Design methodology of a fully parallelized Neural Network on a FPGA

SANTIAGO T. PÉREZ SUÁREZ[1], CARLOS OSORIO ROBAINA[1], JOSÉ L. VÁSQUEZ NÚÑEZ[2], JESUS B. ALONSO HERNÁNDEZ[1], CARLOS M. TRAVIESO GONZÁLEZ[1]

[1]Signal and Communication Department
University of Las Palmas de Gran Canaria
Campus University of Tafira,Las Palmas de Gran Canaria, Las Palmas
SPAIN
c.osorio85@gmail.com{santiago.perez, jesus.alonso, carlos.travieso}@ulpgc.es

[2] Department of Computer Science
University of Costa Rica
Sededel Atlántico, Turrialba, Cartago, Costa Rica
COSTA RICA
jose.vasquez@ucr.ac.cr

Abstract: In this work a methodology of a parallelized neural network has been designed. It explains a way to design a Neural Network using Mathworks and Xilinx Tools. Initially, the floating point algorithm was evaluated using MatlabNeural Network Toolbox. Afterwards, the fixed point algorithm was designed on a Field Programmable Gate Array (FPGA).The architecture was fully parallelized. The design tool used is System Generator of Xilinx, which works over Simulink. Finally the System Generator design is compiled for Xilinx Integrated System Environment (ISE).

Key-Words: Neural Network,FPGA, floating point, fixed point, Matlab, Simulink, System Generator, ISE

1 Introduction

Artificial Neural Networks (ANN) have been used as identifiers of patterns during last years, its advantages and benefits are well-known [1]. This paper exposes a methodology for designing a parallelized ANN, and a series of environmentswhere it has been used.

Firstly, the paper exposes the definition of a completely parallelized NN and his float point design using Matlab [2]. Afterward it shows the available technologies for fixed point design and why FPGA is the best choice to do this task [3]. Finally, the paper shows the complete design of this NN and some conclusions.

2 Fully Parallelized Neural Network

The Artificial Neural Networks are a paradigm of learning and automatic processing inspired in animals' nervous system.It´s a combination of different systems, called neurons (Fig. 1), networked between themtocreatean excitationin the exit. First the inputs are multiplied by the weights and the bias is added. The new value is the input of the activation or transfer function (Fig. 2), which usually is a sigmoid function.

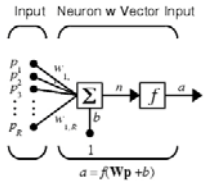
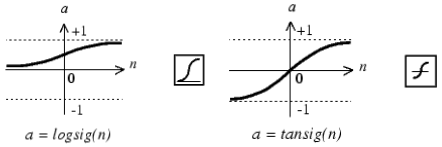


Fig.1. Artificial Neuron



$$a = \frac{1}{1+e^{-n}} \qquad a = \frac{2}{1+e^{-2n}} - 1$$

Fig.2. Sigmoid Functions, logsig and tansig

ISBN: 978-960-474-359-9

Neural Networks have two different architectures; this paper shows the parallel one.The main advantage of this architecture is that it can be used for real time applications. The NNs in general are composed ofthree layers:input, hidden and output layers (Fig. 3).

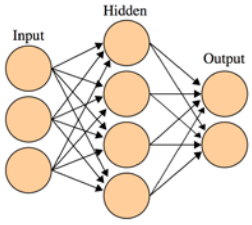


Fig.3. Parallel Neural Network

3 Floating Point Neural Network Design

The ANNs have been designed with MatlabNeural Network Toolbox[4].This toolbox helps to design a floating point ANN to check the functionalities of it and produces the “golden rule”.

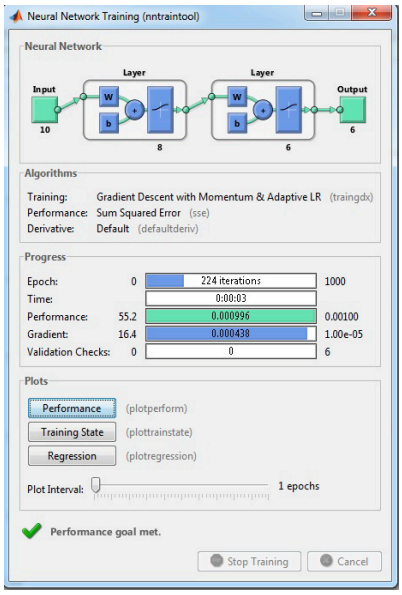


Fig.4. Matlab Neural Network Toolbox

When the design isimplemented in fixed point format it has to reach the functionality of the golden rule.First the ANN has to be trained; the designer has to configure number of layer, number of neuron and the transfer functions, and then introduces the database for training.

In Fig.4 can be seen an ANN with eight and six neuronsin the hidden and output layer. The transfer functions for these layers are the logsig type. This is a particular case, this configuration can be modified depending on the application. When thetraining has finished, the synapticweights and bias are obtained for the two layers.

4 Technologies Available

The algorithm should be developed in a digital circuit using fixed point arithmetic in two's complement.One alternative is to use an Application Specific Integrated Circuit (ASIC). The ASIC has low area occupation, low power consumption and high speed. But its disadvantages are: high price, difficult debugging and verification, big time to market, does not allow reprogramming and high non-recurring engineering costs. For these reason, ASIC is undesirable to develop prototypes where the number of units produced is small.

On the other hand, Digital Signal Processors (DSPs) can be used, which are cheaper than ASICs. The DSPs reach higher clock frequencies, but the data rate that can be processed is limited because of the parallelism of the data, the size and format of the data, and the pipelined are fixed. All this is imposed by its predetermined architecture.

Finally, the use of Field Programmable Gate Arrays (FPGA) has the advantages: low price, no non-recurring engineering cost, minimum development time, ease of debugging and verification, short time to market, high data parallelism, flexible data format and flexible pipelined. Although the clock frequency is not as high as in the DSPs, with the above characteristics is achieved to increase the data rate. Moreover FPGAs have higher power consumption, but they are appropriate for individual prototypes because FPGAs can be reprogrammed by the designer.

5 Design methodology

This methodology is based on Simulink ofMatlab [5]. The designer canimplement the ANN usingSystem Generator [6], this is a Xilinx application which runs over Simulink and allows to design in block diagrams, a fast and flexible way. It uses specific blocks from Xilinx FPGA so guaranties the complete compatibility with their products.

Once the architecture has been defined, the values of the NN and its parameters are loaded with an initialization Matlab file.These values were obtained in the training process (Fig.4), but they are not preparing for working in FPGA, so it will be necessary to turn them to Fixed Point. This transformation was realized with another Matlabfile.

ISBN: 978-960-474-359-9

At this point the model and the architecture of the system have been fixed; besides, the fully functionality has been checked with Simulink simulations.

Then the design is compiled with System Generator,for this compilation process the FPGA device must be chosen. Besides, for System Generator compilation a standard Hardware Description Language (HDL) must be chosen, the two possible languages are Verilog and VHDL (Very High Speed Integrated Circuit Hardware Description Language) [7, 8]. After the compilation, a project is generated for Xilinx Integrated System Environment (ISE), which includes the HDL files for the structural description of the system [9].

The ISE software is the Xilinx standard tool for FPGA design. The syntax of the HDL files can be checked, synthesis and behavioral simulation of the ANN can be executed. After that, the design implementation permits the timing simulation of the system.

6 Neural Network Design on FPGA

For the methodology description of the NN it has used a particular scene (see 7.1 Pejibaye Palm Identification).The Fig. 5 shows the complete ANN implemented with System Generator.This picture shows the design of Fig. 4: ten inputs, eight neurons in the hidden layer and six neurons in the output layer.

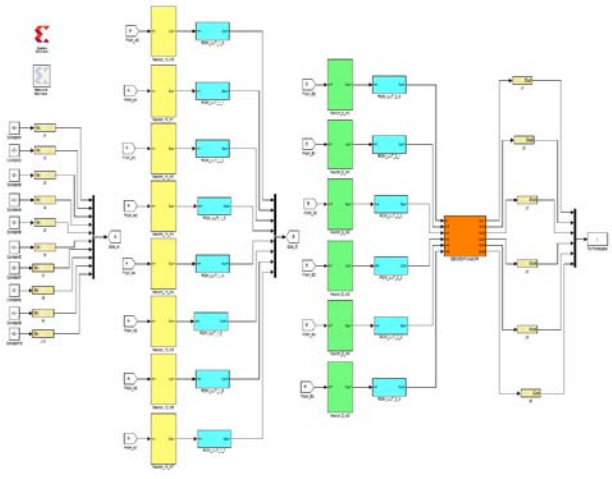


Fig.5. Artificial Neural Network designed with System Generator

In this particulars case of NN was chosen ten entrances for the different parameters of the information.These entrances were connected to each neuron of an eight neurons intermediate layer. The first stage of the implementation of each neuron is represented in Fig.6. This figure shows each entrance multiply for each weight and the addition of the bias (yellow block in Fig.5).

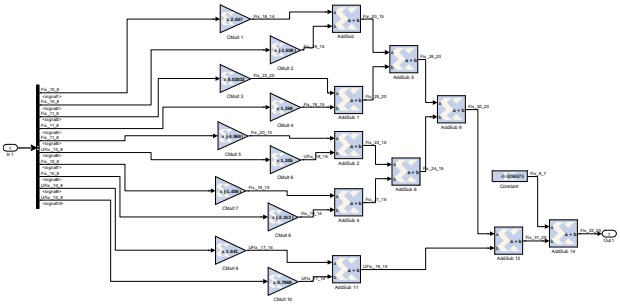


Fig.6. Artificial Neuron in System Generator

Figure 7 shows the second stage ofthe neuron, this is the activationfunction design, which uses a ROM memory for storing samples of the no linear function.

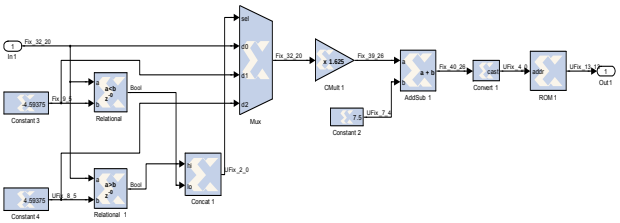


Fig.7. Activation Function

In the output layer was used the same activation function (Fig.7) but in this case it has only six neurons.At the end, there was a digital circuit, which compares all the exits looking for the neuron with the highest value.

7 Environments Tested

This design methodology has been used for many applications:
Pejibayepalmidentification,temperatureprediction, electrocardiogram signal anda binary equalizer.

7.1 PejibayePalm Identification

The Pejibaye palm (BactrisgasipaesKunth) is a species native from tropical America. The pejibaye adult stem is used as wood, young shoots and fruits are used as human and animal food. The database used in this work comes from the Germ Plasma Bank of the University of Costa Rica. The study was conducted with data from molecular markers RAPD (Random Amplification of Polymorphic DNA). Samples are previously classified into six primitive races.

ISBN: 978-960-474-359-9

The objective was to design a reconfigurable prototype with low response time, low size, low power consumption and good portability.The NN was designed with the number of entries equal to the number of parameters. The number of outputs is the number of classes. The performances of the NN, which is the Fig. 5,were reached with a single hidden layer. It was concluded that the optimal number of neurons in the intermediate layer was eight. The unipolar sigmoid activation function was specified for all neurons. The NN was testing and 100% accuracy was reached. In fact, this design is shown in Fig. 5.

7.2 Temperature Prediction

In this design was used a Time Delay Neural Network (TDNN) which is a neural network with input cascadeddelay elements (Fig.8).

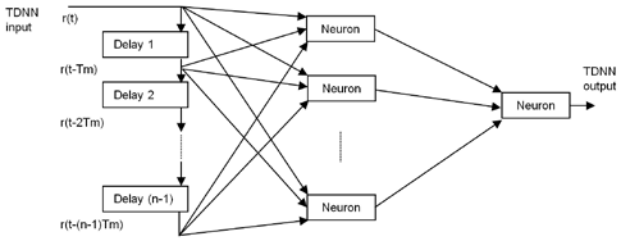


Fig.8. Time Delay Neural Network

The data used in this application came from a weather station located inCosta Rica. It includes information from 2005 to 2009 with a sampling frequency of every half an hour. For the construction of the ANN the cases were obtained using a sliding window on each data series. The data enclosed by the window represent a study case, on a recurring basis, the window will move one to one on the data series until to obtain all the patterns, see Fig. 6.

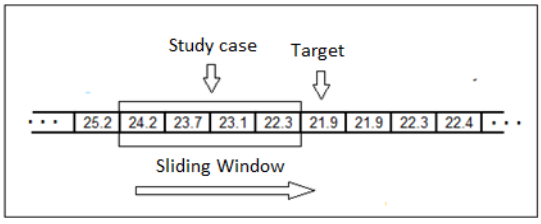


Fig.6. Getting a study case from a time series

The base of this model was an Artificial Neural Network Multilayer Perceptron Feed-Forward with Back-Propagation training algorithm, with only one hidden layer witheight neurons. The Artificial Neural Network had five inputs, and the output layer had only one element that indicates the estimated value of temperature. To train the ANN we used data from a sampling year, which corresponds to 20% of the database.

7.3Electrocardiogram Signal

In this study has been proposed to discriminate eight heartbeats. The main aim of this work was to implement this automatic detector of heart diseases over a Field Programmable Gate Array. For this purpose the signal was processed with the Discrete Wavelet Transform. These types of logical programmable devices are quite suitable to wavelet digital filters.

7.4 Binary Equalizer

One objectiveofthis implementation was to check if a TDNN canbe usedas a preamplifierorequalizer; increasing the outputSignal to Noise Relation (Fig. 8). Furthermore, it was proposed a design methodology over a FGPA for the TDNN.

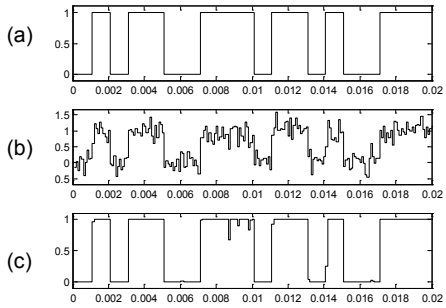


Fig.8. Original data signal (a), noisy sampled signal with +10 dB of SNR (b) and TDNN output (c)

8 Conclusions

A design methodology of a fully parallelized ANN is presented using Matlabover a FPGA. The design can be differentiated in three phases; Matlab supports the first two. In the first stage is used the Matlab Neural Network Toolbox for fixing the floating point architecture, parameters and the performance of the neural network, the information obtained can be called the "golden rule".

In the second stage Xilinx System Generator is used, which operates on Matlab Simulink. In this phase the system is designed in fixed point format according to the golden rule. In System Generator

ISBN: 978-960-474-359-9

the circuits are handled with low level of detail, for this reason the simulations are very fast and the functionality of the system can checked completely. During this stage a poor estimation of the area is calculated, and neither power consumption nor speed of the circuit, are evaluated. Moreover, the effect of the number of bits in different parts of the design can be tested. The fixed-point format has implications on the functionality of the system and the hardware resources occupied.

In the third step, the system description obtained with System Generator is used by Xilinx Integrated System Environment. This design tool uses a high level of detail of the circuits; this allows estimation of physical performances: hardware resources, power consumption and maximum clock frequency.

It should be noted that the description of the system obtained by System Generator is not portable to other manufacturers. The design could have been done for Altera, the second FPGA manufacturer in importance. Altera offers DSP Builder, which is a similar tool over Simulink. In the same way these designs are only valid for Altera FPGAs. As a future line, could be used Matlab HDL Coder, where files are portable to all manufacturers. The HDL Coder designs can be compared with the designs obtained with the FPGA manufacturers' tools. The results will depend on the compilers. Provide the portability using a hand coded hardware description language is not a good alternative. The design of complex systems directly in a hardware description language is long and tedious, and not flexible for changes.

It was assumed the same error in the binary representation of the input and coefficients in the two layers. In general it should be analyzed the effect of different errors for input and each layer coefficients. The conclusions should focus on functionality and physical performances of the system. This study should be automated with a Matlab program for executing the models designed with System Generator.

References:

[1] R. O. Duda, P. E. Hart, D. G. Stork, *Pattern Classification*, 2nd ed, Wiley-Interscience Danvers, USA, 2001.

[2] Matlab. http://www.mathworks.com/products/ (accessed on 20 October 2013).

[3] S. Hauck, A. DeHon, *Reconfigurable Computing*, 1st ed, Elsevier, Netherlands, 2008.

[4] Neural Network Toolbox. http://www.mathworks.com/products/neural-network (accessed on 20 October 2013).

[5] Simulink. http://www.mathworks.com/help/simulink/ (accessed on 20 October 2013).

[6] System Generator. http://www.xilinx.com/tools/sysgen.htm (accessed on 20 October 2013).

[7] S. Palnitkar, *Verilog HDL*, 2nd ed, Prentice Hall, Upper Saddle River, USA, 2003.

[8] V. A. Pedroni, *Circuit Design with VHDL*, 1st ed, The MIT Press, London, England, 2004.

[9] Integrated System Environment. http://www.xilinx.com/products/design-tools/ise-design-suite/index.htm (accessed on 20 October 2013).