Neural Networks 79 (2016) 128–140

Contents lists available at ScienceDirect

Neural Networks

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

FPGA implementation of neuro-fuzzy system with improved PSO learning

Cihan Karakuzu [a,*], Fuat Karakaya [b], Mehmet Ali Çavuşlu [c]

[a] Bilecik Şeyh Edebali University, Faculty of Engineering, Department of Computer Engineering, Gülümbe Campus, 11210, Bilecik, Turkey
[b] Niğde University, Faculty of Engineering, Department of Electrical and Electronics Engineering, Central Campus, 51240, Niğde, Turkey
[c] Koc Information and Defence Technologies Inc., METU Technopolis, ODTÜ-Teknokent, 06800 Ankara, Turkey

HIGHLIGHTS

- A novel approach for implementation of neuro-fuzzy system on FPGA is proposed.
- Hardware implementation includes meta-heuristic learning of the system.
- A novel membership function implementation approach is also presented.
- The proposed approaches have been experimentally tested for both benchmark and practical problems.
- The obtained results show that the proposed method is effective and acceptable in handling various types of real life problems.

ARTICLE INFO

Article history:
Received 18 August 2015
Received in revised form 20 December 2015
Accepted 8 February 2016
Available online 8 April 2016

Keywords:
Neuro-fuzzy network
FPGA implementation
VHDL
System identification
Metaheuristic learning

ABSTRACT

This paper presents the first hardware implementation of neuro-fuzzy system (NFS) with its metaheuristic learning ability on field programmable gate array (FPGA). Metaheuristic learning of NFS for all of its parameters is accomplished by using the improved particle swarm optimization (iPSO). As a second novelty, a new functional approach, which does not require any memory and multiplier usage, is proposed for the Gaussian membership functions of NFS. NFS and its learning using iPSO are implemented on Xilinx Virtex5 xc5vlx110-3ff1153 and efficiency of the proposed implementation tested on two dynamic system identification problems and licence plate detection problem as a practical application. Results indicate that proposed NFS implementation and membership function approximation is as effective as the other approaches available in the literature but requires less hardware resources.

1. Introduction

Neuro-fuzzy system (NFS) is an intelligent technique for system control (Oğuz & Güney, 2010), identification/modelling and prediction (Chen, 2013). It is generally trained by algorithms based on gradient. But these algorithms have disadvantages such as needing complex gradient computations and getting stuck at a local minima.

Particle swarm optimization algorithm (PSO) (Kennedy & Eberhart, 1995), inspired by behaviour of animal swarms, has stochastic global search feature and it has been successfully used for neuro-fuzzy (Chen, 2013; Ghomsheh, Shoorehdeli, & Teshnehlab, 2007) and neural network (Çavuslu, Karakuzu, & Karakaya, 2012) training to solve nonlinear problems. Field programmable gate arrays (FPGAs), capable of parallel processing, have been extensively used in real time applications with heavy computational load (Martinez, Toledo, Fernandez, & Ferrandez, 2008). Recently, FPGAs gain more and more importance as a preferable embedded system platform in many neural, fuzzy and neuro-fuzzy applications (Abdelmoula, Rouabeh, & Masmoudi, 2012; Blake & Maguire, 1998; Campo, Echanobe, Bosque, & Tarela, 2008; Çavuslu et al., 2012; Çavuslu, Karakuzu, & Sahin, 2006; Çavuslu, Karakuzu, Şahin, & Yakut, 2010; Chou, Kung, Vu Quynh, & Cheng, 2013; Echanobe, del Campo, & Bosque, 2008; Glackin, Maguire, & McGinnity, 2004; Hima, Anitha, & Jegan, 2012; Huang, Pan, Zhou, & Chang, 2014; Juang & Hsu, 2005; Lin & Lee, 2009; Lin & Tsai, 2008; McKenna & Wilamowski, 2001; Pande, Paikrao, Chaudhari, 2013; Sanchez-Solano, Cabrera, Baturone, Moreno-Velo, & Brox, 2007; Tamas & Brassai, 2015).

* Corresponding author.
E-mail addresses: cihan.karakuzu@bilecik.edu.tr (C. Karakuzu), fkarakaya@nigde.edu.tr (F. Karakaya), ali.cavuslu@kocsavunma.com.tr (M.A. Çavuşlu).

http://dx.doi.org/10.1016/j.neunet.2016.02.004

Table 1
Comparison table of this study and the related literature.

Ref.	NFS type	Training type	Training algorithm	Membership function type/approximation	Number format
Blake and Maguire (1998)	RBFN	Off-line	–	Gaussian function approximation	Fixed point
Glackin et al. (2004)	ANFIS, DFNN	Off-line	–	Functional approximation given in Jang (1993)	Fixed point
Echanobe et al. (2008)	ANFIS	Off-line	–	Triangular	Fixed point
Tamas and Brassai (2015)	NFC	Off-line	–	Functional approximation given in Jang (1993)	Integer
Huang et al. (2014)	ANFIS	Off-line	Backpropagation algorithm	Triangular	–
Pande et al. (2013)	ANFIS	Off-line	–	Triangular	Fixed point
Abdelmoula et al. (2012)	ANFIS	Off-line	–	LUT	Floating point
Chou et al. (2013)	RBF	Off-line	–	Triangular and description of exponential function using Taylor series	Signed number
Juang and Hsu (2005)	RFC	Semi-trained on FPGA	Online gradient descent learning	LUT	–
Hima et al. (2012)	ANFIS	On FPGA	Error correction	Second order approximation	Floating point
Campo et al. (2008)	ANFIS	On FPGA	LSE and gradient descent	Triangular	–
Lin and Lee (2009)	RNFN	On FPGA	Modified simultaneous perturbation	Gaussian function approximation	Fixed point
Lin and Tsai (2008)	WNN	On FPGA	PSO	LUT approximation based on Taylor series	Fixed point
This study	NFS	On FPGA	Improved PSO	Functional approximation given in Jang (1993) and new proposed function approximation	Floating point

In the literature, some studies related to implementation of neuro-fuzzy systems on FPGA were presented. These studies can be divided into two groups: implementation of neuro-fuzzy networks with offline training (Abdelmoula et al., 2012; Blake & Maguire, 1998; Chou et al., 2013; Echanobe et al., 2008; Glackin et al., 2004; Huang et al., 2014; Pande et al., 2013; Tamas & Brassai, 2015) and with online training (Campo et al., 2008; Hima et al., 2012; Juang & Hsu, 2005; Lin & Lee, 2009; Lin & Tsai, 2008).

In Blake and Maguire (1998), Blake et al., RBFN type neuro-fuzzy network's parameters have been adjusted by training in a software environment and then RBFN has been implemented on FPGA with second order nonlinear Gaussian function approximation and fixed point number format representation. Glackin et al. (2004) implemented ANFIS and DFNN networks, trained in a software environment, on FPGA. They used a functional approximation given in Jang (1993) for Gaussian membership functions (MFs) and fixed point number format for data representation. Echanobe et al. (2008) implemented offline trained ANFIS with triangular MFs on FPGA using fixed point number format. Tamas and Brassai (2015) implemented a neuro-fuzzy controller Intellectual Property (IP) core. They used a functional approximation given in Jang (1993) for Gaussian membership functions (MFs) and integer number format for data representation. Huang et al. (2014) implemented ANFIS on FPGA and they used triangular membership functions to keep stable the temperature in the chamber. Pande et al. (2013) implemented an ANFIS trained off-line beforehand in MATLAB environment. They used triangular membership function and different sizes fixed point number format. Abdelmoula et al. (2012) implemented two intelligent control approaches on FPGA. The first one was based on fuzzy logic and the other one was based on hybrid-type neuro-fuzzy ANFIS. They used look up table for member functions implementation and floating point number format for the two approaches. Chou et al. (2013) implemented RBF using triangular membership function and Taylor series for activation function approach in NN layer. They used signed data format in the implementation. Juang and Hsu (2005) designed a recurrent fuzzy controller (RFC) for temperature control system using a hybrid evolutionary method including Simplex method and PSO in a software environment, then they implemented this design adding/adapting it to online gradient descent learning for only consequent parameters on FPGA and look-up-table (LUT) method was executed for Gaussian MFs. This study can be considered as hardware implementation of semi-trained neuro-fuzzy network on FPGA. Hima et al. (2012) implemented ANFIS system and trained by error correction algorithm on FPGA. They used floating point number format and the second order approximation approach for Gaussian membership function. Campo et al. (2008) implemented piecewise multi linear ANFIS structure using triangular MFs together with an online hybrid learning algorithm (forward pass: the consequent parameters are updated by LSE, backward pass: the antecedent parameters are updated by gradient descent). Lin and Lee solved time sequence prediction and dynamic system identification problems in Lin and Lee (2009) using recurrent neural fuzzy network (RNFN) together with its "modified simultaneous perturbation method" learning on FPGA. They used fixed point number format for data representation and an approximation proposed in Blake and Maguire (1998) for implementing Gaussian MFs. In Lin and Tsai (2008), Lin and Tsai showed implementation of a wavelet neural network (WNN) and its learning using PSO on FPGA for prediction of a chaotic signal and dynamic system identification problem. They used fixed point number format for data representation and LUT approximation based on Taylor series for the network's nonlinear activation functions. For better comparison, the properties of the implementations available in the literature and the properties of the proposed one are listed in Table 1.

In this paper NFS and its iPSO (Çavuslu et al., 2012) learning module is implemented on FPGA using single precision (32 bit) floating point number format and it is tested on dynamic system identification and licence plate localization problems. Two different membership function implementation approaches are used. The first approach is taken from Jang (1993). This approach needs adder, multiplier and a divider arithmetic operations for Gaussian MF implementation on FPGA. Second approach, proposed in this paper, is one of the novel contributions of this study. This approach does not need any multiplier or memory compared to look-up table method. This study has been implemented on Xilinx Virtex 5 xc5vlx110-3ff1153 FPGA.

Rest of the paper is organized as follows: Section 2 introduces NFS architecture and its learning using iPSO. In Section 3, FPGA implementation details of NFS and its learning by means of iPSO are given. Performance of the work is tested based on three examples and results are given in Section 4. And lastly, Section 5 presents the conclusion.