WANN-TAGGER
A Weightless Artificial Neural Network Tagger for the Portuguese Language

Hugo C. C. Carneiro, Felipe M. G. França
Systems Engineering and Computer Science Program/COPPE, Universidade Federal do Rio de Janeiro
Caixa Postal 68511, 21941-972, Rio de Janeiro, RJ, Brazil

Priscila M. V. Lima
Department of Mathematics, Universidade Federal Rural do Rio de Janeiro, BR-465, Km 47
Campus Universitário, 23890-000, Seropédica, RJ, Brazil

Keywords: WiSARD, DRASiW, POS-Tagger, Weightless artificial neural networks.

Abstract: Weightless Artificial Neural Networks have proved to be a promising paradigm for classification tasks. This work introduces the WANN-Tagger, which makes use of weightless artificial neural networks for labelling Portuguese sentences, tagging each of its terms with its respective part-of-speech. A first experimental evaluation using the CETENFolha corpus indicates the usefulness of this paradigm and shows that it outperforms traditional feedforward neural networks in both accuracy and training time, and also that it is competitive in accuracy with the Hidden Markov Model in some cases. Additionally, WANN-Tagger shows itself capable of incrementally learning new tagged sentences during runtime.

1 INTRODUCTION

In many fields in which there are text mining and information extraction tasks, many steps must be taken in order to acquire the main information from some text. The first of these steps is part-of-speech tagging, which consists of obtaining the grammatical tags of words in a sentence.

This paper presents the WANN-Tagger, a **W**eightless **A**rtificial **N**eural **N**etwork (WANN) model that makes use of the WiSARD (**Wi**lkie, **S**tonham and **A**leksander's **R**ecognition **D**evice) architecture and its advantages (Alexander et al., 1984) in order to create a tagger that is both faster and more accurate than traditional connectionist models and is also capable of learning almost instantly new tagged sentences during runtime.

In the next section some basic concepts are introduced and background knowledge for the subsequent sections provided. So, in Section 2, the concepts of weightless artificial neural networks, the WiSARD model and the DRASiW (an important WiSARD generalization) are discussed. Some related works are also described in this section. In Section 3 the specification of the WANN-Tagger is presented and its application to POS tagging explained together with the discussion of the main changes to the original WiSARD model for it to perform POS tagging. An experimental framework is presented in Section 4, which explains how to use CETENFolha corpus (Linguateca, 2009) to train the WANN-Tagger, compares it with a feedforward neural network and with a Hidden Markov Model, and discusses the experimental results. Section 5 offers some conclusions on the effectiveness of WANN-Tagger.

2 BASIC CONCEPTS

2.1 Weightless Artificial Neural Networks and the WiSARD Model

Weightless Artificial Neural Networks (WANNs) are a set of artificial neural network models in which there is no synaptic weight balancing during its training phase. The main idea behind a WANN is that the knowledge is not strictly in the synapses. There is a large variety of WANNs, for instance: WiSARD (Alexander et al., 1984); Sparse

Carneiro H., França F. and Lima P..
WANN-TAGGER - A Weightless Artificial Neural Network Tagger for the Portuguese Language.
DOI: 10.5220/0003082303300335
In *Proceedings of the International Conference on Fuzzy Computation and 2nd International Conference on Neural Computation* (ICNC-2010), pages 330-335
ISBN: 978-989-8425-32-4

Distributed Memory (Kanerva, 1988); Probabilistic Logic Nodes (Alexander and Kan, 1987); Goal Seeking Neuron (Carvalho Filho et al., 1991); Generalizing Random Access Memory (Alexander, 1990); General Neural Unit (Alexander and Morton, 1991).

In this paper, the pioneering WiSARD (Wilkie, Stonham and Aleksander's Recognition Device) model (Alexander et al., 1984) is used. It employs a set of N RAM-discriminators (see Figure 1), where N is the number of existent classes, to determine which class a pattern belongs to. Each RAM-discriminator consists of a set of X Boolean RAMs, or Boolean neurons, with n input bits each, and a summation device Σ (see Figure 2). Each RAM-discriminator (or simply discriminator) receives nX inputs that are pseudo-randomly obtained from the input retina, the binary vector (see Figure 2) used in both training and recognition phases of the WiSARD model. The summation device integrates all X Boolean RAM outputs and is activated during the recognition phase.

2.1.1 Training Phase

In order to train a target discriminator, one must set all of its memory locations to "0" and then, for each dataset entry, a "1" should be written in the memory locations addressed by the given input pattern. Unlike other artificial neural network models based on synaptic weights, the training phase of a given discriminator ends after all the input entries belonging to the corresponding training set are presented just once, not requiring recursive iterations to achieve convergence.

2.1.2 Recognition Phase

Once all discriminators are trained, one can present an unseen input pattern to all discriminators. Each discriminator will produce a response *r* to that given input, so that the discriminator presenting the highest response indicates that such target input pattern belongs to the corresponding class with confidence *d* (see Figure 1), which is usually defined as the difference between the two best discriminator responses produced. The performance of WiSARD strongly depends on *n*. WiSARD generalisation capability grows inversely with *n*.

2.2 DRASiW

When one trains a large dataset, usually there are ties, i.e., different discriminators producing the same (highest) responses during the recognition phase. In order to avoid this saturation effect (also named overfitting), DRASiW (Soares et al., 1998), an extension to the WiSARD model, is provided with the ability of producing pattern examples, or prototypes, derived from learned categories, what was proved helpful in such disambiguation process (Grieco et al., 2010).

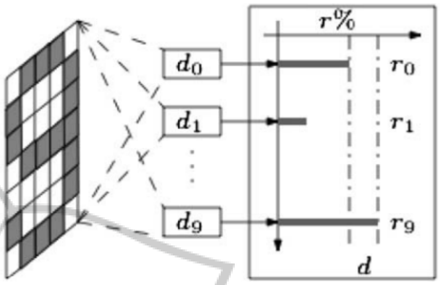

Figure 1: A 10 RAM-discriminator architecture (from (Grieco et al., 2010)).

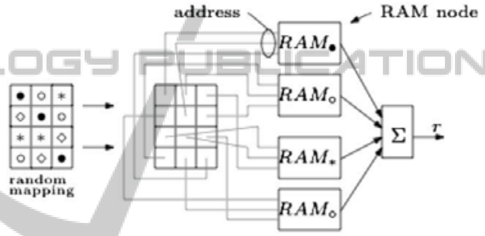

Figure 2: The retina and a RAM-discriminator (from (Grieco et al., 2010)).

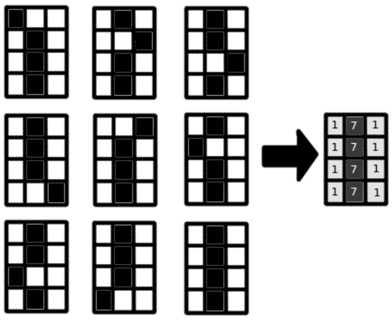

Figure 3: Training samples and the respective mental image.

A very simple modification to the WiSARD training process is needed in order to have DRASiW disambiguating discriminator responses. When an input pattern is presented, RAM locations addressed by the input pattern are incremented by 1, instead of simply storing a "1" at those memory locations. This makes RAM locations that are more accessed during the training step have higher values than the ones that are less accessed.

The change above allows WiSARD model to build "mental images" that are no longer black ("1") and white ("0"), but "greyscale" (see Figure 3). To retrieve a "black and white mental image" one must, at the end of the training step, assume value "1" in the memory locations that have values higher or equal to a luminance threshold l and assume value "0" in the other memory locations. Having $l = 0$ as an initial condition, one can iteratively increase l upon detecting ties between discriminator responses, until l reaches a value meaning that just a single winner discriminator exists (Grieco et al., 2010).

2.3 Related Works

Most part-of-speech (POS) taggers are implemented using probabilistic and statistical frameworks, e.g. Hidden Markov Models (Rabiner, 1989) (Villavicencio et al., 1995), Maximum Entropy Markov Models (McCallum et al., 2000) and Conditional Random Fields (Lafferty et al., 2001). These approaches, although discovering parts-of-speech with high accuracy, lack cognitive background and do not represent how human beings deduce the parts-of-speech of a sentence.

Other approaches include rule based taggers, such as the Constraint Grammar approach (Karlsson et al., 1995), in which the tags are obtained based on which word is being tagged and other constraints. And there are yet other ones, like transformation based learning (TBL) taggers (Brill, 1995) and neural taggers (Schmid, 1994) (Marques and Lopes, 1996). The former works as the rule based taggers, but the rules are automatically induced from the data, like some HMM taggers. The latter makes use of artificial neural networks, mainly feedforward ones.

3 WANN-TAGGER

3.1 General Architecture

The WANN-Tagger is a part-of-speech tagger that makes use of the WiSARD model in order to tag sentences in Portuguese. Words are represented as a probability vector with values $p(tag_i|word)$ when the word is relevant to the corpus, i.e., appears in at least 0.5% of the corpus, and are represented as $p(tag_i|word\ suffix)$ otherwise.

Besides the probability of a word having a particular tag, the WANN-Tagger also takes into account the context in which the word is presented. In order to consider it, the WANN-Tagger employs a context window possessing b words before and a words after the current one plus the current word itself. Thus, the WANN-Tagger input is composed of $a+b+1$ words, i.e. $a+b+1$ vectors with N probabilities each. When there are less than b words before the current word or less than a words after it, then the probability vectors corresponding to the spaces of the context window lacking words have '0' in such positions.

The WANN-Tagger, as a Boolean neural network, does not support the use of floating points as inputs. In order to use the probability vectors one must discretize them. In this discretization process, probabilities are encoded as Boolean vectors according to the following condition: given two distinct probabilities, the one with higher value will be encoded as a Boolean vector with more 1's than the other.

WANN-Tagger has a large variety of classes, which are listed in its tagset. As mentioned in the previous section, ties can often happen during the classification step when a WiSARD model that has a large training set is used. In order to avoid ties, the WANN-Tagger extensively uses the DRASiW mechanism to find the best value of the luminance threshold l that prevents such ties.

In (Grieco et al., 2010) it is stated that these ties can be avoided by gradually increasing the luminance threshold. However, when one uses a large variety of classes, this can take too much time until no tie can be found. To avoid this time spending, a "binary search" luminance threshold adjustment is used, which consists of an iterative process containing the following statements:

- if it is the first iteration of the "binary search" luminance threshold adjustment one must set a minimum threshold to the number of times the most accessed RAM-discriminator was accessed; test the WANN-Tagger using $l = 1$, i.e., if an entry in the memory location is greater or equal to "1", then assume value "1" in this memory location, otherwise assume value "0"; check which discriminator produced the highest response and call this value V;
- if there were ties in the previous iteration and the greatest value was V, then set the maximum threshold to the arithmetic mean of the minimum threshold and the former maximum threshold and test the WANN-Tagger with the recently adjusted maximum threshold as l;
- if the greatest result in the previous iteration was less than V, then set the minimum threshold to the arithmetic mean of the former minimum

threshold and the maximum threshold and test the WANN-Tagger with the recently adjusted minimum threshold as *l*.

Repeat this process until having a result in which the greatest value is *V* and there is no tie.

3.2 Architecture for Portuguese

WANN-Tagger has been implemented to be tested in Portuguese, so that its capability of tagging accurately a highly inflected and somewhat free word order language would be guaranteed.

Several experiments have been done to discover which WANN-Tagger parameter values that maximize its accuracy. They employ a dataset containing 2000 sentences randomly extracted from CETENFolha corpus (Linguateca, 2009) and several distinct context windows, probability discretization degrees and number of inputs in each of the RAMs of the RAM-discriminators.

The results show that the context window $W = (1,1)$ is the one which the WANN-Tagger obtains the best accuracy with, and that the best probability discretization degree is $d = 8$ and the best number of inputs in each of the RAMs of the RAM-discriminators is $n = 16$. Any context window larger than $W = (1,1)$ or values for d and n greater than the ones obtained in the experiments imply in the WANN-Tagger becoming overfitted.

4 METHODOLOGY AND EXPERIMENTS

4.1 Dataset

In order to test the WANN-Tagger, different samples from the CETENFolha corpus (Linguateca, 2009) have been used. These samples were created by extracting random sentences from the corpus. This extraction has consisted in dividing the corpus in *N* parts with the same amount of sentences, and extracting one random sentence from each part. The amounts of sentences in these samples are logarithmically gradual.

The tagset used in this paper is a simplified version of CETENFolha corpus one, as shown in Table 1. This simplified version was created because CETENFolha corpus tagset, PALAVRAS (Bick, 2000), contains very meticulous tags, in which even gender and number (or finiteness, person, tense and mood in the case of a verb) are shown.

Table 1: The tagset used in the experiments.

Tag	Description	Tag	Description
PROP	Proper Noun	ADV	Adverb
N	Noun	PRP	Preposition
PERS	Personal Pronoun	KC	Coordinate Conjunction
POSS	Possessive Pronoun	KS	Subordinate Conjunction
DET	Determiner	NUM	Numeral
ADJ	Adjective	IN	Interjection
V	Verb	PT	Punctuation
SPEC	Specifier		

4.2 Experiments

To prove the effectiveness of the WANN-Tagger, it has been compared with (i) a feedforward neural network trained via Backpropagation, and (ii) with a Hidden Markov Model (HMM).

The feedforward neural network uses the architecture presented in (Schmid, 1994). Like in WANN-Tagger the probabilities of a word (or a suffix) possess a particular tag are obtained by using their relative frequencies in the corpus. The epoch size is equal to the amount of sentences of each sample used in the experiments. Its learning rate is constant and equal to 0.001. The halting criteria consist in either the convergence of the training, or reaching 1000 epochs.

Several experiments have been done and their results showed that the context window that best fits with the feedforward neural network model is $W = (1, 1)$, i.e., one word before the current one and one after it and that the amount of nodes in the hidden layer that produces the best accuracy is $n = 14$. The two experiments mentioned above imply that the feedforward model which presented the best accuracy rates was 45-14-15.

The Hidden Markov Model used is the one presented in LingPipe (Alias-i, 2009), a Java library suite for natural language processing. It was set to use Laplace smoothing and thus causing it to estimate the part-of-speech of words that it does not possess in its vocabulary, based only on their contexts. To test the accuracy of all the models 10-fold cross-validation was used.

4.3 Results and Discussion

By testing these three models it is notable that the WANN-Tagger is considerably better than feedforward neural network in both accuracy and training time and that it is not as good as the Hidden Markov Model in accuracy rate but, unlike it, its accuracy rate standard deviation is very small and thus more reliable. HMM can achieve accuracy rates of 94%, but varying from 65% to 100%, showing that it can be outperformed by the WANN-Tagger in some cases, as can be seen in Figure 4. The training time of the HMM, however, always outperforms the one of the WANN-Tagger as it is shown in Figure 5.

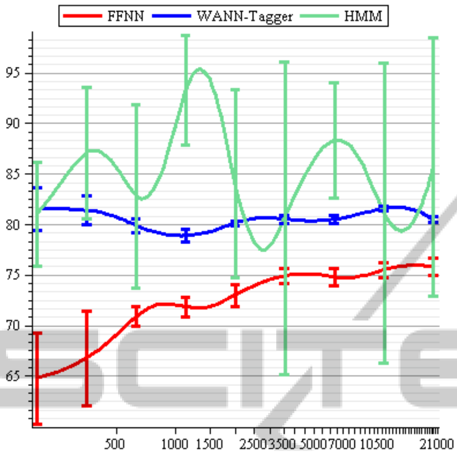

Figure 4: Accuracy rate. The horizontal axis represents the number of sentences in the sample, the vertical represents the accuracy achieved by the model. Each curve indicates the mean accuracy and its 95% confidence interval.

One can note, however, that HMM performed better than the other two models because it employs heuristics that are used by neither the WANN-Tagger nor the feedforward neural network. HMM, through the forward-backward algorithm (also known as Baum-Welch algorithm) (Baum, 1972), a special case of the Expectation-Maximization or EM algorithm (Dempster et al., 1977), is able to obtain the best Markov model that maximizes the probability of a POS sequence given a sequence of words. This algorithm allows HMM to know the whole clause, whereas both WANN-Tagger and the feedforward neural network model only know the part of the clause within the context window.

One may also note that for each word in a sentence the HMM must read two entries only, the word and the tag (already stored in memory), which was the tag of the previous word. On the other hand, the WANN-Tagger has a context window which contains three words. It is clear that for each word in a sentence the tagger must discretize a number of elements equal to three times the number of classes, in this case 15. The feedforward neural network, however, does not have a discretization process. The main reason why it is outperformed in training time by both HMM and WANN-Tagger is due to the training convergence nature.

It is important to perceive that as WANN-Tagger makes use of WiSARD it does not require recursive iterations to achieve convergence and therefore, unlike feedforward neural networks, when it receives a new tagged sentence to be trained it does not need to retrain itself, it only needs to increment the values in the memory locations addressed by this input pattern. As it is only training one sentence its training time is almost zero, as can be seen in Figure 5, making it capable of learning new tagged sentences during runtime.

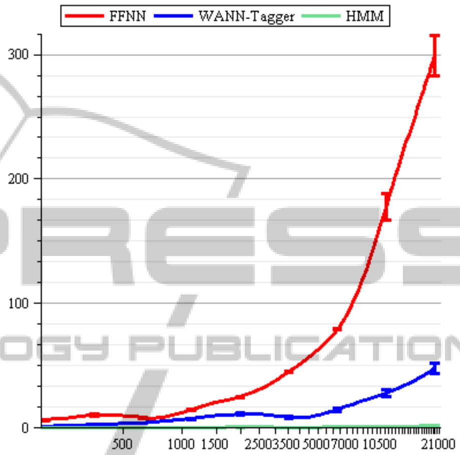

Figure 5: Time spent during experiments. The horizontal axis represents the number of sentences in the sample, the vertical represents the time spent in seconds in each fold. Each curve indicates the mean time spent and its 95% confidence interval.

In addition, it is worth noticing that Portuguese, as a highly inflected language, has a flexible word order in its sentences. This word order flexibility is responsible for the accuracy rates presented in Figure 4, being a little lower than the ones obtained in papers that use less inflected languages, such as English (Schmid 1994).

Tests with a corpus written in English, the Brown Corpus (Francis and Kučera, 1982), have also been done and the results show that WANN-Tagger performs almost as well as when it is trained with CETENFolha. WANN-Tagger performance achieves 79.56% when using the same configuration used for Portuguese. It is worth noting that Brown Corpus uses far more tags than it was used for Portuguese. If WANN-Tagger had the same amount of tags its accuracy would possibly be greater.

5 CONCLUSIONS

A weightless artificial neural network model for tagging Portuguese language sentences was introduced in this paper. The model showed itself as an alternative to feedforward neural network part-of-speech taggers as the former is faster and had better accuracy rate than the latter. WANN-Tagger also showed that its accuracy rate is as high as of Hidden Markov Models in some cases. However, the WANN-Tagger does not show itself as being competitive in training time with the Hidden Markov Model, as it needs a context window and a large number of inputs for each of its RAMs.

In order to make this model more competitive with the Hidden Markov Model, the possibility of adding Markov information to the WANN-Tagger inputs, for instance the tag obtained during the tagging phase of the previous word, will be left for future work. This way, the model would be able to know more information about the whole sentence and not only about a small part of it that is within the context window.

In addition, it is worth noting that WANN-Tagger performed well with both a free and a fixed word order language and showed itself as a good model to be used when training is required during runtime.

ACKNOWLEDGEMENTS

This work was partially supported by CNPq (306070/2007-3) and FAPERJ (E-26/102.957/2008) Brazilian research agencies.

REFERENCES

Alexander, I., Thomas, W. V., Bowden, P. A., 1984. WiSARD: a radical step forward in image recognition. In *Sensor Review*, pp. 120-124.

Alexander, I., Kan, W. W., 1987. A probabilistic logic neuron network for associative learning. In *IEEE Proceedings of the First International Conference on Neural Networks*, pp. 541-548.

Alexander, I., 1990. Ideal neurons for neural computers. In *Parallel Processing in Neural Systems and Computers*, North-Holland, Amsterdam, pp. 225-228.

Alexander, I., Morton, H., 1991. General neural unit: retrieval performance. In *IEE Electronics Letters*, 27, pp. 1176-1178.

Alias-i, 2009. LingPipe 3.9.2 http://alias-i.com/lingpipe (accessed October 1, 2009).

Baum, L. E., 1972. An inequality and associated maximization technique in statistical estimation for probabilistic functions of Markov processes. In Shinsha, O. (Ed.), *Inequalities III: Proceedings of the 3rd Symposium on Inequalities*, University of California, Los Angeles, pp. 1-8. Academic Press.

Bick, E., 2000. *The Parsing System Palavras – Automatic Grammatical Analysis of Portuguese in a Constraint Grammar Framework*. Århus University Press.

Brill, E., 1995. Transformation-based error-driven learning and natural language processing: a case study in part-of-speech tagging. In *Computational Linguistics*, 21(4), pp. 543-566.

Carvalho Filho, E. C. B., Fairhurst, M. C., Bisset, D. L., 1991. Adaptive pattern recognition using goal-seeking neurons. In *Pattern Recognition Letters*, 12, pp. 131-138.

Dempster, A. P., Laird, N. M., Rubin, D. B., 1977. Maximum likelihood from incomplete data via the EM algorithm. In *Journal of the Royal Statistical Society*, 39(1), pp 1-21.

Francis, W. N., Kučera, H. (1982). *Frequency Analysis of English Usage*. Houghton Mifflin, Boston.

Grieco, B. P. A., Lima, P. M. V., De Gregorio, M., França, F. M. G., 2010. Producing pattern examples from "mental" images. In *Neurocomputing*, 73, pp. 1057-1064.

Kanerva, P., 1988. *Sparse Distributed Memory*. MIT Press.

Karlsson, F., Vuotilainen, A., Heikkilä, J., Anttila, A., 1995. Constraint Grammar: A Language-Independent System for Parsing Unrestricted Text. Mouton de Gruyter.

Lafferty, J., McCallum, A., Pereira, F., 2001. Conditional random fields: probabilistic models for segmenting and labeling sequence data. In *Proceedings of the 18th International Conference on Machine Learning*.

Linguateca, 2009. CETENFolha http://www.linguateca.pt/CETENFolha/index_info.html (accessed October 1, 2009).

Marques, N. C., Lopes, G. P., 1996. Using neural nets for portuguese part-of-speech tagging. In *Proceedings of the 5th International Conference on the Cognitive Science of Natural Language Processing*.

McCallum, A., Freitag, D., Pereira, F., 2000. Maximum entropy Markov models for information extraction and segmentation. In *Proceedings of the 17th International Conference on Machine Learning*.

Rabiner, L. R., 1989. A tutorial on hidden Markov models an selected applications in speech recognition. In *Proceedings of the IEEE*, 77(2), pp. 257-286.

Schmid, H., 1994. Part-of-speech tagging with neural networks. In *Proceedings of the 15th Conference on Computational Linguistics*, 1, pp. 172-176.

Soares, C. M., da Silva, C. L. F., De Gregorio, M., França, F. M. G., 1998. Uma implementação em software do classificador WiSARD. In Proceedings of the V Simpósio Brasileiro de Redes Neurais (SBRN 1998), 2, pp. 225-229. (In Portuguese).

Villavicencio, A., Marques, N. M., Lopes, J. G. P., Villavicencio, F., 1995. Part-of-speech tagging for portuguese texts. In *Proceedings of the 12th Symposium on Artificial Intelligence*, 991, pp. 323-332.