ESANN 2022 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 5-7 October 2022, i6doc.com publ., ISBN 978287587084-1. Available from http://www.i6doc.com/en/.

A WiSARD-based conditional branch predictor

Luis A. Q. Villon[1], Zachary Susskind[2], Alan T. L. Bacellar[1], Igor D. S. Miranda[3], Leandro S. de Araújo[4], Priscila M. V. Lima[1], Mauricio Breternitz Jr.[5], Lizy K. John[2], Felipe M. G. França[1,6] and Diego L. C. Dutra[1] *

1- UFRJ, Rio de Janeiro, Brazil, 2- UT Austin, Austin, USA,
3- UFRB, Cruz das Almas, Brazil, 4- UFF, Niterói, Brazil,
5- ISCTE, Lisbon, Portugal, 6- Instituto de Telecomunicações, Portugal

Abstract. Conditional branch prediction is a technique used to speculatively execute instructions before knowing the direction of conditional statements. Perceptron-based predictors have been extensively studied, however, they need large input sizes for the data to be linearly separable. To learn nonlinear functions from the inputs, we propose a conditional branch predictor based on the WiSARD weightless neural network model and compare it with two state-of-the-art predictors: TAGE-SC-L and the Multiperspective Perceptron. We show that the WiSARD-based predictor, with a smaller input size outperforms the perceptron-based predictor by about 0.09% and achieves similar accuracy to that of TAGE-SC-L.

1 Introduction

In recent decades, different types of neural networks have been applied to address several topics in computer microarchitecture [1]. Specifically, innovative techniques for implementing branch prediction were covered using perceptron [2] [3], feedforward neural networks [4], recurrent networks and convolutional networks [5] [6].

Weightless neural networks (WNNs) are a category of neural model which use neurons called RAM nodes to perform prediction. The neurons are made up of lookup tables (LUTs) and do not perform complex arithmetic operations. The main advantage of WNNs is the capacity to learn non-linear functions of their inputs, which is not possible in a conventional weighted neural network, such as the perceptron. The WiSARD (Wilkie, Stoneham and Aleksander's Recognition Device) [7] is considered to be the first WNN to achieve a commercial success, and is the neural network model adopted in this paper.

Due to the ability to learn non-linear features indirectly represented by the inputs, the WiSARD model is an attractive alternative to traditional neural-based predictors. Nevertheless, there is no previous work, to our knowledge, that uses WiSARD in a conditional branch predictor. Consequently, this work aims to explore the potential gain of accuracy using a WiSARD-based branch predictor that requires smaller input sizes when compared to state-of-the-art predictors, the TAGE-SC-L and the Multiperspective perceptron. Our results show

*This paper was partially supported by (a) CAPES - Brazil - Finance Code 001; (b) CNPq - Brazil; (c) Fundação para a Ciência e a Tecnologia, I.P. (FCT): ISTAR Projects: UIDB/04466/2020 and UIDP/04466/2020, and; (d) FCT/COMPETE/FEDER, FCT/CMU IT Project FLOYD: POCI-01-0247-FEDER-045912.

ESANN 2022 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 5-7 October 2022, i6doc.com publ., ISBN 978287587084-1. Available from http://www.i6doc.com/en/.

that our predictor can achieve similar accuracy and even outperform both state-of-art branch predictor models, TAGE-SC-L and Multiperspective perceptron, depending on the analyzed dataset. On average, the WiSARD-based predictor is tied in accuracy with TAGE-SC-L, and outperforms the Multiperspective perceptron by 0.09%.

2 Background and related work

Branch predictor: It is an essential part of modern computer microarchitectures. Instead of stopping when a conditional branch (*if/if-else*) is encountered in the execution of a program, a processor uses branch predictor to fetch and speculatively execute instructions along a predicted path. As computer architectures become more complex and the number of instructions issued per cycle increases, the penalty for a prediction error increases [2]. Most modern branch predictors are variants of the TAGE [8] and/or perceptron branch predictors [2]. In particular, the TAGE-SC-L [9] predictor is considered the state-of-the-art in the industry [10]. This predictor uses a neural-based statistical corrector to detect some unlikely predictions and to revert them.

Neural based branch predictors: Another state-of-the-art branch predictors are the perceptron-based predictors. The first relevant work used a single-layer perceptron [2]. This work was improved later in a hashed predictor, the Multiperspective Perceptron predictor, based on the idea of viewing branch history from multiple perspectives [11]. The success of perceptron-based predictors confirms that neural networks can be useful in branch prediction for industrial applications.

WiSARD [7]: This WNN is a supervised learning model employed for classification tasks. It consists of a n-tuple classifier composed of class discriminators. Each discriminator is a set of N RAM nodes having n address lines each [12]. The learning phase consists of writing 1's in each RAM node in the respective discriminator that is designated by a hash function of the input pattern value. In the classification phase, all RAM nodes similarly designated by the input are read, and their contents are summed to produce a *response* value. The index of the discriminator with the highest response value is taken as the predicted class. To deal with the learning saturation problem, the content of the RAM nodes are implemented as an access counter which is increased at each access during the training phase (Fig. 1). The RAM node's counter must have a value higher than a threshold called bleaching [13]. On inference the output of a RAM is 1 if the addressed value is greater than the bleaching threshold, otherwise it is 0.

3 WiSARD-based predictor architecture

The WiSARD-based predictor is designed to perform one-shot online training and the respective classification phase performs a binary classification. The binary input is a linear combination of different sources of current and recent

ESANN 2022 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 5-7 October 2022, i6doc.com publ., ISBN 978287587084-1. Available from http://www.i6doc.com/en/.

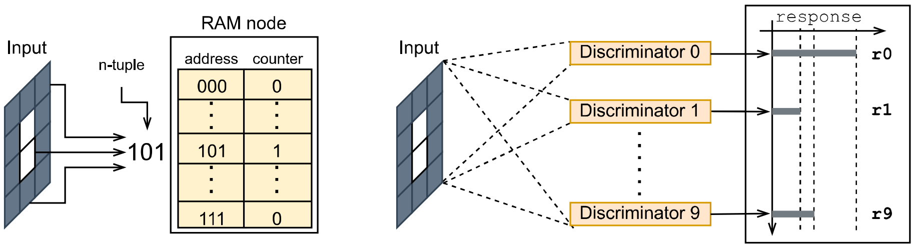

Fig. 1: A representation of the WiSARD model. In this example, the input image contains "0". An outline of the training phase is showed in the left side. In the right side, the corresponding discriminator produces the strongest response in the classification phase.

branch address information. Thus, the input can be expressed by:

$$input = a \cdot PC + b \cdot GHR + c \cdot PCxorGHR + \sum_{i=0}^{N-1} d_i \cdot LHR_i + e \cdot GPHR$$

Where: PC (program counter) represents the least significant bits from the current branch address, GHR is the global history register from the last conditional branches outcomes, $PCxorGHR$ is the xor operation for the PC and GHR, LHR_i are several local history registers from the current branch and $GPHR$ is the global path history register which stores the 8 less significant bits from the last 8 conditional branches. The additional parameters a, b, c, d and e represent the strength of a given field of the input.

The WiSARD-based predictor architecture is shown in Fig. 2. The classification phase is performed first since the predictor is designed to operate using an online learning methodology. In this phase, the current input information is pseudo-randomly divided in *n-tuple* sizes in order to get the address of a RAM node located in two discriminators: Discriminator "0" that represents a not taken branch and Discriminator "1" otherwise. A response is generated in both discriminators and the one with the higher value determines the corresponding final output. In addition, there is a bleaching implementation every time there is a tie in the classification process. As soon as the classification phase for the current input ends, it goes to the training phase, where it is split again in *n-tuple* sizes to get the address of all RAM nodes located in the respective Discriminator. In this latter process, the counters in each RAM node are updated accordingly. All this procedure, including the classification and training phase, is performed for all the following inputs of a given dataset.

4 Evaluation Methodology

Datasets description The dataset used was obtained from the Kaggle dataset (`https://www.kaggle.com/dmitryshkadarevich/branch-prediction`) which corresponds to data extracted from benchmarks of the 3rd Championship Branch

ESANN 2022 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 5-7 October 2022, i6doc.com publ., ISBN 978287587084-1. Available from http://www.i6doc.com/en/.

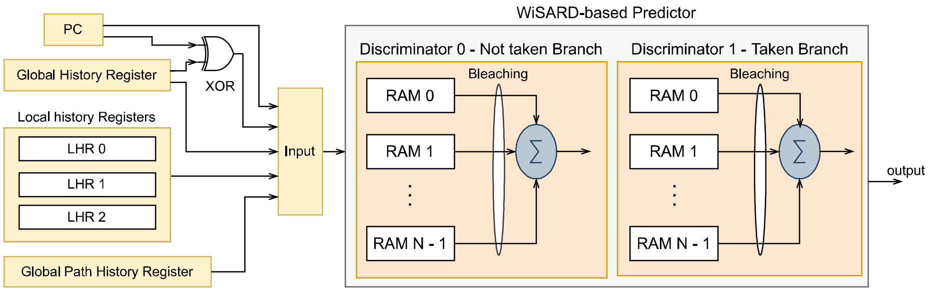

Fig. 2: A depiction of the WiSARD-based predictor. In this example there are three local history registers

Prediction (CBP-3) organized by the JILP Workshop on Computer Architecture Competitions (JWAC). The information is composed only of conditional branch information and is distributed in 3 categories, according to the benchmark application class: integer workloads (I1 and I2), multimedia (M1 and M2) and server (S1 and S2) applications. All of them have $4x10^5$ conditional branch instructions except dataset M1, which has $3x10^5$ elements.

Methodology. In order to obtain more accurate results, 100 experiments were performed on each group of data sets. Thus, the data obtained that are shown later represent the average of 100 values. Furthermore, for the purpose of comparison with the WiSARD-based predictor, we also used the TAGE-SC-L and the Multiperspective Perceptron predictors on all datasets as well. The input size for both predictors is 3127 and 2329 bits respectively, and their training and classification phase do not have a random process involved because they are final hardware architecture implementation models.

5 Results and Discussion

The best configuration we found for the parameters is: a = 24, PC = 24bits, b = 12, GHR = 24bits, c = 12, PCxorGHR = 24bits, d_0 = 8. LHR_0 = 24bits, d_1 = 8. LHR_1 = 16bits, d_2 = 8. LHR_2 = 9bits, d_3 = 6. LHR_3 = 7bits, d_4 = 12. LHR_5 = 5bits, e = 8, GPHR = 64bits. Therefore, the size of the input analyzed in this work was: $24{\cdot}24 + 12{\cdot}24 + 12{\cdot}24 + 8{\cdot}24 + 8{\cdot}16 + 8{\cdot}9 + 6{\cdot}7 + 12{\cdot}5 + 8{\cdot}64 = 2158$bits. This input size is smaller than the TAGE-SC-L and the Multiperspective Perceptron counterparts.

The results of the first experiment are shown in Fig. 3. It illustrates how the accuracy varies as the size of the n-tuple increases. Firstly, we notice that the accuracy in the datasets I1 and I2 remains almost constant in this experiment. In the datasets M1 and S1 the accuracy increases up to n-tuple size = 22 and then decreases, being dataset S1 where this effect is more pronounced. On the other hand, in the datasets S2 and M2 we see a more prominent accuracy benefit. On average (black line), the accuracy increases up to n-tuple size = 25.

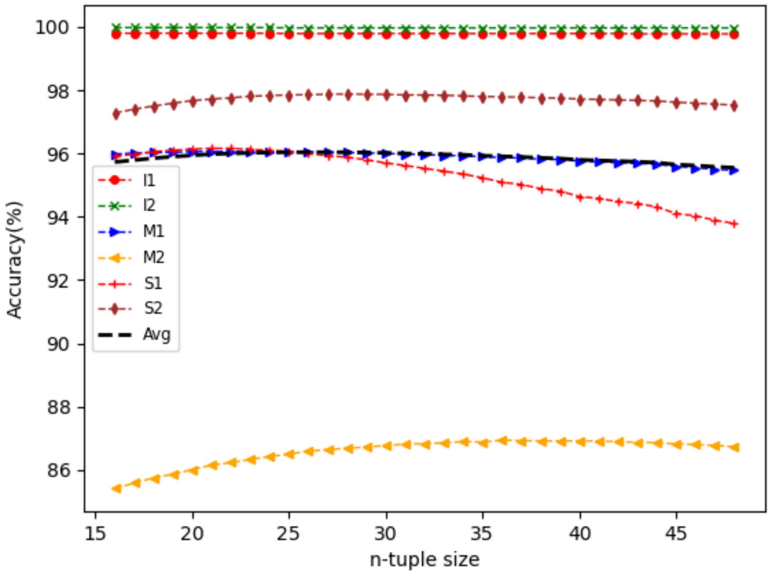

Fig. 3: Results of accuracy obtained by the WiSARD-based predictor in the classification phase as the size of the n-tuple increases. The "Avg" line represents the average accuracy of all datasets. Higher is better.

Taking this result into account, we compared it with TAGE-SC-L and Multiperspective Perceptron (shown in Table 1 where W, T and MP stand for the WiSARD-based, TAGE-SC-L and Multiperspective Perceptron predictors respectively). On average, the WiSARD-based predictor achieves approximately the same accuracy as the TAGE-SC-L and slightly outperforms the Multiperspective Perceptron in 0.09%. We emphasize that our predictor shows a higher accuracy value on the M2 dataset compared to the other predictors.

Predictor	I1	I2	M1	M2	S1	S2	Average
W	99.79±.00	99.97±.00	96.05±.02	86.50±.15	96.07±.03	97.85±.01	96.04±.06
T	99.81±.00	99.98±.00	96.14±.00	85.86±.00	96.32±.00	97.87±.00	96.00±.00
MP	99.77±.00	99.98±.00	96.23±.00	85.75±.00	96.21±.00	97.76±.00	95.95±.00

Table 1: Comparison of the best configurations of the WiSARD-based predictor with the state-of-the-art (TAGE-SC-L) and the multiperspective perceptron predictor. They were code named W, T and MP respectively.

Finally yet importantly, several experiments were performed using the same input from TAGE-SC-L and Multiperspective Perceptron in the WiSARD-based predictor. Interestingly, the best results with these inputs, on average, where 77.19% and 89.77% respectively. This implies that our predictor has a completely

ESANN 2022 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 5-7 October 2022, i6doc.com publ., ISBN 978287587084-1. Available from http://www.i6doc.com/en/.

different knowledge acquisition process than the other predictors, which was intuitively expected.

6 Conclusion and future work

In this paper we propose a conditional branch predictor based on WNNs, particularly on the WiSARD model. We experimented the WiSARD-based predictor in order to compare it with TAGE-SC-L, the state-of-the-art, and with the Multiperspective Perceptron, a neural-based predictor. Using a smaller input size our predictor achieves, on average, similar accuracies than the TAGE-SC-L and outperforms the Multiperspective Perceptron by 0.09%.

This work can be further extended by using Bloom filters [14] to allow for a compact hardware area, whilst reducing memory and power consumption with less latency, making training and classification phases faster.

References

[1] Drew D Penney and Lizhong Chen. A survey of machine learning applied to computer architecture design. *arXiv preprint arXiv:1909.12373*, 2019.

[2] Daniel A Jiménez and Calvin Lin. Dynamic branch prediction with perceptrons. In *Proceedings HPCA Seventh International Symposium on High-Performance Computer Architecture*, pages 197–206. IEEE, 2001.

[3] Daniel A Jiménez. Fast path-based neural branch prediction. In *Proceedings. 36th Annual IEEE/ACM Int Symp on Microarchitecture, 2003. MICRO-36.*, pages 243–252, 2003.

[4] Andrew Smith. Branch prediction with neural networks: Hidden layers and recurrent connections. *Dept of Comp Science UC, San Diego La Jolla, CA*, 92307, 2004.

[5] Stephen J Tarsa, Chit-Kwan Lin, Gokce Keskin, Gautham Chinya, and Hong Wang. Improving branch prediction by modeling global history with convolutional neural networks. *arXiv preprint arXiv:1906.09889*, 2019.

[6] Yonghua Mao, Huiyang Zhou, Xiaolin Gui, and Junjie Shen. Exploring convolution neural network for branch prediction. *IEEE Access*, 8:152008–152016, 2020.

[7] I. Aleksander, W.V. Thomas, and P.A. Bowden. WISARD·a radical step forward in image recognition. *Sensor Review*, 4(3):120–124, 1984.

[8] André Seznec and Pierre Michaud. A case for (partially) tagged geometric history length branch prediction. *The Journal of Instruction-Level Parallelism*, 8:23, 2006.

[9] A. Seznec. Tage-sc-l branch predictors. In *JILP-Championship Branch Prediction*, 2014.

[10] Yonghua Mao, Z Huiyang, and Xiaolin Gui. Exploring deep neural networks for branch prediction. *ECE Department, NC University*, 2017.

[11] Daniel A Jiménez. Multiperspective perceptron predictor. In *5th JILP Workshop on Computer Architecture Competitions: Championship Branch Prediction (CBP-5)*, 2016.

[12] Leopoldo AD Lusquino Filho, Luiz FR Oliveira, Aluizio L Filho, Gabriel P Guarisa, Priscila MV Lima, and Felipe MG França. Prediction of palm oil production with an enhanced n-tuple regression network.

[13] Bruno PA Grieco, Priscila MV Lima, Massimo De Gregorio, and Felipe MG França. Producing pattern examples from "mental" images. *Neurocomputing*, 73(7-9):1057–1064, 2010.

[14] Leandro Santiago, Leticia Verona, Fabio Rangel, Fabricio Firmino, Daniel S Menasché, Wouter Caarls, Mauricio Breternitz Jr, Sandip Kundu, Priscila MV Lima, and Felipe MG França. Weightless neural networks as memory segmented bloom filters. *Neurocomputing*, 416:292–304, 2020.