

# Fast Neural Network Adaptation with Associative Pulsing Neurons

Adrian Horzyk

Dept. Automatics and Biomedical Engineering  
AGH University of Science and Technology  
Krakow, Poland  
horzyk@agh.edu.pl

Janusz A. Starzyk

University of Information Technology and Management in Rzeszow  
and School of EECS, Ohio University  
Rzeszow, Poland, and Athens, OH 45701.J., USA  
starzykj@ohio.edu

**Abstract** — This paper presents a fast self-organization of neural network structure using a new simplified pulsing model of neurons. These neurons incorporate the concept of time while simplifying many functional aspects of spiking models. This model is attractive because it is computationally very efficient. It allows for fast association of experimental data using conditional plasticity rules built-in neurons. It can be used for the representation of sequential and non-sequential data in neural network architectures. It also allows for the creation of synaptic connections that represent similarity, sequence, proximity, or defining dependencies between data and objects. Thus, this model can be used to develop complex neural graph structures for knowledge representation and retrieval. Such neural structures can be further used for fast search of related data or objects, clustering, classification, recognition, data mining, knowledge exploration, data retrieval, as well as for various cognitive tasks.

**Keywords** — *spiking neurons, associative pulsing neurons, associative models, computational complexity, self-organization.*

## I. INTRODUCTION

In computational intelligence, we usually use the second generation of neuron models based on the continuous nonlinear activation functions and artificial neural networks that are trained using various optimization criteria and gradient descent-based methods. The third generation of neuron models [6] [7] [16] incorporate the concept of time and try to model biological neurons more precisely in order to discover and understand internal information processes that are substantial to understand how the knowledge is formed in the brain and how the intelligence works [4] [6] [7] [22]. Today, we have many third generation spiking models, e.g. integrate-and-fire (IF), leaky integrate-and-fire (LIF), exponential integrate-and-fire (EIF) models as well as conductance-based spiking neuron models [6] [11] [16] [19]. These models are biologically more faithful, but also more complex and more difficult to use for various practical computational tasks than the second generation of neuron models. On the other hand, some experiments have shown that certain aspects of the biological platform are unnecessary to achieve desired functionality expected from artificial neural networks [6] [8] [9] [20].

This paper introduces a new model of neurons (named as associative pulsing neurons or APNs) which incorporates the concept of time-related associative abilities of real neurons. The presented model expands the model introduced by Horzyk in [8] and [9] and shows how it is used in fast self-organization of

associative neural network structures. APNs do not intend to precisely model membrane potentials, transduction of stimuli, and electrochemical nonlinear excitation changes of the neuronal states which are observed in real neurons. Instead, they use the main features of the biological neurons based on the accumulation of charges over time and changing synaptic connections between neurons to self-organize. This idea was demonstrated on a group of connected neurons extracting knowledge from data [14] [23].

Moreover, in neurobiology-based signal processing, there are two competing approaches to explain neuron communication: through a rate of pulses or through temporal differences between pulses [16]. In this paper, temporal differences between neuron activations are used to influence the strength of the synaptic connections to postsynaptic neurons. APNs simulation uses a simple event-driven approach which makes this neuron model very efficient developing and using large networks of neurons. Activations of various neurons occur at different times which is crucial for the interpretation of what is processed and what results.

The main objective of this paper is to describe fast self-organization and simulation of neural networks built of APNs. Such networks have both good associative and temporal storage properties making them good candidates for associative short-term memories. They can sort, cluster, recognize, and classify data [10], extract knowledge from input sequences [8], and make new associations [24] to store findings. In this paper, a novel neural network model based on temporal patterns of activations and associative pulsing neurons is presented. Dynamics of the model is described in detail so that it can be easily reproduced. Future research will compare storage properties of APN networks to recurrent LSTM networks [25] and to cortical structures of hierarchical temporal memories [26].

## II. RECEPTORS, NEURONS, AND SENSORY INPUT FIELDS

### A. Receptors and Sensory Input Fields

Receptors of various kinds play a very important role in real neuronal systems. They supply those systems with external data which frequent or unique combinations must be associated and remembered to satisfy needs of the system in future interactions with its environment. The receptors are placed in various sensory input fields (SIFs) which represent different features or parameters of modeled objects, their sequences, or routines in

the neural system. The observed objects can stimulate the receptors with different strength for various periods of time. The strength depends on the receptor placement in its SIF or the similarity of the input data to the value(s) represented by the receptor, i.e. the value(s) to which the receptor is most sensitive. The receptors stimulate special neurons (called sensory neurons) which are charged over time depending on how long the input data are presented on the SIFs and how much the input data are close to the data ranges represented by receptors. If the stimulation lasts long enough, then some sensory neurons achieve their firing thresholds at various time and start to stimulate other connected neurons. The sensory neurons are automatically created and connected to receptors, but the connections to other neurons are conditionally dependent on the plasticity rules built in these neurons as described in [10]. Real neurons, as well as APNs, have no external supervisor to train them correctly to process data, so the incoming data, their frequency of various combinations as well as their uniqueness have to adapt neurons to associate objects correctly and proceed to control various tasks.

Biological receptors [17] are not sensitive to a wide variety of input stimuli, but they are most sensitive to very limited ranges (rods and cones in retina) harmonics (hearing receptors) or subsets (olfactory or taste receptors) of input stimuli [12] [15]. Moreover, receptors are connected to dedicated neurons (called here sensory neurons) which are charged by stimuli coming from these receptors. If the receptors are stimulated long and strong enough, then the connected sensory neurons achieve their firing threshold levels, fire, and send appropriate signals to the higher parts of the nervous system [12] [15]. The fact, that various receptors charge various combinations of neurons, allows for plastic adaptation of neurons which fire at the same time or in close succession of each other. Such neurons reinforce existing connections or create new connections to neurons in their proximity [12] [17]. Neurons together with connections and their parameters reproduce important relations which allow for association of various features, objects, classes, and routines in nervous systems.

Generally, real neurons use their plastic connections both for data transfer and as modifiers of a data processing algorithm, not like contemporary microprocessors which use data buses only to transfer data. In the APN model, the results of neuronal computations can be extracted from the speed, frequency, and succession of spikes of activated neurons and from their connections which define what kind of information these neurons represent. The APNs focus on functional dependencies between neurons and ability to represent various concepts in the form of combinations of input stimuli over time [5] [13]. Each APN represents these spatiotemporal combinations of input stimuli that charge it to its firing threshold. Therefore, a firing APN communicates to other connected neurons information that it has just recognized an object, class, or activity represented these spatiotemporal combinations. The output result produced by an APN can be a part of other spatiotemporal combinations represented by other (higher order) neurons [28]. Each spatiotemporal combination can consist of stimuli coming from different APNs as well as from receptors representing input data. Input data sensed by receptors can be of various kinds. They can

represent different features describing some objects as well as represent already processed data, names of objects or classes.

### B. Receptors and Associative Pulsing Neurons

Each receptor  $R_{v_i}^{a_k}$  is defined as a graph element in the neural structure which is placed in a sensory input field (SIF), it is sensitive to the presented values on its SIF, and is charging a single connected sensory neuron  $S_{v_i}^{a_k}$  with a different strength (1) according to the presented input data (Fig. 1-2). If input data are symbolic or boolean then each receptor  $R_{v_i}^{a_k}$  is sensitive only to a single value (if  $v^{a_k} = v_s^{a_k}$  then  $x_{v_s}^{a_k} = 1$ ) which it represents and is insensitive to all other values (else  $x_{v_s}^{a_k} = 0$ ). When input data are numerical (integer or float) then receptors are mostly sensitive to a single represented value  $v_i^{a_k}$  and less sensitive to other values according to the following formula:

$$x_{v_i}^{a_k} = \begin{cases} 1 - \frac{|v_i^{a_k} - v^{a_k}|}{r^{a_k}} & \text{if } r^{a_k} > 0 \\ \frac{|v_i^{a_k}|}{|v_i^{a_k}| + |v_i^{a_k} - v^{a_k}|} & \text{if } r^{a_k} = 0 \end{cases} \quad (1)$$

where the  $x_{v_i}^{a_k}$  is receptor's  $v_i$  stimulation strength,  $v^{a_k}$  is an input value presented on the SIF,  $v_i^{a_k}$  is a value represented by the receptor  $R_{v_i}^{a_k}$ , and  $r^{a_k}$  is a current range of values of the attribute  $a_k$  defined as:

$$r^{a_k} = v_{max}^{a_k} - v_{min}^{a_k} \quad (2)$$

where  $v_{min}^{a_k} = \min\{v_i^{a_k}\}$  and  $v_{max}^{a_k} = \max\{v_i^{a_k}\}$  are the current minimum and maximum values of the attribute  $a_k$ .

A sensory neuron  $S_{v_i}^{a_k}$  that does not represent the stimulated input value  $v^{a_k}$  fires with some delay according to the relative difference between receptor value  $v_i^{a_k}$  and value  $v^{a_k}$ . The period of time which passes from the receptor stimulation to the sensory neuron firing determines the degree of similarity between these values. Moreover, the input value  $v^{a_k}$  will cause the receptor  $R_{v_i}^{a_k}$  to activate the connected sensory neuron  $S_{v_i}^{a_k}$  less frequently with larger difference between the values  $v^{a_k}$  and  $v_i^{a_k}$ . While the external stimulus is presented on the SIF (Fig. 1-2) the stimulated receptor  $R_{v_i}^{a_k}$  is continuously stimulating and charging the connected sensory neuron  $S_{v_i}^{a_k}$  with the strength (1). The sensory neuron  $S_{v_i}^{a_k}$  achieves its pulsing thresholds after time  $t_{v_i}^{a_k}$  specified in (3) when this neuron is solely stimulated by this receptor long enough:

$$t_{v_i}^{a_k} = \begin{cases} \frac{r^{a_k}}{(r^{a_k} - |v_i^{a_k} - v^{a_k}|)} & \text{if } r^{a_k} > |v_i^{a_k} - v^{a_k}| \\ \infty & \text{if } r^{a_k} = |v_i^{a_k} - v^{a_k}| \\ 1 + \frac{|v_i^{a_k} - v^{a_k}|}{v_i^{a_k}} & \text{if } r^{a_k} = 0 \end{cases} \quad (3)$$

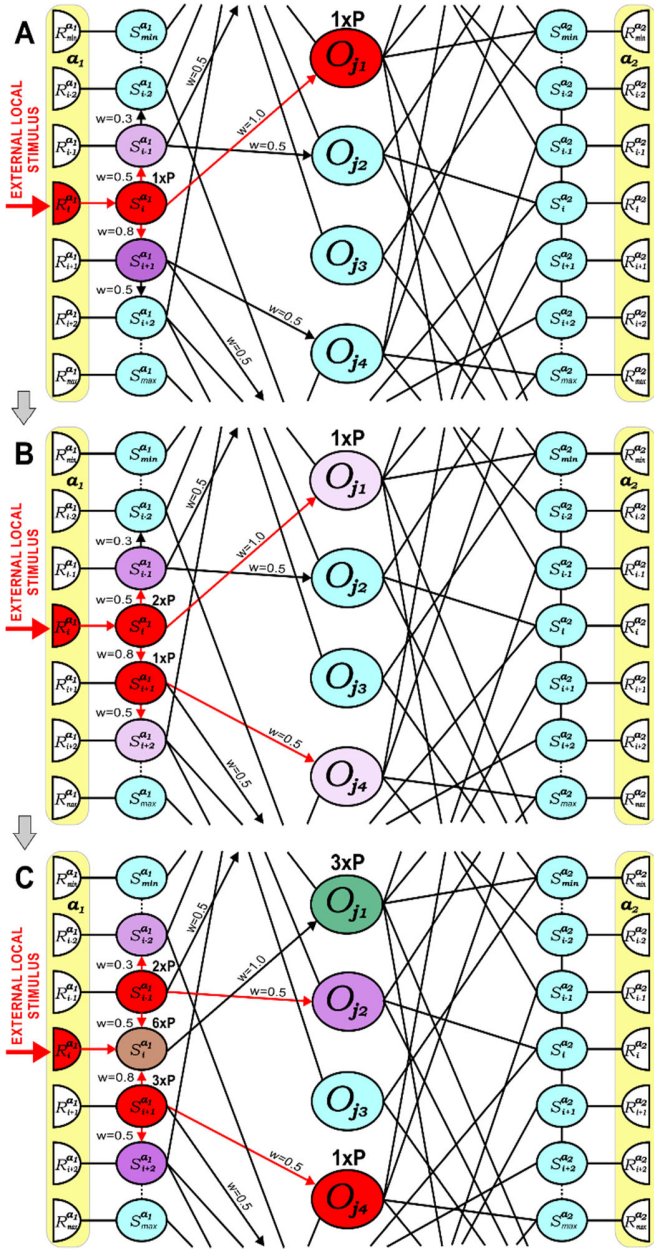


Fig. 1. Neural state changes according to the continuous input stimulus of the receptor  $R_i^{a_k}$  and the forwarded pulses after activation of neurons, where A, B, and C phases illustrate associated state changes shown in Fig. 3.

The time differences in achievement of pulsing thresholds by various sensory neurons are crucial for the next neuronal processes defined in the following section and for input data processing. The sensory neurons not only stimulate other connected neurons but they also automatically associate close values and similar objects by creating connections between neurons representing these objects as described in [10].

Sensory neurons are connected if their receptors represent orderable or neighbor data. The conditional plasticity rules are used for the creation of connections between such sensory neurons [10]. In real neural networks, connection weights and

the sensitivity of sensory neurons to input values are adapted over time, whereas the APN model allows to quickly calculate weights of connections between sensory neurons directly from similarity of values represented by the connected sensory neurons in view of its feature range  $r^{a_k}$ :

$$w_{S_{v_i}^{a_k}, S_{v_j}^{a_k}} = 1 - \frac{|v_i^{a_k} - v_j^{a_k}|}{r^{a_k}} \quad (4)$$

where  $v_i^{a_k}$  and  $v_j^{a_k}$  are mostly sensitive by the receptors  $R_{v_i}^{a_k}$  and  $R_{v_j}^{a_k}$  stimulating connected sensory neurons  $S_{v_i}^{a_k}$  and  $S_{v_j}^{a_k}$ .

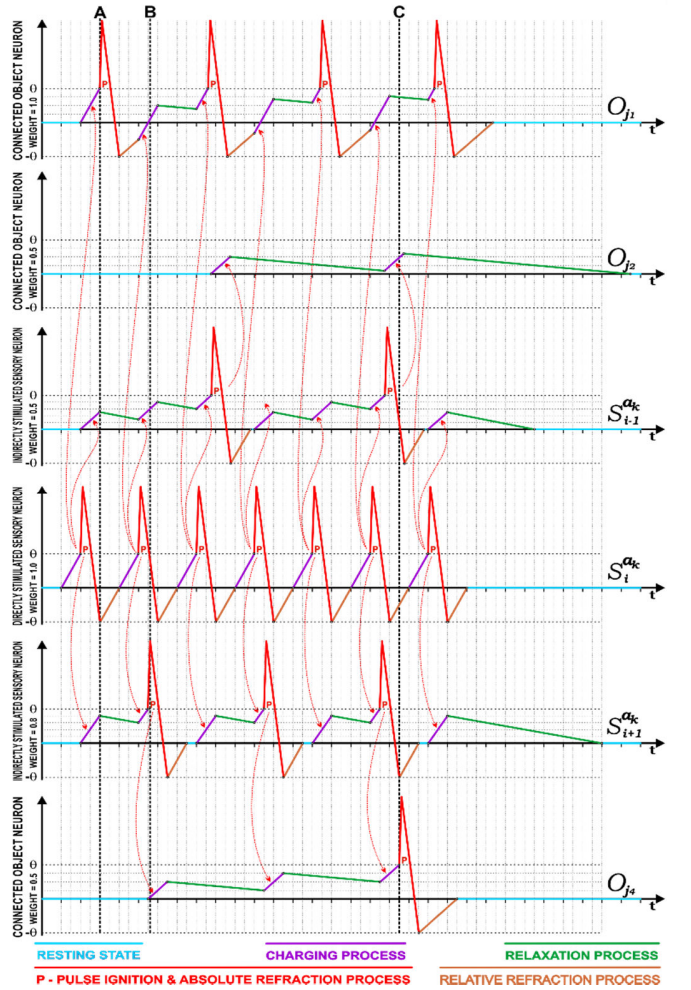


Fig. 2. Synaptic dependencies between receptors, sensory and object neurons. The lower part of the figure shows time domain state and activation levels for object neurons  $O_{j1}$ ,  $O_{j2}$ , and  $O_{j4}$  and sensory neurons  $S_{i-1}^{a_k}$ ,  $S_i^{a_k}$ , and  $S_{i+1}^{a_k}$ . Only the sensory neuron  $S_i^{a_k}$  is directly stimulated by the receptor  $R_i^{a_k}$  and fires most often. The other neurons responses are based on their stimulation strength through associative weighted connections. Object neuron  $O_{j1}$  fires more frequently than object neurons  $O_{j2}$  and  $O_{j4}$  because it is directly and more strongly connected to the most active neuron  $S_i^{a_k}$ , however the object neuron  $O_{j2}$  fires earlier than the object neuron  $O_{j4}$  since it has stronger synaptic connection from the sensory neuron  $S_i^{a_k}$  to  $S_{i-1}^{a_k}$  than to  $S_{i+1}^{a_k}$ .

The sensory neurons  $S_{v_i}^{a_k}$  can also be connected to object neurons  $O_j$  representing objects defined by combinations of input values or other objects. The APN model calculates weights of these connections directly after the  $N_{v_i}^{a_k}$  number of times the given value is used to define various objects:

$$w_{S_{v_i}^{a_k}, O_j}^{Tn} = \frac{1}{N_{v_i}^{a_k}} \quad (5)$$

The reciprocal connections from the object neurons  $O_j$  to the sensory neurons  $S_{v_i}^{a_k}$  representing the defining values are typically equal one:

$$w_{O_j, S_{v_i}^{a_k}}^{Tn} = 1 \quad (6)$$

The weights (4)-(6) do not need to be stored in the connections because they can always be calculated very fast for all outgoing connections before a given neuron fires. One can save a lot of memory and avoid updating their values when the ranges of values  $r^{a_k}$  or the number of connections  $N_{v_i}^{a_k}$  change in time.

The thresholds of all sensory neurons  $S_{v_i}^{a_k}$  are always equal one ( $\theta_{S_{v_i}^{a_k}} = 1$ ). The pulsing thresholds  $\theta_{O_j}$  (7) of object neurons  $O_j$  are defined as the sum  $W_{O_j}$  (8) of the weights of all connections coming from the sensory neurons  $S_{v_i}^{a_k}$  if this sum is less than one because the defining combination of the input stimuli representing an object represented by the object neuron should be able to activate it:

$$\theta_{O_j} = \begin{cases} 1 & \text{if } W_{O_j} \geq 1 \\ W_{O_j} & \text{if } W_{O_j} < 1 \end{cases} \quad (7)$$

$$W_{O_j} = \sum_{S_{v_i}^{a_k}} w_{S_{v_i}^{a_k}, O_j}^{a_k} \quad (8)$$

### III. EVENT DRIVEN SIMULATION WITH APNS

In reference to previous works of many researchers [1] [2] [3] [6] [8] [9] [11] [16] [18] [19] [20] [21], the associative pulsing neuron (APN) is a new model of neurons which incorporates the concept of time, as well as spiking models, and plasticity of real neurons, which expands the adaptivity of the neural network. This is similar to spiking models, but unlike them the APNs also enable automatic conditional association of objects represented by this kind of neurons as proposed in [8]. APNs have built-in plastic mechanisms which enable them to connect conditionally. These mechanisms connect neurons which are frequently active in the similar time. In this way, newly created object neurons connect to recently activated sensory neurons, representing combinations of training samples. The plasticity conditions always take into account the time elapsed between activations of APNs to connect neurons and determine connection weights as described in [10].

Spikes of neurons in the neural network result in activation of effectors that represent the output conclusions or actuators that act on the environment. The output defined by the series of spikes may be reinforced, represented by object neurons, remembered, and used in the future if the similar context will be observed again. The context is defined by the internal or external stimulation of the selected subset of neurons and/or receptors. The same frequently used contexts charge the same neurons and recall the same or similar sequences of spikes in all neurons. The neurons which most frequently spike/pulse and their most frequently stimulated effectors represent the strongest associations with the input context and can be treated as an answer of the neural network. Since APNs can quickly calculate connection weights, neuronal activities can be remembered in the contexts of the data combinations which initiated the stimulations and spikes. This lets us to associate data defining classes of objects, similarities between them, their proximity in space and time. The APN model is computationally very efficient because it updates neurons only in the moments of time when they switch their internal states, uses simple linear equations, and does not need to solve differential equations typically used in spiking neuron models. Moreover, all internal processes are efficiently managed by the discussed in this section internal process queue (IPQ), which is created for each neuron, and all IPQs are triggered by events managed by the global event queue (GEQ). In the GEQ, all events of all neurons are sorted during simulation of this model and executed in proper order in the simulation time. The discrete moments of updating neuronal states and their activation levels together with periods of time necessary to perform each process allow for successful simulation of parallelism of all processes in the neural network which further increases its computational efficiency.

This section describes how event-driven simulation of a network consisting of APNs and connected receptors is performed and how synaptic connections are modified during the neural network self-organization and learning process.

#### A. Simulation Algorithm: APN State Related Processes

The APNs can be in one of the seven states shown in Fig. 3. These states are charging, discharging, relaxing, pulsing, absolute refraction, relative refraction, and resting. Stimuli coming from other neurons charge or discharge a postsynaptic neuron for the period of time  $d^{CH}$ , here arbitrarily set to 1, but with a different strength. Receptors can also charge neurons with different strength for as long as the input data are presented on their SIFs. Input stimuli coming from various sources can naturally overlap in time (as shown in the upper part of Fig. 3), but APNs always add and combine overlapping stimuli transforming them into a sequence of successive internal processes (shown in the bottom part of Fig. 3).

Each new external excitation or inhibition stimulus is combined with the overlapped charging or discharging processes of the IPQ of the stimulated neuron. Each process in the IPQ is defined as  $P_k = (r_k, t_k, d_k, s_k, p_k)$ , where  $r_k$  defines a process type as charging (CH), discharging (DC), relaxation (RX), absolute refraction (AR), or relative refraction (RR),  $t_k$  indicates the time when the process starts,  $d_k$  is the duration of the process,  $s_k$  is the strength of the process defining the positive



or negative change of the neuronal activation level when this process is finished without interruption, and  $p_k$  is a pointer to the event  $E_n = (t_k + d_k, p_n^N)$  in the GEQ that watches for the moment of time  $(t_k + d_k)$  when this process will finish. The neuron pointed by  $p_n^N$  must be updated at the time  $(t_k + d_k)$ , and its process must be switched to the next one. During the update, the neuronal state and the activation level are usually changed. The pointer  $p_k$  allows to quickly remove the outdated event  $E_n$  if the process  $P_k$  is interrupted and must be changed. The strength  $s_k$  is positive for all charging and relative refraction processes and negative for all discharging and relaxation processes. Pulsing (P) of a neuron is not an internal time-span process added to an IPQ or watched by a GEQ. Pulsing is a moment in which the neuron fires and triggers stimulation of other connected neurons, adds an AR process to the IPQ, and can conditionally start plasticity processes which can change its threshold, connection weights, and conditionally connect or disconnect this neuron to other neurons. Plasticity processes which connect sensory neurons representing similar values were already described in [10]. In [8], [24], and [27] plasticity processes, which allow connecting neurons to represent sequential objects, are described. If a neuron is in the resting state it has no process in its IPQ.

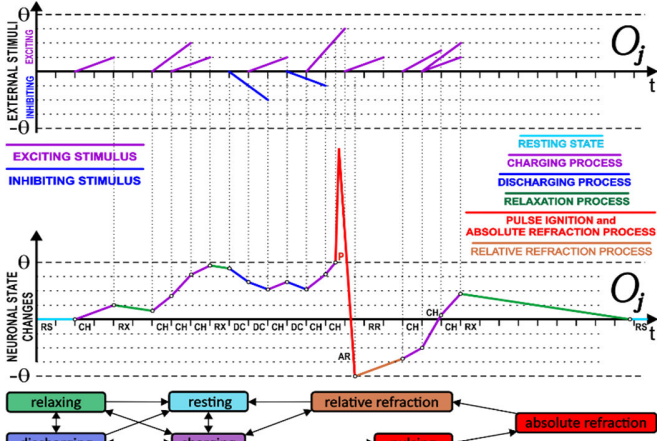


Fig. 3. External stimuli, activation level changes, and neuronal states over time. A state diagram on the bottom of the figure shows transitions between neuronal states that start various internal processes of an associative pulsing neuron  $O_j$ .

### B. Simulation Algorithm: Time Progression

All processes stored in the IPQ are sorted by their starting time and never overlap in time. When a new external stimulus comes, the neuron must combine it with processes in the IPQ if the stored processes overlap with this stimulus in time. The external stimulus is defined as  $S = (t_s, d_s, s_s)$ , where  $t_s$  indicates the time when the stimulus comes,  $d_s$  is the duration of this stimulus, and  $s_s$  is its strength that can be positive or negative depending on its excitatory or inhibitory character. The duration  $d_s$  for the receptor stimulus is set to infinity because we do not know how long the receptor will stimulate the connected sensory neuron. Such a receptor stimulation is

interrupted in the moment when the external stimulus is removed from the SIF. Therefore, the external receptor stimuli cannot be combined with the internal stimuli coming from neurons and are not represented in the IPQ. If the external stimulation overlap with processes stored in the IPQ, then they are temporarily combined to determine the expected pulsing threshold time that is watched out by an appropriate event in the GEQ. The temporal combinations assume the addition of the external receptor stimulus to all subsequent processes of the IPQ until this stimulus is stopped. If the receptor stops stimulating the sensory neuron or changes the stimulation strength, then the expected time of the event is recalculated and the associated event is updated. Hence, when the receptor stimulation  $S^{CH} = (t_s, d_s, s_s)$  is present, it is temporarily combined with the current internal process  $P_k = (r_k, t_k, d_k, s_k, p_k)$  creating a temporal process  $\bar{P}_k = (r_k, t_s, d_k, s_k + s_s, p_k)$ . Next, it is checked whether during the temporal process  $\bar{P}_k$  the pulsing threshold can be achieved in the same way as for the internal process  $P_k$ , and the event watching the moment of activation or the moment of the end of this process is determined according to what will be earlier.

The IPQ is empty when the neuron is in its resting state (RS). New discharging processes during a resting state or a relative refraction process have no influence on this state or this refraction process. However, a new charging process  $P_k = (CH, t_s, d_s, s_s, p_k)$  defined by the internal neuronal charging stimulus  $S^{CH} = (t_s, d_s, s_s)$  is simply added to empty IPQ. If the queue contains a relaxation process, then this process is interrupted, the neuron updates its actual activation level  $X_{t_s}$  in the interruption time  $t_s$  determined by the new stimulus, the IPQ is completely emptied, and a new process of charging or discharging is added. Moreover, we have to check whether the neuron can achieve its firing threshold during the calculation of its next internal process before it is added to the IPQ and watched by the GEQ. We check this by the evaluation of the inequality  $X_{t_s} + s_s > \theta$  for the currently added charging stimulus  $S^{CH} = (t_s, d_s, s_s)$ . If the inequality is true, then we have to shorten the time of the defined new process because the achievement of the firing threshold will come faster ( $t^{SP} = t_s + d_s \cdot \frac{\theta - X_{t_s}}{s_s}$ ) than the end ( $t_s + d_s$ ) of this stimulus. In the similar way, we have to check, whether the neuron can achieve its resting state before the end of the discharging process. In this case, we have to check the inequality  $X_{t_s} + s_s < 0$  for the currently discharging stimulus  $S^{DC} = (t_s, d_s, s_s)$ . If this inequality is true, then the time of the defined new process will come faster ( $t^{RS} = t_s + d_s \cdot \frac{-X_{t_s}}{s_s}$ ) than the end ( $t_s + d_s$ ) of this stimulus. In other cases, the new event contains the time  $(t_s + d_s)$  when the new charging or discharging process is expected to finish. The internal activation level of a neuron  $X_{t_s}$  in the moment  $t_s$  of the interruption caused by a new stimulus  $S = (t_s, d_s, s_s)$  is updated according to:

$$X_{t_s} = X_{t_0} + s_0 \cdot \frac{t_s - t_0}{d_0} \quad (9)$$

where  $t_0$  defines a starting time of the currently running process  $P_0$  in the considered neuron which is interrupted at the time  $t_s$ , and  $d_0$  is an expected duration of the interrupted process  $P_0$ . Fig. 4 presents how the current sequence of processes  $P_0$  and  $P_1$  are changed to a new sequence of processes  $\hat{P}_0$ ,  $\hat{P}_1$ , and  $\hat{P}_2$  at the interruption time  $t_s = t_3$  as a result of the new discharging external stimulus  $S_3 = (t_3, d_3, s_3)$ . The interruption stops the currently running process  $P_0$ , updates the neuronal internal activation level (9), and modifies the unexecuted part of the  $P_0$  process and all subsequent processes in the IPQ which overlap in time with the new stimulus  $S_3$ .

If the IPQ contains a relative refraction process (RR), then it can be interrupted only by a new excitation external stimulus. If so, the neuron updates its actual activation level (9) in the interruption time  $t_s$ , and combines the new charging process  $P_k = (CH, t_s, d_s, s_s, p_k)$  with the rest of the relative refraction process updating the IPQ. If the queue contains an absolute refraction process (AR), then all external stimuli are ignored, and no action is undertaken.

### C. Simulation Algorithm: Process Concatenation

If a new stimulus  $S = (t_s, d_s, s_s)$  arrives and the IPQ contains one or more charging and discharging processes  $P_0 = (r_0, t_0, d_0, s_0, p_0)$ ,  $P_1 = (r_1, t_1, d_1, s_1, p_1)$  etc., then the currently running process  $P_0$  is interrupted, the neuron updates its internal activation level (9), and the new stimulus  $S$  is combined with the remaining part of the interrupted process  $P_0$ , and a new process  $\hat{P}_0$  described by (10) is created.  $S_3$  is also combined with the subsequent process (11) or processes (12) and (13) in the IPQ that overlap with this new stimulus in time. As a result, the old processes  $P_0, P_1$  etc. overlapping with the new stimulus  $S$  are removed from the IPQ, and the events pointed by  $p_0$  and  $p_1$  are removed from the GEQ as well (Fig. 4). In Fig. 4, new processes  $\hat{P}_0, \hat{P}_1, \hat{P}_2$  etc. are calculated, and an updated events of the new processes are added to the GEQ. An example of these new processes  $\hat{P}_0, \hat{P}_1$ , and  $\hat{P}_2$  resulting from adding new stimulus  $S_3 = (t_{s3}, d_{s3}, s_{s3})$  are illustrated on the bottom right part of Fig. 4.

If the interrupted process ends earlier than the new stimulus or at the same time, i.e. when  $t_0 + d_0 \leq t_s + d_s$ , then we add the remaining part  $(s_0 \cdot \frac{d_0 - (t_s - t_0)}{d_0})$  of the interrupted process  $P_0$  to the overlapping part  $(s_s \cdot \frac{d_0 - (t_s - t_0)}{d_s})$  of the new stimulus  $S$  defining a new process  $\hat{P}_0$  in the following way (Fig. 4):

$$\hat{P}_0 = (\hat{r}_0, t_s, d_0 - (t_s - t_0), \hat{s}_0, \hat{p}_0) \quad (10)$$

where  $\hat{s}_0 = s_0 \cdot \frac{d_0 - (t_s - t_0)}{d_0} + s_s \cdot \frac{d_0 - (t_s - t_0)}{d_s}$  is the combined strength of the new process  $\hat{P}_0$ . The remaining part  $(s_s \cdot \frac{d_s - (d_0 - (t_s - t_0))}{d_s})$  of the new stimulus signal value is added to the next subsequent process  $P_1$  of the IPQ in a similar way.

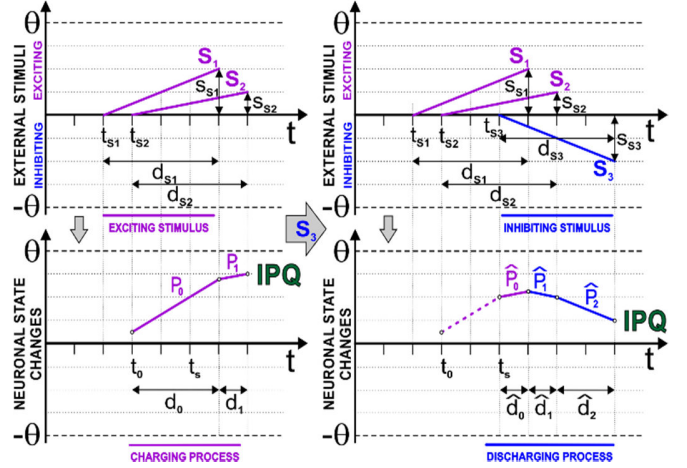


Fig. 4. The illustration of the operation that combines the new stimulus  $S_3$  with the processes  $P_1$  and  $P_2$  in the IPQ and the result of this operation.

If there is no subsequent process in the IPQ then we add this remaining part to the IPQ as a new subsequent process:

$$\hat{P}_1 = (\hat{r}_1, t_0 + d_0, d_s - (d_0 - (t_s - t_0)), \hat{s}_1, \hat{p}_1) \quad (11)$$

where  $\hat{s}_1 = s_s \cdot \frac{d_s - (d_0 - (t_s - t_0))}{d_s}$  is the combined strength of the new process  $\hat{P}_1$ . In the above defined processes (10) and (11),  $\hat{r}_0$  and  $\hat{r}_1$  determine the charging (CH) or discharging (DC) types of the new processes  $\hat{P}_0$  and  $\hat{P}_1$  fixed after the types and the resultant strength of the old processes  $P_0$  and  $P_1$  combined with the new stimulus  $S$ , i.e. if  $\hat{s}_n \geq 0$ , then  $\hat{r}_n = CH$  else  $\hat{r}_n = DC$ , where  $n \in \{0, 1\}$  (Fig. 4).

If there are subsequent processes  $P_1, P_2$  etc. in the IPQ then they must be combined with the remained part  $(s_s \cdot \frac{d_s - (d_0 - (t_s - t_0))}{d_s})$  of the new stimulus  $S$ . The new stimulus can end at the same time (9), after (9), or before (10) the end of the subsequent process  $P_1$ . Depending on the situation we use either equation (9) or (10):

$$\hat{P}_1 = (\hat{r}_1, t_1, d_1, s_1 + s_0 \cdot \frac{d_1}{d_s}, \hat{p}_1) \quad (12)$$

$$\hat{P}_1 = (\hat{r}_1, t_1, \hat{d}_1, \hat{s}_1, \hat{p}_1) \quad (13)$$

where  $\hat{d}_1 = d_s - (d_0 - (t_s - t_0))$  is a new duration of  $\hat{P}_1$  and  $\hat{s}_1 = s_1 \cdot \frac{d_s - (d_0 - (t_s - t_0))}{d_1} + s_s \cdot \frac{d_s - (d_0 - (t_s - t_0))}{d_s}$  is a new strength.

Moreover, if the new stimulus ends after the subsequent process  $P_1$  then it is necessary to recurrently combine the remaining part  $(s_s \cdot \frac{d_s - (d_0 - (t_s - t_0)) - d_1}{d_s})$  of it with the next subsequent processes  $P_2, P_3$  etc. if they exist or add this remaining part to the IPQ in a similar way as was described for the processes  $P_0$  and  $P_1$  (10)-(13).



#### D. Simulation Algorithm: Managing Simulation with the GEQ

Thanks to the linear time-domain approximation of the applied stimuli, all neuronal processes can be easily combined with new stimuli. As a result, new processes combining new stimuli with the overlapping processes from the IPQ replace the overlapping ones as shown in Fig. 4. Each charging process is checked for a possibly reaching the pulsing threshold  $\theta$  and each discharging process is checked for a possibly reaching the resting state in order to correctly add the new event to the GEQ. If one of these states can be achieved, then the expected pulsing or resting moment is computed, and an appropriate event is added to the GEQ. The GEQ will watch out for this moment in the event driven simulation in order to finish the currently running process  $P_0$ , update the neuronal activation level and trigger the next process or remain in the resting state. The anticipated time moment of such an event can change in future after another external stimulus that can come before the process is finished and this watching event is triggered. If it happens, the outdated event is removed from the GEQ, and a new one is added due to the new situation. If the pulsing moment comes, then all processes in the IPQ of the pulsing neuron are removed, and a new absolute refraction process  $P_{AR}$  is added to the IPQ. The AR process  $P_k = (AR, t^{AR}, d^{AR}, s^{AR}, p^{AR})$  changes the APN activation level to  $X_{t^{SP}+d^{AR}} = -\theta$  and its duration  $d^{AR}$  is preset to 1, where  $t^{AR}$  is the time when the neuron achieved its pulsing threshold,  $s^{AR} = -2 \cdot \theta$ , and  $p^{AR}$  is a pointer to the event in the GEQ where the absolute refraction process of this neuron ends. Each new external excitation stimulus always interrupts charging, discharging, relaxation, and relative refraction processes or a resting state of a neuron. Each new external inhibition stimulus can interrupt only charging, discharging, and relaxation processes. The absolute refraction

process cannot be interrupted by any stimulus. The relaxation and relative refraction processes are dependent on the actual internal state of the neuron when these processes start, on the pulsing threshold  $\theta$  of this neuron, and on the assumed maximum relaxation period, e.g.  $p^{RX} = 10$ , or the assumed maximum relative refraction period, e.g.  $p^{RR} = 5$  respectively.

It can be noticed that each IPQ typically consists only of a few processes, and each neuron has maximum one event in the GEQ which watches out its most recent predicted update time associated with its currently running processes. Thus, the GEQ consists of a number of events equal to the number of non-resting neurons in the whole network.

Each APN in the neural network is defined as a node in the associative neuronal graph structure that can have many input and output connections. Connections are always directed, but they can also connect neurons mutually. Each output stimulus of APNs is equal to one and is transferred to the postsynaptic neuron by the weight ranging from 0 to 1 or from 0 to -1 respectively for excitatory or inhibitory connection (Fig. 3). Most of the time, neurons can be charged with the exception of absolute refraction periods. They can be discharged only if their activation level are positive ( $X > 0$ ), so a neuron cannot be discharged during relative refraction processes when the activation level is negative ( $X < 0$ ).

The APN neurons have been implemented and used to represent various training data sets in such a way that receptors represented unique attribute values of all samples. Each receptor was connected with a sensory APN that was charged to the pulsing level (activation threshold) for a different period of time dependently on the strength with which the connected receptor

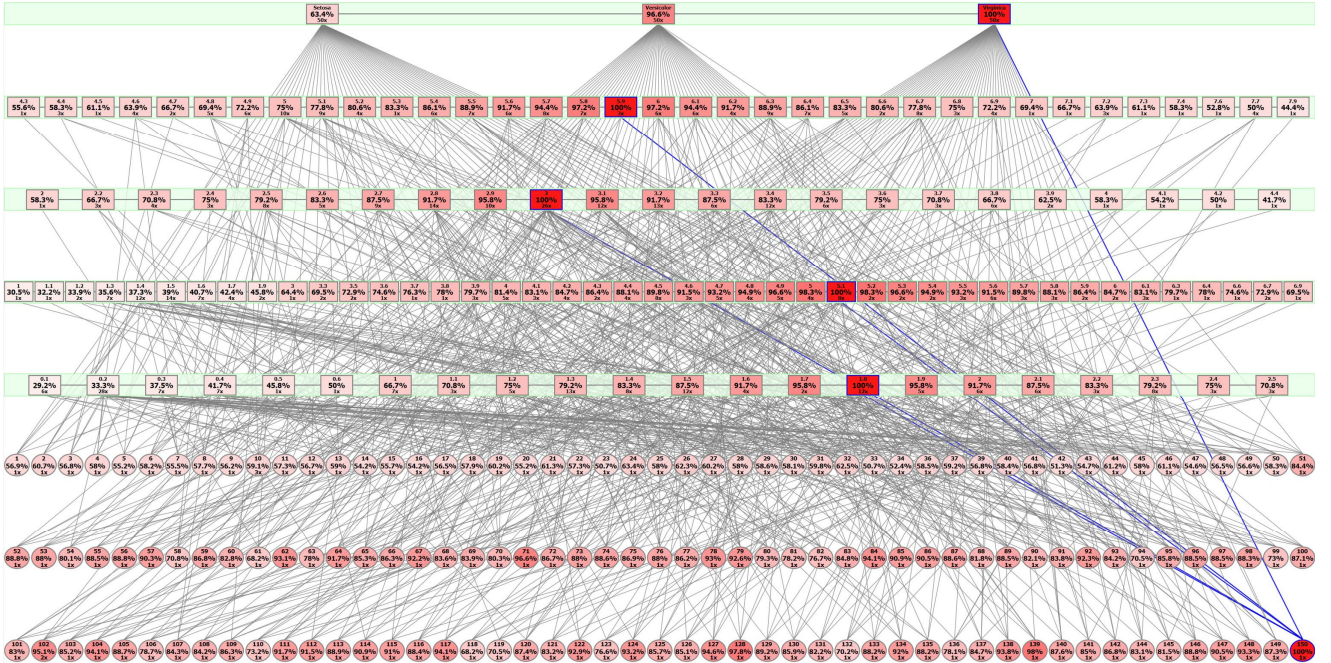


Fig. 5. Neural network structure automatically created for Iris data using plasticity rules for sensory associative pulsing neurons (squares) representing unique input values and for object associative pulsing neurons (circles) which can be used for recognition or classification as shown above.

was stimulating it. The activated neurons generated pulses which according to the connection weights were stimulating connected object APNs. Such kind of neuronal graph can be further used to classify training samples if not all input values are given. As a result, stimulated sensory APNs stimulate connected object APNs, and they stimulate back all sensory APNs which are not in the absolute refraction period. Thus, we finally achieve activation of the selected sensory neurons and the object neuron and classification as shown in Fig. 5.

#### IV. CONCLUSIONS

This paper introduced event-driven simulation with associative pulsing neurons, which allows for fast inference about similarities, clustering, recognition, and classification of objects on the basis of the speed and frequencies of pulses produced by the APNs. The presented approach uses linear approximation of internal neuronal processes, which enables efficient management of these processes using a queue, in which processes are sorted by the starting time. The simulated network may update neurons in discrete sparse moments when the processes must be switched or new external stimuli come to influence neuronal states. The presented APN model is very fast because all internal operations take constant time. Only addition of new events to the GEQ (implemented as a sorted list) takes logarithmic time of the number of neurons which are not in their resting states. This makes the formation of the associative neural network structure and its simulation very efficient.

The important finding is that the network with such neurons uses time and frequency of spikes to formulate its answer about found relations between objects and features which define them. Even more important finding is that spatial correlation can be treated exactly in the same manner as correlation over time, establishing a causal relationship in both time and space domains. Effectively, sequential succession over time and spatial associations use the same mechanism to trigger activations in the developed associative pulsing neural network.

Comparing to spiking neuron models, APN neurons simulation is much faster due to the linear approximations of all internal processes and constant processing time of all internal operations and many other when APNs interact. APN neurons show additional useful properties not observed in spiking neuron models. They can change their thresholds (i.e. sensitivity to input stimuli) if neurons are frequently activated, which results in better clustering properties, and they display fatigue, that make them resistant to repeated stimuli over a short period of time [24]. The APN can also automatically and conditionally create and reconfigure connections, so the associative pulsing neural networks are self-organizing. We are currently investigating other properties of APN in combination with the mini-column concept to increase the resolution of semantic memories using these models [26].

#### ACKNOWLEDGMENT

This work was supported by a grant from the National Science Centre of Poland DEC2016/21/B/ST7/02220 and by AGH 11.11.120.612.

#### REFERENCES

- [1] B.J. Baars. *A Cognitive Theory of Consciousness*. Cambridge, UK: Cambridge University Press, 1988.
- [2] G.A. Carpenter, S. Grossberg. Adaptive resonance theory. In *The Handbook of Brain Theory and Neural Networks*, M. Arbib (Ed.), MIT Press, Cambridge, MA, 2003, pp. 87–90.
- [3] L. Deuker et al. Memory Consolidation by Replay of Stimulus-Specific Neural Activity. *Jour. of Neuroscience*, vol. 33(49):19373–19383, 2013.
- [4] W. Duch. Brain-inspired conscious computing architecture. *Journal of Mind and Behaviour*, 26:1–22, 2005.
- [5] S. Franklin, T. Madl, S. D’Mello, and J. Snider, LIDA: A Systems-level Architecture for Cognition, Emotion, and Learning. *IEEE Trans. on Autonomous Mental Development*, 6(1):19–41, 2014.
- [6] W. Gerstner and W. Kistler. *Spiking Neuron Models* Cambridge University Press, 2002.
- [7] D. Graupe. *Deep Learning Neural Networks*. World Scientific, 2016.
- [8] A. Horzyk. How Does Generalization and Creativity Come into Being in Neural Associative Systems and How Does It Form Human-Like Knowledge? *Neurocomputing*, Elsevier, 2014, pp. 238–257.
- [9] A. Horzyk. Innovative types and abilities of neural networks based on associative mechanisms and a new associative model of neurons. *Proc. Of Int. Conf. ICAISC 2015*, Springer Verlag, LNAI 9119, 2015, pp. 26–38.
- [10] A. Horzyk, Neurons Can Sort Data Efficiently, *Proc. of ICAISC 2017*, Springer-Verlag, LNCS, Vol. 10245, 2017, pp. 64–74, DOI: 10.1007/978-3-319-59063-9\_6.
- [11] E. Izhikevich. Neural excitability, spiking, and bursting. *Int. J. Bifurcat. Chaos*, 10:1171–1266, 2000.
- [12] J.W. Kalat. *Biological grounds of psychology*. PWN, Warsaw, 2006.
- [13] J.E. Laird. Extending the Soar Cognitive Architecture. in *Proc. of the First Conference on AGI*, Memphis, Tenn, 2008, pp. 224–235.
- [14] D.T. Larose. *Discovering knowledge from data. Introduction to Data Mining*. PWN, Warsaw, 2006.
- [15] A. Longstaff. *Neurobiology*. PWN, Warsaw, 2006.
- [16] W. Maass. Networks of spiking neurons: The third generation of neural network models, *Neural Networks*, Vol. 10, Issue 9, Elsevier, 1997, pp. 1659–1671.
- [17] J.Z. Nowak and J.B. Zawilska. *Receptors and Mechanisms of Signal Transfer*. PWN, Warsaw, 2004.
- [18] V.A. Nguyen, J.A. Starzyk, W-B. Goh, D. Jachyra. Neural Network Structure for SpatioTemporal Long-Term Memory. *IEEE Trans. on Neur. Networks and Learning Systems*, vol. 23, no. 6, June, 2012, pp. 971–983.
- [19] S. Ostojic and N. Brunel. From Spiking Neuron Models to Linear-Nonlinear Models. *PLoS Comput Biol* 7(1): e1001056, 2011.
- [20] J.W. Pillow. Prediction and decoding of retinal ganglion cell responses with a probabilistic spiking model. *J Neurosci* 25: 11003–11013, 2005.
- [21] J.A. Starzyk, J. Graham. MLECOG - Motivated Learning Embodied Cognitive Architecture. *IEEE Systems Journal*, Vol. PP, Issue 99, pp. 1–12, 2015.
- [22] R. Tadeusiewicz. New Trends in Neurocybernetics. *Computer Methods in Materials Science*, 10 (1): 1–7, 2010.
- [23] Sha Zongyao, Li Xiaolei. Mining local association patterns from spatial dataset, 7th Int.Conf. on Fuzzy Systems and Knowledge Discovery, 2010.
- [24] A. Horzyk, J.A. Starzyk, Basawaraj. Emergent creativity in declarative memories, *IEEE Symposium Series on Computational Intelligence*, Athens, Greece, Dec. 6-9, 2016.
- [25] S. Hochreiter and J. Schmidhuber. Long short-term memory, *Neural Computation* 1997, vol. 9: issue 8, pp. 1735–1780.
- [26] Hawkins, J. et al. 2016. *Biological and Machine Intelligence*. Release 0.4. Accessed at <http://numenta.com/biological-and-machine-intelligence/>.
- [27] A. Horzyk, J.A. Starzyk, J. Graham. Integration of Semantic and Episodic Memories, *IEEE Transactions on Neural Networks and Learning Systems*, 2017, DOI: 10.1109/TNNLS.2017.2728203 (in press)
- [28] A. Horzyk. Deep Associative Semantic Neural Graphs for Knowledge Representation and Fast Data Exploration Proc. of KEOD 2017, SCITEPRESS Digital Library, 2017 (in press)