ESANN 2019 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 24-26 April 2019, i6doc.com publ., ISBN 978-287-587-065-0.
Available from http://www.i6doc.com/en/.

Prediction of Palm Oil Production with an Enhanced n-Tuple Regression Network

Leopoldo A. D. Lusquino Filho[1], Luiz F. R. Oliveira[1], Aluizio L. Filho[1], Gabriel P. Guarisa[1], Priscila M. V. Lima[2], Felipe M. G. França[1] *

1- PESC/COPPE 2- NCE
Universidade Federal do Rio de Janeiro, RJ, Brazil

Abstract. This paper introduces Regression WiSARD and ClusRegression WiSARD, two new weightless neural network models that were applied in the challenging task of predicting the total palm oil production of a set of 28 differently located sites under different climate and soil profiles. Both models were derived from the n-tuple regression weightless neural model and obtained error (MAE) rates of 0.08737% and 0.08938%, respectively, which are very competitive with the state-of-art (0.07569), whilst being four (4) orders of magnitude faster during the training phase.

1 Introduction

Regression is one of the most important machine learning tasks, given the wide range of practical situations in the real world where it is necessary to predict values in a given continuum space. Due to its great utility, it is desirable that simple devices, such as small sensors, could perform regression with online training. Weightless artificial neural networks (WANNs), due to its lean, RAM-based architecture, seems to be ideal for this type of task.

This paper presents and explores the use of WANNs in the KDD18 competition [5], a challenge which goal is to predict the palm oil harvest productivity of a set of 28 different production fields using data provided by an agribusiness company. The dataset contains information about palm trees varieties, harvest dates, atmospheric data during the development of the trees, and soil characteristics of the fields where the trees are located in. The novel WANN models are based on the n-tuple Regression Network [3], which has been proved successful when compared to other classical regression approaches in non-linear plant approximation, and Mackey-Glass chaotic time series prediction tasks.

The remainder of this text is organized as follows: Section 2 presents the two weightless models proposed for regression, as well as the basic concepts behind the models that inspired it: WiSARD [1] and n-tuple Regression Network. Section 3 discusses the various approaches used in the KDD18 competition, as well as a comparison with state-of-the-art methods and other relevant results. Conclusion and future work are presented in Section 4.

*This work was partially supported by CAPES, CNPq, FAPERJ and FINEP, Brazilian research agencies.

ESANN 2019 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 24-26 April 2019, i6doc.com publ., ISBN 978-287-587-065-0.
Available from http://www.i6doc.com/en/.

2 Enhancing n-tuple regression

2.1 WiSARD's and n-tuple regression basics

- WiSARD[1]: a n-tuple classifier composed by class *discriminators*; each discriminator is a set of N RAM nodes having n address lines each. All discriminators share a structure called *input retina*, from which a pseudo-random mapping of its $N * n$ bits composes the input address lines of all of its RAM nodes. In this model, the learning phase consists of writing "1"s; classification is done via reading all RAM nodes and performing a summation of the addressed contents;
- ClusWiSARD: an extension of the WiSARD model that uses more than one discriminator per class and has a policy of creating new discriminators if the example presented to the network is not sufficiently similar to those learned by already existing discriminators; this way, ClusWiSARD creates sub-profiles for learned classes; this model can be used in a supervised, semi-supervised and unsupervised manner; in a challenge of financial credit analysis, ClusWiSARD outperformed SVM by two orders of magnitude in training time, while remaining competitive in accuracy [2];
- n-tuple regression network[3]: modification of the basic n-tuple classifier architecture, which allows it to operate as a non-parametric kernel regression estimator; it is also capable of approximating probability density functions (pdfs) and deterministic arbitrary function mappings; for this, the n-tuple regression network uses a RAM-based structure, where each memory location stores a counter and a weight, which is updated through the LMS algorithm [4].

2.2 RegressionWiSARD

RegressionWiSARD (ReW) is an extension of the n-tuple Regression Network, which adds to its original structure some characteristics of the WiSARD. Here is a description of its general architecture:

- Each RAM location in the ReW model has two dimensions: a counter, and a "partial" y, a value formed by the predictions learned by the network, both updated at each pattern training; initially all values are set to zero;
- ReW accepts binary data with exactly the size of its retina ($N*n$) as input, what normally require some kind of preprocessing to transform the input data into binary representation; each pseudo-randomly mapped group of n bits of the input retina will access the position corresponding to its values a neuron (RAM node);
- In the training phase, k pairs ($\mathbf{x}_i$, y_i) are submitted to the ReW network, and each of their corresponding addressed memory positions will have their two values updated; the counter is incremented and partial access is summed with the y_i of the example that generated the access;

ESANN 2019 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 24-26 April 2019, i6doc.com publ., ISBN 978-287-587-065-0.
Available from http://www.i6doc.com/en/.

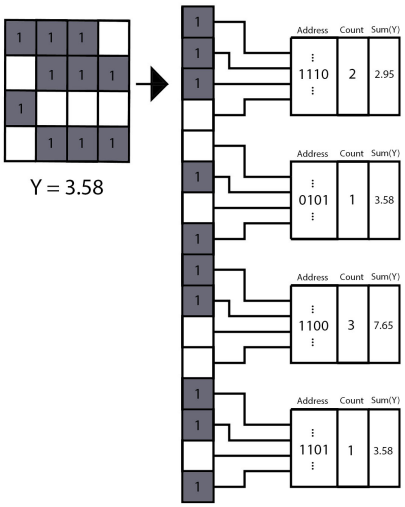

Fig. 1: Example of a ReW model behavior in the training phase. A binary input and a float value y are presented to the model. The pseudo-random mapping is applied to the binary input and the new pattern is divided into n-tuples, each one being assigned to one of the regression RAMs. The values related to the address corresponding to the tuple are updated in the following way: the counter is incremented by 1, while the summation is incremented by the value of y.

- In the prediction phase the sum of counters ($\sum c$) and partial y ($\sum y$) of the positions accessed by a given $\mathbf{x}$ are used to calculate the corresponding y; unlike the n-tuple Regression Network that uses only simple mean ($\frac{\sum y)}{\sum c}$) for this calculation, ReW can also use:
  - power mean: $(\frac{1}{n}\sum_{k=1}^{n}(y_k^{c_k}))^{(\sum c)}$
  - median: central value of $\frac{y_i}{c_i}$, with i in range [0, n])
  - harmonic mean: $(\frac{\sum_{k=1}^{n} y_i^{-1}}{n})^{-1}$
  - harmonic power mean: $\frac{n}{\sum_{i=0}^{n}\frac{c_i}{y_i}}$
  - geometric mean: $(\prod_{i=0}^{n}\frac{y_i}{c_i})^{\frac{1}{n}}$
  - exponential mean: $log(\frac{\sum_{i=0}^{n} e^{\frac{y_i}{c_i}}}{n})$
- In ReW it is possible to ignore the contribution of an address that does not satisfy a certain minimum of "0"s in its formation; the same is possible with "1"s.

ESANN 2019 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 24-26 April 2019, i6doc.com publ., ISBN 978-287-587-065-0.
Available from http://www.i6doc.com/en/.

2.3 ClusRegressionWiSARD

Inspired by ClusWiSARD, the ClusRegressionWiSARD (CReW) is a network formed by several ReWs, each with distinct mappings, but with retinas of the same size and same address size as well. In the training phase, when a pair ($\mathbf{x}_i$, y_i) is submitted to the network, $\mathbf{x}$ is presented for each ReW, which behaves as a class discriminator of the WiSARD in the classification phase, that is, each ReW will return a score obtained from the number of positions of its memories that have been accessed and have counter with value greater than zero. All ReWs with the highest scores are trained with ($\mathbf{x}_i$, y_i).

In the prediction phase, when an input $\mathbf{x}$ is submitted to the CReW, it will be sorted by each ReW and the highest score will predict its corresponding y. If there is a tie between the ReWs, the tie-break policy known as bleaching, native to WiSARD, will be used. In it a threshold initialized with value zero is incremented with each tie and a new classification occurs, being considered for the punctuation of each discriminator only the positions of memory whose counter is superior to the bleaching. If there is an absolute tie, that is, the value of the bleaching is greater than the cardinality of the training set used, a previously chosen ReW default is elected.

3 Experimental Results

3.1 Experimental setup

The data available to the competitors was divided into tree types of files: first, the training and testing files, containing 5243 and 4110 observations, respectively. Both files contains as features i) the id of the observation; ii) the id of the field the observation was planted; iii) the age of the palm tree; iv) the type of the palm tree; v) the year of harvest and vi) the month of harvest. The training file also has information regarding the target y, which is the total amount of palm oil produced by the tree. Second, a file containing information regarding the soil properties of the field in which the palm tree is planted. Finally, 28 files containing data regarding historical weather measured in each field from January 2002 to December 2007.

The initial modeling removes the id and the $field\_id$ and adds additional information from the other files. First, a time window named tw is defined in order to search weather information in a specific period of time going backwards from the month prior to the harvest of the tree. Second, all 66 features related to the soil data are added. The new observation is then composed by $2+66+8\times tw$ features.

In a second round of experiments, variations of the initial modeling were performed. One of them ignores the soil data by creating a total of 28 ReWs, each one responsible for predicting the production of trees planted in a specific field. Other variations aims to overcome the problem of the *type* feature: there are values in the testing file that are not present in the training file. These variations included the removal of the feature and the usage of one-hot encoding

ESANN 2019 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 24-26 April 2019, i6doc.com publ., ISBN 978-287-587-065-0. Available from http://www.i6doc.com/en/.

of all possible values

Since the features must be binarized, a thermometer encoding is applied. Due to the short space of time, it was not possible to perform experiments aiming for the best thermometer value for each feature. As a result, the same value was applied to all features. However, a small set of different values were used for an empirical evaluation. In addition, since a binary word of size w can me divided into different sizes of n tuples, all possible n values that are less than 32 were tested.

3.2 Analysis of the experiments

The best of RAM-based solutions reached the sixth position of 51 teams. In this section we will present the best accuracy for the solutions using ReW, CReW and ensemble of both (Table 1) and a comparison of accuracy and speed of the models proposed here and of those that had better performance in the challenge (Table 2). The experimental environment used here is a Intel Core i5 1,8 GHz with 8 GB DRR.

Method	Parameters	Mean Absolute Error
Ensemble	200 CReWs, therm = 10, tw = 20, n = [11, 17], min0 = min1 = [0, 5], limit = [3, 9], geometric mean	**0.08468**
Ensemble	55 CReWs + 55 ReWs, therm = 10, tw = 20, n = [11, 17], min0 = min 1 = [0, 3], limit = [3, 9], geometric mean	0.08514
Ensemble	200 comittees of 28 CReWs each (1 CReW per field), therm = 10, tw = 20, n = 14, limit = 6, simple mean	0.08537
Ensemble	30 ReWs + 30 CReWs, therm = 15, tw = 24, n = [15,30], min0 = min1 = [3, 7], median	0.8690
ReW	therm = 17, tw = 11, n = 17, simple mean	0.08737
CReW	therm = 10, tw = 20, n = 32, min0 = 0, min1 = 1, limit = 200	0.08938

Table 1: The best results of RAM-based approach in KDD18 Challenge (the parameters of the networks and ensembles listed here are fruits of an exhaustive search in the space of the hyperparameters); therm = number of bits used in thermometer, tw = size of time window, min0/min1 = minimum of 0s/1s required in memory position, n = number of bits per RAM, limit = maximum of ReWs per CReW

ESANN 2019 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 24-26 April 2019, i6doc.com publ., ISBN 978-287-587-065-0. Available from http://www.i6doc.com/en/.

Model	MAE	Training time	Prediction time
XGBoost	**0.07569**	4.12962484	0.08239889145
GradientBoost	0.08287	3864.08913588	0.00241994858
n-Tuple Regression	0.09211	0.0037262439727	0.000348329544
RegressionWiSARD	0.08737	**0.00035619736**	**0.00017619133**
ClusRegressionWiSARD	0.08938	0.00040984154	0.00021290781

Table 2: State-of-the-art and other relevant results; MAE = mean absolute error, the time is given in seconds.

Some considerations: when the n of a net increases, it becomes naturally more sparse, so this confidence in your answers decresease; both ReW and CReW always have low standard deviation; the output of an ensemble is obtained by the average output of all its members, using the same modalities used internally in the ReWs. It can be seen from the results of Table 2 that both proposed models had a minimal difference of the state of the art, surpassing its speed in many orders of magnitude. Experiments with ensembles with randomly generated networks showed a slight improvement in the performance of the models (as seen in Table 1), despite the natural drop in speed.

4 Conclusion

This work presented two new weightless neural networks for regression tasks based on the n-tuple Regression Network model, both competitive in terms of state-of-the-art accuracy and other results relevant to the prediction problem of productivity of palm oil of the KDD18 competition. In terms of speed in the learning and prediction phases, the two models proved to be superior to all other solutions and due to their simplicity, these networks are ideal candidates for situations that require online learning and low computational costs.

Using Regression WiSARD and ClusRegression WiSARD in other regression datasets and adding new policies to update the partial y, the possibility of different addressing sizes for the CReW discriminators, with a new decision policy adapted to different amounts of neurons and traditional ensemble strategies, such as AdaBoost, to the committee of regression RAM-based networks are immediate future works that can be cited.

References

[1] I. Aleksander, W. Thomas, and P. Bowden, *WISARD, a radical new step forward in image recognition*, Sensor Rev., 4(3), 120-124, 1984.

[2] Cardoso, D.O. et al., *Financial credit analysis via a clustering weightless neural classifier*, Neurocomputing, vol. 183, pp. 70-78, 2016.

[3] Kolcz, A. and Allinson N.M., *n-tuple Regression Network*, Neural Networks, vol. 9, pp. 855-869, 1996.

[4] Widrow, B. and Stearns, S.D., *Adaptive signal processing*. Englewood Cliffs, NJ: Prentice-Hall, 1985.

[5] https://www.kaggle.com/c/kddbr-2018/