ESANN 2021 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 6-8 October 2021, i6doc.com publ., ISBN 978287587082-7. Available from http://www.i6doc.com/en/.

Weightless Neural Networks for text classification using *tf-idf*

Massimo De Gregorio[1], Antonio Sorgente[1] and Giuseppe Vettigli[2]

1- Istituto di Scienze Applicate e Sistemi Intelligenti – CNR – Italy
2- Centrica plc – United Kingdom

Abstract. While Weightless Neural Networks (WNN) have been proven effective in Natural Language Processing (NLP) applications, they require the use of highly customized features as they work on binary inputs. However, recent advancements have brought methodologies able to adapt WNN to real numbers showing competitive results on many classification tasks, but they often struggle on sparse data. In this paper, we show that WNN can successfully use sparse linguistic features, like *tf-idf*, using appropriate transformations. We also show that WNN can be used to improve the performances of existing models for Mixed Language Sentiment Analysis and that it has competitive performances for news categorization.

1 Introduction

Weightless Neural Networks have already been proven effective for specific Natural Language Processing applications, like Part of Speech tagging [1]. However, given the binary nature of the input data, they require a heavy adaptation of traditional features used in the field. The introduction of the *WiSARD classifier* changes the game enabling WNN to use real numbers as input. Unfortunately, features like *tf-idf* (term frequency-inverse document frequency [2]), that are an important building block for many NLP applications and have been proven effective in the most recent NLP challenges [3, 4], have a sparse nature which, as demonstrated in [5], makes them not ideal for WNN.

In this paper we provide experimental evidence suggesting that using appropriate transformations, WNN can operate on *tf-idf* with good results. For our experiments, we considered 7 different datasets and 4 different feature transformations. The datasets cover tasks falling under the umbrella of affectual states prediction (such as Sentiment Analysis, subjectivity detection, irony detection) and text categorization. We also provide results for ad hoc models built for Mixed Language Sentiment Analysis and news categorization.

In Section 2 we briefly introduce the *WiSARD classifier*. In Section 3 we introduce the feature transformations considered for the experiments, while in Section 4 we describe the experiments performed and comment on the results. Finally, in Section 5 we offer some conclusions.

2 The WiSARD Classifier

The WiSARD, originally conceived as a pattern recognition device mainly focusing on binary image processing [6], belongs to the class of weightless neural systems. Its peculiarity is that of using lookup tables (RAM) to store the function computed by each neuron rather than storing it in weights of neuron connections. The functionality of a classical neuron can be modified by changes in the weights, while for the weightless counterpart by changes in the RAM contents. WiSARD has been adopted with very good results in a wide variety of applications [7]. Because of its characteristic of receiving only binary inputs, for every single application, the WiSARD has always to be tailored to the intended domain of application.

With the proposal of the *WiSARD classifier*, firstly introduced in [8] and later in [9], the obstacle of providing binary input has been overcome by *ad hoc* data input transformations. In fact, the *WiSARD classifier* can receive in input real and integer numbers as well as nominal data.

3 *tf-idf* features and transformations

The *tf-idf* representation associates each term t of the corpus with the value $tf(t,d) \times idf(t)$, where $tf(t,d)$ is the count of t in the document d and $idf(t)$ is defined as

$$idf(t) = \log \frac{|D|}{1 + |\{s : t \in s\}|},$$

where D is the set containing all the documents in the corpus. Note that we consider as a term a contiguous sequence of n items in the document, these sequences are called n-grams. The input of our system is obtained by arranging these values in a matrix where the rows correspond to documents and columns correspond to terms. This matrix is known as the document-term matrix and its columns are features that describe the content of the documents. We consider this matrix as input for our applications.

For our experiments we considered the original document-term matrix and the following transformations applied to it:

- Random Trees Embedding (*RTE*) – This transformation is performed ensembling a set of Decision Trees fitted on the distribution of the input data, each sample is encoded according to leaves of the trees in which it falls [10]. Any leaf of each tree is associated with a binary value of the encoding in output. We chose this transformation because it is completely binary and can be used with the basic formulation of WiSARD.

- Singular Value Decomposition (SVD) – This transformation linearly projects the data into a lower dimensional space where the variance is maximized [11]. This transformation reduces the columns of the document-term matrix and has been successfully applied in many NLP applications.

- *K-Best* features selection – This methods selects the K most important features according to the ANOVA F-value [12]. This allows to reduce the sparsity of the input data focusing only on features that are highly correlated with the target.

ESANN 2021 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 6-8 October 2021, i6doc.com publ., ISBN 978287587082-7. Available from http://www.i6doc.com/en/.

- A combination of features extracted using *SVD* and the *K-Best* features (*S/KB*) – The idea behind this combination is that the features extracted using *SVD* can complement the *K-Best* features with information brought by groups of features loosely correlated with the target but still helpful.

Notice that all the transformations just mentioned have the goal to reduce the dimensionality of the input data apart from *RTE*, which usually produces good results expanding the dimensionality of the dataset by reducing the granularity of the columns. The parameters for each model, including input and output dimensionality, have been selected using a grid search.

4 Experiments and results

In our experiments we evaluated the effectiveness of using the *WiSARD classifier* with different transformations of the *tf-idf* features on various datasets and tasks. Then, we built upon the best performing setups to create expanded models for two of the tasks.

4.1 Datasets

We collected 7 datasets, covering multiple languages and different types of classification problems. The datasets considered for our experiments are: *Sentiment140*, *StockMarket*, *sentipolc*, *SentiMix*, *BBC News* and *Amazon HTC*. We will now introduce them briefly.

Sentiment140, introduced in [13], contains a set of tweets about general topics annotated according to their polarity. *StockMarket*, released in [14], contains tweets regarding financial topics annotated again according to the polarity. These two datasets are both used for polarity prediction. *Sentipolc*, which was used for 3 different tasks at the EVALITA competition in 2016 [15]. The competition tasks were about predicting polarity, subjectivity, and irony from Italian tweets. The Hinglish and Spanglish dataset used for the *SentiMix* competition at SemEval 2020 [16]. It was a particularly challenging competition about predicting the polarity from tweets written mixing different languages. *BBC News* contains articles from the BBC website annotated according to their category.[1] *Amazon HTC* contains a set of customer reviews from Amazon.com annotated according to the categories of the items under review.[2]

4.2 Evaluation of *tf-idf* transformations

The goal of this experiment is to prove that the performances of the *WiSARD classifier* using *tf-idf* features can be improved using an appropriate transformation. For each task related to the datasets mentioned in Section 4.1, we evaluated the *WiSARD classifier* using the *tf-idf* representation based on unigrams as features and applying each transformation reported in Section 3. For

[1] http://mlg.ucd.ie/datasets/bbc.html

[2] https://www.kaggle.com/kashnitsky/hierarchical-text-classification

Dataset (Task)	***tf-idf***	***RTE***	***SVD***	***K-Best***	***S/KB***
Sentiment140	63.24%	60.14%	66.83%	66.00%	**69.88%**
StockMarket	61.01%	62.83%	58.35%	64.73%	**65.19%**
SentiMix (Hinglish)	49.27%	43.55%	53.12%	57.08%	**58.41%**
SentiMix (Spanglish)	**40.60%**	36.28%	34.07%	38.08%	40.56%
Sentipolc (Task 1)	38.80%	52.53%	53.77%	53.04%	**54.40%**
Sentipolc (Task 2)	41.41%	55.73%	53.06%	**60.31%**	59.22%
Sentipolc (Task 3)	43.76%	46.26%	47.01%	48.29%	**48.63%**
BBC news	93.04%	92.98%	96.24%	**97.10%**	97.05%
Amazon-HTC (cat 1)	79.87%	84.35%	84.81%	82.53%	**86.55%**

Table 1: F_1-scores achieved using *WiSARD classifier* with different transformations of the *tf-idf* features and using cross validation with 5 folds. The parameters of each model were picked using grid search.

datasets that are only in English, a stemmer was used to preprocess the text. Emojis are considered as single terms. The evaluation was performed using cross-validation with 5 folds and using the F_1-score as a performance metric. Each step of the cross-validation considers the full pipeline, from terms extraction to classification.

The Table 1 shows the results obtained by the implementation of WiSARD-classifier[3] defined in [9]. In this table, one can note that *RTE* decreases the performances in half of the cases. These results can be due to *RTE* removing too much information in the binary conversion. The *SVD* and *K-Best* lead to improvements in the majority of the cases, with *K-Best* providing better results on average. This suggests that WiSARD can perform well on sparse data as long as the features are highly correlated with the target. Finally, *S/KB* produces the best scores for most of the tasks, strengthening the hypothesis that features extracted by *SVD* can be complementary to those selected by *K-best*.

4.3 Mixed Language Sentiment Analysis

This task aims to predict the polarity (positive, negative, and neutral) of tweets where Hindi and English are mixed. The goal of this experiment is to validate that the *WiSARD classifier* can be used to improve the performances of classifiers that are known to perform well on this specific task.

With this in mind, we considered a set of features that have already been proven relevant as reported in [17]. The features are uni-grams, bi-grams, and tri-grams at word and character level. Also the percentage of words in English, a categorical feature that indicates the predominant language, the number of words in the text, the number of non-alphanumerical characters, and the number of uppercase characters. All features were transformed using *S/KB*. As a preprocessing step, we removed 10% of the most frequent words in an attempt to filter out stopwords from both languages. We then created a classifier that

[3] Available at https://github.com/giordamaug/WisardClassifier-C_vectors

ensembles Logistic Regression, Ridge Classifier, and *WiSARD classifier* via soft voting. Finally, we evaluated the results with and without WiSARD on the final test set of the competition and using cross-validation with 5 folds on the training data. The rationale behind this choice is that linear classifiers have already been proven effective on these features and that WiSARD is based on a very different logic, hence the WiSARD can improve over the linear classifier taking into account information previously ignored.

The ensemble that includes WiSARD achieved an F_1-score of 67.19% on the final test set of the competition, which is above the average of the leaderboard ranking 25^{th} out of 62 submissions. This represents a gain of 0.80% compared to the ensemble that only uses linear classifiers. Note that such improvement would gain 10 positions on the leaderboard. The improvement of the *WiSARD classifier* was also observed using cross-validation where it increased the score by 1.08%.

4.4 News categorization

The goal of this experiment is to demonstrate that the WiSARD is competitive with other classifiers known in the literature for the task of text categorization. For such experiment we have used *BBC News* dataset consisting of 2225 documents from the BBC news website corresponding to stories in five topical areas from 2004-2005: business, entertainment, politics, sport, tech. We have preprocessed the text removing the stopwords and we extracted the *tf-idf* features considering bi-grams. Finally, we apply *K-Best* selecting 1000 features.

In Table 2 we compare the results of the *WiSARD classifier* with a set of classifiers that produce good results on the features considered for this specific dataset. In this table, we note that the results of WiSARD are comparable to the results of other classifiers.

5 Conclusions

In this work we tested the *WiSARD classifier* on NLP tasks of various nature. Our preliminary results show that WiSARD can successfully use *tf-idf* if appro-

Model	**Accuracy**	**Precision**	**Recall**	$\boldsymbol{F_1}$	$\boldsymbol{\Delta F_1}$
WiSARD	97.17%	97.17%	97.11%	97.10%	0.00% (-)
Random Forest	96.18%	96.31%	96.06%	96.15%	-0.95% (↓)
Logistic Reg.	96.63%	96.85%	96.43%	96.58%	-0.51% (↓)
MultinomialNB	96.36%	96.58%	96.13%	96.29%	-0.80% (↓)
K-Neighbors	86.74%	89.78%	86.17%	87.06%	-10.04% (↓)
LinearSVC	97.57%	97.61%	97.56%	97.56%	**0.46%** (↑)

Table 2: Experimental results achieved using cross-validation with 10 folds on BBC News dataset. The column ΔF_1 reports the difference of F_1-score between the *WiSARD classifier* and the other methods.

priately transformed. Also, our results on specific tasks suggest that WiSARD can be competitive with other classification techniques and that it can complement other classification methods.

For our future experiments, we plan to use different types of linguistic features and to use WiSARD as building block for more complex NLP models.

References

[1] H.C.C. Carneiro, F.M.G. França, and P.M.V. Lima. Multilingual part-of-speech tagging with weightless neural networks. *Neural Networks*, 66:11–21, 2015.

[2] T. Roelleke and J. Wang. Tf-idf uncovered: a study of theories and probabilities. In *Proc. of the* 31st *ACM SIGIR*, pages 435–442, 2008.

[3] G. Vettigli and A. Sorgente. EmpNa at WASSA2021: A lightweight model for the prediction of empathy, distress and emotions from reactions to news stories. In *Proc. of the* 11th *WASSA*. Association for Computational Linguistics, 2021.

[4] F. Mele, A. Sorgente, and G. Vettigli. SentNa@ATE_ABSITA: Sentiment analysis of customer reviews using boosted trees and lexical feature. In *Proc. of the* 7^{t}*h EVALITA, Online. CEUR. org*, 2020.

[5] L.F. Kopp, J. Barbosa-Filho, P.M.V. Lima, and C.M. de Farias. Modeling sparse data as input for weightless neural network. In *Proc. of the* 27th *ESANN*, pages 331–336, 2019.

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

[7] M. De Gregorio, F.M.G. França, P.M.V. Lima, and W.R. de Oliveira. Advances on weightless neural systems. In *22th European Symposium on Artificial Neural Networks, ESANN 2014, Bruges, Belgium, April 23-25, 2014*, 2014.

[8] M. De Gregorio and M. Giordano. The WiSARD classifier. In *Proc. of the* 24th *ESANN*, pages 447–452, 2016.

[9] M. De Gregorio and M. Giordano. An experimental evaluation of weightless neural networks for multi-class classification. *Applied Soft Computing*, 72:338–354, 2018.

[10] F. Moosmann, B. Triggs, and F. Jurie. Fast discriminative visual codebooks using randomized clustering forests. In *Proc. of the* 20th *NIPS*, pages 985–992. MIT Press, 2006.

[11] N. Halko, P. Martinsson, and J.A. Tropp. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM review*, 53(2):217–288, 2011.

[12] D.M. Diez, C.D. Barr, and M. Cetinkaya-Rundel. *OpenIntro statistics.* OpenIntro, 2012.

[13] A. Go, R. Bhayani, and L. Huang. Twitter sentiment classification using distant supervision. *CS224N project report, Stanford*, 1(12):2009, 2009.

[14] Yash Chaudhary. Stock-market sentiment dataset, 2020.

[15] F. Barbieri, V. Basile, D. Croce, M. Nissim, N. Novielli, and V. Patti. Overview of the evalita 2016 sentiment polarity classification task. In *Proc. of the* 3rd *CLiC-it &* 5th *EVALITA*, 2016.

[16] P. Patwa, G. Aguilar, S. Kar, S. Pandey, S. PYKL, B. Gambäck, T. Chakraborty, T. Solorio, and A. Das. Semeval-2020 task 9: Overview of sentiment analysis of code-mixed tweets. *arXiv e-prints*, pages arXiv–2008, 2020.

[17] E. Bear, D.C. Hoefels, and M. Manolescu. Tuemix at SemEval-2020 task 9: Logistic regression with linguistic feature set. In *Proc. of the* 14th *Workshop on Semantic Evaluation*, pages 1316–1321, 2020.