2017 International Symposium on Nonlinear Theory and Its Applications,
NOLTA2017, Cancun, Mexico, December 4-7, 2017

Winner-Take-All Neural Network with Distributed Winner Search Circuit

Kazuki Hanada†, Shoya Ueda†, Hidetaka Ito†, and Hiroomi Hikawa†

†Faculty of Engineering Science, Kansai University
3–3–35 Yamate-cho, Suita, Osaka 56408680, Japan
Email: k306543@kansai-u.ac.jp, h.ito@kansai-u.ac.jp, hikawa@kansai-u.ac.jp

Abstract—

This paper proposes a new winner-take-all (WTA) circuit, in which winner search is distributed among neurons in the network. The winner neuron that has the shortest vector distance is searched by a simple logic circuit. In the circuit, the shortest vector distance is searched by a bit-by-bit comparison scheme. With this approach, the winner search method is embedded in all neurons, and global winner search module that hinders expandability is not needed. Winner-take all network (WTANN) with the proposed WTA was designed with VHDL, and its performance was examined by simulations and experiments. It was revealed that the speed of the winner search provided by the proposed WTA is extremely faster than the other WTA circuits.

1. Introduction

A winner-take-all neural network (WTANN) is a supervised neural network. Due to its operating nature, WTANN is treated as classifier and it has been adopted to solve problems in pattern recognition of images and voices, or signal processing [1][2]. The most important function required by the WTANN is its ability to determine a winner neuron that has a weight vector nearest to the input vector, which is called a winner-take-all (WTA) operation. The WTA operation is also used in other neural networks, such as self-organizing map (SOM) [3].

Since the neural network includes a parallel structure, a considerable speed-up can be achieved by using parallel or custom architectures implemented on hardware.

This paper proposes a new type of WTA circuit distributed among neurons using numerical computation. In the proposed WTA, vector distances of every neuron are loaded into registers, and they are computed on bit-by-bit base in parallel. With the proposed scheme, comparison process is distributed among all neurons, which communicate with each other by a mutual WTAbus. Additional neurons can be installed by connecting them to the bus, and scalability of WTANN is improved because the number of neurons can be increased easily. Using the proposed WTA circuit, hardware WTANN is implemented on field programmable gate array (FPGA), and its performance as a vector classifier is verified by simulations and experiments.

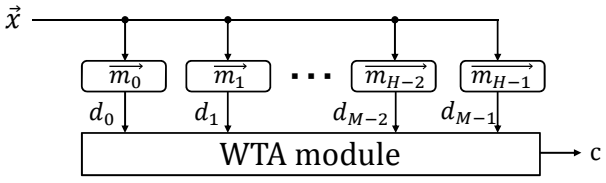

Figure 1: Winner-take-all neural network.

2. Winner-Take-All Neural Network

2.1. WTANN

General structure of WTANN is outlined in Figure 1. The WTANN consists of multiple neurons and the WTA module that finds the winner neuron. Here, H denotes the number of neurons.

A D-dimensional vector, $\vec{m}_i$ called a weight vector, is assigned to all neurons.

$$\vec{m}_i = \{\mu_{i0}, \mu_{i1}, \cdots, \mu_{i(D-1)}\} \tag{1}$$

The operation of WTANN can be divided into two phases, learning and recall. Supervised learning is carried out in the learning phase, and weight vectors of all neurons are trained with a set of training vectors.

In the learning phase, each neuron is assigned to one of the classes to be recognized. A training vector $\vec{x}^t$ that belongs to the assigned class t is given to the neuron.

$$\vec{x}^t = \{\xi_0^t, \xi_1^t, \cdots, \xi_{D-1}^t\} \tag{2}$$

The neuron that is assigned to the class t updates its weight vector toward the training vector.

$$\mu_{ij} = \mu_{ij} + \alpha(\xi_i^t - \mu_{ij}) \tag{3}$$

α is a learning rate ($0 \leq \alpha \leq 1$). By repeating the above computation, the weight vector of each neuron is properly placed in the middle of the vector clusters, each of which belongs to one of the vector classes.

During the recall phase, distances to all weight vectors are calculated for each input vector, and the winner neuron c that has the smallest distance is determined. Euclidean distance is commonly used to measure the vector distance. But many hardware neural networks employ the Manhattan

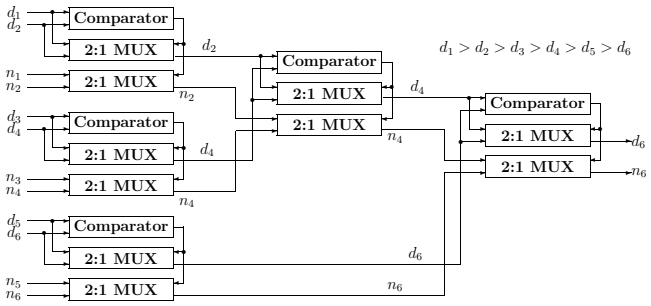

Figure 2: Binary tree search WTA.

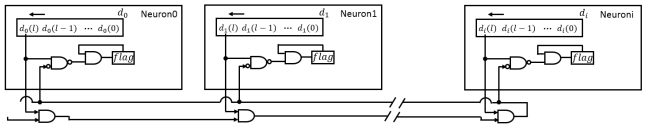

Figure 3: Kohonen's WTA.

metric d_i instead of the Euclidean distance to reduce the computing cost.

$$d_i = \sum_{j=0}^{D-1} |\xi_j - \mu_{ij}| \tag{4}$$

where, ξ_j is a vector element of $\vec{x}$, which is a D-dimensional input vector. The class of $\vec{x}$ is unknown.

$$\vec{x} = \{\xi_0, \xi_1, \cdots, \xi_{D-1}\} \tag{5}$$

The neuron c that has the smallest d_i is chosen as the winner neuron.

$$c = \arg\min_i d_i \tag{6}$$

From the winner neuron, the class of input vector can be identified, and the identified class is given as the vector classification result.

2.2. Related work

Popular winner search circuit method is binary-tree search [4][5]. The WTA circuit based on the binary-tree search is shown in Fig. 2. The search is carried out in tournament fashion. Given distance data are compared in pairs and the smaller one is selected. The selected distances are again compared and selected until the smallest one is found. Kohonen described a very simple winner search circuit that is shown in Fig. 3 [3]. This circuit find the smallest value in serial manner. In the circuit, most significant bit (MSB) of the values are compared, and if the MSB of a value is '1' while any of other value's MSB is '0', then the value is not the smallest one and its $flag$ is set to '0'. By shifting all values to the left, this operation is repeated for all bits. After processing all bits, the node with $flag$ = '1' has the smallest distance.

3. Proposed WTANN

This section describes the proposed WTA circuit based on numerical computation by a simple logic circuit.

3.1. Winner search based on bit-by-bit comparison

The proposed WTANN is shown in Fig. 4. This WTA is carried out through bit-by-bit comparison. The circuit compares all neuron's vector distance bits from MSB toward least significant bit (LSB). All bits in the same position are compared simultaneously, and the bigger ones are excluded from further comparison. At the end of the LSB comparison, the neuron having the smallest distance is identified. Comparison results that indicate whether the neuron is still candidate for the winner or not, are propagated to the lower bit through survival signal S_k. The comparison starts from MSB. N-bit WTAbus is connected to all neuron, and it is driven by tri-state buffers. If any of MSBs in all neurons is '0', then WTAbus$_{N-1}$ ='0', otherwise '1'. If the MSB of a neuron is '0' while WTAbus$_{N-1}$ ='0' then that neuron still has a chance to be a winner, and a survival signal S_{N-2} ='1' is sent to the lower bit. If the MSB of a neuron is '1' while WTAbus$_{N-1}$ ='0', then the neuron is not a winner, and S_{N-2} ='0' is sent to the lower bit. WTAbus$_{N-1}$ ='1' means that all MSBs are '1', therefore S_{N-2} ='1' is propagated to all neurons. Note that the survival signal input to MSB is always S_{N-1} ='1'. If the lower-bit comparison receives S_k ='1' then the same logic operation as the upper-bit is carried out. If the received S_k is '0', then S_{k-1} ='0' is propageted to the next lower bit because that neuron is no longer a candidate for the winner. If a neuron's survival signal at LSB (S_0) is '1', then that neuron is the winner. The proposed winner search operation does not require the clock signal because it is implemented as a combinatorial circuit.

3.2. On-chip learning circuit

The weight vectors $\vec{m}$ in Fig. 1 are provided by an on-chip learning circuit contained in neurons. The on-chip learning circuit performs the computation given in equ. (3), and its block diagram is shown in Fig. 5. The on-chip learning circuit contains D sets of modules. Each module is made of a register, two adders and a multiplier. Each module performs the update of one weight vector element. Note that a bit-shift operation can substitute the multiplier if α is a negative powers of two ($\alpha = 1/2^{\beta}$).

During the on-chip learning, each neuron in the WTANN is assigned to one of the classes. Training vector elements ξ_i^t belonging to that class are fed to the neuron while its `update` signal is activated so that its weight vector elements are updated to be closer to ξ_i^t.

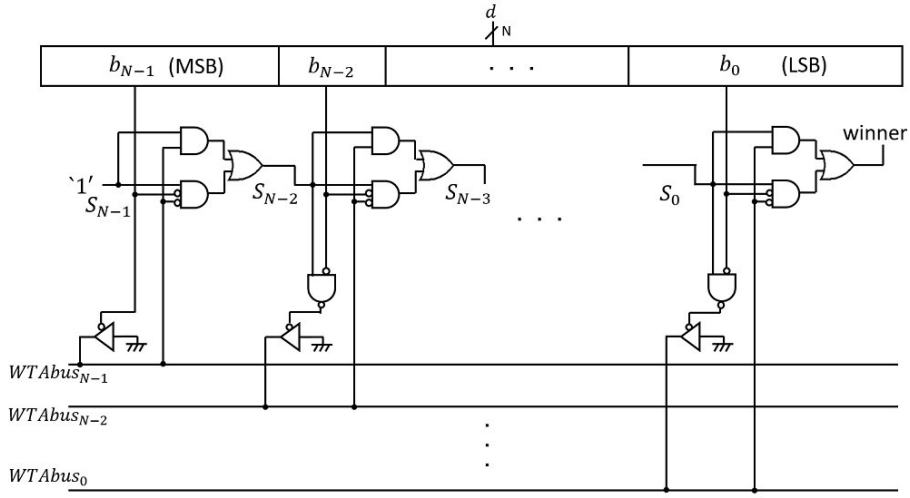

Figure 4: Proposed WTA circuit.

Table 1: IRIS classifications (N=18)

Trial	1	2	3	4	5	6	7	8	9	10	Average
Recognition rate (%)	94.7	96.0	93.3	94.7	94.7	97.3	94.7	96.0	94.7	93.3	94.9 %

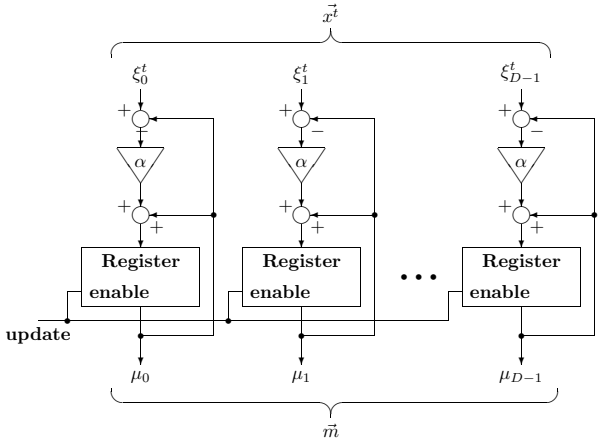

Figure 5: On-chip learning circuit.

4. Verification

This section discusses the properties of WTANN as classifier as well as its hardware obtained by simulations and experiments.

4.1. Classification of IRIS data set

First, the proposed WTANN was described in VHDL, and VHDL simulation was conducted to examine the operation of the WTANN. Iris data set in the machine learning repository (MLR)[6] of the University of California, Irvine (UCI) was employed for the test.

Table 2: Comparison of calssification accuracies.

Proposed	K-NN	LDA	ALH
94.9 %	96.7 %	98.0 %	97.3 %

Half of the instances of each class were used to train the proposed WTANN, and the remaining instances were used as test data to obtain its classification accuracy. Before the classification test, the WTANN was set to the training mode, and the training data were fed to the WTANN, and its weight vectors were updated by the on-chip learning circuit. The learning coefficient in equ. (3) was set to $\alpha = 1/16$, thus the multipliers in Fig. 5 were implemented by using simple 4-bit right shift operations. After the training, all test data were fed to the WTANN and its recognition rate was measured. IRIS test data set was made of 75 instances. This training/test trial was repeated 10 times, where different combinations of instances were assigned to the training and testing data sets, then the average recognition rate was obtained.

Table 1 shows the 10 recognition rates and their average. The average recognition rate was 94.9 %. Tab. 2 compares the recognition accuracy of the roposed WTANN with other algorithms, which are k-nearest neighbor (k-NN), linear discriminant analysis (LDA), and adaptive local hyperplane (ALH) [7]. The table indicates that the recognition performance of the proposed WTANN is a bit lower than others, but it is respectable even though the learning method of the proposed WTANN was very sim-

Table 3: Comparison of the proposed winner search and others.

	Maximum freq. (MHz)			Winner search time (ns)			Circuit size (Slices)		
Number of neurons	16	32	64	16	32	64	16	32	64
Proposed	271	231	206	3.69	4.33	4.85	248	336	531
Kohonen	293	297	294	61.42	60.60	61.22	216	306	488
Binary-tree	183	116	108	5.46	8.62	9.26	302	493	626

ple.

4.2. Network size and speed

The proposed WTANN was implemented on an FPGA, and tested by experiments. Xilinx ISE 14.7 and Spartan-6 XC6SLX16 FPGA were used for the implementation. Hardware cost and operation speed were measured and compared with other methods.

An experiment to examine the relation between the number of neurons and the operation speed of the proposed WTA was carried out. Each neuron used for this test contains only the WTA with pre-defined vector distances. As the operation performance, the highest clock frequencies for different number of neurons were measured. The same experiments were conducted using the binary-tree based WTA and Kohonen WTA to compare with the proposed WTA.

Tab. 3 summarizes the results, which shows the maximum operable clock frequency, corresponding search-time for a single input, and circuit size. While the proposed WTA and Kohonen WTA maintained the same clock frequency even though the number of neurons were increased, the frequency of the binary-tree search based WTA was lowered as the number of neurons increased. Due to its sequential operation that requires more clock cycles to complete the winner search, the search speed of the Kohonen WTA was the slowest. In terms of the search speed, the proposed WTA exhibited the best performance.

In terms of the circuit size, the proposed WTA requires less hardware cost than that of the binary-tree WTA while the Kohonen's WTA was the smallest.

5. Conclusion

This paper proposed a new WTA, in which winner search was distributed among neurons in the network. The winner neuron has the shortest vector distance between its weight vector and input vector. In the WTA, the winner neuron was searched by simple logic circuits embedded in all neurons, and neurons compete with each other to become a winner. As a result, any global winner search module that hinders expandability of the WTANN was no longer needed, which makes it easier to increase the number of neurons.

The proposed WTA consists of bit-by-bit competitive comparison of vector distance in all neurons. All neuron's vector distance bits in the same position were compared from MSB to LSB in parallel. Through the comparison process the bigger distances are dropped out from the comparison, and the smallest one is left as the winner neuron.

The proposed WTA neuron was designed with VHDL and experiments were carried out. The relation between the number of neurons and operable clock frequency for winner search was examined, and it was revealed that the maximum frequency was not affected by the number of neurons. The most notable advantage of the proposed WTA is its high speed winner search. Compared to other WTAs, the required number of time to complete the search was extremely less than those required by other WTA circuits.

It was also revealed that the proposed WTA circuit provide easy expandability to WTANN without degrading operation speed. The proposed WTA can be used for SOM, and the development of hardware SOM with the proposed WTA circuit is underway.

References

[1] R. P. Lipmann, "An introduction to computing with neural nets," *IEEE Acoustics, Speech, and Signal Processing Magazine,* pp. 4-22, April 1987.

[2] P. F. Sum, C. S. Leung, K. S. Tam, G. H. Young, W. K. Kan and L. W. Chan, "Analysis for a Class of Winner-Take-All Model," *IEEE Trans. on Neural Networks,* vol. 10, no. 1, pp.64-71, 1999.

[3] T. Kohonen, *Self-Organizing Maps*, 3rd, ser. Springer Series in Information Sciences 30. New York: Springer-Verlag, 2001.

[4] B. Mailachalam and T. Srikanthan, "Area-time issues in the VLSI implementation of self organizing map neural networks," *Microprocessors and Microsystems*, Vol.26, No.9-10, pp.399-406, 2002.

[5] D.C. Hendry, "Comparator trees for winner-take-all circuits," Neurocomputing, Vol. 62, pp.389-403, 2004.

[6] University of California at lrvine web site, http://www.ics.uci.edu/mlearn/MLRepository.html

[7] T. Yang and V. Kecman, "Adaptive local hyperplane classification," *Neurocomputing*, Vol. 71, pp.2409-3050, 2008.