Neural Network Based Fuzzy Cognitive Map

Sima Sabahi [a,*], Paul M. Stanfield [b]

[a] Industrial and Systems Engineering Department, University of Florida, Gainesville, USA. Email: sima.sabahi@ufl.edu

[b] Industrial and Systems Engineering Department, North Carolina A&T State University, Greensboro, USA. Email: stanfiel@ncat.edu

Abstract

A Fuzzy Cognitive Map (FCM) is a powerful technique for modeling and analyzing complex systems. In this study, we propose a novel learning algorithm that, unlike existing FCM-based learning algorithms, ensures matching the desired system state by computing the otherwise "unexplained" biases in the model. Our learning algorithm considers both the whole system bias and the individual biases for each system factor (concept). We explore the impact of FCM structure and characteristics for the proposed algorithm and suggest interpretation of computed biases. Finally, we propose an FCM visualization technique which enables comparison between and deeper understanding of modeled systems. As FCMs offer a broader, quantifiable view of the causal relationships between factors, the approach used in this study provides insights into FCM modeling and application to real-world complex systems.

Keywords: Fuzzy Cognitive Maps, Neural Networks, Complex System Analysis

1. Introduction

Fuzzy Cognitive Maps (FCM), first introduced by Kosko (1986), are a powerful technique for modeling and analyzing complex systems. FCM has been used in a wide range of situations such as political developments (Taber, 1991), dolphin, shark, and fish dynamics (Dickerson and Kosko, 1994), organizational behavior and job satisfaction (Craiger et al., 1996), economic/demographics of nations (Schneider et al., 1998), ecological models (Özesmi and Özesmi, 2004), relationship management in airline service (Kang et al., 2004), information system evaluation (Sharif and Irani, 2006), diagnosis of obesity (Giabbanelli et al., 2012), social-ecological systems (Gray et al., 2015), sustainable banking (Ferreira et al., 2016), and risk analysis in the food industry (Rezaee et al., 2018). Using FCM, one can visually demonstrate the cause and effect relationships among important factors in a system where it is difficult to describe them with a traditional mathematical representation. FCMs have a network structure where nodes represent the concepts/system factors, and directed arcs represent the causal

[*] Corresponding author:
E-mail address: sima.sabahi@ufl.edu
Phone: (352) 273-1896 Cell: (336) 926-0051
Department of Industrial and Systems Engineering
University of Florida
303 Weil Hall, PO Box 116595
Gainesville, FL 32611

relationships among the concepts (Giabbanelli et al., 2012). FCMs have three main components that include concepts, relationships between the concepts, and relationship weights. Once constructed, FCMs undergo an iterative (forward analysis) process where concept values are updated based on input relationships, their weights, and source concept values until convergence or some other termination criterion is reached.

Consider an FCM that might be constructed to model the interdependent community health factors (concepts) of healthcare, education, employment, and public safety. That FCM model might be applied to several communities with varying levels of "health" as determined by associated factor measures. The value of a factor measure is a function of the overall community health, the impact of other factors as modeled by the FCM, and unique characteristics of that factor without that community not explained by the community health or FCM model.

One would desire an expanded version of the FCM where concept values converge to values representative of the modeled community. FCM convergence to a desired state is typically pursued through learning algorithms (Jamshidi et al., 2018). Learning algorithms minimize the prediction errors made by the forward analysis (Kim et al., 2008). Another reason for using learning algorithm for FCM is to update the knowledge obtained from decision-makers or historical data to get refined relationship weights (Papageorgiou, 2012) and increase the efficiency and robustness of FCMs (Papageorgiou et al., 2005).

Having the objective of optimizing the weight matrix for reaching the desired state, many learning algorithms have been proposed in the literature for learning FCMs. One drawback of the existing learning algorithms is that their learning process is mainly concentrated on the impact of relationship weights on the concept values, ignoring the influences of other external forces or circumstances that may affect the whole system. Additionally, such learning algorithms might allow relationship weights to change to illogical values. By considering "biases" which we define as system influencing factors outside of the relationship weights, our proposed learning algorithm matches concept values associated with the real-world complex system state.

Our training approach can refine the interrelationships in the FCM model and discover biases unique to a specific complex system. As a result, the approach is inherently able to produce matches for real-world concept values while maintaining coherent connection weights, and producing additional system insight based on learned bias values. In the context of FCM, our use of the term bias is based on the bias input in error backpropagation neural networks. Bias accounts for the circumstances that create a deviation from the expected concept outcome based on the interrelationship weights alone.

Having designed and validated the proposed learning algorithm, we conduct an experiment to discover the impact of the FCM structure on its behavior. In to the experiment, weights are analyzed in terms of size and bound, along with three more sets of features including initial concept values (system health), number of nodes, and level of connectedness. Results are consistent with the expected function of the algorithm, providing further validation. Finally, we consider a more content rich method of representing the resulting FCM that enables rapid visual comparison of similar systems.

This paper is organized as follows. In the next section, we first provide a background on FCM and its structure. In Section 3, we propose our learning algorithm. Section 4 represents the detail of our algorithm using an example, community health example. In Section 5, we present our empirical experimentation approach and the results. Finally Section 6 discusses our findings and the future directions.

2. Research Background

2.1 FCM Model and Structure

In an FCM model, concepts are the main factors/variables that impact a system and get initial fuzzy normalized values that might be obtained from experts, historical data, or a combination of both. The fuzzy values for concepts take values ranging from 0 to 1, where 0 indicates that the concept is not activated in the system while 1 says it is fully active. Equation (1) shows a vector of concepts which is defined as vector C where $C_i \in [0, 1]$, $i = 1, 2, \ldots, N$ and N is the total number of concepts.

$$C = [C_1, C_2, \ldots, C_N] \quad (1)$$

Relationships between the concepts create the feedback containing potential balancing or reinforcing causal loops depending on the interdependencies between the concepts. The intensity of interdependencies between the concepts determines the complexity of the system. The fuzzy values for the cause and effect relationship can take values ranging from -1 to 1. Where a negative number indicates concept *i* negatively affects concept *j* and a positive number shows an increase in the value of concept *i* increases the value of concept *j*. Having considered fuzzy values for the concepts and the cause and effect relationship among them, FCMs strengthen cognitive maps (Ferreira et al., 2016). Equation (2) shows a matrix of weights which is defined as matrix W where $w_{ij} \in [-1, 1]$, $i, j = 1, 2, \ldots, N$ represents the strength of the causal relationship between concept *i* and *j*. Because a concept rarely causes itself (Ferreira et al., 2016), in this study, we assume all the entries on the main diagonal of the weight matrix is equal to zero. Figure 1 shows an example of an FCM model.

$$W = \begin{bmatrix} w_{11} & \cdots & w_{1N} \\ \vdots & \ddots & \vdots \\ w_{N1} & \cdots & w_{NN} \end{bmatrix} \quad (2)$$

After setting the initial states of concepts, using a function that is presented in Equation (3), the simulation starts where the activation level of each concept at each iteration is adjusted based on the value of its interconnected concepts at the previous iteration. The condition for terminating the simulation is value convergence of all the concepts.

$$C_j^{t+1} = f(C_j^t + \sum_{i=1}^{n} C_i^t w_{ij}) \quad (3)$$

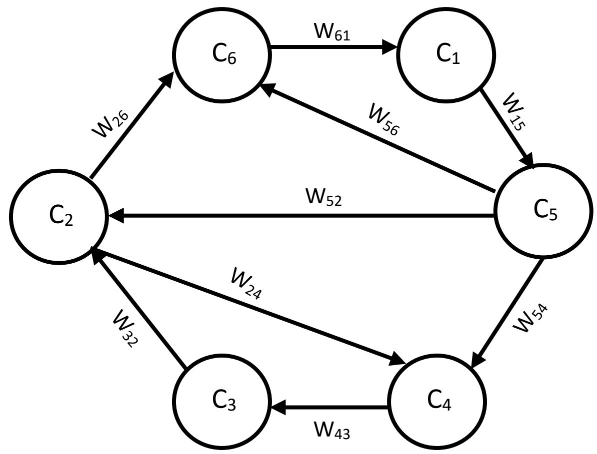

Figure 1. Example of FCM structure.

In Equation (3), C_j^t is the activation level of concept j at iteration t, and f is the threshold or transformation function which can be sigmoid, hyperbolic tangent (usually tanh(x)), bivalent (x = 0 or 1), or trivalent function (x = -1, 0 or 1). The comparison conducted by Tsadiras (2008) has shown sigmoid function works better than the others in general. The sigmoid is attractive in its quasi-linear shape about 0 and its asymptotic conduct at the extremes modeling "diminishing returns" behavior. Equation (4) shows the formulation of the sigmoid function where $c > 0$ is a constant parameter determining the steepness of the function.

$$f(x) = \frac{1}{1+e^{-cx}} \tag{4}$$

The objective of an FCM model is to learn the weight matrix in a way that the difference between the initial (in our case, target/real-world actual) concept values and the response, which is the final steady-state of concepts, is minimized. This result indicates the weight structure provides an explanation of those target concept values. To this end, many learning algorithms have been proposed in the literature. In general, these existing algorithms may fail to match the target values and may produce connection weights that are illogical.

3. FCM Learning Algorithm and Visualization

In the same vein with Amit and Meir (2019) who suggest that bias in the learning process is two-tiered, within each task and between tasks, we argue that bias needs to be divided into the circumstances around the whole system (system bias) which affects all concepts; and circumstances that impact a specific concept (concept bias). In the community health example, the state of healthcare might be biased by the overall community condition and by conditions unique to healthcare. Interpreting the real-world source of the bias requires analysis on the part of system experts.

FCM learning Algorithm and Visualization (FLAV) contains three phases to learn the system bias, FCM weight matrix, and individual concept biases using an approach similar to the neural network error backpropagation mechanism. The error backpropagation in neural networks minimizes the error function in relationship weights between the nodes in a network by using the methods of gradient descent (Rojas, 1996). In backpropagation, the error function of every neuron is calculated after processing initial data as the input (Rumelhart et al., 1986). Iteration in feed-forward backpropagation network is influenced mainly by training set variation, while in our approach it is influenced by the iterative process toward convergence that exists in the FCM. More detail about the backpropagation method can be found in (Rumelhart et al., 1986; Hecht-Nielsen, 1992).

Figure 2 shows the complete process of FLAV. As it is shown in the overview of the algorithm, in order to get the algorithm started, we need the target concept values, expert guesses at relationship weights along with any desired bounds, and other learning parameters. Weight bounds are set at rational limits by an expert on the modeled system. In practice, weights are often set based on translation of linguistic fuzziness from experts into bounds. Experts would indicate direction of correlation – positive or negative, magnitude along some measure of influence – very weak to very strong, and weight bound range based on some sense of certainty – from very loose to very tight.

FLAV consists of three main phases: 1) learning the system bias: computing the bias within the system and learning it to reach a bias value which minimizes the error between the network's given output and desired output. At this phase, the desired output refers to the initial value of nodes which is C_j^0, 2) learning the weights: updating weights using feed-forward backpropagation. Like phase 1, at this phase, the desired output refers to the initial value of nodes which is C_j^0, and 3) Learning concept bias: learning individual concept bias using the learned weight matrix and learned system bias.

Figure 3 shows the mechanism of FLAV based on the sigmoid function. In this figure, we assume there are three concepts. The black circles show the desired value of the concepts, while the purple circles show the values of the concepts after the FCM simulation without our learning process. As shown in the figure, using our proposed learning process we try to minimize the concept deviation from the desired values by learning the weighs in different phases of the algorithm.

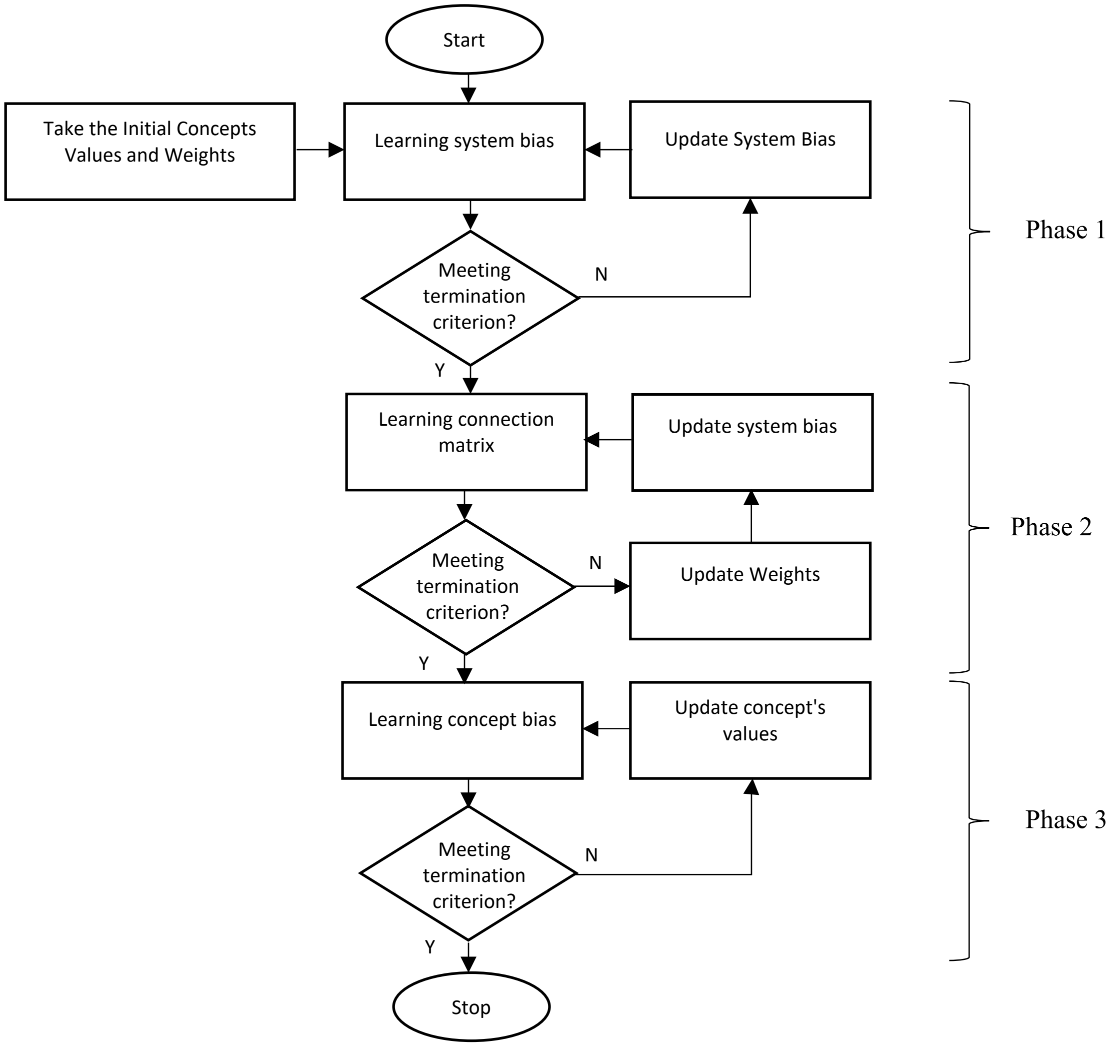

Figure 2. Overview of FLAV

Here we explain each step with more detail. We also presented the inputs needed to get the algorithm started in Table 1.

Phase 1: learning the system bias

Step 1.1. Iteratively compute the final values of concepts using Equation (5) until the model reaches a steady state which means $C_j^t = C_j^{t+1}\ \forall j = 1, \dots, n$. In Equation (5), C_j^{t+1} is the value of concept *j* at iteration *t+1*, C_j^t is the value of concept *j* at iteration *t*, w_{ij} represents the relationship weight between concept *i* and *j*, S_{bias}^e is the bias within the system at epoch *e* and *n* represents the number of concepts.

$$C_j^{t+1} = \frac{1}{1+e^{-c(C_j^t+S_{bias}^e+\sum_{i=1}^{n} w_{ij}C_i^t)}} \tag{5}$$

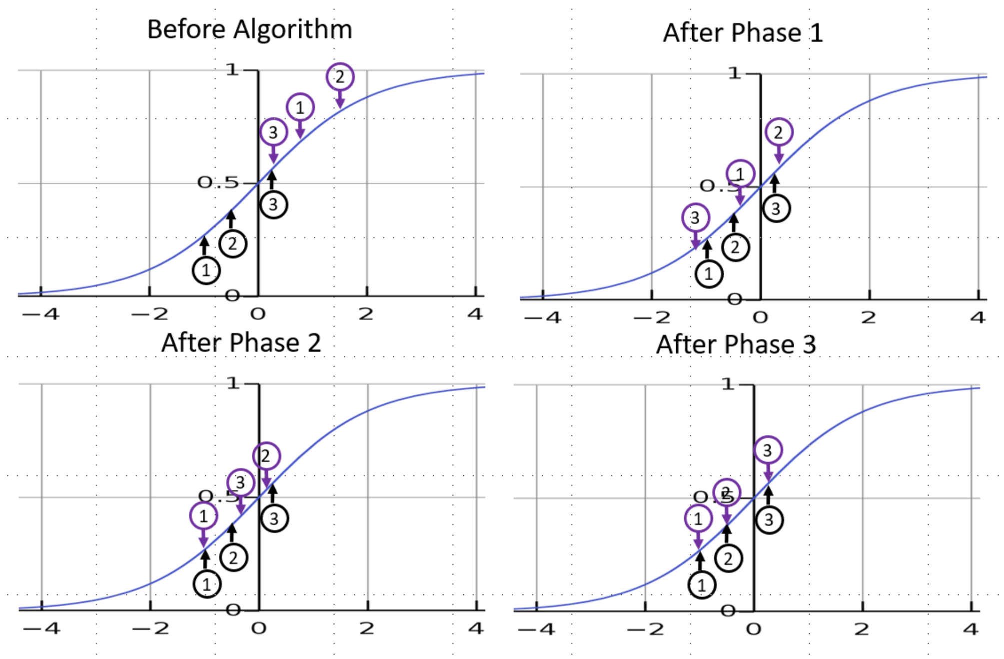

Figure 3. Mechanism of FLAV based on sigmoid function

Step 1.2. Update the value of S_{bias}^{e} using Equations (6) where S_{bias}^{e-1} is the value of bias at epoch *e*-1, γ is the learning rate in phase 1, and C_j^0 is the initial (target) value of concept *j*. The description of all other terms is the same as stated above. Note that we need to initialize S_{bias}^{e} for *e=0*.

$$S_{bias}^{e} = S_{bias}^{e-1} + \gamma \sum_{j=1}^{n} (C_j^t - C_j^0) C_j^t \, (1 - C_j^t) \tag{6}$$

Step 1.3. If $\left|\sum_{j=1}^{n} (C_j^t - C_j^0) C_j^t \left(1 - C_j^t\right)\right| \leq \xi$ or $m \geq M$, save the S_{bias}^{e} and go to the next phase which is learning the weight matrix, otherwise, go to step 1.1. Note that ξ is the accepted error in phase 1, m is the number of epochs at this phase, and M is the maximum number of epochs.

Phase 2: learning the weight matrix

Step 2.1. Using the final S_{bias}^{e} value obtained from phase 1, iteratively compute the values of concepts using Equation (7) until the values reach a steady state which means $C_j^t = C_j^{t+1} \; \forall j = 1, \dots, n$. Make $S_{bias}^{e'} = S_{bias}^{e}$ for *e=0* where $S_{bias}^{e'}$ is the value of bias in phase 2 (note that system bias keeps updating in phase 2 until the algorithm meets the termination criteria of Phase 2).

$$C_j^{t+1} = \frac{1}{1+e^{-c(C_j^t + + S_{bias}^{e'} + \sum_{i=1}^{n} w_{ij} C_i^t)}} \tag{7}$$

Step 2.2. Update the values of w_{ji} for $w_{ji} \neq 0$, using Equations (8) and (9) where $w_{ji}^{change^{e'}}$ is the amount of change in w_{ji} at epoch e',$w_{ji}^{e'}$ is the new value of w_{ji} at epoch e', $w_{ji}^{e'-1}$ is the value of w_{ji} at epoch $e'-1$, w_{ji}^{min} is the (expert set) minimum value of w_{ji}, w_{ji}^{max} is the (expert set) maximum value of w_{ji}, and γ_1' is the learning rate for weight change. The description of all other terms is the same as stated earlier. The mount of weight change is calculated using backpropagation. Please refer to Rumelhart et al (1986) for more information.

$$w_{ij}^{change^{e'}} = (C_j^0 - C_j^t)(1 - C_j^t)C_j^t C_i^t \quad (8)$$

$$w_{ij}^{e'} = \min(\max\left(w_{ij}^{e'-1} + \gamma_1' w_{ij}^{change^{e'}}, w_{ij}^{min}\right), w_{ij}^{max}) \quad (9)$$

Step 2.3. Update the $S_{bias}^{e'}$using Equation (10). Where $S_{bias}^{e'-1}$ is the value of bias at epoch $e'-1$, γ_2' is the learning rate for system bias in phase 2, and the description of the other terms is the same as stated earlier.

$$S_{bias}^{e'} = S_{bias}^{e'-1} + \gamma_2' \sum_{j=1}^{n}(C_j^t - C_j^0)C_j^t\,(1 - C_j^t) \quad (10)$$

Step 2.4. If $\sum_{j=1}^{n}(C_j^t - C_j^0)^2 \leq \xi'$, or $m' \geq M$, go to the next phase which is learning the final status of the system considering the learned bias value and weight matrix, otherwise, go to step 2.1. ξ' is the accepted error in phase 2 and m' is the number of epochs in this phase. If the terminating criteria is the error, then the network has essentially been learned and Phase 3 is not required. In such a case, phase 3 produces very small concept biases to explain the small, acceptable amount of the error function remaining in the FCM.

Table 1

Algorithm inputs

Input	Description
c	Sigmoid factor (set to 1 in this research)
C_i^0	Initial values of concepts
w_{ji}	Weight matrix
S_{bias}^{e}	Initial value of system bias
γ	Learning rate in phase 1
ξ	Acceptable error in phase 1
γ_1'	Learning rate for weight change in phase 2
γ_2'	Learning rate for system bias in phase 2
ξ'	Acceptable error in phase 2
M	Maximum number of epochs
γ''	Learning rate in phase 3
ξ'	Acceptable error in phase 3

Phase 3: Learning concept bias

Step 3.1. Iteratively compute the values of concepts using Equation (11) until all concepts converge to a steady-state. In Equation (11), $C\_bias_j^{e''}$ represents the bias in concept j at epoch e'', $S_{bias}^{e'}$ is the final value of bias obtained from phase 2, and w_{ji}^{final} is the final value of w_{ji} obtained from phase 2. The description of all other terms is the same as mentioned earlier. Note that for concepts with no associated error at the end of stage 1, there may be some very small concept bias determined based on the correction of those concepts which had more substantial error.

$$C_j^{t+1} = \frac{1}{1+e^{-c(C_j^t + S_{bias}^{e'} + C\_bias_j^{e''} + \sum_{i=1}^{n} w_{ij}^{final} C_i^t)}} \qquad (11)$$

Equation (12) shows the detail of calculating the concept bias. Where γ'' is the learning rate in phase 3, C_j^0 is the initial value of concept j, and C_j^t is the final steady value of concept j at iteration *t* of epoch e''.

$$C\_bias_j^{e''} = C\_bias_j^{e''-1} + \gamma''(C_j^t - C_j^0) \qquad (12)$$

Step 3.2. If $\sum_{i=1}^{n}(C_j^t - C_j^0)^2 \leq \xi''$, or $m'' \geq M$ stop, otherwise go to step 3.1. ξ'' is the accepted error in phase 2 and m'' is the number of epochs in this phase.

4. Community Health Example

To facilitate understanding of algorithm function, consider a simple FCM of community health with four concepts: education, healthcare, employment and public safety. Suppose an initial matrix of connection weights between these factors was proposed by a panel of experts. Additionally, four different communities are considered with varying levels of "health" as determined by measures in the concept areas. Table 2 shows the outcomes of the three phases of FLAV. In the table, the two sets of results show the impact of weight change being "loose" (+/- 0.4 from expert value) or "tight" (+/- 0.2 from expert value). Loose bounds are shown to the left in the table and tight bounds to the left. For each data set, the target concept values are shown for each of the four communities.

Phase 1 computes the system bias, a measure of overall community health. This bias provides an indication of overall system health when compared across systems. It increases across the community as the level of overall health increases. The Phase 1 system bias result is the same for both tight and loose weights because those weights are not changed in the initial phase of FLAV.

Phase 2 adjusts the weights and system bias in an attempt to match the target concept values. Weights that have hit the bound are shown in red. As expected, more weights hit the bound when the weights are tighter. As a result, the changes in system bias during Phase 2 tend to be higher with tighter bounds. When all the weights coming into a concept hit their bound during training, it means that concept may not be fully explained by the system bias and weights (as bounded). This situation is what necessitates Phase 3 of the learning algorithm. With loose bounds, the high health system is fully explained after Phase 2 so concept bias is 0. With tight bounds, the high health system has three concepts (C1-C3) with all incoming weights at their bound, requiring explanation by concept bias. Weights at a bound are shown in red.

In Phase 3, we learn the deviation from concept target values not explained by system bias or weights. With the tight bounds - high health scenario, C1-C3 are fully constrained by weight bounds. As a result, in Phase 3 the concept bias adjusts to enable the concept to match its target. A positive concept bias denote a positive deviant (concept factor that is above what would be expected due to system bias and weight matrix). Note that an unconstrained concept like C4 may develop a small concept bias as other concept biases are learned, but this value will be negligibly small.

At the end of the three phases of the algorithm, you have produced an FCM that matches the target concept values and does not violate any weight constraints.

Table 2

Community Health Algorithm Results

Comparison with weight bounds of +/- 0.4						Comparison with weight bounds of +/- 0.2				
Low Health										
	Con Target	0.3	0.2	0.4	0.1	Con Target	0.3	0.2	0.4	0.1
Phase 1	Sys Bias	-1.330				Sys Bias	-1.330			
Phase 2	Sys Bias	-1.329				Sys Bias	-1.317			
	Weight matrix					Weight matrix				
		C1	C2	C3	C4		C1	C2	C3	C4
	C1	0.000	0.000	0.700	-0.400	C1	0.000	0.200	0.500	-0.200
	C2	0.400	0.000	0.400	-0.600	C2	0.200	0.000	0.200	-0.400
	C3	0.400	-0.200	0.000	-0.800	C3	0.200	0.000	0.000	-0.600
	C4	-0.320	-0.400	0.400	0.000	C4	-0.400	-0.200	0.200	0.000
Phase 3	Con Bias	-0.026	-0.137	0.193	-0.408	Con Bias	0.089	-0.310	0.301	-0.601
Comparison with weight bounds of +/- 0.4						Comparison with weight bounds of +/- 0.2				
Moderate Health										
	Con Target	0.45	0.35	0.55	0.25	Con Target	0.45	0.35	0.55	0.25
Phase 1	Sys Bias	-0.780				Sys Bias	-0.780			
Phase 2	Sys Bias	-0.796				Sys Bias	-0.806			
	Weight matrix					Weight matrix				
		C1	C2	C3	C4		C1	C2	C3	C4
	C1	0.000	0.013	0.616	-0.176	C1	0.000	0.200	0.500	-0.174
	C2	0.254	0.000	0.325	-0.384	C2	0.200	0.000	0.200	-0.390
	C3	0.296	-0.200	0.000	-0.617	C3	0.200	0.000	0.000	-0.600
	C4	-0.426	-0.276	0.224	0.000	C4	-0.400	-0.200	0.200	0.000
Phase 3	Con Bias	0.000	0.000	0.000	0.000	Con Bias	0.075	-0.203	0.112	0.002
Comparison with weight bounds of +/- 0.4						Comparison with weight bounds of +/- 0.2				
High Health										
	Con Target	0.6	0.5	0.7	0.4	Con Target	0.6	0.5	0.7	0.4
Phase 1	Sys Bias	-0.283				Sys Bias	-0.283			
Phase 2	Sys Bias	-0.299				Sys Bias	-0.387			
	Weight matrix					Weight matrix				
		C1	C2	C3	C4		C1	C2	C3	C4
	C1	0.000	0.068	0.486	-0.063	C1	0.000	0.200	0.500	-0.015
	C2	0.223	0.000	0.197	-0.270	C2	0.200	0.000	0.200	-0.225
	C3	0.243	-0.199	0.000	-0.476	C3	0.200	0.000	0.000	-0.419
	C4	-0.443	-0.256	0.141	0.000	C4	-0.400	-0.200	0.200	0.000
Phase 3	Con Bias	0.000	0.000	0.000	0.000	Con Bias	0.113	-0.153	0.054	-0.003

Comparison with weight bounds of +/- 0.4						Comparison with weight bounds of +/- 0.2				
Very High Health										
	Con Target	0.8	0.7	0.9	0.6	Con Target	0.8	0.7	0.9	0.6
Phase 1	Sys Bias	0.519				Sys Bias	0.519			
Phase 2	Sys Bias	0.511				Sys Bias	0.337			
	Weight matrix					Weight matrix				
		C1	C2	C3	C4		C1	C2	C3	C4
	C1	0.000	0.008	0.578	-0.079	C1	0.000	0.200	0.500	-0.010
	C2	0.207	0.000	0.274	-0.287	C2	0.200	0.000	0.200	-0.220
	C3	0.216	-0.200	0.000	-0.491	C3	0.200	0.000	0.000	-0.413
	C4	-0.441	-0.317	0.220	0.000	C4	-0.400	-0.200	0.200	0.000
Phase 3	Con Bias	0.000	0.000	0.000	0.000	Con Bias	0.169	-0.230	0.300	0.003

5. Experiment

In this section, we aim to validate and understand the function of FLAV and facilitate the interpretation of its output. In developing these experiments, the following practices are followed:

1. Concepts are defined to be normalized fuzzy numbers [0,1] such that 1 is the most preferred or attractive state and 0 is the least preferred state.
2. An expert may be allowed to place tighter bounds on a relationship, but in all cases, relationship weights are in the range [-1, 1].
3. Where an expert asserts that no causal relationship exists from factor A to factor B, the associated relational weight is maintained at 0 and not modified by the algorithm.

We use three sets of networks with a different number of nodes, connectedness level, weight size, weight bounds, and health levels (which are referred to different concept inputs) to examine the impact of various factors on the learning process. The level of health within a system is associated with the average concept value and system bias. We also develop more understanding in terms of different ways to compare one system to another.

5.1 FCM Experimental Design Factors

We define measures for understanding the behavior of an FCM model and use these measures for model comparison. Table 3 shows a list of factors we will use to compare different FCM model structures.

As shown in the table, we examine the impact of the number of concepts, level of connectedness, weight, and systems health on FLAV behavior on an FCM model. Using a different number of concepts, we aim to understand if the size of a system with similar features will have a significant impact on FLAV behavior. Connectedness is the level of connection between nodes which could be full, meaning mostly all the nodes are dependent on each other, or partial, meaning nodes are only partially connected. Figure 4 shows an example of fully and partially connected models.

Table 3

Measures for comparing different FCM model Structures

Experimental measures	Levels			Description
Number of concepts	4*N*	10*N*	20*N*	Networks with 4, 10, and 20 nodes.
Connectedness	Partial		Full	If (No. of relations)/(*n*(*n*-1)) is equal or less than 0.5, the connectedness is considered to be *Partial*, otherwise *Full*.
Weight value	Small		Large	If the average weight is equal or less than 0.3, weight values are assumed to be *Small*, otherwise, *Large*.
Weight bound	Loose		Tight	*Loose* bound is where lower and upper bounds of weights are set to be -1 and 1, respectively *Tight* bound is where weights are allowed to move +/- 0.2 from original expert weight estimate.
System health	Low	Medium	High	System health is assumed to be *Low*, *Medium*, or *High*, if the average of initial values of concepts is in [0,0.3], (0.3, 0.7], or (0.7,1] interval, respectively.

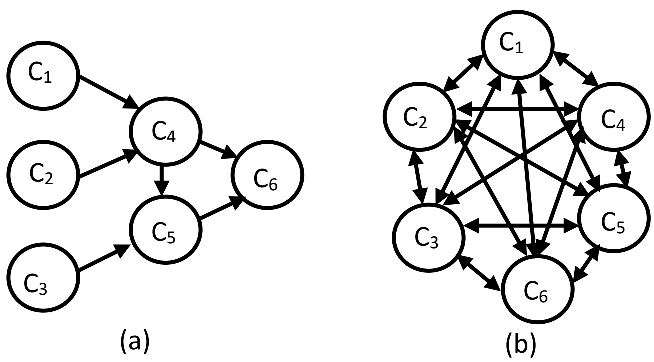

Figure 2. (a) Example of a partial system, (b) and example of a full system.

With the weight size and bound measures, we analyze to what extent the behavior of the algorithm is dependent on the interrelationship weights. By eliciting information on the strength of causality between each pair of cause-effect concepts, it is possible to provide richer recommendations and comparisons about different decision options or systems. We also will analyze the impact of system health on model performance.

5.2 Dataset

The design presented in Table 3 is tested at the levels shown in the table. Table 4 shows our assumptions regarding the initial setup of the models in the three phases of FLAV. The total set of initial values and weights we generated for all the experiments was 180 as there are 12 levels in each experiment for the three network sizes, and the number of experiments is 5. We used a weight bound measure to see the difference between tightly bound models with loosely bound models, we repeated all the experiments twice, but one of them with loosely bound weights and the other one with tightly bound weights. Thus, overall, we have 360 models with different measures and levels.

Table 4

Initial setup assumptions

	Sigmoid factor	Initial system bias	Initial concept bias	Learning rate	epsilon	Max No. iterations
Phase 1	c = 1	$S_{bias}^{e} = 0$	-	$\gamma = 1$	$\xi = 0.00001$	M=10000
Phase 2	c = 1	-	-	$\gamma'_1 = 1, \gamma'_2 = 1$	$\xi' = 0.00001$	M=10000
Phase 3	c = 1	-	$C\_bias_i^{e''} = 0$	$\gamma'' = 1$	$\xi'' = 0.00001$	M=10000

5.3 Model Structure Comparison

The measures we used to compare different models include system bias at phases 1 and 2, average of learned weights in phase 2, number of weights hitting the bounds in phase 2, values of concept biases in phase 3, and number of concepts with approximately zero bias at the end of phase 3 (we assumed the bias in a concept is zero if the absolute value of concept bias obtained at the end of phase 3 is equal or less than 0.001). We coded FLAV using Python and PyCharm Community Edition 2018.3.5.

As expected, FLAV learned all concept values. The result of this analysis is summarized visually in Figure 5. As shown in Figures 5.a and b, the values of system bias at phases 1 and 2 are higher in fully connected models than the partially connected ones. Because the concepts are all nonnegative and typically with positive correlation, higher levels of connection push further to the right in the threshold function requiring more system bias correction.

The number of weights hitting the bound (Figure 5.d) is also greater in fully connected models because we have a greater number of arcs in fully connected models so more weights hit the bound. However, the percentage of arcs hitting the bounds in partial models is 37% while this percentage is 17% for full models. We may conclude that fully connected models have more ability to learn the system and accordingly adjust the weights compared to partial models. Moreover, as the number of concepts with zero bias (Figure 5.f) is greater in fully connected models, we can observe that in fully connected models the bias embedded in individual concepts is less than that of partial models. In other words, a higher number of interrelationships between the concepts helps the individual concepts to reach the desired values. As one would expect, the fully connected networks possess a greater ability to fully explain the individual concept value without requiring concept bias.

Regarding the impact of weight size (Figure 5.a and b) on system characteristics, we see that the values of system bias at both phases 1 and 2 are higher in large weight size models compared to small ones in 4 and 10 nodes models. However, we see a reverse pattern in 20 nodes models. More specifically, in 20 nodes models, the values of system bias are higher in small weight size models. Also, as we expected the average weight, as depicted in Figure 5.c, is higher in large weight size models compared to small ones. Concerning concept bias, as shown in Figure 5.e, we see a radical increase in the values of concept bias in 20 nodes large weight size models compared to small weight size models. This result is supported in Figure 5.f, as we see a fewer number of concepts have reached zero bias in 20 nodes large weight size models compared to the small weight size ones, more specifically the percentage of concepts that have reached zero bias is 37% and 50% in 20 nodes large and small weight size models, respectively.

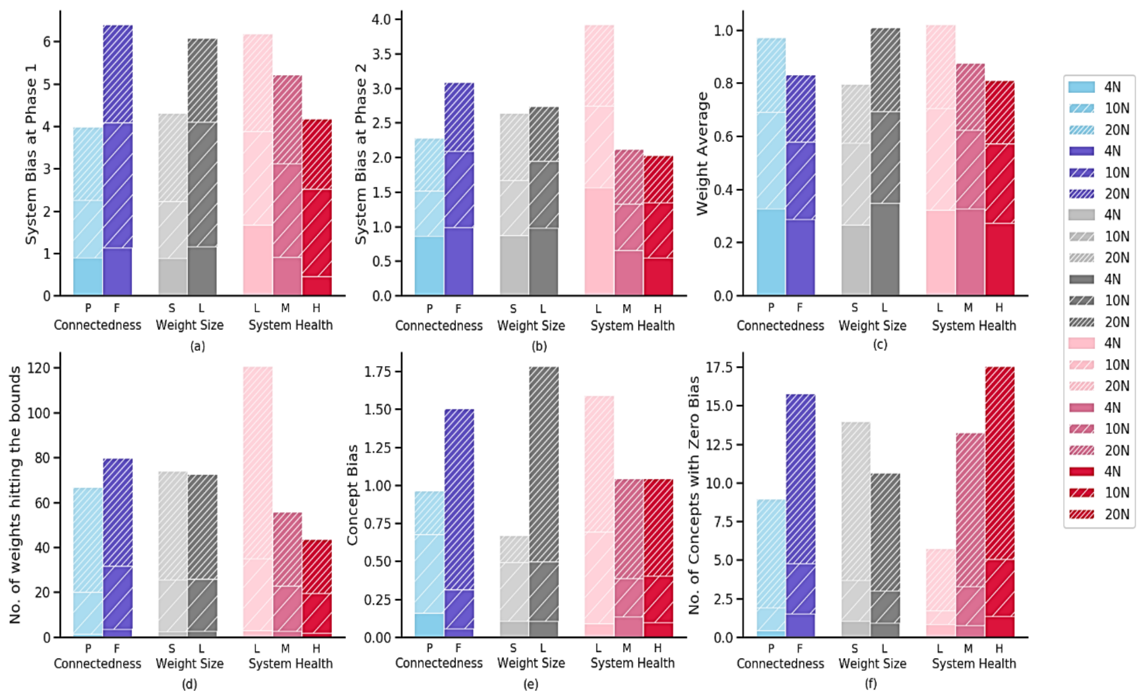

Figure 3. Comparison of partially connected with fully connected, small weight size with large weight size, and three levels (low, medium, and high) of system health in terms of (a) system bias at phase 1, (b) system bias at phase 2, (c) weight average, (d) overall number of weights hitting the bounds, (e) concept bias, and (f) number of concepts with zero bias.

The results of system health comparison in different models are in line with our expectations. As we move from a low health system to a healthy system, we expect the values of system bias, concept bias, and the number of weights hitting the bounds to decrease. This outcome is connected to the behavior of the selected threshold function. In the same vein, we expected the number of concepts with zero bias would be higher in healthier systems compared to medium and low health systems since we expect less amount of concept bias would be embedded in healthy models. The results depicted in all six plots of Figure 5 support our initial expectation.

Although Figure 5, gives us a general idea about the performance of different FCM model structures, we pursued our analysis with statistical testing to find out whether the differences we observed are statistically significant. Our statistical tests showed almost all differences shown in Figure 5 are statistically significant. More specifically:

- System bias values at phase 1 and 2 are significantly higher in fully connected models compared to partially connected ones in most cases.
- The number of concepts with zero bias is significantly higher in all fully connected models compared to partially connected ones.
- System bias at phase 1 is significantly higher in 4 and 10 nodes large weight size models.

- Average weight is significantly higher in 4 and 20 nodes models with large weight size compared to small weight size models.
- In 20 nodes models with small weight size the value of bias is significantly lower and the number of concepts with zero bias is significantly greater compared to large weight size models.
- System bias at phases 1 and 2 is significantly higher is low health models.
- There is a significant difference between systems with different health levels and the amount of deviation from their desired state.

5.4 FLAV visualization

Next, we propose a novel visualization for FLAV to facilitate interpretation. Figure 6 shows our proposed visualization for a three-concept model where each node contains information regarding the concept name, initial value, and concept bias after running the algorithm. Different colors represent the direction of biases. A concept is black if the concept bias is zero and it is green or red if the bias reflects positive deviant or negative deviant behavior, respectively. The term positive deviant stems from the sociology to describe individuals who experience uncommon success relative to others in the same environment. This representation also enables us to visualize if a learned weight has hit its expert set bound. An arc is green if the learned weight hits the upper bound and it is red when it hits the lower bound.

Note that where there is concept bias yielding a green or red circle, all the arcs going into that circle will be of the same color. This behavior is a function of the fact that concept values are non-negative.

We can show the magnitude of system or concept bias with varying line thickness. We also show the final value of the system bias with a circle around the model along with indicating magnitude with the line thickness. The system bias will be green, black, or red if it has positive, zero, or negative deviance, respectively.

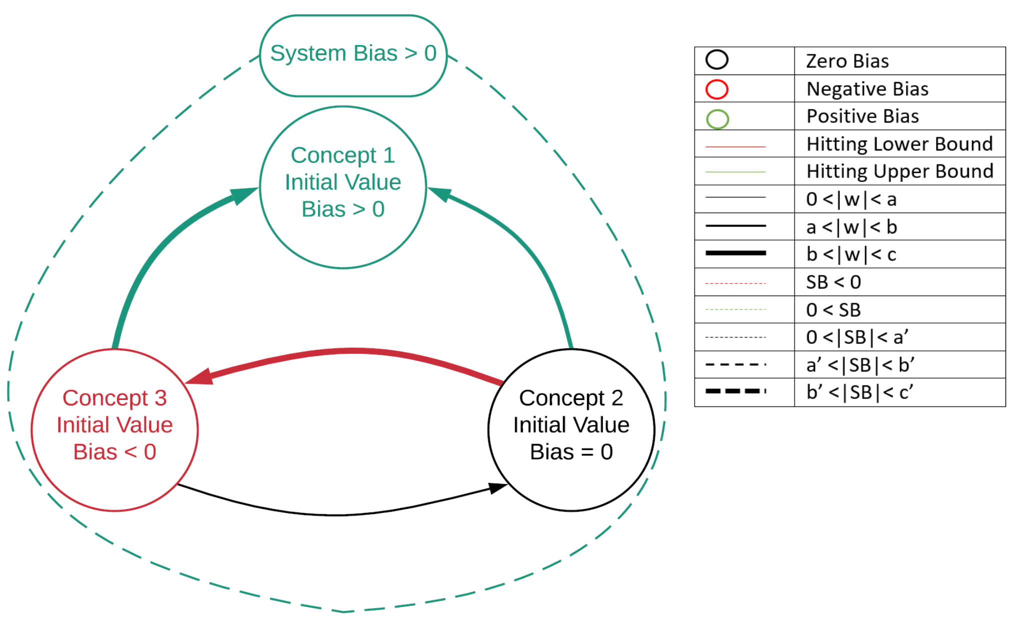

Figure 4. New FCM visualization.

This visualization facilitates the rapid understanding of a single system and comparison between multiple systems. Where relationships are at bounds, experts might consider whether that bound might be altered. Positive and negative deviants are quickly identified focusing future study to determine the underlying real-world cause of the deviation. In the case of positive deviance, what is learned from this study might be translated to other systems.

Figure 7 shows FCM results representing the “health” of two communities and modeled factors of healthcare, education, employment, and public safety. This new representation gives us a quick view of the FCM model and where bias exists. Although the interpretation of bias in real life is dependent on the threshold function, this representation helps us to compare the biases relatively. Figure 7.a, for example, shows the initial input of the model is mostly biased as most of the arcs and concepts are either in green or red while in Figure 7.b shows a system with small bias as most of the arcs and concepts are in black. Regarding the importance of the concepts, since all the arcs are presented in the same thickness, it is suggested that all the relationships are of equal importance. In this situation, community developer would want to discover the reasons for positive healthcare and safety and negative education shown in the Figure 7.a.

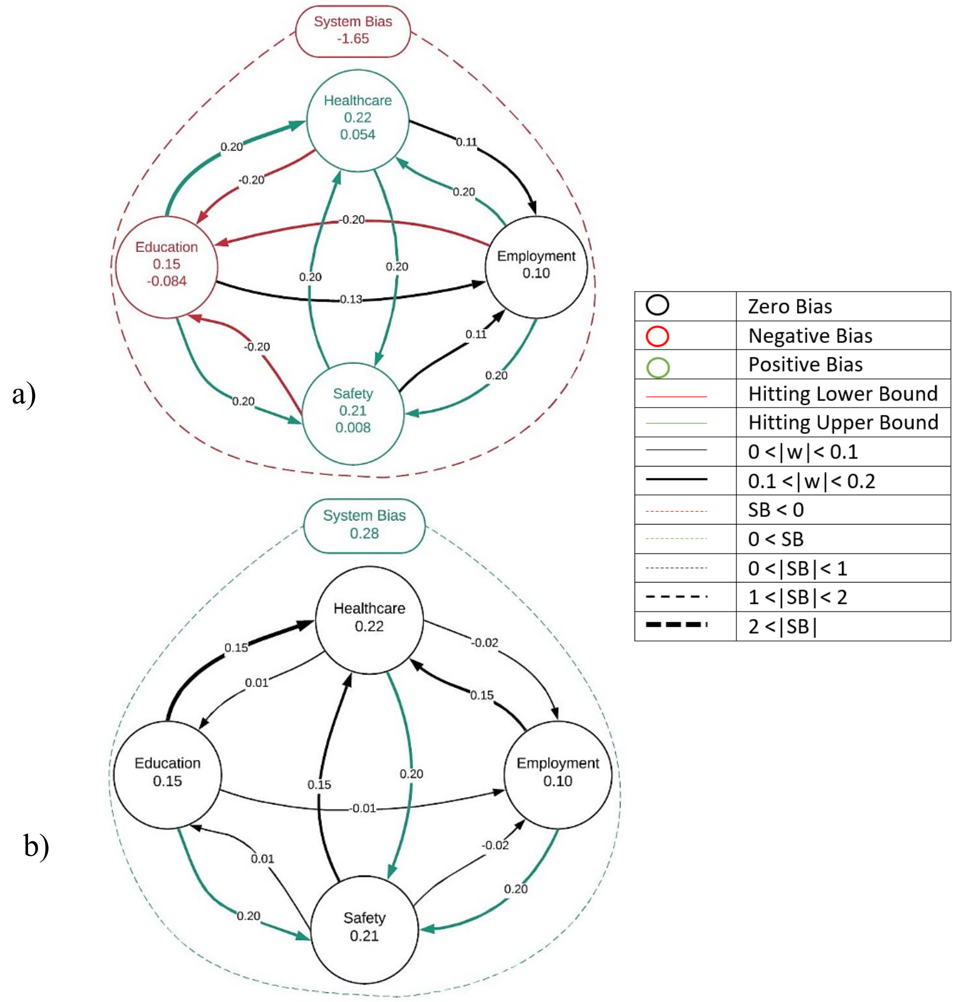

Figure 5. New FCM representation: a) full tight 4 nodes large weights low health level model, b) full tight 4 nodes large weights high health level model (Experiment 5).

6. Conclusion and Future Research

In this study, we propose a new learning algorithm for FCM and implement a new approach for FCM analysis which provides important contributions to the theory and practice in the FCM literature. Our first contribution is related to the introduction of FLAV which can match target concept values while maintaining logical connection weights. The algorithm accomplishes this by computing biases for the whole system and each of the individual concepts based on the neural networks' error backpropagation mechanism. Considering bias in the learning process can make the algorithm more flexible to simulate the real-world and fit the actual state. *This method enables detailed quantitative and visual modeling of complex systems that is generally applicable. The method begins with a general model created by experts and is able to adaptively learn a specific system in a manner that facilitates improvement and comparison*. Unlike other FCM learning algorithms that only focus on the error which is based on the differences between the initial and final concept values, we argue that external circumstances beyond relational weights have an impact on the algorithm output and in order to make the learning process more realistic we need to take them into account. Finally, the iterative nature of learning makes small changes to the weights as balancing corrections made for experts who may tend to consistently overstate or understate relationship weights.

In addition to algorithm development, an experimental design assessing the behavior and logic of this method was conducted with results summarized. Finally, we propose a novel format for FCM visualization. Using this new visualization, we can easily engage the audience with the amount of bias in the model as well as the nature and importance of the relationships.

Future studies using real-world datasets would assist with demonstrating the explanatory power of this approach. Second, as indicated in the detail of the experimental section, in order to get the algorithm started we need to initialize various learning parameters including the biases. Future studies are needed to validate and investigate the impact of the learning parameters on the learning process of the algorithm. Such analysis would help us in understanding how the biases should be initialized. Third, in this study, we only considered 4, 10, and 20 nodes models, while in real-world situations we may have to deal with larger scale FCMs. The impact of the threshold function, either the sigmoid parameter or other function might be considered. Another direction for future research is analyzing the proposed algorithm for large-scale FCMs and studying the limitations it may have. A final area of future research is use of a learned FCM using FLAV might be used to discover high leverage opportunities for systems improvement.

References

Amit, R., & Meir, R. (2019). Lifelong learning and inductive bias. *Current Opinion in Behavioral Sciences, 29*, 51-54. Retrieved from http://www.sciencedirect.com/science/article/pii/S2352154618301955. doi:https://doi.org/10.1016/j.cobeha.2019.04.003

Craiger, J. P., Goodman, D. F., Weiss, R. J., & Butler, A. (1996). Modeling organizational behavior with fuzzy cognitive maps. *International Journal of Computational Intelligence and Organizations, 1*(3), 120-123.

Dickerson, J. A., & Kosko, B. (1994). Virtual worlds as fuzzy cognitive maps. *Presence: Teleoperators & Virtual Environments, 3*(2), 173-189.

Ferreira, F. A., Jalali, M. S., & Ferreira, J. J. (2016). Integrating qualitative comparative analysis (qca) and fuzzy cognitive maps (fcm) to enhance the selection of independent variables. *Journal of Business Research, 69*(4), 1471-1478.

Giabbanelli, P. J., Torsney-Weir, T., & Mago, V. K. (2012). A fuzzy cognitive map of the psychosocial determinants of obesity. *Applied soft computing, 12*(12), 3711-3724.

Gray, S. A., Gray, S., De Kok, J. L., Helfgott, A. E., O'Dwyer, B., Jordan, R., & Nyaki, A. (2015). Using fuzzy cognitive mapping as a participatory approach to analyze change, preferred states, and perceived resilience of social-ecological systems. *Ecology and Society, 20*(2).

Hecht-Nielsen, R. (1992). Theory of the backpropagation neural network. In *Neural networks for perception* (pp. 65-93): Elsevier.

Jamshidi, A., Ait-kadi, D., Ruiz, A., & Rebaiaia, M. L. (2018). Dynamic risk assessment of complex systems using fcm. *International Journal of Production Research, 56*(3), 1070-1088.

Kang, I., Lee, S., & Choi, J. (2004). Using fuzzy cognitive map for the relationship management in airline service. *Expert Systems with Applications, 26*(4), 545-555.

Kim, M.-C., Kim, C. O., Hong, S. R., & Kwon, I.-H. (2008). Forward–backward analysis of rfid-enabled supply chain using fuzzy cognitive map and genetic algorithm. *Expert Systems with Applications, 35*(3), 1166-1176.

Kosko, B. (1986). Fuzzy cognitive maps. *International journal of man-machine studies, 24*(1), 65-75.

Özesmi, U., & Özesmi, S. L. (2004). Ecological models based on people's knowledge: A multi-step fuzzy cognitive mapping approach. *Ecological modelling, 176*(1-2), 43-64.

Papageorgiou, E. I. (2012). Learning algorithms for fuzzy cognitive maps—a review study. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), 42*(2), 150-163.

Papageorgiou, E. I., Parsopoulos, K. E., Stylios, C. S., Groumpos, P. P., & Vrahatis, M. N. (2005). Fuzzy cognitive maps learning using particle swarm optimization. *Journal of Intelligent Information Systems, 25*(1), 95-121.

Rezaee, M. J., Yousefi, S., Valipour, M., & Dehdar, M. M. (2018). Risk analysis of sequential processes in food industry integrating multi-stage fuzzy cognitive map and process failure mode and effects analysis. *Computers & Industrial Engineering, 123*, 325-337. Retrieved from http://www.sciencedirect.com/science/article/pii/S0360835218303279. doi:https://doi.org/10.1016/j.cie.2018.07.012

Rojas, R. (1996). The backpropagation algorithm. In *Neural networks* (pp. 149-182): Springer.

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. *nature, 323*(6088), 533.

Schneider, M., Shnaider, E., Kandel, A., & Chew, G. (1998). Automatic construction of fcms. *Fuzzy sets and systems, 93*(2), 161-172.

Sharif, A. M., & Irani, Z. (2006). Exploring fuzzy cognitive mapping for is evaluation. *European Journal of Operational Research, 173*(3), 1175-1187.

Taber, R. (1991). Knowledge processing with fuzzy cognitive maps. *Expert Systems with Applications, 2*(1), 83-87.

Tsadiras, A. K. (2008). Comparing the inference capabilities of binary, trivalent and sigmoid fuzzy cognitive maps. *Information Sciences, 178*(20), 3880-3894.