Learning and Convergence of Fuzzy Cognitive Maps Used in Pattern Recognition

Peer-reviewed author version

NAPOLES RUIZ, Gonzalo; PAPAGEORGIOU, Elpiniki; Bello, Rafael & VANHOOF, Koen (2016) Learning and Convergence of Fuzzy Cognitive Maps Used in Pattern Recognition. In: NEURAL PROCESSING LETTERS, 45 (2), pag. 431-444.

DOI: 10.1007/s11063-016-9534-x
Handle: http://hdl.handle.net/1942/22971

Learning and convergence of Fuzzy Cognitive Maps used in pattern recognition

Gonzalo Nápoles[1,2,*], Elpiniki Papageorgiou[3,1], Rafael Bello[2] Koen Vanhoof [1]

[1]Faculty of Business Economics, Hasselt University, Belgium

[2]Department of Computer Sciences, Central University of Las Villas, Cuba

[3]Department of Computer Engineering, Technological Education Institute of Central Greece, Greece

Abstract. In recent years Fuzzy Cognitive Maps (FCM) have become an active research field due to their capability for modeling complex systems. These recurrent neural models propagate an activation vector over the causal network until the map converges to a fixed-point or a maximal number of cycles is reached. The first scenario suggests that the FCM converged, whereas the second one implies that cyclic or chaotic patterns may be produced. The non-stable configurations are mostly related with the weight matrix that defines the causal relations among concepts. Such weights could be provided by experts or automatically computed from historical data by using a learning algorithm. Nevertheless, from the best of our knowledge, population-based algorithms for FCM-based systems do not include the map convergence into their learning scheme and thus, non-stable configurations could be produced. In this research we introduce a population-based learning algorithm with convergence features for FCM-based systems used in pattern classification. This proposal is based on a heuristic procedure, called Stability based on Sigmoid Functions, which allows improving the convergence of sigmoid FCM used in pattern classification. Numerical simulations using six FCM-based classifiers have shown that the proposed learning algorithm is capable of computing accurate parameters with improved convergence features.

Keywords. Fuzzy Cognitive Maps, learning algorithm, convergence.

[*] Corresponding author: gonzalo.napoles@uhasselt.be

I. Introduction

Fuzzy Cognitive Maps (FCM) are Recurrent Neural Networks for modeling dynamical systems using causal relations [1]. Essentially, a FCM involves an information network where graph nodes represent objects, states, concepts or entities of the investigated system and they comprise a precise meaning for the problem domain. These concepts are equivalent to neurons in neural models, and they are connected by causal relationships that take values in the range $[-1,1]$. These elements interact during the inference stage to update the activation value of each neuron by using a rule similar to the standard McCulloch-Pitts schema [2]. This updating procedure is iteratively repeated until (i) the FCM-based system converges to a fixed-point attractor or (ii) a maximal number of iterations is reached. The former implies that a hidden pattern was discovered [3] whereas the latter suggests that the system responses are cyclic or completely chaotic.

The non-stable configurations are mostly related with the causal weight matrix that describes the whole system. More explicitly, a perfectly symmetric weight matrix implies the existence of large number of positive cycles in the modeled system. These cycles provide the system with positive feedback loops that amplify any initial change and thus lead to exponential growth or decline [4]. On the other hand, antisymmetric causal weight matrixes imply the existence of negative cycles with odd number of connections, providing the FCM with negative feedback loops that counteract any stimulus. Thus, after time period equal to the length of the cycle the neuron to which the initial change was introduced will receive an influence that has an opposite sign from the initial change. This leads the system to periodic behavior and the creation of limit cycles.

Such weights can be provided by domain experts or automatically computed from historical data by using a learning algorithm. Existing learning methods can be grouped into two large groups: Hebbian-based and population-based algorithms [5]. The first ones only require a single instance to adjust the model, however, numerical experiments reported by Papakostas et al. [6] have shown that population-based learning algorithms are preferred when developing FCM-based classifiers. Unfortunately, these algorithms do not include any convergence feature into their learning scheme and therefore, estimated parameters could induce non-stable behaviors.

Another challenging research field is related to the development of accurate FCM-based classifiers since they often show lower prediction rates regarding to traditional classifiers (e.g., decision trees, neural networks, support vector machines). However, in contrast to FCM-based models, traditional classifiers perform like *black-boxes* and therefore they are difficult to interpret. Roughly speaking, a FCM-based classifier can work in two types of architectures [6]:

- ***Class-per-output architecture***. Each decision class is mapped as an output neuron. During the exploitation of the FCM-based classifier, the predicted decision class corresponds to the output neuron with the highest activation value.
- ***Single-output architecture***. Each decision class is enclosed into the activation space of the decision neuron. By doing so, two possibilities have been identified:
    a) *Using a clustering approach*. During the training phase, the center of each cluster is determined and labeled. In the testing phase, the center having the closest distance to the projected activation value is assigned to the input pattern.
    b) *Using a thresholding approach*. During the training phase, a pair of thresholds for each decision class are determined. In the testing phase, the interval comprising the projected activation value is assigned to the input pattern.

From the best of our knowledge, only a few studies addressing the convergence on FCM-based classifiers have been proposed. For example, Boutalis et al. [7] and Kottas et al. [8] investigated the existence and uniqueness of equilibrium values of neurons in FCM equipped with sigmoid transfer functions, using the contraction mapping theorem. Knight et al. [9] proposed a slightly different theoretical result related with the inclination of the sigmoid function. However, Nápoles et al. [10] numerically verified that these theoretical results cannot be directly used in solving pattern classification problems since a FCM-based classifier with a single fixed point-attractor will produce the same decision class for all input patterns.

In this paper we introduce a population-based learning algorithm that attempts to compute accurate parameters (i.e., the causal weights that define the interaction among map neurons, and the sigmoid inclination of each transfer function) having convergence features. It implies that the FCM-based classifier must be capable of effectively recognizing the input patterns in a stable fashion, that is, reducing the variability on the responses for consecutive iterations. To accomplish that, we extend the basic principle of a heuristic algorithm called Stability based on Sigmoid Functions (SSF) that allows improving the convergence of FCM-based classifiers [10] [11]. It should be mentioned that the proposed learning algorithm provides high flexibility and allows computing the parameters of FCM-based classifiers having different decision architectures.

The rest of the paper is organized as follows: in Section II the background about the FCM theory is provided, whereas in Section III we describe the SSF algorithm. In Section IV we introduce the proposed algorithm to compute the causal weights and the sigmoid parameters in a stable fashion, including some important definitions and theorems. Section V provides numerical simulations that allow evaluating our learning methodology across six FCM-based classifiers, whereas in the last section we discuss relevant remarks and further research aspects.

II. Fuzzy Cognitive Maps

Essentially a FCM is a fuzzy digraph that describes the behavior of an intelligent system in terms of concepts. Each concept represents an object, a state, a variable or a characteristic of the system under investigation [12]. These concepts (or neurons) define a set $\mathcal{C} = \{C_1, C_2, \dots, C_M\}$ where M is the number of neurons in the weighted graph. On the other hand, the relation between two neurons C_i and C_j is defined by a function $\mathcal{W}: (C_i, C_j) \rightarrow w_{ij}$ where $w_{ij} \in [-1,1]$. It allows characterizing the interaction between two map neurons by using causal relations.

The reasoning rule of a FCM computes the activation value of a map neuron C_i at each iteration step by using a function $\mathcal{A}: (C_i) \rightarrow A_i^{(t)}$. The activation degree of a neuron plays a relevant role during the FCM interpretation: the higher the activation value of a neuron, the stronger its influence over the system [13]. Moreover, a transfer function $f: \mathbb{R} \rightarrow I$ is used to keep the activation value of neurons in the interval $I = [0,1]$ or $I = [-1,1]$. The three most widely used transfer functions include: the bivalent, the trivalent and the sigmoid variants [14]. Equation (1) displays how these elements interact to iteratively compute the activation vector $A^{(t)} = \left(A_1^{(t)}, A_2^{(t)}, \dots, A_M^{(t)}\right)$ using the state vector $A^{(0)} = \left(A_1^{(0)}, A_2^{(0)}, \dots, A_M^{(0)}\right)$ as the initial stimulus.

$$A_i^{(t+1)} = f\left(\sum_{j=1}^{M} w_{ji}\, A_j^{(t)} + w_{ii} A_i^{(t)}\right), i \neq j \qquad (1)$$

In the same way to other recurrent models - such as the Hopfield model - at each discrete time a new state vector is produced [15]. After a large enough number of cycles T and depending on the transfer function adopted, the map will arrive in one of the following states:

- **Fixed-point** ($\exists t_\alpha \in \{1,2,\dots,(T-1)\} : A^{(t+1)} = A^{(t)}, \forall t \geq t_\alpha$): the system will produce the same output after the cycle t_α, so $A^{(t_\alpha)} = A^{(t_\alpha+1)} = A^{(t_\alpha+2)} = \cdots = A^{(T)}$.
- **Limit cycle** ($\exists t_\alpha, P \in \{1,2,\dots,(T-1)\} : A^{(t+P)} = A^{(t)}, \forall t \geq t_\alpha$): the map will produce the same output periodically after the cycle t_α, so $A^{(t_\alpha)} = A^{(t_\alpha+P)} = A^{(t_\alpha+2P)} = \cdots = A^{(t_\alpha+jP)}$ where obviously $t_\alpha + jP \leq T$, such that $j \in \{1,2,\dots,(T-1)\}$.
- **Chaos**: the system continues to produce different state vectors for successive cycles. In such cases the FCM is unable to converge, leading to confusing system responses.

It should be commented that the transfer function plays a pivotal role in the convergence of FCM-based systems. Discrete (bivalent or trivalent) FCM always converge to a fixed-point attractor or limit cycle since FCM are deterministic models, and so, the number of distinct states is finite [16]. In contrast, FCM equipped with continuous transfer functions (e.g., sigmoid FCM) can exhibit chaotic behaviors since the FCM could produce infinite different states freely distributed in the space defined by the $[-1,1]^M$ hypercube. In spite of this fact, Bueno and Salmeron [14] concluded that the sigmoid transfer function has a superior predictive capability.

III. Stability based on Sigmoid Functions

In this section we describe a heuristic procedure called Stability based on Sigmoid Functions (SSF) for non-discrete FCM-based systems [10] that allows improving the system convergence without altering the weights configuration. The original SSF algorithm is focused on FCM-based classifiers using a class-per-output architecture, but it could be extended to other models as will be illustrated in Section IV. The foundations of this algorithm emerged from empirical simulations, where the authors observed that using a different sigmoid function for each neuron (instead of using the same transfer function for all map neurons) the convergence of the FCM-based classifier suffered some

changes. Being more explicit, we noted a significant reduction of the system entropy in some cases, while in others, the convergence was seriously affected. From such experiments we concluded that variations on the parameter λ_i in Equation (2) lead to significant changes (positive or negative) on the FCM behavior. This suggests that a learning procedure could improve the stability properties of the FCM-based classifier, without altering its capability for predicting new patterns (i.e., without changing the weight configuration that has been previously estimated).

$$f_i(A_i, \lambda_i) = 1/\left(1 + e^{-\lambda_i(A_i - 0.5)}\right) \tag{2}$$

Based on the above assumption, Nápoles et al. [10] developed a heuristic method that reduces the variability on the system responses, without affecting the system ability to recognize new patterns. To accomplish that, this algorithm estimates a family of sigmoid functions $\{f_1(A_1),, \dots, f_M(A_M)\}$ where the ith sigmoid function will be used for transforming the activation value of the ith neuron. This is equivalent to compute the sigmoid inclination $\lambda_i > 0$ for each transfer function $f_i(A_i, \lambda_i)$. Equation (3) shows the modified neural inference rule used in this learning methodology where the activation value of the ith neural processing entity is influenced by the free interaction of the connected neurons and also by its steepness parameter $\lambda_i > 0$.

$$A_i^{(t+1)} = f_i\left(\sum_{j=1}^{M} w_{ji}\, A_j^{(t)} + w_{ii} A_i^{(t)}, \lambda_i\right), i \neq j \tag{3}$$

From the learning point of view, we must estimate a parameter $\lambda_i > 0$ for each neuron C_i that minimizes the difference between two consecutive responses. If this value exists, then the system will produce similar (even identical) output vectors for each initial pattern $A^{(0)}$. Equation (4) shows the objective function to be minimized in order to improve the stability of the ith neuron, where K is the number of training patterns, whereas T is the maximal number of iterations used during the

inference. In this equation $A_{ik}^{(t)}$ is the activation degree of the neuron C_i at each cycle for an initial condition, which is codified from the kth training pattern.

$$min \rightarrow \phi_i(f_i) = \sum_{k=1}^{K} \sum_{t=2}^{T} \left| A_{ik}^{(t)} - A_{ik}^{(t-1)} \right| \tag{4}$$

Equation (5) generalizes the above reasoning for all neurons. Notice that the learning procedure minimizes the absolute difference between the numerical responses for two consecutive discrete-time steps, for all patterns stored in the training dataset.

$$min \rightarrow \phi(f_1, f_2, f_3, \dots, f_M) = \sum_{k=1}^{K} \sum_{i=1}^{M} \sum_{t=2}^{T} \left| A_{ik}^{(t)} - A_{ik}^{(t-1)} \right| \tag{5}$$

The search space of the optimization problem is defined by I^M where $I = [0.1,25]$ is the domain of the steepness factors λ_i. It means that a solution X can be codified as a M-dimensional vector $X = (\lambda_1, \lambda_2, \lambda_3, \dots, \lambda_M)$ which produces a family $\{f_i\}_{i=1}^{M}$ of sigmoid functions. It should be noticed that we could use any continuous optimizer (e.g., Evolutionary Algorithms) for solving the related optimization problem. In our study we adopt an heuristic search approach since population-based metaheuristics are capable of finding near-optimal solutions in a reasonable execution time, thus ignoring analytical properties of the target function (e.g., convexity, continuity, differentiability or gradient information). In the next section we present the learning method for computing the causal weights and the sigmoid parameters in FCM-based classifiers.

IV. The proposed learning algorithm

In this section we formulate a new learning algorithm with convergence features for FCM-based classifiers having a single decision neurons. This methodology is based on the theoretical study

conducted by Nápoles et al. [11] that corrects several drawbacks on the foundations of the original SSF procedure. After introducing the proposed learning methodology, we first formalize general concepts and definitions regarding to pattern recognition problems that will be adopted in the rest of the proposal (i.e., pattern classification problem, FCM-based classifiers, and decision classes in FCM-based classifiers using a thresholding decisions scheme).

Definition 1. Let us assume a pattern recognition problem $\wp$ described by N numerical variables and a discrete decision variable. The solution for $\wp$ is equivalent to compute a model $\Omega: \mathbb{R}^N \rightarrow \mathcal{D}$, where $\mathcal{D}$ is the set of all decision classes, from a set of training examples that comprises instances of the problem $\wp$ previously associated to the decision class.

Definition 2. Let us suppose a FCM-based system $\mathcal{M}$ comprising a set of M neural processing entities $\mathcal{C} = \{C_1, \ldots, C_N, C_{N+1}, \ldots, C_M\}$ that can be organized in two subsets $\mathcal{I} \subset \mathcal{C}$ and $\mathcal{O} \subset \mathcal{C}$, such that $\mathcal{I} \cap \mathcal{O} = \emptyset$. The subset $\mathcal{I} = \{C_1, C_2, \ldots, C_N\}$ includes the *input neurons*, whereas the subset $\mathcal{O} = \{C_{N+1}, C_{N+2}, \ldots, C_M\}$ involves the *output neurons*. We say that $\mathcal{M}$ is a FCM-based classifier if we can reconstruct the model $\Omega: \mathbb{R}^N \rightarrow \mathcal{D}$ from a mapping $\mathcal{M}: [0,1]^N \rightarrow [0,1]^{M-N}$ that encloses the activation value of input and output neurons, respectively.

The above definition provides a general framework for FCM-based classifiers. For example, let us suppose a class-per-output architecture with N input neurons, and $N - M$ output decision classes. Assuming that all features are numerical, the set of neurons $\mathcal{C} = \{C_1, \ldots, C_N, C_{N+1}, \ldots, C_M\}$ can be activated by using a min-max normalization where the excitation of the ith neuron according to the kth instance Y_k is given by $A_{ik}^{(0)} = (Y_{ki} - min(F_i))/(max(F_i) - min(F_i))$ where $min(F_i)$ and $max(F_i)$ denote the minimum and maximum value for the problem feature F_i, respectively. In the second step, the mapping $\mathcal{M}: [0,1]^N \rightarrow [0,1]^{M-N}$ corresponds to the updating rule that computes

the activation value of output neurons. Finally, the mapping $\psi_{(\mathcal{M})}: [0,1]^{M-N} \rightarrow \mathcal{D}$ decides the class label as $\psi_{(\mathcal{M})}(\mathcal{A}^T(\mathcal{O})) = argmax_{N+i}\left\{A_{N+i}^{(T)}\right\}$, $i \in \{1, \dots, M-N\}$, where $A_{N+i}^{(T)}$ is the activation value of the $(N+i)$th output neuron at the last iteration step.

In this research, we conducted our experimentation to FCM-based classifiers using a thresholding single-output architecture since Papakostas et al. [6] empirically concluded that this approach has a superior predictive capability. Definition 3 mathematically formalizes the concept of *decision classes* in FCM-based classifiers using a single-output architecture.

Definition 3. Let us assume that $\mathcal{M}$ is a FCM-based classifier using a thresholding single-output architecture (i.e., FCM-based classifier with a single decision neuron). A decision class $d_j \in \mathcal{D}$ is a closed partition of the activation space of the decision neuron C_M bounded by a lower threshold L_j and an upper threshold U_j, such that the class rank $U_j - L_j > 0$ and $\left[L_j, U_j\right] \cap [L_h, U_h] = \emptyset$, $\forall h \in \{1, \dots, |\mathcal{D}|\}/\{j\}$. In this model, the mapping $\psi_{(\mathcal{M})}: [0,1] \rightarrow \mathcal{D}$ determines the predicted class label by allocating the activation value into the corresponding interval.

The learning goal of our model is to estimate the parameters that define the performance of FCM-based classifier, that is, the weights and the sigmoid parameters. It means that a candidate solution for the optimization problem must involve a weight matrix and a family of sigmoid functions that ensure high prediction rates and acceptable convergence features. Therefore, a candidate solution comprises a $(M^2 + M)$-dimensional vector, assuming a fully connected network with M neurons. This suggests that the population-based optimizer (e.g., Evolutionary Algorithm or Particle Swarm Optimization) must compute solutions with the following structure:

$$X = [w_{11}, \dots, w_{1M}, w_{21}, \dots, w_{2M}, \dots, w_{M1}, \dots, w_{MM}, \lambda_1, \lambda_2, \dots, \lambda_M]$$

In this paper we adopted the Particle Swarm Optimization (PSO) search method [17] for generating the solutions since PSO-based algorithms have proven to be competent for solving real-parameter optimization tasks, however, other optimizers could be used as well.

Equation (6) shows the error function to be optimized, where X is the candidate solution generated by the selected optimizer, K denotes the number of training patterns, $0 \leq F(.) \leq 1$ is a function that computes the prediction error achieved by the classifier, whereas $0 \leq H(.) \leq 1$ represents the accumulated convergence error during updating the activation value of neurons. On the other hand, the parameters $\alpha_1, \alpha_2 \in [0,1]$ establish the importance of the classifier's accuracy regarding to the FCM stability. In this learning scheme, $\alpha_1 + \alpha_2 = 1$ guarantees that the error function is always confined into the interval $[0,1]$. We strongly suggest that $\alpha_1 \gg \alpha_2$ since the key goal of the proposed methodology is the prediction accuracy, even if the system is unable to produce the same decision class for successive discrete-time steps. We may confidently assume that $\alpha_2 = 1 - \alpha_1$ due to the fact that $\alpha_1 + \alpha_2 = 1$, which leads to a simpler parametrized model.

$$min \rightarrow E(X) = \alpha_1 G(X) + \alpha_2 H(X) \tag{6}$$

Equation (7) shows a basic strategy for estimating the prediction error by computing the number of misclassified patterns regarding to the cardinality of the sample. In practice, the function $F(.)$ quantifies the differences between the expected decision class S_k associated with the kth training pattern, and the decision class predicted by the decision model $\psi_{(X)}\left(A_{Mk}^{(T)}\right)$. The decision model is defined from the candidate weights configuration and the activation value of the decision neuron at the last iteration step $A_{Mk}^{(T)}$. Observe that the responses at the previous discrete-time steps are not considered since they are not used when computed the predicted class, instead, such responses are evaluated when analyzing the convergence of the FCM-based classifier.

$$G(X) = \frac{1}{K}\sum_{k=1}^{K}\begin{cases} 0, \psi_{(X)}\left(A_{Mk}^{(T)}\right) = S_k \\ 1, \psi_{(X)}\left(A_{Mk}^{(T)}\right) \neq S_k \end{cases} \tag{7}$$

Equation (8) formalizes the function $H(X)$ used for computing the accumulated convergence error during updating the activation value of neurons. This function quantify the dissimilarity between the activation vector at each discrete-time step and the activation vector produced at the last cycle, where M is the number of neurons and T is the number of discrete-time cycles. This is equivalent to calculate the differential $\left|A_{ik}^{(t)} - A_{ik}^{(T)}\right|$ for each sigmoid neuron, using the kth training pattern as the initial stimulus. Moreover, our learning algorithm assumes that each iteration step has different significance for the convergence and prediction of the FCM-based classifier. In this research, the weighting parameter ω_t is linearly increased with time (i.e., iteration steps).

$$H(X) = \sum_{k=1}^{K}\sum_{i=1}^{M}\sum_{t=1}^{T}\frac{2\omega_t\left(A_{ik}^{(t)} - A_{ik}^{(T)}\right)^2}{KM(T+1)} \tag{8}$$

More explicitly, in the FCM-based classifiers defined above, a decision class is a closed partition of the activation space of this neuron, and so numerical responses could embrace the same decision class. It should be noticed that this learning algorithm can be used for adjusting the parameters in FCM-based classifiers with several architectures (e.g., class-per-output or single-output using either a thresholding or clustering approach). In such cases, the proposed heuristic learning scheme will be likely to compute a *slightly non-stable* FCM as the worst scenario.

Definition 4. Let us assume that $\mathcal{M}$ is a FCM-based classifier where $\psi_{(\mathcal{M})}: [0,1]^{M-N} \rightarrow \mathcal{D}$ is the decision model, whereas $\mathcal{A}^{(t)}(\mathcal{O}) = \left(A_{N+1}^{(t)}, \ldots, A_{N+i}^{(t)}, \ldots, A_{M}^{(t)}\right), i \in \{1, \ldots, M-N\}$, is the activation vector for output-type neurons at each discrete-time step $t \in \{1,2, \ldots, T\}$. We say that the FCM-based classifier $\mathcal{M}$ is *slightly non-stable* if $\nexists t_\alpha \in \{1,2,\ldots,(T-1)\} : A_{N+i}^{(t_\alpha)} = A_{N+i}^{(t_\alpha+1)}$, however, $\exists t_\alpha \in \{1,2,\ldots,(T-1)\} : \psi_{(\mathcal{M})}\left(\mathcal{A}^{(t_\alpha)}(\mathcal{O})\right) = \psi_{(\mathcal{M})}\left(\mathcal{A}^{(t_\alpha+1)}(\mathcal{O})\right), \forall t \geq t_\alpha$.

Another issue to be discussed is the definition of the search space. In the case of dimensions related to causal weights, the search space is defined as $I_1 = [-1,1]$ since in FCM-based classifiers the causality could be either negative or positive. In the cases of dimensions related with the sigmoid parameters the search space must be carefully defined in order to avoid situations on which the system only produces a single decision class. Knight et al. [9] proved that if $\lambda > 0$ is small enough then there is a unique fixed-point attractor. On the contrary, if $\lambda > 0$ is large enough, then there can be multiple fixed-points, where many of such equilibrium points may be linearly stable (see next Theorem 1). However, Nápoles et al. [10] numerically verified that a FCM-based classifier having a single fixed-point attractor will produce the same decision class for all input patterns. It suggests that the theorem cannot be used in pattern recognition scenarios, but in control scenarios where the system requires to be consistent to external perturbations.

Theorem 1. The number of solutions of Equation (1) depends on the size of λ:

- If $\lambda > 0$ is small enough then there is a unique solution. This fixed point of the sigmoidal FCM is linearly stable.
- If $\lambda > 0$ is large enough then there can be multiple solutions, where many of these fixed points may be linearly stable.

On the other hand, Knight et al. [9] introduced a theorem where the upper bound $\bar{\lambda}(M)$ for "small enough" values of λ is estimated (see Theorem 2). Therefore, if $0 \leq \lambda < \bar{\lambda}(M)$ then the system will produce the same decision class regardless the input pattern. It suggests that the selected optimizer must produce candidate solutions with sigmoid parameters $\lambda_i \geq \bar{\lambda}(M)$. This result implies that the FCM-based classifier could have several linearly stable fixed-points attractors.

Theorem 2. For $A \in \mathbb{R}^M \times \mathbb{R}^M$, the sigmoid FCM, has a unique fixed-point for all λ such that $0 \leq \lambda < \bar{\lambda}(M)$, this fixed point is stable. $\bar{\lambda}(M)$ satisfies (9) where B_i^M are the binomial coefficients, and b_i given by the recursion relation $b_i = ib_{i-1} + (-1)^i$, $b_0 = 1$.

$$\left(1 - \frac{\bar{\lambda}(M)}{4}\right)^M - \sum_{i=1}^{M} b_i B_i^M \left(\frac{\bar{\lambda}(M)}{4}\right)^i = 0 \tag{9}$$

Based on the above analysis, we can conclude that the solution space for dimensions associated to sigmoid parameters must be defined by the space $I_2 = [\bar{\lambda}(M) + 0.01, \bar{\lambda}(M) + \sigma]$, where M is the number of map neurons, while $\sigma > 0$ is the rank of the search space. In this research $\sigma = 10$ since sigmoid functions with larger inclinations tend to reach the behavior of discrete functions, which produce qualitative results. It could affect the prediction rate of the FCM-based classifier since the system could produce very similar responses for quite dissimilar inputs.

V. Numerical simulations

In this section we study the behavior of the proposed learning methodology using a real problem concerning the resistance mechanism of HIV-1 proteins to existing inhibitors. These FCM-based classifiers showed good classification rates by using historical data, however, the system stability cannot be ensured and therefore new mutations could be misclassified.

A. Description of the FCM-based models used for evaluation

Recently, Nápoles et al. [18] introduced a FCM-based model for predicting the resistance of new HIV-1 *protease* mutations to existing inhibitors. The *protease* sequence is defined by a chain of 99 amino acids, however, with the goal of reducing the model complexity only positions related with drug resistance were used [19]. Such sites were biologically determined from clinic assays in infected patients and they allowed an averaged reduction rate of 80% regarding the total number of sequence positions. In the proposed topology sequence positions related with drug resistance were taken as input neurons, whereas a decision neuron for the resistance feature was also defined with the goal of computing the resistance class for each input pattern. This FCM-based classifier embraces two kinds of causal relations, which are in correspondence with the biological system. Direct relationships connect input neurons with the resistance feature, while indirect relationships establish connections between all sequence positions.

Based on the above topology, Nápoles et al. [18] obtained six FCM-based classifiers where each map represents the protein behavior for the following drugs: Amprenavir (APV), Indinavir (IDV), Saquinavir (SQV), Nelfinavir (NFV), Ritonavir (RTV) and Atazanavir (ATV). Each inhibitor has associated a high-quality filtered dataset [20] comprising reported mutations and their resistance value, where the amino acids are encoded according to their contact energy [21]. Therefore, each training pattern comprises the activation value of the N input neurons, and the expected resistance class for the inhibitor (i.e., 0-*susceptible* and 1-*resistant*).

In this single-output architecture, the predicted decision class for each input pattern is computed from the activation value of the decision neuron at the last iteration step, and a biologically defined threshold. More explicitly, input sequences having resistance degree below the resistance threshold

will be labeled as *susceptible*, otherwise the mutation will be labeled as *resistant*. Equation (10) formalizes the decision model used in the single-output classifier, where ξ denotes the biologically determined threshold. Note that we need to change this decision model if the architecture changes, however, the learning algorithm discussed in this paper holds.

$$\psi_{(\mathcal{M})}\left(A_{Mk}^{(T)}\right) = \begin{cases} 0, & A_{Mk}^{(T)} \leq \xi \\ 1, & A_{Mk}^{(T)} > \xi \end{cases} \tag{10}$$

In order to compute the causal weights, Nápoles et al. [18] used a standard learning algorithm that allows reducing number of misclassified patterns. This is equivalent to minimize the function $G(.)$ for all patterns stored in the training set. Simulations reported by Grau et al. [22] have shown that the adjusted FCM-based classifiers have high prediction rates, notably outperforming other well-known classifiers. However, in most cases the stability is comprised.

B. Evaluation of the proposed learning algorithm

In this section we evaluate the proposed learning algorithm by using the six FCM-based systems described above. As mentioned, to minimize the error function (6) we adopt the constricted PSO Type-1 introduced by Clerk and Kennedy [23] as numerical optimizer, where the number of swarm particles is set to 80, the inertia weight $\omega = 0.7298$ and $c_1 = c_2 = 1.496$. The learning algorithm will stop when a maximal number of cycles $NC = 200$ is reached, or alternatively when a fixed-point attractor is discovered. As a final point, the parameter $\alpha_1 = 0.8$ and $\alpha_2 = 1 - \alpha_1$, while the maximal number of iterations steps in the inference rule is set to 100. To perform this experiment, we have used the FCM WIZARD software [24] since it provides several experimentation facilities. Actually, we included the proposed learning algorithm into this software.

Aiming at highlighting the difficulty of the six problems used for validation, Table 1 summarizes the features of each learning system. In this table, features correspond to the number of sequence positions associated with drug resistance for each drug, thus the number of relations in the network is equivalent to the number of weights to be estimated. Notice that the number of parameters to be estimated by the learning algorithm (i.e., dimensionality of the optimization problem) is equal to $(N^2 + N + 1)$ where N denotes the number of input neurons.

Table 1. Features of each learning problem adopted for evaluation.

Inhibitor	Patterns	Features	Relations	Parameters	Threshold	Accuracy
APV	96	19	361	381	0.007	0.95
IDV	137	17	289	307	0.006	0.99
SQV	139	17	289	307	0.003	0.95
NFV	204	12	144	157	0.005	0.95
RTV	151	14	196	211	0.004	0.97
ATV	69	15	225	241	0.004	0.96

Figures 1, 2, 3 show the behavior of the proposed learning algorithm, regarding to the approach adopted by Nápoles et al. [18]. Being more explicit, in these figures we show the activation value of the decision neuron (i.e., the system response) for each iteration step. The dashed line represents the numerical response after optimizing the FCM parameters using the standard approach, whereas the solid line denotes the numerical response after optimizing the FCM parameters using the model presented in this paper. From these simulations we can conclude that our learning model is capable of computing FCM-based classifiers having superior convergence features regarding the standard approach. It implies that the proposed learning method allows reducing the overall variability on the activation values of the decision neuron. Furthermore, we observed that the model shows better convergence rate for those problems that converge to a fixed-point attractor regardless the learning scheme, so lower number of discrete-time steps are required.

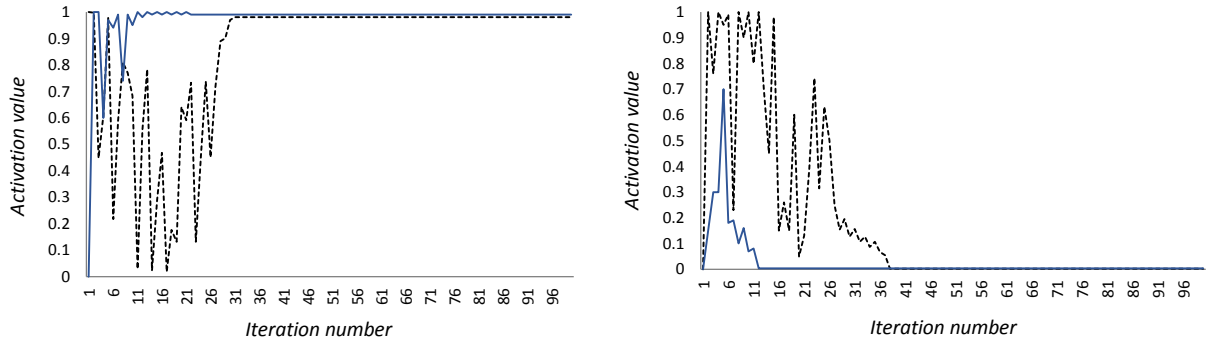

Figure 1. Activation value of the decision neuron each iteration step for two stable configurations. The dashed line denotes the numerical response after applying the standard learning approach, whereas the solid line represents the numerical response after applying the proposed learning algorithm.

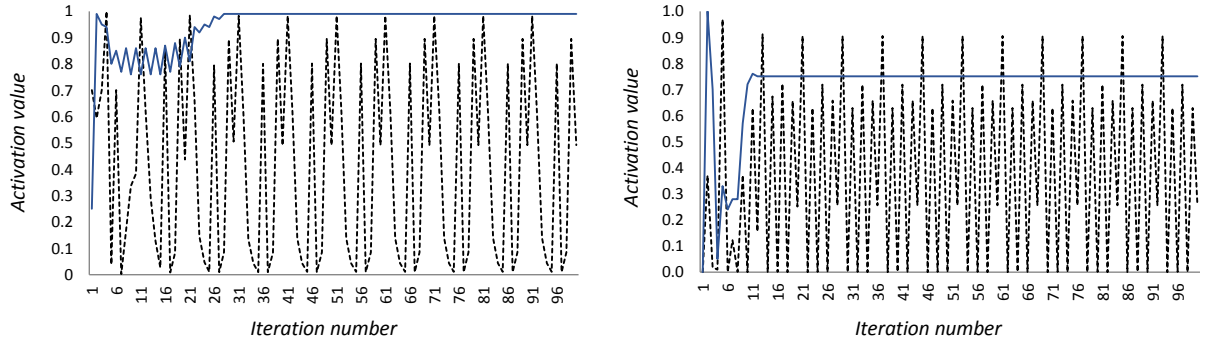

Figure 2. Activation value of the decision neuron each iteration step for two cyclic configurations. The dashed line denotes the numerical response after applying the standard learning approach, whereas the solid line represents the numerical response after applying the proposed learning algorithm.

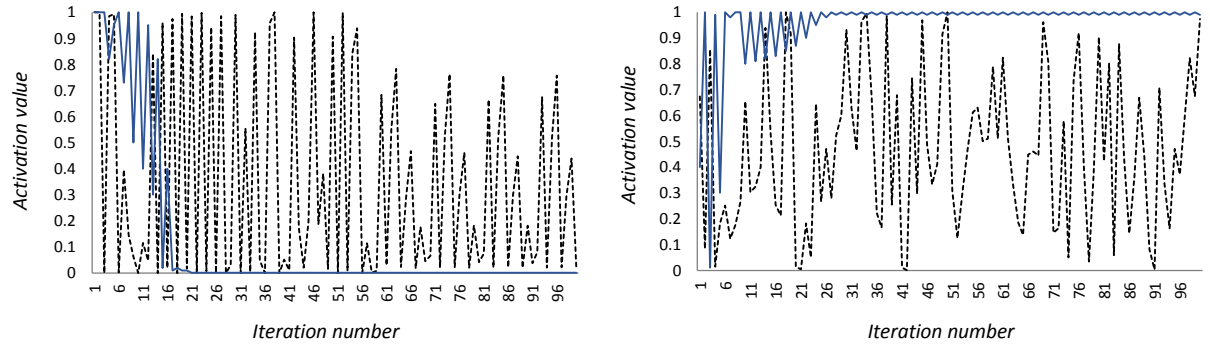

Figure 3. Activation value of the decision neuron each iteration step for two chaotic configurations. The dashed line denotes the numerical response after applying the standard learning approach, whereas the solid line represents the numerical response after applying the proposed learning algorithm.

The following experiment is oriented to determine whether the improvements in the convergence of the FCM-based classifier are statistically significant. By doing so, we compute Equation (11) for each pattern φ_k belonging to training set according to (i) the map $\mathcal{M}_1$ obtained after applying the standard learning algorithm that only considers the prediction rates, and for (ii) the map $\mathcal{M}_2$ obtained after applying the learning algorithm discussed in this study. Therefore, we achieve two ordered sets $\Gamma^1 = \{\Gamma_1^1(\mathcal{M}_1, \varphi_1), \ldots, \Gamma_K^1(\mathcal{M}_1, \varphi_K)\}$ and $\Gamma^2 = \{\Gamma_1^2(\mathcal{M}_2, \varphi_1), \ldots, \Gamma_K^2(\mathcal{M}_2, \varphi_K)\}$ for each dataset. It should be mentioned that, due to the stochastic nature of our learning method, each entry $\Gamma_k(\mathcal{M}, \varphi_k)$ is calculated from the average of 10 independent trails.

$$\Gamma_k(\mathcal{M}, \varphi_k) = \sum_{i=1}^{M} \sum_{t=1}^{T} \frac{\omega_t \left(A_{ik}^{(t)} - A_{ik}^{(T)} \right)^2}{M(T+1)} \tag{10}$$

Table 2 summarizes the p-values achieved by Wilcoxon signed rank test [25], using a significance degree $\alpha = 0.05$, which corresponds with the 95% confidence interval. The Wilcoxon signed rank test is a nonparametric method employed in hypothesis testing situations, involving a design with two samples. This pairwise test allows spotting significant differences between two sample means, that is, the behavior of two algorithms [26]. According to the results, the test suggests rejecting the null hypothesis (p-value < 0.05) for drugs RTV, IDV, NFV and SQV. In the case of APV and ATV, the conservative hypothesis is accepted (p-value > 0.05) although in such problems the sum of negative ranks (R^-) is lower than the sum of positive ranks (R^+). Moreover, Table 2 reports the mean and standard deviation achieved for the convergence measure after running 10 independent trials of the proposed algorithm. The reader may observe that the learning methodology is capable of estimating high-quality solutions with small standard deviation.

Table 2. Results achieved by the Wilcoxon test.

	Patterns	**Mean**	R^-	R^+	p**-value**	**Hypothesis**
APV	96	$0.06094\ (\pm 1.6E-3)$	61	35	0.069	Accepted
IDV	137	$0.01521\ (\pm 2.2E-4)$	111	26	0.000	Rejected
SQV	139	$0.04470\ (\pm 8.8E-4)$	117	22	0.000	Rejected
NFV	204	$0.04954\ (\pm 1.0E-3)$	159	45	0.000	Rejected
RTV	151	$0.05203\ (\pm 2.0E-3)$	123	28	0.000	Rejected
ATV	69	$0.07755\ (\pm 1.9E-3)$	39	30	0.081	Accepted

The above results confirm that our model is capable of producing configurations that lead to FCM-based classifiers having superior convergence features. This is actually expected since as far as we know there is no population-based learning procedure including the system convergence into the learning scheme. More importantly, during simulations we observed that our learning method was capable of producing the same classification accuracy (see Table 1) for APV, IDV, RTV and ATV, while for the remaining inhibitors it achieved better results. The prediction rates for SQV and NFV are 0.96 and 0.97, respectively. These positive outcomes are unexpected but totally consistent with our learning scheme since it is well-known that the convergence is closely related with the ability of the FCM-based classifier for recognizing new patterns.

It should be mentioned that we could use another information theoretic coefficient to measure the divergence or disparity between two probability densities. A widely accepted divergence measure is the Kullback–Leibler divergence [27] between the true model and the approximating candidate model. This measure and its variants [28] have been used in developing learning rules for artificial neural networks [29][30] and could be adapted to the semantic of fuzzy cognitive mappers used in pattern classification environments. For example, we could use a learning rule minimizing a) the Kullback–Leibler divergence between the expected responses and the approximate outputs, and b) the Kullback–Leibler divergence between the response at each discrete-time step and the produced

output. Of course, the recurrent nature of FCM-based classifiers makes this integration challenging and leads to new research avenues to be explored as a future work.

VI. Conclusions

This paper presents a population-based learning algorithm for FCM-based classifiers. It attempts computing the required parameters (i.e., the causal weights that define the interaction between map neurons, and the sigmoid inclination of each transfer function) leading to high prediction rates and improved convergence features. Numerical simulations have shown that our learning methodology is able of computing high-quality FCM-based classifiers, that is, systems that effectively recognize new input patterns in a stable fashion. On the other hand, we observed several scenarios on which our proposal achieved better prediction rates, with regard to the standard learning scheme that only considers the classification accuracy into its learning goal. From the best of our knowledge, there is no learning algorithm allowing the convergence of the FCM-based classifier, without affecting the classification rates. As a future work, the authors will be focused on hybridizing the proposed learning algorithm with existing learning rules for neural networks.

Acknowledgments

The authors would like to thank PhD Student Isel Grau from Free University of Brussels, Belgium, for her valuable help on implementing the proposal into FCM WIZARD. As well, we would like to thank reviewers for their constructive suggestions during the revision process.

References

[1] B. Kosko, "Fuzzy cognitive maps," *Int. J. Man. Mach. Stud.*, vol. 24, no. 1, pp. 65–75, 1986.

[2] W. McCulloch and W. Pitts, "A logical calculus of the ideas immanent in nervous activity," *Bull. Math. Biophys*, vol. 7, pp. 115–133, 1943.

[3] B. Kosko, "Hidden patterns in combined and adaptive knowledge networks," *Int. J. Approx. Reason.*, vol. 2, no. 4, pp. 377–393, 1988.

[4] A. K. Tsadiras and K. G. Margaritis, "An experimental study of the dynamics of the certainty neuron fuzzy cognitive maps," *Neurocomputing*, vol. 24, no. 1, pp. 95–116, 1999.

[5] E. I. Papageorgiou, "Learning Algorithms for Fuzzy Cognitive Maps - A Review Study," *IEEE Trans. Syst. Man, Cybern. - Part C Appl. Rev.*, vol. 42, no. 2, pp. 150–163, 2012.

[6] G. A. Papakostas, D. E. Koulouriotis, A. S. Polydoros, and V. D. Tourassis, "Towards Hebbian learning of Fuzzy Cognitive Maps in pattern classification problems," *Expert Syst. Appl.*, vol. 39, no. 12, pp. 10620–10629, 2012.

[7] Y. Boutalis, T. L. Kottas, and M. C. Christodoulou, "Adaptive Estimation of Fuzzy Cognitive Maps With Proven Stability and Parameter Convergence," *IEEE Trans. Fuzzy Syst.*, vol. 17, pp. 874–889, 2009.

[8] T. L. Kottas, Y. S. Boutalis, and M. A. Christodoulou, "Bi-linear adaptive estimation of Fuzzy Cognitive Networks," *Appl. Soft Comput.*, vol. 12, pp. 3736–3756, 2012.

[9] C. Knigh, L. D. Lloyd, and A. S. Penn, "Linear and Sigmoidal Fuzzy Cognitive Maps: an Analysis of Fixed Points," *Appl. Soft Comput.*, vol. 15, pp. 193–202, 2014.

[10] G. Nápoles, R. Bello, and K. Vanhoof, "How to improve the convergence on sigmoid Fuzzy Cognitive Maps?," *Intell. Data Anal.*, vol. 18, no. 6S, pp. S77–S88, 2014.

[11] G. Nápoles, E. Papageorgiou, R. Bello, and K. Vanhoof, "On the convergence of sigmoid Fuzzy Cognitive Maps," *Inf. Sci. (Ny).*, vol. 350, pp. 154–171, 2016.

[12] B. Kosko, *Neural Networks and Fuzzy systems, a dynamic system approach to machine intelligence*. Prentice-Hall, Englewood Cliffs, 1992.

[13] G. Nápoles, I. Grau, R. Pérez-García, and R. Bello, "Learning of fuzzy cognitive maps for simulation and knowledge discovery," in *Studies on Knowledge Discovery, Knowledge Management and Decision Making, EUREKA 2013*, 2013, pp. 27–36.

[14] S. Bueno and J. L. Salmeron, "Benchmarking main activation functions in fuzzy cognitive maps," *Expert Syst. Appl.*, vol. 36, no. 3, pp. 5221–5229, 2009.

[15] E. I. Papageorgiou and J. L. Salmeron, "A Review of Fuzzy Cognitive Maps Research During the Last Decade," *IEEE Trans. Fuzzy Syst.*, vol. 21, no. 1, pp. 66–79, 2013.

[16] A. K. Tsadiras, "Comparing the inference capabilities of binary, trivalent and sigmoid fuzzy cognitive maps," *Inf. Sci. (Ny).*, vol. 178, pp. 3880–3894, 2008.

[17] R. Poli, J. Kennedy, and T. Blackwell, "Particle Swarm Optimization – An overview," *IEEE Trans. Evol. Comput.*, vol. 1, pp. 37–57, 2007.

[18] G. Nápoles, I. Grau, R. Bello, and R. Grau, "Two-steps learning of Fuzzy Cognitive Maps for prediction and knowledge discovery on the HIV-1 drug resistance," *Expert Syst. with Appl.*, vol. 41,

pp. 821–830, 2014.

[19] V. A. Johnson, V. Calvez, H. F. Günthard, R. Paredes, D. Pillay, R. W. Shafer, A. M. Wensing, and D. D. Richman, "Update of the Drug Resistance Mutations in HIV-1: March 2013," *Top. Antivir. Med.*, vol. Special Co, no. March, pp. 6–14, 2013.

[20] S. Rhee, M. J. Gonzales, R. Kantor, B. J. Betts, J. Ravela, and R. W. Shafer, "Human immunodeficiency virus reverse transcriptase and protease sequence database," *Nucleic Acids Res.*, vol. 31, no. 1, pp. 298–303, 2003.

[21] S. Miyazawa and R. L. Jernigan, "Self-Consistent Estimation of Inter-Residue Protein Contact Energies Based on an Equilibrium Mixture Approximation of Residues," *PROTEINS Struct. Funct. Genet.*, vol. 34, pp. 49–68, 1999.

[22] I. Grau, G. Nápoles, and M. M. García, "Predicting HIV-1 Protease and Reverse Transcriptase Drug Resistance Using Fuzzy Cognitive Maps," in *CIARP 2013, Part II, LNCS 8259*, 2013, pp. 190–197.

[23] M. Clerc and J. Kennedy, "The Particle Swarm — Explosion , Stability , and Convergence in a Multidimensional Complex Space," *IEEE Trans. Evol. Comput.*, vol. 6, no. 1, pp. 58–73, 2002.

[24] "FCM WIZARD," 2016. [Online]. Available: www.fcmwizard.com.

[25] F. Wilcoxon, "Individual comparisons by ranking methods," *Biometrics*, vol. 1, pp. 80–83, 1945.

[26] J. Luengo, S. García, and F. Herrera, "A study on the use of statistical tests for experimentation with neural networks: Analysis of parametric test conditions and non-parametric tests," *Expert Syst. Appl.*, vol. 36, no. 4, pp. 7798–7808, 2009.

[27] S. Kullback and R. A. Leibler, "On information and sufficiency," *Ann. Math. Stat.*, vol. 22, pp. 76–86, 1951.

[28] A.-K. Seghouane and S.-I. Amari, "The AIC Criterion and Symmetrizing the Kullback–Leibler Divergence," *IEEE Trans. Neural Networks*, vol. 18, no. 1, pp. 97–106, 2007.

[29] J. Cid-Sueiro, J. I. Arribas, S. Urban-Munoz, and A. R. Figueiras-Vidal, "Cost Functions to Estimate A Posteriori Probabilities in Multiclass Problems," *IEEE Trans. Neural Networks*, vol. 10, no. 3, pp. 645–656, 1999.

[30] J. I. Arribas and J. Cid-Sueiro, "A Model Selection Algorithm for a Posteriori Probability Estimation With Neural Networks," *IEEE Trans. Neural Networks*, vol. 16, no. 4, pp. 799–809, 2005.