Available online at www.sciencedirect.com
SCIENCE DIRECT®
ELSEVIER International Journal of Approximate Reasoning 37 (2004) 219–249
INTERNATIONAL JOURNAL OF APPROXIMATE REASONING
www.elsevier.com/locate/ijar

Active Hebbian learning algorithm to train fuzzy cognitive maps

E.I. Papageorgiou [a,*], C.D. Stylios [b,1], P.P. Groumpos [a]

[a] *Laboratory for Automation and Robotic, Department of Electrical and Computer Engineering, University of Patras, Rion 26500, Greece*

[b] *Department of Computer Science, University of Ioannina, P.O. Box 1186, Ioannina, Greece*

Received 1 August 2003; accepted 1 January 2004
Available online 28 April 2004

Abstract

Fuzzy cognitive map is a soft computing technique for modeling systems, which combines synergistically the theories of neural networks and fuzzy logic. Developing of fuzzy cognitive map (FCM) relies on human experience and knowledge, but still exhibits weaknesses in utilization of learning methods. The critical dependence on experts and the potential uncontrollable convergence to undesired steady-states are important deficiencies to manage FCMs. Overcoming these deficiencies will improve the efficiency and robustness of the FCM methodology. Learning and convergence algorithms constitute the mean to improve these characteristics of FCMs, by modifying the values of cause–effect weights among concepts. In this paper a new learning algorithm that alleviates the problem of the potential convergence to a steady-state, named Active Hebbian Learning (AHL) is presented, validated and implemented. This proposed learning procedure is a promising approach for exploiting experts' involvement with their subjective reasoning and at the same time improving the effectiveness of the FCM operation mode and thus it broadens the applicability of FCMs modeling for complex systems.

Keywords: Fuzzy cognitive maps; Unsupervised learning; Hebbian learning; Artificial neural networks

* Corresponding author. Tel.: +302-610-997293; fax: +302-610-997309.

E-mail addresses: epapageo@ee.upatras.gr (E.I. Papageorgiou), stylios@cs.uoi.gr (C.D. Stylios), groumpos@ee.upatras.gr (P.P. Groumpos).

[1] Tel.: +302-651-098818.

0888-613X/$ - see front matter
doi:10.1016/j.ijar.2004.01.001

1. Introduction

Fuzzy cognitive map (FCM) is a soft computing technique, which is capable of dealing with complex systems in situations exactly as a human does using a reasoning process that can include uncertain and ambiguity descriptions. FCM is a promising modeling method for describing particular domains showing the concepts (variables) and the relationships between them while it encompasses advantageous features. The most pronounced features for FCMs are the flexibility in system design, model and control; the comprehensive operation and the abstractive representation of behavior for complex systems. These advantageous modeling features of FCMs, encourage us to investigate their structure, attempting to broaden the FCM functionality and applicability in any problem and system.

FCMs were introduced by Kosko to represent the causal relationship between concepts and analyse inference patterns [1,2]. FCMs represent knowledge in a symbolic manner and they model systems and behaviour relating states, variables, processes, events, values and inputs according to their cause effect relationship. Compared to either expert systems or neural networks, FCMs have several desirable properties such as: FCMs are relatively easily used for representing structured knowledge, and permitting feedback relationships and/or hidden interrelationships. For their inference FCMs utilize IF/THEN rules but finally their inference is computed by simple numerical matrix operation [3].

FCMs are appropriate to represent the knowledge and experience which has been accumulated on the operation of a system. FCM are built by experts with an interactive procedure of knowledge acquisition [4]. FCMs have gained considerable research interest and have been applied in different scientific areas such as biomedical engineering [5–7], manufacturing and supervisory systems [8,10], organization behavior [9], political science [11], decision making for geographic information systems [12,13]. However, FCMs had initially some deficiencies that sometimes restricted their application, they are also inefficient in adapting experts' knowledge via optimization and learning techniques which are crucial in many applications. Thus the FCM methodology needs enhancement, and improvement by eliminating its weaknesses such as the dependence on experts' intervention for the FCM design. Following this direction, learning algorithms have been investigated in order to improve the FCM capabilities [14–19].

This research work proposes an advanced learning algorithm based on the unsupervised Hebbian learning rule to improve the FCM structure, to eliminate the deficiencies in the usage of FCM and to enhance the flexibility and dynamical behavior of the FCM model. The proposed learning algorithm is a modified version of the general unsupervised Hebbian learning algorithm for neural networks [20]. We called this algorithm "Active Hebbian Learning" (AHL), because it introduces the determination of the sequence of activation concepts. Furthermore it is based on the well-known Hebbian principles for neural adaptation. This training algorithm increases the robustness of FCM, and it is accompanied with the utilization of the acquired knowledge of the given system. The updated FCM structure, after training with AHL, guarantees the successful implementation of the proposed modeling procedure for real case problems.

The outline of this paper is as follows. Section 2 presents an overview of FCM modeling technique, how FCMs are developed and how they model a system. Section 3 discusses different learning methods of neural networks concentrating in unsupervised Hebbian learning methods and especially presenting the requirements of training for FCMs. Section 4 introduces the Active Hebbian Learning algorithm and it presents the mathematical formulation and justification of the algorithm for FCMs. Section 5 describes a methodology how to implement the AHL algorithm for training FCMs. In Section 6, the proposed AHL algorithm is applied to train the FCM model of a process control problem, while conclusions are provided in Section 7.

2. Fuzzy cognitive maps

Axelord [11] for the first time used cognitive maps as a formal way to represent social scientific knowledge and to model decision-making in social and political systems. Then, Kosko [1,2] introduced fuzzy cognitive maps. An FCM illustrates the model of a system using a graph of concepts and showing the cause and effect among concepts (Fig. 1). An FCM describes the behavior of a system in terms of concepts; each concept represents a state, variable or a characteristic of the system. Values of concepts (nodes) change over time, and take values in the interval $[0, 1]$. The causal links between nodes are represented by directed weighted edges that illustrate how much one concept influences the interconnected concepts, and the causal weights of the interconnections belong in the interval $[-1, 1]$.

FCM graphical illustration reflects which concept influences other concepts, showing the interconnections between concepts and facilitates suggestions in

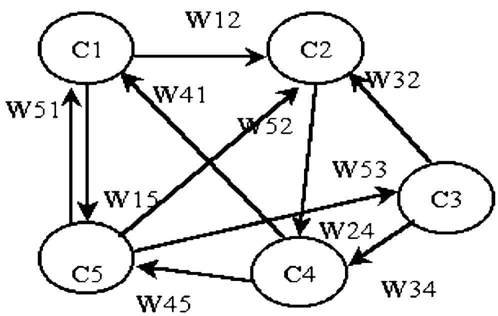

Fig. 1. A simple fuzzy cognitive map.

the reconstruction of the FCM, as the adding or deleting of an interconnection or a concept [21]. If the sign of the weight indicates positive causality $w_{ij} > 0$ between concept C_i and concept C_j, then an increase in the value of concept C_i will cause an increase in the value of concept C_j and a decrease in the value of concept C_i will cause a decrease in the value of concept C_j. When there is negative causality between two concepts, $w_{ij} < 0$, then an increase in the value of the first concept C_i causes a decrease in the value of the second concept and a decrease of concept C_i causes an increase in value of C_j. When there is no relationship between two concepts, then $w_{ij} = 0$. The strength of the weight w_{ij} indicates the degree of influence between concept C_i and concept C_j.

Generally, the value of each concept at every simulation step is calculated, computing the influence of the interconnected concepts to the specific concept, [8], by applying the following calculation rule:

$$A_i(t) = f\left(A_i(t-1) + \sum_{\substack{j=1 \\ j\neq 1}}^{n} A_j(t-1) \cdot w_{ji} \right) \tag{1}$$

where $A_i(t)$ is the value of concept C_i at time t, $A_j(t-1)$ is the value of concept C_j at time $t-1$, w_{ji} is the weight of the interconnection from concept C_j to concept C_i and f is the sigmoid function.

The sigmoid function, f, belongs to the family of squeezing functions, and usually the following function is used:

$$f(x) = \frac{1}{1+\exp(-\lambda x)} \tag{2}$$

This is the unipolar sigmoid function, where $\lambda > 0$ determines the steepness of the continuous function $f(x)$.

All the values of concepts and weights on the FCM have fuzzy nature representing issues, states and variables using linguistic notion. These fuzzy variables need to be defuzzified in order to use mathematical functions and calculate the corresponding results. Thus, values of concepts belong to the interval $[0,1]$ and values of weights to the interval $[-1,1]$. Using the sigmoid function the calculated values of concepts after each simulation step will belong to the interval $[0,1]$ where concepts take values.

Experts design and develop the fuzzy graph structure of the FCM, consisting of concept-nodes that represent the key principles-factors-functions of the system operation. Then, they determine the structure and the weighted interconnections of the FCM using fuzzy conditional statements [22]. More specifically, experts are asked to describe the relationships between concepts and they use IF–THEN rules to justify their cause and effect suggestions among concepts, inferring a linguistic weight for each interconnection. 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 and determines the grade of causality between the two concepts. Then the inferred fuzzy weights, are aggregated, as they are suggested by experts, and an overall linguistic weight is produced, which with the defuzzification method of Center of Area (CoA) [23], 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 $w^{\text{initial}} = [w_{ji}]$, $i, j = 1, \ldots, N$, with $w_{ii} = 0$, $i = 1, \ldots, N$, is obtained, that gathers the weights of all the interconnections of the FCM.

3. Learning methods

Learning in Artificial Neural Networks (ANNs), is the process of searching a multidimensional parameter space for a Neural Network (NN) state that optimizes a predefined criterion function J. Learning algorithms can be supervised, reinforcement or unsupervised and the function J is commonly referred to an error function, or a cost function, or an objective function. In fact, most of the learning algorithms have well-defined analytical criterion functions [24–26]. Learning rules implement local search technique (gradient descent) to obtain weight vector solutions, which optimize the associated criterion function. Therefore, it is the criterion function that mainly determines the implementation of the learning rule; there are different minimization methods that have been used for training feed forward neural networks (NNs) [27].

A learning algorithm is a mathematical method that determines the weights and outlines the convergence for an ANN to reach the steady state of its parameters successfully. Typically one starts with an error function, which is expressed in terms of the weights and the values of the output nodes. The learning objective is to reach a minimum error that corresponds to a set of weights for the NN. When the error is zero or conveniently small, then the steady state for the NN and the weights is reached. The steady-state weights define the learning process and the ANN model [28]. Proofs of convergence for these algorithms, and presentation of the different methods can be found in [29–31].

The general weight-learning rule, which is shown in Fig. 2, has the following general form:

$$\Delta \mathbf{w}_i = \rho \cdot r(\mathbf{w}_i, \mathbf{x}) \cdot \mathbf{x} \tag{3}$$

where ρ is a positive number called the learning constant which determines the rate of learning, r is the learning signal which is in general a function of $\mathbf{w}_i$ and $\mathbf{x}$ is the input signal. This learning form indicates that the increment of the weight vector $\mathbf{w}_i$ (the weights of all the other nodes towards node i that means

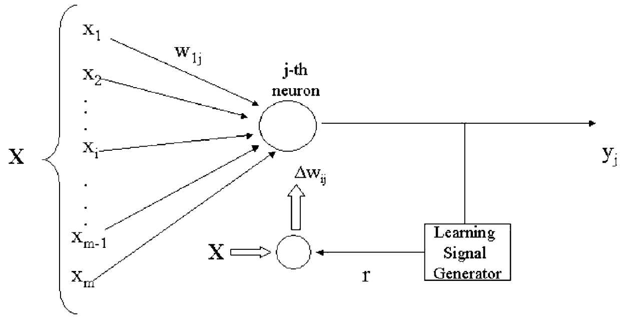

Fig. 2. The general unsupervised weight-learning rule for neural networks.

the ith column of the weight matrix **w**) is proportional to the product of the learning signal **r** and the input **x**.

One of the main algorithms used in unsupervised learning is the Hebbian learning algorithm [30]. The simplest networks consist of a set of input vectors **x** and outputs **y** connected by a weight matrix, **w**, where w_{ij} connects x_i to y_j. Then the problem in unsupervised learning is to find the values of the weights, **w**, which will minimize the error function. During the training session, the NN receives as input many different excitations, or input patterns, and it arbitrarily organizes the patterns into categories. The Hebb's learning law is usually implemented by

$$w_{ij}(k+1) = w_{ij}(k) + \rho \cdot y_j \cdot x_i \tag{4}$$

The weight-learning rule requires the definition and calculation of a criterion function. The objective is the criterion function to reach a minimum error that corresponds to a set of weights of NN. The steady-state weights define the learning process and the NN model. Thus, the minimization of an objective function is the ultimate goal [32–34].

3.1. Learning methods for FCMs

Learning of FCM involves updating the strengths of causal links. A learning strategy is to improve FCMs by fine-tuning its initial causal link or edge strengths applying training algorithms similar to that of artificial neural networks. Up-to-date, there have been proposed some FCM learning algorithms [2,15–17,19].

Kosko has initially proposed the Differential Hebbian Learning (DHL), as a form of unsupervised learning, but without any mathematical formulation and implementation in real problems [1,2]. The Balanced differential learning algorithm for FCM training, based exactly on the DHL, has also investigated [14]. This algorithm is a modified version of the DHL and seems to do better in learning patterns and in modeling a given domain than the classical approach. But till today no concrete procedure exist for applying DHL and balanced differential learning algorithm in FCMs. Another proposed approach for FCMs training is the Adaptive Random FCMs based on the theoretical aspects of Random Neural Networks [16]. This algorithm starts from an initial state and an initial weight matrix of the FCM and adapt the weights in order to compute a weight matrix that leads the FCM to a desired steady-state.

Recently, another unsupervised learning algorithm, the Nonlinear Hebbian Learning (NHL), have been investigated to train FCMs [17]. This algorithm is based on the nonlinear Hebbian-type learning rule and updates only the initially suggested (non-zero) weights of the FCM. These weights are updating synchronously at each iteration step till the termination of the algorithm. The calculated values of weights keep their initial signs and directions, as suggested by experts. All the other weights remain zero and no new interconnections are assigned.

In addition to the unsupervised learning-based techniques for FCMs, methods based on Evolutionary Computation techniques have been investigated. Particle Swarm Optimization (PSO) method has been proposed and used for first time for FCM learning giving very promising results [18]. PSO algorithms are a part of swarm intelligence, which is a rapidly growing area of artificial intelligence. This method provides a search procedure, which optimizes a problem-depended fitness function $\varphi()$, by maintaining and evolving a swarm of candidate solutions. Using this learning approach a number of appropriate weight matrices can be derived leading the system to desired convergence regions. Furthermore, Evolution Strategies have been used for the computation of the desired output concepts' values and system's configuration [19]. Exactly the same technique has been used in neural networks training; it does not 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. More investigation is needed for the evolutionary computation methods.

After this discussion it is clear that a formal methodology and a learning algorithm suitable for FCMs convergence has not yet proposed and accepted widely.

4. The Active Hebbian Learning algorithm

Here, an unsupervised learning algorithm to train FCMs is proposed and developed, namely Active Hebbian Learning (AHL) algorithm, which introduces the determination of the sequence of activation concepts. When the experts develop the FCM they determine the sequence of activation, the steps of activation and the cycle of simulation. At every simulation step, one (or more) concept(s) becomes Activation concept that triggers the other interconnecting concepts, which in turn, at the next simulation step, become Activation concept. When all the concepts have become Activation concepts, according to the sequence of activation the simulation cycle has closed and a new one starts.

A simulation cycle is consisted of steps, at each simulation step one or more concepts are the Activation concepts that influence the interconnected concepts and so on till the termination of the sequence of activation that close the cycle. This concept, at the next iteration step, becomes Activated concept. For example, let us say the *j*th concept C_j, is the triggering concept that influences concept C_i, as shown in Fig. 3. The concept C_j is declared the Activation concept, with the value A_j^{act} and it triggers the interconnected corresponding concept C_i, which is the Activated concept. At next iteration step, the concept C_i influence the other interconnected concepts C_l and so on. It is assumed that there is asynchronous stimulation mode due to which the concept C_i is becoming the Activation concept that triggers C_l and the other interconnected concepts and there is a sequence of activation steps. During every activation step the weight w_{ji} of the causal interconnection of the related concepts is updated and the modified weight $w_{ji}^{(k)}$ is derived for each iteration step k.

In addition to the determination of sequence of activation concepts; a limited number of concepts are selected (by experts) as outputs for each specific problem and these concepts are defined as the Activation Decision Concepts

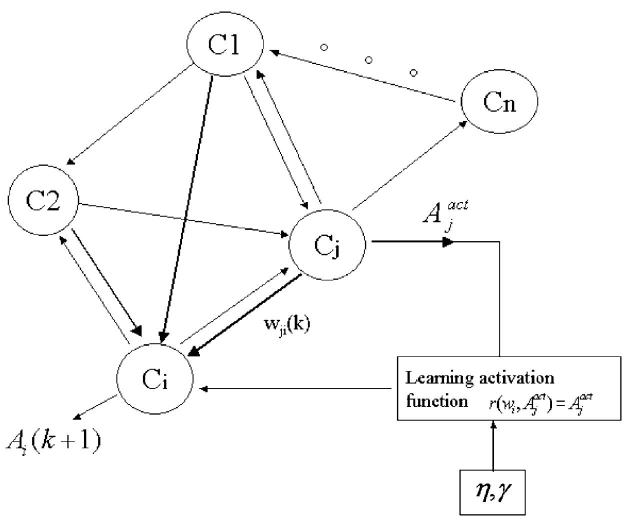

Fig. 3. The proposed activation weight-learning process for FCMs.

(ADCs). These concepts are in the center of interest; they stand for the main factors and characteristics of the system, known as outputs, whose values we want to estimate and they represent the final state of the system.

The distinction of FCM concepts as inputs, intermediate or outputs depends on the modeled system and the focus of experts. In general, all the concepts of FCM for the given system may be inputs which take their values externally, intermediates which are influenced by other concepts and influence the outputs concepts. However, the training phase is conducted in selecting a limited number of concepts as outputs (those we want to estimate their values). The expert's intervention is the only way to address this definition. The design of this learning algorithm extracts the valuable knowledge of experts and can increase the operation of FCMs and implementation in real case problems just by analyzing previous information and experts' knowledge about the given systems.

Fig. 3 illustrates an FCM-model, consisting of n-nodes, with the following parameters: C_i is the ith concept with value $A_i(k)$, $1 \leqslant i \leqslant n$; w_{ji} is the weight describing the influence from C_j to C_i, and its value is modified using the AHL rule; $A_j^{\mathrm{act}}(k)$ is the activation value of concept C_j, which triggers the interconnected concepts, behaving as Activation concept; γ is the weight decay parameter; η is the learning rate parameter, depending on simulation cycle c and $A_i(k)$ is the value of Activated concept C_i.

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

$$A_i(k+1) = f\left(A_i(k) + \sum_l A_l^{\mathrm{act}}(k) \cdot w_{li}(k)\right) \tag{5}$$

where A_l are the values of Activation concepts C_l that influence the concept C_i, and $w_{li}(k)$ are the corresponding weights that describe the influence from C_l to C_i. For example, in Fig. 3, the l takes the numbers 1, 2 and j, and A_1, A_2 and A_j are the values of Activation concepts C_1, C_2 and C_j, respectively, that influence C_i in this simulation step. Thus the value of Activated concept C_i is calculated using the following equation:

$$\begin{aligned} A_i(k+1) = f(&A_i(k) + A_1^{\mathrm{act}}(k) \cdot w_{1i}(k) + A_2^{\mathrm{act}}(k) \cdot w_{2i}(k) \\ &+ A_j^{\mathrm{act}}(k) \cdot w_{ji}(k)) \end{aligned} \tag{6}$$

The AHL algorithm relates the values of concepts and values of weights in the FCM model. We introduce a mathematical formalism for incorporating the learning rule, with the learning parameters and the introduction of the sequence of activation.

The proposed rule has the general mathematical form:

$$\Delta w_{ji} = \eta \cdot r(w_{ji}, A_j^{\mathrm{act}}) \cdot A_i - \gamma \cdot w_{ji} \tag{7}$$

where the coefficients η, γ are positive learning factors called learning parameters. The function $r(w_{ji}, A_j^{\mathrm{act}})$ is analogous to the learning signal function $r = r(w_i, x)$ in the case of general weight-learning rule [23].

In this algorithm, we propose the learning activation function r to be equal to the Activation value A_j^{act} of concept C_j that is considered as the Activation concept influencing the other concepts of FCM

$$r = r(w_{ji}, A_j^{\mathrm{act}}) = A_j^{\mathrm{act}} \tag{8}$$

From the learning equation (8), substituting the learning function r of Eq. (7), the learning rule for adjusting the FCM model, for discrete time dynamical type is

$$\Delta w_{ji} = \eta \cdot A_j^{\mathrm{act}}(k-1) \cdot A_i(k-1) - \gamma \cdot w_{ji}(k-1) \tag{9}$$

where the η is the learning rate parameter and γ is the weight decay parameter. The role and values of parameters η and γ will be explained later.

This simple rule states that if $A_i^{(k-1)}$ is the value of concept C_i at iteration $k-1$, and A_j^{act} is the value of the Activation concept C_j triggering the concept C_i at iteration step $k-1$, the corresponding weight from concept C_j towards the concept C_i increases proportional to their product multiplied with the learning rate parameter minus the weight decay.

Solving Eq. (9), the training weight rule takes the following form:

$$w_{ji}(k) = (1-\gamma) \cdot w_{ji}(k-1) + \eta \cdot A_j^{\mathrm{act}}(k-1) \cdot A_i(k-1) \tag{10}$$

In order to prevent indefinitely growing of weight values, we suggest normalization of weight at value 1, $\|\mathbf{w}\| = 1$, at each step update:

$$w_{ji}(k) = \frac{(1-\gamma) \cdot w_{ji}(k-1) + \eta \cdot A_j^{\mathrm{act}}(k-1) \cdot A_i(k-1)}{\left[\sum_{\substack{j=1 \\ j \neq i}} \left((1-\gamma) \cdot w_{ji}(k-1) + \eta \cdot A_j^{\mathrm{act}}(k-1) \cdot A_i(k-1)\right)^2\right]^{1/2}} \tag{11}$$

where the addition in the denominator covers just all the synapses connected to the Activated concept C_i.

When the learning rate parameter η is sufficiently small, then the second and higher-order terms can be neglected and the denominator, (considering the ith row of the weight matrix $\mathbf{w}$ constant and calculating the weights belonging to jth column of the weight matrix $\mathbf{w}$), becomes

$$\begin{aligned}
&\sum_{\substack{j=1\\ j\neq i}}^{N}((1-\gamma)\cdot w_{ji}(k-1)+\eta\cdot A_j^{\text{act}}\cdot A_i(k-1))^2\\
&\quad\simeq\sum_{j=1}^{N}\Big((1-\gamma)^2\cdot w_{ji}^2(k-1)+2\eta\cdot(1-\gamma)\cdot w_{ji}(k-1)\cdot A_j^{\text{act}}\cdot A_i(k-1)\Big)\\
&\quad\simeq(1-\gamma)^2+\sum_{j=1}^{N}(2\eta\cdot(1-\gamma)\cdot A_j^{\text{act}}(k-1)\cdot A_i(k-1)\cdot w_{ji}(k-1))\\
&\quad=(1-\gamma)^2+2\eta\cdot(1-\gamma)\cdot(A_j^{\text{act}}(k-1))^2\\
&\quad=(1-\gamma)^2\left[1+\frac{2\eta}{(1-\gamma)}\cdot(A_j^{\text{act}}(k-1))^2\right]
\end{aligned}$$

where N is the number of concepts.

The parameter γ is also a small weight decay parameter (it is defined in Section 4.3) and thus it can be considered that $(1-\gamma)^2\simeq 1$. Continuing the approximation, one gets:

$$\begin{aligned}
\frac{1}{\sqrt{1+\frac{2\eta}{(1-\gamma)}\cdot(A_j^{\text{act}}(k-1))^2}}&\simeq\frac{1}{1+\frac{\eta}{(1-\gamma)}\cdot(A_j^{\text{act}}(k-1))^2}\\
&\simeq 1-\frac{\eta}{(1-\gamma)}\cdot(A_j^{\text{act}}(k-1))^2 \qquad (12)
\end{aligned}$$

So the weight updating rule in Eq. (11), after normalization, using Eq. (12), becomes

$$\begin{aligned}
w_{ji}(k)=&[(1-\gamma)\cdot w_{ji}(k-1)+\eta\cdot A_j^{\text{act}}(k-1)\cdot A_i(k-1)]\\
&\cdot\left(1-\frac{\eta}{(1-\gamma)}\cdot(A_j^{\text{act}}(k-1))^2\right)\Rightarrow \qquad (13)
\end{aligned}$$

$$\begin{aligned}
w_{ji}(k)=&(1-\gamma)\cdot w_{ji}(k-1)+\eta\cdot A_j^{\text{act}}(k-1)\cdot A_i(k-1)\\
&-\eta\cdot w_{ji}(k-1)\cdot(A_j^{\text{act}}(k-1))^2 \qquad (14)
\end{aligned}$$

Thus the weight-training rule can—without much loss of precision— be simplified to

$$\begin{aligned}
w_{ji}(k)=&(1-\gamma)\cdot w_{ji}(k-1)+\eta\cdot A_j^{\text{act}}(k-1)\cdot[A_i(k-1)\\
&-w_{ji}(k-1)\cdot(A_j^{\text{act}}(k-1))] \qquad (15)
\end{aligned}$$

The first two terms $(1-\gamma)\cdot w_{ji}(k-1)+\eta\cdot A_j^{\text{act}}(k-1)\cdot A_i(k-1)$ in Eq. (14) represent the usual Hebbian modification of weights w_{ji}. They account for the self-amplification effect responsible for the self-organizing nature of the described infrastructure. The last term $-w_{ji}(k-1)\cdot(A_j^{\text{act}}(k-1))^2$, prevents an unlimited growth of w_{ji} and is responsible for stabilization.

If we consider a hypothetical concept C_i' with the $A_i'(k-1)$ where:

$$A_i'(k-1) = A_i(k-1) - w_{ji}(k-1)\cdot A_j^{\text{act}}(k-1) \tag{16}$$

then Eq. (15) is in direct correspondence to the classical Hebbian learning of Eq. (4).

The special form of decay of the previous equation (15) stops the weights from growing too large and helps the convergence of the learning process [27].

Thus, Eq. (1) that calculates the value of each concept of FCM is updating, taking the form of Eq. (5) where the value of weight $w_{ji}(k)$ is calculated using Eq. (15).

4.1. First criterion: objective function

The proposed Active Hebbian Learning algorithm has an asynchronous stimulation mode. Some concepts are considered as Activation and Activated concepts in each iteration step, where the Activated concepts are stimulated by the other interconnected Activation concepts. Also there are defined the outputs or Activation Decision Concepts that indicate the final states of the corresponding concepts after the applied stimulations.

The proposed learning algorithm of FCMs, is to some extent similar to correlation learning networks and Hebbian learning. A criterion function J is proposed for the AHL, which examines the values of outputs concepts that are the values of Activation Concepts we are interested about. We propose the criterion function J

$$J = \|\text{ADC}_i - A_i^{\min}\|^2 + \|\text{ADC}_i - A_i^{\max}\|^2 \tag{17}$$

where $A_i^{\min}$ is the minimum target value of the concept ADC_i and $A_i^{\max}$ is the corresponding maximum target value of ADC_i. This type of criterion function is appropriate for the AHL rule for FCMs. At the end of each cycle, the value of J calculates the Euclidean distance of ADC_i value from the minimum and maximum target values of the desired ADC_i, respectively. The minimization of the criterion function J is the ultimate goal, according to which we update the weights and determine the learning process.

If we consider the case of an FCM-model, where there are m Activation Decision Concepts, then for the calculation of J, we take the sum of the square differences between the m-ADCs values and the Eq. (17) takes the following form:

$$J = \sqrt{\sum_{i=1}^{m}\left[(\text{ADC}_i - A_i^{\min})^2 + (\text{ADC}_i - A_i^{\max})^2\right]} \tag{18}$$

The objective is to reach a minimum value of the criterion function J, for a set of weights. When criterion function J is minimized the desired equilibrium point-steady state of the FCM is reached.

4.2. Second criterion function

In addition to the previous statements, we introduce one more criterion for the AHL algorithm of FCMs in order to terminate the algorithm after a limited number of cycles, when the desired values for ADC(s) are reached, and so the system converges in the steady state (or fixed point). This second criterion is determined by the variation of the subsequent values of ADC_i concept, for simulation cycle c, yielding a value e, which has to be minimum, taking the form:

$$|\text{ADC}_i^{(c+1)} - \text{ADC}_i^{(c)}| < e \tag{19}$$

where ADC_i is the value of ith concept.

The term e is a tolerance level keeping the variation of values of ADC(s) as low as possible and it is proposed equal to $e = 0.001$, satisfying the termination of iterative process.

Thus we proposed and developed two criteria functions for the AHL algorithm, the first one ensures the convergence to the desired values for ADCs with the minimization of the criterion function J. The second one ensures the minimization of the variation of two subsequent values of Activation Decision Concepts (ADCs). Both criterion functions are represented in Eqs. (18) and (19), respectively, and they determine and terminate the iterative process of the learning algorithm.

4.3. Determination of learning parameters

In Eq. (15), η and γ are the learning parameters of the proposed algorithm. The learning rate parameter η is a small positive scalar parameter that is defined to decrease exponentially with simulation cycle, following Eq. (20):

$$\eta^{(c)} = b_1 \cdot \exp(-\lambda_1 \cdot c) \tag{20}$$

Convergence of FCMs depends on the step size $\eta^{(c)}$ decay with time, thus $\eta^{(c)}$ is selected to decrease and the rate of decrease depends on the speed of convergence to the optimum solution and on the weight updating mode. The parameters b_1 and λ_1 are positive learning factors, which are determined using the trial and error method [14].

The learning factor $\eta^{(c)}$ takes the following values that ensures convergence of concepts values:

$$\eta^{(c)} = 0.02 \cdot \exp(-0.2 \cdot c) \tag{21}$$

The weight decay coefficient γ may be zero, constant or may decrease by the number of simulation cycles c. This depends on the problem's constraints and the desired region. The parameter γ can be selected for each specific problem to ensure that the learning process converges in a desired steady state. If the parameter γ is selected as a decreasing function at each simulation cycle c, the following form is proposing:

$$\gamma^{(c)} = b_2 \cdot \exp(-\lambda_2 \cdot c) \tag{22}$$

where b_2 and λ_2 are positive constants which are determined using trial and error experimental process. These values influence the rate of convergence to the desired region and the termination of the algorithm.

The "learning rate" η determines the amount that the value of ADC_i is incremented during each simulation cycle. The weight decay γ determines the amount of the previous connection weight that is carried forward, i.e. how much the previous time weight value affect the next calculated weight value. Updating of weights continues until the final calculated weights no more change or change in a negligible amount for the process. High values of parameters η, γ may cause the objective function and FCM system to oscillate. The convergence process in desired equilibrium points is very sensitive to the values of η and γ. Thus the suggested bounds for these parameters are within [0 0.1]. Values of learning rate parameter η greater than around 0.1 cause the FCM to oscillate.

5. Implementation of AHL algorithm for training fuzzy cognitive maps consisting of *n*-concepts

In the previous section there were presented the theoretical justification of the proposed AHL rule, there were introduced the determination of the sequence of activation concepts, the Activation Decision Concepts (ADCs) and there were introduced two objective criteria functions. Here the proposed learning algorithm will be structured into a set of steps explaining the method of implementing the AHL algorithm.

During the development of FCMs, experts determine the concepts of the FCM, based on their knowledge and experience that models the behavior and the operation of the system. They know the relevant factors, the main characteristics of the system, and thus they determine the number and kind of concepts that consist of the FCM. Then, they determine the structure and the interconnections of the FCM using fuzzy conditional statements [8]. Also, they determine the sequence of activation concepts, the mode of the activation among concepts (synchronous-asynchronous) and they select the ADC(s).

The proposed training algorithm is consisted of seven steps. The third step is consisted of $n - p + 1$ sub-steps, where $1 \leqslant p \leqslant n$, n is the number of concepts and p is the number of synchronously (at the same step) Activated concepts. Thus, experts have to determine which of the concepts are the synchronously Activated concepts for the same iteration step. For example, for an FCM model consisting of six nodes, experts may determine that the three of the six nodes are triggered at the same iteration step that being the Activated concepts; this means that p is equal to 3, so the number of sub-steps would be $n - p + 1 = 4$.

- If $p = n - 1$, all the n-concepts are synchronously activated and from asynchronous type learning we pass to synchronous,
- when $p = 1$, all the n-concepts are activated asynchronously and the number of sub-steps is n, equal to the number of FCM concepts.

The number of all $n - p + 1$ sub-steps are considered as a recursive cycle, declared as c-cycle.

A flowchart of the proposed learning procedure is given in the Fig. 4. Considering an n-node FCM-model, the learning procedure using the AHL rule is consisted of the following steps:

Step 1: Initial values for concepts vector $\mathbf{A}^0$ and weight matrix $\mathbf{w}^{\text{initial}}$ are assigned. Experts determine the sequence of activation concepts and the Activation Decision Concepts (ADCs).

Step 2: Determination of learning parameters $\eta^{(c)}$ and $\gamma^{(c)}$, and the first simulation cycle starts ($c = 1$).

Step 3: Consisting of the following sub-steps, taking into account the sequence of activation:

Sub-step 1: The first Activation concept is C_j which triggers concept C_i, which is the Activated concept at this step. The value of Activated concept C_i is calculated for the k step by Eq. (5) using the previous (determined) activation values of the interconnected concepts and the updated weight matrix. Notably only the weights $w_{ji}(k)$ from the Activation concept C_j to its interconnecting Activated concepts C_i are updating, through Eq. (15) and all other weights remain unchanged at this step.

Sub-step 2: The Activation concept C_i affects the next Activated concepts C_l according to the sequence of activation. The value of step k is increased by 1, and concept C_l is updating. Its value $A_l^{(k)}$ is calculated from Eq. (5). At the same time only the weights $w_{il}(k)$ from the Activation concept C_i to its interconnected Activated concepts C_l are updating, through Eq. (15) and all other weights remain unchanged at this step.

...

Sub-step $n - p + 1$: The nth Activated concept C_n is fired due to influences from the Activation concepts C_l at this iteration step

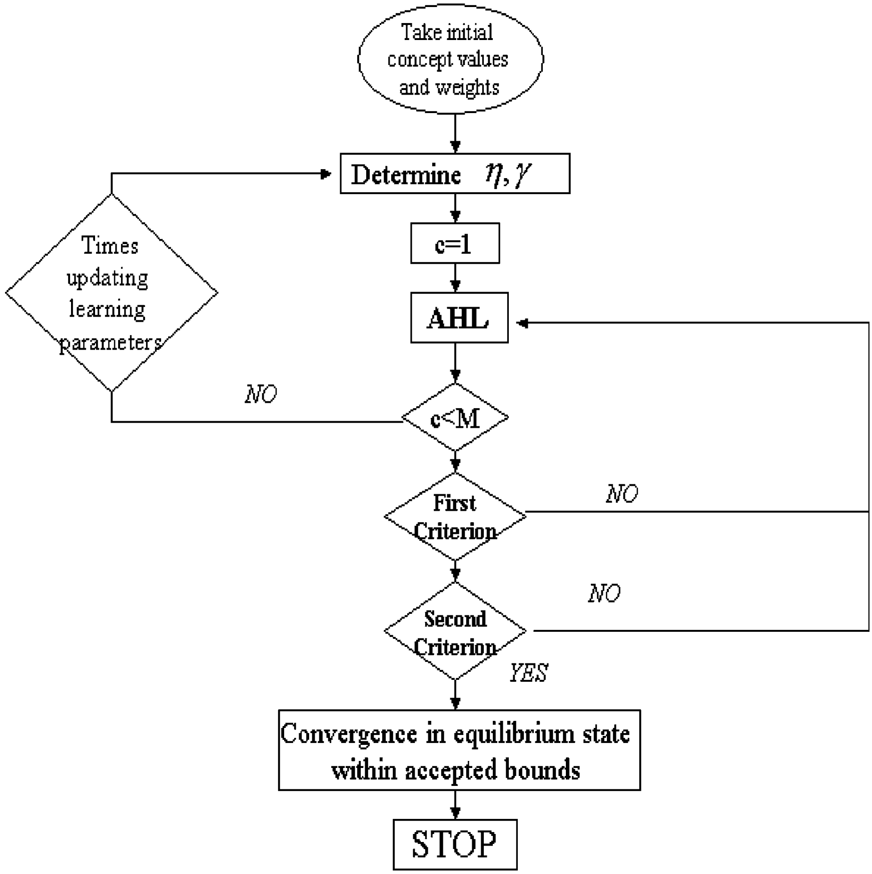

Fig. 4. Flowchart of the proposed AHL procedure.

$k = n - p + 1$. This nth Activated concept C_n supposed that it is selected by experts as the output concept and is defined as the Activation Decision Concept with value ADC_n.

When the cycle has closed the concept vector $\mathbf{A}^{\mathrm{act}}_{\mathrm{final}}$ is formed that represents the new state vector of the whole system after the simulation sub steps.

At this point, one recursive cycle is accomplished, after $k = n - p + 1$ substeps. The value(s) $\mathrm{ADC}^{(c)}_n$ of the Activation Decision Concept(s) at c-cycle is (are) used in the following steps.

Step 4: IF $c < M$, where $M = 100$ cycles, the objective function J is calculated for the c-cycle. ELSE GO TO Step 2 and redefine the parameters $\eta^{(c)}$ and $\gamma^{(c)}$, which have to be within their initially suggested bounds in subsection 4.3 and in order to lead the system in convergence.

Step 5: IF $J(c-2) > J(c-1) > J(c)$ is true GO TO next step, ELSE return to step 3, and a new cycle starts with simulation step $c = c+1$ and new value for $k = k+1$.

Step 6: Examination of second criterion and calculation of the difference between values of ADC_n for two subsequent recursive cycles, where the variation of ADC_n is less than the tolerance level, e.
IF Eq. (19) is false GO TO step 3.

Step 7: The two criteria are synchronously satisfied and the system converges in equilibrium state within accepted bounds. Thus the process STOPs.

Step 8: When the learning parameters have updating for 10 times then experts are asked to reconstruct the FCM model and the process starts from the beginning. The new initial weight matrix which derived after reconstruction of the FCM model, is used next in the AHL process till the system's convergence in equilibrium states.

This learning algorithm drives the system to converge in a desired region values for ADC concepts. The iteration of training process stops when the two suggested criteria are fulfilled simultaneously. The process of computing the objective function J, and adjusting the weights is repeated until a minimum value of J and a minimum variation of subsequent values for Activation Decision Concepts (ADC_n) are reached.

6. Implementation of AHL algorithm in FCM model for a process control problem

6.1. Statement of the problem

In this section the new proposed Activation Hebbian Learning (AHL) is implemented to train the FCM model for a process control problem. The most important component in developing the FCM is the determination of the concepts that best describe the system and the direction and grade of causality between concepts. These aspects will be represented through the following example. A part of a chemical plant is considered consisting of two tanks, three valves, one heating element and two thermometers for each tank, as depicted in Fig. 5.

Each tank has an inlet valve and an outlet valve. The outlet valve of the first tank is the inlet valve of the second tank. The objective of the control system is firstly to keep the height of liquid, in both tanks, between some limits, an upper limit $H_{\max}$ and a low limit $H_{\min}$, and secondly the temperature of the liquid in both tanks must be kept between a maximum value $T_{\max}$ and a minimum value $T_{\min}$. The temperature of the liquid in tank 1 is regulated through a heating element. The temperature of the liquid in tank 2 is measured through a sensor

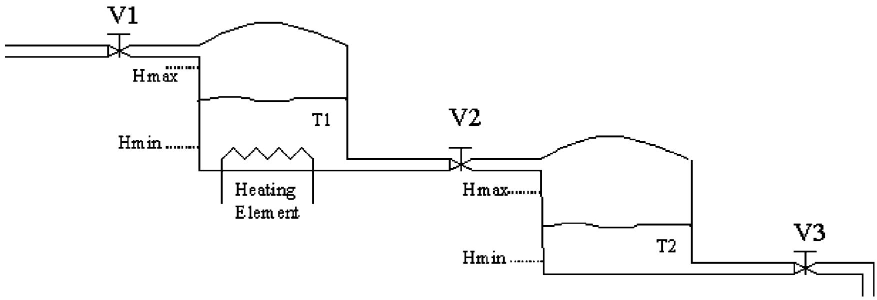

Fig. 5. The illustration of the chemical process example.

thermometer; when the temperature of the liquid two decreases, valve 2 needs opening, so hot liquid comes into tank 2 from tank 1. The control objective is to keep values of these variables in the following range of values:

$$\begin{aligned} &H^1_{\min} \leqslant H^1 \leqslant H^1_{\max} \\ &H^2_{\min} \leqslant H^2 \leqslant H^2_{\max} \\ &T^1_{\min} \leqslant T^1 \leqslant T^1_{\max} \\ &T^2_{\min} \leqslant T^2 \leqslant T^2_{\max} \end{aligned} \tag{23}$$

Fuzzy cognitive map that models and controls this system is developed and depicted on Fig. 6. Three experts constructed the FCM and jointly determined the concepts of the FCM [8,22]. Variables and states of the system, such as the

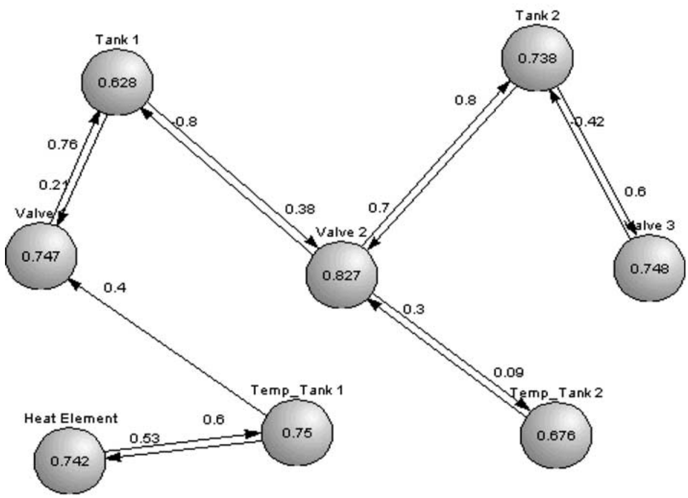

Fig. 6. The FCM model of the process problem.

height of the liquid in each tank or the temperature, will be the concepts of the FCM model, which describes the system. Then concepts are assigned for the system's elements that affect the variables such as the state of the valves. The values of the concepts correspond to the real measurements of the physical magnitude. Each concept of the FCM takes a value, which ranges in the interval $[0, 1]$ and it is obtained after threshold the real measurement of the variable or state, which each concept represent. The values of the weights will be determined after training the FCM using AHL algorithm. The Activation Decision Concepts (ADCs) in this problem, as experts propose them, are the concepts: C_1 of "Tank 1", C_2 of "Tank 2", C_6 of "Temperature 1" and C_7 of "Temperature 2".

For this simple system eight concepts are proposed and they give a good model of the system:

Concept 1. The height of the liquid in tank 1. The height of liquid is dependent on state of valve 1 and valve 2.

Concept 2. The height of the liquid in tank 2. The height of liquid is dependent on state of valve 2 and valve 3.

Concept 3. The state of the valve 1. The valve is open, closed or partially open.

Concept 4. The state of the valve 2. The valve is open, closed or partially open.

Concept 5. The state of the valve 3. The valve is open, closed or partially open.

Concept 6. The temperature of the liquid in tank 1.

Concept 7. The temperature of the liquid in tank 2.

Concept 8. Describes the operation of the heating element, which has different levels of operation and which increases the temperature of the liquid in tank 1.

Experts described how these concepts are connected with each other. The interconnections among concepts can easily be changed and any new can be added or others can be removed if the human operator decides so, in order to have a better model and operation of the system. Three experts used the methodology of Section 2 to determine the cause-effect relationship among concepts [22]. As an example, experts describe the influence of valve 1 (concept 3) on the amount of liquid in tank 1 (concept 1) using a set of fuzzy rules from which it is inferred that there is positive influence, transformed in numerical weight 0.76. Each connection between concepts has a weight, which ranges between $[-1, 1]$ and was determined by the group of experts.

Experts have suggested the initial weights for the FCM model that are shown in the following weight matrix:

$$\mathbf{w}^{\text{initial}} = \begin{bmatrix} 0 & 0 & 0.21 & 0.38 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.70 & 0.60 & 0 & 0 & 0 \\ 0.75 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -0.80 & 0.70 & 0 & 0 & 0 & 0 & 0.09 & 0 \\ 0 & -0.42 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.40 & 0 & 0 & 0 & 0 & 0.50 \\ 0 & 0 & 0 & 0.30 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.40 & 0 \end{bmatrix} \tag{24}$$

For the FCM model of the process control problem, experts determined that the values of concepts change asynchronously. Experts determined the sequence of Activation concepts and which concepts are fired at the same iteration step. Concept C_1 is defined as the first Activated concept. Concepts C_3 and C_4 are the synchronously Activated concepts, at next sub step, that are the second Activated concepts. Concept C_2 is the third Activated concept, concept C_5 is the fourth Activated concept, concepts C_6 and C_8 are the synchronously fifth Activated concepts and concept C_7 is the sixth Activated concept. Thus the *c*-cycle is consisted of six sub-steps.

The desired regions for the Activation Decision concepts (outputs), which reflect the proper operation of the modelled system, have been defined by the experts:

$$\begin{aligned} &0.55 \leqslant \mathrm{ADC}_1 \leqslant 0.75 \\ &0.75 \leqslant \mathrm{ADC}_2 \leqslant 0.8 \\ &0.75 \leqslant \mathrm{ADC}_6 \leqslant 0.82 \\ &0.65 \leqslant \mathrm{ADC}_7 \leqslant 0.75 \end{aligned} \tag{25}$$

Before the implementation of the proposed AHL algorithm, we apply Eq. (1) to check the FCM model interactions for the previously described chemical process control problem. The initial values of concepts are given in vector $\mathbf{A}^0 = [0.48\ 0.57\ 0.58\ 0.68\ 0.59\ 0.59\ 0.52\ 0.58]$, and they represent the real data of the physical process (after thresholding). These values and the initial weights from matrix $\mathbf{w}^{\text{initial}}$ are used in Eq. (1) to calculate the final state of the process without training. After 10 iteration steps an equilibrium state is reached and Fig. 7 gives the subsequent values of calculated concepts. The values of concepts for this equilibrium state are:

$$\mathbf{A}^{\text{equil}} = [0.6256\quad 0.7334\quad 0.7675\quad 0.8600\quad 0.7704\quad 0.7390\quad 0.6810\quad 0.7548]$$

It is observed that the values of concepts C_2 and C_6, in the final state, are out of the suggested desired regions in Eq. (25).

6.2. Training scenarios

In this section, three different training scenarios will be examined. In the first scenario, experts propose the initial values of concepts, they determine the

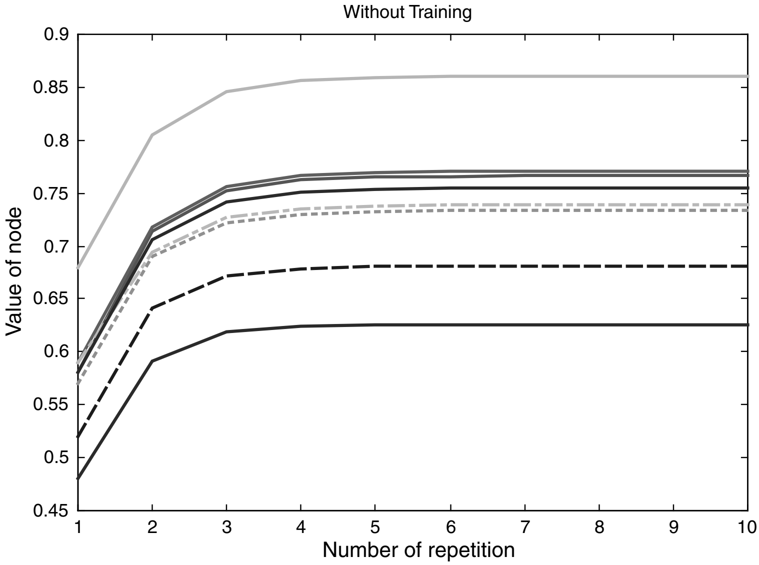

Fig. 7. The values of concepts for 10 simulation steps without training.

sequence of activation, and the parameter γ is considered equal to zero. When we implement the proposed AHL rule for the asynchronous learning process an updating weight matrix is derived. In the second scenario, the AHL algorithm is implemented for the same initial concepts values and the same sequence of activation, but for a constant value for parameter $\gamma = 0.02$ and new values of concepts and new convergence regions of Activation Decision Concepts (ADCs) are calculated. In the third scenario, we follow the same process implementing the AHL using an exponential attenuation equation for parameter γ. In this case the desired convergence regions for the proposed Activation Decision Concepts (ADCs) are succeeded.

6.2.1. First scenario

The training process starts by applying the initial values of concepts $\mathbf{A}^0_{\text{first}} = [0.48\ 0.57\ 0.58\ 0.68\ 0.58\ 0.59\ 0.52\ 0.58]$, representing the initial data of the process, and using the initial weights $\mathbf{w}^{\text{initial}}$. At second step of algorithm, the learning parameters η, γ are determined and the first simulation cycle starts. The parameter γ is defined equal to zero and the learning rate parameter η takes the suggested value of Eq. (21). At third simulation step (which is the first sub step) and for iteration number $k = 1$, the concept C_1 is defined as the first Activation concept and the value of Activation concept $A_1^{(1)}$, at iteration step $k = 1$, is calculated by Eq. (5). At the same time we calculate the weight values $w_{13}^{(1)}$, $w_{14}^{(1)}$ using Eq. (15). At second sub step the concept C_1, with its new value

$A_1^{(1)}$, triggers the interrelated C_3 and C_4 concepts. The concepts C_3 and C_4 are now the Activated concepts, and their activation values $A_3^{(2)}$ and $A_4^{(2)}$, for iteration number $k = 2$, are also calculated by Eq. (5). At the same time the weight values $w_{31}^{(2)}$, $w_{42}^{(2)}$, $w_{47}^{(2)}$ are calculated using Eq. (15). At the third sub step the previously activated concepts affect the concept C_2, which now is the Activated concept with value $A_2^{(3)}$, for $k = 3$. At the same time we calculate the weight values $w_{24}^{(3)}$, $w_{25}^{(3)}$ using Eq. (15). Now the concept C_2, with its calculated value $A_2^{(3)}$ acts as Activation concept, triggering the concept C_5. The C_5 as the forth Activated concept, takes the value $A_5^{(4)}$, for iteration $k = 4$, and triggers subsequently the concepts C_6 and C_8. The C_6 and C_8 as the fifth Activated concepts, take the values $A_6^{(5)}$ and $A_8^{(5)}$ respectively, using Eq. (15) for iteration $k = 5$. At this step we calculate the weight values $w_{63}^{(5)}$, $w_{68}^{(5)}$, $w_{86}^{(5)}$ using Eq. (15). The last Activated concepts C_6, C_8 act as Activation concepts, triggering the concept C_7. The C_7 as the sixth Activated concept, takes the value $A_7^{(6)}$, for iteration $k = 6$ and for the next simulation step triggers the other interconnected concepts.

Notably, only the weights connected from the Activation Concepts to the Activated concepts are updated at each iteration step using Eq. (15). All other weights remain unchanged at each iteration sub-step.

Thus, this AHL algorithmic procedure continues, until the synchronously satisfaction of the two objective criteria are met. The result of training the FCM is a set of connection weights w_{ji} that minimize the objective function J and satisfy synchronously the second criterion. The AHL algorithm iteratively updates the connection weights based on Eq. (15) and calculates the values of concepts based on the previously described asynchronous updating mode.

The AHL process stops after eight cycles where the two proposed criteria, Eqs. (18) and (19) are satisfied. The objective function J depends on the values of Activation Decision Concepts-ADCs and the value of weights. Fig. 8 shows the subsequent calculated values of activation concepts for eight recursive cycles.

The vector of concepts values in this equilibrium region is

$$\mathbf{A}_{\text{first}}^{\text{act}} = [0.7057 \quad 0.7658 \quad 0.8213 \quad 0.8965 \quad 0.8183 \quad 0.7964 \quad 0.7404 \quad 0.8105]$$

The updated matrix of weights after eight cycles, is the following:

$$\mathbf{w}_{\text{first-scen}} = \begin{bmatrix} 0 & 0.022 & 0.244 & 0.413 & 0.039 & 0.040 & 0.035 & 0.041 \\ 0.043 & 0 & 0.045 & 0.721 & 0.617 & 0.046 & 0.039 & 0.047 \\ 0.754 & 0.026 & 0 & 0.055 & 0.047 & 0.049 & 0.042 & 0.050 \\ -0.695 & 0.7052 & 0.054 & 0 & 0.052 & 0.080 & 0.131 & 0.054 \\ 0.045 & -0.381 & 0.047 & 0.052 & 0 & 0.048 & 0.042 & 0.049 \\ 0.044 & 0.027 & 0.429 & 0.051 & 0.044 & 0 & 0.041 & 0.524 \\ 0.041 & 0.024 & 0.042 & 0.335 & 0.040 & 0.040 & 0 & 0.041 \\ 0.045 & 0.027 & 0.047 & 0.052 & 0.044 & 0.428 & 0.042 & 0 \end{bmatrix} \tag{26}$$

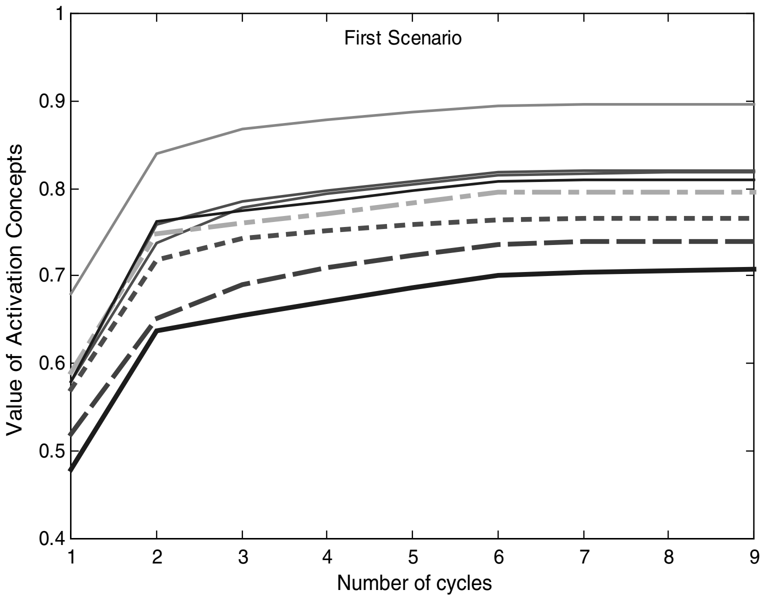

Fig. 8. The variation of concepts for eight cycles for the first scenario.

These new values for weights describe new relationships among the concepts of FCM. Actually, a new FCM model for the process has been produced. It is noticeable that the initial zero weights no more exist, and new interconnections with new weights have been assigned but 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. For example, the weighted arc w_{13} with initial value 0.21, after eight cycles takes the value 0.244, which means that the initial influence of C_1 towards C_3 increases at a small amount. The initial zero value of weights w_{23} has changed and after eight cycles is 0.045, which means that the concept C_2 affects the concept C_3 finally. Moreover, the weighted arc w_{41} with initial value −0.80 takes the value −0.695, which means that there is a negative decrease of the influence from concept C_4 to the "Tank 1" C_1. More specifically, this means that when the valve 2 takes the desired value, it influences negatively but at a smaller amount the height of liquid in tank 1, which will decrease opening the valve 2. Also, there are relative influences and physical meaning for all other weights. Thus, this AHL affects the dynamical behavior of the system and the equilibrium values for ADCs are within desired regions defined at Eq. (25).

6.2.1.1. Testing first scenario for 1000 random cases. We implemented the proposed AHL algorithmic approach and tested the FCM model for 1000 different $\mathbf{A}^0_{\text{random}}$ with random initial values of concepts, calculating the convergence regions for the four Activation Decision Concepts. All the results fall into the ranges:

$$\begin{aligned} &0.55 \leqslant \mathrm{ADC}_1 \leqslant 0.73 \\ &0.75 \leqslant \mathrm{ADC}_2 \leqslant 0.85 \\ &0.74 \leqslant \mathrm{ADC}_6 \leqslant 0.82 \\ &0.55 \leqslant \mathrm{ADC}_7 \leqslant 0.80 \end{aligned} \tag{27}$$

We observe that the above convergence values for the Activation Decision Concepts are different from the desired by experts. This means that using the proposed values for learning parameters η, γ the system reaches different than the desired equilibrium regions, for any random set of initial concepts values.

6.2.1.2. Evaluation of the modified weight matrix. Let us make now a testing for a random initial vector $\mathbf{A}^0_{\mathrm{random}}$ but using the previously derived with AHL algorithm weight matrix, $\mathbf{w}_{\mathrm{first-scenario}}$ as initial. It is $\mathbf{A}^0_{\mathrm{random}}$: $\mathbf{A}^0_{\mathrm{random}} =$ [0.1 0.45 0.37 0.20 0.85 0.04 0.18 0.01]. Applying the AHL algorithm, it will stop after 9 simulation steps and the derived concept vector A_{random} will be: $\mathbf{A}_{\mathrm{random}} =$ [0.7058 0.7658 0.8214 0.8965 0.8183 0.7964 0.7404 0.8105]. This new state vector has almost the same values as the previous state vector $\mathbf{A}^{\mathrm{act}}_{\mathrm{first}}$, in the convergence-desired region, that means if we use the $\mathbf{w}_{\mathrm{first-scenario}}$ for any initial A_{random}, FCM will reach the same equilibrium region. The Fig. 9 represents the variation values of concepts for this scenario for nine simulation steps.

6.2.2. Second scenario

At this scenario the learning parameter γ takes a constant value, equal to 0.02. For the same process example and following the same procedure described in first scenario, with the same values for learning rate parameter η, same initial values for concepts and weights, a new equilibrium region is reached. Eq. (15) is used for updating the weights and Eq. (5) is used for calculating the values of concepts. At every recursive c-cycle the two criteria are calculated and when they are successfully and simultaneously satisfied the training process terminates and the final Activation Decision Concepts states are reached. This requires 15 cycles where the final state vector $\mathbf{A}^{\mathrm{act}}_{\mathrm{second}}$ is derived

$$\mathbf{A}^{\mathrm{act}}_{\mathrm{second}} = [0.6972 \;\; 0.7410 \;\; 0.7882 \;\; 0.8577 \;\; 0.7857 \;\; 0.7667 \;\; 0.7226 \;\; 0.7787].$$

Fig. 10 illustrates all the subsequent values of calculated concepts for the 15 cycles.

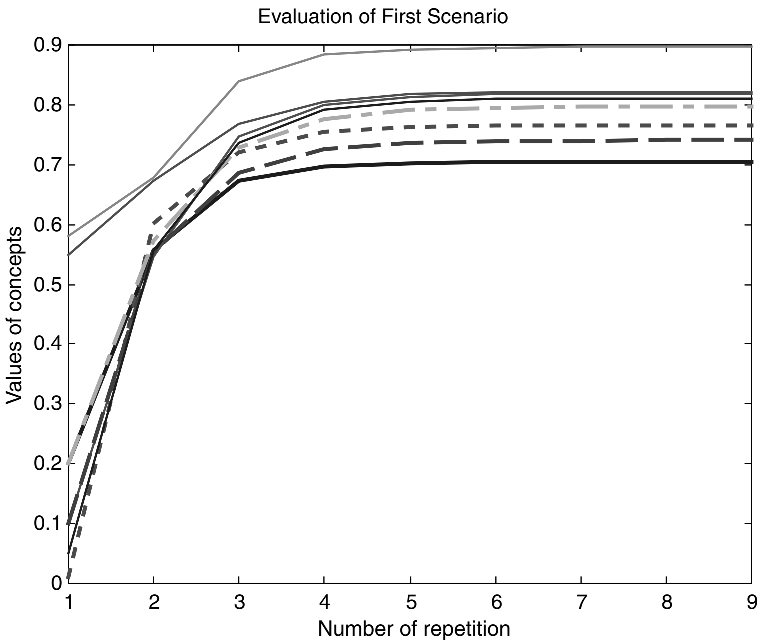

Fig. 9. The variation of concepts for nine simulation steps for random initial values.

The updated weight matrix of FCM is

$$\mathbf{w}_{\text{second-scen}} = \begin{bmatrix} 0 & 0.018 & 0.186 & 0.313 & 0.031 & 0.032 & 0.028 & 0.033 \\ 0.035 & 0 & 0.036 & 0.545 & 0.467 & 0.036 & 0.032 & 0.037 \\ 0.559 & 0.022 & 0 & 0.040 & 0.037 & 0.036 & 0.033 & 0.039 \\ -0.510 & 0.533 & 0.043 & 0 & 0.041 & 0.042 & 0.101 & 0.043 \\ 0.036 & -0.286 & 0.037 & 0.041 & 0 & 0.038 & 0.033 & 0.039 \\ 0.035 & 0.022 & 0.325 & 0.040 & 0.035 & 0 & 0.033 & 0.397 \\ 0.032 & 0.019 & 0.033 & 0.255 & 0.032 & 0.032 & 0 & 0.033 \\ 0.036 & 0.021 & 0.037 & 0.041 & 0.035 & 0.325 & 0.033 & 0 \end{bmatrix} \tag{28}$$

Also, in this scenario we observe that new relationships among the concepts of FCM are described; the initial zero weights no more exist, and new weights have been assigned. This means that all concepts affect the related concepts, and the weighed arcs show the degree of this relation. It is noticeable that some of the weights diverge significantly from the initial weight values suggested by experts. This happens because the algorithm affects the behavior of the systems, determining new weights for the desired convergence equilibrium point.

6.2.2.1. Testing second scenario for 1000 random cases. The convergence regions for the four Activation Decision Concepts resulting after simulation results for 1000 different cases with random initial values are the following:

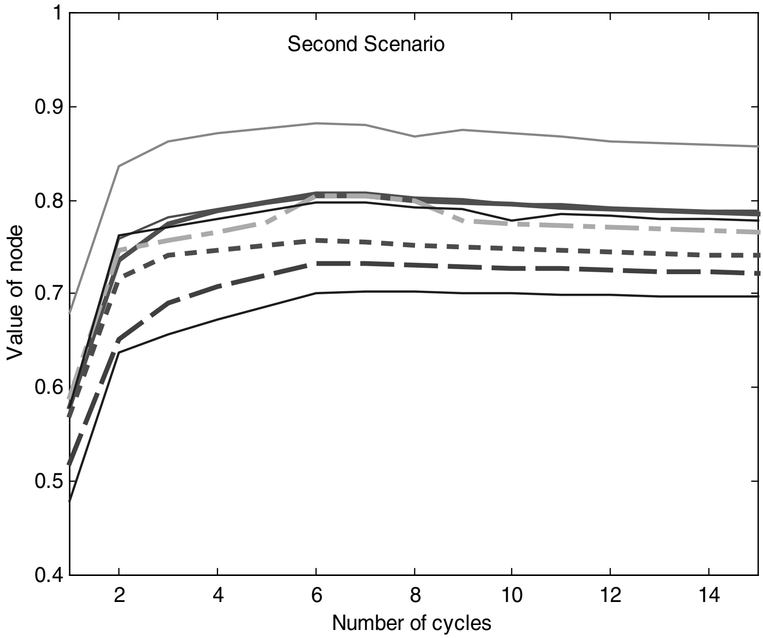

Fig. 10. The variation of concepts for 15 cycles in third scenario.

$$\begin{aligned} &0.5 \leqslant \mathrm{ADC}_1 \leqslant 0.75 \\ &0.72 \leqslant \mathrm{ADC}_2 \leqslant 0.78 \\ &0.74 \leqslant \mathrm{ADC}_6 \leqslant 0.79 \\ &0.58 \leqslant \mathrm{ADC}_7 \leqslant 0.77 \end{aligned} \tag{29}$$

It is observed that the two concepts C_1 and C_7 have broaden regions than the values required by experts. Only the concepts C_2 and C_6 keep their values in the initial constraint regions.

6.2.3. Third scenario

In this scenario, the exponential attenuation for parameter γ is used. The learning parameter γ is a learning weight decay coefficient defined equal to:

$$\gamma^{(c)} = 0.08 \cdot \exp(-c) \tag{30}$$

where the coefficient $b_2 = 0.08$ and $\lambda_2 = 1$ of Eq. (22) are positive constants that are determined using trial and error experimental results. The parameter γ is selected as a decreasing function at each simulation cycle c. The rate of convergence to the desired equilibrium region and the termination of algorithm are depend on the rate of decrease that mean on the selection of parameters b_2 and λ_2.

For this scenario, the same initial values for concepts and weights and the same learning rate parameter η are used with the first scenario. The training process terminates and the final Activation Decision Concepts states are reached when the two proposed criteria synchronously satisfied. The values of concepts after nine simulation cycles, are given in the following state vector: $\mathbf{A}^{\text{act}}_{\text{third}} = [0.7045\ 0.7707\ 0.8058\ 0.8790\ 0.8042\ 0.7828\ 0.7334\ 0.7955]$, and these values are within the desired convergence regions.

The updated weight matrix derived after nine cycles is

$$\mathbf{w}_{\text{third-scen}} = \begin{bmatrix} 0 & 0.035 & 0.213 & 0.358 & 0.037 & 0.038 & 0.033 & 0.038 \\ 0.041 & 0 & 0.042 & 0.624 & 0.535 & 0.043 & 0.038 & 0.044 \\ 0.634 & 0.042 & 0 & 0.051 & 0.044 & 0.045 & 0.039 & 0.046 \\ -0.576 & 0.612 & 0.050 & 0 & 0.048 & 0.049 & 0.116 & 0.050 \\ 0.042 & -0.305 & 0.044 & 0.048 & 0 & 0.045 & 0.039 & 0.045 \\ 0.041 & 0.039 & 0.375 & 0.047 & 0.041 & 0 & 0.038 & 0.455 \\ 0.038 & 0.036 & 0.039 & 0.293 & 0.037 & 0.038 & 0 & 0.039 \\ 0.042 & 0.040 & 0.043 & 0.048 & 0.041 & 0.372 & 0.039 & 0 \end{bmatrix} \tag{31}$$

6.2.3.1. Testing third scenario for 1000 random cases. We tested this scenario by implementing the AHL procedure for 1000 different cases using random initial values for concepts and for all cases the results for the Activation Decision Concepts where within the intervals:

$$\begin{aligned} &0.55 \leqslant \text{ADC}_1 \leqslant 0.72 \\ &0.76 \leqslant \text{ADC}_2 \leqslant 0.79 \\ &0.77 \leqslant \text{ADC}_6 \leqslant 0.80 \\ &0.65 \leqslant \text{ADC}_7 \leqslant 0.75 \end{aligned} \tag{32}$$

It is observed that the four Activation Decision Concepts take values in the desired suggested regions in Eq. (27), and specifically the two Concepts C_1, C_2 and C_6 take values in very narrow regions.

Fig. 11 represents the variation values of concepts for nine cycles after implementing the AHL algorithm.

Therefore it is proved that using the AHL algorithm we improve the FCM model, which exhibit equilibrium behaviour within the desired regions. With the proposed procedure the experts suggest the initial weights of the FCM, and then using the AHL algorithm a new weight matrix is derived that can be used for any set of initial values of concepts. The FCM converge to a steady state contributing to an updated weight matrix. Determining the learning parameters η, γ for the specific problem, and particularly, proposing the γ to be exponentially attenuated with the iteration steps, as described in third scenario, the system converges within the desired regions for all ADCs and the two of the four ADCs, take values in a range smaller than the proposed one.

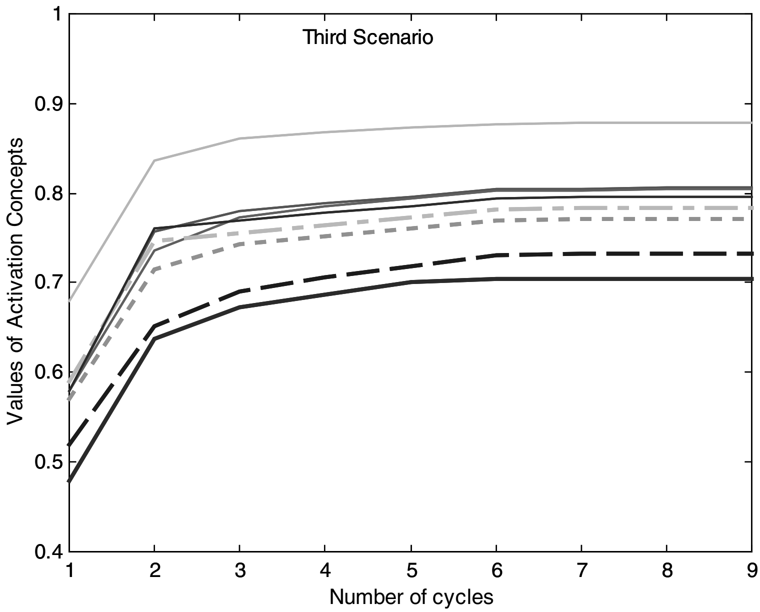

Fig. 11. The variation of concepts for nine cycles in second scenario.

The main advantage of AHL is that it determines new FCM causal links between all the concepts in order to succeed desired behavior of the system, and not only modify the initial causal links. The AHL is problem-dependent, starts using the initial weight matrix but all the process is independent from the initial values for concepts and the system succeed to converge in desired equilibrium regions for appropriate learning parameters.

7. Conclusions

The most significant weaknesses of the FCMs, namely their dependence on the expert's beliefs, and the potential convergence to undesired steady states, can be overcome by learning procedures. A new unsupervised learning methodology for Fuzzy cognitive maps training has been introduced, presented, implemented and tested for a chemical process control problem. The proposed AHL algorithm adjusts and modifies the weights of FCMs improving the FCM's efficiency and adaptability.

This paper proposes the mathematical analysis of Active Hebbian Learning algorithm. The proposed mathematical formulation and the implementation of the algorithm have been effectively investigated. Experimental results based on simulations of a process control system, verify the effectiveness, validity and especially the advantageous behavior of the proposed algorithm. Benefits of the proposed rule are in accordance with the practical framework of FCMs; easy to be used combined with flexibility and wide adaptability. The proposed AHL algorithm sustains a formal methodology for FCMs training, improving the functional FCM reliability and providing the FCM developers with learning parameters to adjust the influence of concepts. This type of learning rule accompanied with the good knowledge of the given system, guarantee the successful implementation of the proposed process.

In conclusion, this innovative training approach improves the FCM operation, eliminates the weaknesses, establish functional reliability and prove the practical importance of the proposed AHL algorithm through a process control application.

Acknowledgement

The work of E.I. Papageorgiou was supported by a grant from the Greek Scholarships Foundation (I.K.Y.).

References

[1] B. Kosko, Fuzzy cognitive maps, International Journal of Man–Machine Studies 24 (1986) 65–75.
[2] B. Kosko, Neural Networks and Fuzzy Systems, Prentice-Hall, Englewood Cliffs, 1992.
[3] B. Kosko, Fuzzy Engineering, Prentice-Hall, New Jersey, 1997.
[4] M. Hagiwara, Extended fuzzy cognitive maps, IEEE International Conference in Fuzzy Systems, 8–12 March, San Diego, CA, 1992, pp. 795–801.
[5] E.I. Papageorgiou, C.D. Stylios, P.P. Groumpos, An integrated two-level hierarchical decision making system based on fuzzy cognitive maps (FCMs), IEEE Transactions on Biomedical Engineering 50 (12) (2003) 1326–1339.
[6] E.I. Papageorgiou, C.D. Stylios, P.P. Groumpos, Decision Making in external beam radiation therapy based on FCMs, in: Proceedings of the 1st International IEEE Symposium on Intelligent Systems, IS 2002, Varna, Boulgaria, 2002, CD-ROM.
[7] E.I. Papageorgiou, P.P. Spyridonos, P. Ravazoula, C.D. Stylios, P.P. Groumpos, G.N. Nikiforidis, A fuzzy cognitive map model for grading urinary bladder tumours, in: Proceedings of NNESMED/CIMED 2003, 5th International Conference on Neural Networks and Expert Systems in Medicine, Sheffield, England, 2003, pp. 55–60.
[8] C.D. Stylios, P.P. Groumpos, Fuzzy cognitive maps in modeling supervisory control systems, Journal of Intelligent and Fuzzy Systems 8 (2) (2000) 83–98.
[9] R. Taber, Fuzzy cognitive maps model social systems, AI Expert 9 (1994) 8–23.
[10] C.E. Pelaez, J.B. Bowles, Using fuzzy cognitive maps as a system model for failure modes and effects analysis, Information Sciences 88 (1996) 177–199.
[11] R. Axelrod, Structure of Decision, the Cognitive Maps of Political Elites, Princeton University Press, Princeton, New Jersey, 1976, p. 404.
[12] Y. Miao, Z.Q. Liu, On causal inference in fuzzy cognitive maps, IEEE Transactions on Fuzzy Systems 8 (2000) 107–119.

[13] Z.Q. Liu, R. Satur, Contextual fuzzy cognitive map for decision support in geographic information systems, IEEE Transactions of Fuzzy Systems 5 (1999) 495–507.
[14] A.V. Huerga, A balanced differential learning algorithm in fuzzy cognitive maps, in: Proceedings of the Sixteenth International Workshop on Qualitative Reasoning 2002, 10–12 June, Barcelone, Spain.
[15] E.I. Papageorgiou, C.D. Stylios, P.P. Groumpos, Activation Hebbian learning rule for fuzzy cognitive maps, in: Proceedings of 15th IFAC World Congress of International Federation of Automatic Control, Barcelona, Spain, July 21–26, 2002, CD-ROM.
[16] J. Aguilar, Adaptive random fuzzy cognitive maps, in: F.J. Garijio, J.C. Riquelme, M. Toro (Eds.), IBERAMIA 2002, 28 July–1 August, Edmonton, Alberta, Canada, Lecture Notes in Artificial Intelligence, 2527, Springer-Verlag, Berlin, Heidelberg, 2002, pp. 402–410.
[17] E.I. Papageorgiou, C.D. Stylios, P.P. Groumpos, Fuzzy cognitive map learning based on nonlinear Hebbian Rule, in: T.D. Gedeon, L.C.C. Fung (Eds.), 16th International Conference on Artficial Intelligence, AI, Lecture Notes in Artificial Intelligence, 2903, Springer-Verlag, Berlin Heidelberg, 2003, pp. 254–266.
[18] E.I. Papageorgiou, K.E. Parsopoulos, P.P. Groumpos, M.N. Vrahatis, Fuzzy cognitive maps learning through swarm intelligence, in: International Conference Artificial Intelligence and Soft Computing, (ICAISC 2004), Zakopane, Poland, 7–11 June, 2004, Lecture Notes in Computer Science, in press.
[19] D.E. Koulouriotis, I.E. Diakoulakis, D.M. Emiris, Learning fuzzy cognitive maps using evolution strategies: a novel schema for modeling a simulating high-level behavior, in: Proceedings of IEEE Congress on Evolutionary Computation, Seoul, Korea, vol. 1, 2001, pp. 364–371.
[20] D.O. Hebb, The Organization of Behaviour: A Neuropsychological Theory, John Wiley, New York, 1949.
[21] O.X. Cordero, C.E. Pelaez, Fuzzy numbers for the improvement of Causal Knowledge Representation in FCMs, in: Proceedings of the 18th National Conference on Artificial Intelligence and 14th Conference on Innovative Applications of Artificial Intelligence (AAAI), 2002, pp. 949–950.
[22] C.D. Stylios, V. Georgopoulos, P.P. Groumpos, Fuzzy cognitive map approach to process control systems, Journal of Advanced Computational Intelligence 3 (5) (1999) 409–417.
[23] C.T. Lin, G. Lee, Neural Fuzzy Systems: A Neuro-Fuzzy Synergism to Intelligent Systems, Prentice-Hall, Upper Saddle River, 1996.
[24] P. Simpson, Artificial Neural Systems, Pergamon Press, Oxford, 1989.
[25] E. Oja, H. Ogawa, J. Wangviwattana, Learning in nonlinear constrained Hebbian networks, in: T. Kohonen et al. (Eds.), Artificial Neural Networks, North-Holland, Amsterdam, 1991, pp. 385–390.
[26] R.H. White, Competitive Hebbian learning: algorithm and demonstration, Neural Networks 5 (1992) 261–275.
[27] S. Haykin, Neural Networks: A Comprehensive Foundation, second ed., Prentice-Hall Inc, Englewood Cliffs, NJ, 1999.
[28] P. Baldi, K. Hornik, Learning in linear neural networks: a survey, IEEE Trans. Neural Networks 6 (1995) 837–858.
[29] T.D. Sanger, Optimal unsupervised learning in a single layer linear feed forward neural network, Neural Networks 2 (6) (1989) 459–473.
[30] M. Hassoun, Fundamentals of Artificial Neural Networks, MIT Press, Bradford Book, Massachusetts, 1995.
[31] S. Kartalopoulos, Understanding Neural Networks and Fuzzy Logic, Basic Concepts and Applications, IEEE Press, Inc, New York, 1996.
[32] R.J. Williams, D. Zipser, A learning algorithm for continually running fully recurrent neural networks, Neural Computation 1 (2) (1989) 270–280.

[33] E. Oja, Neural networks, principal components and subspaces, International Journal of Neural Systems 1 (1989) 61–68.
[34] P.P. Van der Smagt, Minimization methods for training feed forward neural networks, Neural Networks 7 (1) (1994) 1–11.