Fuzzy Cognitive Map Learning Based on Nonlinear Hebbian Rule

Elpiniki Papageorgiou[1], Chrysostomos Stylios[2], and Peter Groumpos[1]

[1] Laboratory for Automation and Robotics
Department of Electrical and Computer Engineering
University of Patras, Rion 26500, Greece
Tel. +30 2610 997293, Fax. +30 26120 997309
{epapageo,groumpos}@ee.upatras.gr
[2] Computer Science Department, University of Ioannina
P.O. Box 1186, Ioannina, Greece
Tel. +30651298818
stylios@cs.uoi.gr

Abstract. Fuzzy Cognitive Map (FCM) is a soft computing technique for modeling systems. It combines synergistically the theories of neural networks and fuzzy logic. The methodology of developing FCMs is easily adaptable but relies on human experience and knowledge, and thus FCMs exhibit weaknesses and dependence on human experts. The critical dependence on the expert's opinion and knowledge, and the potential convergence to undesired steady states are deficiencies of FCMs. In order to overcome these deficiencies and improve the efficiency and robustness of FCM a possible solution is the utilization of learning methods. This research work proposes the utilization of the unsupervised Hebbian algorithm to nonlinear units for training FCMs. Using the proposed learning procedure, the FCM modifies its fuzzy causal web as causal patterns change and as experts update their causal knowledge.

Keywords: Unsupervised learning, Nonlinear Hebbian learning, fuzzy cognitive maps, neural networks, Hebbian rule.

1 Introduction

FCMs were proposed by Kosko to represent the causal relationship between concepts and analyze inference patterns [1,2]. FCMs represent knowledge in a symbolic manner and relate states, variables, events, outputs and inputs in a cause and effect approach. Comparing FCMs to either expert system or neural networks, they have several desirable qualities such as: FCM is relatively easy to represent structured knowledge, and the inference is computed by numeric matrix operation. FCMs are

T.D. Gedeon and L.C.C. Fung (Eds.): AI 2003, LNAI 2903, pp. 256-268, 2003.

appropriate to explicit the knowledge, which has been accumulated for years observing the operation and behavior of a complex system. Fuzzy Cognitive Maps have already been applied in many scientific areas, such as medicine, manufacturing, decision support systems, political science [3,4,5,6,7,8].

Till today, very few research efforts have been made to investigate and propose a learning algorithm suitable for FCMs [9,10]. This research work proposes a learning procedure based on the nonlinear Hebbian learning rule to improve the FCM structure. The introduction of FCM training eliminates the deficiencies in the usage of FCM and enhances the dynamical behavior and flexibility of the FCM model. A criterion function similar to that of the Hebbian rule for linear units is used and the proposed learning algorithm is used to train the FCM model of a process chemical control problem proving the efficiency of the algorithm.

The extension of the Hebbian learning rule suggesting nonlinear units drew the attention of research community [11,12,13] and it was applied in many problems [12,14]. There were showed that the use of units with nonlinear activation functions, which employ Hebbian learning, might lead to robust principal component analysis and also a nonlinear Hebbian learning rule by minimizing a given criterion function was proposed [15]. Furthermore, for a better understanding of the implementation of nonlinear units by exploring the statistical characteristics of the criterion function, i.e. how the operation of the nonlinear activation is being optimized and interpreted using a probability integral transformation you can refer to [16,17].

The outline of this paper follows. Section 2 describes the FCM modeling methodology, how a FCM is developed and how it models a system. Section 3 discusses nonlinear Hebbian learning algorithm. Section 4 introduces the Hebbian Learning Algorithm to nonlinear units for FCM and it presents the mathematical justification of the algorithm for FCMs and a methodology to implement this algorithm. In section 5, the proposed algorithm is implemented to train the FCM model of a process control problem and section 6 concludes the paper and discusses the usefulness of the new training methodology for FCMs.

2 Fuzzy Cognitive Maps Background and Description

Fuzzy cognitive maps have their roots in graph theory. Euler formulated the first graph theory in 1736 [18] and later on the directed graphs (digraphs) was used for studying structures of empirical world [19]. Signed digraphs were used to represent the assertions of information [20] and the term “cognitive map” described the graphed causal relationships among variables. The term “fuzzy cognitive map” was used for first time by Kosko [1] to describe a cognitive map model with two significant characteristics: (a) Causal relationships between nodes are fuzzified and (b) The system has dynamical involving feedback, where the effect of change in one node affects other nodes, which in turn can affect the node initiating the change.

The FCM structure is similar to a recurrent artificial neural network, where concepts are represented by neurons and causal relationships by weighted links connecting the neurons.

Concepts reflect attributes, characteristics, qualities and senses of the system. Interconnections among concepts of FCM signify the cause and effect relationship that a concept has on the others. These weighted interconnections represent the direction and degree with which concepts influence the value of the interconnected concepts. Figure 1 illustrates the graphical representation of a Fuzzy Cognitive Map.

The interconnection strength between two nodes C_j and C_i is w_{ji}, with w_{ji} taking on any value in the range -1 to 1.

There are three possible types of causal relationships among concepts:

- $w_{ji} > 0$ which indicates positive causality between concepts C_j and C_i. That is, the increase (decrease) in the value of C_j leads to the increase (decrease) on the value of C_i.
- $w_{ji} < 0$ which indicates negative causality between concepts C_j and C_i. That is, the increase (decrease) in the value of C_j leads to the decrease (increase) on the value of C_i.
- $w_{ji} = 0$ which indicates no relationship between C_j and C_i.

The directional influences are presented as all-or-none relationships, so the FCMs provide qualitative as well as quantitative information about these relationships [9]. Generally, the value of each concept is calculated, computing the influence of other concepts to the specific concept, [5], by applying the following calculation rule:

$$A_i^{(k+1)} = f(A_i^{(k)} + \sum_{\substack{j \neq i \\ j=1}}^{N} A_j^{(k)} \cdot w_{ji}) \quad (1)$$

where $A_i^{(k+1)}$ is the value of concept C_i at time $k+1$, $A_j^{(k)}$ is the value of concept C_j at time k, w_{ji} is the weight of the interconnection between concept C_j and concept C_i and f is the sigmoid threshold function.

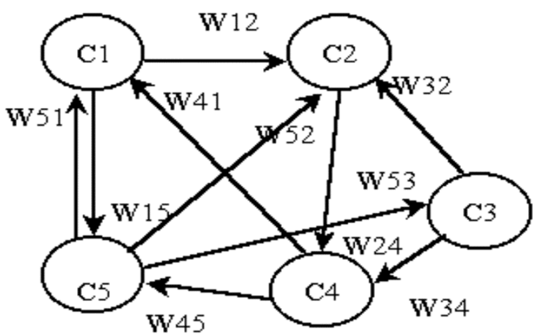

Fig.1. A simple Fuzzy Cognitive Map

The methodology for developing FCMs is based on a group of experts who are asked to define concepts and describe relationships among concepts and use IF-THEN rules to justify their cause and effect suggestions among concepts and infer a linguistic weight for each interconnection [8]. Every expert describes each interconnection with a fuzzy rule; the inference of the rule is a linguistic variable, which describes the relationship between the two concepts according to everyone expert and determines the grade of causality between the two concepts.

Then the inferred fuzzy weights that experts suggested, are aggregated and an overall linguistic weight is produced, which with the defuzzification method of Center of Area (CoA) [21,22], is transformed to a numerical weight w_{ji}, belonging to the interval [-1, 1] and representing the overall suggestion of experts. Thus an initial matrix $\mathbf{w}^{initial} = [w_{ji}]$, i,j=1,…,N, with $w_{ii} = 0$, i=1,..,N, is obtained.

The most significant weaknesses of the FCMs are their dependence on the expert's opinion and the uncontrollable convergence to undesired states. Learning procedures is a mean to increase the efficiency and robustness of FCMs, by modifying the FCM weight matrix.

3 Nonlinear Extension of Hebbian Rule

A weight-learning rule requires the definition and calculation of a criterion function (error function) and examining when the criterion function reaches a minimum error that corresponds to a set of weights of NN. When the error is zero or conveniently small then a steady state for the NN is reached; the weights of NN that correspond to steady-state define the learning process and the NN model [23].

According to the well-known Hebb's learning law, during the training session, the neural network receives as input many different excitations, or input patterns, and it arbitrarily organizes the patterns into categories. Hebb suggested the biological synaptic efficacies change in proportion to the correlation between the firing of the pre-and post-synaptic neuron [22,24]. Given random pre-synaptic input patterns **x**, weight vector **w**, and output $y = \mathbf{w}^T \mathbf{x}$, the criterion function J maximized by Hebb's rule may be written as:

$$J = E\{y^2\} \tag{2}$$

An additional constraint such as $\|\mathbf{w}\| = 1$ is necessary to stabilize the learning rule derived from eq. (2). A stochastic approximation solution based on (1) leads to the single neuron Hebbian rule [11]:

$$\Delta w_{ji} = \eta_k y_i (x_j - w_{ji} y_i) \tag{3}$$

where η_k is the learning rate at iteration k.

An extension of (3) to nonlinear units proposed with various possible nonlinear activation functions and their criterion functions [12]. Considering the case where the output of the linear unit is transformed using a nonlinear activation function (usually a sigmoid type function) with the criterion function in eq. (2). The following optimization problem is solving adjusting the nonlinear units' weights adaptively:

$$\begin{aligned}&\text{maximize} \quad J = E\{z^2\}\\&\text{subject to:} \quad \|\mathbf{w}\| = 1\end{aligned} \tag{4}$$

where $z = f(y)$, and f is a sigmoid function.

It is difficult to determine a closed form solution of eq. (4); therefore we employ a stochastic approximation approach, which leads to the following nonlinear Hebbian learning rule (the derivation is given in the [17])

$$\Delta w_{ji} = \eta_k z \frac{dz}{dy}(x_j - w_{ji} y_i) \tag{5}$$

The learning rule in eq. (5) is one of the three possible cases of nonlinear Hebbian rules discussed in Oja et al. [12,13]. If z is a linear function of y, say $z = y$, then the learning rule is reduced to the simple Hebbian rule for a linear unit.

For the case of the Hebbian learning rule in a linear unit, the objective function in eq. (2) has an obvious interpretation, i.e., projection of the input patterns in the direction of maximum variance, or maximum output value of neuron. On the other hand the same criterion function applied to nonlinear units, may lead to results radically different from those produced by linear units. Note that both linear and nonlinear learning rules are seeking a set of weight parameters such that the outputs of the unit have the largest variance. The nonlinear unit constraints the output to remain within a bounded range, e.g., $z = 1/(1 + \exp(-y))$ limits the output within [0,1]. The restriction of the nonlinear unit outputs significantly distinguishes nonlinear units from their linear counterparts and affects the mechanism of variance maximization.

4 Learning in FCMs

FCM learning involves updating the strengths of causal links so that FCM converge in a desired region. An advanced learning strategy is to modify FCMs by fine-tuning its initial causal link or edge strengths through training algorithms similar to that in artificial neural networks.

There are just a few FCM learning algorithms [9,10], which are based on artificial neural networks training. Kosko proposed the Differential Hebbian, a form of unsupervised learning, but without mathematical formulation and implementation in real problems [9]. There is no guarantee that DHL will encode the sequence into the FCM and till today no concrete procedures exist for applying DHL in FCMs. Another algorithm is named Adaptive Random for FCMs based on the definition of Random Neural Networks [10]. Recently, a different approach has been proposed for FCM learning based on evolutionary computation [25], where evolution strategies have been used for the computation of the desired system's configuration. This technique is exactly the same used in neural networks training; it doesn't take into consideration the initial structure and experts' knowledge for the FCM model, but uses data sets determining input and output concepts in order to define the cause-effect relationships satisfied the fitness function. The calculated weights appear large deviations from the actual FCM weights and in real problems they have not the accepted physical magnitude. So, a formal methodology and general learning algorithm suitable for FCM learning and for practical applications is still needed.

4.1 The Proposed Approach Based on Nonlinear Hebbian Learning (NHL)

The proposed learning algorithm is based on the premise that all the concepts in FCM model are triggering at each iteration step and change their values. During this triggering process the weight w_{ji} of the causal interconnection of the related concepts is updated and the modified weight $w_{ji}^{(k)}$ is derived for iteration step k.

The value $A_i^{(k+1)}$ of concept C_i, at iteration $k+1$, is calculated, computing the influence of interconnected concepts with values A_j to the specific concept C_i due to modified weights $w_{ji}^{(k)}$ at iteration k, through the following equation:

$$A_i^{(k+1)} = f(A_i^{(k)} + \sum_{\substack{j \neq i \\ j=1}}^{N} A_j^{(k)} \cdot w_{ji}^{(k)}) \tag{6}$$

Furthermore, some of concepts are defined as output concepts (OCs). These concepts stand for the factors and characteristics of the system that interest us, and we want to estimate their values, which represent the final state of the system. The distinction of FCM concepts as inputs or outputs is determined by the group of experts for each specific problem. Any of the concepts of the FCM model may be inputs or outputs. However, experts select the output concepts and they consider the rest as initial stimulators of the system. The learning algorithm that extracts hidden and valuable knowledge of experts can increase the effectiveness of FCMs and their implementation in real problems.

Taking the advantage of the general nonlinear Hebbian-type learning rule for NNs, [12], we introduce the mathematical formalism incorporating this learning rule for FCMs, a learning rate parameter and the determination of input and output concepts. This algorithm relates the values of concepts and values of weights in the FCM model.

The proposed rule has the general mathematical form:

$$\Delta w_{ji} = \eta_k A_j (A_i - A_j w_{ji}) \tag{7}$$

where the coefficient η_k is a very small positive scalar factor called learning parameter. The coefficient has determined using experimental trial and error method that optimizes the final solution.

This simple rule states that if $A_i^{(k)}$ is the value of concept C_i at iteration k, and A_j is the value of the triggering concept C_j which triggers the concept C_i, the corresponding weight w_{ji} from concept C_j towards the concept C_i is increased proportional to their product multiplied with the learning rate parameter minus the weight decay at iteration step k.

The training weight algorithm takes the following form:

$$w_{ji}^{(k)} = w_{ji}^{(k-1)} + \eta_k A_j (A_i^{(k)} - A_j w_{ji}^{(k-1)}) \tag{8}$$

At every simulation step the value of each concept of FCM is updated, using the equation (6) where the value of weight $w_{ji}^{(k)}$ is calculated with equation (8).

Also, we introduce two criteria functions for the proposed algorithm. One criterion is the maximization of the objective function J, which has been defined by Hebb's rule in equation (4). The objective function J has been proposed for the NHL, examining the desired values of output concepts (OCs), which are the values of the activation concepts we are interested about. The J is defined as:

$$J = \sum_{i=1}^{l} (OC_i)^2 \tag{9}$$

where l is the number of OCs.

The second criterion is the minimization of the variation of two subsequent values of OCs, represented in equation:

$$\left| OC_j^{(k+1)} - OC_j^{(k)} \right| < e \tag{10}$$

These criteria determine when the iterative process of the learning algorithm terminates. The term e is a tolerance level keeping the variation of values of OC(s) as low as possible and it is proposed equal to $e = 0.001$.

Through this process and when the termination conditions are met, the final weight matrix $\mathbf{W}^{updated}$ is derived.

4.2 The Proposed Learning Algorithmic Process

The schematic representation of the proposed NHL algorithm is given in Figure 2. Considering an *n*-node FCM-model, the execution phase of the proposed algorithm is consisted of the following steps:

Step 1: Read input state $\mathbf{A}^0$ and initial weight matrix $\mathbf{w}^0$
Step 2: Repeat for each iteration step k

2.1: Calculate A_i according to equation (6)

2.2: Update $w_{ji}^{(k)}$ according to equation (8)

2.3: Calculate the two criterion functions

Step 3: Until the termination conditions are met.
Step 4: Return the final weights $w^{updated}$ and concept values in convergence region.

All the FCM concepts are triggering at the same iteration step and their values are updated due to this triggering process.

5 Implementation to a Process Control Problem

In this section the proposed Nonlinear Hebbian Learning (NHL) is implemented to modify the FCM model of a simple process control problem [8].

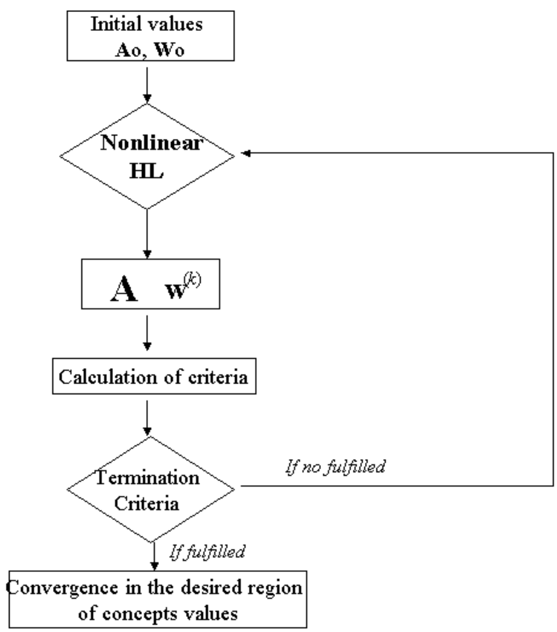

Fig. 2. Schematic representation of NHL training algorithm

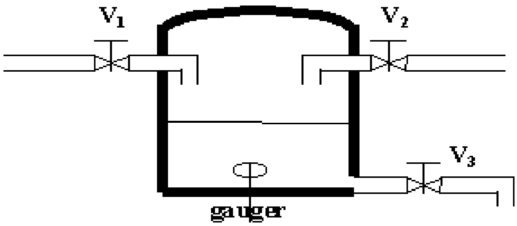

Fig. 3. The illustration for simple process example

At this process problem there is one tank and three valves that influence the amount of liquid in the tank; figure 3 shows an illustration of the system. Valve 1 and valve 2 empty two different kinds of liquid into tank1 and during the mixing of the two liquids a chemical reaction takes place into the tank. A sensor measures the specific gravity of the produced liquid into tank. When value of specific gravity is in the range between (G_{max}) and (G_{min}), this means that the desired liquid has been produced in tank. Moreover, there is a limit on the height of liquid in tank, which cannot exceed an upper limit (H_{max}) and a lower limit (H_{min}). So the control target is to keep these variables in the following range of values:

$$0.74 \leq G \leq 0.80$$

$$0.68 \leq H \leq 0.70$$

A FCM model for this system is developed and depicted on figure 4. Three experts constructed the FCM, they jointly determined the concepts of the FCM and then each expert drawn the interconnections among concepts and assigned fuzzy weight for each interconnection [5].
The FCM is consisted of five concepts:

Concept 1 – the amount of the liquid that Tank 1 contains is depended on the operational state of Valves 1, 2 and 3;
Concept 2 – the state of Valve 1 (it may be closed, open or partially opened);
Concept 3 - the state of Valve 2 (it may be closed, open or partially opened);
Concept 4 - the state of Valve 3 (it may be closed, open or partially opened);
Concept 5 –the specific gravity of the liquid into the tank.

For this specific control problem, experts have determined the initial values of concepts and weights, and which concepts are the desired output concepts (OCs). For this problem the desired output concepts are the concepts C_1 and C_5.

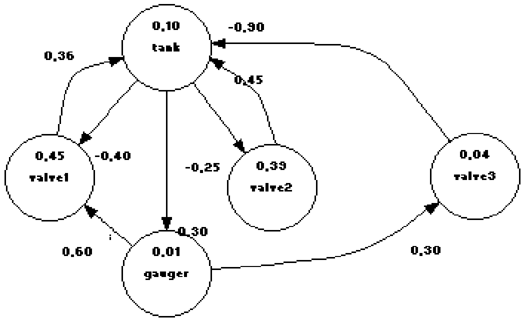

Fig. 4. The FCM model of the chemical process

Experts suggested the appropriate initial weights of the FCM model that are shown in the following weight matrix:

$$\mathbf{W}^{initial} = \begin{bmatrix} 0 & -0.4 & -0.25 & 0 & 0.3 \\ 0.36 & 0 & 0 & 0 & 0 \\ 0.45 & 0 & 0 & 0 & 0 \\ -0.9 & 0 & 0 & 0 & 0 \\ 0 & 0.6 & 0 & 0.3 & 0 \end{bmatrix}$$

Now the NHL algorithm can be applied to train the FCM and modify the weights. The training process starts by applying the initial values of concepts $\mathbf{A}^{0}_{first}$ =[0.4 0.708 0.612 0.717 0.3] and weights $\mathbf{W}^{initial}$. The suggested value of learning rate η, at equation (8), after trial and error experiments, is determined as 0.01. Notably, at each iteration step, all weights are updated using the equation (6).

The algorithmic procedure continues, until the synchronously satisfaction of the two termination criteria are met. The result of training the FCM is a new connection weight matrix $\mathbf{W}$ that maximizes the objective function J and satisfy synchronously the second criterion. This algorithm iteratively updates the connection weights based on the equation (8), and equation (6) calculates the activation values of concepts based on the described asynchronous updating mode.

The convergent state is reached, after 16 recursive cycles, and it is $\mathbf{A}$ =[0.692 0.738 0.676 0.747 0.743]. The updated weight matrix after 16 cycles, is:

$$\mathbf{W}^{updated} = \begin{bmatrix} 0 & -0.207 & -0.112 & 0.064 & 0.264 \\ 0.298 & 0 & 0.061 & 0.069 & 0.067 \\ 0.356 & 0.062 & 0 & 0.063 & 0.061 \\ -0.516 & 0.070 & 0.063 & 0 & 0.068 \\ 0.064 & 0.468 & 0.060 & 0.268 & 0 \end{bmatrix}$$

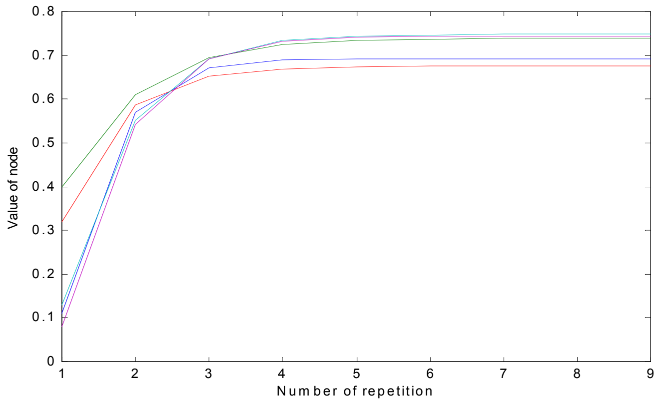

Fig. 5. Variation values of concepts for 9 simulation steps

These new values for weights describe new relationships among the concepts of FCM. It is noticeable that the initial zero weights no more exist, and new interconnections with new weights have been assigned, only diagonal values remain equal to zero. This means that all concepts affect the related concepts, and the weighed arcs show the degree of this relation.

5.1 Evaluation of the Modified Weight Matrix

Let's make now a testing using a random initial vector $\mathbf{A}^{0}_{random}$, and with the previously derived weight matrix, $\mathbf{w}^{updated}$ as initial. The randomly produced initial values are

$\mathbf{A}^{0}_{random}$ =[0.1 0.45 0.37 0.04 0.01]. Applying the NHL algorithm, it stops after 9 simulation steps and the derived concept vector is A_{random} :

$\mathbf{A}_{random}$ =[0.693 0.738 0.676 0.747 0.744], as shown in Fig. 5. This new state vector has the same values as the previous state vector $\mathbf{A}$, driving the FCM in the same convergence-desired region.

Also, we have tested this FCM model for 1000 test cases with random initial values of concepts, using the weight matrix $\mathbf{w}^{updated}$ and we end up at the same result for concepts values. So, if we use the FCM model of the process control problem with the modified values of weights for any initial values of concepts this model is driving in the desired region of concept values.

The NHL rule, which represented with equation (8), is suitable for updating the weights of FCMs with the same manner as in neural networks. Here the output concept is the one that have been fired by its interconnected concepts and it is initially defined for each specific problem. This output concept is the learning rate signal in the unsupervised learning procedure and due to the feedback process fires sequentially the other concepts and updates their cause-effect interconnections of FCM through the previous described Hebbian-type learning rule.

6 Conclusions and Future Directions

In this paper, the unsupervised learning method (NHL) based on nonlinear Hebbian-type learning rule is introduced to adapt the cause-effect relationships among concepts of FCM and to eliminate the deficiencies that appear in operation of FCMs. In this way, we take into account the dynamic characteristics of the learning process and the environment.

The unsupervised Hebbian rule is introduced to train the FCM and accompanied with the good knowledge of a given system or process can contribute towards the establishment of FCM as a robust technique. Also a formal methodology suitable for practical application has been developed and desired convergence regions for FCM process control have been produced.

In future work, further developments of the learning algorithm will be examined and the implementation of this learning approach on more complex problems will be investigated.

References

[1] Kosko, B.: Fuzzy Cognitive Maps, Int. J. Man-Machine Studies, Vol. 24 (1986), 65-75.

[2] Kosko, B.: Neural Networks and Fuzzy Systems, Prentice-Hall, New Jersey, (1992).

[3] Papageorgiou, E., Stylios, C.D., Groumpos, P.P.: Decision Making in external beam radiation therapy based on FCMs, Proceedings of the 1st International IEEE Symposium on Intelligent Systems, IS 2002, Varna, Boulgaria, 10-12 September, (2002), CD-ROM

[4] Papageorgiou, E.I., Stylios, C.D., Groumpos, P.P.: An Integrated Two-Level Hierarchical Decision Making System based on Fuzzy Cognitive Maps, IEEE Transactions on Biomedical Engineering, vol. 50, no. 12, (December 2003).

[5] Stylios, C.D., Groumpos, P.P.: Fuzzy Cognitive Maps in Modelling Supervisory Control Systems, Journal of Intelligent & Fuzzy Systems, vol. 8, (2000), 83-98.

[6] Khan, S., Chong, A., and Gedeon, T.A.: Methodology for Developing Adaptive Fuzzy Cognitive Maps for Decision Support, *Proc. 3rd Australia-Japan Evolutionary Systems,* Canberra 23-25 November, (1999), 93-100.

[7] Papageorgiou, E.I., Stylios, C.D., Groumpos, P.P.: Activation Hebbian Learning Rule for Fuzzy Cognitive Map Learning, *Proc. of 15th IFAC World Congress of Inter. Federation of Automatic Control,* Barcelona, Spain, July 21-26,CD-ROM, 2002.

[8] Stylios, C.D., Groumpos, P.P., Georgopoulos, V.C.: An Fuzzy Cognitive Maps Approach to Process Control Systems, *Journal of Advanced Computational Intelligence*, vol. 3, No. 5, 1999, pp. 409-417.
[9] Kosko, B.: Fuzzy Engineering, Prentice-Hall, New Jersey, (1997).
[10] Aguilar, J.: Adaptive Random Fuzzy Cognitive Maps. In: Garijio, F.J., Riquelme, J.C., Toro, M. (eds.): IBERAMIA 2002, Lecture Notes in Artificial Intelligence 2527, Springer-Verlag Berlin Heidelberg (2002), 402-410.
[11] Oja, E.: A simplified neuron model as a principal component analyzer. Journal of Mathematical Biology, Vol. 16, (1982), 267-273
[12] Oja, E., Ogawa, H., Wangviwattana, J.: Learning in nonlinear constrained Hebbian networks. In. T. Kohonen et al.(eds.): Artificial Neural Networks (1991), Amsterdam: North-Holland, 385-390.
[13] Oja, E.: Neural networks, principal components and subspaces. International Journal of Neural Systems, Vol. 1, (1989), 61-68
[14] Lee, W.C., Oslhausen, B.A.: A nonlinear Hebbian network that learns to detect disparity in random-dot stereograms. Neural Computation, Vol. 8, (1996), 545-566.
[15] Oja, E., Karhunen, J.: Nonlinear PCA, algorithm and applications. Report A18, Finland: Helsinski University of Technology (1993).
[16] Hassoun, M.: *Fundamentals of Artificial Neural Networks,* MIT Press, Bradford Book, Massachusetts, (1995).
[17] Sudjianto, A., Hassoun, M.: Statistical basis of nonlinear hebbian learning and application to clustering. Neural Netwoks, Vol. 8, No. 5, (1995), 707-715.
[18] Biggs, N.L., Llooyd, E.K., Wilson, R.J.: Graph Theory 1736-1936. Clarendon Press, Oxford, (1976).
[19] Harary, F., Norman, R.J., Cartwright, D.: Structural models: an introduction to the theory of directed graphs. John Wiley & Sons, New York, (1965), pp.201.
[20] Axelrod, R.: Structure of Decision, the cognitive maps of political elites. Princeton University Prass, Princeton, New Jersey, (1976), pp. 404
[21] Lin, C.T., Lee, C.S. G: *Neural Fuzzy Systems: A Neuro-Fuzzy Synergism to Intelligent Systems*, Prentice Hall, Upper Saddle River, N.J, (1996).
[22] Jang, J. S. R., Sun, C. T., Mizutani, E.: *Neuro-Fuzzy & Soft Computing.* Prentice-Hall, Upper Saddle River, (1997).
[23] Haykin, S.: Neural Networks: A Comprehensive Foundation (2^{nd} ed.), Macmillan, (1994).
[24] Zurada, J.: Introduction to Artificial Neural Systems. West Publishing Company, (1992)
[25] Koulouriotis, D.E., Diakoulakis, I.E., Emiris, D.M.: Learning Fuzzy Cognitive Maps using evolution strategies: A novel schema for modeling a simulating high-level behavior. *Pro. 2001 IEEE Congress on Evolutionary Computation,* Seoul, Korea, 2001, vol. 1, pp. 364-371.