34

Fuzzy Cognitive Maps

E.I. Papageorgiou and C.D. Stylios

34.1 Introduction

Fuzzy cognitive maps (FCMs) are a modeling methodology based on exploiting knowledge and experience. FCMs are originated from the theories of fuzzy logic, neural networks, and evolutionary computing and generally from soft computing and computational intelligence techniques. FCMs are consisted of concepts representing conceptual entities, which could be considered as information granules. Concepts of FCMs interact; one influences the others and exchanges information. Thus, FCMs belong to the granular computing area, which refers to the conceptualization and processing of information granules – concepts.

The FCM structure is consisted of concepts representing information granules and the casual relationships among the information granules. These information granules could be physical quantities, qualitative characteristics, abstract ideas, and generally conceptual entities. A group of experts design the FCM; they determine the most important entities which are essential to model a system. Then they define the causal relationships among these entities, indicating the relative strength of the relationships using a fuzzy linguistic variable. The directions of the causal relationships are indicated with arrowheads. FCMs conceptual models are constructed by experts, who have a sufficient and abstract understanding of the specific system and its operation.

FCMs is a qualitative modeling tool; they provide a simple and straightforward way to model the relationships among different factors. FCM can describe any system using a model with three distinct characteristics: (1) signed causality indicating positive or negative relationship, (2) the strengths of the causal relationships take fuzzy values, and (3) the causal links are dynamic where the effect of a change in one concept/node affects other nodes, which in turn may affect other nodes. The first characteristic implies both the direction and the nature of the causality. The second characteristic assigns a fuzzy number or linguistic value to reflect the strength of the causality or the degree of association between concepts. Finally, the third characteristic reflects a feedback mechanism that captures the dynamic relationship of all the nodes, which may have temporal implications.

This chapter is going to thoroughly present FCMs and the different contributions to the field. After a short literature review on FCMs, a general method for designing and developing FCMs will be presented along with the most important methodologies for training and upgrading FCMs.

34.1.1 History of Fuzzy Cognitive Mapping and Applications

Cognitive maps are originated from graph theory, which Euler introduced in 1736 [1] based on directed graphs. In directed graphs, there are links (connections) between variables (nodes) with a direction [2].

Handbook of Granular Computing Edited by Witold Pedrycz, Andrzej Skowron and Vladik Kreinovich

Anthropologists applied signed digraphs to represent the different social structures in human society [3]. Axelrod [4] was the first who used digraphs to describe causal relationships among variables and to mimic the way human do. He called these digraphs cognitive maps. Cognitive mapping has been applied in many areas for decision making as well as to describe expert's perceptions on complex social systems [4–11].

Kosko [12] modified Axelrod's cognitive maps, with binary values; he suggested the use of fuzzy causal functions taking numbers in $[-1, 1]$, so he introduced the FCM. Kosko examined the behavior of FCMs explaining the inference mechanism of FCM. He applied FCMs to model the effect of different policy options using a computational method [13]. There was a very important update of cognitive maps combined them with fuzzy logic, which gave them the temporal and causal nature [14, 15]. FCMs express causality over time and allow for causality effects to fluctuate as input values change. Non-linear feedback can only be modeled in a time-based system. FCMs are intended to model causality, not merely semantic relationships between concepts. The latter relationships are more appropriately represented with semantic networking tools like SemNet [16]. By modeling causality over time, FCMs facilitate the exploration of the implications of complex conceptual models, as well as represent them with greater flexibility.

In general, changes in the topology or in the weight parameters of the FCM model may result in totally different inference outcomes. The extended fuzzy cognitive map (eFCM) [17] introduced time delay in the FCM parameters and discussed non-linear properties. Actually, the eFCM was an extension of FCM that simply uses binary-valued functions for weights. As a consequence, it had little improvement over FCM in terms of inference performance.

FCM have been successfully applied in many different scientific fields for modeling and decision making: political developments [18], electrical circuits [19], virtual sea world of dolphins, shark and fish [20], organizational behavior and job satisfaction [21], and the economic demographics of world nations [22]. FCMs were combined with data mining techniques to further utilize expert knowledge [23, 24].

Skov and Svenning [25] combined FCM with a geographic information system so that to apply expert knowledge to predict plant habitat suitability for a forest. Mendoza and Prabhu [26] used cognitive mapping to examine the linkages and interactions between indicators obtained from a multicriteria approach to forest management. FCMs were used to support the esthetical analysis of urban areas [27] and for the management of relationships among organizational members of airline services [28]. Furthermore, evaluation procedure for specifying and generating a consistent set of magnitudes for the causal relationships of an FCM, utilizing pairwise comparison techniques, has been presented [29].

Liu and Satur [30] investigated inference properties of FCMs and they proposed contextual FCMs introducing the object-oriented paradigm for decision support and they applied contextual FCMs to geographical information systems [31]. Miao and Liu proposed FCM as a robust and dynamic mechanism to represent the strength of the cause and the degree of the effect [32]. They stated that FCMs lack the temporal concept that is crucial in many applications, so that FCMs cannot effectively describe the dynamic nature of the cause. Miao *et al*. investigated the properties of the FCMs, in particular, *dynamics*, because causal inference systems by nature are dynamic systems with uncertainties and they introduced dynamic cognitive networks (DCNs), taking into account the three major causal inference factors, namely, the direction of the causal relationship, the strength of the cause, and the degrees of the effect, thus improving the capability of FCMs [33]. The DCN introduces a mechanism to quantify the description of concepts with the required precision. The arcs of DCNs are able to describe not only the causal relationship but also how it will make the effect and how long it takes for the effect to build up. Nevertheless, further research on learning paradigms and self-restructuring mechanisms in the DCN could have a significant impact on the development of intelligent systems and it could provide a flexible structure for effective causal inference in real-world applications [33].

Fuzzy cognitive networks (FCNs) [34] constitute an extension of the well-known FCMs [14] to be able to operate in continuous interaction with the physical system while at the same time to keep tracking the various operational equilibrium points met by the system. FCNs can model dynamical complex systems that change with time following non-linear laws. FCNs have been used to address the maximum power point trackers (MPPTs) in photovoltaic (PV) power systems by giving a good maximum power operation of any PV array under different conditions such as changing isolation and temperature [35]. Moreover, FCNs have been applied to control an anaerobic digestion process [36].

In bioengineering area, FCMs have been used for diagnosis and medical decision making. FCMs were used to model and analyze the radiotherapy process and they were successfully applied for decision making in radiation therapy planning systems [37]. FCMs have also been used to analyze the problem of specific language impairment diagnosis using several experts' opinions and introducing competitive learning methods [38]. Competitive learning methods for FCMs were successfully applied for medical diagnosis problems because they ensure that only one diagnosis will come up. FCMs have been used to model the supervisor for decision support actions during the labor [39]. Furthermore, an augmentation of FCMs based on learning methods was proposed to assist grade diagnosis of urinary bladder tumors [40]. An advanced medical decision support system was developed based on supplementing FCMs with case-based reasoning techniques taking advantages of the most essential characteristics of both methodologies [41].

34.2 Background of Fuzzy Cognitive Maps

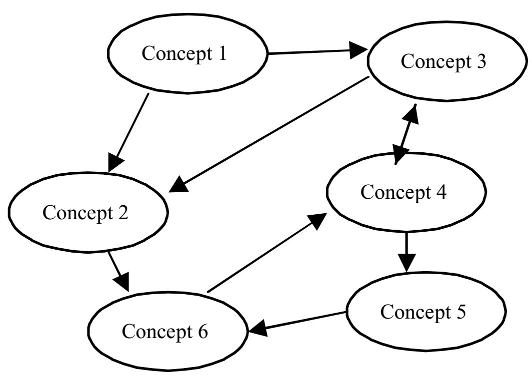
Kosko describes FCMs as fuzzy directional diagrams where feedback is permitted [42]. Like traditional causal concept maps, FCMs are consisted of nodes which represent variable concepts. The links between the concepts are signed with $+$ or $-$ to represent the positive or negative relationship between nodes and fuzzy-logic-based approach describes the degree of causality. FCMs can be used to create and model dynamic systems based on a causal explanation of interrelationships among concepts.

An FCM consisted of n concepts is represented in an $n \times n$ matrix. Generally, the causality between concepts is described by non-linear function $e(C_i, C_j)$, which describes the degree to which C_i influences C_j. The influence function takes values in the interval $[-1, 1]$.

See Table 34.1 for an example of a simple FCM matrix proposed by Kosko where weights take only bipolar values and Figure 34.1 illustrates a FCM with fuzzy weights.

Suppose that an FCM consists of n concepts. An $1 \times n$ matrix **A** represents the values of the n concepts and an $n \times n$ matrix **W** represents the causality of the relationships. In the weight matrix the row i represents the causality between concept i and all other concepts in the map. Noone concept is assumed to cause itself; thus, the diagonal of the matrix is zeroed. Each element w_{ji} of the matrix **W** indicates

Table 34.1 A simple FCM matrix



To/From	Concept 1	Concept 2	Concept 3	Concept 4	Concept 5	Concept 6
Concept 1	0	1	1	0	0	0
Concept 2	0	0	0	0	0	1
Concept 3	0	1	0	1	0	0
Concept 4	0	0	1	0	1	0
Concept 5	0	0	0	0	0	1
Concept 6	0	0	0	1	0	0

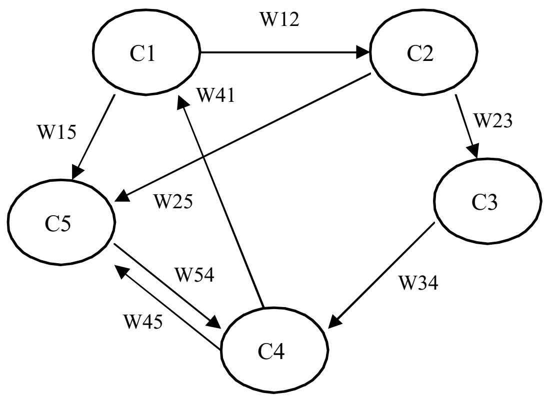


Figure 34.1 A simple FCM representation

the value of the weight w_{ji} between concept C_j and C_i. Equation (1) can be transformed as follows to describe the FCM operation using a compact mathematical equation:

$$\mathbf{A}^{(k)} = f(\mathbf{A}^{(k-1)} + \mathbf{A}^{(k-1)} \cdot \mathbf{W}) \tag{1}$$

where $\mathbf{A}^{(k)}$ is the column matrix with values of concepts at iteration step k, and f is the threshold function.

The FCM model of any system takes the initial values of concepts based on measurements from the real system and it is free to interact. The interaction is also caused by the change in the value of one or more concepts.

This interaction continues until the model [43]:

- Reaches an equilibrium fixed point, with the output values stabilizing at fixed numerical values.
- Exhibits limit cycle behavior, where the concept values falling in a loop of numerical values.
- Exhibits a chaotic behavior, where concept values reach a variety of numerical values in a nondeterministic, random way.

Compared to expert systems, FCMs are relatively quicker and easier to acquire knowledge and experience, especially exploiting the human approaches that do not usually take decisions using precise mathematics but using linguistic variables. The FCMs construction methods utilize different knowledge sources with diverse knowledge and different degrees of expertise. These knowledge sources can all be combined into one augmented FCM [44, 45]. The main advantage of the FCM development method is that there is no restriction on the number of experts or on the number of concepts.

34.2.1 General Methodology for Developing FCMs

The development and construction methodology of FCM has great importance for its success in modeling. FCM represents the human knowledge on the operation of the system and experts develop FCMs using their experience and knowledge on the system. Experts know what the main factors that influence the system are and what the essential elements of the system model may be; they determine the number and kind of concepts that the FCM is consisted of. Construction methodologies rely on the exploitation of experts' experience on system's modeling and behavior. Expert has observed the main factors that influence the behavior of the system; each one of these factors is represented by a concept at the FCM model. Experts, according to their experience, determine the concepts of FCM that may stand for events, actions, goals, values, and trends of the system. Moreover experts know which elements of the system influence other elements; for the corresponding concepts they determine the negative or positive effect of one concept on the others, with a fuzzy degree of causation. Causality is the key in representing human cognition and the human way in reaching a decision. In this way, an expert decodes his own knowledge on the behavioral model of the system and transforms this knowledge in a weighted graph, the FCM [46].

Interconnections among concepts express the cause-and-effect relationship that exists between two concepts; this relationship can be direct or indirect. Interconnections describe the influence that has the variation on the value of one concept in the value of the interconnected concept. This causal relationship is characterized with vagueness, because of its nature, as it represents the influence of one qualitative factor on another one and is determined using linguistic variables. The following definitions describe the procedure of determining the cause-and-effect relationship between concepts.

Definition 1. Direction of correlation between two concepts.
The causal relationship between two concepts can have the following directions:
Concept C_i influence concept C_j and there is a connection between $i \to j$, so $\delta_{i,j} = 1$. Either there is a connection with the reverse direction $j \to i$ when concept C_j influences concept C_i and so $\delta_{j,i} = 1$, or there is no connection between these two concepts:

$$i, j = \begin{Bmatrix} \delta_{i,j} = 1 \\ \delta_{j,i} = 1 \\ 0 \end{Bmatrix}.$$

Definition 2. Sign of correlation between two concepts.
Correlation between two concepts can be positive or negative:

(i) $W_{ij} > 0$*, which means that when value of concept C_i increases, the value of concept C_j increases, and when value of concept C_i decreases, the value of concept C_j decreases.*
(ii) W_{ij} $\quad 0$*, which means that when value of concept C_i increases, the value of concept C_j decreases, and when value of concept C_i decreases, the value of concept C_j increases.*

Definition 3. Degree of correlation between two concepts.
The value of the weight for the interconnection w_{ij} between concept C_i and concept C_j expresses the degree of correlation of the value of one concept on the calculation of the value of the interconnected concept. Linguistic variables are used to describe the strength of influence from one concept to another and the crisp transformation of linguistic values of weights w_{ij} belongs to the interval $[-1, 1]$.

Different methodologies have been proposed to develop FCMs and extract knowledge from experts [47, 48]. Experts are asked to describe the causality among concepts and the influence of one concept to another, using linguistic notions. At first step experts use Definition 1 to describe the direction of causality between two concepts, and then they determine the kind of the relationship using Definition 2. Then they describe the degree of the causal relationship between two concepts according to Definition 3 using the linguistic variable *Influence* and the grade of influence is described with a linguistic variable such as 'strong,' 'weak,' and etc.

Influence of one concept on another is interpreted as a linguistic variable taking values in the universe $U = [-1, 1]$ and its term set *T(influence)* is proposed to be

T(influence) = {negatively very strong, negatively strong, negatively medium, negatively weak, zero, positively weak, positively medium, positively strong, positively very strong}.

The semantic rule *M* is defined as follows and these terms are characterized by the fuzzy sets whose membership functions are shown in Figure 34.2. *M*(negatively very strong) = the fuzzy set for 'an influence below -75%' with membership function μ_{nvs}.

- *M*(negatively very strong) = the fuzzy set for 'an influence more than -75%' with membership function μ_{nvs}

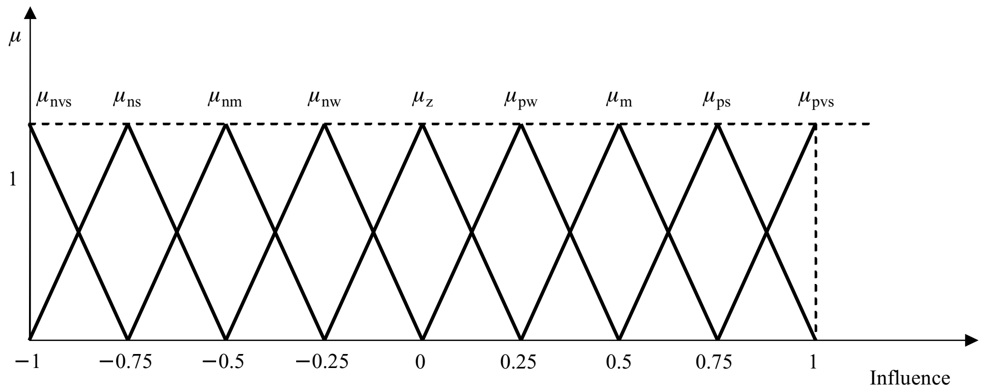


Figure 34.2 The nine membership functions corresponding to each one of the nine linguistic variables

- M(negatively strong) = the fuzzy set for 'an influence close to −75%' with membership function μ_{ns}
- M(negatively medium) = the fuzzy set for 'an influence close to −50%' with membership function μ_{nm}
- M(negatively weak) = the fuzzy set for 'an influence close to −25%' with membership function μ_{nw}
- M(zero) = the fuzzy set for 'an influence close to 0' with membership function μ_z
- M(positively weak) = the fuzzy set for 'an influence close to 25%' with membership function μ_{pw}
- M(positively medium) = the fuzzy set for 'an influence close to 50%' with membership function μ_{pm}
- M(positively strong) = the fuzzy set for 'an influence close to 75%' with membership function μ_{ps}
- M(positively very strong) = the fuzzy set for 'an influence above 75%' with membership function μ_{pvs}

The values of variable *Influence* belong to a set of nine members that can describe sufficiently the relationship between two concepts. This nine-member set is in correspondence with the human matter describing the causal relationship among concepts. The set of values of the linguistic variable *Influence* could be consisted by a great number but in this case the description of relationships would be very detailed. And an expert could not describe the relationship as 'very very much strong influence' and discern this relationship to the 'very much strong influence.' On the other hand the definition of the grades of influence must be quite detailed and not to have a few members, e.g., three members describing the influence with only three statements as weak, medium, and strong.

A group of experts develop the FCM. For each interconnection of the FCM, every expert assigns a linguistic variable that describes the influence from one concept to the other. For each interconnection the M experts assign M linguistic variables and so a set of M linguistic variables describe each interconnection. The M linguistic variables are combined using the fuzzy logic method of minmax and so an overall linguistic weight is produced which represents the strength of this interconnection, which is then transformed in numeric value using the defuzzification method of center of area [15]. The overall linguistic variable is transformed in the interval $[-1, 1]$. The same procedure is applied to all the interconnections among the N concepts of the FCM [47].

The main advantage of this methodology is that experts are asked to describe the grade of causality among concepts using linguistic variables and they do not have to assign numerical causality weights [48]. Thus, an initial weight matrix

$$W^{\text{initial}} = \begin{bmatrix} w_{11} & w_{12} & \dots & w_{1N} \\ w_{21} & w_{22} & \dots & w_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ w_{N1} & w_{N2} & \cdots & w_{NN} \end{bmatrix}$$

with $w_{ii} = 0, i = 1, \ldots N$ is obtained. Using the initial concept values, A_i, and the matrix W^{initial}, the FCM interacts through the application of the rule of equation (1).

The potential convergence to undesired steady states is a major deficiency of FCMs. Thus, new techniques are proposed that could further refine the experts' knowledge and significantly enhance their performance. Learning algorithms are used to increase the efficiency and robustness of FCMs by updating the weight matrix so as to avoid convergence to undesired steady states.

34.3 Learning Methods for FCMs

The methodology of developing FCMs mainly relies on human expert experience and knowledge. The external intervention (typically from experts) for the determination of FCM parameters, the recalculation of the weights, and causal relationships every time a new strategy is adopted, as well as the potential convergence to undesired regions for concept values were significant FCM deficiencies. It is necessary to overcome these deficiencies so that to improve efficiency and robustness of FCM. Weight adaptation methods are very promising as they can alleviate these problems by allowing the creation of less error-prone FCMs where causal links are adjusted through a learning process.

Experts are involved in the construction of FCM by determining concepts and causality among them. This approach may yield to a distorted model, because experts may not consider the most appropriate factors and may assign inappropriate causality weights among FCM concepts. A better conductance of FCMs is obtained by combining them with approaches based on neural network characteristics and integrating their advantages. Specifically, neural learning techniques can be used to train the FCM and modify the most appropriate weights of interconnections among concepts. The result is a hybrid neurofuzzy system. Learning methods have been proposed for FCM training, where the gradient for each weight is calculated by the application of general rules:

$$w'_{ij} = g(w_{ij}, A_i, A_j, A'_i, A'_j) \tag{2}$$

Learning rules can fine-tune FCM cause-and-effect relationships, meaning adjusting the interconnections between concepts, as if they were synapses in a neural network.

Kosko [12] mentioned for first time that adaptation and learning methodologies based on unsupervised Hebbian-type rules can be used to adapt the FCM model and adjust its weights. He proposed the differential Hebbian learning (DHL), as a suitable unsupervised learning technique, to train FCM, but without any mathematical formulation or and implementation in any problem [14, 20]. DHL method is based on the law expressed in the equation (3), which correlates changes between causal concepts:

$$\dot{w}_{ij} = -w_{ij} + \Delta C_i \Delta C_j, \tag{3}$$

where $\dot{w}_{ij}$ is the change of weight between concept *i*th and *j*th and where w_{ij} is the current value of this weight and $\Delta C_i \Delta C_j$ are changes in concepts *i*th and *j*th values, respectively.

The learning process iteratively updates the values of all weights at the FCM graph. Considering that value, ΔC_i is defined as the difference of *i*th concept values in two successive steps, which ranges between -1 and 1. The values of C_i and C_j concepts increase or decrease at the same direction only when $\Delta C_i \Delta C_j > 0$. If $\Delta C_i \Delta C_j$ 0, then one of the concept values decreases when the other one increases. In general, the weights of outgoing edges for a given concept node are modified when the corresponding concept value changes. The weights are updated according to the following formula:

$$w_{ij}(t+1) = \begin{cases} w_{ij}(t) + c_t[\Delta C_i \Delta C_j - w_{ij}(t)], \Delta C_i \neq 0 \\ w_{ij}(t), \Delta C_i = 0 \end{cases}, \tag{4}$$

where w_{ij} denotes the weight of the edge between concepts C_i and C_j, ΔC_i represent the change in the C_i concept's value, *t* is the iteration number, and c_t is a decreasing learning coefficient; e.g.,

$$c_t = 0.1\left[1 - \frac{t}{1.1N}\right], \tag{5}$$

where t is the current iteration number and the parameter N should be chosen to ensure that the learning coefficient c_t never becomes negative.

Huerga proposed an extension of DHL algorithm by introducing new rules to update edge values [49]. This algorithm was called balanced differential learning algorithm (BDLA). It eliminates the limitation of the initial DHL method where weight adaptation for an edge connecting two concepts (nodes) is dependent only on the values of these two concepts. But in BDLA, weights are updated taking into account all the concept values that change at the same time. This means that the formula for calculating $w_{ij}(t+1)$ takes into consideration not only changes ΔC_i and ΔC_j, but changes in all the other concepts if they occur at the same iteration and in the same direction. The BDLA algorithm was applied to adjust the structure of FCM models, which use bivalent transformation function, based on a historical data consisting of a sequence of state vectors. The goal of BDLA was to develop FCM, which would be able to generate identical sequence of state vectors given the same initial state vector. BDLA was improved in comparison to DHL method [49, 50], but both learning methods were applied only to FCMs with binary concept values, which significantly restricts their application areas. Another similar approach was the adaptive random FCMs based on the theoretical aspects of random neural networks [51].

The initial FCM model proposed by Kosko has two limitations. First, the model cannot describe many-to-one (or -many) causal relations. Second, the recall model gets trapped within limit cycles and is therefore not applicable in real-time decision-making problems. Konar and Chakraborty have proposed an extension of Kosko's model that can represent many-to-one (or -many) causal relations by using Petri nets. The proposed recall model was free from limit cyclic behavior. Other researchers selected fuzzy Petri nets (FPNs) [52, 53] to model FCMs due to the fact that FPNs support the necessary many-to-one (or -many) causal relations and have already proved themselves useful as an important reasoning [54, 55] and learning tool [56]. This unsupervised learning and reasoning process is realized with the adaptation of weights in an FPN using a different form of Hebbian learning. This proposed model converges to stable points in both encoding and recalls phases.

Moreover, a novel scheme of supervised learning on a fuzzy Petri net has been proposed in [57], providing semantic justification of the hidden layers and being capable of approximate reasoning and learning from noisy training instances. The algorithm for training a feedforward fuzzy Petri net and the analysis of its convergence have been successfully applied in object recognition from two-dimensional geometric views.

A different learning objective for FCM was presented by Khan and Chong in 2003 [58]. Instead of training the structure of FCM model, their approach was to find the initial state vector (initial conditions) that leads a given model to the desired fixed-point attractor or limit cycle.

Most of the existing learning approaches for weight updating do not take into account the feedback of the real system. A weight updating method for FCMs based on system feedback has been proposed by Boutalis *et al.* [34]. This method supposed that FCM reaches its equilibrium point using direct feedback from the node values of the real system and the learning limitations are imposed by the reference nodes. Moreover, the updating procedure of the method is enhanced using information on previous equilibrium points of the system operation. This is achieved by storing knowledge from already encountered operational situations into some fuzzy if–then rules.

Furthermore, two novel Hebbian-based approaches for FCM training, the active Hebbian learning (AHL) and the non-linear Hebbian learning (NHL) algorithms [59, 60], were proposed. The AHL algorithm takes into consideration the experts' knowledge and experience for the initial values of the weights, which are derived from the summation of experts' opinions. AHL algorithm supposes that there is a sequence of activation concepts, which is depending on the specific problem's configuration and characteristics. A seven-step AHL procedure is proposed to adjust the FCM weights. Mathematical formulation, implementation, and analysis of AHL supported by examples can be found in [59].

The core of the second approach, the NHL algorithm, is a non-linear extension to the fundamental Hebbian rule [61]. The main idea behind NHL is to update only those weights that experts determined, i.e., the non-zero weights. Weight values of FCM are updated synchronously, and yet they have fixed signs for the entire learning process. As a result, the NHL algorithm retains the structure of the obtaining model, which is enforced by the expert(s), but at the same time it requires human intervention before starting the learning process.

Moreover, evolutionary computation-based methods have been proposed for learning FCM causal weights by training the connection matrix of input data and thus eliminating expert involvement during development of the model. In Section 34.5, it is presented the evolutionary learning algorithms for FCMs proposed till today and referred in the corresponding literature.

34.4 Unsupervised Learning Algorithms for FCMs

34.4.1 The Active Hebbian Learning Algorithm

AHL algorithm has been introduced recently [59]. The novelty of this algorithm is based on supposing sequence of influence from one concept to another; in this way the interaction cycle is dividing in steps. During the FCM developing phase, experts are asked to determine the sequence of activation concepts, the activation steps, and the activation cycle. At every activation step, one (or more) concept(s) becomes activated concept(s), triggering the other interconnecting concepts, and in turn, at the next simulation step, may become activation concept. When all the concepts have become activated concepts, the simulation cycle has closed and a new one starts until the system converges in an equilibrium region.

In addition to the determination of sequence of activation concepts; experts select a limited number of concepts as outputs for each specific problem which are defined as the activation decision concepts (ADCs). These concepts are in the center of interest; they stand for the main factors and characteristics of the system, known as outputs and their values represent the final state of the system.

Suppose that there is the FCM shown on Figure 34.3, where experts determined the following activation sequence: $C_1 \rightarrow C_2, C_j \rightarrow C_i \rightarrow C_m \rightarrow C_n$. At second step of the cycle, according to the activation sequence concept C_j is the triggering concept that influences concept C_i, as shown in Figure 34.3. This concept C_j is declared the activation concept, with the value A_j^{act} that triggers the interconnected corresponding concept C_i, which is the activated concept. At the next iteration step, concept C_i influences the other interconnected concepts C_m and so forth. This learning algorithm has asynchronous stimulation mode which means that when concept C_j is the activation concept that triggers C_i, the corresponding weight w_{ji} of the causal interconnection is updated and the modified weight $w_{ji}^{(k)}$ is derived for each iteration step k.

Figure 34.3 is an instance of the FCM model during the activation sequence. The FCM model consists of n nodes and at the second activation step, the activation concept C_j influences the activated concept C_i. The following parameters are depicted in Figure 34.3:

C_i is the ith concept with value$A_i(k)$, $1 \leq i \leq n$.
w_{ji} is the weight describing the influence from C_j to C_i.

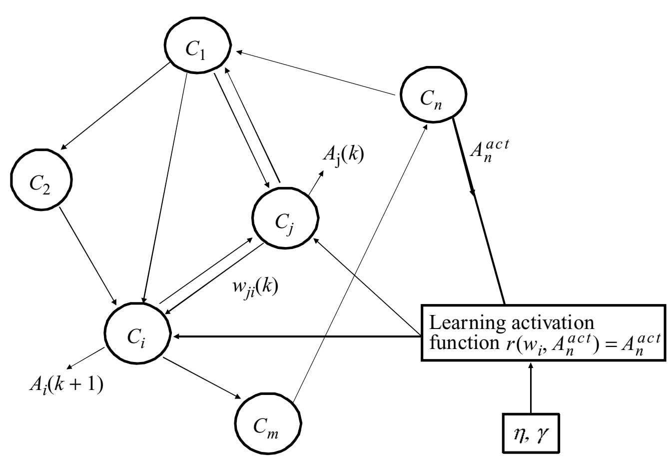


Figure 34.3 The activation weight-learning process for FCMs

$A_j^{act}(k)$ is the activation value of concept C_j, which is triggering the interconnected concept C_i.
γ is the weight decay parameter.
η is the learning rate parameter, depending on simulation cycle c.
$A_i(k)$ is the value of activated concept C_i, at iteration step k.

The value $A_i(k+1)$ of the activated concept C_i, at iteration step $k+1$, is calculated, computing the influence from the activation concepts with values A_l^{act} to the specific concept C_i due to modified weights $w_{li}(k)$ at iteration step k, through the equation

$$A_i(k+1) = f(A_i(k) + \sum_{l \neq i} A_l^{act}(k) \cdot w_{li}(k)), \tag{6}$$

where A_l are the values of 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 Figure 34.3, l takes values 1, 2, and j, and A_1, A_2, and A_j are the values of concepts C_1, C_2, and C_j that influence C_i. Thus, the value A_i of the concept, after triggering at step $k+1$, is calculated:

$$A_i(k+1) = f(A_i(k) + A_1^{act}(k) \cdot w_{1i}(k) + A_2^{act}(k) \cdot w_{2i}(k) + A_j^{act}(k) \cdot w_{ji}(k)). \tag{7}$$

The AHL algorithm relates the values of concepts and values of weights to the FCM model. We introduced a mathematical formalism for incorporating the learning rule, with the learning parameters and the introduction of the sequence of activation [59].

The proposed rule has the general mathematical form:

$$w_{ji}(k) = (1-\gamma) \cdot w_{ji}(k-1) + \eta \cdot A_j^{act}(k-1) \cdot A_i(k-1), \tag{8}$$

where the coefficients η and γ are positive learning factors called learning parameters.

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

$$w_{ji}(k) = \frac{(1-\gamma) \cdot w_{ji}(k-1) + \eta \cdot A_j^{act}(k-1) \cdot A_i(k-1)}{\left[\sum_{\substack{j=1 \\ j \neq i}} ((1-\gamma) \cdot w_{ji}(k-1) + \eta \cdot A_j^{act}(k-1) \cdot A_i(k-1))^2\right]^{1/2}}, \tag{9}$$

where the addition in the denominator covers all of the interconnections from the activation concepts C_j to the activated concepts C_i.

For low learning rates of parameters η, γ, equation (8) can – without any loss of precision – be simplified to

$$w_{ji}(k) = (1-\gamma) \cdot w_{ji}(k-1) + \eta \cdot A_j^{act}(k-1) \cdot [A_i(k-1) - w_{ji}(k-1) \cdot \left(A_j^{act}(k-1)\right)]. \tag{10}$$

The equation (1) that calculates the value of each concept of FCM takes the form of equation (6), where the value of weight $w_{ji}(k)$ is calculated using equation (10).

The learning parameters η and γ are positive scalar factors. The learning rate parameter η is exponentially attenuated with the number of activation-simulation cycles c so that the trained FCM converges fast. Thus, $\eta^{(c)}$ is selected to be decreased where the rate of decrease depends on the speed of convergence to the optimum solution and on the updating mode. Thus, the following equation is proposed:

$$\eta^{(c)} = b_1 \cdot \exp(-\lambda_1 \cdot c) \tag{11}$$

Depending on the problem's constraints and the characteristics of each specific case, the parameters b_1 and λ_1 may take values within the following bounds: 0.01 b_1 0.09 and $0.1 < \lambda_1$ 1, which are determined using experimental trial and error method for fast convergence.

The parameter γ is the weight decay coefficient which is decreasing following the number of activation cycles c. The parameter γ is selected for each specific problem to ensure that the learning process converges in a desired steady state. When the parameter γ is selected as a decreasing function at each activation cycle c, the following form is proposed:

$$\gamma^{(c)} = b_2 \cdot \exp(-\lambda_2 \cdot c), \tag{12}$$

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

In addition, for the AHL algorithm, two criteria functions have been proposed [59]. The first one is the criterion function J which examines the desired values of outputs concepts, which are the values of activation concepts we are interested in.

The criterion function J has been suggested as

$$J = \sqrt{\sum_{j=1}^{m} \left(\left[ADC_j - A_j^{\min} \right]^2 + \left[ADC_j - A_j^{\max} \right]^2 \right)}, \tag{13}$$

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

One more criterion for this learning algorithm of FCMs has been proposed. This second criterion is determined by the variation of the subsequent values of ADC_j concept, for simulation cycle c, yielding value e, which has to be minimum and takes the form

$$\left| ADC_j^{(c+1)} - ADC_j^{(c)} \right| \quad e, \tag{14}$$

where ADC_j is the value of jth concept.

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

Thus, for training FCM using the asynchronous AHL algorithm two criteria functions have been proposed. The first one is the minimization of the criterion function J and the second one is minimization of the variation of the two subsequent values of ADCs, represented in equations (13) and (14), respectively, in order to determine and terminate the iterative process of the learning algorithm.

The proposed algorithm is based on defining a sequence of concepts that means distinction of FCM concepts as inputs, intermediates, and outputs; this distinction depends on the modeled system and the focus of experts. During the training phase a limited number of concepts are selecting as outputs (those we want to estimate their values). The expert's intervention is the only way to address this definition. This learning algorithm extracts the valuable knowledge and experience of experts and can increase the operation of FCMs and implementation in real case problems just by analyzing existing data, information, and experts' knowledge about the given systems.

The training process implementing the AHL into an n-concept FCM is described analytically in [59]. The schematic representation of this training process is given in Figure 34.4. This learning algorithm drives the system to converge in a desired region of concepts values within the accepted-desired bounds for ADCs concepts.

34.4.2 Non-Linear Hebbian Learning Algorithm for FCMs

The second proposed unsupervised algorithm for training FCMs is based on the non-linear Hebbian-type learning rule for ANNs learning [62–64]. This unsupervised learning rule has been modified and adapted for the FCM case, introducing the NHL algorithm for FCMs.

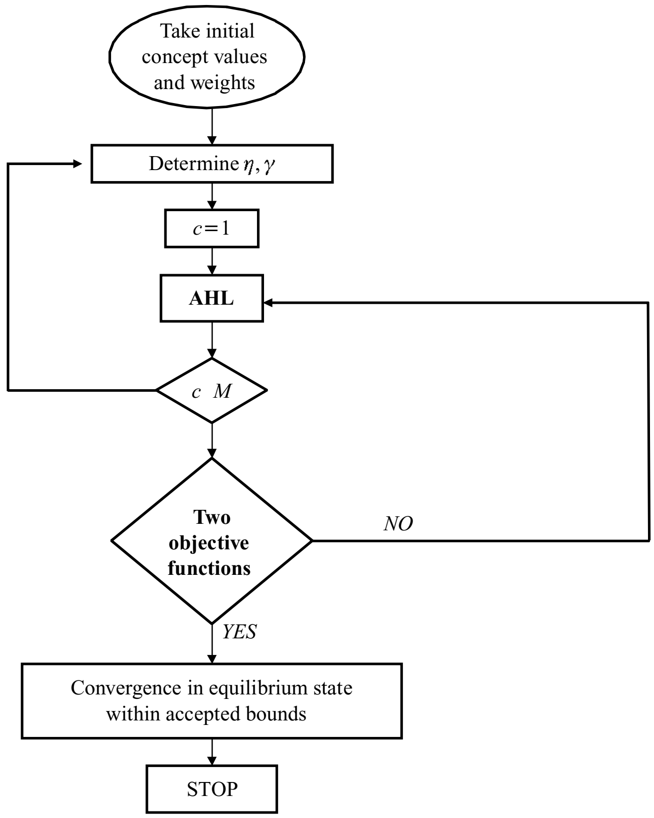


Figure 34.4 The flowchart of the training process using the AHL technique

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

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

$$A_i^{(k+1)} = f\left(A_i^{(k)} + \sum_{\substack{j \neq i \\ j=1}}^{N} A_j^{(k)} \cdot w_{ji}^{(k)}\right). \tag{15}$$

Taking the advantage of the general non-linear Hebbian-type learning rule for neural networks [64–66], we introduce the mathematical formalism incorporating this learning rule for FCMs. This algorithm relates the values of concepts and values of weights in the FCM model, and it takes the general mathematical form

$$\Delta w_{ji} = \eta A_i^{(k-1)}\left(A_j^{(k-1)} - w_{ji}^{(k-1)} A_i^{(k-1)}\right). \tag{16}$$

where the coefficient η is a very small positive scalar factor called learning parameter, which is determined using experimental trial-and-error method in order to optimize the final solution.

Equation (16) is modified and adjusted for FCMs and the following form of the non-linear weight-learning rule for FCMs is proposed:

$$w_{ji}^{(k)} = \gamma \cdot w_{ji}^{(k-1)} + \eta A_i^{(k-1)} \left(A_j^{(k-1)} - \operatorname{sgn}\left(w_{ji}^{(k-1)} \right) w_{ji}^{(k-1)} A_i^{(k-1)} \right), \tag{17}$$

where the γ is the weight decay learning coefficient.

The value of each concept of FCM is updated through the equation (15), where the value of weight $w_{ji}^{(k)}$ is calculated using equation (17).

Indeed, when experts develop an FCM, they usually propose a quite spare weight matrix $\mathbf{W}$. Using the NHL algorithm the initially non-zero weights are updating synchronously at each iteration step through the equation (17), until the termination of the algorithm. The NHL algorithm does not assign new interconnections and all the zero weights do not change value. When the algorithm termination conditions are met, the final weight matrix $\mathbf{W}_{NHL}$ is derived.

Implementation of NHL algorithm requires determination of upper and lower bounds for the learning parameter η; using trial-and-error experiments the values of learning rate parameter η were determined to belong in $0 < \eta < 0.1$. For any specific case-study problem, a constant value for η is calculated [67].

34.4.2.1 Two Termination Conditions

During the FCM development stage, experts define the desired output concepts (DOCs). These concepts stand for the main characteristics and outputs of the system that we want to estimate their values, which reflect the overall state of the system. The distinction of FCM concepts as inputs and outputs is determined by the group of experts for each specific problem. Experts select the output concepts and they consider the rest as initial stimulators or interior concepts of the system. The proposed learning algorithm extracts hidden and valuable knowledge of experts and it can increase the effectiveness of FCMs and their implementation for real problems.

Two complementary termination conditions of the NHL process have been proposed: the first termination condition is the minimization of the following cost function F_1:

$$F_1 = \sqrt{\left\| DOC_j^{(k)} - T_j \right\|^2}, \tag{18}$$

where T_j is the mean target value of the concept DOC_j. At each step, the value of F_1 calculates the square of the Euclidean distance of actual DOC_j value and mean target value T_j of the DOC_j values.

Let us assume that we want to calculate the cost function F_1 of concept C_j. It is required that DOC_j take values in the range $DOC_j = [T_j^{\min}, T_j^{\max}]$. Then the target value T_j of the concept C_j is determined as

$$T_j = \frac{T_j^{\min} + T_j^{\max}}{2}. \tag{19}$$

If we consider the case of an FCM model, where there are m DOCs, then for the calculation of F_1, we take the sum of the square differences between the m DOC values and the m Ts mean values of DOCs, and the equation (19) takes the following form:

$$F_1 = \sqrt{\sum_{j=1}^{m} (DOC_j^{(k)} - T_j)^2}. \tag{20}$$

The objective of the training process is to find the set of weights that minimize function F_1.

In addition to the previous statements, one more criterion for the NHL has been introduced so as to terminate the algorithm after a limited number of steps. This second criterion is based on the variation of the subsequent values of DOC_j concepts, for iteration step k, yielding a very small value e, Algorithm: "Nonlinear Hebbian Learning"

Step 1: *Read input concept state* $\mathbf{A}^{o}$ *and initial weight matrix* $\mathbf{W}^{o}$

Step 2: *For iteration step k.*

Step 3: *Update the weights:*

$$w_{ij}^{(k)} = w_{ij}^{(k-1)} + \eta A_j^{(k-1)}(A_i^{(k-1)} - sgn(w_{ij})w_{ij}^{(k-1)}A_j^{(k-1)})$$

Step 4: *Calculate* $A_i^{(k)}$ *according to the eq. (13)*

Step 5: *Calculate the two termination functions*

Step 6: *Until both the termination conditions are met, go to step 2*

Step 7: *Return the final weights* $\mathbf{W}_{NHL}$.

Figure 34.5 NHL algorithm for FCMs

taking the form

$$F_2 = \left| DOC_j^{(k+1)} - DOC_j^{(k)} \right| \quad 0.002, \tag{21}$$

where $DOC_j^{(k)}$ is the value of jth concept at iteration step k.

The constant value $e = 0.002$ has been proposed after a large number of simulations for different FCM cases. When the variation of two subsequent values of DOC_j is less than this number, it is pointless for the system operation to continue the training process.

When both terminations functions F_1 and F_2 are satisfied, the learning algorithm terminates and the desired equilibrium region for the DOCs is reached.

A generic description of the proposed NHL algorithm for FCMs is given in Figure 34.5 and the flowchart in Figure 34.6 describes the NHL-based algorithmic procedure.

34.5 Enhancing FCMs Using Evolutionary Computation Techniques

Evolutionary computation (EC) has become a standard term to indicate problem-solving techniques, which use design principles inspired from models and the natural evolution of species. Historically, there are three main algorithmic developments within the field of EC: evolution strategies [68, 69], evolutionary programming [70], and genetic algorithms [71, 72]. Common on these approaches is that they are population-based algorithms, which use operators inspired by population genetics to explore the search space. (The most typical genetic operators are reproduction, mutation, and recombination.) Each individual in the algorithm represents directly or indirectly (through a decoding scheme) a solution to the problem under consideration. The reproduction operator refers to the process of selecting the individuals that will survive and be part of the next generation. This operator typically uses a bias toward good-quality individuals: the better the objective function value of an individual, the higher the probability that the individual will be selected and therefore it will be part of the next generation. The recombination operator (often also called crossover) combines parts of two or more individuals and generates new individuals, also called offspring. The mutation operator is a unary operator that introduces random modifications to one individual.

Differences among the different EC algorithms concern the particular representation chosen for the individuals and the way genetic operators are implemented [68, 69, 71, 73]. For example, genetic algorithms typically use binary or discrete-valued variables to represent information in individuals and

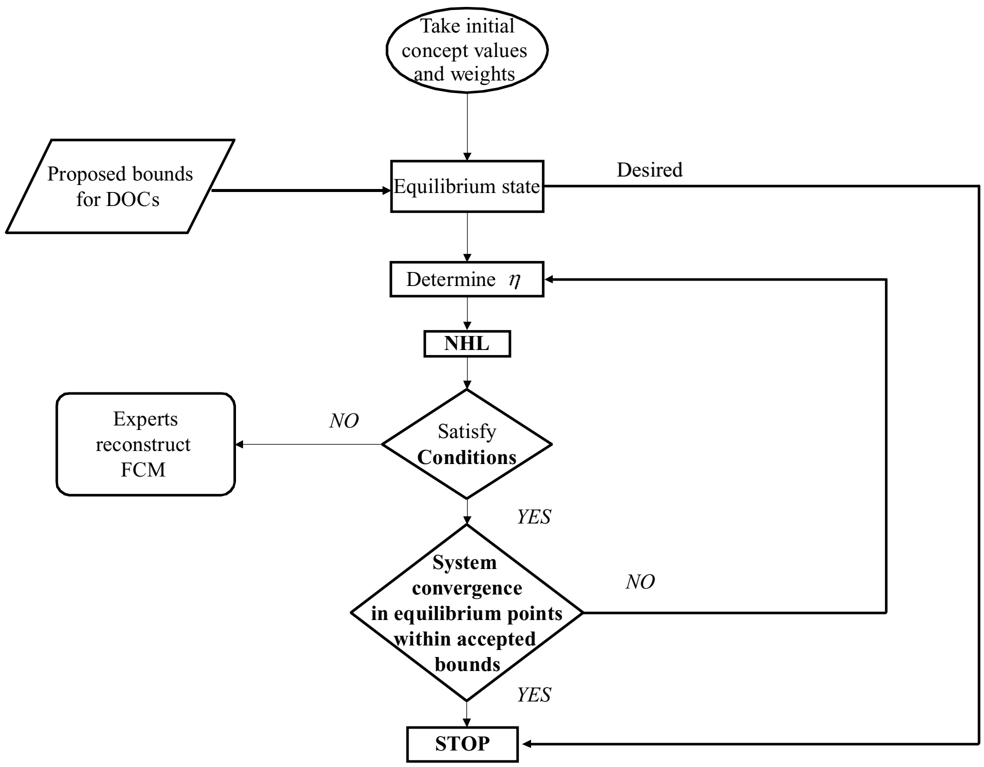


Figure 34.6 Flowchart of the training process using NHL technique

they favor the use of recombination, while evolution strategies and evolutionary programming often use continuous variables and put more emphasis on the mutation operator [72, 74, 75].

Koulouriotis *et al*. [76] were the first who applied evolution strategies for learning FCMs. Their technique was exactly the same used in neural networks training. In this case, the learning process is based on a collection of input/output pairs, which are called examples. Particular values of inputs and outputs depend on the designer's choice. Inputs are defined as the initial state vector values, whereas outputs are the final state vector values, i.e., values of state vector when the FCM simulation terminates. One of its main drawbacks is that it does not take into consideration the initial structure and experts' knowledge for the FCM model, but uses data sets determining input and output patterns in order to define the cause-and-effect relationships which satisfy the fitness function. Another main drawback is the need for multiple state vector sequences (input/output pairs), which might be difficult to obtain for many real-life problems. The calculated weights appear as large deviations from the actual FCM weights.

Recently, two different approaches based on the application of genetic algorithms for learning FCM connection matrix have been proposed. The first approach involving genetic algorithms performs a goal-oriented analysis of FCM [77]. This learning method did not aim to compute the weight matrix, but to find the initial state vector, which leads a predefined FCM (with a fixed weight matrix) to converge to a given fixed-point attractor or limit cycle solution. They viewed the problem of the FCM backward inference as one of optimization and applied a genetic algorithm-based strategy to search for the optimal stimulus state.

The second, more powerful, genetic algorithm-based approach has been proposed to develop FCM connection matrix which is based on historical data, consisting of a sequence of state vectors. It uses a real-coded genetic algorithm (RCGA) which allows eliminating expert involvement during development of the model and learns the connection matrix for an FCM that uses continues transformation function, which is a more general problem than the one considered in [78, 79]. The RCGA learning method is fully automated. Based on historical data given as time series (called input data) it establishes the FCM model (called candidate FCM), which is able to mimic the data. This approach is very flexible in terms of input data: it can use either one time series or multiple sets of concepts values over successive iterations. The central part of this method is the real-coded genetic algorithm, which is a floating-point extension to genetic algorithm [72]. The RCGA learning approach was intensively tested and experiments proved its effectiveness and high quality [79]. The analysis of RCGA learning ability is depending on the available size of historical data. Increasing the size of input data improves the accuracy of learning and on the other hand insufficient size of input data may result in poor quality of learning. In the latter case multiple different models that mimic input data of small size can be generated, and most of them fail to provide accurate results for experiments with new initial conditions, which were unseen during learning process [80]. The main advantage of this method is the absence of human intervention but the RCGA method needs investigation in terms of its convergence and more investigation to associate the GA parameters with the characteristics of a given experimental data set.

Another novel FCM learning procedure has been suggested by Parsopoulos *et al*. [81, 82] based on particle swarm optimization (PSO), which is a stochastic, population-based optimization algorithm. PSO belongs to the class of swarm intelligence algorithms [83], exploiting a population, called a swarm, of individuals, called particles, to probe the search space. Each particle moves with an adaptable velocity within the search space and retains a memory of the best position it ever encountered. The best position ever attained by all individuals of the swarm is communicated to all the particles in the case of global variant of PSO [84–87]. The PSO-based learning algorithm utilizes historical data consisting of a sequence of state vectors that leads to a desired fixed-point attractor state. It provides a search procedure, which optimizes a problem-depended fitness function $\varphi()$, by maintaining and evolving a swarm of candidate solutions. The individual of the swarm yielding the best fitness value throughout all generations, so that to give the optimal solution. Using the PSO method, a number of appropriate weight matrices can be derived leading the system to desired convergence regions. This approach is very fast and efficient to calculate the optimum cause-and-effect relationships of the FCM model and to overcome the main drawback of the FCM, which is the recalculation of the weights every time a new real case is adopted. This training method was successfully applied to an industrial control problem [82].

The detection of an appropriate weight matrix that leads the FCM to a steady state at which the output concepts lie in their corresponding bounds, while the weights retain their physical meaning, is the main aspect of this learning process. This aspect is attained by imposing constraints on the potential values assumed by weights. To succeed that, an objective function F was considered, which is non-differentiable (described in [81]) to calculate the desired values of output concepts by producing the appropriate weight matrix of the system. The PSO approach is used for the minimization of the objective function, due to the non-differentiability of F, and the global minimizers of the objective function are weight matrices that lead the FCM to a desired steady state. Generally, a plethora of weight matrices are produced through the PSO algorithm that lead the FCM to convergence to the desired regions. It is quite natural to obtain such suboptimal matrices that differ in subsequent experiments because PSO is a stochastic algorithm. All these matrices are proper for the FCM design and follow the constraints of the problem, though each matrix may have different physical meaning for the system.

Statistical analysis of the obtained weight matrices may help in the better understanding of the system's dynamics as it is implied by the weights, as well as in the selection of the most appropriate suboptimal matrix. The aforementioned procedure uses only primitive information from the experts. Nevertheless, any information available a priori may be incorporated to enhance the procedure, either by modifying the objective function in order to exploit the available information or by imposing further constraints on the weights. The proposed approach has proved to be very efficient in practice and its operation is illustrated on two different application areas in industry and medicine [81, 82].

34.6 Conclusion

FCMs are a conceptual modeling technique, consisted of concepts, the information entities, which interact following the causal relationships among the granular entities. FCMs take advantage of theories and approaches derived from discipline areas. FCMs synergistically utilize these theories of fuzzy sets, neurocomputing, evolutionary computing, and naturally inspired computing. FCMs representing good knowledge on a given system or process and accompanying them with computational intelligence techniques can contribute toward the establishment of FCMs as a robust technique.

One of the criticisms directed to FCMs is that they require human experts to come up with fuzzy causal relationship values, which may sometimes be inaccurate. Learning techniques can alleviate this problem by allowing the creation of less error-prone simple FCMs with causal links, which are then fuzzified through an adaptive learning process. The applicability of the proposed weight adaptation methodologies to real-world industrial problems has been successfully applied [88].

This chapter discussed the main aspects of FCMs representation, development, and then the learning approaches for FCMs. It presents their origin, the background, and the research efforts and contributions till today. Much research effort has been put on the utilization of neurocomputing methods, such as Hebbian learning algorithms for training FCMs and so developing augmented FCMs.

The most important characteristics of FCM model, which make it to differ from other knowledge-based models, is its interpretability, transparency, and efficiency in modeling and operation behavior of complex systems.

The main features of FCM model are summarized in the following:

- Representation of human knowledge on the modeling and operation behavior
- Exploitation of human's operator experience
- Selection of different factors of system
- Relations among different parts of system
- Symbolic representation of system's behavior
- Human-like reasoning
- Qualities for planning, decision making, and failure detection

All these qualities of FCMs make them suitable and applicable for modeling complex systems with many real applications.

Acknowledgments

The work of E.I. Papageorgiou was supported by a postdoctoral research grant from the Greek State Scholarship Foundation 'I.K.Y.'

References

[1] N.L. Biggs, E.K. Lloyd, and R.J. Wilson. *Graph Theory: 1736-1936*. Clarendon Press, Oxford, 1976.
[2] F. Harary, R.Z. Norman, and D. Cartwright. *Structural Models: An Introduction to the Theory of Directed Graphs*. John Wiley & Sons, New York, 1965.
[3] P. Hage and F. Harary. Structural models. In: *Anthropology*. Cambridge University Press, Cambridge, 1983.
[4] R. Axelrod. *Structure of Decision: The Cognitive Maps of Political Elites*. Princeton University Press, Princeton, NJ, 1976.
[5] M.G. Bougon, K. Weick, and D. Binkhorst. Cognition in organizations: An analysis of the Utrecht Jazz Orchestra. *Adm. Sci. Q.* 22 (1977) 606–639.
[6] S.M. Brown. Cognitive mapping and repertory grids for qualitative survey research: some comparative observations. *J. Manage. Stud.* 29 (1992) 287–307.
[7] K. Carley and M. Palmquist. Extracting, representing and analyzing mental models. *Soc. Forces* 70 (1992) 601–636.
[8] P. Cossete and M. Audet. Mapping of an idiosyncratic schema. *J. Manage. Stud.* 29 (1992) 325–347.
[9] J.A. Hart. Cognitive maps of three Latin America policy makers. *World Polit.* 30 (1977) 115–140.
[10] J.H. Klein and D.F. Cooper. Cognitive maps of decision makers in a complex game. *J. Oper. Res. Soc.* 33 (1982) 63–71.

[11] K. Nakamura, S. Iwai, and T. Sawaragi. Decision support using causal knowledge base. *IEEE Trans. Syst. Man Cybern.* SMC12 (1982) 765–777.
[12] B. Kosko. Fuzzy cognitive maps. *Int. J. Man-Mach. Stud.* 24 (1986) 65–75.
[13] Kosko. Hidden patterns in combined and adaptive knowledge networks. *Int. J. Approx. Reason.* 2 (1988) 377–393.
[14] B. Kosko. Fuzzy associative memory systems. In: *Fuzzy Expert Systems*.. CRC Press, Boca Raton, FL, 1992, pp. 135–162.
[15] C.T. Lin and C.S.G. Lee. *Neural Fuzzy Systems: A Neuro-Fuzzy Synergism to Intelligent Systems*. Prentice Hall, Upper Saddle River NJ, 1996.
[16] J.F. Sowa. *Principles of Semantic Networks: Explorations in the Representation of Knowledge*. Morgan Kaufmann Publishers, San Mateo, CA, 1991.
[17] M. Hagiwara. Extended fuzzy cognitive maps. In *Proceedings of the 1st IEEE International Conference on Fuzzy Systems*, New York, March 1992, pp.795–801.
[18] R. Taber. Knowledge processing with fuzzy cognitive maps. *Expert Syst. Appl.* 2 (1991) 83–87.
[19] M.A. Styblinski and B.D. Meyer. Fuzzy cognitive maps, signal flow graphs, and qualitative circuit analysis. In: *Proceedings of the 2nd IEEE International Conference on Neural Nets (ICNN-87)*, San Diego, CA, 1988, pp. 549–556.
[20] J. Dickerson and B. Kosko. *Virtual Worlds as Fuzzy Cognitive Maps. Presence*, Vol. 3, no. 2. MIT Press, MA, 1994, pp. 173–189.
[21] J.P. Craiger, D.F. Goodman, R.J. Weiss, and A. Butler. Modeling organizational behavior with fuzzy cognitive maps. *J. Comput. Intell. Organ.* 1 (1996) 120–123.
[22] M. Schneider, E. Shnaider, A. Kandel, and G. Chew. Automatic construction of FCMs. *Fuzzy Sets Syst.* 93 (1998) 161–172.
[23] T. Hong and I. Han. Knowledge-based data mining of news information on the Internet using cognitive maps and neural networks. *Expert Syst. Appl.* 23 (2002) 1–8.
[24] K.C. Lee, J.S. Kin, N.H. Chung, and S.J. Kwon. Fuzzy cognitive map approach to web-mining inference amplification. *J. Experts Syst. Appl.* 22 (2002) 197–211.
[25] F. Skov and J.-C. Svenning. Predicting plant species richness in a managed forest. *Forest Ecol. Manage.* 620 (2003) 1–11.
[26] G.A. Mendoza and R. Prabhu. Qualitative multi-criteria approaches to assessing indicators of sustainable forest resource management. *Forest Ecolog. Manage.* 174 (2003) 329–343.
[27] G. Xirogiannis, J. Stefanou, and M. Glykas. A fuzzy cognitive map approach to support urban design. *Expert Syst. Appl.* 26 (2) (2004) 257–268.
[28] I.I. Kang and S. Lee. Using fuzzy cognitive map for the relationship management in airline service. *Expert Syst Appl* 26 (4) (2004) 545–555.
[29] K. Muata and O. Bryson. Generating consistent subjective estimates of the magnitudes of causal relationships in fuzzy cognitive maps. *Comput. Oper. Res* 31 (8) (2004) 1165–1175.
[30] Z.Q. Liu and R. Satur. Contextual fuzzy cognitive map for decision support in geographical information systems. *J. IEEE Trans. Fuzzy Syst.* 7 (1999) 495–507.
[31] Z.Q. Liu. *Fuzzy Cognitive Maps: Analysis and Extension*. Springer, Tokyo, 2000.
[32] Y. Miao and Z.Q. Liu. On causal inference in fuzzy cognitive map. *IEEE Trans. Fuzzy Syst.* 8 (1) (2000) 107–120.
[33] Y. Miao, Z. Liu, C. Siew, and C. Miao. Dynamical cognitive network-an extension of fuzzy cognitive map. *IEEE Trans. Fuzzy Syst.* 9 (5) (2001) 760–770.
[34] Y.S. Boutalis, T.L. Kottas, B. Mertzios, and M.A. Christodoulou. A fuzzy rule based approach for storing the knowledge acquired from dynamical FCMs. In: *Proceedings of the 5th International Conference on Technology and Automation (ICTA'05)*, Thessaloniki, Greece, October 2005, pp. 119–124.
[35] Y. Boutalis, A. Karlis, and T. Kottas. Fuzzy cognitive networks + fuzzy controller as a self adapting control system for tracking maximum power point of a PV-Array. In: *Proceedings of 32nd Annual Conference of the IEEE Industrial Electronics Society*, *IECON 2006*, Paris, France, November 7–10, 2006, pp. 4355–4360.
[36] T. Kottas, Y. Boutalis, V. Diamantis, O. Kosmidou, and A. Aivasidis. A fuzzy cognitive network based control scheme for an anaerobic digestion process. In: *Proceedings of the 14th Mediterranean Conference on Control and Applications*, *Session TMS – Process Control*, Ancona, Italy, June 28–30, 2006, pp. 1–7.
[37] E.I. Papageorgiou, C.D. Stylios, and P.P. Groumpos. An integrated two-level hierarchical decision making system based on fuzzy cognitive maps (FCMs). *IEEE Trans. Biomed. Eng.* 50 (12) (2003) 1326–1339.
[38] V.C. Georgopoulos, G.A. Malandraki, and C.D. Stylios. A fuzzy cognitive map approach to differential diagnosis of specific language impairment. *Artif. Intell. Med.* 679 (2003) 1–18.
[39] C.D. Stylios, G. Georgoulas, and P.P. Groumpos. The challenge of using soft computing for decision support during labour. In: *Proceedings of 23rd Annual International Conference of the IEEE Engineering in Medicine & Biology Society*. Istanbul, Turkey, CD-ROM, October 25–28, 2001.

[40] E.I. Papageorgiou, C.D. Stylios, and P.P. Groumpos. Unsupervised learning techniques for fine-tuning fuzzy cognitive map causal links. *Int. J. Hum-Comput. Stud.* 64 (2006) 727–743.
[41] V.C. Georgopoulos and C.D. Stylios. Augmented fuzzy cognitive maps supplemented with case base reasoning for advanced medical decision support. In: M. Nikravesh, L.A. Zadeh, and J. Kacprzyk (eds), *Soft Computing for Information Processing and Analysis Enhancing the Power of the Information Technology, Studies in Fuzziness and Soft Computing*. Springer-Verlag, Berlin/Heidelberg, 2005, pp. 391–405.
[42] B. Kosko. Adaptive bi-directional associative memories. *IEEE Trans. Syst. Man Cybern.* 18 (1) (1988) 49–60.
[43] B. Kosko. *Fuzzy Engineering*. Prentice Hall, NJ, 1997.
[44] C.D. Stylios and P.P. Groumpos. A soft computing approach for modeling the supervisor of manufacturing systems. *J. Intell. Robot. Syst.* 26 (3–4) (1999) 389–403.
[45] C.D. Stylios, V. Georgopoulos, and P.P. Groumpos. introducing the theory of fuzzy cognitive maps in distributed systems. In: *Proceedings of the 12th IEEE International Symposium on Intelligent Control*, Istanbul, Turkey, July 16–18, 1997, pp. 55–60.
[46] C.D. Stylios and P.P. Groumpos. Modeling complex systems using fuzzy cognitive maps. *IEEE Trans. Syst., Man Cybern.* 34 (1) (2004) 155–162.
[47] C.D. Stylios, P.P. Groumpos, and V.C. Georgopoulos. A fuzzy cognitive maps approach to process control systems. *J. Adv. Comput. Intell.* 3 (5) (1999) 409–417.
[48] C.D. Stylios and P.P. Groumpos. Fuzzy cognitive maps in modeling supervisory control systems. *J. Intell. Fuzzy Syst.* 8 (2000) 83–98.
[49] A.V. Huerga. A balanced differential learning algorithm in fuzzy cognitive maps. In: *Proceedings of 16th International Workshop on Qualitative Reasoning2002*, Sitges, Spain, Poster, June 2, 2002.
[50] A. Vazquez. *A Balanced Differential Learning Algorithm in Fuzzy Cognitive Maps*. Technical Report. Departament deLlenguatges I Sistemes Informatics, Universitat Politecnica deCatalunya (UPC) 2002.
[51] J. Aguilar. Adaptive random fuzzy cognitive maps. In: F.J. Garijio, J.C. Riquelme, and M. Toro (eds), *Proceedings of IBERAMIA*, *Lecture Notes in Artificial Intelligence 2527*. Springer-Verlag, Berlin/Heidelberg, 2002, pp. 402–410.
[52] A.J. Bugarin and S. Barro. Fuzzy reasoning supported by Petri nets. *IEEE Trans. Fuzzy Syst.* 2 (2) (1994) 135–150.
[53] A. Daltrini and F. Gomide. An extension of fuzzy Petri nets and its applications. *IEEE Trans. Syst. Man Cybern.* 23 (5) (1993) 1255–1264.
[54] A. Konar and A.K. Mandal. Uncertainty management in expert systems using fuzzy Petri nets. *IEEE Trans. Knowl. Data Eng.* 8 (1) (1996) 96–105.
[55] A. Konar. *Artificial Intelligence and Soft Computing: Behavioral and Cognitive Modeling of the Human Brain*. CRC Press, Boca Raton, FL, 1999.
[56] W. Pedrycz and F. Gomide. A generalized fuzzy Petri net model. *IEEE Trans. Fuzzy Syst.* 2 (4) (1994) 295–301.
[57] A. Konar and U.K. Chakraborty. Reasoning and unsupervised learning in a fuzzy cognitive map. *Inf. Sci.* 170 (2–4) (2005) 419–441.
[58] M.S. Khan and A. Chong. Fuzzy cognitive map analysis with genetic algorithm. In: *Proceedings of the 1st Indian International Conference on Artificial Intelligence*, IICAI 2003, Hyderabad, India, December 18–20, 2003, IICAI 2003, pp. 1196–1205.
[59] E.I. Papageorgiou, C.D. Stylios, and P.P. Groumpos. Active Hebbian learning algorithm to train fuzzy cognitive maps. *Int. J. Approx. Reason.* 37 (3) (2004) 219–247.
[60] E.I. Papageorgiou and P.P. Groumpos. A weight adaptation method for fuzzy cognitive maps to a process control problem. In: M. Bubak, G.D.V. Albada, P.M.A. Sloot, and J.J. Dongarra (eds), *Proceedings of International Conference on Computational Science*, *ICCS 2004, Krakow, Poland, Lecture Notes in Computer Science 3037 Vol. II*. Springer-Verlag, Berlin/Heidelberg, 2004, pp. 515–522.
[61] E. Oja, H. Ogawa, and J. Wangviwattana. Learning in nonlinear constrained Hebbian networks. In: T. Kohonen, K. Mäkisara, O. Simula, and J. Kangas (eds), *Artificial Neural Networks*. North-Holland, Amsterdam, 1991, pp. 385–390.
[62] D.O. Hebb. *The Organization of Behaviour: A Neuropsychological Theory*. John Wiley, New York, 1949.
[63] E. Oja. Neural networks, principal components and subspaces. *Int. J. Neural Syst.* 1 (1989) 61–68.
[64] M. Hassoun. *Fundamentals of Artificial Neural Networks*. MIT Press, Bradford Book, MA, 1995.
[65] E. Oja. Data compression, Feature extraction, and autoassociation in feed- forward neural networks. In: T. Kohonen, K. Mäkisara, O. Simula, and J. Kangas (eds), *Artificial Neural Networks*. Elsevier Science Publishers B.V., North-Holland, Amsterdam 1991.
[66] L. Xu. Theories for unsupervised learning: PCA and its nonlinear extensions. In: *Proceedings of IEEE International Conference on Neural Networks*, Vol. II, New York, 1994, pp. 1252–1257.

[67] E.I. Papageorgiou, K.E. Parsopoulos, P.P. Groumpos, and M.N. Vrahatis. A first study of fuzzy cognitive maps learning using particle swarm optimization. In: *Proceedings of IEEE 2003 Congress on Evolutionary Computation*. IEEE Press, Canberra, Australia, December 8–12, 2003, pp. 1440–1447.

[68] I. Rechenberg. Evolution strategy, In: J.M. Zurada, R.J. Marks II, and C. Robinson (eds), *Computational Intelligence: Imitating Life*. IEEE Press, Piscataway, NJ, 1994, pp. 147–159.

[69] H.P. Schwefel. *Evolution and Optimum Seeking*. Wiley, New York, 1995.

[70] D. Fogel. *Evolutionary Computation: Towards a New Philosophy of Machine Intelligence*. IEEE Press, Piscataway, NJ, 1995.

[71] J.H. Holland. *Adaptation in Natural and Artificial Systems*. University of Michigan Press, MI, 1975.

[72] D.E. Goldberg. *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley/Prentice Hall, MA, 1989.

[73] A.P. Engelbrecht. *Fundamentals of Computational Swarm Intelligence*. John Wiley & Sons, IN, 2005.

[74] Z. Michalewicz. *Genetic Algorithms + Data Structures = Evolution Programs*. Springer, Berlin, 1994.

[75] M. Mitchell. *An Introduction to Genetic Algorithms*. MIT Press, Cambridge, MA, 1996.

[76] D.E. Koulouriotis, I.E. Diakoulakis, and D.M. Emiris. Learning fuzzy cognitive maps using evolution strategies: A novel schema for modeling a simulating high-level behavior. *Proc. IEEE Congr. Evol. Comput.* 1 (2001) 364–371.

[77] M.S. Khan, S. Khor, and A. Chong. Fuzzy cognitive maps with genetic algorithm for goal-oriented decision support. *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.* 12 (2004) 31–42.

[78] W. Stach, L. Kurgan, W. Pedrycz, and M. Reformat. Evolutionary development of fuzzy cognitive maps. In: *Proceedings of 2005 IEEE International Conference on Fuzzy Systems (FUZZ IEEE 2005)*, Reno, NV, 2005a, pp. 619–624.

[79] W. Stach, L. Kurgan, W. Pedrycz, and M. Reformat. Genetic learning of fuzzy cognitive maps. *Fuzzy Sets Syst.* 153 (3) (2005b) 371–401.

[80] W. Stach, L. Kurgan, W. Pedrycz, and M. Reformat. Learning fuzzy cognitive maps with required precision using genetic algorithm approach. *Electron. Lett.* 40 (24) (2004) 1519–1520.

[81] K.E. Parsopoulos, E.I. Papageorgiou, P.P. Groumpos, and M.N. Vrahatis. Evolutionary computation techniques for optimizing fuzzy cognitive maps in radiation therapy systems. In: K. Deb, R. Poli, W. Banzhaf, H.-G. Beyer, E. Burke, P. Darwen, D. Dasgupta, D. Floreano, J. Foster, M. Harman, O. Holland, and P.L. Lanzi (eds), *Proceedings of Genetic and Evolutionary Computation Conference (GECCO), Seattle, Washington, USA, June 26–30, 2004, Lecture Notes in Computer Science 3102*. Springer-Verlag Publications, Berlin/Heidelberg, 2004b, pp. 402–413.

[82] E.I. Papageorgiou, K.E. Parsopoulos, C.D. Stylios, P.P. Groumpos, and M.N. Vrahatis. Fuzzy cognitive maps learning using particle swarm optimization. *Int. J. Intell. Inf. Syst.* 25 (1) (2005) 95–121.

[83] J. Kennedy and R.C. Eberhart. *Swarm Intelligence*. Morgan Kaufmann Publishers, San Francisco, 2001.

[84] R.C. Eberhart, P. Simpson, and R. Dobbins. *Computational Intelligence PC Tools*. Academic Press, Professional (APP), New York, 1996.

[85] J. Kennedy and R.C. Eberhart. Particle swarm optimization. In: *Proceedings of IEEE International Conference on Neural Networks*, Vol. IV. IEEE Service Center, Piscataway, NJ, 1995, pp. 1942–1948.

[86] K.E. Parsopoulos and M.N. Vrahatis. Recent approaches to global optimization problems through particle swarm optimization. *Natural Computing* 1 (2–3) (2002) 235–306.

[87] K.E. Parsopoulos and M.N. Vrahatis. On the computation of all global minimizers through particle swarm optimization. *IEEE Trans. Evol. Comput.* 8 (3) (2004a) 211–224.

[88] E.I. Papageorgiou, C.D. Stylios, and P.P. Groumpos. Fuzzy cognitive map learning based on nonlinear hebbian rule. In: T.D. Gedeon and L.C.C. Fung (eds), *Proceedings on Artificial Intelligence. Lecture Notes in Artificial Intelligence 2903.* Springer-Verlag, Berlin/Heidelberg, 2003b, pp. 254–266.

Part Three

Applications and Case Studies