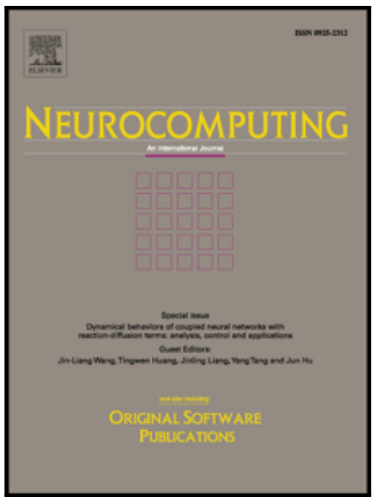

Reconfigurable FPGA implementation of neural networks

Zbigniew Hajduk

PII: S0925-2312(18)30539-3
DOI: 10.1016/j.neucom.2018.04.077
Reference: NEUCOM 19558

To appear in: *Neurocomputing*

Received date: 9 October 2017
Revised date: 16 March 2018
Accepted date: 30 April 2018

Please cite this article as: Zbigniew Hajduk , Reconfigurable FPGA implementation of neural networks, *Neurocomputing* (2018), doi: 10.1016/j.neucom.2018.04.077

Reconfigurable FPGA implementation of neural networks

Zbigniew Hajduk* zhajduk@kia.prz.edu.pl
Rzeszów University of Technology, ul. Powstańców Warszawy 12, 35-959 Rzeszów, Poland.
*Corresponding author.

Abstract

This brief paper presents two implementations of feed-forward artificial neural networks in FPGAs. The implementations differ in the FPGA resources requirement and calculations speed. Both implementations exercise floating point arithmetic, apply very high accuracy activation function realization, and enable easy alteration of the neural network's structure without the need of a re-implementation of the entire FPGA project.

Keywords

FPGA; Neural networks

1. Introduction

Most of the existing artificial neural networks (ANNs) applications, particularly for commercial environment, are developed as software. Yet, the parallelism offered by hardware may deliver some advantages such as higher speed, reduced cost, and higher tolerance of faults (graceful degradation) [1, 2]. Among various developed methods of ANNs implementations in field programmable gate arrays (FPGAs), e.g., [3 - 6], there is a breed of implementation which allows the structure of the ANN (i.e., the number of layers and/or neurons, etc.) to be altered without the need of re-synthesizing and re-implementation of the whole FPGA project. This feature increases the ANNs implementation flexibility to the similar level as offered by software, at the same time maintaining the advantages delivered by hardware. Unfortunately, existing solutions, e.g., [7 - 9], are based on fixed point arithmetic, have strongly limited calculations accuracy of the activation function, and require dedicated software tools for the formulation of a set of user instructions controlling the ANN calculations in the developed hardware. Some of them [9, 10] do not employ parallel architecture exploiting only a single neuron block for the calculations of the whole ANN. In