Spiking Neural Model for New Clusters Detection in Sequential Mode

Artem Dolotov, Iryna Pliss

Control Systems Research Laboratory, Kharkiv National University of Radio Electronics, Lenin Ave. 14, 61166 Kharkiv Ukraine

artem.dolotov@gmail.com, pliss@kture.kharkov.ua

Abstract. *Self-learning fuzzy spiking neural network is proposed to detect new clusters during data processing. Learning algorithm for the neural networks is proposed as a generalization of well-known temporal Hebbian learning rule on the base of 'Winner-Takes-More' approach.*

Keywords

Inductive modelling, computational intelligence, self-learning spiking neural network, new cluster detection

1 Introduction

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. At the same time, fairly widespread data clustering methods are means of computational intelligence that cover artificial neural networks, fuzzy systems, and hybrid systems of them. 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.

One of the fast and efficient data clustering tools of computational intelligence is self-learning spiking neural networks [1]. Moreover, being more realistic models of biological neural systems than artificial neural networks of the previous generations, spiking neural networks present another step of computational intelligence paradigm evolving toward biologically plausible computing.

Hybrid intelligent systems designed on the base of self-learning spiking neural networks and fuzzy clustering approaches were shown to be successfully applied for data clustering under uncertainty [2]. Within scope of problem being considered, a promising architecture of the system that successfully process data with varying number of clusters would be the one based on fuzzy possibilistic clustering. It is common knowledge that the fuzzy possibilistic method can detect outliers and minimize their impact on data partitioning. As a rule, outlier is a pattern located rather far from all clusters centers. Several patterns are considered to be outliers until their number becomes sufficient (as compared to existing clusters) to recognize them as another cluster. Noteworthy also are the first few patterns of newly appearing cluster that look like outliers. Using this notion, it is possible to handle new clusters during data processing, thus partially solving the problem considered above.

In this paper, architecture of heterogeneous three-layered self-learning fuzzy spiking neural for new clusters detection during data processing is introduced. The network consists of receptive neurons layer, layers of spiking neurons, and output fuzzy clustering layer. The layer of spiking neuron is adjusted by a separate procedure that increases number of spiking neurons each time when a new cluster appears in data being processed.

2 Architecture of the network

Architecture of the proposed neural network is depicted on figure 1.

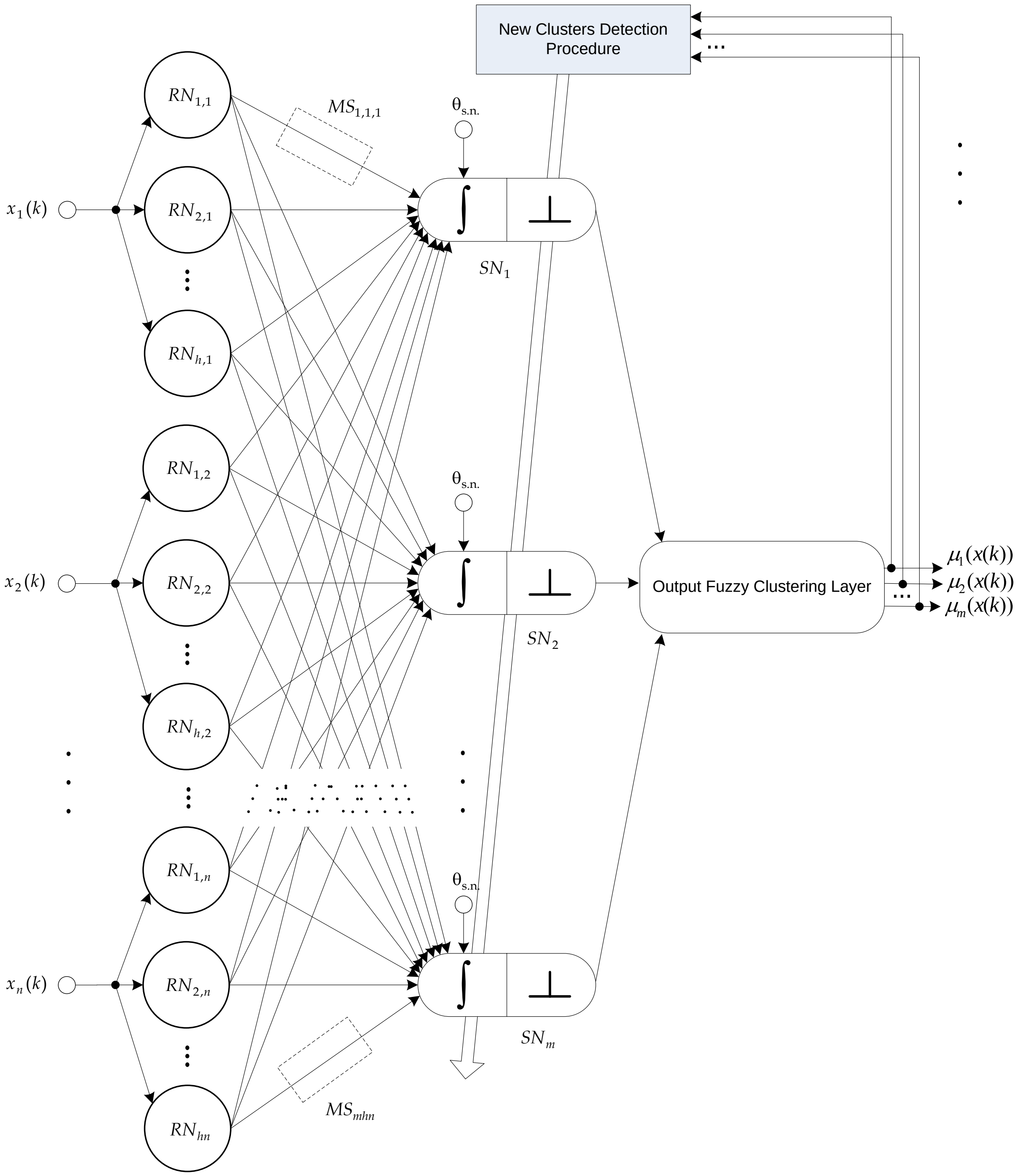

Fig. 1. Self-learning fuzzy spiking neural model for new clusters detection

The first two layers of the proposed neural model is similar to self-learning spiking neural network introduced by S. Bohte [3]. The only difference is that lateral connections in the second hidden layer are eliminated.

The first hidden layer consists of pool of receptive neurons and performs population coding of input signal [3]. Each dimension of input signal is processed by a pool of receptive neurons so the signal dimensionality increases. The population coding can be described in the following general form:

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

where k is a pattern number, $x_i(k)$ is i-th dinesional component of signal $x(k)$, $t_{li}^{[0]}(x_i(k))$ is a spike generated by l-th receptive neuron within pool for $x_i(k)$, $\lfloor \bullet \rfloor$ is the floor function, $\psi(\bullet,\bullet)$, $c_{li}^{[0]}$, σ_{li}, and $\theta_{\text{r.n.}}$ are the receptive neuron's activation function, center, width and dead zone, respectively, -1 indicates that the neuron does not fire, $t_{\max}^{[0]}$ is the length of coding interval.

Internal hidden layers between layer of receptive neurons and output layer are formed by spiking neurons [3]. Spiking neuron is considered to be formed of two constituents, they are: synapse and soma. Synapses of a spiking neuron are multiple structures (figure 2).

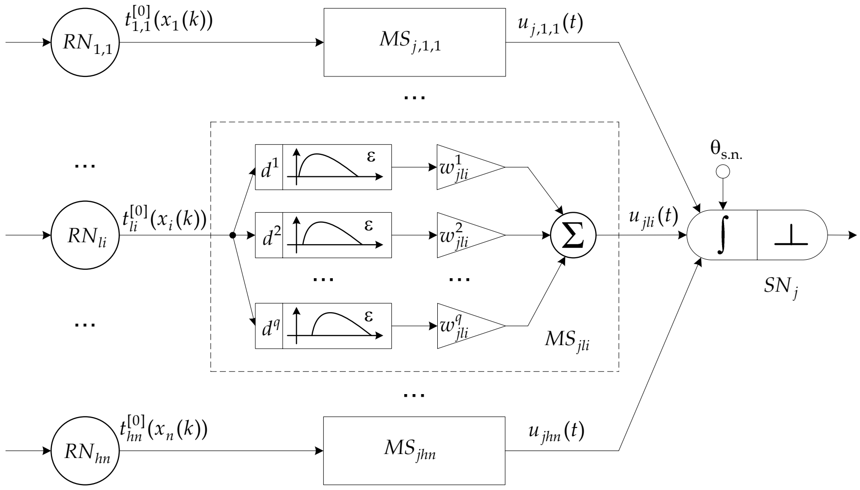

Fig. 2. Multiple synapse

A multiple synapse MS_{jli} (here, j is the number of neuron in the layer) consists of a set of q subsynapses with different time delays d^p, $d^p - d^{p-1} > 0$, $d^q - d^1 > t_{\max}$, and adjustable weights w_{jli}^p. It should be noted that number of subsynapses within a multiple synapse are fixed for the whole network. Having a spike $t_{li}^{[0]}(x_i(k))$ from the li-th receptive neuron, the p-th subsynapse of the j-th spiking neuron produces delayed weighted postsynaptic potential

$$u_{jli}^p(t) = w_{jli}^p \varepsilon_{jli}^p(t) = w_{jli}^p \varepsilon\left(t - \left(t_{jli}^{[0]}(x_i(k)) + d^p\right)\right), \tag{2}$$

where $\varepsilon(\bullet)$ is a spike-response function [2].

Output of the multiple synapse MS_{jli} forms total postsynaptic potential

$$u_{jli}(t) = \sum_{p=1}^{q} u_{jg}^p(t). \tag{3}$$

Each incoming total postsynaptic potential contributes to membrane potential of spiking neuron SN_j as follows:

$$u_j(t) = \sum_{l=1}^{h}\sum_{i=1}^{n} u_{jli}(t), \tag{4}$$

where h and n are the number of receptive neurons in a pool and number of dimensions of input data correspondingly.

Spiking neuron SN_j generates at most one outgoing spike $t_j^{[1]}(x(k))$ during a simulation interval (the presentation of an input pattern $x(k)$), and fires at the instant the membrane potential reaches firing threshold $\theta_{\text{s.n.}}$. After neuron has fired, the membrane potential is reset to the rest value u_{rest} (0 usually) until the next input pattern is presented.

Within the layer, each spiking neuron is assigned to a certain cluster. The first fired neuron defines the cluster that input pattern belongs to (and firing time defines the temporal distance of input pattern to the corresponding cluster center [3]). Thus, a layer of spiking neurons performs crisp data partitioning.

The output layer performs fuzzy partitioning of data using either fuzzy possibilistic approach [2]:

$$\mu_j(x(k)) = \left(1 + \left(\frac{\left(t_j^{[1]}(x(k))\right)^2}{\lambda_j}\right)^{\frac{1}{\zeta-1}}\right)^{-1}, \tag{5}$$

$$\lambda_j = \frac{\sum_{k=1}^{N} \mu_j^{\zeta}(x(k))\left(t_j^{[1]}(x(k))\right)^2}{\sum_{k=1}^{N} \mu_j^{\zeta}(x(k))}, \tag{6}$$

where $\mu_j(x(k))$ is the membership level of pattern $x(k)$, ζ is the fuzzifier (as a rule, $\zeta = 2$), N is the number of pattern in data set.

New clusters detection procedure analyzes output of fuzzy clustering layer on each step of data processing, and if the following criterion is met, adds a new spiking neuron to the layer:

$$\sum_{j=1}^{m} \mu_j(x(k)) \geq \theta_{\text{n.c.}}. \tag{7}$$

where *m* is the number of spiking neurons, $\theta_{\text{n.c.}}$ is the threshold that indicates new cluster (adjusted empirically).

Once several patterns of the new cluster (as a rule, number of patterns should equal average number of patterns over existing clusters) have been detected using criterion (7), another learning stage is initiated to adjust the new spiking neuron to the actual data.

3 Learning Algorithm

The purpose of an unsupervised learning algorithm for network with one layer of spiking neurons is to adjust centers of spiking neurons so as to make each of them to correspond to centroid of a certain data cluster [3]. Such learning algorithm was introduced on the base of two learning rules, namely, 'Winner-Takes-All' rule and temporal Hebbian learning rule [2, 3]. The first one defines which neuron should be updated, and the second one defines how it should be updated. The algorithm adjusts neuron centers through synaptic weights updating on each learning epoch.

A learning epoch consists of two phases. Competition, the first phase defines a neuron-winner. Being laterally linked with inhibitory connections in originally suggested architecture of spiking neural network [3], spiking neurons compete to respond to input pattern. The one wins (and fires) whose center is the closest to the pattern. After competition has been completed, weights adjusting takes place. The learning algorithm adjusts synaptic weights of the neuron-winner to move it closer to the input pattern. It strengthens weights of those subsynapses which contributed to the neuron-winner's firing (i.e. the subsynapses produced delayed spikes right before the neuron firing) and weakens ones which did not contribute (i.e. the delayed spikes appeared right after the neurons firing or long before it).

In this paper, the learning algorithm [3] is generalized using 'Winner-Takes-More' rule that seems to be more natural for fuzzy clustering case as at each epoch all spiking neurons responses are meaningful. This approach implies that there is a cooperation phase before weights adjustment. Neuron-winner determines a local region of topological neighbourhood on each learning epoch. Within this region, the neuron-winner fires along with its neighbours, and the closer a neighbour is to the winner, the more its weights are adjusted. The topological region is represented by the neighbourhood function $\varphi(|\Delta t_{\tilde{j}j}|)$ that depends on difference $|\Delta t_{\tilde{j}j}|$ between the neuron-winner firing time $t_{\tilde{j}}^{[1]}(x(k))$ and the neighbour neuron firing time $t_j^{[1]}(x(k))$ (distance between the neurons is in temporal sense) and a parameter that defines effective width of the region. As a rule, $\varphi(\bullet)$ is a kernel function that is symmetric about its maximum at the point where $\Delta t_{\tilde{j}j} = 0$. It reaches unity at that point and monotonically decreases as $\Delta t_{\tilde{j}j}$ tends to infinity.

The functions that are the most frequently used as neighbourhood function are Gaussian, Epanechnikov, Mexican Hat, and many others [2].

Thus, the learning algorithm based on ‘Winner-Takes-More’ rule can be expressed in the following form for multilayered self-learning spiking neural network:

$$w^p_{jli}(K+1) = w^p_{jli}(K) + \eta_{\mathrm{w}}(K)\varphi(\left|\Delta t_{\tilde{j}j}\right|)L(\Delta t^p_{jli}), \tag{8}$$

where K is a number of epoch, $\eta_{\mathrm{w}}(\bullet) > 0$ is the learning rate [2], $L(\bullet)$ is the learning function [2], Δt^p_{jli} is calculated as follows:

$$\Delta t^p_{jli} = t^{[0]}_{li}(x_i(k)) + d^p - t^{[1]}_j(x(k)). \tag{9}$$

4 Simulation experiment

The proposed fuzzy spiking neural network was tested on satellite image of the Earth surface. Each cluster is indicated by a separate color on the processed image for visualization purposes. RGB pixels of the image were suggested to be input patterns for the spiking neural network. Each component of input patterns were processed by a pool of eight receptive nurons.

At the first stage, a part of the image was used to learn the spiking neural network (figure 3). Eight clusters were originally supposed (i.e. the second hidden layer was formed of eight spiking neurons). Spiking neural network learned at 3 epochs to successfully process the image.

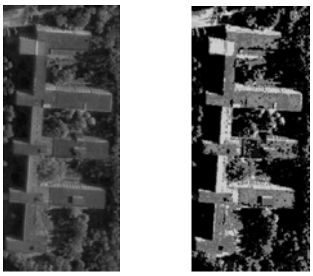

Fig. 3. Original image used for initial spiking neural network learning (left) and the results of its processing (right)

Having learned initially, spiking neural network was presented with larger portion of image where patterns of a new cluster were presented (figure 4). A new stage of learning was initiated, and it took two epochs.

Fig. 4. Image that contains patterns of a new cluster (left) and the results of its processing (right)

Finally, the learned spiking neural network was presented with the whole picture (figure 5). As it can be easily seen, the image was processed successfully.


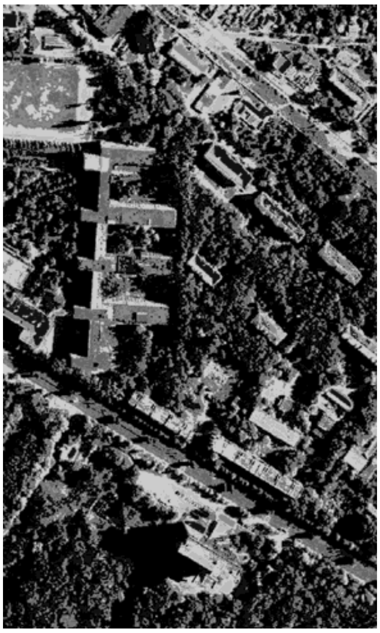

Fig. 5. Full picture used to test the learned spiking neural network (left) and the results of its processing (right)

It makes sense to note here that the proposed spiking neural network can successfully detect new clusters if patterns of only one cluster appear at a time. It also can't handle situation when number of clusters decreases.

5 Conclusion

The simulation experiment confirms that the proposed fuzzy spiking neural network can successfully detect new clusters during data processing. The further work on this direction is synthesis of spiking neural network architecture that is capable not only new clusters detection but also clusters merging (for example, using merging methods based on alphabetical optimality criteria [4]). Such spiking neural network will be able to adjust both its parameters and its structure in unsupervised mode that would be another step in further integration of inductive modelling and computational intelligence.

References

[1] Gerstner W., Kistler W.M.: *Spiking Neuron Models: Single Neurons, Populations, Plasticity.* Cambridge University Press, Cambridge: 400 p., 2002.

[2] Bodyanskiy Ye., Dolotov A.: Analog-Digital Self-Learning Fuzzy Spiking Neural Network in Image Processing Problems. Yung-Sheng Chen (ed.): *Image Processing*, INTECH: pp. 357-380, 2009.

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

[4] Bodyanskiy Ye., Dolotov A.: 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, Vol. 2: pp. 213-216, 2009.