

INTERNATIONAL JOURNAL OF COMPUTERS COMMUNICATIONS & CONTROL
Special Issue on Fuzzy Sets and Applications (Celebration of the 50th Anniversary of Fuzzy Sets)
ISSN 1841-9836, 10(6):825-833, December, 2015.

Why Fuzzy Cognitive Maps Are Efficient

V. Kreinovich, C. Stylios

Vladik Kreinovich*

Department of Computer Science
University of Texas at El Paso
500 W. University
El Paso, Texas 79968, USA
*Corresponding author: vladik@utep.edu

Chrysostomos D. Stylios

Laboratory of Knowledge and Intelligent Computing
Department of Computer Engineering
Technological Educational Institute of Epirus
47100 Kostakioi, Arta, Greece
stylios@teiep.gr

Abstract: In many practical situations, the relation between the experts' degrees of confidence in different related statements is well described by Fuzzy Cognitive Maps (FCM). This empirical success is somewhat puzzling, since from the mathematical viewpoint, each FCM relation corresponds to a simplified one-neuron neural network, and it is well known that to adequately describe relations, we need multiple neurons. In this paper, we show that the empirical success of FCM can be explained if we take into account that human's subjective opinions follow Miller's seven plus minus two law.

Keywords: fuzzy cognitive maps, neural networks, seven plus minus two law.

1 Introduction: Fuzzy Cognitive Maps and Their Puzzling Success

Need for (imprecise) expert estimates. To characterize a real-life system, we must know its properties. Some of these properties come from measurements and are thus represented by real numbers. However, in many cases, a large amount of information comes from expert estimates.

For example, to describe the current state of a patient, it is rarely enough to collect the corresponding measurement results – such a temperature, blood pressure, etc. Medical doctors supplement this information by providing imprecise ("fuzzy") estimates, such as "somewhat soft", "small", "rather high", etc.

Similarly, to adequately describe the financial situation of a company or of a country, it is important to supplement the corresponding numbers with expert estimates describing how probable is a default or, vice versa, how probable in an increase in profitability (and how big this increase can be). A typical expert's opinion sounds like this: "a vert big increase is improbable, but it is reasonable to expect a modest increase in reasonable time".

Fuzzy techniques as a natural way to describe imprecise expert estimates. When an expert completely agrees or completely disagrees with a precise statement (such as "the price of this stock will increase by at least 5% in a year"), in the computer, the resulting expert-estimated truth value of a statement is either "true" or "false". In the computer, "true" is usually represented as 1, and "false" as 0.

When the statement is imprecise, like the one above about a modest increase, the expert is not 100% sure that the price will increase by 5%. Instead, the expert has some degree of

confidence in this 5% increase. Since full confidence in a statement is described by the number 1, and full confidence in its negation is described by the number 0, a reasonable way to describe the expert's partial confidence is by using numbers between 0 and 1: the higher the value, the larger the expert's degree of confidence. The use of numbers from the interval $[0, 1]$ for describing the experts' degrees of confidence is the main idea behind *fuzzy logic* [6, 9, 15].

From individual fuzzy properties to Fuzzy Cognitive Maps. Fuzzy properties describing a system are often interrelated, in the sense that some properties imply others. For example, in medicine, if a person is overweight and not very physically fit, this increases the possibility that this person may get diabetes and thus, may be in a pre-diabetic stage. In financial situations, if a company has many new patents, especially patents in a "hot" area like advanced bioinformatics, it is usually a good indication of its future financial prosperity, etc.

Fuzzy Cognitive Maps (FCM) are a way to describe the relations between different fuzzy properties. To describe these relations, for each property P , we first need to list all the properties P_1, \dots, P_n that directly affect the property P . Once this list is produced, we need to describe how the numerical values $x_1, \dots, x_n \in [0, 1]$ of the properties P_i affect the value x of the property P . In computational terms, we need to come up with an algorithmic function $f(x_1, \dots, x_n)$ that predicts the value x based on the known values x_1, \dots, x_n .

Which functions should we choose? A natural idea is to start with the simplest functions. The simplest possible functions are linear functions, in which case we have

$$x = w_0 + w_1 \cdot x_1 + \dots + w_n \cdot x_n. \quad (1)$$

However, we cannot simply use general linear functions:

- the predicted value should be within the interval $[0, 1]$, but
- for different combinations of weights, the above linear expression can be any real number, not necessarily a number between 0 and 1.

A reasonable idea is that, after we get the above linear combination, we then apply an additional transformation $s(x)$ that maps the whole real line to a number from the interval $[0, 1]$. In other words, instead of the linear expression (1), we use a slightly more complex expression

$$x = s(w_0 + w_1 \cdot x_1 + \dots + w_n \cdot x_n), \quad (2)$$

where $s(x)$ is a pre-selected function that maps the real line into a unit interval $[0, 1]$. This function $s(x)$ is called an *activation function*.

This is the main idea behind *Fuzzy Cognitive Maps* (FCM); see, e.g., [2, 3, 7, 8, 11, 12, 16–20, 22, 24–29]. The FCM model is used when experts provide estimates only for *some* of the properties. In this case, the values of other properties are estimates by using the corresponding formulas of type (2).

Which activations functions are used? Several different activation functions $s(x)$ have been used in FCM; the most frequently used is the *sigmoid function*

$$s(x) = \frac{1}{1 + \exp(-x)}. \quad (3)$$

The main reason why this function is used is the same reason why the same function is used in artificial neural networks: our goal is to describe human reasoning, and the sigmoid function provides a good approximate description of how similar processing is performed by the biological neurons in the brain; see, e.g., [1].

Comment. There are also theoretical reasons explaining why the sigmoid function is, in some reasonable sense, optimal; see, e.g., [10, 14, 23]. These theoretical reasons may also explain why evolution resulted in selecting this particular function in the actual brain – since this function is indeed optimal.

Fuzzy Cognitive Maps are efficient. In many practical applications, Fuzzy Cognitive Maps have led to a reasonably good description of human reasoning; see, e.g., [2, 3, 7, 8, 11, 12, 16–20, 22, 24–29].

This empirical success is puzzling. From the theoretical viewpoint, this empirical success is puzzling. Indeed, as we have mentioned, the output (3) of each corresponding fuzzy rule is the same as the output of a standard non-linear *neuron* [1, 24].

It is known that a 3-layer *neural network* has the *universal approximation property*; see, e.g., [1]. This means that if we use *several* (K) nonlinear neurons, with the outputs

$$x^{(k)} = s \left(w_0^{(k)} + w_1^{(k)} \cdot x_1 + \dots + w_n^{(k)} \cdot x_n \right), \quad (4)$$

and then use an additional linear neuron to combine these output into a single combination

$$x = W^{(0)} + \sum_{k=1}^K W^{(k)} \cdot x^{(k)}, \quad (5)$$

then, for each continuous function $x = f(x_1, \dots, x_n)$ on any box – in particular, on the box $[0, 1] \times [0, 1]$ – and for every $\varepsilon > 0$, we can find the values of the weights $w_i^{(k)}$ and $W^{(k)}$ for which, for every inputs, the final output (5) is ε -close to the desired value $f(x_1, \dots, x_n)$.

It is also known that we need *several* neurons to get the universal approximation property, a single neuron does not have this property; see, e.g., [9]. And here, we have an opposite phenomenon: in many practical cases, already a single neuron provides a good approximation for the desired dependence! This is very puzzling.

Comment. The fact that a single neuron does not have a universal approximation property can be explained if we take into account that when the dependence $x = f(x_1, \dots, x_n)$ is described by the formula (2), then for every i , we get

$$\frac{\partial f}{\partial x_i} = s' \cdot w_i,$$

where $s'(x)$ is the derivative of the activation function $s(x)$. Thus, for every $i \neq j$, we have

$$\frac{\partial f}{\partial x_i} = \text{const} \cdot \frac{\partial f}{\partial x_j},$$

where the constant is the ratio $\frac{w_i}{w_j}$. This property is already not satisfied by the simplest non-linear operation of multiplication $f(x_1, x_2) = x_1 \cdot x_2$, for which

$$\frac{\partial f}{\partial x_1} = x_2 \neq \text{const} \cdot \frac{\partial f}{\partial x_2} = \text{const} \cdot x_1.$$

What we do in this paper. In this paper, we provide a possible explanation for the (puzzling) empirical success of Fuzzy Cognitive Maps.

2 Possible Explanation for the Puzzling Empirical Success of FCMs

Main idea behind our explanation. Our main idea is to take into account the following difference between the general universal approximation property (as used in neural network theory) and what we want in Fuzzy Cognitive Maps.

The difference is that in the general applications of neural networks, the values x_1, \dots, x_n , and x are usually well-defined physical quantities, quantities which can be, in principle, measured with an arbitrary accuracy ε . For example, if we use neural networks to design an appropriate control, we want the resulting control value x to be as close to the optimal value $f(x_1, \dots, x_n)$ as possible.

In contrast, in Fuzzy Cognitive Maps, all the variables x_1, \dots, x_n , and x are degrees of confidence describing expert opinions. These degrees are, by definition, imprecise, so computing them with too much for an accuracy simply does not make sense. An expert may be able to mark his or her degree of confidence by marking 6 on a scale from 0 to 10 – which corresponds to the degree of confidence 0.6 – but a normal expert cannot meaningfully distinguish between degrees of confidence 0.61 and 0.62.

Let us show that this difference can explain the puzzling empirical success of Fuzzy Cognitive Maps.

How accurate are expert estimates: 7 ± 2 rule. Psychologists have found out that we usually divide each quantity into 7 plus plus minus 2 categories – this is the largest number of categories whose meaning we can immediately grasp; see, e.g., [13, 21] (see also [30]). For some people, this "magical number" is $7 + 2 = 9$, for some it is $7 - 2 = 5$. This rule is in good accordance with the fact that in fuzzy logic, to describe the expert's opinion on each quantity, we usually use 7 ± 2 different categories (such as "small", "medium", etc.).

Since on the interval $[0, 1]$, we can only have 7 ± 2 meaningfully different degrees of confidence, the accuracy of these degrees ranges from $1/9$ (for those who use 9 different categories) to $1/5$ (for those who use only 5 different categories).

What is the overall accuracy of the corresponding degrees. A Fuzzy Cognitive Map usually combines knowledge of a large number of experts. Since we have a large number of experts, it is practically certain that these experts include experts of all types: namely, those who can estimate their degree of confidence with a higher accuracy of $1/9$, as well as those who can only estimate their degree of confidence with a much lower relative accuracy of $1/5 = 20\%$.

In general, if we process a large amount of data of different accuracy, the accuracy of the result is determined by the lowest accuracy of the inputs. For example, if we estimate the overall amount of money $m = m_1 + m_2 + m_3$ owned by three people, and we know m_1 and m_2 with an accuracy of 1 cent, but we only know m_3 with an accuracy of 50% (i.e., we only know the ballpark estimate for m_3), then clearly our estimate for the sum m will be very inaccurate as well.

From this viewpoint, since FCM contains lower-accuracy data, with the accuracy 20%, we cannot expect the estimation results to be more accurate than that.

How accurate should our predictions be? Based on the above arguments, it makes sense to estimate the dependence of x on x_1, \dots, x_n with accuracy 20%. Attempts to get a more accurate estimation would be, in general, a useless computational exercise which is not related to the desired problem – of estimating the expert's degrees of confidence.

For example, if the expert's degree is 0.6, and our formula predicts 0.65, it is a very good match, and there is no need to come up with a formula that predicts exactly 0.6.

So, how many neurons do we need to make predictions with this accuracy: let us start our analysis. Let us show that in general, if we want predictions with accuracy 20%,

then one neuron is sufficient.

Specifically, we will show that if, instead of taking the neuron that provides the largest contribution to the prediction, we consider both neurons, then – within the given accuracy – the result will not change.

It makes sense to treat the outputs of two neurons as random variables. As we have mentioned, for a general neural network, the result is a sum of the terms corresponding to different neurons. Let t_1 and t_2 be terms corresponding to the two neurons.

In general, these terms depends on many factors, so it makes sense to treat them as random variables.

As usual in statistics, we can somewhat simplify the problem by subtracting the means $E[t_i]$ from the corresponding variables. In precise terms, instead of the original random variables t_i , we can consider the differences $d_i \stackrel{\text{def}}{=} t_i - E[t_i]$ for which the mean value is 0: $E[d_i] = 0$.

What we compare. We compare the two situations:

- a situation in which we consider the sum $d_1 + d_2$ of both neural terms, and
- a situation in which we only have a single neuron, the one that provides the largest contribution:
 - we consider d_1 if $|d_1| \geq |d_2|$, and
 - we consider d_2 if $|d_2| \geq |d_1|$.

It is reasonable to assume that the variables corresponding to different neurons are independent. Since we have no reason to believe that the variables corresponding to different neurons are correlated, it makes sense to assume that the variables t_1 and t_2 – and thus, the corresponding differences d_1 and d_2 – are independent. This conclusion is in line with the general Maximum Entropy approach to dealing with probabilistic knowledge: if there are several possible probability distributions consistent with our knowledge, it makes sense to select the one which has the largest uncertainty (entropy; see, e.g., [4, 5]), i.e., to select a distribution for which the entropy

$$S = - \int \rho(x) \cdot \ln(\rho(x)) dx$$

attains the largest possible value, where $\rho(x)$ is the probability density function (pdf).

In particular, for the case when for two random variables, we only know their marginal distributions, with probability densities $\rho_1(x_1)$ and $\rho_2(x_2)$, the Maximum Entropy approach selects the joint probability distribution with the probability density $\rho(x_1, x_2) = \rho_1(x_1) \cdot \rho_2(x_2)$ that corresponds exactly to the case when these two random variables are independent.

This independence make perfect sense for neural networks: when we train a neural network, we want to get a model which is as accurate as possible, and if we use two highly correlated neurons, we waste the second neuron to describe what the first neuron describes already.

How can we estimate the size of each random variable? For a random quantity with 0 mean, a natural measure of its size is its standard deviation σ .

If we only consider the term d_s corresponding to a single neuron, then this size can be described by the corresponding standard deviation σ_s . If we consider both neurons, then the size of the sum $d_1 + d_2$ can be similarly characterized by its standard deviation σ_{12} . Since the variables are independent, the variance σ_{12}^2 of this sum is equal to the sum $\sigma_1^2 + \sigma_2^2$ of the corresponding variances. Thus, the standard deviation σ_{12} of the sum has the form

$$\sigma_{12} = \sqrt{\sigma_1^2 + \sigma_2^2}. \quad (6)$$

What we plan to prove. We plan to prove that the change caused by adding the second neuron is, in general, below the desired accuracy bound, i.e., that

$$\left| \frac{\sigma_{12} - \sigma_s}{\sigma_s} \right| \leq 0.2. \quad (7)$$

Let us estimate the sizes σ_s and σ_{12} corresponding to two possible situations. We do not have much information about the size of the signals d_i corresponding to different neurons. We may guess some bounds $\underline{d} \leq d_i \leq \bar{d}$.

If all know about the probability distribution is that its values are always located on the interval $[\underline{d}, \bar{d}]$, then the Maximum Entropy approach recommends that we select a uniform distribution on this interval. This recommendation is in perfect accordance with common sense: if we have no reason to believe that some values from this interval are more probable or less probable than others, then it is reasonable to assume that all these values have the exact same probability, i.e., that the distribution is indeed uniform.

For a uniform distribution on the interval $[\underline{d}, \bar{d}]$, the mean value is known to be equal to the midpoint $\frac{\underline{d} + \bar{d}}{2}$ of this interval. Since we are interested in random variables d_i with 0 mean, this means that this point must be equal to 0, i.e., that we have $\underline{d} = -\bar{d}$.

Since the mean is 0, the variance is equal to the expected value of d_i^2 . Here, $d_i^2 = a_i^2$, where by $a_i \stackrel{\text{def}}{=} |d_i|$ we denoted the absolute value of the d_i . One can easily check that this absolute value a_i is uniformly distributed on the interval $[0, \bar{d}]$, with a constant probability density $\rho_i(x) = \frac{1}{\bar{d}}$, so its variance $\sigma^2 = \int x^2 \cdot \rho(x) dx$ is equal to

$$\sigma_i^2 = \int_0^{\bar{d}} x^2 \cdot \frac{1}{\bar{d}} dx = \frac{1}{3} \cdot (\bar{d})^3 \cdot \frac{1}{\bar{d}} = \frac{1}{3} \cdot (\bar{d})^2. \quad (8)$$

Due to the formula (6), we thus have

$$\sigma_{12} = \sqrt{\frac{1}{3} \cdot (\bar{d})^2 + \frac{1}{3} \cdot (\bar{d})^2} = \sqrt{\frac{2}{3}} \cdot \bar{d}. \quad (9)$$

Now, we need to estimate the variance σ_s^2 of the case when we only select one of the neurons, i.e., the expected value of the square of the selection d_s . Similarly to the one-neuron case, since $d_s^2 = |d_s|^2$, this variance is equal to the expected value of a_s^2 , where we denoted $a_s \stackrel{\text{def}}{=} |d_s|$. By definition, $a_s = |d_s| = \max(|d_1|, |d_2|) = \max(a_1, a_2)$.

We know that a_1 and a_2 are two independent random variables which are uniformly distributed on the interval $[0, \bar{d}]$. The distribution of the maximum can be described in terms of the cumulative distribution functions (cdf) $F(x) \stackrel{\text{def}}{=} \text{Prob}(X \leq x)$.

For the uniformly distributed variable a_1 , we have $F_1(x) = \text{Prob}(a_1 \leq x) = \frac{x}{\bar{d}}$. Similarly, $F_2(x) = \text{Prob}(a_2 \leq x) = \frac{x}{\bar{d}}$. For the maximum $a_s = \max(a_1, a_2)$, we have

$$F_s(x) = \text{Prob}(a_s \leq x) = \text{Prob}(\max(a_1, a_2) \leq x).$$

Since the maximum of the two numbers is smaller than or equal to x if and only both of these numbers are $\leq x$, we conclude that $F_s(x) = \text{Prob}((a_1 \leq x) \& (a_2 \leq x))$. The variables a_1 and

a_2 are independent, so

$$F_s(x) = \text{Prob}(a_1 \leq x) \cdot \text{Prob}(a_2 \leq x) = \frac{x}{\bar{d}} \cdot \frac{x}{\bar{d}} = \frac{x^2}{(\bar{d})^2}. \quad (10)$$

From this cdf, we can compute the corresponding pdf $\rho_s(x)$:

$$\rho_s(x) = \frac{dF_s(x)}{dx} = \frac{2x}{(\bar{d})^2}. \quad (11)$$

Thus, the desired variance σ_s^2 has the form

$$\sigma_s^2 = \int_0^{\bar{d}} x^2 \cdot \frac{2x}{(\bar{d})^2} dx = \frac{2}{(\bar{d})^2} \cdot \int_0^{\bar{d}} x^3 dx = \frac{2}{(\bar{d})^2} \cdot \frac{1}{4} \cdot (\bar{d})^4 = \frac{1}{2} \cdot (\bar{d})^2. \quad (12)$$

Thus,

$$\sigma_s = \sqrt{\frac{1}{2}} \cdot \bar{d}. \quad (13)$$

Final step: checking that the desired inequality (7) is indeed satisfied. Now that we have expressions (9) and (13) for the sizes σ_{12} and σ_s , we can plug them into the inequality (7) and check that this inequality is satisfied – i.e., that within the desired accuracy of 20%, adding the second neuron, on average, does not matter. Indeed, substituting expressions (9) and (13) into the left-hand side of the formula (7) and dividing both the numerator and the denominator by the common factor \bar{d} , we get the ratio

$$r = \left| \frac{\sqrt{\frac{2}{3}} - \sqrt{\frac{1}{2}}}{\sqrt{\frac{1}{2}}} \right|.$$

Dividing both terms in the numerator by the denominator, we get

$$r = \left| \sqrt{\frac{4}{3}} - 1 \right| = \left| \frac{2}{\sqrt{3}} - 1 \right| = \left| \frac{2}{3} \cdot \sqrt{3} - 1 \right|.$$

For $\sqrt{3} = 1.73\dots$, we get

$$r = \left| \frac{2 \cdot 1.73\dots}{3} - 1 \right| = \left| \frac{3.46\dots}{3} - 1 \right| = |1.15\dots - 1| = 0.15\dots < 0.2.$$

The statement is proven.

3 Conclusion

Thus, we have explained why Fuzzy Cognitive Maps (i.e., 1-neuron neural networks) are adequate for describing the dependence between the experts' degree of confidence, when the relative accuracy of 20% is quite sufficient.

Comment.

We have proven that, *on average*, the relative error does not exceed 20%. This explains why Fuzzy Cognitive Maps are efficient in many practical situations. However, the fact that this inequality is satisfied on average does not necessarily mean that it is satisfied always. There may be cases when Fuzzy Cognitive Maps do not work that well – in this case, it makes sense to describe the corresponding dependencies $x = f(x_1, \dots, x_n)$ by generic (multi-neuron) neural networks.

Acknowledgment

This work was supported in part by the National Science Foundation grants HRD-0734825 and HRD-1242122 (Cyber-ShARE Center of Excellence) and DUE-0926721. This work was performed when C. Stylios was a Visiting Researcher at the University of Texas at El Paso.

Bibliography

- [1] C.M. Bishop (2006), *Pattern Recognition and Machine Learning*, Springer, New York.
- [2] E. Bourgani, C.D. Stylios, G. Manis, V. Georgopoulos (2013), A study on Fuzzy Cognitive Map Structures for Medical Decision Support Systems, *Proc. of the 8th Conf. of the European Society for Fuzzy Logic and Technology EUSFLAT'2013*, Milano, Italy, September 11-13, 744-751.
- [3] Y. Boutalis, T. Kottas, M. Christodoulou M. (2009), Adaptive Estimation of Fuzzy Cognitive Maps with Proven Stability and Parameter Convergence, *IEEE Trans. on Fuzzy Systems*, 17(4): 874-889.
- [4] B. Chokr, V. Kreinovich (1994), How Far Are We from the Complete Knowledge: Complexity of Knowledge Acquisition in Dempster-Shafer Approach, In: R.R. Yager, J. Kacprzyk, M. Pedrizzi (Eds.), *Advances in the Dempster-Shafer Theory of Evidence*, Wiley, New York, 555-576.
- [5] E.T. Jaynes, G.L. Bretthorst (2003), *Probability Theory: The Logic of Science*, Cambridge University Press, Cambridge, UK.
- [6] G. Klir, B. Yuan (1995), *Fuzzy Sets and Fuzzy Logic*, Prentice Hall, Upper Saddle River, New Jersey.
- [7] C. Knight, D. Lloyd, A. Penn (2014), Linear and Sigmoidal Fuzzy Cognitive Maps: An Analysis of Fixed Point, *Applied Soft Computing*, 15: 193-202.
- [8] B. Kosko (1986), Fuzzy Cognitive Maps, *International J. of Man-Machine Studies*, 7: 65-75.
- [9] V. Kreinovich, A. Bernat (1994), Parallel Algorithms for Interval Computations: An Introduction, *Interval Computations*, 1994(3):3, 6-62.
- [10] V. Kreinovich, C. Quintana (1991), Neural Networks: What Non-Linearity to Choose?, *Proc. of the 4th University of New Brunswick Artificial Intelligence Workshop*, Fredericton, New Brunswick, Canada, 627-637.
- [11] W. Lu, J. Yang, X. Liu (2014), Numerical Prediction of Time Series Based on FCMs with Information Granules, *International J. of Computers Communications & Control*, 9(3): 313-324.
- [12] Y. Miao, C.Y. Miao, X.H. Tao, Z.Q. Shen, Z.Q. Liu (2010), Transformation of Cognitive Maps, *IEEE Trans. on Fuzzy Systems*, 18(1): 114-124.
- [13] G.A. Miller (1956), The Magical Number Seven, Plus or Minus Two: Some Limits on Our Capacity for Processing Information", *Psychological Review*, 63(2): 81-97.
- [14] H.T. Nguyen, V. Kreinovich, *Applications of Continuous Mathematics to Computer Science*, Kluwer, Dordrecht.

-
- [15] H.T. Nguyen and E.A. Walker (2006), *A First Course in Fuzzy Logic*, Chapman and Hall/CRC, Boca Raton, Florida.
- [16] E. Papageorgiou, C. Stylios (2008), Fuzzy Cognitive Maps, In: W. Pedrycz, A. Skowron, V. Kreinovich, *Handbook of Granular computing*, John Wiley & Sons, 755-776.
- [17] E.I. Papageorgiou, C. Stylios, P.P. Groumpos (2006), Introducing Interval Analysis in Fuzzy cognitive Map Framework, In: G. Antoniou et al. (eds), *Proc. of the 4th Hellenic Conf. on Artificial Intelligence SETN'2006*, Heraklion, Crete, May 18-20, 2006, Springer Lecture Notes in Artificial Intelligence, 3955: 571-575.
- [18] W. Pedrycz (2010), The Design of Cognitive Maps: A Study in Synergy of Granular Computing and Evolutionary Optimization, *Expert Systems with Applications*, 37(10): 7288-7294.
- [19] W. Pedrycz, W. Homenda, From Fuzzy Cognitive Maps to Granular Cognitive Maps, *IEEE Trans. on Fuzzy Systems*, 22(4): 859-869.
- [20] Y.G. Petalas, E.I. Papageorgiou, K.E. Parsopoulos, P.P. Groumpos, M.N. Vrahatis (2005), Interval Cognitive Maps, *Proc. of Intl. Conf. of Numerical Analysis and Applied Mathematics ICNAAM'05*, Rhodes, Greece, September 16-20, 2005, 1120-1123.
- [21] S.K. Reed (2010), *Cognition: Theories and Application*, Wadsworth Cengage Learning, Belmont, California.
- [22] J.T. Rickard, J. Aisbett, R.R. Yager (2015), A New Fuzzy Cognitive Map Structure Based on the Weighted Power Mean, *IEEE Trans. on Fuzzy Systems*.
- [23] O. Sirisaengtaksin, V. Kreinovich, H.T. Nguyen (1995), Sigmoid Neurons Are the Safest Against Additive Errors, *Proc. of the First Intl. Conf. on Neural, Parallel, and Scientific Computations*, Atlanta, Georgia, May 28-31, 1995, 1: 419-423.
- [24] H.J. Song, C.Y. Miao, Z.Q. Shen, W. Roel, D.H. Maja, C. Francky (2010), Design of Fuzzy Cognitive Maps Using Neural Networks for Predicting Chaotic Time Series, *Neural Networks*, 23: 1264-1275.
- [25] H. Song, C. Mia, R. Wuyts, Z. Shen, M. Hondt, F. Catthoor (2011), An Extension to Fuzzy Cognitive Maps for Classification and Prediction, *IEEE Trans. on Fuzzy Systems*, 19(1): 116-135.
- [26] W. Stach, L. Kurgan, W. Pedrycz (2008), Numerical and Linguistic Prediction of Time Series With the Use of Fuzzy Cognitive Maps, *IEEE Trans. on Fuzzy Systems*, 16(1): 61-72.
- [27] C.D. Stylios, P.P. Groumpos (2000), Fuzzy Cognitive Maps in Modeling Supervisory Control Systems, *Journal of Intelligent & Fuzzy Systems*, 8(2): 83-98.
- [28] C. Stylios, P. Groumpos (2004), Modeling Complex Systems Using Fuzzy Cognitive Maps, *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, 34(1): 155-162.
- [29] R. Taber (1991) Knowledge Processing with Fuzzy Cognitive Maps, *Expert Systems with Applications*, 2(1): 83-87.
- [30] R. Trejo, V. Kreinovich, I.R. Goodman (2002), J. Martinez, R. Gonzalez, A Realistic (Non-Associative) Logic And a Possible Explanations of 7 ± 2 Law, *International Journal of Approximate Reasoning*, 29: 235-266.
- [31] L.A. Zadeh (1965), Fuzzy Sets, *Information and Control* 8: 338-353.