

Investigation on MLP Artificial Neural Network Using FPGA for Autonomous Cart Follower System

Liew Yeong Tat, S.S.N. Alhady, W.A.F.W. Othman
and Wan Rahiman

Abstract The future of the autonomous cart follower system will be equipped with lots of sensory data, due to the ever lower cost of sensory devices. This provides design challenge on handling large data and firmware complexity. This paper investigates an alternative approach of running the autonomous cart follower system on neural network model using Field Programmable Gates Array (FPGA). A microcontroller based autonomous cart follower system is modified to use the FPGA board and implemented via the System on Chip (SOC) approach. The neural network model is trained off line then implemented as software code in the SOC. By observation the firmware footprint of the neural network model remains small size regardless of the neural network size. The result shows that with 40 % more additional resource utilization, the overall system improvement of 27 times is achieved with the usage of hardware acceleration block in SOC compared to SOC without hardware acceleration.

Keywords Autonomous cart follower · MLP artificial neural network · FGPA · SOC

1 Introduction

Current developments in autonomous cart following system rely on the microcontroller to process the input data, and to determine the control output. Such implementation would require developer to hand coding the firmware code. The complexity of the firmware code usually proportional to the number of the sensory input and the control output, and most importantly the level of processing rules

L.Y. Tat · S.S.N. Alhady (✉) · W.A.F.W. Othman · W. Rahiman
School of Electrical and Electronic Engineering, Universiti Sains Malaysia,
Engineering Campus, 14300 Nibong Tebal, P. Pinang, Malaysia
e-mail: sahal@eng.usm.my

L.Y. Tat
e-mail: ytliw82@gmail.com

applied in the system. Besides, the increasing firmware footprint will become a challenge for fitting in the limited memory size in the microcontroller system. This work investigates the implementation of the artificial neural network in the autonomous cart following system, which was implemented via FPGA. Such approach offers an alternative cart following implementation, where the artificial neural network is used as the control system. In this scenario, developer emphasize on the training MLP network rather than optimizing the firmware code.

1.1 Autonomous Navigation Systems

Autonomous Land Vehicle in a Neural Network (ALVINN) demonstrates the utilization of the multilayer perceptron network (MLP) in learning to control the steering wheels of the vehicle [1]. To train the system, a human driver is required to drive the vehicle while the cameras in front of the vehicle capture image and feed the input nodes of the network. The output nodes of the network are produced to determine the angle of the steering direction.

1.2 Performance Over Software Implementation

A neural network based system has been proposed for wire length placement estimation and record shows that, due to the advantage of parallel execution of FPGA implementation, the execution period of the hardware implementation is unchanged regardless of the size of the circuit [2]. A full pipeline MLP neural network architecture on FPGA chip has been studied and could achieve performance improvement of up to 36 times better as compared with the software implementation on the general purpose processor [3]. The system overall frequency has not been affected by the number of the inputs into the neural network, as the FPGA logic is able to support the parallel fan out of the nodes to the downstream nodes. Study has been made on neural network based intruder detection system on FPGA, and able to record the overall speed improvement of up to 69 times quicker, over the traditional software implementation over microcontroller system [4].

2 Methodology

An existing autonomous cart follower system is used as the base framework [5]. A FPGA board is used to replace the microcontroller board and enhancement is made on the overall sensory inputs. The DE-0 Nano board has the Altera Cyclone IV EP4CE22F17C6 N FPGA chip, which equips 22320 logic elements (LE) for configuration [6]. This could provide a flexible and finer system on

implementation of the neural network model for the autonomous cart following system. In this work, the autonomous cart control system is proposed to run on System on Chip (SOC) design which is based on the Nios II processor. The SOC system design is created via the Qsys tools from Altera.

2.1 Algorithm of Control System

Based on the motor control data and the range of sensory data, the autonomous cart follower system is designed to track and follow a predefined pattern which is pasted on a subject. If the subject moves away from the cart for more than the threshold value (>30 cm), then the cart will move forward to stay close to the subject. If the subject moves closer to the cart (<10 cm), then the cart will move away from the subject to avoid the possible collision. The speed of the cart will increase or decrease depending on the distant of the cart and the subject, and urgency of the situation.

2.2 Neural Network Model for Autonomous Cart Follower System

MLP network model with single layer of hidden neurons is proposed to implement the control system of the autonomous cart follower system. Matlab Neural Network Toolbox is used to train the neural network model of the system [7]. In this work Levenberg-Marquardt backpropagation (LM), Bayesian regularization backpropagation (BR), and Scaled conjugate gradient backpropagation (SCG) training functions have been selected. The selected training functions are then trained with training vectors, and the performance of training functions are evaluated in the aspect of Mean Squared Error (MSE), epoch counts, and correlation value (R). The R value represents how accurate the neural model is comparatively to the training vector, in the range of $[0, 1]$ of which 1 indicates maximum correlation.

2.3 Realization of Neural Network on FPGA

Research has been carried out to evaluate the FPGA implementation of neural network with both on-chip SOC and FSM solution, and suggested that the hardware acceleration will significantly improve the Nios II CPU execution performance in running neural network model [8]. Existing Nios II custom instruction blocks could be used to achieve this purpose [9]. The network size, weight, and bias information

will be extracted from the chosen trained neural network. Nios II Software Build Tools for Eclipse is then used to build the control system software for the autonomous cart follower system.

2.4 Nios II Custom Instruction

Nios II soft-core processor could be enhanced with single precision hardware floating-point support with the usage of custom instruction block, in which will increase the multiplication and accumulation performance as the result could be computed within few cycles rather than go through the lengthy software emulation execution. Since the hardware floating-point unit only support single precision, thus the floating-point operation must strictly limit to the float type variables. To keep the variables in register without being swapped out, the “register” keyword is used to reduce such swapping frequency of critical floating-point value during software execution.

2.5 Activation Function Approximation

There are several ways to implement the sigmoidal activation function in FPGA, by using linear approximation method or lookup table for instance. The existing method of using lookup table is either utilizing pipelined LE block, or initialize the data in a dedicated memory block for read back. Since the FPGA board has an on-board 32 MB SDRAM memory; this memory space is used to construct the lookup table for activation function. The lookup table data can be generated during program start time by using the software emulation code execution to get the table data.

3 Results and Discussions

3.1 System Resource Utilization

The addition of the hardware floating-point acceleration block increase the LE resource utilization about 2496 LE, about 40 % from the basic default system as shown in Table 1. However, such increment is acceptable as the utilization still within 40 % of the total LE. The increment for hardware acceleration is preferable as it does not affect the system frequency, while benefited from the execution cycle improvement.

Table 1 Hardware utilization report

Report	SOC system without hardware floating-point acceleration	SOC system with hardware floating-point acceleration
Frequency	80 MHz	80 MHz
LE usage	6395/22320 (29 %)	8891/22320 (40 %)
Multiplier	4/132 (3 %)	13/132 (10 %)
On-chip memory bit	26672/608256 (4 %)	42032/608256 (7 %)

3.2 Training Function Evaluation for Motor Control System Neural Network Model

The LM backpropagation and BR backpropagation both show a smooth MSE performance after network size of 8 hidden neurons as shown in Fig. 1. However, the result for SCG backpropagation MSE is significantly higher than the LM and BR backpropagation. The performance of SCG is trending toward higher MSE value over the increment of the network size.

The BR backpropagation is apparently utilizing more epochs with the increment of the network size after 8 hidden neurons as shown in Fig. 2. This can be a problem when a bigger size network is used. The LM and SCG backpropagation are able to achieve target performance within the average count of less than 300 epochs.

Both LM and BR backpropagation show a good correlation with the training vector for network size of 5 or more hidden neurons as shown in Fig. 3. While the SCG backpropagation is slightly lower (with three hidden networks) and fracture along the network size change. From Fig. 3, it can be seen that the LM backpropagation is preferable over BR and SCG backpropagation techniques. The LM backpropagation demonstrates a good balance of MSE performance, quicker convergent time with less epoch counts, and still preserve a good correlation with training vector.

Fig. 1 Comparison of MSE over network size for motor control system

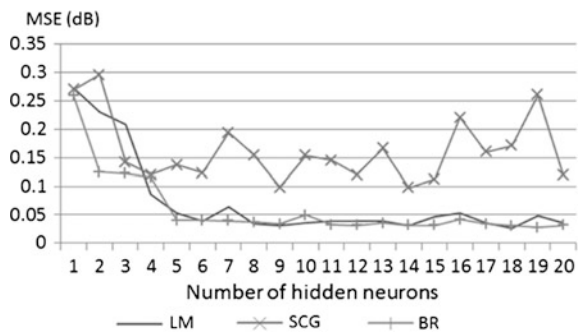


Fig. 2 Comparison of epoch counts over network size for motor control system

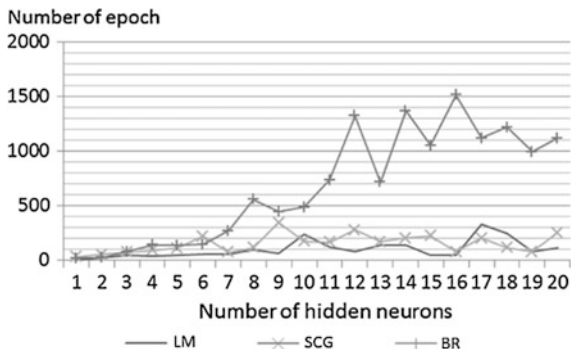
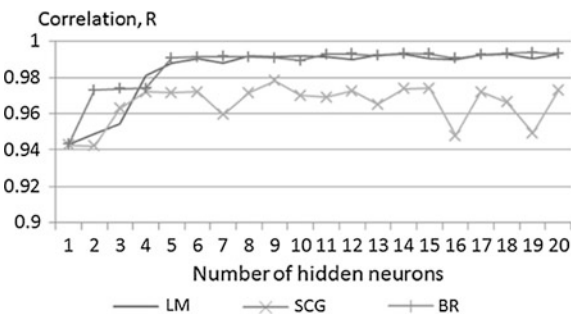


Fig. 3 Comparison of R value over network size for motor control system



The chosen neural network model is one with the size of 18 hidden neurons, which achieve lowest MSE in the group.

3.3 Software Performance

The performance counter report shows that the SOC without hardware acceleration requires 509,924 clock cycles to complete the control system execution, while with the present of the hardware acceleration, it is able to reduce the execution cycle to 18,459 clock cycles. This shows that the hardware acceleration of the custom instruction is improving the system performance by the magnitude of 27 times.

4 Conclusion

The result shows that with appropriate hardware acceleration block, the overall system performance of the autonomous cart follower system in neural network could be improved by 27 times, with cost of additional 40 % area utilization. It

presents a well-balanced approach in implementing autonomous cart follower control system with multiple sensor inputs. The choice of SOC based approach offer flexibility in implementing the neural network model with high level software language. By observation the software footprint of the neural network model remains small size and does not increase drastically on the scale of the system complexity.

Acknowledgments The authors would like to thank Universiti Sains Malaysia (USM) for the support of the research under grant no 304/PELECT/60313019.

References

1. Pomerleau DA (2012) Neural network perception for mobile robot guidance
2. Liu Q, Ma J, Zhang Q (2012) Neural network based pre-placement wirelength estimation in Field-Programmable Technology (FPT). In: 2012 International conference on, 2012. pp 16–22
3. Ferreira APDA, Barros ENDS (2010) A high performance full pipelined architecture of MLP. In: 2010 17th IEEE International Conference on, neural networks in FPGA, in electronics, circuits, and systems (ICECS) 2010. pp 742–745
4. Hassan A, Elnakib A, Abo-Elsoud M (2008) FPGA-based neuro-architecture intrusion detection system. In: International Conference on, Computer Engineering and Systems, 2008. ICCES 2008, pp 268–273
5. Sani NA, Nazli Alhady SS, Othman W, Kaharuddin S (2012) Cordless Cart Follower for Wheelchair User. In: Trends in intelligent robotics, automation, and manufacturing. pp 252–262
6. Terasic Technologies (2012) DE0-Nano User Manual
7. Demuth H, Beale M (2015) Neural network toolbox user's guide
8. Lozito G-M, Laudani A, Riganti Fulginei F, Salvini A (2014) FPGA implementations of feed forward neural network by using floating point hardware accelerators. *Adv Electr Electron Eng* 12:30–39
9. Suyyagh A, Abandah G (2013) FPGA parallel recognition engine for handwritten Arabic words. *J Signal Process Syst* 78:163–170