HYBRID SYSTEMS OF COMPUTATIONAL INTELLIGENCE EVOLVED FROM SELF-LEARNING SPIKING NEURAL NETWORK

Yevgeniy Bodyanskiy, Artem Dolotov

***Abstract**: Computational intelligence paradigm covers several approaches for technical problems solving in an intelligence manner, such as artificial neural networks, fuzzy logic systems, evolutionary computation, etc. Each approach provides engineers and researchers with the smart and powerful tools to handle various real-life concerns. Even more powerful tools were designed at the joint of different computational intelligence approaches. Neuro-fuzzy systems, for example, are well-known and advanced intelligent tool that combines capabilities of neural networks and fuzzy systems together in a synergetic way. Among them, one of the prominent hybrid systems type is self-learning fuzzy spiking neural networks. They were evolved from fuzzy logic systems and self-learning spiking neural networks, and revealed considerable computational capabilities. There were proposed several architectures of self-learning fuzzy spiking neural networks, each handling a particular kind of data processing tasks (processing fuzzy data, fuzzy probabilistic and possibilistic clustering, batch and adaptive methods, new clusters detection, irregular form clusters detection, etc). In this paper, known architectures of self-learning hybrid systems based on spiking neural network are reviewed, compared, and summarized. A generalized architecture and learning algorithm for self-learning fuzzy spiking neural networks are proposed.*

***Keywords**: computational intelligence, hybrid systems, self-leaning spiking neural network, fuzzy clustering, temporal Hebbian learning, 'Winner-Takes-More' rule, control theory, inductive modelling, clusters merging, new clusters detection, hierarchical clustering .*

***ACM Classification Keywords**: I.2.6 [Artificial Intelligence]: Learning – Connectionism and neural nets; I.2.8 [Artificial Intelligence]: Problem Solving, Control Methods, and Search – Control theory; I.5.1 [Pattern Recognition]: Models – Fuzzy set, Neural nets; I.5.3 [Pattern Recognition]: Clustering – Algorithms.*

Introduction

Same as any well-established science, each of primary scientific directions of computational intelligence paradigm develops in two fundamental manners, namely, in intensive way – by improving present methods and approaches, and in extensive way – by adopting achievements of adjacent directions. Considering artificial neural networks direction, prominent result of the former is the third generation of neural networks, known as spiking neural networks [Gerstner, 2002; Maass, 1997]. Well-known result of the latter is hybrid neuro-systems, among them the most advanced are neuro-fuzzy systems [Jang, 1997].

Neuro-fuzzy systems successfully combine capabilities of neural networks of the second generation and fuzzy systems in a synergetic way and provide researchers and engineers with powerful tools to handle various technical problems. Following this notion, hybrid systems of self-learning spiking neural networks and fuzzy logic methods were introduced and shown to be effective in data processing [Bodyanskiy, 2009b]. The third generation of artificial neural networks appeared to be capable of being combined with fuzzy logic approach on different levels of architecture, and that led to different types of hybrid systems, each solving a certain kind of data processing tasks. Having particular hybrid systems evolved from self-learning spiking neural network, the synthesis of generalized architecture of self-learning fuzzy spiking neural network is of theoretical interest, and that is the main purpose of the paper.

Before proceeding any further, it is worth to note that spiking neural networks differ from neural networks of the second generation essentially. Whereas conventional neural networks are driven by value of signal and disregard time in the most cases, spiking neural networks utilize time as the only computational resource, and that makes the third generation of neural networks related to biological neural systems in higher degree than networks of the previous generations are. Such difference becomes sharper if one considers both generations from control theory point of view. In this case, conventional artificial neural networks are nothing other than pulse-amplitude systems, and spiking neural networks are pulse-position ones [Bodyanskiy, 2009b]. Hybrid systems evolved from spiking neural networks introduce concept of time as a computational resource through spiking neural networks into other areas of computational intelligence, and that will confidently lead to new discoveries in future.

Self-Learning Spiking Neural Networks

The first notion of spiking neurons as self-learning systems was stated by J. Hopfield in [Hopfield, 1995]. He discovered that spiking neuron acts as a radial basis function, namely, its response depends on distance between incoming pattern and one encoded into the neuron's synapses (it is named as spiking neuron center). The closer pattern is to the center, the earlier neuron fires. If pattern is far from the center, the neuron does not fire at all. Here the distance is treated not in common sense, it reflects degree of coincidence (synchrony) of spikes emitted in synapses and incoming to soma. Among the simple ways to calculate the distance, there are two commonly known ones, they are generalized in [Bodyanskiy, 2009d].

By adjusting synaptic weights of spiking neuron, one can encode center of a certain data cluster into neuron's center. Several neurons adjusted this way and linked with lateral inhibitory connections allowing only one neuron to fire (the one that is closer to incoming pattern) can successfully perform crisp data clustering. The first known architecture of network with spiking neurons layer that is applicable for data processing tasks solving was proposed by T. Natschlaeger and B. Ruf in [Natschlaeger, 1998]. The network consists of input neurons layer that transforms input patterns into spikes and spiking neurons layer that performs data clustering. The proposed self-learning spiking neural network is able to detect clusters in input data set successfully only in cases when number of clusters does not exceed number of input dimensions. S. Bohte, J. Kok, and H. La Poutre improved spiking neural network clustering capabilities by introducing the layer of population coding [Bohte, 2002] instead of layer of Natschlaeger-Ruf input neurons. Now Bohte's self-learning spiking neural network is the base for any further modifications and improvements.

It is remarkable that spiking neuron functioning can be described from control theory perspective, in terms of Laplace transform [Bodyanskiy, 2009b]. By this means spiking neuron synapse is nothing other than a second-order critically damped response unit, and spiking neuron soma is a threshold-detection unit. Description of spiking neuron functioning in terms of the Laplace transform makes it possible to state spiking neural network architecture on a general technical ground that can be used in the following for constructing various hardware implementations of self-learning spiking neural network and hybrid systems evolved from it.

Fuzzy Clustering Based on Spiking Neurons

Fuzzy clustering is the more advanced approach for unsupervised data processing as compared to crisp clustering. It allows of successful data processing under a prior and current uncertainty thus providing researchers and engineers with flexible tool to handle various complex technical problems. Fuzzy clustering analysis covers several techniques. Among them, algorithms based on objective function are the most mathematically rigorous [Bezdek, 1981]. Such algorithms solve data processing tasks by optimizing a preset cluster quality criterion. As a rule, the criterion rests on a certain distance between input patterns and cluster centers. Considering different distance metrics, one can originate various fuzzy clustering algorithms. For example, Euclidean metric originates well-known and widely used conventional FCM and PCM algorithms [Bezdek, 2005].

Within scope of the paper subject, it is instructive to find the way of how self-learning spiking neural networks can be combined with fuzzy clustering approaches. Such uniting factor was originally proposed in [Bodyanskiy, 2008a] on the base of spiking neurons firing time. As far back as the first architectures of self-learning spiking neural network, it has been noted that firing time of spiking neuron accounts for similarity ([Natschlaeger, 1998]) or distance ([Bohte, 2002]) between the neuron center and input pattern. Using this notion, fuzzy clustering algorithm was introduced for spiking neurons layer as follows:

$$\mu_j(x(k)) = \frac{\left(t_j^{[1]}(x(k))\right)^{\frac{2}{1-\zeta}}}{\sum_{\iota=1}^{m}\left(t_\iota^{[1]}(x(k))\right)^{\frac{2}{1-\zeta}}}, \tag{1}$$

where $x(k)$ is the input pattern ($k = 1, 2, ..., k, ..., N$ is a pattern number), $\mu_j(x(k))$ is the membership level of pattern $x(k)$ to the j-th cluster, $t_j^{[1]}(x(k))$ is the firing time of the j-th spiking neuron in the layer ('1' indicates number of spiking neuron layer, layer of population coding is marked by '0'), $\zeta \geq 0$ is the fuzzifier, m is number of clusters (and number of spiking neurons in the layer).

Hybrid system based on self-learning spiking neural network for accomplishing fuzzy clustering task (self-learning fuzzy spiking neural network) includes three layers, namely, population coding layer, spiking neurons layer (lateral connections are eliminated), and layer that performs fuzzy partitioning (1). Obviously lateral connections in the second hidden layer should be eliminated in order fuzzy clustering layer to receive information on distances from all spiking neurons. The major advantage of the proposed network is its rapidity in data clustering as compared to conventional FCM algorithm. As it was shown [Bodyanskiy, 2008a-b], fuzzy spiking neural network requires about 3-4 epochs to successfully process satellite images while FCM requires about 30-50 epochs for the same tasks. One more advantage of the hybrid system is that it needs not to calculate cluster's centroids explicitly thus simplifying the clustering algorithm.

Developing the proposed notion further, possibilistic clustering on self-learning spiking neural network base was put forth [Bodyanskiy, 2008d]. Its form is similar to PCM algorithm but firing time is used as distance between input pattern and cluster center:

$$\mu_j(x(k)) = \left(1 + \left(\frac{\left(t_j^{[1]}(x(k))\right)^2}{\lambda_j}\right)^{\frac{1}{\zeta-1}}\right)^{-1}, \tag{2}$$

$$\lambda_j = \frac{\sum_{k=1}^{N}\mu_j^{\zeta}(x(k))\left(t_j^{[1]}(x(k))\right)^2}{\sum_{k=1}^{N}\mu_j^{\zeta}(x(k))}, \tag{3}$$

where $\lambda_j \geq 0$ is j-th penalty term.

Both (1) and (2), (3) are algorithms for data processing in batch mode that implies the data set is defined before neural network learning stage. Every so often new data arrive in the course of learning (on-line mode). This case requires adaptive form of clustering algorithms. Utilizing adaptive version of conventional PCM [Bodyanskiy, 2005] in the same way as it was shown above for batch mode PCM, self-learning spiking neural network can be tuned to accomplish possibilistic clustering in on-line mode [Bodyanskiy, 2008c].

The proposed modification in spiking neurons layer requires the network learning algorithm to be updated. The original learning algorithm of self-learning spiking neural network is based on two rules, namely, on 'Winner-Takes-All' rule and temporal Hebbian rule [Gerstner, 1996; Natschlaeger, 1998; Bohte, 2002; Berredo, 2005]. 'Winner-Takes-All' rule is applied to define which spiking neuron to be updated: the one that fired first for the provided input pattern. Temporal Hebbian rule is used to update the neuron so that its center is moved closer to input pattern. A learning epoch of the algorithm consists of two phases. Spiking neurons compete to respond to input pattern on the first phase. Synaptic weights of the neuron-winner are adjusted on the second phase. Centers of spiking neurons of the learned network represent centriods of data clusters so the network can successfully perform crisp data clustering. In order to accomplish fuzzy clustering (1) or (2), (3), the output layer should received information on distances from input patters to all spiking neurons centers so evidently the learning algorithm should take into account that information on each learning epoch as well. Such consideration leads us to the generalized version of learning algorithm [Bodyanskiy, 2009b] where 'Winner-Takes-All' rule is replaced with 'Winner-Takes-More' rule as follows:

$$w^p_{jli}(K+1) = w^p_{jli}(K) + \eta_w(K)\varphi(\left|\Delta t_{\tilde{j}j}\right|)L(\Delta t^p_{jli}), \tag{4}$$

where K is the epoch number, w^p_{jli} is the synaptic weight of the p-th subsynapse between the li-th receptive neuron and the j-th spiking neuron, $\eta_w(\bullet) > 0$ is the learning rate, $\varphi(\bullet)$ is the neighbourhood function, $\left|\Delta t_{\tilde{j}j}\right|$ is the difference between the neuron-winner $\tilde{j}$ firing time and the j-th neuron firing time, $L(\bullet)$ is the learning function [Gerstner, 1996], Δt^p_{jli} is the time delay between delayed spike $t^{[0]}_{li}(x_i(k)) + d^p$ and spiking neuron spike $t^{[1]}_j(x(k))$. Learning algorithm (4) naturally fits spiking neural network architecture where lateral inhibitory connections are eliminated.

Thus, as is evident from the foregoing, layer of spiking neurons can be easily "fuzzified". Firing time of spiking neuron appeared to be a good ground to link new type of neural networks and conventional clustering algorithms considerably reducing data processing time. This is the most obvious and easy to implement notion of evolving hybrid systems from self-learning spiking neural network. More sophisticated approached are stated in the following sections.

Layer of Fuzzy Receptive Neurons

Except the main purpose of input layer of spiking neural network that is transforming input signal into spikes, population coding layer of Bohte's self-learning spiking neural network was designed also to improve the network computational capabilities [Bohte, 2002]. Population coding layer consists of several pools of receptive neurons. Number of pools corresponds to number of components (dimensions) of input signal. Each pool of receptive neurons processes a certain input component, thus distributing information over pool neurons and increasing the input dimensionality.

Receptive neuron is so constructed that input signal of either pulse-amplitude or continuous-time form is transformed to pulse-position form. Receptive neuron's activation function is bell-shaped. The closer input signal is to the activation function center, the earlier receptive neuron emits spike. Generally the transformation can be expressed as follows (for pulse-amplitude signal):

$$t^{[0]}_{li}(x_i(k)) = \begin{cases} \left\lfloor t^{[0]}_{\max}\left(1 - \psi\left(\left|x_i(k) - c^{[0]}_{li}\right|, \sigma_{li}\right)\right)\right\rfloor, & \psi\left(\left|x_i(k) - c^{[0]}_{li}\right|, \sigma_{li}\right) \geq \theta_{\text{r.n.}}, \\ -1, & \psi\left(\left|x_i(k) - c^{[0]}_{li}\right|, \sigma_{li}\right) < \theta_{\text{r.n.}}, \end{cases} \tag{5}$$

where $x_i(k)$ is the *i*-th component of input pattern $x(k)$, $t_{li}^{[0]}(x_i(k))$ is the spike emitted by the *l*-th receptive neuron of the *i*-th pool, $t_{\max}^{[0]}$ is the maximum possible firing time for a neuron in the input layer, $\lfloor \bullet \rfloor$ is the floor function (ceiling function can be used as well), $\psi(\bullet,\bullet)$, $c_{li}^{[0]}$, σ_{li}, and $\theta_{r.n.}$ are respectively the receptive neuron's activation function, center, width, and dead zone, -1 is used to indicate not fired neuron.

Originally it was proposed to set receptive neurons in a pool so that their activation functions were uniformly shifted and overlapped [Bohte, 2002; Berredo, 2005]. In addition neurons with narrow and wide activation functions were considered. One can see that the way of receptive neurons setting has nothing to do with the input data nature. In the absence of a priori knowledge of problem being solved, the cited method is acceptable. However, if researcher or engineer possess some knowledge regarding the problem, it would be worthwhile to consider it when setting neural network parameters in order to improve the network performance.

It is common knowledge that fuzzy logic rule-based systems allow for ease of their strucutre interpretation, thus providing researchers with ability to incorporate knowledge of the problem being solved into the system. Let's look at population coding layer of spiking neural network from fuzzy inference systems perspective [Bodyanskiy, 2009b]. One can readily see that pool of receptive neurons is closely similar to a certain linguistic variable, and receptive neuron in the pool is similar to a linguistic term. Using this notion, we can consider activation function $\psi_{li}(x_i(k))$ of receptive neuron as membership function of corresponding linguistic term. Obviously in this context bell-shaped form is no longer requirement for activation function. It can be trapezoidal, or triangular, or any other that is the best suited to express expert knowledge. By this means population coding layer can be treated as the one that transforms input data set to a fuzzy set that is defined by values of activation-membership functions $\psi_{li}(x_i(k))$ and is expressed over time domain in form of firing times $t_{li}^{[0]}(x_i(k))$. Firing time $t_{li}^{[0]}(x_i(k)) = 0$ indicates that input $x_i(k)$ belongs to the *l*-th linguistic term of the *i*-th linguistic variable in the highest degree. And vice versa firing time $t_{li}^{[0]}(x_i(k)) = -1$ means $x_i(k)$ does not belong to the linguistic term at all.

Thus, interpretation of population coding layer as a fuzzification layer allows researchers to incorporate a priori knowledge into layer structure, and that may be resulted in better clusters partitioning. Here it makes sense to note that the preset linguistic terms can be occasionally lacking of necessary density of overlapping activation-membership functions. In this case researcher should observe balance between customizing activation functions of receptive neurons regarding his knowledge and keeping the required density, but that is the subject for another research.

Adjustable Architectures of Self-Learning Spiking Neural Network

Efficiency of data clustering depends to a large degree on quality of corresponding mathematical model chosen by researcher. One of the challenging problems in area of effective mathematical model designing is varying number of clusters in data being processed. That case can be successfully handled by adaptive models that are capable of adjusting not only their parameters but also their structure. Idea of designing optimal in some respect structures underlies the scientific direction known as inductive modelling. From this point of view, it may be supposed that combining methods of inductive modelling and computational intelligence together would increase quality of data processing in the presence of varying number of clusters or when clusters are of irregular form. It is possible to apply the following inductive modelling methods to self-learning spiking neural network:

- clusters merging by decreasing number of spiking neurons;
- new clusters detecting by adding new spiking neurons to layer;
- hierarchical clustering on the base of multilayered self-learning spiking neural network.

One of the efficient methods for clusters merging was proposed by Yo. Nakamori and M. Ryoke [Nakamori, 1996] the essence of which was minimizing sum of concentration hyperellipsoid volumes. In [Bodyanskiy, 2009c], Nakamori-Ryoke merging procedure was improved on the base of experimental design solutions and successfully applied to self-learning fuzzy spiking neural network. Thus, the proposed hybrid system overcame the problem of necessity to set correct number of clusters initially, before data processing. Researcher just needs to set a certain number of clusters that is exactly higher than actual number of clusters, and the merging procedure will automatically decrease number of spiking neurons to the proper one.

New clusters detection procedure for self-learning spiking neural network was proposed in [Dolotov, 2010] on the basis of fuzzy possibilistic clustering algorithm. The procedure analyses output of fuzzy clustering layer on each step of data processing, and if the sum of membership levels over all clusters is less than the preset threshold, a new neuron is added to spiking neurons layer. The proposed hybrid system can handle new clusters only in case if patterns of one new cluster only appear at a time.

Multilayered spiking neural network for hierarchical clustering (and for complex clusters detecting correspondingly) was originally proposed in [Bohte, 2002]. Feed-forward adjustable lateral connections in hidden layers of spiking neurons and learning algorithm for them were suggested in the architecture. The spiking neural network brought out remarkable results in detecting clusters of irregular form. Nonetheless improved architecture of multilayered self-learning spiking neural network was proposed later on [Bodyanskiy, 2009a]. Instead of two learning algorithms (one for interlayer connections, another for lateral connections), it was proposed to use the generalized learning algorithm (4). Updating not only neuron-winner, but also its neighbours made it possible to eliminate lateral connection from the network and thus to simplify its architecture.

A Generalized Architecture of Self-Learning Fuzzy Spiking Neural Network

A generalized architecture of self-learning fuzzy spiking neural network that incorporates approaches to design hybrid systems on the base of spiking neural network that are outlined above is depicted on Figure 1.

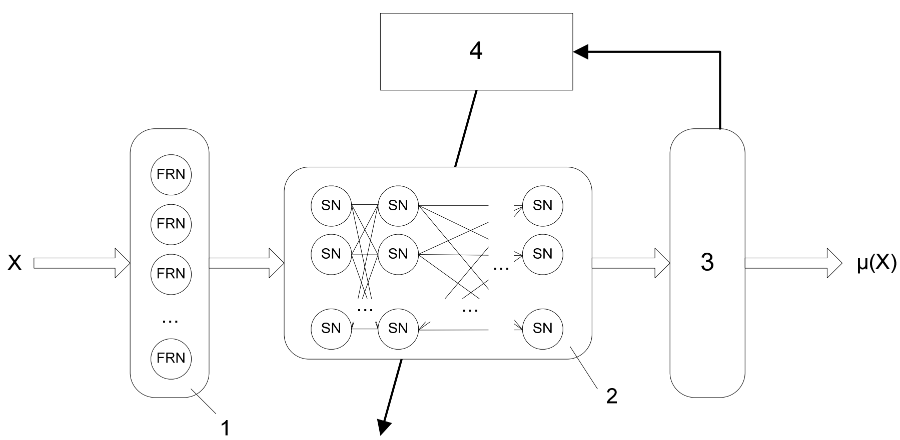

Figure 1. Architecture of a generalized self-learning fuzzy spiking neural network: 1 is the fuzzification layer; 2 is the hierarchical clustering subsystem; 3 is the output fuzzy clustering layer; 4 is the subsystem for adaptive number of clusters controlling; X is the input data set; FRN is a fuzzy receptive neuron; SN is a spiking neuron; $\mu(X)$ *is the fuzzy partition of X*

Its first layer (fuzzification layer) transforms input data into spikes having regard to a priori knowledge of the problem being solved. In the absence of such knowledge, fuzzy receptive neurons are set automatically, and the layer becomes properly a conventional population coding layer. Hierarchical clustering subsystem consists of several layers of spiking neurons. Number of layers depends on the data processing task complexity and again on a priori knowledge of the problem. The output fuzzy clustering layer performs fuzzy partitioning of input data either using probabilistic (1) or possibilistic (2), (3) approach in batch or adaptive mode. The subsystem for adaptive number of clusters controlling adjusts number of spiking neurons in the last layer of the hierarchical clustering subsystem. The hybrid system is learned with the generalized learning algorithm (4). If it is suggested to implement the system on hardware base, the Laplace transform description of the system functioning can be used.

On the one hand, the described hybrid system architecture is a generalized sketch allowing researchers and engineers to derive a certain specific hybrid system subject to the problem being considered. On the other hand, it can be used for solving complex real-life problems where several approaches are required to be applied.

It is necessary to stress that the proposed architecture is not absolutely complete. There are still some open questions. Interpretation of firing time of fuzzy receptive neurons output as membership level can be further developed. Suggestion criteria to select proper number of layers of spiking neurons and methods to control number of neurons in each of the layers would be a significant improvement. Other approaches for merging and new clusters detection can be considered for the system as well.

Conclusion

The known methods to construct self-learning hybrid systems on spiking neural network base are reviewed. The generalized architecture of self-learning fuzzy spiking neural network is proposed. Further directions of research are suggested.

Bibliography

[Berredo, 2005] R.C.de Berredo. A review of spiking neuron models and applications. M. Sc. Dissertation. Pontifical Catholic University of Minas Gerais, Belo Horizonte, 2005.

[Bezdek, 1981] J.C.Bezdek. Pattern Recognition with Fuzzy Objective Function Algorithms. Plenum Press, New York, 1981.

[Bezdek, 2005] J.C.Bezdek, J.Keller, R.Krishnapuram, N.R.Pal. Fuzzy Models and Algorithms for Pattern Recognition and Image Processing. Springer, New York, 2005.

[Bodyanskiy, 2005] Ye.Bodyanskiy. Computational intelligence techniques for data processing. Lecture Notes in Informatics, Vol. 72, 2005, P. 15-36.

[Bodyanskiy, 2008a] Ye.Bodyanskiy, A.Dolotov. A self-learning spiking neural network for fuzzy clustering task. Sci. Proc. of Riga Technical University: Information Technology and Management Science, 36, 2008, P. 27-33.

[Bodyanskiy, 2008b] Ye.Bodyansky, A.Dolotov. Image processing using self-learning fuzzy spiking neural network in the presence of overlapping classes. Proc. 11th Int. Biennial Baltic Electronics Conf. "BEC 2008" (October 6-8, 2008, Tallinn/Laulasmaa, Estonia), Tallinn University of Technology, Tallinn, 2008, P. 213-216.

[Bodyanskiy, 2008c] Ye.Bodyanskiy, A.Dolotov, I.Pliss. Adaptive possibilistic clustering algorithm based on spiking neural network. Proc. 4th Int. School-Seminar “Theory of Decision Making” (September 29 – October 4, 2008, Uzhhorod, Ukraine), Uzhhorod National University, Uzhhorod, 2008, P. 25-26 (in Ukrainian).

[Bodyanskiy, 2008d] Ye.Bodyanskiy, A.Dolotov, I.Pliss, Ye.Viktorov. Fuzzy possibilistic clustering using self-learning spiking neural network. Proc. 15th East West Fuzzy Colloquium (September 17-19, 2008, Zittua, Germany), Wissenschaftliche Berichte der Hochschule Zittau/Goerlitz, Heft 100, Nr. 2360-2395, 2008, S. 53-60.

[Bodyanskiy, 2009a] Ye.Bodyanskiy, A.Dolotov. A multilayered self-learning spiking neural network and its learning algorithm based on ‘Winner-Takes-More’ rule in hierarchical clustering. Sci. Journal of Riga Technical University: Information Technology and Management Science, 40, 2009, P. 66-74.

[Bodyanskiy, 2009b] Ye.Bodyansky, A.Dolotov. Analog-digital self-learning fuzzy spiking neural network in image processing problems. In: Image Processing. Ed. Yu.-S.Chen. INTECH, Vukovar, 2009.

[Bodyanskiy, 2009c] Ye.Bodyanskiy, A.Dolotov. Fuzzy spiking neural network and design of experiments alphabetical optimality criteria using in data clustering problem. Proc. Int. Conf. "Intellectual Systems for Decision Making and Problems of Computational Intelligence (ISDMCI'2009)" (May 18-22, 2009, Yevpatoria, Ukraine), Kherson National Technical University, Kherson, Vol. 2, 2009, P. 154-157.

[Bodyanskiy, 2009d] Ye.Bodyanskiy, A.Dolotov, I.Pliss. Adaptive Gustafson-Kessel fuzzy clustering algorithm based on self-learning spiking neural network. In: Intelligent Information and Engineering Systems. Eds. G.Setlak, K.Markov. Int. book series “Information Science and Computing”, No. 13. Rzeszow University of Technology – Institute of Information Theories and Applications FOI ITHEA, Rzeszow – Sofia, 2009, P. 17-24.

[Bohte, 2002] S.M.Bohte, J.S.Kok, H.La Poutre. Unsupervised clustering with spiking neurons by sparse temporal coding and multi-layer RBF networks. IEEE Trans. on Neural Networks, 13, 2002, P. 426-435.

[Dolotov, 2010] A.Dolotov, I.Pliss. Spiking neural model for new clusters detection in sequential mode. Proc. 3rd Int. Conf. on Inductive Modelling “ICIM 2010” (May 16-22, 2010, Yevpatoria, Ukraine), Kherson National Technical University, Kherson, 2010, P. 107-112.

[Gerstner, 1996] W.Gerstner, R.Kempter, J.L.van Hemmen, H.Wagner. A neuronal learning rule for sub-millisecond temporal coding. Nature, 383, 1996, P. 76-78.

[Gerstner, 2002] W.Gerstner, W.M.Kistler. Spiking Neuron Models. The Cambridge University Press, Cambridge, 2002.

[Hopfield, 1995] J.J.Hopfield. Pattern recognition computation using action potential timing for stimulus representation. Nature, 376, 1995, P. 33-36.

[Jang, 1997] J.-S.R.Jang, C.-T.Sun, E.Mizutani. Neuro-Fuzzy and Soft Computing. Prentice Hall, Upper Saddle River, 1997.

[Maass, 1997] W.Maass. Networks of spiking neurons: the third generation of neural network models. Neural Networks, 10, 1997, P. 1659-1671.

[Nakamori, 1996] Yo.Nakamori, M.Ryoke. Hyperellipsoidal clustering. In: Fuzzy Modelling: Paradigms and Practice. Ed. W.Pedrycz. Kluwer Academic Publishers, Boston Dordrecht London, 1996.

[Natschlaeger, 1998] T.Natschlaeger, B.Ruf. Spatial and temporal pattern analysis via spiking neurons. Network: Computations in Neural Systems, 9, 1998, P. 319-332.

Authors' Information

Yevgeniy Bodyanskiy *– Professor, Dr.-Ing. habil., Scientific Head of Control Systems Research Laboratory, Kharkiv National University of Radio Electronics, 14 Lenin Ave., Office 511, 61166 Kharkiv, Ukraine; e-mail: bodya@kture.kharkov.ua*

Major Fields of Scientific Research: Artificial neural networks, Fuzzy systems, Hybrid systems of computational intelligence

Artem Dolotov *– Ph. D. student, Junior Researcher of Control Systems Research Laboratory, Kharkiv National University of Radio Electronics, 14 Lenin Ave., Office 511, 61166 Kharkiv, Ukraine; e-mail: dolotov.artem@gmail.com*

Major Fields of Scientific Research: Spiking neural networks and hybrid systems based on them