Multidimensional Systems and Signal Processing, 6, 333–346 (1995)

KWTA Networks and their Applications

TAEK MU KWON AND MICHAEL ZERVAKIS tkwon@ub.d.umn.edu
Department of Computer Engineering, University of Minnesota, Duluth, 271 MWAH, 10 University Dr., Duluth, MN 55812

Received January 8, 1994; Revised May 31, 1994

Abstract. Winner-Take-All (WTA) or K-Winner-Take-All (KWTA) networks have been frequently used as the basic building blocks of complex neural networks. This paper introduces a new selection rule for network connections that implements stable KWTA networks. To widen the applications of WTA networks, a new class of WTA networks is proposed, and their efficient design methods are presented. We demonstrate the properties of the generalized class of WTA networks, through three application examples.

1. Introduction

In many neural network models, the selection of a node that encodes the best match of the testing pattern is an essential part of the processing. This selection process is usually implemented using a class of networks called the WTA (Winner-Take-All) network. Noteworthy systems that use the WTA network are the ART networks [1], [2], Kononen's self-organizing feature map and LVQ [3], the Lippmann's hamming net [4], and DBNN [5].

Due to the important role of the WTA network in the design of neural networks (especially for those networks that utilize competitive learning), the implementation of the WTA networks using various types of node models has been studied [6], [7], [8], [9], [4]. Recently, the WTA network was extended to the K-Winner-Take-All (KWTA) network where the network allows multiple winners [10], [11]. This relaxation in the number of winners is useful in a wider variety of applications. For example, Erlanson et al. [12] developed the analog decoding network of error correcting codes, where noisy vectors transmitted over are correctly decoded at the receiver using a KWTA network.

In this paper we suggest two general design techniques for the KWTA network. The first one develops the design using the recurrent structure of the Hopfield-type network. Through this technique, we extend the work of Majani et al. [10] on the KWTA network to general network connections. More specifically, we develop recurrent networks for both inhibitory and excitatory connections. The second technique is based on a feedforward network structure, in which stability is of minor importance.

Although the KWTA network relaxes the stringent requirement of the conventional WTA network on the number of winners, there is still a need to define other types of WTA networks that can be utilized in a variety of applications. Hereafter, we will refer to the conventional WTA as the pure-WTA network and the term WTA will be used to represent general classes of winner related networks. We introduce a generalized class that includes several new types of WTA networks. One of these networks is called the Kth-WTA (Kth-Winner-Take-ALL) network which identifies the Kth largest out of N values. Similarly, we introduce the LTA (Looser-Take-All) network, which seeks the minimum of input values. The corresponding modification results in the KLTA and the Kth-LTA networks. We further develop the rank-range WTA (RRWTA) network that identifies any range of ranks from the given set of input values. This network structure defines the generalized class of networks that include all the above WTA and LTA structures.

Based on the new class of WTA networks introduced, we demonstrate three application examples: the design of an analog sorting network, an order-statistics filter, and the design of an analog fault-tolerant system. All the networks proposed are readily implemented in hardware with basic electronic components (such as resistors, capacitors, and op amps, etc.), and thus they are attractive for real-time applications.

This paper is organized as follows. Section 2 introduces a new parameter selection in the design of KWTA networks, discusses practical implementation issues, and proves the convergence of the K winners in the appendix. Section 3 introduces a generalized class of WTA networks and proposes a design method based on the KWTA networks developed in Section 2. Section 4 demonstrates three applications that utilize the general WTA networks. Section 5 concludes this paper.

2. Design Methods of the KWTA Network

Two design approaches for KWTA networks are presented in this section. The first one is developed based on a recurrent network structure (the Hopfield network model [13]), and the second one is established using a feedforward network.

2.1. Recurrent KWTA Networks

The development of a KWTA network using the Hopfield network is motivated by two reasons: i) the network is stable, and ii) it is easily implementable in hardware using basic electronic components. The continuous Hopfield network [13] is described by a system of first order differential equations given by

$$C\frac{du_i}{dt} = -\lambda_i u_i + \sum_{j=1}^{N} T_{ij}g(u_j) + I_i, \quad \text{for } i = 1, \ldots, N, \tag{1}$$

where

$$\lambda_i = \frac{1}{\rho_i} + \sum_{j=1}^{N} |T_{ij}| \text{ and } C > 0.$$

The function $g(\cdot)$ is non-linear and is defined as a sigmoid function, which is usually implemented by an op amp. The sigmoid transfer function we consider is expressed as $g(u_i) = \tanh(\beta u_i)$, where u_i is the input, $g(u_i)$ is the output of the amplifier, and β is the sigmoid gain that can be assumed very high (i.e. $\beta \to \infty$). Each parameter T_{ij} is a real number referred to as the connection strength between nodes i and j and is implemented through a resistor. The current I_i is the total external input to node i, and ρ_i is interpreted as the input impedance of the amplifier i. Since the admittance $1/\rho_i$ is usually very small (especially for op amps), it is ignored in the subsequent analysis.

The advantage of utilizing eqn. (1) is that the overall system is guaranteed to converge to a stable equilibrium state, provided that the matrix $T = [T_{ij}]$ is symmetric [13]. Moreover, the dynamics of this model follow the dynamics of electronic circuits, implying that hardware implementation is always feasible.

Traditionally, the WTA or KWTA networks have been designed with inhibitory connections. However we also found that the WTA or KWTA networks can be readily designed using excitatory connections. Overall, networks for both types of connections are characterized by the following two lemmas.

LEMMA 1 *Consider the system given by (1). If its parameters are chosen as*

$$\begin{aligned} &T_{ij} = -w \ \ \forall\, i \neq j, \ \ \text{where} \ \ w > 0 \\ &T_{ii} = a \ \ \forall\, i, \ \ \text{where} \ \ -w < a < w \\ &I_i = w(2K - N), \end{aligned} \tag{2}$$

then the system is a KWTA network. With this parameter selection, only K out of N nodes in the network converge to a positive equilibrium state, and the K nodes are the ones that have the K largest initial values.

LEMMA 2 *Consider the system characterized by (1). If its parameters are chosen by negating w and I_i in (2), i.e.*

$$\begin{aligned} &T_{ij} = w > 0 \ \ \forall\, i \neq j \\ &T_{ii} = a \ \ \forall\, i, \ \ \text{where} \ \ w < a < 3w \\ &I_i = -w(2K - N), \end{aligned} \tag{3}$$

then the system is still a KWTA network.

The proofs of both lemmas are similar, and thus we only show the proof of Lemma 2 in the Appendix. Lemmas 1 and 2 provide a selection of w which generalizes the consideration in ref. [10] that fixes $w = 1$. It is interesting to observe from Lemmas 1 and 2 that there exists a parametric duality between the two network structures.

LEMMA 3 *For every inhibitory design of the KWTA network in Lemma 1, there exists a unique excitatory design of the KWTA network as in Lemma 2.*

The proof is clear from Lemmas 1 and 2.

In actual hardware implementation, we must consider the limitation on the slope of the sigmoid in the transition region. To analyze this aspect, we consider the relation (A.2) in the Appendix at the equilibrium state. By substituting the value of I_i from (3), we obtain

$$|u_i| = \frac{a - w}{(N-1)w + a}. \tag{4}$$

Assuming that N is very large and $a - w \approx w$ is selected, (4) is further simplified to

$$|u_i| \approx \frac{1}{N+1} \tag{5}$$

Hence if N is very large, $|u_i|$ could be very small and could fall within the high-slope region of the sigmoid, in which the dynamics of Lemma 1 and 2 networks fail. However the limit imposed on N by a practical op amp is not much restrictive. Indeed, consider a practical comparator (such as the LM106). The output of a typical comparator changes its transition from 0 to 5 volt during the changes of differential inputs from about 0 millivolt to 0.2 millivolt. This means that u_i must be greater than 0.2×10^{-3} to maintain the proper KWTA dynamics, which yields a bound:

$$\frac{1}{N+1} > 0.2 \times 10^{-3} \tag{6}$$

Thus, we should be able to consider up to $N \approx 5,000$ inputs for KWTA computation using typical comparators. This is usually more than sufficient condition for practical uses of KWTA networks. A more important problem in practical implementation, however, may occur due to the limitations on the fan-in and fan-out of the op-amps. Since such a problem is heavily dependent on the implementation, or on the device characteristics, it will not be discussed here. At this point, we show a SPICE simulation result of a KWTA network that is built using Lemma 1. The implemented prototype network is a 2-WTA network with $N = 3$ and the trajectories of three initial values 1V, 2V, and 3V are shown in Fig. 1. We also constructed a KWTA network using discrete components and obtained the same result as the SPICE simulation.

2.2. *Feedforward Network Design*

It is also possible to design the KWTA network using a feedforward network. Let us define a two-input neural node with a hard-limiter activation, or simply a comparator, as

$$h(x, y) = \begin{cases} 1 & x \geq y \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

The rank of a value r_i out of N values is then readily obtained by

$$\text{rank}(r_i) = \sum_{j=1}^{N} h(r_i, r_j) \tag{8}$$

Thus, if $\text{rank}(r_i) = K$, there must exist K comparators that are in "on" states. This design is illustrated in Fig. 2. The input values are multiplexed until the sum of comparators become equal to K. In actual implementation, a small bias may be added to the comparator to safely derive $h(x, y) = 1$ when $x \approx y$. This design is less noble than the recurrent designs shown in the previous subsection, but derives an efficient sorting-network described in Section 4.1.

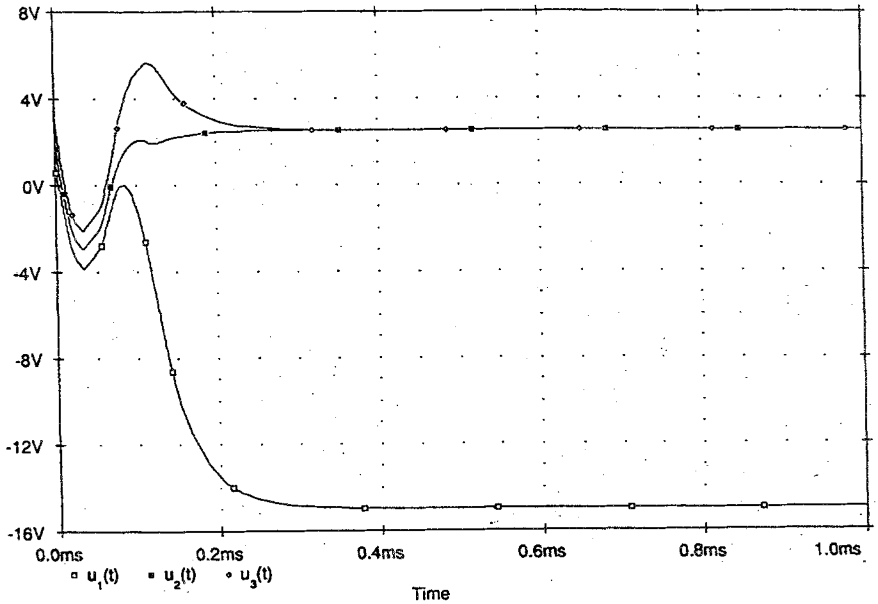

Figure 1. Trajectory of $u_1(t)$, $u_2(t)$, and $u_3(t)$ with initial values: $u_1(0) = 1$, $u_2(0) = 2$, $u_3(0) = 3$.

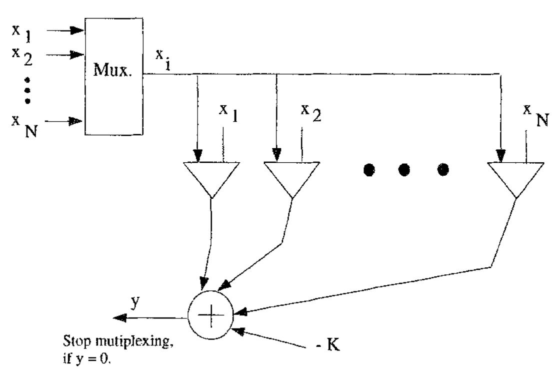

Figure 2. A KWTA network design using a feedforward network.

3. A Generalized Class of WTA Networks

In this section, we develop a new class of WTA networks to generalize the design and broaden the application range of the existing WTA network family. Design examples are provided.

Definition 1. The Kth-WTA network is defined as the network that identifies the Kth largest (Kth winner) from N input values.

The Kth-WTA network includes N nodes that represent the state of input signals. At convergence, the node that corresponds to the Kth ranked value (or the Kth winner) is represented by the on-state of the node model while all other nodes are set to off-states. Notice that the pure-WTA network is a special case of this network, where K is equal to one. Using the KWTA networks developed in Section 2, we can readily design the Kth-WTA network. One particular design technique is presented in the following.

Design 1 (Kth-WTA). The network constructed by the XOR (exculsive or) connection of a KWTA and a $(K - 1)$-WTA network is a Kth-WTA network.

Next we define a variation of the KWTA network, which is referred to as the KLTA (K-Losers-Take-All) network.

Definition 2. The KLTA network is a network that identifies the K smallest (losers) out of N values.

The design of the KLTA network is simple. Since the (N-K)-WTA network has K loosers represented by off-states, the KLTA network can be obtained by inverting the $(N - K)$-WTA network. The Kth-LTA network is defined similar to the Kth-WTA network with the appropriate operational modification.

In Design 1, the Kth-WTA network was designed using XOR connections. We can now present its design based on the KLTA network, where AND connections are used.

Design 2. The network that is constructed by the AND connection of the node outputs of a KWTA and a $(N - K + 1)$-LTA networks is a Kth-WTA network.

In certain applications, we need to identify the values within a specific range of ranks. For example, in signal processing, the a-trimmed mean (TM) filter averages a certain number of signal values around the median [17]. This filter is efficient in removing mixed Gaussian and impulsive noise processes. In order to represent structures similar to the a-TM filter, we introduce a generalization of the WTA network that defines the class of rank-range WTA (RRWTA) networks.

Definition 3. The $(K : L)$-RRWTA network is defined as the network that can identify rank K (Kth largest) through rank L (Lth largest) values where $K < L$.

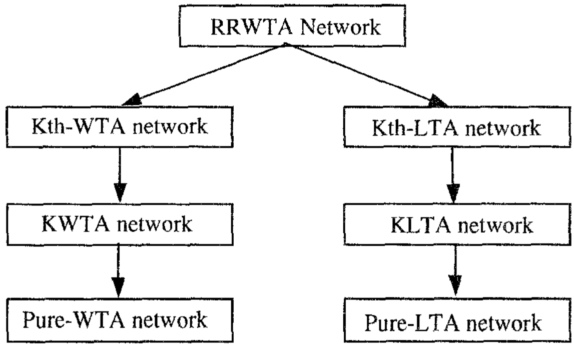

Figure 3. A general class of WTA networks.

It is noteworthy that the RRWTA network generalizes Kth-WTA, which is obtained in the special case that $K = L$. The RRWTA networks can be efficiently implemented using KWTA networks as shown in the following design.

Design 3 ((K:L)-RRWTA). The network constructed by the XOR connection of a $(K - 1)$-WTA and a L-WTA network is a $(K : L)$-RRWTA network.

The generalized class of RRWTA networks includes all the previous WTA and LTA structures. A classification of such networks is presented in Fig. 3, where each level represents a subset of its parent level. A related generalization of order statistic filters is presented in [18].

4. Applications of WTA Networks

This section demonstrates three application examples using the proposed WTA networks.

4.1. Analog Sorting Network

Three design approaches are presented here.

Design 4. One of the simplest way of designing an analog sorting network comprises sequential applications of data to a pure-WTA network. The largest value is identified by the first pass of the N data to the pure-WTA network, then the largest value is removed from the N values, and the remaining $(N - 1)$ values are passed through the same network again, in which step the second largest value is identified. Performing these steps $(N - 1)$ times in total provides a complete set of sorted data. This approach was used in ART1 and ART2 networks [1], [2]. However, due to the nature of sequential search, this approach requires additional processing time and control circuits to monitor both the state of the WTA network and the selection of input values.

Design 5. We propose a different design, which sorts all input values in parallel. In this design, the data is presented to the sorting network once, and the sorted values are simultaneously derived. The information of N sorted outputs is represented by an $N \times N$ two-dimensional array of nodes. Suppose that the five real values $\{r_1, r_2, r_3, r_4, r_5\}$ are sorted through a network with 5×5 nodes. The network may represent the sorted result as:

$$\begin{array}{cccccc} & 1st & 2nd & 3rd & 4th & 5th \\ r_1 & 0 & 0 & 0 & 0 & 1 \\ r_2 & 1 & 0 & 0 & 0 & 0 \\ r_3 & 0 & 1 & 0 & 0 & 0 \\ r_4 & 0 & 0 & 0 & 1 & 0 \\ r_5 & 0 & 0 & 1 & 0 & 0 \end{array} \tag{9}$$

where the node's values 0 and 1 represent off- and on-states of the nodes, respectively. The on state node of the Kth column represents the Kth largest value (or the Kth winner) out of N real values. Thus the sorted order of the above example can be read from left to right as $r_2 > r_3 > r_5 > r_4 > r_1$. With this two-dimensional representation, the states of the Kth-column in (9) are equivalent to the states of the Kth-WTA network. Therefore, a general sorting network can be designed by arranging the Kth-WTA networks in parallel from left to right, 1st-WTA to Nth-WTA network, for sorting N values. Notice that the Kth-WTA network shown in Design 1 uses a KWTA and a $(K-1)$-WTA (except for $K = 1$). From those components, the $(K-1)$-WTA network can be shared by the Kth-WTA network and the $(K-1)$th-WTA network. Thus, for sorting N real numbers, only N KWTA networks $(K = 1, \ldots, N)$ are required.

Design 6. An efficient parallel sorting network can be derived using the feedforward structure of the KWTA network in Section 2.2. This KWTA network utilizes a sequential multiplexing of each input to a row of N comparators. Due to the relation in (8), the multiplexing of each value derives the rank information of the value. Hence, allocating N values to the N rows of N-comparators ($N \times N$ comparator matrix) derives a parallel sorting network that sorts N values. For instance, considering the five values used in Design 5, the resulting output of 5×5 comparators would be

$$\begin{array}{ccccccc} & r_1 & r_2 & r_3 & r_4 & r_5 & \sum \\ r_1 & 1 & 1 & 1 & 1 & 1 & 5 \\ r_2 & 0 & 1 & 0 & 0 & 0 & 1 \\ r_3 & 0 & 1 & 0 & 1 & 0 & 2 \\ r_4 & 0 & 1 & 1 & 1 & 1 & 4 \\ r_5 & 0 & 1 & 1 & 0 & 1 & 3 \end{array} \tag{10}$$

The summation of bits in each row represents the rank of the value fed in that row. For example, $\text{rank}(r_1) = 5$, $\text{rank}(r_2) = 1$, etc. Note also that the summation of the bits in each column exactly represents the reverse order of the sorted result. To decode the sorted analog values, this approach requires a little more complicated decoding circuitry than Design 5. This type of design appears in [16].

Now we evaluate each design in terms of its complexity. The connection complexity of the recurrent KWTA networks designed based on the Hopfield network model is $O(N^2)$, if a completely connected feedback architecture is used. However, using a slight modification on the connection structure by replacing $S = \sum_{j=1}^{N} T_{ij} g(u_j) \quad \forall\ i$ with a single node as suggested in [10], the connection complexity can be reduced down to $O(N)$. Hence, the connections required in Design 4 are $O(N)$. In Design 5, since N KWTA networks are required for sorting N real values, the connection complexity of this network is $O(N^2)$. On the other hand, if the sorting network is designed using an $N \times N$ feedback network (similarly to the Hopfield and Tank's traveling-salesman network [14]), the connection complexity increases to $O(N^4)$. Clearly, such a design is too expensive and impractical. The connection complexity of Design 6 is $O(N)$, the same as Design 5, but this design requires more complicated decoding circuit.

In addition to the connection complexity, we consider the computational time in each design. The network in Design 4 requires $O(N)$ time due to its sequential search, while those of the Design 5 and 6 require only a single time unit, i.e. $O(1)$. Considering that the best computation time for a digital parallel computer to sort N values is $O(N \log N)$ [15], the sorting networks by Design 5 and 6 present a significant advantage in computational time. Moreover, such networks are devoided of A/D and D/A conversion processing, which requires additional hardware and computing time.

4.2. *A Noise Filtering Example*

We consider the cameraman image that consists of 256×256 pixels with 8-bit resolution per pixel. This image is contaminated by 20 dB of white Gaussian noise and 5% of impulsive noise. The result is shown in Fig. 4.a. The variance of the total noise and the Root Mean Square Error (RMSE) of this image from the original image are 286 and 21.8, respectively. For comparison purposes, we first process the image using the conventional Wiener filter implemented in DFT domain with a noise variance of 286. This result is depicted in Fig. 4.b. The RMSE of this estimate is reduced to 12.7, but the Wiener filter is not able to remove the impulsive noise. To reduce the effect of the impulsive noise, the same data is processed using the Wiener filter with a noise variance of 2,860 (10 times larger than the previous case). This estimate yields RMSE equals 15.21 and is depicted in Fig. 4.c. Although the RMSE is increased, the setting of high noise variance in the Winner filter results in a smoother estimate improving the visual quality from the previous case.

Finally, we consider the restoration of the contaminated data in Fig. 4.a using a (K:L)-RRWTA network. For each pixel $p(i, j)$, the 3×3 neighborhood pixels are processed using a (K:L)-RRWTA network with $K = 4$ and $L = 6$, by which the three pixel values having ranks 4 to 6 are selected. Let the intensity of these pixels be denoted by $p_{rank_4}(i, j)$,

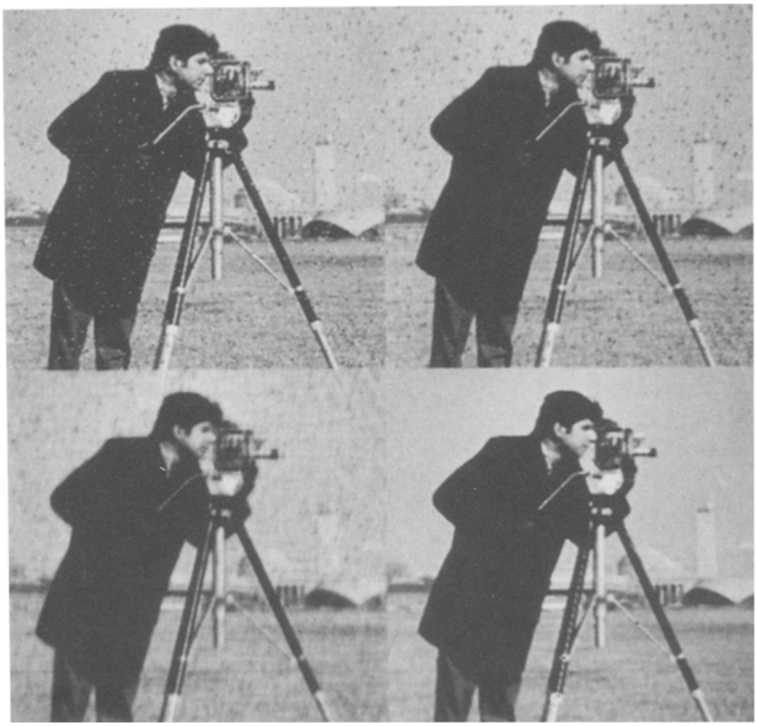

Figure 4. Image restoration example: (a) degraded image by 20 db Gaussian noise and 5% impulsive noise (upper left); (b) DFT Wiener estimate using noise variance 286 (upper right); (c) DFT Wiener estimate using noise variance 2,860 (lower left); (d) estimate of the (K:L)-RRWTA network (lower right).

$p_{rank_5}(i, j)$, and $p_{rank_6}(i, j)$. The filtering is performed by averaging these values:

$$p(i, j) = \frac{p_{rank_4}(i, j) + p_{rank_5}(i, j) + p_{rank_6}(i, j)}{3} \tag{11}$$

for all i and j.

The result of this filtering is depicted in Fig. 4d which has the RMSE equal to 13.19. Notice that the impulsive noise is almost completely removed, presenting a drastic improvement of the visual quality from the estimates of the conventional Wiener filter. This (K:L)-RRWTA network actually implements the so called a-TM filter [17], which is one of the most useful robust L-estimators. Under the influence of impulsive noise processes, the a-TM filter usually works efficiently outperforming conventional filters.

This application on image restoration simply illustrates one example of the proposed WTA networks. Due to the sorting capability of the WTA, the proposed network can implement all types of order statistics filters.

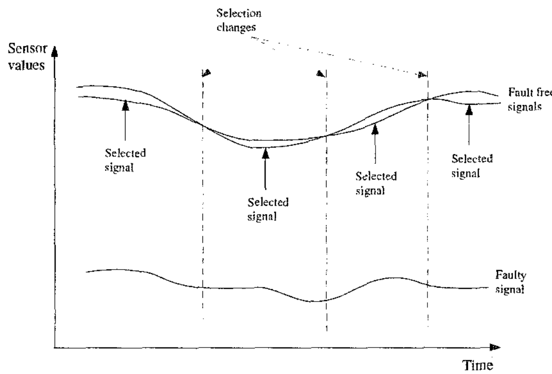

Figure 5. The mid-value select technique

4.3. *Design of an Analog Fault-Tolerant System*

We consider the design of an analog fault tolerant system based on the analog Tripple-Modular Redundancy (TMR). The TMR approach is a frequently used design technique in fault tolerant systems, especially for critical-computation applications [19]. The basic concept of TMR is to triplicate the hardware and perform a majority vote to determine the output of the system. If one of the module becomes faulty, two-remaining fault-free modules mask the result of the faulty module. One major problem arizes if the result of each module is an analog signal. In such cases, the three signals may not completely agree in value even if the system functions with no-fault. One technique that alleviates this problem is the mid-value select technique [19], which selects the middle value of each triplet. This selection technique is illustrated in Fig. 5. Obviously, a 3-input 2nd-WTA network perfectly matches this selection method and thus can be directly used for this application.

5. Conclusion

The main goal of this paper is to generalize WTA networks into a new class of networks that broadens the range of WTA applications. Therefore, several new types of WTA networks and their designs are developed and organized into a general framework of WTA networks. We also explore three possible applications using the proposed WTA networks, which include designs of an analog sorting network, an order-statistics filter, and an analog fault-tolerant system. One of the most attractive features of the proposed WTA networks is the easy and efficient way of implementing them in hardware, which provides a powerful means for real-time signal processing.

Appendix

Proof of Lemma 2. The selection of parameters in (3) derives

$$\lambda_i = \lambda = (N-1)w + a > 0, \quad \forall i. \tag{A.1}$$

Eqn. (1) for the ith node becomes

$$C\frac{du_i}{dt} = -\lambda u_i + (a-w)g(u_i) + (I_i + w\sum_{j=1}^{N} g(u_j)). \tag{A.2}$$

Since the connection matrix imposed by (3) is symmetric, we already know that a system characterized by (A.2) converges to a stable equilibrium state. Let the equilibrium state vector be $\boldsymbol{u}^e$, which satisfies the condition:

$$\frac{du_i^e}{dt} = 0 \quad \forall\, i. \tag{A.3}$$

By applying this equilibrium condition to (A.2), we obtain:

$$\lambda u_i^e = (a-w)g(u_i^e) + (I_i + w\sum_{j=1}^{N} g(u_j^e)). \tag{A.4}$$

Since $\lambda > 0$, the following inequalities are derived:

$$I_i + w\sum_{j=1}^{N} g(u_j^e) > -(a-w)g(u_i^e), \quad \text{for} \quad u_i^e > 0 \tag{A.5}$$

and

$$I_i + w\sum_{j=1}^{N} g(u_j^e) < -(a-w)g(u_i^e), \quad \text{for} \quad u_i^e < 0 \tag{A.6}$$

Due to the monotonously increasing characteristic of the sigmoid $g(\cdot)$, u_i^e and $g(u_i^e)$ always have the same sign. Assuming that the sigmoid function has a very high gain ($\beta \to \infty$), the nonlinearity is approximated by $g(u_i^e) \simeq +1$ or -1 for $u_i^e > 0$ or $u_i^e < 0$, respectively. Since I_i is a constant value independent of i, imposing the combined constraint (A.5) and (A.6) gives

$$-(a-w) < I_i + w\sum_{j=1}^{N} g(u_j^e) < (a-w). \tag{A.7}$$

Let the number of nodes that have $g(u_i^e) \simeq +1$ be K^+. Then, (A.7) can be expressed a

$$-(a-w) < I_i + w(2K^+ - N) < (a-w). \tag{A.8}$$

Substituting the value of I_i given in (3) into (A.8) gives:

$$-\frac{a-w}{2w}+K<K^{+}<\frac{a-w}{2w}+K. \tag{A.9}$$

Following the constraint $w < a < 3w$ from (3), one can readily prove that an integer solution for K^+ is unique and $K^+ = K$. This proves that there exist exactly K out of N nodes on equilibrium state which are in on-states ($u_i^e > 0$) under the constraints in (3).

Next, it is necessary to prove that the K on-states of the network correspond to its K largest initial values. From (A.2), and for $i \neq j$, we can write:

$$C\frac{d(u_i-u_j)}{dt}=-\lambda(u_i-u_j)+(a-w)(g(u_i)-g(u_j)). \tag{A.10}$$

Since $[u_i(0)-u_j(0)]=0 \Rightarrow [u_i(t)-u_j(t)]=0$ for all t, the order of initial values at each node can not be reversed for all t. Hence there exist exactly K nodes with output 1 that are the K largest initial values.

Finally we show that the output of each node at equilibrium is located in the region near the upper or lower bound of the sigmoid, which also justifies the assumption used in (A.7). If we rewrite (A.2) near an equilibrium point of node i, then

$$C\frac{d(u_i-u_i^e)}{dt}=-\lambda(u_i-u_i^e)+(a-w)[g(u_i)-g(u_i^e)]. \tag{A.11}$$

According to the mean-value theorem, there exists x that satisfies

$$g(u_i)-g(u_i^e)=g'(x)(u_i-u_i^e), \tag{A.12}$$

where $u_i < x < u_i^e$. Since u_i and u_i^e can be very close (in the region of equilibrium), we can approximate (A.11) as:

$$C\frac{d(u_i-u_i^e)}{dt}\simeq[-\lambda+(a-w)g'(u_i^e)](u_i-u_i^e). \tag{A.13}$$

Therefore, we obtain the stability condition of an equilibrium state as

$$g'(u_i^e)<\frac{\lambda}{a-w}. \tag{A.14}$$

Substituting the value of λ in (A.1) yields that the slope of the sigmoid at an equilibrium point is bounded by:

$$g'(u_i^e)<\frac{(N-1)w+a}{a-w} \tag{A.15}$$

Since we assumed that $\beta \to \infty$, (A.15) indicates that all the equilibrium points are located in the level regions of the sigmoid (i.e., the region that $|g(u_i^e)| \approx 1$). This condition is always satisfied for all $N \neq \infty$ in theory. In practice, however, the slope of the sigmoid at its transition region is limited, so the number of inputs that can be processed is also limited. This bound is discussed in Section 2. In fact for implementation with commercial comparators, up to $N \approx 5,000$ is satisfied, which is far more than adequate in most practical applications.

■

References

1. G. A. Carpenter and S. Grossberg, "A massively parallel architecture for a self-organizing neural pattern recognition machine," *Computer Visions, Graphics, and Image Processing*, vol. 37, 1983, pp. 54–115.
2. G. A. Carpenter and S. Grossberg, "ART 2: Self-organization of stable category recognition codes for analog output patterns," *Applied Optics*, vol. 26, 1987, pp. 4919–4930.
3. T. Kohonen, *Self-Organization and Associative Memory*, New York: Spring-Verlag, 1988.
4. R. P. Lippmann, B. Gold, and M. L. Malpass, "A comparison of Hamming and Hopfield nets for pattern classification," Technical Report TR-769, Lincoln Laboratories, 1987.
5. S. Y. Kung. *Digital Neural Networks*, Englewood Cliffs, NJ: PTR Prentice-Hall, Inc., 1993.
6. S. Grossberg, "Contour enhancement, short term memory, and constancies in reverberating neural networks," *Studies in Applied Mathematics*, vol. LII (52), no. 3, 1973, pp. 213–257.
7. J. A. Feldman and D. H. Ballard, "Connectionist models and their properties," *Cognitive Science*, vol. 6, 1982, pp. 205–254.
8. J. Lazzaro, S. Ryckebusch, M. A. Mahovald, and C. A. Mead, "Winner-take-all networks of $O(N)$ complexity," in *Advances in Neural Information Processing Systems I*, D. S. Touretzky, Ed., Palo Alto, CA: Morgan Kaufmann, pp. 703–711, 1989.
9. D. S. Touretzky, "Analyzing the energy landscapes of distributed winner-take-all networks," in *Advances in Neural Information Processing Systems I*, D. S. Touretzky, Ed., Palo Alto, CA: Morgan Kaufmann, pp. 626–633, 1989.
10. E. Majani, R. Erlanson, and Y. Abu-Mostafa, "On the k-winners-take-all network," in *Advances in Neural Information Processing Systems I*, D. S. Touretzky, Ed., Palo Alto, CA: Morgan Kaufmann, pp. 634–642, 1989.
11. W. J. Wolfe, D. Mathis, C. Anderson, J. Rothman, M. Gottler, G. Brady, R. Walker, G. Duane, and G. Alaghband, "K-winner network," *IEEE Trans. on Neural Networks*, vol. 2, 1991, pp. 310–315.
12. R. Erlanson and Y. Abu-Mostafa, "Analog neural networks as decoders," in *Advances in Neural Information Processing Systems I*, R. P. Lippmann, J. E. Moody, and D. S. Touretzky, Eds., San Mateo, CA: Morgan Kaufmann, pp. 585–588, 1991.
13. J. J. Hopfield, "Neurons with graded response have collective computational properties like those of two-state neurons," *Proc. Natl. Acad. Sci. USA*, vol. 81, 1984, pp. 3088–3092.
14. J. J. Hopfield and D. W. Tank, "'Neural' computation of decision in optimization problems," *Biol. Cybern.*, vol. 52, 1985, pp. 141–152.
15. R. S. Francis, and I. D. Mathieson, "Benchmark parallel sort for shared memory multiprocessors," *IEEE Tans. on Computers*, vol. C-37, 1988, pp. 1619–1626.
16. P. Shi and R. K. Ward, "OSNet: A Neural Network Implementation of Order Statistic Filters," *IEEE Trans. on Neural Networks*, vol. 4, 1993, pp. 234–241.
17. A. C. Bovic, T. S. Huang, and D. C. Muson, "A generalization of median filtering using combinations of order statistics," *IEEE Trans. on Acoustics, Speech, and Signal Processing*, vol. ASSP-31, 1983, pp. 1342–1350.
18. I. Pitas and A. N. Venetsanopoulos, "Edge detectors based on nonlinear filters," *Signal Processing*, vol. 10, 1986, pp. 395–413.
19. B. Johnson, *Design and Analysis of Fault Tolerant Digital Systems*, Reading, MA: Addison-Wesley, 1989.