV|+1}, \ldots, d_{|\Omega|}\}$, on the other hand, consists of documents for which the categories are unknown to the categorization system. After being (tunned and) trained with TV, the categorization system is used to predict the set of categories of each document in Te.

A multi-label categorization system typically implements a real-valued function $f : \mathcal{D} \times \mathcal{C} \rightarrow \mathbb{R}$ that returns a value for each pair $\langle d_j, c_i \rangle \in \mathcal{D} \times \mathcal{C}$ that, roughly speaking, represents the evidence for the fact that the test document d_j should be categorized under the category c_i. The real-valued function $f(.,.)$ can be transformed into a ranking function $r(.,.)$, which is a one-to-one mapping onto $\{1, 2, \ldots, |\mathcal{C}|\}$ such that, if $f(d_j, c_1) > f(d_j, c_2)$, then $r(d_j, c_1) < r(d_j, c_2)$. If C_j is the set of proper categories for the test document d_j, then a successful categorization system will tend to rank categories in C_j higher than those not in C_j. Those categories that rank above a threshold τ (i.e., $c_k | f(d_j, c_k) \geq \tau$) are then assigned to the test document d_j.

We have used the four multi-label evaluation metrics discussed in [11] for examining the categorization performance of VG-RAM WNN, namely *hamming loss*, *one-error*, *coverage*, and *average precision*. While *hamming loss* evaluates the exact set of categories predicted for the test document d_j, that is, those categories that rank above the threshold τ, the metrics *one-error*, *coverage*, and *average precision* evaluate the real-valued function $f(.,.)$, that is, the ranking quality of different categories for each test document. We present each of these metrics below.

Hamming Loss (hloss$_j$) evaluates how many times the test document d_j is misclassified, i.e., a category not belonging to the document is predicted or a category belonging to the document is not predicted.

$$\text{hloss}_j = \frac{1}{|\mathcal{C}|} |P_j \Delta C_j| \tag{1}$$

where $|\mathcal{C}|$ is the number of categories and Δ is the symmetric difference between the set of predicted categories P_j and the set of appropriate categories C_j of the test document d_j.

One-error (one-error$_j$) evaluates if the top ranked category is present in the set of proper categories C_j of the test document d_j.

$$\text{one-error}_j = \begin{cases} 0 & \text{if } [\arg\max_{c \in \mathcal{C}} f(d_j, c)] \in C_j \\ 1 & \text{otherwise.} \end{cases} \tag{2}$$

where $[\arg\max_{c \in \mathcal{C}} f(d_j, c)]$ returns the top ranked category for the test document d_j.

Coverage (coverage$_j$) measures how far we need to go down the rank of categories in order to cover all the possible categories assigned to a test document.

$$\text{coverage}_j = \max_{c \in C_j} r(d_j, c) - 1 \tag{3}$$

where $\max_{c \in C_j} r(d_j, c)$ returns the maximum rank for the set of appropriate categories of the test document d_j.

Average Precision (avgprec$_j$) evaluates the average of precisions computed after truncating the ranking of categories after each category $c_i \in C_j$ in turn:

$$\text{avgprec}_j = \frac{1}{|C_j|} \sum_{k=1}^{|C_j|} precision_j(R_{jk}) \tag{4}$$

where R_{jk} is the set of ranked categories that goes from the top ranked category until a ranking position k where there is a category $c_i \in C_j$ for the test document d_j, and $precision_j(R_{jk})$ is the number of pertinent categories in R_{jk} divided by $|R_{jk}|$. If there is a category $c_i \in C_j$ at the position k and $f(d_j, c_i) = 0$ then $precision_j(R_{jk}) = 0$.

For p test documents, the overall performance is obtained by averaging each metric, that is $\text{hloss} = \frac{1}{p} \sum_{j=1}^{p} \text{hloss}_j$, $\text{one-error} = \frac{1}{p} \sum_{j=1}^{p} \text{one-error}_j$, $\text{coverage} = \frac{1}{p} \sum_{j=1}^{p} \text{coverage}_j$, and $\text{avgprec} = \frac{1}{p} \sum_{j=1}^{p} \text{avgprec}_j$. The smaller the value of *hamming loss*, *one-error*, and *coverage*, and the larger the value of *average precision*, the better the performance of the categorization system. The performance is perfect when $\text{hloss} = 0$, $\text{one-error} = 0$, $\text{coverage} = \frac{1}{p} \sum_{j=1}^{p} (|C_j| - 1)$, and $\text{avgprec} = 1$.

3. Vg-ram wnn

RAM-based neural networks, also known as n-tuple categorizers or weightless neural networks (WNN), do not

store knowledge in their connections but in Random Access Memories (RAM) inside the network's nodes, or neurons. These neurons operate with binary input values and use RAM as lookup tables: the synapses of each neuron collect a vector of bits from the network's inputs that is used as the RAM address, and the value stored at this address is the neuron's output. Training can be made in one shot and basically consists of storing the desired output in the address associated with the input vector of the neuron [1].

In spite of their remarkable simplicity, RAM-based neural networks are very effective as pattern recognition tools, offering fast training and test, and easy implementation [2]. However, if the network input is too large, the memory size of the neurons of WNN becomes prohibitive, since it must be equal to 2^n, where n is the input size. Virtual Generalizing RAM (VG-RAM) networks are RAM-based neural networks that only require memory capacity to store the data related to the training set [8]. In the neurons of these networks, the memory stores the input-output pairs shown during training, instead of only the output. In the test phase, the memory of VG-RAM neurons is searched associatively by comparing the input presented to the network with all inputs in the input-output pairs learned. The output of each VG-RAM neuron is taken from the pair whose input is nearest to the input presented—the distance function employed by VG-RAM neurons is the *hamming distance*. If there is more than one pair at the same minimum distance from the input presented, the neuron's output is chosen randomly among these pairs.

Figure 1 shows the lookup table of a VG-RAM WNN neuron with three synapses (X_1, X_2 and X_3). This lookup table contains three entries (input-output pairs), which were stored during the training phase (entry $\#1$, entry $\#2$ and entry $\#3$). During the test phase, when an input vector (input) is presented to the network, the VG-RAM WNN test algorithm calculates the distance between this input vector and each input of the input-output pairs stored in the lookup table. In the example of Figure 1, the *hamming distance* from the input to entry $\#1$ is two, because both X_2 and X_3 bits do not match the input vector. The distance to entry $\#2$ is one, because X_1 is the only non-matching bit. The distance to entry $\#3$ is three, as the reader may easily verify. Hence, for this input vector, the algorithm evaluates the neuron's output, Y, as category 2, since it is the output value stored in entry $\#2$.

To categorize text documents using VG-RAM WNN, we represent a document as a multidimensional vector $V = \{v_1, \ldots, v_{|V|}\}$, where each element v_i corresponds to the number of times a term in the vocabulary of interest appears in this document. We use single layer VG-RAM WNN (Figure 2) whose neurons' synapses $X = \{x_1, \ldots, x_{|X|}\}$ are randomly connected to the network's input $N = \{n_1, \ldots, n_{|N|}\}$, which has the same size of the vectors

lookup table	$\mathbf{X_1}$	$\mathbf{X_2}$	$\mathbf{X_3}$	**Y**
entry #1	1	1	0	category 1
entry #2	0	0	1	category 2
entry #3	0	1	0	category 3
	↑	↑	↑	↓
input	1	0	1	**category 2**

Figure 1. VG-RAM WNN **neuron lookup table.**

representing the documents, i.e., $|N| = |V|$. Note that $|X| < |V|$ (our experiments have shown that $|X| < |V|$ provides better performance). Each neuron's synapse x_i forms a minchinton cell with the next, x_{i+1} ($x_{|X|}$ forms a minchinton cell with x_1) [9]. The type of the minchinton cell we have used returns 1 if the synapse x_i of the cell is connected to an input element n_j whose value is larger than that of the element n_k to which the synapse x_{i+1} is connected (i.e. $n_j > n_k$); otherwise, it returns zero.

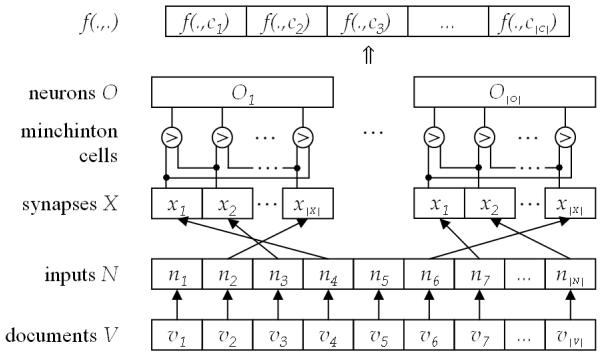

Figure 2. VG-RAM WNN **architecture.**

During training, for each document in the training set, the corresponding vector V is connected to the VG-RAM WNN's input N and the neurons' outputs $O = \{o_1, \ldots, o_{|O|}\}$ to one of the categories of the document. All neurons of the VG-RAM WNN are then trained to output this category with this input vector. The training for this input vector is repeated for each category associated with the corresponding document. During test, for each test document, the inputs are connected to the corresponding vector and the number of neurons outputting each category is counted. The network's output is computed by dividing the count of each category by the number of neurons of the network. This output is organized as a vector whose size is equal to the number of categories. The value of each vector element varies from 0 to 1 and represents the percentage of neurons which presented the corresponding category as output (the sum of the values of all elements of this vector is always equal to 1). This way, the output of the network implements the function $f(.,.)$, defined in Section 2. A threshold τ may be used with the function $f(.,.)$ to define the set of categories to be assigned to the test document.

Table 1. Characteristics of the Web page data sets (after term selection). NC denotes the number of categories, NT the number of terms in the vocabulary, PMC the percentage of documents belonging to more than one category, ANC the average number of categories of each document, and PRC the percentage of rare categories, i.e., those categories associated with less than 1% of the documents of a the data set.

Data set	NC	NT	Training(-and-validation) set			Test set		
			PMC	ANC	PRC	PMC	ANC	PRC
Arts&Humanities	26	462	44.50%	1.63	19.23%	43.63%	1.64	19.23%
Business&Economy	30	438	42.20%	1.59	50.00%	41.93%	1.59	43.33%
Computers&Internet	33	681	29.60%	1.49	39.39%	31.27%	1.52	36.36%
Education	33	550	33.50%	1.47	57.58%	33.73%	1.46	57.58%
Entertainment	21	640	29.30%	1.43	28.57%	28.20%	1.42	33.33%
Health	32	612	48.05%	1.67	53.13%	47.20%	1.66	53.13%
Recreation&Sports	22	606	30.20%	1.41	18.18%	31.20%	1.43	18.18%
Reference	33	793	13.75%	1.16	51.52%	14.60%	1.18	54.55%
Science	40	743	34.85%	1.49	35.00%	30.57%	1.43	40.00%
Social&Science	39	1 047	20.95%	1.27	56.41%	22.83%	1.29	58.97%
Society&Culture	27	636	41.90%	1.71	25.93%	39.97%	1.68	22.22%

4. Experimental Evaluation

The Web page data employed in our experiments was extracted from the Yahoo directory[1] (`http://dir.yahoo.com`). Currently, the top level of the Yahoo directory consists of 14 Web page categories (i.e., "Arts&Humanities", "Business&Economy", "Computers&Internet", and so on) and each category is further categorized into a number of second-level subcategories. By focusing on these subcategories, one can devise 14 independent text categorization problems. Zhang and Zhou [14] used 11 of these 14 problems to evaluate the performance of ML-KNN. To reduce the dimensionality of each data set, they used a simple term selection method based on document frequency (the number of documents containing a specific term)—only the top 2% terms with highest document frequency were retained in the final vocabulary. After term selection, each document in the data set was also described as a multidimensional vector using the "Bag-of-Words" representation. Table 1 summarizes the characteristics of the Web page data sets[2]. For each data set, the training(-and-validation) set contains 2000 documents while the test set contains 3000 documents.

To tune the parameters of the VG-RAM WNN categorizer for these data sets, we divided the 2000 documents training(-and-validation) set of each problem into a 1500 documents training set, which was used to inductively build the categorizers, and a 500 documents validation set, which was used to evaluate the performance of the categorizers in the series of experiments aimed at parameter optimization. The VG-RAM WNN categorizer has three parameters: (i) the number of neurons, $|O|$; (ii) the number of synapses per neuron, $|X|$; and (iii) the threshold, τ (Section 3). We tested networks with number of neurons equal to 256, 512, 1024, and 2048; number of synapses per neuron equal to 32, 64, 128 and 256; and threshold τ equal to 0.1, 0.2, 0.3, 0.4, and 0.5. Table 2 shows, for each one of the 11 text categorization problems, the parameters that yield the best VG-RAM WNN performance.

Once its parameters are estimated, we can use VG-RAM WNN to predict the set of categories of the test documents. We compared VG-RAM WNN categorization performance with that of: the multi-label lazy learning approach ML-KNN [14], the boosting-style algorithm BOOSTEXTER [11], the multi-label decision tree ADTBOOST.MH [6], and the multi-label kernel method RANK-SVM [7]. We believe that these categorizers are representative of some of the most effective multi-label text categorization methods currently available.

For ML-KNN, the number of nearest neighbors, k, was set to 10, which yield the best performance on a bioinformatic data set studied in [14]. For BOOSTEXTER and ADTBOOST.MH, the number of boosting rounds was set to be 500 and 50, respectively, because on all data sets studied in [14], the performance of these two algorithms did not significantly change after the specified boosting rounds. For RANK-SVM, polynomial kernels with degree 8 were used, which yield the best performance as shown in the litera-

[1] Data set available at `http://www.inf.ufes.br/~alberto/yahoo.tar.gz`.

[2] The characteristics of the Web page data sets were obtained from the work presented in [14].

Table 2. Parameters of VG-RAM WNN that yield the best performance. $|O|$ denotes the number of neurons, $|X|$ the number of synapses per neuron, and τ the threshold used to compute the output of the multi-label categorizer.

Data set	$\|O\|$	$\|X\|$	τ	Data set	$\|O\|$	$\|X\|$	τ
Arts&Humanities	64	1024	0.2	Recreation&Sports	64	1024	0.2
Business&Economy	64	1024	0.2	Reference	64	1024	0.5
Computers&Internet	64	1024	0.4	Science	64	1024	0.2
Education	128	1024	0.4	Social&Science	128	1024	0.4
Entertainment	128	1024	0.3	Society&Culture	64	1024	0.3
Health	128	1024	0.2				

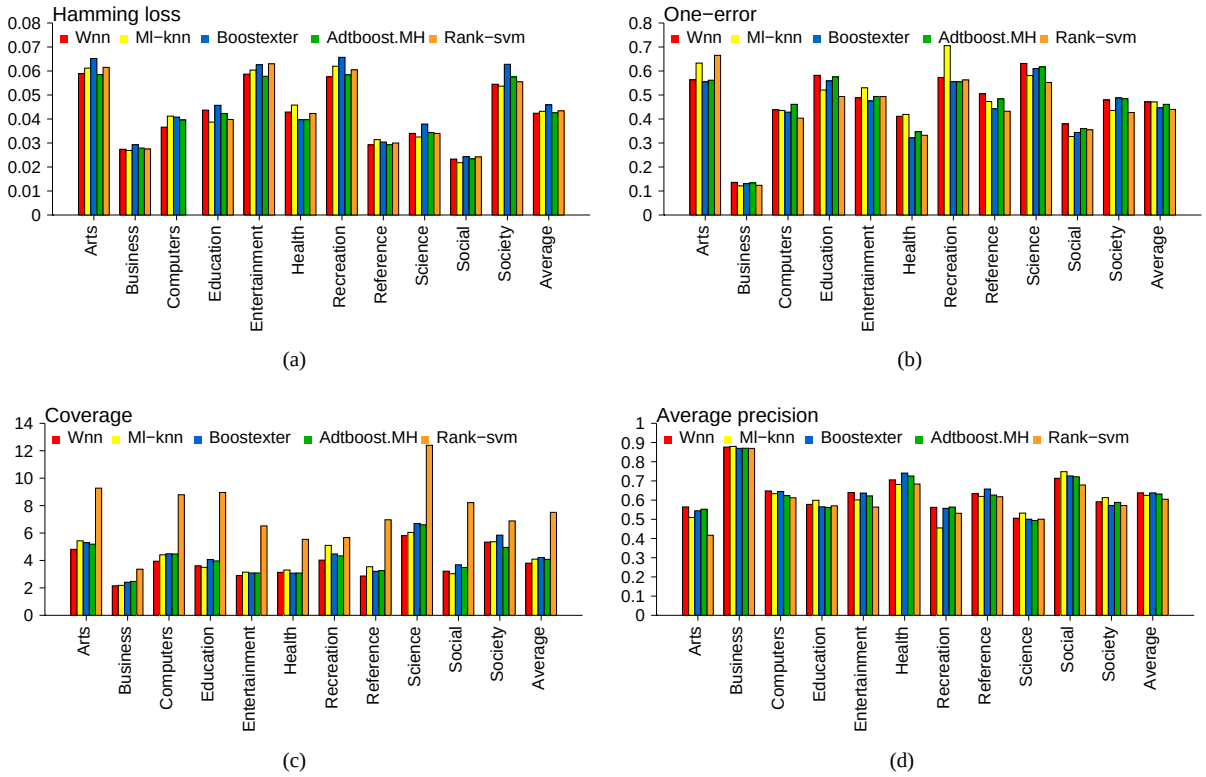

Figure 3. Experimental results of each multi-label learning algorithm on the Web page data sets in terms of *hamming loss*, *one-error*, *coverage*, and *average precision*. The smaller the value of *hamming loss*, *one-error*, and *coverage*, and the larger the value of *average precision*, the better the performance of the categorizer.

ture [7][3]. For each data set, the multi-label algorithms were trained with the 2000 documents in the the training(-and-validation) set and tested with the 3000 documents in the test set.

Figures 3(a) to 3(d) show the experimental results of each multi-label categorization technique on all the Web page data sets in terms of *hamming loss*, *one-error*, *coverage*, and *average precision*, respectively. These plottings also show the averages for each evaluation metric over all data sets. On average, VG-RAM WNN performs better than the other algorithms in terms of *hamming loss*, *coverage*, and *average precision*, and shows inferior, although com-

[3]The results for ML-KNN, BOOSTEXTER, ADTBOOST.MH, and RANK-SVM were obtained from the work presented in [14].

parable, performance than the other algorithms in terms of *one-error*. It is worth to note that all categorizers examined here perform poorly in terms of *one-error*; on average, in 45.81% of the documents tested the top-ranked category was not in the set of appropriate categories (Figure 3(b)).

In terms of *hamming loss* (Figure 3(a)), VG-RAM WNN shows performance equivalent to ADTBOOST.MH and outperforms ML-KNN, BOOSTEXTER and RANK-SVM, showing average gains of 2%, 8%, and 2%, respectively. In terms of *coverage* (Figure 3(c)), VG-RAM WNN shows a better performance than ML-KNN, BOOSTEXTER, and ADTBOOST.MH, with average gains of 7%, 10% and 7%, respectively, and presents a far superior performance than RANK-SVM, showing an average gain of 49% and a maximum gain of 61% with the "Social" data set. Finally, in terms of *average precision* (Figure 3(d)), VG-RAM WNN shows performance equivalent to BOOSTEXTER and ADTBOOST.MH, and performs better than ML-KNN and RANK-SVM, with average gains of 2% and 5%, respectively.

5. Conclusions and Future Work

In this paper, we presented an experimental evaluation of the performance of Virtual Generalizing Random Access Memory Weightless Neural Networks (VG-RAM WNN [2]) on multi-label text categorization. We compared the performance of VG-RAM WNN, on the categorization of Web pages, with that of the multi-label lazy learning technique ML-KNN [14], the boosting-style algorithm BOOSTEXTER [11], the multi-label decision tree ADTBOOST.MH [6], and the multi-label kernel method RANK-SVM [7]. Our experimental results showed that, on average, VG-RAM WNN outperforms the comparing algorithms in terms of *hamming loss*, *coverage* and *average precision*, showing average gains of up to 8% (compared with BOOSTEXTER), 49% (compared with RANK-SVM) and 5% (compared with RANK-SVM), respectively.

A direction for future work is to examine correlated VG-RAM WNN [4] and other mechanisms for taking advantage of the correlation between categories. Other direction for further research is to evaluate the categorization performance of VG-RAM WNN using different multi-label categorization problems, such as image annotation and gene function prediction.

6. Acknowledgments

We would like to thank Min-Ling Zhang for all the help with the ML-KNN categorization tool and Web page data sets. We would also like to thank *Receita Federal do Brasil*, *Conselho Nacional de Desenvolvimento Científico e Tecnológico* — CNPq-Brasil (grants 308207/2004-1, 471898/2004-0, 620165/2006-5, 309831/2007-5), *Financiadora de Estudos e Projetos* — FINEP-Brasil (grants CT-INFRA-PRO-UFES/2005, CT-INFRA-PRO-UFES/2006), and *Fundação Espírito Santense de Tecnologia* — FAPES-Brasil (grant 37711393/2007) for their support to this research work.

References

[1] I. Aleksander. Self-adaptive universal logic circuits. *IEEE Electronic Letters*, 2(8):231–232, 1966.

[2] I. Aleksander. *RAM-Based Neural Networks*, chapter From WISARD to MAGNUS: a Family of Weightless Virtual Neural Machines, pages 18–30. World Scientific, 1998.

[3] M. R. Boutell, J. Luo, X. Shen, and C. M. Brown. Learning multi-label scene classification. *Pattern Recognition*, 37(9):1757–1771, 2004.

[4] R. Carneiro, S. S. Dias, D. F. Jr., S. Oliveira, A. S. d. Garcez, and A. F. De Souza. Improving VG-RAM neural networks performance using knowledge correlation. *Lecture Notes in Computer Science*, 4232:427–436, 2006.

[5] A. Clare and R. D. King. Knowledge discovery in multi-label phenotype data. In *Lecture Notes in Computer Science*, volume 2168, pages 42–53, 2001.

[6] F. D. Comité, R. Gilleron, and M. Tommasi. Learning multi-label alternating decision tree from texts and data. In *Lecture Notes in Computer Science*, volume 2734, pages 35–49. Springer, 2003.

[7] A. Elisseeff and J. Weston. A kernel method for multi-labelled classification. In *Advances in Neural Information Processing Systems*, volume 14, pages 681–687. MIT Press, 2002.

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

[9] R. J. Mitchell, J. M. Bishop, S. K. Box, and J. F. Hawker. *RAM-Based Neural Networks*, chapter Comparison of Some Methods for Processing Grey Level Data in Weightless Networks, pages 61–70. World Scientific, 1998.

[10] E. Romero, L. Màrquez, and X. Carreras. Margin maximization with feed-forward neural networks: a comparative study with svm and adaboost. *Neurocomputing*, 57:313–344, 2004.

[11] R. E. Schapire and Y. Singer. BoosTexter: a boosting-based system for text categorization. *Machine Learning*, 39(2/3):135–168, 2000.

[12] F. Sebastiani. Machine learning in automated text categorization. *ACM Computing Surveys*, 34(1):1–47, 2002.

[13] M.-L. Zhang and Z.-H. Zhou. Multi-label neural networks with applications to functional genomics and text categorization. *IEEE Transactions on Knowledge and Data Engineering*, 18(10):1338–1351, 2006.

[14] M.-L. Zhang and Z.-H. Zhou. ML-KNN: A lazy learning approach to multi-label learning. *Pattern Recognition*, 40(7):2038–2048, 2007.