Implementation of a Multi-Layer Perceptron Neural Networks in Multi -Width Fixed Point Coding

A.G. Blaiech, K. Ben Khalifa, M. Boubaker, and M.H. Bedoui

***Abstract*—The artificial neural networks (ANN) have emerged as interesting approaches in various fields of research and industry. ANNs are able to solve non-linear and complex problems where the classic methods do not provide solutions. These learning algorithms are particularly interesting as they are implemented in embedded systems. In order to obtain an efficient implementation, a compromise of time and area is needed**

In this paper, we will develop a methodology for automatic coding of fixed-point data for the implantation of artificial neural networks that are typically specified as floating-point. This methodology should determine the optimal encoding of various blocks of our ANN, to maximize accuracy and minimize the application area. The proposed methodology allows the automatic generation of VHDL code within an encoding in multi-width of multi-layer perceptron (MLP) model in the decision phase after a simulation in the learning phase with fixed point operators. This methodology has allowed a gain of 6% in LUTs compared to a single coding accuracy with the same network topology.

***Index Terms*—Floating point, fixed point, implementation, RNA, simulation, VHDL.**

I. Introduction

Several connectionist methods with supervised and non-supervised learning have been used to quantify the vigilance state starting from processing the electroencephalographic(EEG) signal derivation collected at the cortex of the subject [1],[2]. In our work, we have used multi-layer perceptron (MLP) in the decision phase and in the learning phase as a tool to classify the vigilance states. The approach proposed in this paper is a continuation of this work with the aim of proposing an entirely embedded real-time detection of reduced vigilance. In this context, we are interested in the issue of implementing this model on a neural circuit based on Field Programmable Gate Arrays (FPGA).

Various approaches to implementations of the MLP on FPGAs have been carried out [3],[4].Several authors have looked at the implementation phase of the MLP decision [5],[6] and in the learning phase [7]. Considering the complexity of these two phases, algorithms, which is generally proportional to the increase in the number of neurons and layers, an optimization methodology of implementation is necessary. In [7], the authors developed a methodology for optimizing the latency architectures described in VHDL through four stages: the topology generation, the definition of accuracy and ranges of values, the choice of arithmetic representation (fixed or floating point) and selecting the type of learning. In [6] the authors developed a methodology for optimizing the operating surfaces where they used serial and parallel operators, coded in fixed point, which fit with the nature of the spread of data pipelining: inter-neuron and intra-neuron. All these methods exploit a single width for all variables and all operations and hence result in a non-optimal implementation of operators and generates an overflow at the level of operators .To remedy this drawback, we propose to exploit a methodology for optimizing multi-width precision.

This paper is organized as follows: In the first section, we describe the optimization methodology which aims to reduce the size of the information while retaining the performance of the MLP model. In the second and final part, we present the results of the implementation of an MLP on a FPGA by exploiting our methodology.

II. Proposed Methodology

Our goal is to propose a methodology for optimizing the implementation of an artificial neural network (ANN) of an MLP type .We direct our research towards the implementation of limited accuracy operators by minimizing the size of the data without destroying the performance of RNA. After determining the corpus of work and the topology of our MLP, we have carried out our methodology in two phases. First, we have passed a simulation phase to convert the learning algorithm, initially coded in a floating point, to fixed point coding. Second, we have used an implementation phase, trying to collect all information necessary for our MLP (Topology, accuracy ...) and generate the corresponding VHDL code. Figure 1 illustrates this methodology.

A. Corpus work and topology of the MLP

To assess our methodology, we have used a work body based on data extracted from signals of the EEG recorded in real conditions. These signals characterize physiological states related to the vigilance variation.

The topology of the MLP adopted throughout our work is formed by 21 neurons in the input layer, 10 neurons in the hidden layer and 2 neurons in the output layer. We have

Manuscript received April 10, 2012; revised May 12, 2012.

A. G. Blaiech. is now with the TIM Team, Laboratory of Biophysics, Faculty of Medicine of Monastir, University of Monastir, 5019 Tunisia. (e-mail:ahmedghaziblaiech@yahoo.fr)

K. Ben Khalifa is now with TIM Team, Laboratory of Biophysics, Faculty of Medicine of Monastir, University of Monastir, 5019 Tunisia(e-mail: khaled.benkhalifa@issatso.rnu.tn)

M. Boubaker is now with TIM Team, Laboratory of Biophysics, Faculty of Medicine of Monastir, University of Monastir, 5019 Tunisia(e-mail: Boubaker_mohamed@yahoo.fr)

M. H. Bedoui is now withTIM Team, Laboratory of Biophysics, Faculty of Medicine of Monastir, University of Monastir, 5019 Tunisia (e-mail: medhedi.bedoui@fmm.rnu.tn)

used the hyperbolic tangent as a transfer function in the hidden and output layers while a linear function is adopted in the input layer.

B. Simulation phase

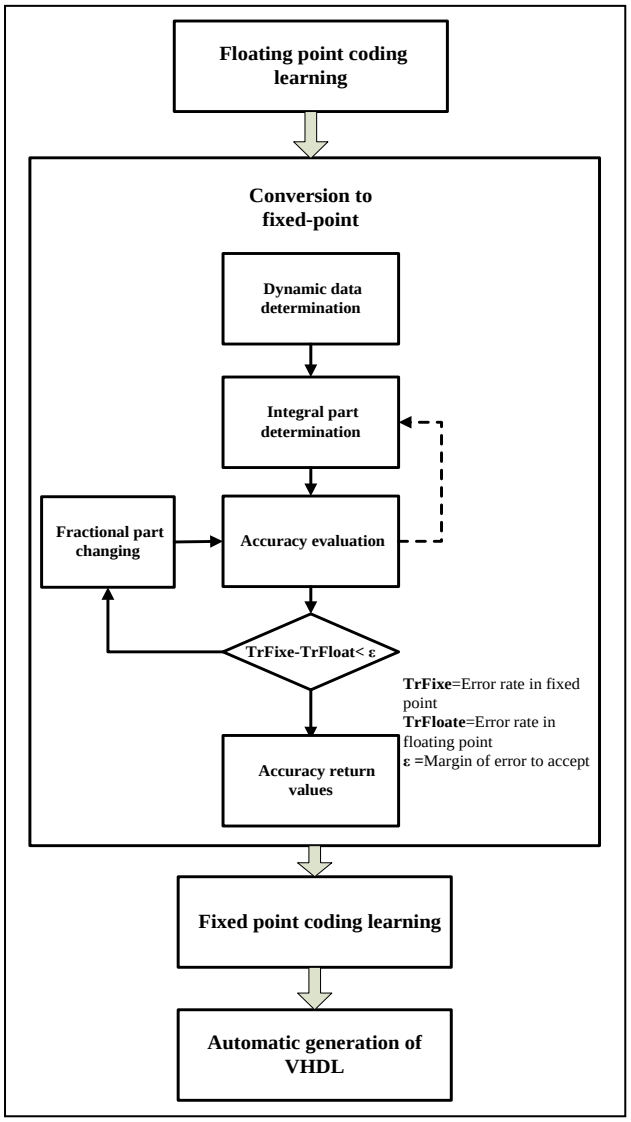

Fig. 1. Methodology proposed

The majority of researches are directed towards the study of the ANN implementation in single precision. This impedes the accuracy optimization because generally fixed and single width coding of all information is set. To overcome these limits, we propose to use a multi-width coding. In our MLP, we have reserved a precision format for entries in the first layer and formats for weights, multipliers and adders for the hidden and output layers. The hyperbolic tangent transfer function has been simulated by a CORDIC algorithm [8],[9] .To make the conversion of the learning algorithm, initially coded as a floating point algorithm, into a fixed point algorithm, we have to estimate in the first place the dynamic data enough to gain the size of the integer part and evaluate the accuracy. Then, we should be able to determine the size of the fractional part.

1) Dynamic data estimation

The dynamics of a given data correspond to the interval containing all values taken by this data over time. To determine the dynamic variables, two approaches can be used: statistical approaches, based on the analysis of simulation results of the application [10],[11],[12]; and analytical approaches, based on a static analysis of the application [13],[14],[15],[16]. In our methodology, we have opted to use a statistical approach in calculating the maximum value of inputs, weights, adders and multipliers in floating-point encoding.

2) Integer part determination

This first phase is to determine the minimum size of the integer part. The decimal position used for a variable x depends on the stretch estimated for this variable. Knowing the extent $[\underline{x}, \overline{x}]$ to guarantee the absence of overflow, the decimal position is equal to:

$$m_x = \min_{k \in z} k \mid 2^k > \max(-\underline{x}, \overline{x}) = \lceil \log_2 \max(|x|) \rceil \quad (1)$$

In other word, we look for the smallest integer k in a way that 2^k is larger in absolute value at the terminals of the variable x. Subsequently, we add one bit for negative numbers (two's complement).

Given that the statistical method used in estimating the dynamics of the data does not guarantee the overflow, we have integrated indicators defined as variables that control the excesses within the operators and the intermediate signals. These indicators control the overflow dynamically. The determination of the integral part of the data is described by the algorithm 1.

Algorithm 1: *Determination of the integral part*

1. Estimate the dynamic data;
2. Determine the integer part;
3. Assess accuracy of fixed point coding with the absolute error rate of success, which shows the performance of our algorithm;
4. If there is an overflow back to 2 ,else go to 5;
5. Return the width of the integer part.

3) Accuracy evaluation

The use of fixed point arithmetic limits the accuracy of the calculations and leads to an error quantization in output calculations for the difference between the signal at infinite and finite precisions. In addition, to reduce the hardware cost, truncations of data must be made at different points to further increase the imprecision. Indeed, some operations, such as multiplication, increase the number of output bits of the operation. Thus, calculation accuracy should be assessed basing on data widths to ensure that it exceeds a threshold fixed by the user and corresponding to the precision constraint. Several criteria exist to assess the accuracy of a fixed-point implementation.

The absolute error calculated by the difference between success rates in fixed point coding and floating-point coding and where it reflects the performance of our MLP, will be the test of our application. To determine the noise power output of the algorithm for a specific instance of data formats, two approaches can be used. First, the statistical approach is to determine the statistical parameters of the quantization error from the simulation of the algorithm in fixed point and floating point [17]-[19] The second analytical approach determines the analytical expression of noise power by propagating a noise model, depending on the sizes within the flow graph of the algorithm [20]-[22]. In our methodology, we have chosen using a statistical method to assess the accuracy. Since our methodology takes account of

different widths, we have used a method called "adding one bit for all method", where we add a bit of fractional part for all blocks (inputs, weights, adders, multipliers) to obtain the desired performance, This method ensures good performance of the network. Algorithm 2 describes a method.

Algorithm 2: *Adding one bit for all algorithm*

1. Add one bit to the fractional part of the entries in the input layer and the weights adders and multipliers in the hidden and output layers;
2. Assess the accuracy of the fixed point coding with the absolute error rate of success, which shows the performance of our algorithm;
3. If the result of evaluation is acceptable compared to a predefined margin of error, go to 4 ;else, return 1;
4. Return the width of the fractional part.

4) Simulation results

The proposed methodology consists in converting the learning algorithm, initially coded in floating point into an algorithm coded in multi-width fixed point encoding

The simulation phase has allowed more details of the entries, weights, multipliers and adders for each layer. The simulation results are given in Table I

Based on the hypothesis which says that uniformed width coding, which, consists in adopting the maximum width specified by the multiple widths encoding for the different blocks of our MLP, then we can consider a gain of 10 bits over a single-coding accuracy.

TABLE I MULTIPLE WIDTHS FOR THE DIFFERENT COMPONENTS OF RNA

Layers	Specifications	Integer part	Fractional part	Total Width
Input layer	Inputs	4	12	16
Hidden layer	Weight	6	12	18
	Multipliers	4	12	16
	Adders	6	12	18
Output layer	Weight	3	12	15
	Multipliers	3	12	15
	Adders	6	12	18

C. MLP Implementation in FPGA

The last step of our optimization methodology of coding accuracy is to implement our MLP phase decision on an FPGA VHDL code generator. In this step, we try to collect all information necessary for our MLP (Topology, accuracy ...) and generate the corresponding VHDL. The hyperbolic tangent transfer function has implemented using the CORDIC method [8],[9].

III. IMPLEMENTATION RESULTS

We have synthesized and implemented our MLP (21 input neurons, 10 hidden neurons, 2 output neurons) on the FPGA Virtex-E XCV 3200E in multi-width and uniform width fixed point coding.

Table II presents the results of the implementation of an MLP on FPGAs by exploiting our methodology compared to the uniform width implementation.

TABLE II RESULTS OF IMPLEMENTATIONS OF MLP IN UNIFORM WIDTHS AND MULTIPLE WIDTHS CODING

Logical Resources	Multiple Widths		Uniform Width		Gain (%)
	Used	*Occupation (%)*	*Used*	*Occupation (%)*	
I/O primitives	372	-	414	-	11
IBUF	336	-	378	-	12
OBUF	36	-	36	-	0
Total LUTS	28472	43	30035	46	6

This table shows the gain of resources generated by a fixed-point coding in multiple widths compared to uniform widths to the inputs, the output signals and the total LUTs.

IV. CONCLUSION

In this work, we presented a methodology for automatic coding of fixed-point data for the implementation of artificial neural networks. This methodology has determined the optimal coding of the various blocks of our RNA, to maximize accuracy and minimize the size of the application without destroying the network performance. The proposed methodology has allowed the automatic generation of VHDL coding with multiple widths of the MLP model in the decision phase, after a simulation in the learning phase using fixed point operators. By adopting this methodology, we have considered a gain of 6% in LUTs compared to a single coding accuracy with the same network topology.

REFERENCES

[1] A. Vuckovic, V. Radivojevic, AC. Chen, and D. Popovic, "Automatic recognition of alertness and drowsiness from EEG by an artificial neural network," *Med Eng & Phys.* 2002 Jun; 24(5), p.349-60.

[2] K. Ben Khalifa F. Alexandre, N. Kerkeni, and M. H. Bedoui, "Artificial neural networks to extract knowledge from eeg," In *The IASTED International Conference on Biomedical Engineering - BioMED2005*, Innsbruck, Austria, Feb 2005.

[3] M. Moradi, M. A. Poormina, and F. Razzazi, "FPGA Implementation of Feature Extraction and MLP Neural Network Classifier for Farsi Handwritten Digit Recognition," in *Third UKSim European Symposium on Computer Modeling and Simulation* , Athens, Greece,2009.

[4] D. Anguita ,S. Bencetti ,A. De Gloria ,G. Parodi ,D. Ricci, and S. Rieella ."FPGA Implementation of High Precision Feedforward Networks," Department of Biophysical and Electronic Engineering .University of Genova ,Italy.p-103-106 ,2006

[5] C. Torres and B. Girau, "A. Gauffriau .Hardware codesign for embedded implementation of Neural Networks," *Springerlin*k, vol. 4419,P167-178, Juin 2007

[6] S. Vitabile, V. Conti, F. Gennaro, and F. Sorbello .I.CA.R. "Efficient MLP Digital Implementation on FPGA," Italien National Research Concil viale dele Scienze ,90128 Palermo,Italy and Dipartimento Ingegneria Informatica-University of Palermo,viale delle Scienze ,90128 Palermo ,Italy,2005.

[7] W. Savich, M. Moussa, and S. Areibi," The impact of arihmetic representation on implementing MLP-Bp on FPGAs,A study," *IEEE transactions on neural networks*,vol.18,no.1,January 2007.

[8] H. Lin and H. Sips, "On-Line CORDIC Algorithms," *IEEE Transactions on Computers*, C-39(9) :1038-1052, August 1990.

[9] K. Meher, J. Valls, T-Bing Juang, K. Sridharan, and K. Maharatna *50 Years of CORDIC: Algorithms, Architectures, and Applications*,vol. 56, no. 9, September 2009.

[10] K. Kum. Kim and W. Sung, "Fixed-Point Optimization Utility for C and C++ Based Digital Signal Processing Programs," *IEEE Transactions on Circuits and Systems II*, vol. 45, no 11, no. 1998

[11] R. Cmar, L. Rijnders, P. Schaumont, and I. Bolsens, "A Methodology and Design Environment for DSP ASIC Fixed Point Refinement ," in *Proceedings of the Design Automation and Test in Europe Conference (DATE 99)*, p. 271–276, Munich, 1999.

[12] K. Kum, J. Kang, and W. Sung, "AUTOSCALER for C: An optimizing floating-point to integer C program converter for fixed-point digital signal processors," *IEEE Transactions on Circuits and Systems II - Analog and Digital Signal Processing*, vol. 47, p. 840–848, September 2000.

[13] A. Benedetti and P. Perona, "Bit-width optimization for configurable dsp's by multi-intervalanalysis," in *Signals, Systems and Computers, 2000. Conference Record of the Thirty-Fourth Asilomar Conference*, vol. 1, p. 355 359, Pacific Grove, CA, USA, oct. 2000

[14] M. Stephenson, J. Babb, and S. Amarasinghe, "Bidwidth analysis with application to silicon compilation," in *Proceedings of the ACM SIGPLAN 2000 conference on Programming language design and implementation*, p. 108–120, ACM Press, 2000.

[15] L. H. de Figueiredo, and J. Stolfi, "Affine Arithmetic: Concepts and Applications ," *Numerical Algorithms*, vol. 37, no 1, p. 147–158, 2004.

[16] R. Kearfott, "Interval Computations : Introduction, Uses, and Resources ," *Euromath Bulletin*, vol. 2, no 1, p. 95–112, 1996.

[17] S. Kim, K. Kum, and W. Sung, "Fixed-Point Optimization Utility for C and C++ Based Digital Signal Processing Programs," in *Workshop on VLSI and Signal Processing '95*, Osaka, Nov. 1995.

[18] M. Coors, H. Keding, O. Luthje, and H. Meyr, "Integer Code Generation For the TI TMS320C62x," in International Conference on *Acoustics, Speech and Signal Processing 2001 (ICASSP 01)*, Sate Lake City, US, May 2001.

[19] D. Coster, M. Ade, R. Lauwereins, and J. Peperstraete, "Code Generation for Compiled Bit True Simulation of DSP Applications ," in *Proceedings of the 11th International Symposium on System Synthesis (ISSS 98*), Taiwan, December 1998.

[20] S. Wadekar, and A. Parker, "Accuracy Sensitive Word-Length Selection for Algorithm Optimization," in *IEEE/ACM International Conference on Computer Design (ICCAD'98*), p. 54– 61, San Jose, CA , USA, nov. 1998.

[21] D. Menard, and O. Sentieys, "Automatic Evaluation of the Accuracy of Fixed-point Algorithms," in *IEEE/ACM conference on Design, Automation and Test in Europe 2002* (DATE-02), p. 529–535, Paris, France, mars 2002.

[22] G. A. Constantinides, P. Y. K. Cheung, and W. Luk, "Synthesis and Optimization of DSP Algorithms," Kluwer Academic, 2004.

A. G. Blaiech completed his Masters in computer real time and actually is a PHD student, attached to the Laboratory of Biophysics, Faculty of Medicine of Monastir, University of Monastir, Tunisia. He is an assistant at the ISIMG of GABES, Tunisia. He has five paper published, he interested to study the impact of arithmetic representation for optimized implementation in FPGA.

K. B. Khaled obtained his doctorate since 2006. He is attached to the Laboratory of Biophysics, Faculty of Medicine of Monastir, University of Monastir, Tunisia. He is an assistant professor at the ISSATs of Sousse, university of Sousse,Tunisia.

M. Boubaker obtained his doctorate since 2009. He is attached to the Laboratory of Biophysics, Faculty of Medicine of Monastir, University of Monastir, Tunisia. He is an assistant professor at the ISIM of Monastir, university of Monastir, Tunisia.

M. H. Bedoui received his PhD degree from Lille University in 1993. He currently teaches with the position of Professor of biophysics in the Faculty of Medicine of Monastir (FMM), Tunisia. He is a member of Medical Technology and image processing team (TIM), UR 08-27. His research interests are real-time and embedded systems, image & signal processing and hardware/software design in medical field, electronic applications in biomedical instrumentation. He is the president of the Tunisian Association of Promotion of Applied Research**.**