International Journal of Engineering and Advanced Technology (IJEAT)
ISSN: 2249-8958 (Online), Volume-9 Issue-1, October, 2019

FPGA Implementation of Weighted Online Sequential Extreme Learning Machine for Data Classification

Susanta Kumar Rout, Bhanja Kishor Swain, Pradyut Kumar Biswal

Abstract*: To design an efficient embedded module field-programmable gate array (FPGA) plays significant role. FPGA, a high speed reconfigurable hardware platform has been used in various field of research to produce the throughput efficiently. A now-a-days artificial neural network (ANN) is the most prevalent classifier for many analytical applications. In this paper, weighted online sequential extreme learning machine (WOS-ELM) classifier is presented and implemented in hardware environment to classify the different real-world bench-mark datasets. The faster learning speed, remarkable classification accuracy, lesser hardware resources, and short-event detection time, aid the hardware implementation of WOS-ELM classifier to design an embedded module. Finally, the developed hardware architecture of the WOS-ELM classifier is implemented on a high speed reconfigurable Xilinx Virtex (ML506) FPGA board to demonstrate the feasibility, effectiveness, and robustness of WOS-ELM classifier to classify the data in real-time environment.

Keywords:* Artificial neural network (ANN), Field-programmable gate array (FPGA), Hardware architecture, Hardware implementation, Weighted online sequential extreme learning machine (WOS-ELM).

I. INTRODUCTION

The hardware implementation of artificial neural networks (ANNs) in a high speed reconfigurable digital platform designed in parallel manner provide efficient throughput with flexibility and high processing speed. Many researchers have been validating the simulation results in terms of performance, hardware resources, and power consumption [1]. Field-programmable gate array (FPGA) has been used in different areas of research, such as signal processing [2], image processing [3], and sensor networks [4]. The major advantages of FPGA over application specific integrated circuits (ASIC) are reusability, flexibility, and high clock speed. Recently, machine learning is an important topic in several domain of research, which can accept meaningful features from data extracted using a conventional signal processing technique for classification, and detection purpose. ANNs generally consists of nodes similar to biological neurons to process the information. ANNs has been used frequently in signal and image processing due to extraction of sequential information from complex data. Now-a-days single layer feed-forward neural networks (SLFN) are used efficiently due to its capability to transfer the information sequentially. Huang *et al.* [5] , [6] are presented a single layer feed forward neural networks (SLFN) with maximum N number of hidden neurons for classification and regression of data. In [7-9] authors proposed a new single layer feed-forward neural network named as extreme learning machine (ELM) to solve both regression and classification problems. ELM produces promising classification accuracy, high learning speed, automatic tuning of parameters, and use of wide range of activation functions are the key advantages as compared to support vector machine (SVM) [10], least-square support vector machine [11], and other state-of-the-art machine learning algorithms. In case of ELM the input weight and biases are chosen randomly, that may reduce the effectiveness of ELM. The above disadvantage of ELM is avoided by using effective extreme learning machine (E-ELM) presented in [12]. The training phase of ELM requires more time, for large data size. So the variants of ELM are proposed by different researchers to increase the effectiveness of the ELM. Huang *et al.* [13-15] proposed incremental extreme learning machine (I-ELM) to replace the hidden neurons during the training phase. It chooses proper nodes from a bunch of nodes and eliminates redundant nodes. Bidirectional extreme learning machine (B-ELM) is presented in [16] to reduce the network size of the classical ELM. Rong *et al.* [17], presented the fast pruned extreme learning machine (P-ELM) and Zhang *et al.* proposed optimally pruned extreme learning machine (OP-ELM) in [18], to eradicate the hidden nodes during training phase. In [19], authors presented adaptive extreme learning machine (A-ELM), in which number of hidden nodes may increase, decrease or remain the same during training phase. A fast and accurate online sequential learning algorithm presented by Liang *et al.*

Revised Manuscript Received on October 30, 2019.
* Correspondence Author
Susanta Kumar Rout*, Department of Electrical and Electronics Engineering, Siksha 'O' Anusandhan Deemed to be University, Bhubaneswar , India. Email: susantarout.iter@gmail.com
Bhanja Kishor Swain*, Department of Electrical Engineering, Siksha 'O' Anusandhan Deemed to be University, Bhubaneswar , India. Email: bkswain123@gmail.com
Pradyut Kumar Biswal, Department of Electronics and Telecommunication Engineering, International Institute of Information Technology, Bhubaneswar, India. E-mail: pradyut@iiit-bh.ac.in

© The Authors. Published by Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP). This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/)

[20], and Online sequential extreme learning machine (OS-ELM) is proposed by Ye *et al.* [21], are applied on real-world dataset in which data may arrive one by one or chunk by chunk with a fixed or variable step size. OS-ELM produces better classification and regression results.

Moreover, several efforts have been applied to increase the accuracy and effectiveness of the different variants of ELM classifier in software environment, but as far the authors knowledge, very few research articles have presented the hardware implementation of machine learning technique to solve the real-world problem. Decherchi *et al.* [22], proposed the digital implementation of extreme learning machine in a field-programmable gate array for classification purpose. In [23], Villora *et al.* presented the hardware implementation of real-time extreme learning machine in a field-programmable gate array for data classification.

In this paper, authors presented the hardware implementation of weighted online-sequential extreme learning machine (WOS-ELM) classifier for testing phase using cosine activation function to increase the overall testing classification accuracy with minimum number of hidden nodes. Finally, the proposed digital architecture of WOS-ELM classifier is implemented in a high speed reconfigurable FPGA environment to validate the effectiveness of the classifier. The results of the digital architecture in terms of clock speed, hardware resources, and power consumption is computed. The hardware implementation block diagram of the WOS-ELM classifier for data classification is shown in Fig. 1.

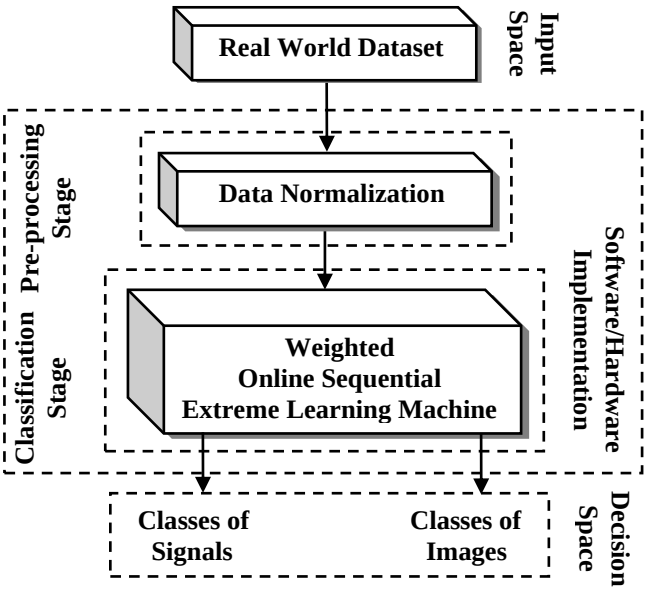

Fig. 1. The hardware implementation block diagram of the weighted online-sequential extreme learning machine classifier for data classification.

The remaining section of the paper is formulated as follows. Section II describes the materials and methods. Section III presents the results and discussion. Finally section IV draws the conclusion of the novel method.

II. MATERIALS AND METHODS

A. Dataset Details

In this paper, authors use publicly available online UCI machine learning repository [24], to test the performance and effectiveness of the weighted online-sequential extreme learning machine (WOS-ELM) classifier. To execute the experiment, authors considered three datasets named as diabetes datasets, image segment datasets, and land satellite image datasets. The diabetes dataset contains 8 features, 2 classes, 576 training samples, and 192 testing samples. The image segment dataset has 19 features, 7 classes, 1500 training samples, and 810 testing samples. The land satellite image dataset has 36 features, 6 classes, 4435 training samples, and 2000 testing samples.

B. Weighted Online-Sequential Extreme Learning Machine (WOS-ELM)

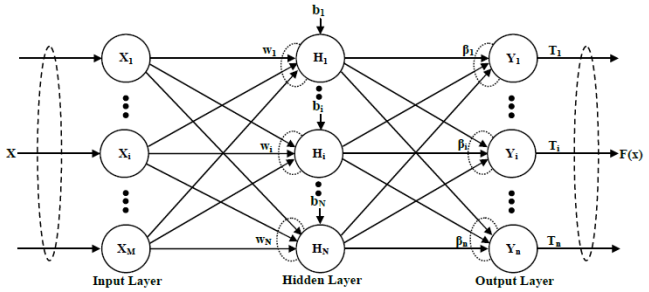

Fig. 2. Structure of Extreme learning Machine.

The disadvantages of the extreme learning machine (ELM) classifier is its incapability of computing the optimum value of number of hidden nodes, weight, and bias to acquire the maximum classification accuracy. In case of multiclass feature vectors, the WOS-ELM is presented to allot separate weights for every class and depending on the size of chunk, number of the hidden layers are decided to obtain a training root-mean-square error (RMSE) less than 0.0075. WOS-ELM gives better result than unweighted OS-ELM [25]. The structure of ELM is presented in Fig. 2.

Consider M number of input samples of the i^{th} feature vector for input class $[X_{i1}, X_{i2}, \cdots, X_{iM}]^T \in R^m$. The ELM has N number of hidden nodes with input and hidden layer weight w_i is $[w_{i1}, w_{i2}, \cdots, w_{iN}]^T$, and the hidden layer bias b_i is $[b_{i1}, b_{i2}, \cdots, b_{iN}]^T$. Number of output classes n, has a target of $T_i = [T_{i1}, T_{i2}, \cdots, T_{in}]^T \in R^n$ and the output weight vector $\beta_i = [\beta_{i1}, \beta_{i2}, \cdots, \beta_{in}]^T$ is calculated from output and hidden nodes. The analytical miniature of standard ELM taking non-regular, infinitely differentiable activation function $K(y)$ with the network output $Y_i = [Y_{i1}, Y_{i2}, \cdots, Y_{in}]^T$ is

$$\sum_{i=1}^{N} \beta_i K_i(X_j) = \sum_{i=1}^{N} \beta_i K_i(w_i \cdot X_j + b_i) = Y_j, \quad j = 1,2,\cdots,M \quad (1)$$

ELM can take the appropriate value of w_i, b_i and β_i to achieve zero mean error, i.e. $\sum_{j=1}^{M} \left\| Y_j - T_j \right\| = 0$.

Such that

$$\sum_{i=1}^{N} \beta_i K_i(w_i \cdot X_j + b_i) = T_j, \quad j = 1,2,\cdots,M \tag{2}$$

The hidden layer output matrix H of the ELM [5], [6] are presented in terms of (2) as,

$$H \cdot \beta = T \tag{3}$$

where

$$H = \begin{bmatrix} K(w_1 \cdot X_1 + b_1) & \cdots & K(w_N \cdot X_1 + b_N) \\ \vdots & \ddots & \vdots \\ K(w_1 \cdot X_M + b_1) & \cdots & K(w_N \cdot X_M + b_N) \end{bmatrix}_{M \times N};$$

$$\beta = \begin{bmatrix} \beta_1^T \\ \vdots \\ \beta_N^T \end{bmatrix}_{N \times n} \quad and \quad T = \begin{bmatrix} T_1^T \\ \vdots \\ T_M^T \end{bmatrix}_{M \times n}$$

The output weight matrix is calculated as

$$\hat{\beta} = H^{\dagger} \cdot T \tag{4}$$

where $H^{\dagger}$ is the *Moore-Penrose generalized inverse* of hidden layer output matrix H [26], [27].

To avert singularities using the original projection method [26], the $\hat{\beta}$ is defined as,

$$\hat{\beta} = \begin{cases} H^T\left(HH^T\right)^{-1}T & if \quad HH^T \neq 0 \\ \left(H^T H\right)^{-1}H^T T & if \quad H^T H \neq 0 \end{cases} \tag{5}$$

$$F(X) = \begin{cases} h(X) \cdot H^T\left(HH^T\right)^{-1}T & if \quad M < N \\ h(X) \cdot \left(H^T H\right)^{-1}H^T T & if \quad M > N \end{cases} \tag{6}$$

The initial training pattern of the input feature vector is represented as, $M_0 = \{(X_i, t_i)\}_{i=1}^{M_0} \in R^m \times R^n$ *and* $M_0 > N$. Calculate the hidden layer output matrix $[H_0]_{M_0 \times N}$ using (3) and the output weight matrix $\beta^0 = \left(H_0^T H_0\right)^{-1}H_0^T T = A_0^{-1}H_0^T T$. Where $A_0 = H_0^T H_0$.

Take the latest chunk of feature vector M_1 with M_0, the recent output weight matrix

$$\beta^1 = A_1^{-1}\begin{bmatrix} H_0 \\ H_1 \end{bmatrix}^T \begin{bmatrix} T_0 \\ T_1 \end{bmatrix} \tag{7}$$

where

$$A_1 = \begin{bmatrix} H_0 \\ H_1 \end{bmatrix}^T \begin{bmatrix} H_0 \\ H_1 \end{bmatrix} = \begin{bmatrix} H_0^T & H_1^T \end{bmatrix} \begin{bmatrix} H_0 \\ H_1 \end{bmatrix} = A_0 + H_1^T H_1 \tag{8}$$

and

$$\begin{bmatrix} H_0 \\ H_1 \end{bmatrix}^T \begin{bmatrix} T_0 \\ T_1 \end{bmatrix} = H_0^T T_0 + H_1^T T_1 = D_0\beta^0 + H_1^T T_1 = \left(D_1 - H_1^T T\right)\beta^0 + H_1^T T_1 = D_1\beta^0 + H_1^T H_1\beta^0 + H_1^T H_1 \tag{9}$$

Combining (8) and (9) $\beta^1 = \beta^0 + A_1^{-1}H_1^T\left(T_1 - H_1\beta^0\right)$

The input feature pattern of $(m+1)^{th}$ chunk along with all the remark of earlier chunks

$$M_{m+1} = \{(X_i, t_i)\}_{i=\left(\sum_{j=0}^{m} M_j\right)+1}^{\sum_{j=0}^{m+1} M_j}.$$

The partial hidden layer output matrix is

$$H_{m+1} = \begin{bmatrix} K(w_1 \cdot X_{\sum_{j=0}^{m} M_j} + b_1) & \cdots & K(w_N \cdot X_{\sum_{j=0}^{m} M_j} + b_N) \\ \vdots & \ddots & \vdots \\ K(w_1 \cdot X_{\sum_{j=0}^{m} M_j} + b_1) & \cdots & K(w_N \cdot X_{\sum_{j=0}^{m} M_j} + b_N) \end{bmatrix}_{M+1 \times N} \tag{10}$$

$$and \; A_{m+1}^{-1} = \left(A_m + H_{m+1}^T H_{m+1}\right)^{-1} = A_m^{-1} - A_m^{-1}H_{m+1}^T\left(I + H_{m+1}A_m^{-1}H_{m+1}^T\right)^{-1}H_{m+1}A_m^{-1} \tag{11}$$

Let $V_{m+1} = A_{m+1}^{-1}$, then A_{m+1}^{-1} can rewritten as

$$V_{m+1} = V_m - V_m H_{m+1}^T\left(I + H_{m+1}V_m H_{m+1}^T\right)^{-1}H_{m+1}V_m = V_m - \frac{V_m H_{m+1}H_{m+1}^T V_m}{I + H_{m+1}V_m H_{m+1}^T} \tag{12}$$

$$\beta^{m+1} = \beta^m + A_{m+1}^{-1}H_{m+1}^T(T_{m+1} - H_{m+1}\beta^m) \tag{13}$$

The diagonal matrix is a weight matrix i.e., $\omega = diagonal\{\omega_{ii}\}, i = 1,\cdots,M_S$, where M_S is the number of classes and ω is computed the from each class feature chunk. To increase the marginal distance between the classes and to decrease the weighted cumulative error with respect to each class, authors use (0.712:1) golden ratio.

$$Weighted\ scheme\ \omega = \begin{cases} \omega_{ii} = 0.712/M_n & ; \quad if\ \omega_{ii} > AVG(\omega_{ii}) \\ \omega_{ii} = 1/M_n & ; \quad if\ \omega_{ii} \leq AVG(\omega_{ii}) \end{cases} \tag{14}$$

where M_n is the number samples of majority classes.

Authors calculated the regular pattern ω from every class to increase the overall testing classification accuracy of the OS-ELM classifier. The OS-ELM algorithm is upgraded by considering the weight factor ω_{m+1}, the V_{m+1} and β^{m+1} as,

$$\hat{V}_{m+1} = \hat{A}_{m+1}^{-1} = \hat{V}_m - \frac{\hat{V}_m H_{m+1}^T \omega_{m+1} H_{m+1}\hat{V}_m}{I + \hat{V}_m H_{m+1}^T \omega_{m+1} H_{m+1}} \tag{15}$$

$$\hat{\beta}^{m+1} = \hat{\beta}^m + \hat{V}_{m+1}H_{m+1}^T(T_{m+1} - H_{m+1}\hat{\beta}^m) \tag{16}$$

III. RESULT AND DISCUSSION

In this work, to check the effectiveness and performance of the WOS-ELM classifier, three publicly available online datasets are used. To evaluate the overall testing classification accuracy of the WOS-ELM classifier, the feature vector of each dataset is fed to the classifier individually and simulated in MATLAB environment.

The overall testing classification accuracy is 82.68%, 97.46%, and 94.72% for diabetes dataset, image segment dataset, and land satellite image dataset respectively using cosine activation function is presented in Table I.

Finally, the proposed hardware architecture of WOS-ELM classifier is implemented in a high-speed reconfigurable FPGA platform to validate the effectiveness of the classifier.

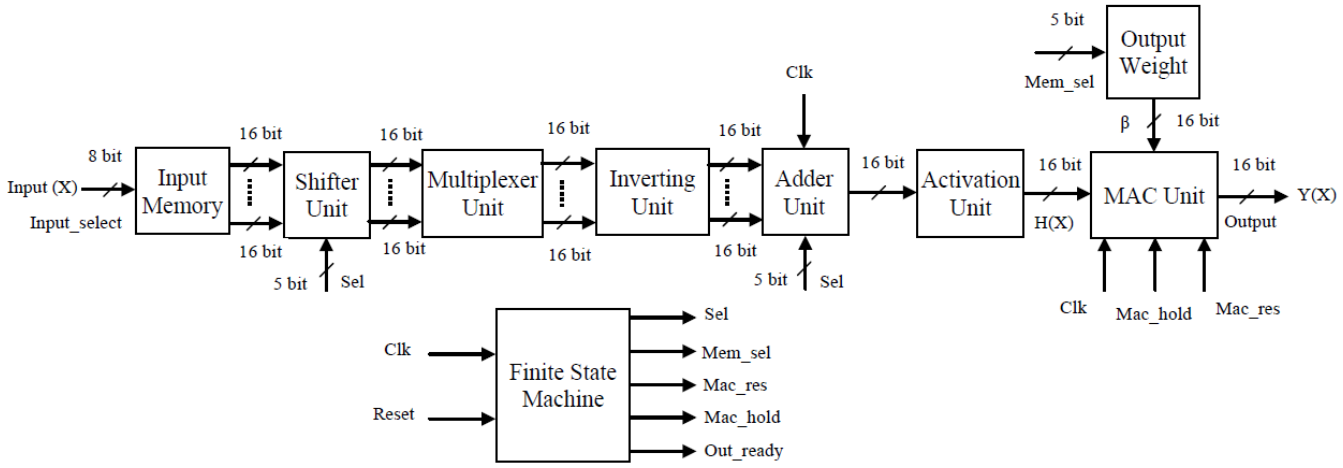

Fig. 3. Hardware implementation architecture of the WOS-ELM classifier.

Table-I: Performance Comparison of WOS-ELM with ELM Classifier for Real-World Dataset.

Authors	Dataset	Activation Function	Testing Accuracy (%)
Huang *et al.* [7]	Diabetes	Sigmoid	76.54
Huang *et al.* [8]	Diabetes	Sigmoid	77.57
	Segment		95.01
	Satimage		89.04
Huang *et al.* [9]	Diabetes	Sigmoid	77.95
		Gaussian	77.52
		Multiquad	78.09
	Segment	Sigmoid	96.07
		Gaussian	96.53
		Multiquad	95.54
	Satimage	Sigmoid	89.8
		Gaussian	92.35
		Multiquad	89.06
Proposed	**Diabetes**	**Cosine**	**82.68**
	Segment		**97.46**
	Satimage		**94.72**

FPGA environment to classify the real-world datasets. The proposed hardware architecture of WOS-ELM classifier is shown in Fig. 3. The overall hardware architecture of WOS-ELM classifier contains the following stages, i.e., input layer, hidden layer, output layer and all the stages are processed together using parallel and pipelining technique to produce the required throughput. To execute the experiment, a signed 2's complement fixed-point representation is used to represent a numerical quantity with 16 bits, where 3 bits are required to represent an integer value, 12 bits are required to represent a fractional value, and a sign bit.

The input layer of the hardware architecture is consisting of random access memory (RAM) to store the input testing data. RAM pops the testing data to the output according to the value of input-select line. To store the input testing data 8 RAMs, 19 RAMs, and 36 RAMs are required for diabetes dataset, image segment dataset, and land satellite image dataset respectively. The diabetes datasets with 192 testing data uses 8 bit input-select line as 2^8=256. Each RAM stores 192 data and fetches all the data to the output one-by-one according to the input-select line value. Similarly the image segment testing dataset and land satellite image dataset are processed according to the value of input-select line.

The hidden layer includes of shifter unit, multiplexer unit, inverting unit, adder unit, and activation unit. In this paper, one hidden node is represented by one barrel shifter to store the random weight (w) as positive quantities, which is the power of two like $0, \pm 1/2, \pm 1/4$ etc. The input layer produces 8 outputs from 8 RAMs and is fed to the N number of barrel shifters. The second stage of hidden layer is a multiplexer unit and each multiplexer has 18 inputs, as 18 hidden nodes are used for WOS-ELM classifier and one output according to the value of 5 bit selection line. The term $w_i \cdot X_i$ is computed using right barrel shifter in the place of multiplier to reduce the computational complexity, and hardware resources. The computed term is fed to the inverting unit with proper sign of hidden layer weight (w). The output of inverting unit is fed to the adder unit and extracts $\sum_{i=1}^{N} w_i \cdot X_i$ as output without considering the hidden layer bias (b) as input to avoid the hardware complexity. The output of adder unit is fed into the cosine activation function to set a threshold value which helps to classify the testing data. The output of the activation function is given by the equation,

$$H_i = K(\sum_{i=1}^{N} w_i \cdot X_i) = K(y) \quad (17)$$

where $y = \sum_{i=1}^{N} w_i \cdot X_i$

The cosine activation function is given by the expression

$$K(y) = cos(y) \quad (18)$$

The representation of cosine activation function using Taylor series expansion is given by,

$$K(y) = 1 - y^2/2! + y^4/4! - y^6/6! + \cdots\cdots \quad (19)$$

To reduce the computational complexity, authors reduce the 6^{th} order terms,

$$K(y) = 1 - y^2/2! + y^4/4! \quad (20)$$

It is very arduous to implement the (19) in FPGA. So the (19) is rewritten using right shifter as,

$$K(y) = 1 - y^2/2 + y^4/4 - y^4/16 - y^4/64 - y^4/256 - y^4/1024 \quad (21)$$

By considering the interval -1 to +1, the cosine activation function is represented as,

$$K(y) = \begin{cases} 0.5403; y < -1 \\ 1 - y^2/2 + y^4/4 - y^4/16 - y^4/64 \\ - y^4/256 - y^4/1024; -1 \le y < \\ 0.5403; y \ge 1 \end{cases} \quad (22)$$

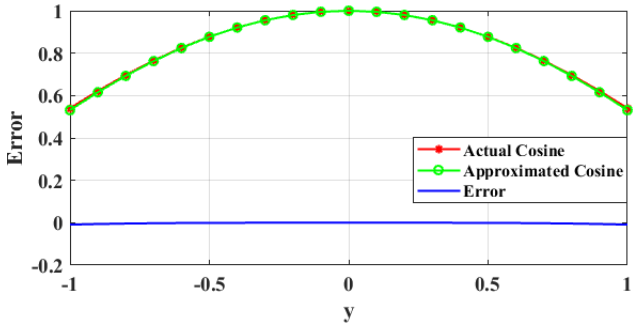

Fig. 4. The error curve between the actual and approximated cosine activation function.

The error curve is plotted in MATLAB environment between the actual cosine activation function given in (18) and approximate cosine activation function in (21). The error curve is shown in Fig. 4 and the error is almost zero. So, the (21) is used to implement the cosine activation function in hardware platform. The hardware architecture of cosine activation function is shown in Fig. 5.

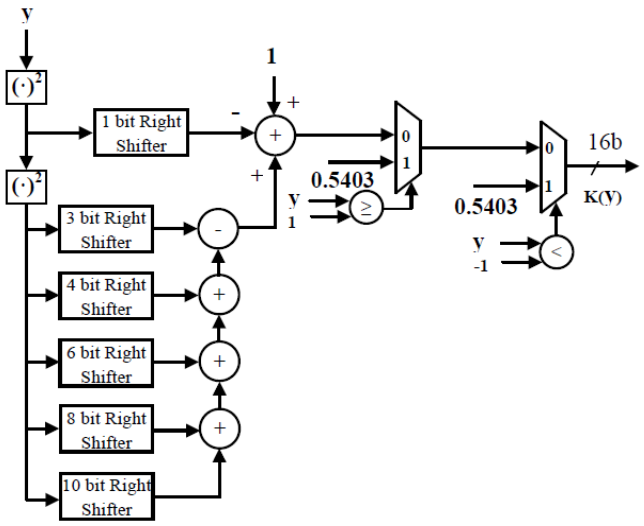

Fig. 5. Hardware implementation architecture of cosine activation function.

The output weight (β) is computed during the training period of the classifier and stored in output weight memory. Finally, the output of the activation unit $H(X)$ and output and accumulate (MAC) unit to compute the desired output $Y(X)$ sequentially taking the control signal from the finite state machine (FSM). The FSM generates the control signals for the entire hardware architecture of WOS-ELM classifier taking the clock (Clk) and reset as inputs. The proposed hardware architecture produces the output when the 'out-ready control' signal is logic high. The state diagram of finite state machine is presented in Fig. 6.

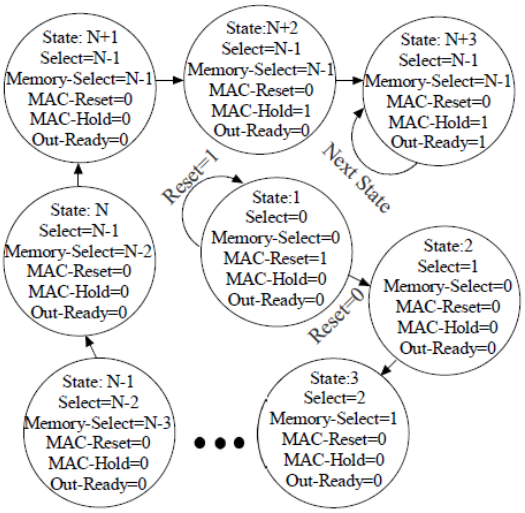

Fig. 6. The state diagram of finite state machine

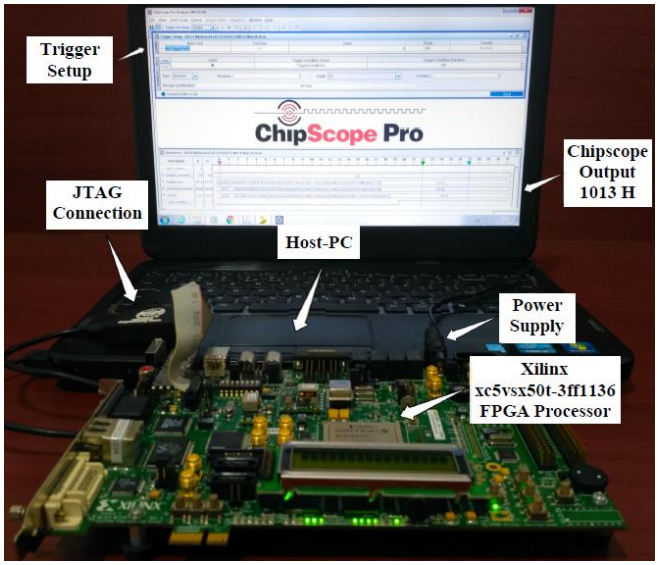

Fig. 7. The hardware interfacing between the host-PC and Xilinx Virtex-5 FPGA for WOS-ELM classifier.

Moreover, to validate the proposed digital architecture in FPGA platform, Xilinx chipscope pro integrated logic analyzer (ILA) debugging environment is used. Fig. 7 shows the hardware interfacing between the host personal computer (PC) and Xilinx Virtex-5 (ML506) FPGA board. The digital architecture of WOS-ELM classifier is simulated using Xilinx ISE design suite 14.5 as shown in Fig. 8.

Fig.9 shows the chipscope pro integrated logic analyzer output of the proposed hardware architecture. From Fig. 8 and 9, authors observed that both the simulation and hardware implementation output have same digital value of 1013 H, i.e., 1.0046 D. The above decimal value indicates that

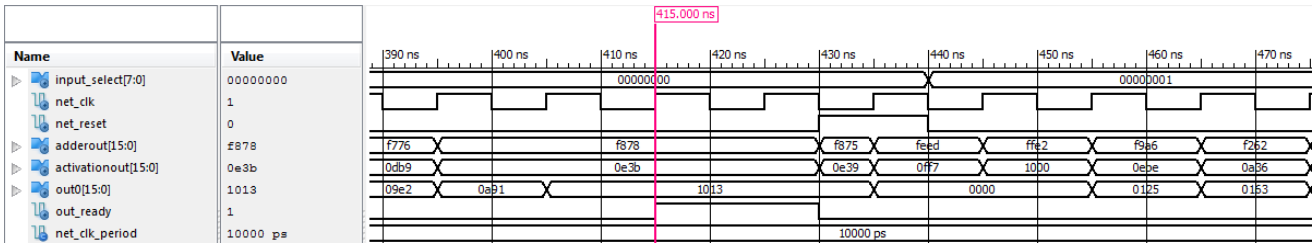

Fig. 8. WOS-ELM hardware implementation architecture simulation output.

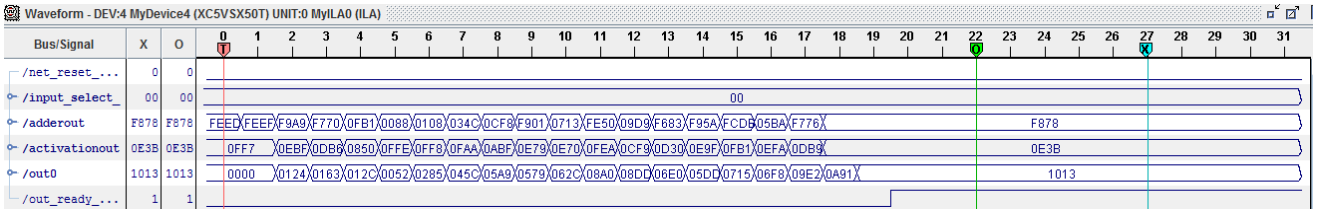

Fig. 9. The chipscope pro integrated logic analyzer output of the proposed hardware architecture.

the given input diabetes data belong to a particular class with an error of 0.0046. The hardware resources used to implement WOS-ELM in digital domain, after place and route with Xilinx ISE design suite 14.5 tool in a Xilinx ML506 FPGA board i.e., xc5vsx50t FPGA with a speed grade of -3. In this experiment diabetes testing dataset is tested by utilizing 63 bounded IOBs, 1877 slice LUTs, two BlockRAMs of size 72 KB, and power consumption of 931mW with clock speed 40.02 MHz. The hardware output is computed in 0.0966 ms with an initial latency of 1.0244 µs. Table II presents the device utilization summary of the proposed hardware architecture of the WOS-ELM classifier.

Table-II: Device Utilization Summary for the Proposed Hardware Architecture of WOS-ELM Classifier.

Logic Utilization	Used	Available	Utilization
Clock Speed (MHz)	40.02	--	--
Arithmetic Precision (bit)	16	--	--
Number of bounded IOBs	63	480	13.12 %
Number of Slice LUTs	1877	32640	5.75 %
Power Consumption (mW)	931	--	--

IV. CONCLUSION

In this paper, authors designed the hardware architecture of WOS-ELM classifier and implemented it in a high-speed reconfigurable FPGA environment to validate the simulation output of the proposed hardware architecture. The superior overall testing classification accuracy, low power consumption, less hardware resources, flexibility, effectiveness, robustness, and short event detection time are the key advantages to make the hardware architecture of the WOS-ELM classifier a promising method to classify the data in real-time environment. The future scope of the article is to implement the hardware architecture of other variants of ELM to achieve better classification accuracy with lesser hardware resources.

REFERENCES

1. J. Misra, and I. Saha, "Artificial neural networks in hardware: A survey of two decades of progress," Neurocomputing, vol. 74, no. 1-3, 2010, pp. 239-255.
2. Y. Wang, Z. Li, L. Feng, H. Bai, and C. Wang, "Hardware design of multiclass SVM classification for epilepsy and epileptic seizure detection," in *IET Circuits, Devices & Systems*, vol. 12, no. 1, pp. 108-115, 2018.
3. R. Dlugosz, T. Talaska, and W. Pedrycz, "Current-Mode Analog Adaptive Mechanism for Ultra-Low-Power Neural Networks," in *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 58, no. 1, pp. 31-35, Jan. 2011.
4. J. Chen, and T. Shibata, "A Neuron-MOS-Based VLSI Implementation of Pulse-Coupled Neural Networks for Image Feature Generation," in *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 57, no. 6, pp. 1143-1153, June 2010.
5. G.B. Huang, and H. A. Babri, "Upper bounds on the number of hidden neurons in feedforward networks with arbitrary bounded nonlinear activation functions," in *IEEE Transactions on Neural Networks*, vol. 9, no. 1, pp. 224-229, Jan. 1998.
6. G.B. Huang, "Learning capability and storage capacity of two-hidden-layer feedforward networks," in *IEEE Transactions on Neural Networks*, vol. 14, no. 2, pp. 274-281, March 2003.
7. G.B. Huang, Q.Y. Zhu, and C.K. Siew, "Extreme learning machine: a new learning scheme of feedforward neural networks," *2004 IEEE International Joint Conference on Neural Networks*, vol.2, pp. 985-990, 2004.
8. G.B. Huang, Q.Y. Zhu, and C.K. Siew, "Extreme learning machine: Theory and applications," Neurocomputing, vol. 70, no. 1–3, pp. 489-501, 2006.
9. G.G. Huang, H. Zhou, X. Ding, and R. Zhang, "Extreme Learning Machine for Regression and Multiclass Classification," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 42, no. 2, pp. 513-529, Apr. 2012.
10. C. Cortes, and V. Vladimir, "Support-vector networks," *Machine learning*, vol. 20, no.3, pp. 273-297, 1995.
11. J.A. Suykens, and J. Vandewalle, "Least squares support vector machine classifiers," *Neural processing letters*, vol.9, no.3, pp. 293-300, 1999.
12. Y. Wang, F. Cao, and Y. Yuan, "A study on effectiveness of extreme learning machine," Neurocomputing, vol. 74, no. 16, pp. 2483-2490, 2011.
13. G.B. Huang, Q.Y. Zhu, and C.K. Siew, "Universal approximation using incremental constructive feedforward networks with random hidden nodes," *IEEE Trans. Neural Networks*, vol. *17*, no. 4, pp.879-892, 2006.
14. G.B. Huang, and L. Chen, "Convex incremental extreme learning machine," Neurocomputing, vol. 70, no.16–18, pp. 3056-3062, 2007.
15. G.B. Huang, and L. Chen, "Enhanced random search based incremental extreme learning machine," Neurocomputing, vol. 71, no.16–18, pp. 3460-3468, 2008.
16. Y. Yang, Y. Wang, and X. Yuan, "Bidirectional Extreme Learning Machine for Regression Problem and Its Learning Effectiveness," in *IEEE Transactions on Neural Networks and Learning Systems*, vol. 23, no. 9, pp. 1498-1505, Sept. 2012.
17. H.J. Rong, Y.S. Ong, A. H. Tan, and Z. Zhu, "A fast pruned-extreme learning machine for classification problem," Neurocomputing, vol. 72, no. 1–3, pp. 359-366, 2003.
18. Y. Miche, A. Sorjamaa, P. Bas, O. Simula, C. Jutten, and A. Lendasse, "OP-ELM: Optimally Pruned Extreme Learning Machine," in *IEEE Transactions on Neural Networks*, vol. 21, no. 1, pp. 158-162, Jan. 2010.
19. R. Zhang, Y. Lan, G. Huang, and Z. Xu, "Universal Approximation of Extreme Learning Machine With Adaptive Growth of Hidden Nodes," in *IEEE Transactions on Neural Networks and Learning Systems*, vol. 23, no. 2, pp. 365-371, Feb. 2012.
20. N.Y. Liang, G.B. Huang, P. Saratchandran, and N. Sundararajan. "A fast and accurate online sequential learning algorithm for feedforward networks," *IEEE Transactions on neural networks*, vol. 17, no. 6, pp.1411-1423, 2006.
21. Y. Ye, S. Squartini, and F. Piazza, "Online sequential extreme learning machine in nonstationary environments," Neurocomputing, vol. 116, pp. 94-101, 2013.
22. S. Decherchi, P. Gastaldo, A. Leoncini, and R. Zunino, "Efficient Digital Implementation of Extreme Learning Machines for Classification," in *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 59, no. 8, pp. 496-500, Aug. 2012.
23. J.V. Frances-Villora, A. Rosado-Muñoz, J. M. Martínez-Villena, M. Bataller-Mompean, J. Fco. Guerrero, and M. Wegrzyn, "Hardware implementation of real-time Extreme Learning Machine in FPGA: Analysis of precision, resource occupation and performance," Computers & Electrical Engineering, vol. 51, pp. 139-156, 2016.
24. C. Blake, and C. J. Merz, "fUCIg repository of machine learning databases," 1998.
25. M. Sahani, and P.K. Dash, "Variational mode decomposition and weighted online sequential extreme learning machine for power quality event patterns recognition," Neurocomputing, Volume 310, pp. 10-27, 2018.
26. C. R. Rao, and S. K. Mitra. "Generalized inverse of a matrix and its applications." *Proceedings of the Sixth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Theory of Statistics*. The Regents of the University of California, 1972.
27. D. Serre, "Matrices: Theory and Applications", Springer-Verlag, New York, Inc., 2002.

AUTHORS PROFILE

Susanta Kumar Rout received the B.Tech. degree in Electronics and Telecommunication Engineering from the Biju Patnaik University of Technology, Rourkela, Odisha, India and M.Tech. degree in Microelectronics from the Siksha 'O' Anusandhan University, Odisha, India, in 2009 and 2013, respectively. He is currently pursuing Ph.D. in Electronics and Telecommunication Engineering at International Institute of Information Technology, Bhubaneswar, Odisha, India. His current research interests include Digital VLSI Architecture Design, Biomedical Signal Processing and Image processing.

Bhanja Kishor Swain received the B.Tech. degree in Instrumentation and Electronics Engineering from the Biju Patnaik University of Technology, Rourkela, Odisha, India and M.Tech. degree in Electronics System and Communication from the National Institute of Technology, Rourkela, Odisha, India, in 2004 and 2007, respectively. He is currently working as an assistant professor in the department of Electrical Engineering, ITER, Siksha 'O' Anusandhan University, Odisha, India. His current research interests include Image Processing, Digital Signal Processing and Artificial Intelligence.

Pradyut Kumar Biswal obtained his B.E. from the Institution of Engineer's, India and M.Tech. from the Visvesvaraya National Institute of Technology, Nagpur, India in 2000 and 2002 respectively. In 2012, he obtained his Ph.D. degree from the Department of Electronics & Electrical Communication Engineering at Indian Institute of Technology, Kharagpur, India. His research interests include Digital Image processing, Biomedical Signal Processing and Image processing, Architecture design for image and signal processing algorithms with FPGA implementations. He is presently positioned as an Assistant Professor in International Institute of Information Technology, Bhubaneswar, Odisha, India.