საქართველოს მეცნიერებათა ეროვნული აკადემიის მოამბე, ტ. 8, №2, 2014
BULLETIN OF THE GEORGIAN NATIONAL ACADEMY OF SCIENCES, vol. 8, no. 2, 2014

Informatics

Determining the Number of Neurons Using the Cluster Identification Methods

Ramaz Khurodze*, Otar Verulava, Mariam Chkhaidze****

* *Academy Member, Georgian Technical University, Tbilisi*
***Faculty of Informatics and Control Systems, Georgian Technical University, Tbilisi*

ABSTRACT. It is known that the number of neurons in the layers in the modern neural networks used for recognition are determined on the basis of experiments without theoretical justification according to the heuristic considerations. In the presented paper a procedure for determining the number of neurons in the layers based on clustering is proposed. Formal neuron's recognizing function, "scalar product", as similarity measure for clustering is used. It provides high level of clustering and recognition identity. On the first stage the number of neurons is determined for only one pattern. On the next stage the correction of the number of neurons takes place taking into consideration other system descriptions (templates). *© 2014 Bull. Georg. Natl. Acad. Sci.*

Key words: *neuron, recognition, learning, clustering*

Recognition using neural network for determining the number of neurons for each pattern, which provides error-free recognition of training set, is discussed. It is known that the numbers of neurons in the layers in the modern neural networks used for recognition are determined on the basis of experiments without theoretical justification according to the heuristic considerations [2]. In the presented paper a procedure for determining the number of neurons in the layers based on clustering is proposed. Formal neuron's recognizing function "scalar product" as similarity measure for clustering is used. It provides high level of clustering and recognition identity [3]. On the first stage the number of neurons is determined for only one pattern. On the next stage the correction of the number of neurons takes place taking into consideration other system descriptions. Introduce the following symbols:

Set of patterns A, set of neurons Ne;

$\{A\} = A_1, A_2, \ldots, A_i, \ldots, A_I$; Card $\{A\} = I$;

$\{Ne\} = Ne_1, Ne_2, \ldots, Ne_i, \ldots, Ne_I$; Card $\{Ne\} = I$.

Realization patterns: *X* set

$\{X\} = \{X_1\}, \{X_2\}, \ldots, \{X_i\}, \ldots \{X_I\}$; $\forall \{X_i\} \in \{X\}$, A_i – pattern realizations; subsets of *X*-set;

$\{x\} = x_1, x_2, \dots, x_n, \dots, x_N - A$ - patterns signs; sign space;

$x_i = x_{1i}, x_{2i}, \dots, x_{ni}, \dots, x_{Ni} - A_i$ - patterns signs; $i = \overline{1, I}$;

Weighing coefficients: W– set;

$\{W\} = \omega_1, \omega_2, \dots, \omega_i, \dots, \omega_I$,

$\{W_i\} = \omega_{1i}, \omega_{2i}, \dots, \omega_{ni}, \dots, \omega_{NI}$;

The weighing coefficients of A_i- patterns neuron Ne_i;

$\Rightarrow$ - is received, $\overset{0}{\Rightarrow}$ - is presented.

$X = \{x\}$ – unknown realization; $Ne = \{W\}$ – formal neuron, weighing coefficients;

$$X \overset{0}{\Rightarrow} Ne = \{W\} \Rightarrow \{W \cdot X\} = \sum_n w_n x_n = net \overset{0}{\Rightarrow} z \Rightarrow F(net) \Rightarrow$$

$$\Rightarrow \begin{cases} F(\cdot) \geq Z \Rightarrow out = 1; & X \in A \\ F(\cdot) < Z \Rightarrow out = 0; & X \notin A \end{cases}$$

Recognition process for A_i - pattern Ne_i neuron:

$$X \overset{0}{\Rightarrow} Ne_i = \{W_i\} \Rightarrow \{W_i \cdot X\} = \sum_n w_{in} x_n = net_i \overset{0}{\Rightarrow} z_i \Rightarrow F_i(net)$$

$$\Rightarrow \begin{cases} F_i(\cdot) \geq Z \Rightarrow & X \in A_i \\ F_i(\cdot) < Z \Rightarrow & X \notin A_i \end{cases}$$

Correct recognition process for A_i - pattern Ne_i neuron:

$$X_i \overset{0}{\Rightarrow} Ne_i = \{W_i \cdot X_i\} = \sum_n w_{in} x_{in} = net_i \overset{0}{\Rightarrow} z_i \Rightarrow F_i(net) \Rightarrow \begin{cases} F_i \geq Z_i \Rightarrow & X_i \in A_i \\ F_i < Z_i \Rightarrow & X_i \notin A_i \end{cases}$$

The process of recognition in a formal neuron can be imagined in the following way: the unknown *X* realization is the ordered sequence of the weighing coefficients: a vector or a matrix. Realization coordinates are multiplied by the corresponding indexes of weighing coefficients elements whereupon the obtained products (vector or matrix elements) are added. The obtained value of the total is presented to the previously selected neuron activation function and thus a corresponding point of the value of the total (activation function value) is obtained. The value of activation function is compared to the neuron threshold (the difference is calculated). If the activation function value is bigger than or equals to the neuron threshold value, then neuron output value equals to one, otherwise (when it is smaller) – to zero. The result of the recognition process can be correct or incorrect [3]. The recognition is correct if to a certain kind of neuron the same kind of realization is presented and, as a result, at the neuron output we obtain the value which equals to one. This means that realization belongs to "the same kind". The recognition is also correct when to one pattern of neuron a different type of realization is presented and as a result we get a value which equals to zero. This means that one type of realization does not belong to a different pattern. Both cases are given in (1) and (2) according to the sequence given in the text and in consideration of the symbols adopted.

$$X_j \overset{0}{\Rightarrow} Ne_j = W_j \Rightarrow net_j = \sum_n \omega_{nj} x_{nj} \Rightarrow F_i(net_j) \overset{0}{\Rightarrow} z_j \Rightarrow out_j = 1 \Rightarrow X_j \in A_j \,. \quad (1)$$

$F_j(\mathrm{net}_j)$ activation function value at presenting X_j realization for Ne_j neuron;

$$X_i \overset{0}{\Longrightarrow} Ne_j = W_j \Rightarrow \mathrm{net}_j = \sum_n \omega_{nj} x_{ni} \Rightarrow F_i(\mathrm{net}_j) \overset{0}{\Longrightarrow} z_j \Rightarrow \mathrm{out}_{ji} = 0 \Rightarrow X_j \notin A_j \quad (2)$$

$F_j(\mathrm{net}_j)$ activation function value at presenting X_i realization for Ne_i neuron.

The realization is not correct, if to A_j pattern neuron the same kind of X_i realization is presented for recognition and, as a result, we get $\mathrm{out}_j = 0$, which means that $X_i \notin A_j$.

The realization is also incorrect, if to Ne_j neuron a different kind of X_i realization is presented for recognition; as a result we get $\mathrm{out}_j = 1$, which means that $X_i \in A_j$.

The incorrect recognition process is given in (3) and (4).

$$X_j \overset{0}{\Longrightarrow} Ne_j \Rightarrow \mathrm{net}_j \Rightarrow F_j(\mathrm{net}_j) \overset{0}{\Longrightarrow} z_j \Rightarrow \mathrm{out}_j = 0 \Rightarrow X_j \notin A_j, \quad (3)$$

$$X_j \overset{0}{\Longrightarrow} Ne_j \Rightarrow \mathrm{net}_j \Rightarrow F_j(\mathrm{net}_j) \overset{0}{\Longrightarrow} z_j \Rightarrow \mathrm{out}_j = 1 \Rightarrow X_i \in A_j. \quad (4)$$

Errors in recognition process occur when (3) or (4) are applied. Analytically, both cases can be represented separately, according to (1):

$$F_i(\mathrm{net}_j) = F_j\left(\sum_n \omega_{nj} X_{nj}\right) < z_j \Rightarrow \mathrm{out}_j = 0 \quad (5)$$

according to (2):

$$F_i(\mathrm{net}_j) = F_i\left(\sum_n \omega_{nj} X_{ni}\right) \ge z_j \Rightarrow \mathrm{out}_j = 1. \quad (6)$$

It is obvious that in case "a" the recognition result will change i.e. the error will be corrected if the inequality symbol in (5) is changed:

$$F_i(\mathrm{net}_j) \ge z_j. \quad (7)$$

In case "b" the error can also be corrected by changing the inequality symbol in (6):

$$F_i(\mathrm{net}_j) < z_j. \quad (8)$$

In order to correct the error in (6) it is necessary to increase W_j weighing coefficient, i.e. implement the awarding procedure or else decrease z_j threshold. In case of (7), the error will be corrected, if we decrease the value of W_i coefficient or increase the value of z_j threshold. From the above mentioned it is obvious that to fulfill the conditions given in (6) and (7) it is necessary to perform mutually exclusive procedures.

In general, the errors made during the recognition process are corrected by means of applying one or several of the procedures stated below: Improving the realizations of patterns which is obtained by perfecting the sign space or through preparation procedures; Describing the patterns and perfecting the templates; Correct selected comparison procedures similarities through the interpretation of their results; Changing neuron threshold (z_i). Unknown realization is included into each element of a neuron set, the product of multiplying realization coordinates by weighing coefficients is received, i.e. the net set for each neuron determines values of activation function, and chooses the maximum F_i value in the case shown in (4), according to which a decision is made: $X \in A_j$ for those patterns, the neurons of which do not have the maximum value, we have: $X \notin A_i$, where $i = \overline{1, I}$, $i \ne j$,

$$\xrightarrow{X} \{Ne\} \to \{net\} \to \{F(net)\} \to F_j = \max\{F(net)\} \xrightarrow{X \in A_j} . \tag{9}$$

The improvement of realizations – preparation procedures imply that sign space dimension is definite and constant (fixed) for all the realizations of recognizable patterns. Apart from this, in our case it appears that we have binary realizations (vectors, matrices), where each sign (property) is presented by one pixel. If the pixel contains an expression element, e.g. a fragment of Georgian symbol, then the value of the corresponding sign equals to one, but if it does not – to zero. Perfection of etalon designations implies carrying out such procedures, which can result in correcting the introduced error; at the same time the results through which the realizations will be correctly recognized should not be changed. Let us take into consideration that in the process of changing the weighing coefficients we use only awarding procedures.

In a formal neuron the measure of similarity is the product of multiplying weighing coefficients by corresponding realization coordinates which is commonly called scalar product, but unlike this the results obtained through the product are not multiplied by the transcendental function, which is of no importance because we have a similarity function of binary realizations for the coordinate product. We have:

$$F_{ij} = \sum_n x_{ni} \cdot x_{nj} \geq 0 . \tag{10}$$

A non-negativity constraint is carried out because according to it $\forall x = \{0;1\}$ we have two A_j and A_i type realizations for scalar product:

$$\sum_n x_{nj} \cdot x_{ni} = \sum_n x_{ni} \cdot x_{nj} , \tag{11}$$

which means that symmetry condition is fulfilled $F_{ij} = F_{ji}$.

The product of realization of any type multiplied by itself does always result in zero value:

$$\sum_n x_{ni} \cdot x_{nj} \neq 0 , \tag{12}$$

which means that the reflexive condition is not fulfilled, accordingly, the measure of similarity a scalar product is not a metric function. According to the above-mentioned it is necessary to choose as a similarity measure such a function which will be connected with the scalar product function and, correspondingly, with the algorithm of a neuron recognition. Let us use the coincidence and non-coincidence selection, which was discussed in [3] according to the condition, pattern realizations are evenly-dimensioned binary vectors or matrices, therefore to describe their coincidence or non-coincidence it is advisable to use the apparatus of logical functions. For describing the coincidence of X_i and X_j realizations we will have:

a) $x_{ni} = x_{nj} = 1;$

b) $x_{ni} = x_{nj} = 0;$

where $n \in N$. For non-coincidence we have different values of X_i and X_j realization coordinates:

c) $(x_{ni} = 1) \cup (x_{nj} = 0)$;

d) $x_{ni} = 0 \cup x_{nj} = 1; \quad n \in N$.

Due the fact that we are interested in those values of one realization, where its coordinates are equal to one, we have chosen “a” and “c” points from “a”, “b” and “c” points which reflect the coincidence or non-coincidence of the given, for example, X_i realization with any other realization. Let us present both cases with the scalar product below:

$$\sum_n x_{ni} x_{nj} . \tag{13}$$

It is obvious that any corresponding value of the function given in (13) is equal to one only if $x_{ni} = 1 \cup x_{nj} = 1$, which corresponds to point "a". For presenting the non-coincidence we will have the following scalar product function:

$$\sum_n x_{ni} (1 - x_{nj}) . \tag{14}$$

It is evident that the value of any summand for (14) function equals to one only if condition $x_{ni} = 1 \cup x_{nj} = 0$ is fulfilled which concerns point "c" because the non-coincidence is calculated by means the coincidence function, e.g.,

$$\sum_n x_{ni} - \sum_n x_{ni} (1 - x_{nj}) ,$$

Therefore, hereafter we will use (13) function by means of which we will identify unknown realizations, conduct a training process and carry out clusterization.

A training process in a neuron implies changing weighing coefficients. The training process starts with determining the value of weighing coefficients or random conferring. In our case we will give the weighing coefficients the values equal to one.

$$W = \{\omega_n\}; \ \ \forall \omega_n = 1; \quad n = \overline{1; N} \cdot \tag{15}$$

For conducting a training process it is necessary to know the set of unknown patterns and create a training combination of realizations for each element, e.g. for A_i set $\{X_i\}$according to (15) by the beginning of the training process will be:

$$W_i = \{\omega_{ni}\}; \quad \forall \omega_{ni} = 1; \tag{16}$$

We will have that the set of training combinations of any kind meets the representativeness conditions. In the process of training, a recognition procedure with scalar product similarity zone is used; e.g. in the process of presenting X_j realization its coordinates are multiplied by A_j - type neuron ω_j and weighing coefficient as the product of $\forall \omega_j = 1$; equals to one only when x coordinate of X_j realization is equal to one. Let us present this process as in (9):

$$X_j \xRightarrow{0} W_j = \{\omega_{nj}\} \Rightarrow \textstyle\sum_n x_{nj}\omega_{nj} \longrightarrow F\left(\sum_n x_{nj}\omega_{nj}\right) \xRightarrow{0} z_j \Rightarrow out_j = \{0,1\} \tag{17}$$

The weighing coefficients of those members of (17) product the x_{nj} coordinates of which equal to one, will be awarded according to the given procedure:

$$\omega_{ni}[k] = \omega_{ni}[k-1] + a, \ldots \tag{18}$$

where $k = 0; 1; 2; \ldots; K$, $\omega_n = 1$, according to (2) $i = \overline{1, I}$; $n = \overline{1, N}$,$a = const$ is the awarding coefficient the selection and determination of which as, a rule, occurs heuristically from the whole set of non-negative numbers. If a = 1 then after presenting the whole training set we get the so-called statistical etalons. If $x_{ni} = 0$, then awarding will not take place and the value of the corresponding weighing coefficient will not change. By conducting procedure (3) we get A_i -type etalon description E_i^0 for each realization of A_i -type training set the dimension of which equals to the isolation of the signs space while the coordinates get values

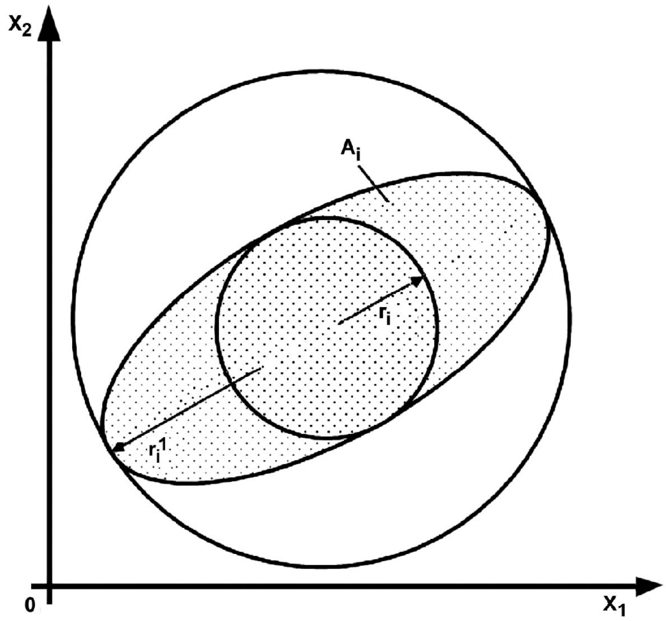

Fig. 1. The indicated situation in two-dimensional sign is shown.

from the set of non-negative whole numbers.

Let us assume that by conducting procedures (17) and (18) for each element (pattern) of the given set of patterns we get a set of pattern descriptions *E*, which contains the same number of elements as in the set of patterns. According to the condition we have as many neurons as the number of the set of recognizable patterns of elements:

$$\mathrm{Card}\{A\} = \mathrm{Card}\{Ne\}, \ \{E\} = E_1, \ E_2, \ \ldots, \ E_i, \ \ldots, \ E_I . \qquad (19)$$

Originally, at the beginning of the training process it was assumed that one neuron represents one pattern (description). After conducting the recognition process it turned out that we had certain errors in this process which were expressed by means of one type of gradation, e.g. by differences in fonts of typed texts, different distortions and disturbances; it became obvious that we need more than one neuron for problem patterns, accordingly, it became necessary to determine the criteria for finding the right number of neurons necessary for the correct recognition of each pattern. Let us mark by M_i the power of the set of A_i pattern training set of realizations.

$$M_i = \mathrm{Card}\{X_i\}; \ \ i = \overline{1, I} \qquad (20)$$

Let us mark by X_i^{mi} any realization of X_i set which we will present to Ne_i for recognition the weighing coefficients of which are the elements of E_i vector (matrix) and according to (17) we will get:

$$X_i^{mi} \Rightarrow W_i = E_i \Rightarrow net_i^{mi} = \sum\nolimits_n x_{nj}^{mi} \omega_{ni} \Rightarrow F(\cdot) \Rightarrow out_i^{mi}, \ \ m_i = \overline{1; M_i}, \ \ i = \overline{1, I} \qquad (21)$$

If we repeat the (26) procedure for each element of X_i set, we will get a set of net_i^{mi} neuron reactions, according to which the recognition i.e. the determination of out_i^{mi} occurs. Let us assume that we get the correct recognition $out_i = 1$ from M_i number of realizations for P_i number of realizations; for the number of incorrectly recognized realizations, then we will have:

$$Q_i = M_i - P_i, \qquad i = \overline{1, I} \qquad (22)$$

Let us mark by net_i^{pi} the value of neural reaction of the correctly recognized X_i^{pi} realizations. Then we will have minimum and maximum values of neural reactions for each correctly recognized realization:

$$a_i = \min_{pi}\left\{net_i^{pi}\right\}; \; b_i = \max_{pi}\left\{net_i^{pi}\right\}, \; i = \overline{1,I}. \tag{23}$$

Let us determine the centre of gravity realizations coordinates which got into the range $[a_i; b_i]$; let us determine the centre of gravity recognized realizations in the sign space. Let us draw a hyper-sphere with r_i radius from the given centre in the sign space:

$$r_i = \frac{b_i - a_i}{2}, \qquad i = \overline{1,I}. \tag{24}$$

The indicated situation in two-dimensional sign space (on the plane) is shown in Fig. 1. Each point on the plane represents A_i pattern of realizations. The correctly recognized realizations are placed inside the r_i^1 radius circumference. It is possible to have two situations with P_i number: $Q_i = 0$ which means that $P_i = M_i$ that is all the realizations are recognized correctly and are error-free. Consequently, we decide that only one neuron is enough to recognize the pattern realization; $Q_i \neq 0$ which means that we have incorrectly recognized realizations beyond the hyper-sphere with $P_i \neq M_i$ radius. In such a case, for those realizations which were not included into r_i^1 hyper-sphere let us calculate the maximum and minimum values given by (22), calculate the radius of the hyper-sphere and draw a hyper-sphere with a new radius around it. We might have the following cases: The hyper-sphere comprises old and new realizations as shown on the circumference with r_i^1 radius (Fig. 1). In this case we should select a different centre of gravity and a different radius which is smaller than r_i^1 radius value. Let us restrict the new hyper-sphere; this process should be continued until we will have only correctly recognized realizations. Consequently, we make a decision that another neuron is necessary, the weighing coefficient of which equals to the coordinates of the new centre of gravity. If there still are realizations left outside the hyper-spheres, then we should continue the procedures given in point 1 until there are no more realizations of the given training set pattern left. By conducting the above mentioned procedures we will get the needed number of neurons per the realization of each recognizable training set pattern for each pattern and by summarizing these quantities we will get the total number of neurons for the whole set of recognizable patterns.

Let us assume that we have pattern descriptions: *E* set which coordinates are the weighing coefficients. For each pattern sets according to (23) have been calculated. The realizations of A_i pattern examination set from $\{X_i\}$ set should be presented to a_i and b_i from another pattern e.g. to A_j neuron (or neurons) for recognition in case when $z_i = a_i$

$$\{x_j\} \Longrightarrow Ne_i = W_i \Rightarrow \textstyle\sum_n x_{nj}\omega_{ni} \Rightarrow F_{ji}\left(\sum_n x_{nj}\omega_{ni}\right) \Longrightarrow z_i = a_i \Rightarrow out_{ji} = \{0; 1\} \tag{25}$$

where $j = \overline{1,J}$; $i = \overline{1,I}$; $i \neq j$.

According to (25) let us determine $\min \sum_n \omega_{nj}\omega_{ni} = a_{ji}$ values $A_j\, j = \overline{1,I},\ j \neq i$. For A_i and A_j patterns we may get two situations:

a) $F_{ji}(\cdot) \geq z_i \Rightarrow out_{ji} = 1 \Rightarrow X_j \in A_i$, (26)

b) $F_{ji}(\cdot) < z_i \Rightarrow out_{ji} = 0 \Rightarrow X_j \notin A_i$. (27)

It is evident that we have an error when the situation given by (26) is carried out. If (26) is not carried out for any of X_i set realizations, then we will decide that the number of neurons for A_i pattern is enough. If the situation given by (27) is accomplished for several realizations, then we will conduct the procedures described bellow for incorrectly recognized realizations and will determine the number of neurons necessary for A_i pattern in connection with A_j pattern. We will apply the same process for each pattern of $\{A\} \setminus A_i$ set and get the number of neurons necessary for A_i pattern for which (neurons) the weighing coefficients of the corresponding neurons together with *E* set elements will be determined in the bottom of the work by implementing the above-mentioned procedures. For all the *A* set patterns we will get the overall number of neurons for the set of recognized patterns. A procedure for determining the number of neurons in the layers based on clustering. Formal neuron's recognizing function – "scalar product" is used as similarity measure for clustering. On the first stage the number of neurons is determined for only one pattern. On the next stage the correction of the number of neurons considered other system descriptions (templates). Improving the realizations of patterns was obtained by perfecting the sign space and preparation procedures; In the process of changing the weighing coefficients we used only awarding procedures. The training process started with determining the value of weighing coefficients or random conferring. For conducting a training process we had the set of training combinations of any kind of represented conditions. In the process of training, a recognition procedure with scalar product similarity measure is used. By implementing the above mentioned procedures for all the *A* set patterns we will get the overall number of neurons for the set of recognized patterns.

ინფორმატიკა

ნეირონების რაოდენობის განსაზღვრა კლასტერირების მეთოდის გამოყენებით ამოცნობის პროცესისათვის

რ. ხუროძე*, ო.ვერულავა, მ. ჩხაიძე****

* *აკადემიის წევრი, საქართველოს ტექნიკური უნივერსიტეტი, თბილისი*
** *საქართველოს ტექნიკური უნივერსიტეტი, ინფორმატიკისა და მართვის სისტემების ფაკულტეტი, თბილისი*

როგორც ცნობილია, თანამედროვე ნეირონულ ქსელებში შრეებში ნეირონების რაოდენობა განისაზღვრება ექსპერიმენტების საფუძველზე, თეორიული დასაბუთების გარეშე, ევრისტიკული მოსაზრებებიდან გამომდინარე. წარმოდგენილ სტატიაში შრეებში ნეირონების რაოდენობის განსაზღვრის პროცედურა შემოთავაზებულია კლასტერირების საფუძველზე. ასევე, მსგავსების ზომად კლასტერირებისათვის გამოყენებულია ფორმალური ნეირონის ამომცნობი ფუნქცია "სკალარული პროდუქტი". კლასტერირება და ამოცნობა სრულდება მაღალი საიმედოობით. პირველ საფეხურზე ნეირონთა რაოდენობა განისაზღვრება მხოლოდ ერთი სახისათვის, შემდეგ ეტაპზე ნეირონების რაოდენობის შესწორებისას მხედველობაში მიიღება სხვა სახეთა აღწერები (ეტალონები).

REFERENCES

1. *C.A. Doukim, J.A. Dargham, A. Chekima* (2010), 10th International Conference on Information Sciences Signal Processing and their Applications (ISSPA) : 606-609, 10-13 May 2010, Kuala Lumpur, Malaysia.
2. *K. Shin-ike* (2010), Proc. of SICE Annual Conference: 238-242, 18-21 Aug. 2010. Taipei, Taiwan.
3. *Q. Y. Zhu, G. B. Huang, C. K. Siew* (2003), Proc. of 4th International Conference on Control and Automation, 2003. ICCA '03:496-50012 June 2003, Montreal, Canada.

Received June, 2014