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 [10]. However, in real-world problems, multi-label categorization, in which any number of categories may be assigned to the same document, is frequently necessary [9, 5, 4, 3, 11, 12].

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 [10]. 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 [4], kernel methods [5, 3] or neural networks [8, 11], and many of them specifically for multi-label text categorization [5, 9, 4, 8, 11, 12].

In this paper, we present an experimental evaluation of the performance of virtual generalizing random access memory weightless neural networks (VG-RAM WNN [6]) 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 [9]. We also compare the VG-RAM WNN performance, according to these metrics, with that of the multi-label lazy learning technique ML-KNN [12], the boosting-style algorithm BOOSTEXTER [9], the multi-label decision tree ADTBOOST.MH [4], and the multi-label kernel method RANK-SVM [5]. Our experimental evaluation shows that VG-RAM WNN has an overall better performance than the other algorithms for the set of metrics considered.

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 [10]. In multi-label text categorization, one or more categories may be assigned to a document.

Let \( D \) be the domain of documents, \( C = \{c_1, \ldots, c_{|C|}\} \) 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 \( C \). 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 \( C \). \( TV \) 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 (tuned 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 \( f : D \times C \to \mathbb{R} \) that returns a value for each pair \( \langle d_j, c_i \rangle \in D \times C \) 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 \( f(\ldots) \) can be transformed into a ranking function \( r(\ldots) \), which is a one-to-one mapping onto \( \{1, 2, \ldots, |C|\} \) 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 \( \tau \) (i.e., \( c_k \in r(d_j, c_k) \geq \tau \)) are then assigned to the test document \( d_j \).

We have used the four multi-label evaluation metrics discussed in [9] 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 \( \tau \), the metrics one-error, coverage, and average precision evaluate the real-valued function \( f(\ldots) \), that is, the ranking quality of different categories for each test document. We present each of these metrics below.

**Hamming Loss (hloss)\)** 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.

\[
hloss_j = \frac{1}{|C|} |P_j \Delta C_j| \tag{1}
\]

where \( |C| \) is the number of categories and \( \Delta \) 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)\)** evaluates if the top ranked category is present in the set of proper categories \( C_j \) of the test document \( d_j \).

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

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

**Coverage (coverage)\)** 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)\)** evaluates the average of precisions computed after truncating the ranking of categories after each category \( c_i \in C_j \) in turn:

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

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 \( \text{precision}_j(R_{jk}) \) 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 \( \text{precision}_j(R_{jk}) = 0 \).

For \( p \) test documents, the overall performance is obtained by averaging each metric, that is \( \text{hloss} = \frac{1}{p} \sum_{j=1}^{p} \text{hloss}_j \), one-error = \( \frac{1}{p} \sum_{j=1}^{p} \text{one-error}_j \), coverage = \( \frac{1}{p} \sum_{j=1}^{p} \text{coverage}_j \), and 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 hloss = 0, one-error = 0, coverage = \( \frac{1}{p} \sum_{j=1}^{p} |C_j| - 1 \), and avgprec = 1.

3. Vg-Ram Wnn

RAM-based neural networks [1], also known as weight-less neural networks (WNN), do not store knowledge in
their connections but in Random Access Memories (RAM) inside the network’s nodes, or neurons. In spite of their remarkable simplicity, WNN 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 [6]. 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 of its input 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.

<table>
<thead>
<tr>
<th>lookup table</th>
<th>$X_1$</th>
<th>$X_2$</th>
<th>$X_3$</th>
<th>$Y$</th>
</tr>
</thead>
<tbody>
<tr>
<td>entry #1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>category 1</td>
</tr>
<tr>
<td>entry #2</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>category 2</td>
</tr>
<tr>
<td>entry #3</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>category 3</td>
</tr>
</tbody>
</table>

![Figure 1. VG-RAM WNN neuron lookup table.](image)

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 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, $x_{i+1}$, which forms a minchinton cell with $x_1$ [7]. 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.

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 $f(\ldots)$, defined in Section 2. A threshold $\tau$ may be used with the function $f(\ldots)$ to define the set of categories to be assigned to the test document.

![Figure 2. VG-RAM WNN architecture.](image)
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, and ANC the average number of categories of each document in the training(-and-validation) set $T_V$ and test set $T_e$.

<table>
<thead>
<tr>
<th>Data set</th>
<th>NC</th>
<th>NT</th>
<th>$T_V$</th>
<th>$T_e$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Arts&amp;Humanities</td>
<td>26</td>
<td>462</td>
<td>1.63</td>
<td>1.64</td>
</tr>
<tr>
<td>Business&amp;Economy</td>
<td>30</td>
<td>438</td>
<td>1.59</td>
<td>1.59</td>
</tr>
<tr>
<td>Computers&amp;Internet</td>
<td>33</td>
<td>681</td>
<td>1.49</td>
<td>1.52</td>
</tr>
<tr>
<td>Education</td>
<td>33</td>
<td>550</td>
<td>1.47</td>
<td>1.46</td>
</tr>
<tr>
<td>Entertainment</td>
<td>21</td>
<td>640</td>
<td>1.43</td>
<td>1.42</td>
</tr>
<tr>
<td>Health</td>
<td>32</td>
<td>612</td>
<td>1.67</td>
<td>1.66</td>
</tr>
<tr>
<td>Recreation&amp;Sports</td>
<td>22</td>
<td>606</td>
<td>1.41</td>
<td>1.43</td>
</tr>
<tr>
<td>Reference</td>
<td>33</td>
<td>793</td>
<td>1.16</td>
<td>1.18</td>
</tr>
<tr>
<td>Science</td>
<td>40</td>
<td>743</td>
<td>1.49</td>
<td>1.43</td>
</tr>
<tr>
<td>Social&amp;Science</td>
<td>39</td>
<td>1047</td>
<td>1.27</td>
<td>1.29</td>
</tr>
<tr>
<td>Society&amp;Culture</td>
<td>27</td>
<td>636</td>
<td>1.71</td>
<td>1.68</td>
</tr>
</tbody>
</table>

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 [12] 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, $\tau$ (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 $\tau$ 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.

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 $\tau$ the threshold used to compute the output of the multi-label categorizer.

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

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 [12], the boosting-style algorithm BOOSTEXTER [9], the multi-label decision tree ADTBOOST.MH [4], and the multi-label kernel method RANK-SVM [5]. 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 [12]. 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 [12], the performance of these two algorithms did not significantly change after the specified boosting rounds. For

\(^1\)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 [12].
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.

RANK-SVM, polynomial kernels with degree 8 were used, which yield the best performance as shown in the literature [5]. 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 comparable, 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.

To present a clearer view of the relative performances of the algorithms, a partial order > is defined on the set of all comparing algorithms for each evaluation metric, where A1 > A2 means that the performance of algorithm A1 is significantly better than that of algorithm A2 on the specific metric (two-tailed paired t-test at 5% significance level). The par-

---

3The results for ML-KNN, BOOSTEXTER, ADTBOOST.MH, and RANK-SVM were obtained from the work presented in [12].
V is rewarded with a positive score +1 if A1 ~ A2 holds, then A1 ~ A2 is penalized with a negative score −1. Based on the accumulated score of each algorithm on all evaluation metrics, a total order ~ is defined on the set of all comparing algorithms, as shown in the last line of Table 3, where A1 ~ A2 means that A1 performs better than A2. The accumulated score of each algorithm is also shown in the parentheses. As shown in Table 3, VG-RAM WNN has an overall better performance than the other algorithms for the set of metrics considered.

5. Conclusions

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 [12], the boosting-style algorithm BOOSTexter [9], the multi-label decision tree ADTBOOST.MH [4], and the multi-label kernel method RANK-SVM [5]. Our experimental results showed that VG-RAM WNN has an overall better performance than the other algorithms for the set of metrics considered.

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 — FAPESP-Brasil (grant 37711393/2007) for their support to this research work.

References