Computers and Electrical Engineering 51 (2016) 139–156

Contents lists available at ScienceDirect

Computers and Electrical Engineering

journal homepage: www.elsevier.com/locate/compeleceng

Hardware implementation of real-time Extreme Learning Machine in FPGA: Analysis of precision, resource occupation and performance☆

Jose V. Frances-Villora[a], A. Rosado-Muñoz[a,*], José M. Martínez-Villena[a], Manuel Bataller-Mompean[a], Juan Fco. Guerrero[a], Marek Wegrzyn[b]

[a] GPDD. Department Electronic Engineering, School of Engineering, University of Valencia, Av. Universitat, s/n. 46100 Burjassot, Valencia. Spain

[b] Institute of Computer Engineering and Electronics, University of Zielona Góra, ul. Podgórna 50, 65-246 Zielona Góra, Poland

ARTICLE INFO

Article history:
Received 12 March 2015
Revised 7 February 2016
Accepted 8 February 2016
Available online 28 February 2016

Keywords:
FPGA
Extreme Learning Machine - ELM
Neural network training
Neural network hardware
On-chip machine learning
Embedded systems

ABSTRACT

Extreme Learning Machine (ELM) proposes a non-iterative training method for Single Layer Feedforward Neural Networks that provides an effective solution for classification and prediction problems. Its hardware implementation is an important step towards fast, accurate and reconfigurable embedded systems based on neural networks, allowing to extend the range of applications where neural networks can be used, especially where frequent and fast training, or even real-time training, is required. This work proposes three hardware architectures for on-chip ELM training computation and implementation, a sequential and two parallel. All three are implemented parameterizably on FPGA as an IP (Intellectual Property) core. Results describe performance, accuracy, resources and power consumption. The analysis is conducted parametrically varying the number of hidden neurons, number of training patterns and internal bit-length, providing a guideline on required resources and level of performance that an FPGA based ELM training can provide.

1. Introduction

The Extreme Learning Machine (ELM) algorithm has revived the interest in Single Layer Feedforward Neural Networks (SLFN) since it eases the neural network training procedure. Hardware implementations have explored other computing paradigms, [1–3], but ELM makes SLFN possible in applications where it was not allowed before due to computational requirements, especially in on-line systems with real-time data processing and tight timing constraints.

ELM algorithm is rooted in the previous idea defined as Random Vector Functional Link (RVFL) networks, [4], but introduces new concepts and advantages that outperforms RVFL, [5]. Thus, ELM is based on random hidden layer weights and a linear adjustment for the output layer. As advantages, it provides similar or better results in generalization performance than previous iterative training methods without the need for manually adjusting parameters to find the best behavior. Further, ELM can use a wide range of activation functions, including those piece-wise linear. Finally, it is remarkable the combination of reduced computation requirements and extremely fast learning speed, surpassing previous training algorithms.

☆ Reviews processed and recommended for publication to the Editor-in-Chief by Associate Editor Dr. R. Cumplido.

* Corresponding author. Tel.: +34 963543808; fax: +34 963544353.

E-mail addresses: jose.v.frances@uv.es (J.V. Frances-Villora), alfredo.rosado@uv.es (A. Rosado-Muñoz), J.Miguel.Martinez-Villena@uv.es (J.M. Martínez-Villena), manuel.bataller@uv.es (M. Bataller-Mompean), juan.guerrero@uv.es (J.Fco. Guerrero), M.Wegrzyn@imei.uz.zgora.pl (M. Wegrzyn).

http://dx.doi.org/10.1016/j.compeleceng.2016.02.007
0045-7906/

The reduced and fixed training time is one of the ELM properties that has contributed to its wide application in different fields, especially in battery operated devices where training time is a key factor. ELM has been successfully applied in the power energy area for failures and overload detection, in electricity distribution lines, and electricity price prediction. Other applications are related to biometrics, soldering inspection, or very fast object detection. In the field of control systems, ELM was already applied to real-time computation of friction in automobiles, new control strategies in non-linear control, or non-linear behavior compensation in fiber optic communications. Additional applications are reported for bank client classification, multi-categories classification applications, discrimination of evoked potentials, [6], or epileptic EEG detection, amongst others.

Moreover, the characteristic fixed and tight temporal requirements enable ELM to land in the hardware implementation arena to obtain efficient classifiers, usually implementing the forward phase. As an example, an FPGA implementation of the neural network forward phase was proposed in [7], which training phase was calculated on a PC. Others approaches have used VLSI techniques, [8]. However, its temporal requirements make ELM a good candidate to drive also on-chip training in real-time applications.

There is a current trend to implement on-chip learning for pattern recognition, facial recognition and complex learning behaviors. As an example, [9] proposed an on-line learning ANN performing a model training every new arriving data, to monitor and forecast the indoor temperature in a smart home, improving the energy efficiency of the HVAC (Heating, Ventilating and Air Conditioning); [10] focused the problem of sequential learning in mobile devices in real-time by a real-time learning algorithm for face recognition related applications; or [11], that proposed a real-time learning of neural networks to the prediction of future opponent robot coordinates in a specific environment.

The interest in hardware implementation of ELM training arises in the applications where input dataset does not increase excessively during the application, as in DBS (Deep Brain Stimulation) surgery, [12]. However, much more potential real-time applications can be driven by an on-line sequential learning algorithms as OS-ELM, which is a variation of basic ELM, and that anyway needs the implementation of the basic ELM training for initialization. Diverse OS-ELM sequential learning applications have been proposed to the date, as an example [5] adapted a gesture recognition model to new users automatically, getting high recognition accuracy.

As the expansion of on-chip learning systems is just beginning, and ELM is a good candidate to implement it, then arises the research question of how many resources usage the ELM training algorithm need, and which performance can it provide on current FPGA devices. Answering these questions parametrically (it is as a function of parameters like word width, number of training patterns or number of hidden neurons) would offer a quick sight to the needs and requirements of any application to be implemented on FPGA using on-chip ELM learning.

To answer this questions, we propose a hardware implementation of an SLFN, including ELM training, and the analysis of the performance, accuracy, resource occupation and power consumption that an FPGA implementation demands. The analysis was conducted parametrically varying the number of hidden neurons, number of training patterns and bit-length. The subject of the analysis was three proposed hardware implementations, all three under FPGA with a fixed-point implementation of the ELM training of a generic SLFN neural network. At the end, the results provide a guideline on expected resources, accuracy and performance that an FPGA implementation can provide for ELM batch training; proposes different hardware implementations for ELM training, describing a customizable core to fit multiple applications by simple parameter modification; and allows to compare between computational methods and parallelizing approaches, balancing among accuracy, logic occupation and performance. The results are applicable to any classification or regression application based on an SLFN neural network, providing a guideline on required resources and level of performance that an FPGA-based ELM learning can demand.

This paper is organized as follows. Details of the proposed architectures and computational methods are described in Section 2. Section 3 describes the hardware structures. Results of the analysis and discussion are presented in Sections 4 and 5, respectively. Finally, Section 6 concludes the paper.

2. Extreme Learning Machine (ELM) training for artificial neural networks

The basic principle of ELM is that parameters of hidden nodes (the input weights and biases for additive hidden nodes or kernel parameters) need not be traditionally tuned by conventional learning algorithms (e.g. gradient descent-based method). The hidden nodes parameters can be randomly assigned. Right after, the output weights linking the hidden layer to the output layer can be analytically determined through simple generalized inverse operation of hidden layer output matrices [13].

We briefly delineate the principles of the ELM algorithm (Section 2.1) and explain and justify the computation method selected (Section 2.2).

2.1. Principles of ELM algorithm

Let $\mathcal{D} = (\mathbf{x}_i, \mathbf{o}_i)$; i = 1, . . . , N, be a set of N patterns where $\{\mathbf{x}_i\} \in \mathbb{R}^{d_1}$ are the input and $\{\mathbf{o}_i\} \in \mathbb{R}^{d_2}$ the output data for the training set, so that the goal is to find a relationship between $\{\mathbf{x}_i\}$ and $\{\mathbf{o}_i\}$. If there are M nodes in the hidden layer, the