ISSN: 2277-9655
Scientific Journal Impact Factor: 3.449
(ISRA), Impact Factor: 1.852

IJESRT

INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH TECHNOLOGY

Delay Constrained Multicast Tree with Neural Networks in Fixed Time

N. Saber, M.Khouil, M. Mestari
Laboratoire SSDIA , ENSET, Mohammedia MAROC

Abstract

This study aimed to propose an artificial neural networks, in fixed time, for the construction of the delay constrained multicast tree. These neural networks are composed mainly of two kinds of neurons: linear and threshold neurons, which are easier to implement in a specialized hardware. Multicast routing consists of transmitting information from a single source to multiple destinations, using the network resources very effectively, and respecting several constraints, such as delay, cost, bandwidth or other. To guarantee an optimal diffusion, it is necessary to determine a tree that connects the source node to all destination nodes minimizing the use of resources. Our approach for solving this problem differs from the conventional approach used in the field of neural networks. Our primary concern is how to organize neurons into a network so that it can solve a specific problem, with an emphasis on fully utilizing the massive parallelism property offered by neural networks.

Keywords: Multicast routing, linear neuron, threshold-logic neuron, minnet, sortnet.

Introduction

Multicast routing is to transmit simultaneously a message from a single source to multiple destinations, subject to minimizing the network resources employed, and satisfying a set of constraints required by the multicast applications. In recent years, this type of routing has attracted the attention of several researchers, because of the fast emerging of multimedia applications such as file sharing, interactive games, videoconferencing....

In graph theory, the problem of multicast routing is called the constrained Steiner tree problem, and is an NP-complete problem [1][2][3].The constrained Steiner problem is to find a Steiner tree that connects the source node to all destinations nodes, such as his cost is minimal and the total delay from the source node to any destination node must be smaller than or equal to a given positive number Δ.

Several algorithms are proposed in the literature to solve routing problem. Chow [4] and Salama et al [5] proposed two exact algorithms for implementing multicast routing, but they are not viable in very large networks, because of their high degree of computational complexity. Heuristics proposed by Kompella et al [6], Widyono [7], Parsa et al [8] are the famous methods used to solve multicast routing problem, because they construct a feasible solution within reasonable time. Many metaheuristics are also proposed to solve this problem: genetic algorithms (Wang et al [9], Tseng et al [10], Lu et al [11]), tabu search (Youssef et al [12]), ant colony optimization (Tseng et al [13], Wang et al [14], Yin et al [15]).

Neural networks are also proposed to solve the multicast routing problem, because of the important property of parallelism that they offer. Rauch and Winarske [16] proposed a modification of the neural networks traveling Salesman algorithm to solve routing problem. In 1982, Hopfield presented the Hopfield Neural Networks [17], since then, many researchers have been exploring HNNs. Pornavalai et al [18] proposed a modification of HNNs to solve constrained multicast routing, but Gee and Prager [19] demonstrated that they are not efficient in large networks. Nozawa [20] and Jain et al [21] proposed a modification of HNNs by adding other properties]. Li and Wang [22] proposed the transiently chaotic neural networks to this problem.

In this paper, we propose a construction of neural networks to solve multicast routing problem in fixed time. For this, we will use only two kinds of neurons: linear and threshold-logic neurons, which are commonly used in neural networks applications [23][24][25][26].

This paper is organized as follows: section 2 will give a formal definition of the problem of multicast routing, section 3 will present the algorithm for the

ISSN: 2277-9655
Scientific Journal Impact Factor: 3.449
(ISRA), Impact Factor: 1.852

construction of the multicast tree, and in section 4 we will propose an implementation of this algorithm by neural networks in fixed time.

Problèm formulation

The communication network is modeled as an undirected graph *G= (V, E, c, d)*, where $V = \{v_1, v_2 \dots v_N\}$ is a set of vertices, and *E* is a set of edges, $E \subseteq \{\{v_i, v_j\} \setminus v_i \in V, v_j \in V,\ v_i \neq v_j, N\}$. $c: E \to IR_+^*$ is a cost function, which associate to any edge *e* a real positive number *c(e)*. $d: E \to IR_+^*$ is a delay function, which associate to any edge *e* real positive number *d(e)* .The cost may measure the monetary utilization or the degree of congestion, and the delay measures the time required to transmit a peace of information through this edge. We denote by *s* the source node, *S* the destination nodes, and *M* the multicast group i.e. $M = \{s \cup S\}$.

A path in a graph *G* is a sequence of vertices $u_1, u_2, \dots u_P$ such that for all $k \in \{1,2, \dots, p\}, \{u_k, u_{k+1}\} \in E$.The cost of a path from u_1 to u_p is equal to $\sum_{k=1}^{p-1} c(\{u_k, u_{k+1}\})$, and its delay is equal to $\sum_{k=1}^{p-1} d(\{u_k, u_{k+1}\})$ .A loop is a path $u_1, u_2, \dots u_P$, such that $u_1 = u_p$.A path is simple if all the vertices on the path are distinct. A shortest path that connects u_1 to u_P is a path from u_1 *to* u_P whose cost is minimal among all possible paths from $u_1$to u_P. *G* is a connected graph if and only if there is a path from any vertex v_i to any other vertex v_j in the graph.A spanning tree of *G* is a connected graph of *G*, without loop, that spans *V*, such as the removal of any edge make it disconnected. The minimal spanning tree of *G* is a spanning tree of *G* whose cost is minimal among all possible spanning trees.

The minimal Steiner tree for *G* and *M* is a minimal tree in *G* that spans *M* and possibly other vertices. The adjacency matrix *A* of *G* is a means of representing which vertices of *G* are adjacent to which other vertices; in other words: $A = (a_{ij})$ is a booleen matrix ,where $1 \leq i,j \leq N$ and:

$$a_{ij} = \begin{cases} 1 & if\ (v_i, v_i) \epsilon E \\ 0 & otherwhise \end{cases} \quad (1)$$

The problem of multicast routing in multimedia is to determine a Steiner tree that connects the source node to all destination nodes, such that the total cost of this tree is minimal ,and the total delay is smaller than or equal to Δ, where Δ can be any positive real number. The Steiner problem is an NP-complete problem.

In this article, we will consider a graph *G* with *N* vertices, numbered from 1 to *N*, and the cost and the delay of each edge are positive, i.e. *c (e)> 0* and *d (e)> 0* . By analogy with the adjacency matrix, *G* can be represented by two square matrices whose coefficients correspond to the costs and the delays of the edges. We denote by $C = (c_{ij})$ and $D = (d_{ij})$ where $1 \leq i \leq N$ *and* $1 \leq j \leq N$ respectiveley the cost and the delay matrices defined by

$$c_{ij} = \begin{cases} c((v_i, v_i)) & if (v_i, v_i) \epsilon E \\ 0 & otherwhise \end{cases} \quad (2)$$

$$d_{ij} = \begin{cases} d((v_i, v_i)) & if (v_i, v_i) \epsilon E \\ 0 & otherwhise \end{cases} \quad (3)$$

Description of algorithm for the multicast tree

In this section, we describe all the steps of the algorithm for multicast tree, which is based on the mathematical model. This algorithm has as input the graph G that models the network and the multicast group M, and the output will be the multicast tree T.
[Inputs: G (V, E, c, d), the multicast group M
Output: multicast tree T]

Step 1: For each edge e ∈ V, dress a list of all edges adjacent to e sorted in ascending order.
We consider that $e_i > e_j$ *if and only if:*

$$\begin{cases} c(e_i) > c(e_j)\ or \\ c(e_i) = c(e_j)\ and\ d(e_i) > d(e_j) \end{cases}$$

Step 2: Initialize the tree T with edge s.
Step 3: Remove s from all lists.
Step 4: Where M is not entirely included in T, then:
Repeat
e ←the nearer edge to T
T ←T+e
Remove e from all lists.

Neural network for multicast routing:

In this section, our main concern is how to implement the proposed algorithm directly in specialized hardware, by neural networks in fixed time.

We will consider in the following that the cost and the delay of an edge are denoted respectively by: $c(e_i) = X_i$ and $d(e_i) = Y_i$.

A-Neurons used:

Two kinds of neurons are employed to implement our networks: the linear and the threshold-logic neurons, with only integer weights (most of the weights being just +1 or -1) and integer threshold. They both assume the well-known linear sum neuron model and differ only in their activation functions: one employs the linear activation function and the other the threshold-logic activation function. Their schematic

ISSN: 2277-9655
Scientific Journal Impact Factor: 3.449
(ISRA), Impact Factor: 1.852

representations are shown in Fig.1 where y is the output.

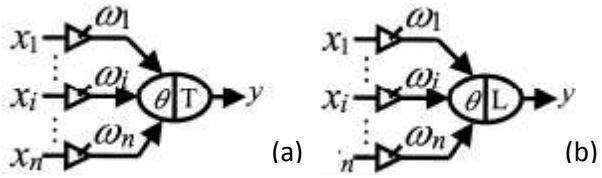

Fig 1: (a) the threshold-logic neuron (b) the linear neuron

y is the output of both neurons, such that:

$$y = \varphi(\sum_{i=1}^{n} w_i x_i - \theta) \quad (4)$$

Where: φ: the activation function

θ : the external threshold

w_i : are the synaptic weights or strengths.

x_i: are the inputs, (i=1, 2, … ,n).

The threshold-logic neuron model uses only the binary function. In this model, a weighted sum of all inputs is compared with a thresholdθ.

$$y = \varphi_T(\sum_{i=1}^{n} w_i x_i - \theta) = \begin{cases} 1 & if \ \sum_{i=1}^{n} w_i x_i \geq \theta \\ 0 & otherwise \end{cases} \quad (5)$$

The linear neuron model uses the linear activation function; the output y is given by:

$$y = \varphi_L(\sum_{i=1}^{n} w_i x_i - \theta) = \sum_{i=1}^{n} w_i x_i - \theta \quad (6)$$

Now that we have defined these two types of neurons, we will explain the way in which they are organized within the networks.

B. Somme basic functions:

In this subsection, we will introduce some basic functions that are essentials for the construction of our networks.

Definition 1: Let (X_i, Y_i) and (X_q, Y_q) be two elements of the input array (X, Y). The comparison function of (X_i, Y_i) and (X_q, Y_q) is defined as follows:

for $i<q$: $comp\left((X_i, Y_i), (X_q, Y_q)\right) =$

$$\begin{cases} 1 \ if \ (X_q, Y_q) \geq (X_i, Y_i) \\ 0 \ if \ (X_q, Y_q) < (X_i, Y_i) \end{cases} \quad (7)$$

and

for $i>q$ $comp\left((X_i, Y_i), (X_q, Y_q)\right) =$

$$\begin{cases} 1 \ if \ (X_q, Y_q) > (X_i, Y_i) \\ 0 \ if \ (X_q, Y_q) \leq (X_i, Y_i) \end{cases} \quad (8)$$

Definition 2:Let (X_q, Y_q) be an element of the input array (X, Y). The order in the input array (X, Y) of (X_q, Y_q) is defined as:

$$ord\left((X_q, Y_q), (X, Y)\right) = \sum_{i<q} comp\left((X_i, Y_i), (X_q, Y_q)\right) + \sum_{i>q} comp\left((X_i, Y_i), (X_q, Y_q)\right) + 1 \quad (9)$$

Definition 3 :let $(X, Y)_{(k)}$denote the k-th largest element of the input array (X, Y).Let (X_q, Y_q) be an element of the input array (X, Y).Then :

$$(X_q, Y_q) = (X, Y)_{(k)} \ \ if \ ord\left((X_q, Y_q), (X, Y)\right) = k.$$

Definition 4: Let (X_i, Y_i) and (X_q, Y_q) be two elemnts of the input array (X, Y).We say that (X_q, Y_q)> (X_i, Y_i) if a nd only if :

1) X_q> X_i or

2) X_q=X_i and Y_q> Y_i

The network for computing the comparison function of (X_i, Y_i) and (X_q, Y_q) given by (7) and (8) is illustrated by the diagram depicted in fig.2.This network is denoted as CN(i,q) (comparison network of (X_i, Y_i) and (X_q, Y_q))

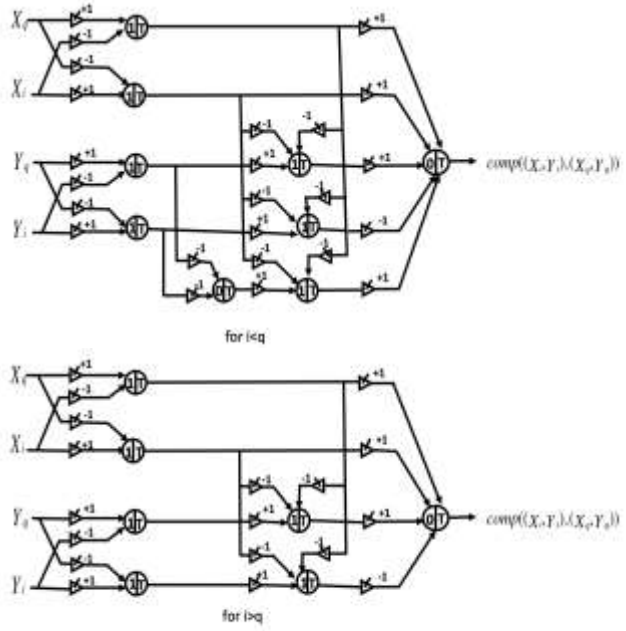

Fig. 2: Comparison network CN(i,q)

C.Order and selection networks:

The function of the order network ON (Fig.3) is to compute the order in the input array (X,Y) of each element (X_q, Y_q). ON consists of (N-1) comparison networks.

ISSN: 2277-9655
Scientific Journal Impact Factor: 3.449
(ISRA), Impact Factor: 1.852

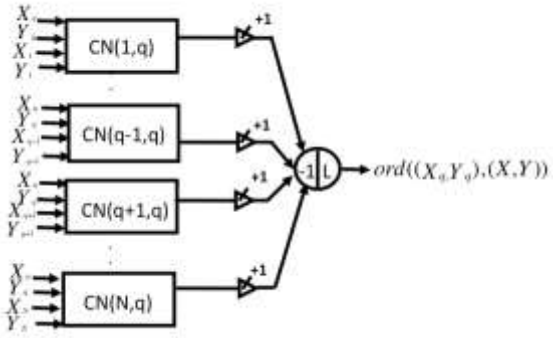

Fig. 3: Order network ON

Figure 4 illustrates the EN: equality network, which determines whether the order of an element is equal or not to a given number k ($1 \le k \le N$). The function computed by the equality network is defined as :

$$eq\left[ord\left((X_i,Y_i),(X_q,Y_q)\right)\right] = \begin{cases} 1 \; if \; ord\left((X_i,Y_i),(X_q,Y_q)\right) = k \\ 0 \; otherwise \end{cases}$$

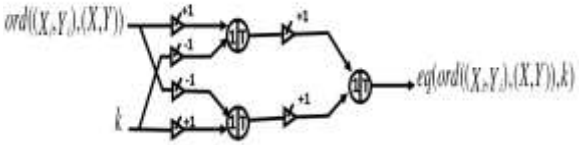

Fig 4: Equality network EN

The DN: detection network is illustrated by Fig.5 .The function of DN is to detect and send to output the k-th largest element $(X,Y)_{(k)}$ of an input array (X,Y)

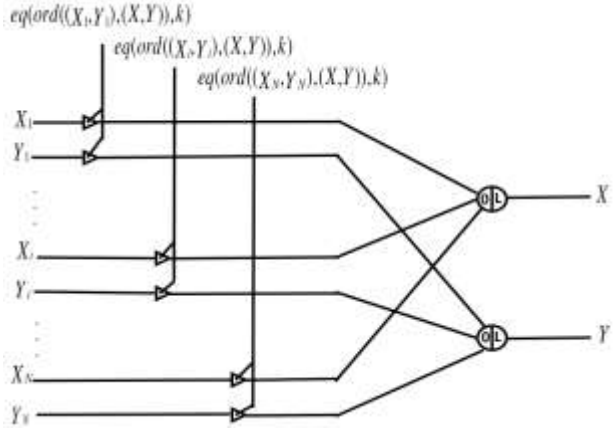

Fig 5: Detection network DN

Figure 6 illustrates the selection network (SN), which selects among all elements of the input array (X,Y) that corresponding to an order k fixed, and transfers it to output. This network is composed of N equality networks (EN) and a detection network (DN).

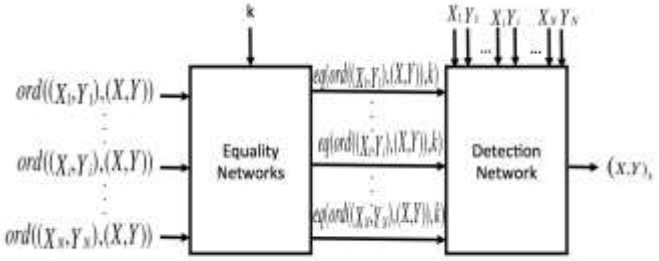

Fig 6: selection network SN

Figure 7 illustrates MINNET , where the input determines which order statistic is to appear at the output. The network shown in Figure 7 consists of two types of neurons arranged in 11 layers.

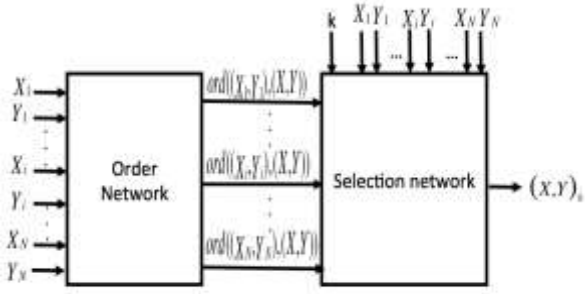

Fig. 7:MINNET

To sort an array, a sorting network must give all order statistics of the array and arranging them either in ascending. An implementation of sorting network (SortNet) is shown in Fig.8. This sorting network is equivalent to N MINNETs implemented in parallel.

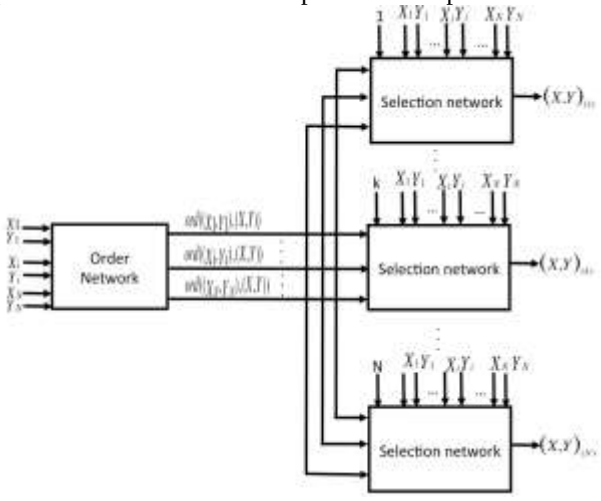

Fig. 8 : Sorting network SortNet

For each vertex of the graph G, a sorting network SortNet sorts the edges adjacent to it in ascending

ISSN: 2277-9655
Scientific Journal Impact Factor: 3.449
(ISRA), Impact Factor: 1.852

order. Recall that the vertices of G are numbered from 1 to N. Then, another MINNET selects the lowest edge and adds it to the final multicast tree.So, by repeting these steps, we will find the multicast tree.

Conclusion

In this paper, we saw a multicast routing algorithm and neural networks for this problem in fixed time. These neural networks are composed mainly of two kinds of neurons: linear and threshold neurons, which are easier to implement in a specialized hardware. These neural networks for the problem of multicast routing in multimedia, have a simple architecture, and require the implementation of a sorting network at each edge, sorting runs in a fixed time. The advantage of this implementation is that regardless of the network size, the sorting is done in parallel, and has a complexity of θ (1) for each edge.

References

[1] *V.Kompella."Multicast Routing Algorithms for multimedia traffic.".Phd thesis*

[2] *F. A. Kuipers and P. Van Mieghem, "MAMCRA: A constrained-based multicast routing algorithm,"Comput. Commun., vol. 25, pp. 802-811,2002*

[3] *P. Venkataram, S. Ghosal, and B. P. V. Kumar, "Neural network based optimal routing algorithm for communication networks," Neural Networks vol. 15, no. 10, pp. 1289–1298, 2002*

[4] *Chow CH. On multicast path finding algorithms. In: Proceedings of INFOCOM; 1991. p. 1274–83R.*

[5] *H. K. Salama, S. D. Reeves, and Y. Vinitois, "Evaluation of Multicast Routing Algorithm for Real-Time networks, " IEEE Journal on Selected Areas in Communications, vol. 25, no. 3, pp. 332-345, Apr. 1997.*

[6] *V. P. Kompella , J. C. Pasquale and G. C. Polyzos "Multicast Routing for Multimedia Communication", IEEE/ACM Trans. Networking, 1993*

[7] *Widyono R. "The design and evaluation of routing algorithms for real-time channels". Technical report TR-94-024. University of California, Berkeley. 1994.*

[8] *M. Parsa, Q. Zhu, J.J. Garcia-Luna-Aceves "An iterative algorithm for delay-constrained minimum-cost multicasting" IEEE/ACM Transactions on Networking, 6 (1998), pp. 461–474*

[9] *Z. Wang, B. Shi, E. Zhao "Bandwidth-delay-constrained least-cost multicast routing based on heuristic genetic algorithm". Computer Communications, 24 (2001), pp. 685–692*

[10] *S.Y. Tseng, Y.M. Huang, C.C. Lin . "Genetic algorithm for delay- and degree-constrained multimedia broadcasting on overlay networks" .Computer Communications, 29 (2006), pp. 3625–3632.*

[11] *T. Lu, J. Zhu . "A genetic algorithm for finding a path subject to two constraints". Applied Soft Computing, 13 (2013), pp. 891–898*

[12] *H. Youssef, A. Al-Mulhem, S.M. Sait, M.A. Tahir. "QoS driven multicast tree generation using Tabu search".Computer Communications, 25 (2002), pp. 1140–1149*

[13] *S.Y. Tseng, C.C. Lin, Y.M. Huang. "Ant colony-based algorithm for constructing broadcasting tree with degree and delay constraints". Expert Systems with Applications, 35 (2008), pp. 1473–1481*

[14] *J. Wang, E. Osagie, P. Thulasiraman, R.K. Thulasiram, HOPNET: a hybrid ant colony optimization routing algorithm for mobile ad hoc network, Ad Hoc Networks 7 (4) (2009) 690–705.*

[15] *P. Yin ,R. Chang, C. Chao, Y. Chu. " Niched ant colony optimization with colony guides for QoS multicast routing". Journal of Network and Computer Applications Volume 40, April 2014,*

[16] *Rauch, H.E., Winarske." Neural Networks for Routing Communication Traffic".IEEE Cont. Syst. Mag. 8(2) (1988) 26-3*

[17] *J.J. Hopfield, "Neural Networks and Physical Systems with Emergent Collective Computational Abilities," Proc. Nat'l Academy of Sciences, vol. 79, pp. 2554-2558, 1982.*

[18] *C. Pornavalai, G. Chakraborty, and N. Shiratori, "A Neural Network Approach to Multicast Routing in Real-Time Communication Networks," Proc. Int'l Conf. Network Protocols (ICNP '95), pp. 332-339, 1995*

[19] *A.H. Gee and R.W. Prager, "Limitations of Neural Networks for Solving Traveling Salesman Problems," IEEE Trans. Neural Networks,* vol. 6, no. 1, pp. 280-282, 1995.

[20] H. Nozawa, "A Neural-Network Model as a Globally Coupled Map and Applications Based on Chaos," Chaos, vol. 2, no. 3, pp. 377-386, 1992.

[21] Jain, S., & Sharma, J. D. Delay bound multicast routing using hopfield neural network.

ISSN: 2277-9655
Scientific Journal Impact Factor: 3.449
(ISRA), Impact Factor: 1.852

International Journal of Computer Theory and, Engineering, *2*(3), 1793–8201. 2010.

[22] L. Wang, S. Li, F. Tian, and X. Fu,"A noisy chaotic neural network for solving combinatorial optimization problems: Stochastic chaotic simulated annealing,"IEEE Trans. Syst. Man Cybern. B, Cybern., vol.34, no. 5, pp. 2119–2125, Oct. 2004.

[23] M. Mestari, "An analog Neural Network Implementation in Fixed Time of Adjutable-Order Statistic Filters and Applications", IEEE Transactions On Neural Networks Vol. 15, No 3, pp. 766-785, May 2004.

[24] M. Mestari and A. Namir, "AMAXNET: A neural network implementation of adjustable MAXNET in fixed time". IFAC-IFIP- IMACS Proc. Internat. Conf. on Control of Industrial Systems, 20-22 May, Belfort, (France), vol. 2, 543-549, 1997.

[25] M. Mestari and A. Namir, "AOSNET: A Neural Network Implementation of Adjustable Order Statistic Filters In Fixed Time", SAMS, 2000, Vol. 36, pp. 509-535.

[26] M. Mestari, A. Namir, K. Akodadi and A. Badi,"θ(1) Time Neural Network Minimum Distance Classifier and its Application to Optical Character Recognition Problem", Applied Mathematical Sciences, Vol. 2, 2008, no. 26, 1253 - 1282, November 2007.