FPGA Based Neural Wireless Sensor Network

Thamer M. Jamel  
Electrical Engineering Department  
University of Technology  
Baghdad, Iraq  
thamerjamel@yahoo.com

Ban M. Khammas  
Information Engineering College  
AI –Nahrain University  
Baghdad, Iraq  
banmoh_79@yahoo.com

Mohammed K. Zibar  
Medical Instruments Department  
Technical Medical Institution  
Baghdad, Iraq  
mkhammas@yahoo.com

Abstract— Wireless sensor networks (WSN) are an exciting emerging technology that scientists believe to become a part of everyday life in the next few years. The present work describes the implementation of the Back propagation algorithm for use in Wireless Sensor Network (WSN) application. The application had been performed over an FPGA (field-programmable gate arrays), and MATLAB software package. Moreover, this paper studies the architecture of a neural wireless sensor network designed to identify technical conditions (temperature, humidity, and light (total solar radiation (TSR) and the photo synthetically active radiation (PAR)) of the base station of wireless sensor networks. A comparison between FPGA and MATLAB software implementation is then made with different output technical conditions of WSN: the differences were not too much.

Keywords- Wireless sensor networks, artificial neural networks, FPGA.

1. Introduction

Wireless communication is one of the most vibrant research areas in the communication field today. While it has been a topic of study since the 60s, the past decade has seen a surge of research activities in the area. This is due to a confluence of several factors. First factor, is the explosive increase in demand for connectivity, driven so far mainly by cellular telephony but is expected to be soon eclipsed by wireless data applications. Second factor, is the dramatic progress in VLSI technology which has enabled small-area and low-power implementation of sophisticated signal processing algorithms and coding techniques [15].

With the requirements for advanced integration, intensive onboard processing, and low power consumption, FPGAs emerge as a technology of choice that strikes an optimal balance between processing power, energy requirements, and flexibility. Through the power of reconfigurable logic, wireless sensor network is containing reprogrammed, which can be upgraded; errors can be fixed, and limited-resource applications can be dynamically reprogrammed in the field [15].

The concept of ANNs is emerged from the principles of brain that are adapted to digital computers. The first works of ANNs developed the models of neurons using mathematics rule [10]. These works show that each neuron in ANNs takes some information as an input from another neuron or from an external input. This information is propagated as an output that is computed as the weighted sum of inputs and applied to the non-linear function [11].

The idea of using FPGA in WSN has been used by Jorge P. Angel de Castro, Eduardo de la Torre, and Teresa Riesgo [8]. They design a modular architecture for nodes in WSN. The architecture is made up of four layers; these layers are (communication, processing, power supply and sensing/actuating). A set of such layers used to create the node for a specific application. Changing of application implies changing the layers or redesigning one or some of them from scratch all the system, which drives to high cost and time inefficiency. Another trend had been used by Stephen J. Kieran Delaney, Brendan O’Flynn, John Barton, Kafil M. Razeeb, and Cian O’Mathuna [3]. Their work partitions the wireless sensor module into a series of layers with area (25 mm X 25 mm). The module has resulted in the specification of a series of layers, including a FPGA for signal processing. Freek Z. [17], shows that WSN and NNs can be combined into a working system; in this case, the system tries to detect the shadows of clouds as they drift over a field.

In this paper, a new trend is proposed that combine NN with WSN for identifying technical condition's parameters (temperature, humidity, and light (total solar radiation (TSR) and the photo synthetically active radiation (PAR))) of the base station and implement the system using FPGA.

The paper is organized as in the following. In section 2, a neural network is introduced in the sensor network context. Furthermore, in section 3 we give a system design procedure of FPGA based sub neural sensor network using back propagation algorithm and multi layer perceptions (MLP). After this, section 4 presents
the experimental results, in which, the performance of the NWSN in our hardware architecture is shown, and a pure software implementation is introduced and compared with its hardware counterpart. Finally, in section 5, some of the important conclusions are given.

2. Wireless Sensor Network and Neural Networks

WSNs are an emerging technology which holds the potential to revolutionize everyday life [4, 9, and 12]. Continuous advances in semiconductor technology have enabled the miniaturization of radios and mechanical structures and deployment of very cost-efficient wireless sensor nodes.

A sensor node basically consists of a microcontroller (processor and memory), sensors, analog-to-digital converter (ADC), transceiver (sender and receiver) and power supply. The abstract architecture of a sensor node and the dependencies of its units are depicted in Figure 1 [7].

The structure of neural networks and wireless sensor networks can both be described as a collection of interconnected simple components. Both kinds of networks are also similar in that they implement a function that maps input values to output values. These similarities between wireless sensor networks and neural networks led to the construction and investigation of a system that combines both technologies. Physical limitations, however, complicate straightforward combination of both technologies [17].

In general, Artificial Neural Networks (ANN) shows characteristics such as distributed representation and processing, massive parallelism, learning and generalization ability, adaptively, inherent contextual information processing, fault tolerance and low computation. Many of those characteristics are either inherent or desirable for WSNs as well. As shown in

Figure 2. The perception of a biological neuron was transferred via an artificial neuron to the sensor node. The functionality h(x) is basically a weighted sum over all inputs. The sensor converts the physical world to an electrical signal which is filtered or preprocessed corresponding to weighting/synapse. The subsequent processing within the processor corresponds to the chemical processing accomplished by the soma or applying the particular functionality h(x), respectively.

Finally, the sensor node sends out the modified sensor reading via the radio link. This strong analogy shows that the sensor node itself can be seen as a biological and artificial neuron, respectively.

Figure 2: The analogy of the biological neuron artificial neuron and the sensor node [7].

The most basic element of the neural network, the neuron, transforms its inputs in the feed forward stage of operation according to the following steps in (1) as shown in Figure 3.

- Multiply each input by its corresponding weight.
- Sum the multiplication results.
- Squash” the sum using a transfer function. [13]

This concept of neuron structure is used in this paper.

Figure 3: A neuron structure.
3. System Design

The system consists of several neural network modules operate in parallel [8, and 11]. Each module is a three-layer neural network and has its own training data set. A high speed back propagation learning algorithm has been used to train the networks. Before learning starts, tolerances are defined for the output units. During learning, the weights are updated only when the output errors exceed the tolerances. The learning data for which the output errors do not exceed the tolerances are eliminated from the training data sets [7]. Parameters that are selected and used for our design are:

- The learning algorithm is the back propagation algorithm.
- Multi Layer Perception (MLP) structure which consists of input, hidden and output nodes of the network.
- The activation function is log-sigmoid function.
- The learning rate is set to 0.1.

The base station system is design to received signals from multiple sensors (temperature sensors, humidity sensors, and light sensors), which measure the total solar radiation (TSR) and the photo synthetically active radiation (PAR).

The number of input neurons equals the number of bits that use from all sensors. To be more precise, this paper used the following number of bits related to the particular sensors (7 bits for temperature (the measure range 0-100°C), 7 bits for humidity (the measure range 0-100% ), and 7 bits for each light sensor, with the TSR and PAR measure range 0-100%. Furthermore, this paper used a software package MATLAB for offline learning method.

The MLP neural network with one hidden layer is used for every sub- neural networks system. Figure 4 shows the default three layers MLP sub- neural network which is designed by MATLAB in this research. Each layer includes some sub-blocks. The security system that design in this research working by controlling four neural network's modules in the base system of wireless sensor network as depicted in Figure 5. The system take extremely short training times and runs entirely autonomously. It does not need any outside control; it provides fast access to match the results, and is designed to work with the infinite stream of data. All these features make it an excellent choice for applications in wireless sensor networks. The initial parameters of the NWSN used in this research are given in Table 1. These parameters are used in MATLAB program to obtain the final correct weights of NN, which will be used in the hardware design of NWSN. These weights must be converted to binary form, and saved in a file for transmitting it later to the Xilinx program.

The initial parameters of the NWSN used in this research are given in Table 1. These parameters are used in MATLAB program to obtain the final correct weights of NN, which will be used in the hardware design of NWSN. These weights must be converted to binary form, and saved in a file for transmitting it later to the Xilinx program.
The sigmoid function is very popular for neural networks, because it performs similar to a step function, but it is everywhere differentiable. The standard form of the sigmoid is:

\[ f(\text{sum}) = \frac{1}{1 + e^{-\text{sum}}} \]  

(1)

Where \( f \) is the transfer function and \( \text{sum} \) is the summation of multiplying the input \( X_j \) and the weight \( W_j \) associated with each input as [16]:

\[ \text{sum} = \sum X_j W_j + b \]  

(2)

Moreover, \( b \) is the biased.

Table 1 Initial parameters of each sub-NN system

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of input neuron</td>
<td>?</td>
</tr>
<tr>
<td>Number of hidden layers</td>
<td>1</td>
</tr>
<tr>
<td>Number of total output neuron</td>
<td>1</td>
</tr>
<tr>
<td>Number of total neurons in hidden layers</td>
<td>3</td>
</tr>
<tr>
<td>Input range</td>
<td>[0, 1]</td>
</tr>
<tr>
<td>Output range</td>
<td>[0, 1]</td>
</tr>
<tr>
<td>Activation function of neurons in hidden layer</td>
<td>logsig</td>
</tr>
<tr>
<td>Activation function of neurons in output layer</td>
<td>logsig</td>
</tr>
<tr>
<td>Type of neural network</td>
<td>Feed forward back propagation network</td>
</tr>
<tr>
<td>Training Function</td>
<td>Trainslm (MATLAB built-in function)</td>
</tr>
<tr>
<td>Number of epochs used in training</td>
<td>10000</td>
</tr>
<tr>
<td>Back propagation weights/bias learning function</td>
<td>Levenberg (MATLAB built-in function)</td>
</tr>
<tr>
<td>Error tolerance</td>
<td>0.000000001</td>
</tr>
<tr>
<td>Performance function</td>
<td>Mean Square Error (MSE)</td>
</tr>
</tbody>
</table>

This equation was selected because it provides the necessary limiting of the outputs while having some properties, which are useful in the learning phase of the algorithm. Unfortunately, this equation contains the transcendental function (exponential term, \( \exp \)), which is somewhat difficult to calculate. Nordstrom and Svensson [16], list several functions, which may be used as an approximation to the function in equation (1). These functions have the same general characteristics. They are continuously increasing, approach 0 at \(-\infty\) and 1 at \(+\infty\), and have a continuous first derivative. This equation is implemented in hardware as a transfer function block which is shown in Figure 7. The activation block in Figure 7, which performs the multiplications of \( X_j \) and \( W_j \) and then performs the summation according to the equation (2), is always on the neuron-chip (or the processing element of the neuron computer). The data flow between these blocks is controlled by the Control Unit that is always on the chip. The data flow is such that the weights from the weights block and the inputs from outside or from the outputs are multiplied and the products are summed in the activation block, then the outputs are obtained in the neuron state block from the transferred sum of the products.

![Figure 7: The block diagram of the hardware neuron](image)

4. Simulation Results

The process of training is shown in the following Figures (8, 9, 10, and 11) in which training curves is approaching its goal through readjustment of weights and biases using MATLAB. Figure 12 shows the comparison results between MATLAB and Xilinx for the outputs of NWSN; the differences were not too much. Figure 13 shows the timing diagram of the base station of NSWN. As shown in the figure, the label h39.out7, the label h41.out7 and the label h42 Out7 represents the output result for temp., PAR, and TSR respectively.
Figure 8: Process of training for Humidity NN

Figure 9: Process of training for PAR NN

Figure 10: Process of training for temper. NN

Figure 11: Process of training for TSR NN

Figure 12: The comparison results between MATLAB and FPGA.
5. Conclusions

This paper presents hardware implementation of Wireless Sensor Network (WSN) using Field Programmable Gate Array (FPGA). The WSN is implemented using a Multi Layer Perceptions (MLP) neural network with the Back-Propagation (BP) algorithm. The network model has a three-layer structure which consists of an input layer, a hidden layer and an output layer. Then the MLP is implemented using FPGA. The simulation results show that the performance of the NWSN using software and hardware implementation were similar; the differences were not too much. The resultant neural networks are modular, compact, and efficient. The number of neurons, number of hidden layers and number of inputs are easily changed shown to be a very powerful embedded system design tool, with low cost, reliability, and multi-faceted applications. As FPGAs allow the hardware design via configuration software control, the improvement of circuitry design is just a matter of modifying, debugging and downloading the new configuration code in a short time.

References

[6] Liao Yihua , Neural Networks in Hardware: A Survey, Department of Computer Science, University of California, Davis One Shields Avenue, Davis, 1999.


