Multi-Class and Multi-Label Classification Using Associative Pulsing Neural Networks

Adrian Horzyk
AGH University of Science and Technology
Dept. Automatics and Biomedical Engineering
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 introduces the use of a new model of associative pulsing neurons (APN) for multi-class and multi-label classification tasks which are usually performed by separate artificial neural networks. The presented associative pulsing neurons have similar capabilities as various spiking models of neurons, but they additionally have built-in conditional plastic mechanisms which allow creating a neural structure with any given training dataset. Associative pulsing neurons can be connected and adapted very quickly. They have been implemented in the described research to define static patterns and their relations. They have been successfully used to automatically construct associative pulsing neural networks (APNN) to provide a classification of some well-known benchmark training data. These networks use special receptors which transform external stimuli into their internal representation of pulses. Receptors charge connected neurons in different periods of time according to the similarity of the presented input value to the values that they represent. This paper also presents the answer to one of the most challenging tasks in neuroscience, i.e. whether neurons communicate by a rate of pulses or temporal differences between pulses, and how the frequency of pulses influences the neural network activations.**

Keywords—classifiers, multi-class, multi-output, and multi-label classification, spiking neurons, neuron modeling, associative pulsing neurons, associative pulsing neural networks.

I. Introduction

Today, we have many computational intelligence methods that can be used for various classification tasks. This paper reveals how a biologically motivated model of neurons can be successfully used for the efficient construction of classic and multi-label classifiers. The classification and multi-classification are performed on the basis of the modeled associative properties of biological neurons. There is no question that biological neurons process input data in a quite different way than it was modeled over half a century ago when first computers were constructed. Contemporary computers and many other devices use the Turing machine as a computational model which operates on symbols [18]. Since then many various models of neurons were developed to imitate the way how real neurons work and process data. Today, many computational methods can be used for classification [7], [22], [26], [28], [29], [30], [31]. Because we are used to the Turing machine computational model implemented in our computers, it is not easy to create a brain-like associative model of data processing. In neuroscience, we still try to find out the answer to the fundamental questions how neurons work, communicate, and transfer information, treating neurons like connected processors. However, real neurons form a special and dynamically changing processing unit – a nervous system – that controls the behavior of biological bodies. Real neurons generate spikes when the accumulated stimulation achieves activation thresholds, but the same neurons can also ignore too weak stimulations and automatically, gradually return to their resting state preparing for new stimulations [16], [19]. They can also be temporarily unproductive from the computational point of view when real neurons are in the absolute refraction periods after spikes. This apparent ineffectiveness of real neurons hides tremendous computational potential because such states are useful for stimulation of other neurons. Each real neuron can inform other connected neurons about its activation using spikes which are produced when the neuron recognizes one of the input combinations of stimuli (patterns, objects) it represents. Moreover, neurons react to their mutual activations creating new or modifying a structure and parameters of existing connections. Real neurons can also grow, requiring stronger or longer stimulations, which allows them to specialize in the representation of subsets of input stimuli and work faster.

Spiking neurons and spiking neural networks (SNN), increase the level of realism modeling the concept of time and change neuronal states in time [6], [15], [20], [25], but they do not create or develop a neuronal structure. Instead, they use a given structure which is adapted during a training process. This creates difficulty of creating desired associations between objects represented by the neural network and interpreting the results represented by the spiking neurons [25].

This paper uses of a new model of associative pulsing neurons (APN) developed in the previous research [8], [9], [10], [11], and [12] for construction of associative pulsing neural networks (APNN) for successful construction of multi-class and multi-label classifiers as well as for clustering and determination of similarities between various training patterns (objects). Multi-class classification problems occur when there are multiple categories (classes), but each pattern is assigned only to one of them [26]. Multi-label classification problems occur when each pattern can be associated with multiple categories (classes), i.e. when we have a set of target labels [23] [24]. The main contribution of this paper is to show how the same single APNN network can perform various computational tasks, i.e. similarity computation, training objects recognition, as well as multi-class and multi-label classification, which are usually performed by several dedicated artificial neural networks. The novelty is also in fast adaptation mechanisms used for the spiking-like |APN neurons, demonstrating that the results of classification can be depicted in terms of pulsing frequencies. It differentiates them from the results achieved for the second generation of artificial neural networks typically presented in the form of real numbers.

The APNN are constructed in the same way regardless of which attribute or attributes will be chosen to play a role of the class label(s). In the APNNs, each attribute can be used as input or output. It means that APNNs are not specifically trained to classify only one initially specified attribute defining class labels but they use associations in a very similar way as people do. For instance, looking at the chair, we can classify it as a chair, furniture, but also as grandma's favorite place to sit or as a four-legged object. Moreover, we can do it without learning of a new special class of four-legged objects! In the same way APNNs work, giving us a versatile tool for organizing memories, representing various relations, and performing multi-class and multi-label classification.

APNNs can also recognize or point out the most similar training patterns to any combination of input stimuli (e.g. any training or testing pattern or its part), and fill up the missing values if any. This versatility is possible thanks to the plastic and associative properties of the APN neurons described in [10] and [12]. When two or more neurons are often active in close time-succession, then they are connected to remember their association which can be next recalled. The APNs like real neurons and spiking models of neurons can be charged, discharged, relax after stimulation, and refract after spikes that generate pulses which stimulate other neurons in APNN neural networks.

II. Associative Pulsing Neurons and Networks

Associative Pulsing Neurons (APN) described in [12] were developed to overcome some adaptive difficulties of spiking neurons and add some plastic capabilities known from real neurons and their networks. The fundamental assumption is that the neural network should automatically develop its structure and computing necessary parameters for given data to emphasize various relations between training patterns and the attribute data which define them. Thus, APNs are added and connected conditionally on demand of training data reproducing the similarities between values and training objects. The wealth of represented relations allow the APNN to find the most similar objects, missing values, clusters, or classifying input data. Developed for APN plastic mechanisms first mentioned and later described in [8], [9], [12], [13], and [14], were used to create a neural structure presented in Fig. 1. In this structure, APN neurons representing similar values of the same attribute

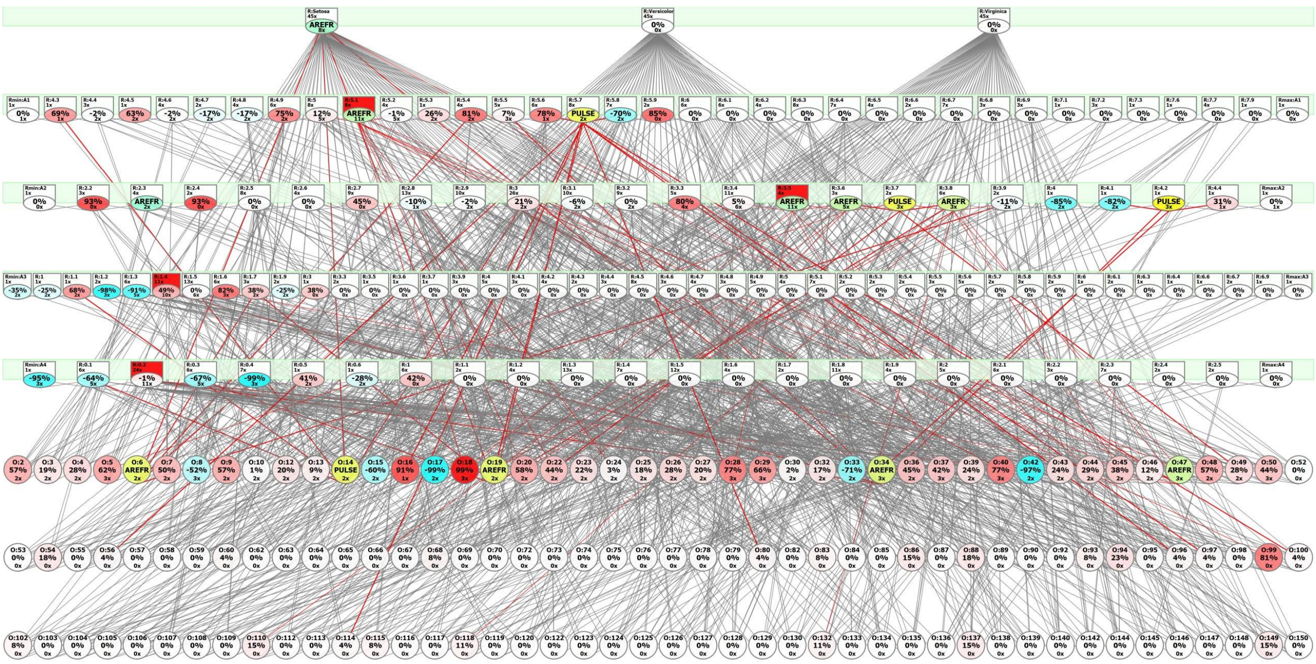

Fig. 1. The APNN structure developed for the Iris training data [32] stimulated by the validation pattern of the Setosa class. Red rectangles represent the externally stimulated receptors which constantly charge the connected value neurons painted below as ellipses. Receptors of each attribute are painted on a long lightgreen attribute rectangles which represent input sensory fields for values of each attribute separately. Each neuron (ellipses) presents its state in percentage computed as a quotient of its state to its activation threshold and the number of pulses which it has generated during this stimulation process. PULSE means the short state of the neurons indicating achievement of the activation thresholds; AREFR indicates neurons (colored in lightgreen) which are in the absolute refraction process; cyan neurons are during the relative refraction process and return to their resting states gradually, pink and red neurons are during the charging or relaxation, and white neurons are in their resting states. The red lines represent currently active connections which transfer stimulations from recently activated (pulsing/spiking) neurons (colored in yellow). All processes in such a network are dynamic and parallel. The upper left neuron in the AREFR state representes the mostly pulsing (8x) value neuron of a class attribute that represents the Setosa class due to the input context.

are connected to allow inference for objects defined by these values. In this case, the plastic mechanism of connecting such neurons is defined by ASSORT-2 algorithm [9]. This mechanism assumes that a neuron representing a sensor value will be connected by two neighbor values represented by two existing neurons in the network under a simple condition. This condition states that when the existing neuron is stimulated stronger by the receptor than by the neighbor neuron, then the connection between these neurons is replaced by connections to the newly created neuron that represents a new value. This simple mechanism, automatically allows connecting neurons to represent a linear order of all values of each input attribute.

Another plastic mechanism implemented in the APNN networks connects all activated neurons representing various attribute values to a new neuron defining a training object (training pattern). This mechanism also automatically aggregates neuronal representation of all duplicated training patterns. If no neuron of those representing objects is activated shortly after presenting the combination of input signals representing a training sample (object), then a new neuron (representing this new object) is created.

APNs can be stimulated many times by the same or different neurons or receptors and sum these stimulations taking into account relaxation in time. Each APN represents all these time-spread combinations of input stimuli which activate this neuron, i.e. charge it over its pulsing threshold. Hence, there is always a set of input stimuli combinations that is represented by such a neuron. In this way, we can treat every neuron as a simple classifier which recognizes all patterns of the represented class. Such neurons can be used to construct more sophisticated classifiers or multi-label classifiers as will be described in the subsequent sections of this paper.

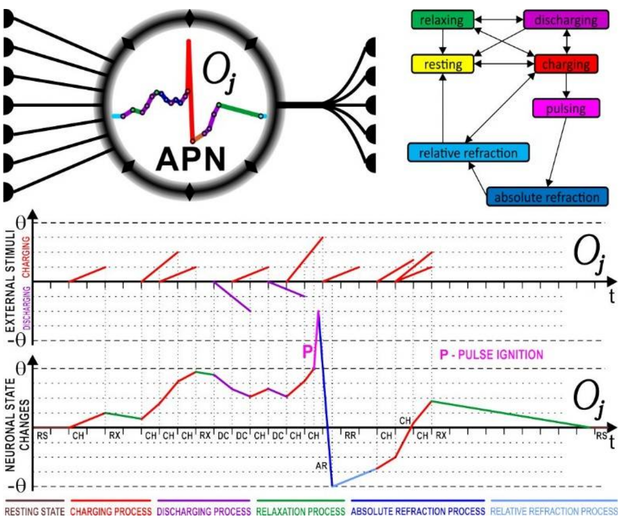

Fig. 2. The Associative Pulsing Neuron (APN) integration of parallel input stimuli and its influence on the neuronal state and internal processes.

Associative Pulsing Neural Networks (APNN) are dynamically developed neural networks consisting of APN neurons and other functionally important elements such as receptors and effectors. Receptors are responsible for the transformation of input data to the internal representation. Effectors work as an output interfaces and can influence or write on output devices or send signals to actuators. Receptors are connected to value neurons internally representing the sensed values by receptors in the APNN network. Moreover, these neurons can be connected to each other when representing similar or neighbor values, especially numerical ones. Each stimulated receptor $R_{v_i}^{a_k}$ by the value v^{a_k} and representing the value $v_i^{a_k}$ constantly charges its connected value neuron $S_{v_i}^{a_k}$ with the strength $x_{v_i}^{a_k}$ computed using (1):

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

where r^{a_k} is the current range of values of the attribute a_k, and q decreases the sensitivity of receptors to close values, i.e. the influence of similar values to those represented by receptors. Moreover, according to the various types of stimulations of receptors in the sensory input fields created for each attribute (Fig. 1) and introduced in [10], only one receptor representing the input value or two neighbor receptors (when the stimulation value is not exactly represented in the network) are stimulated as shown in Figs. 1 and 4. The value neurons representing numerical data are connected with weights calculated as follows:

$$w_{S_{v_i}^{a_k}, S_{v_j}^{a_k}} = \left(1 - \frac{\left|v_i^{a_k} - v_j^{a_k}\right|}{r^{a_k}}\right)^p \qquad (2)$$

where $v_i^{a_k}$ and $v_j^{a_k}$ are the values represented by the connected receptors $R_{v_i}^{a_k}$ and $R_{v_j}^{a_k}$ stimulating connected neurons $S_{v_i}^{a_k}$ and $S_{v_j}^{a_k}$, and p decreases the weight of neighbor connections, i.e. the strength of influence of the connected neuron representing the neighbor value of the same attribute.

Each APN neuron [12] changes its internal states in time. Depending on its internal state and external stimuli, it can be in in a resting state or during one of the following states: charging, discharging, relaxing, pulsing, absolute or relative refraction (Fig. 2). The most important aspect of its work is the way how the parallel and sequential input stimuli and the internal running process are combined and change the state of such a neuron. This model of neurons uses linear integration of the input stimuli and internal processes as shown for exemplary stimuli in Fig. 2.

In the APNN networks, we can use a special kind of APN neurons which internally represent the trained patterns. These neurons are called object neurons and represent training patterns. If training data contain duplicates of training patterns, they are aggregated and represented by the same object neurons. Each object neuron is connected to all value neurons representing the values defining the training pattern which this object neuron represents. The weights of the connections from value neurons $S_{v_i}^{a_k}$ to object neurons O_j are simply determined on the basis of the number of incoming connections to object neurons and computed using (3):

$$w_{S_{v_i}^{a_k},O_j} = \frac{1}{K} \quad (3)$$

The reciprocal connections between the object neurons O_j to the sensory neurons $S_{v_i}^{a_k}$ are equal their activation threshold value:

$$w_{O_j,S_{v_i}^{a_k}} = \theta = 1 \quad (4)$$

where activation thresholds θ of all neurons in the presented APNN networks were set to one (4).

This quite simple configuration of weights and thresholds is powerful thanks to the characteristics of APN neurons, the sparse graph structure which reproduces relations between attribute values defining patterns, and the implementation of the time approach which allows charging different neurons for different periods of time. Moreover, training data not only influence receptors which charge APN neurons but also automatically develop an APNN structure which can differ a lot for various training data.

APNs implement the approach of time which is very important during stimuli integration and modeling of subsequent processes. Time and various periods of the internal processes of the APNs enable to determine the strength of associations and finally differentiate between the answers of the network [12]. Most frequently pulsing APNs represent the most associated values and objects (Figs. 1, 4 and 5), so we can determine the APNN response on the basis of the numbers of pulses of the most pulsing neurons as shown in Figs. 1 and 4.

The APNs in the APNN multi-associate values and objects they represent directly and indirectly. Thanks to this feature, it is unnecessary to initially determine which attribute describes the intended class as in supervised learning because every attribute can be used as a class attribute. In most cases, class attribute represents values which are repeated in many training samples. The missing attribute values during external stimulation are automatically computed and pointed out by the APN with the most numerous pulses indicating the strongest associations to the given input values of the other attributes. It also means that APNNs must be developed and trained in a different way than classical artificial neural networks (ANN). For multi-class and multi-label classification, we use training data that contain information about the desired classes, but we treat the class labels the same way as all the other input parameters. Thus, this learning process is not unsupervised or supervised, so we call it an associative adaptation. Next, the APNN develops a neural structure and calculates all weights to reproduce associations between all values of input parameters.

APNs connect when they are frequently activated in close temporal succession. The synaptic strength depends on the frequency of subsequent activations of connected neurons. For simplicity and higher speed of the organization of the network structure, we can also use the ASSORT-2 algorithm [9] for numerical data which calculates connection weights according to the similarity of close attribute values represented by the connected neurons. In using APNNs for classification, there will also be used the AVB-trees for optimization of receptors organization and access to given input data as described in [10]. In this solution, training patterns are represented by separate APNs called object neurons which are connected to the APNs called value neurons defining these patterns (Fig. 1). The same object neurons represent duplicated training patterns. All duplicated values of each attribute are also aggregated and represented by the same value neurons. Such aggregations are crucial from the associative point of view as described in [8] where elements in sequences are aggregated, in [10] where database entities are aggregated, and in [12] where various values of static patterns are aggregated.

On the other hand, there is a big difference in representation of computational results collected in APNNs in comparison to the second generation of neural networks, e.g. MLP, RBF or SVM, based on non-linear continuous neurons [26]. APNs are usually connected in a complex dynamic sparse graph structure, so there are many loops and neighbor connections in the APNN networks (Fig. 1). APNs connect to other APNs which often start pulsing in close periods of time and the connection weights are established based on the difference in time between their activations and on the number and strength of input stimuli which took part in charging the neurons to their pulsing thresholds. In simple cases, the weights can also be determined on the basis of the similarity or frequency of activations of the connected neurons as a result of activations of the presynaptic neurons similarly to the Hebb's rule [16], [26]. It is sufficient for the multi-class or multi-label classification tasks.

APNNs can have different structures because they are developed on demand and according to the training data. We assume that a single neuron can represent any combination of input stimuli that occurs and we use this feature to represent each unique training pattern by a single object neuron. We also suppose that such neurons can connect between themselves when they often pulse in short intervals. In such cases, APNs are connected, or their existing synaptic weights are reinforced.

The following section discusses how to use APNN networks for classification tasks.

III. Construction of APNN Multi-Class Classifiers

Associative Pulsing Neural Networks for classification tasks are constructed in the following way (Fig. 3):

1. Create an empty APNN network.
2. Add Sensory Input Fields (SIF) for all data attributes, treat the desired class attribute in the same way as all other attributes. Each SIF is the collection of receptors representing aggregated values of a single data attribute. In order to efficiently add, remove, or find represented values in each SIF use AVB-trees described in [10].
3. For all objects (training patterns) in the dataset use ASSORT-2 algorithm [9] to stimulate the existing receptors and find those representing the attribute values of a new object or create new receptors representing new values for each receptor separately. If a new receptor is added, create a new value neuron for it and connect the receptor to it. In such a way, receptors of each SIF aggregate duplicates of all trained objects.

4. Next, simultaneously stimulate all receptors representing attribute values of the new trained object and let them stimulate connected value neurons. When value neurons are charged to their pulsing threshold, let them stimulate connected neighbor value neurons and object neurons. Allow to mutually connect new value neurons to the other value neurons representing the neighbor values according to the ASSORT-2 algorithm [9]. Calculate the neighbor weights using (2).
5. Check whether any object neuron will pulse during the two periods of time necessary for charging a single neuron. If it happens, it means that this object is already represented in the network and will not be duplicated. If no object neuron pulses in this period, add a new object neuron to the network and connect it mutually with all recently pulsing neurons, i.e. the value neurons representing the values defining this object. Set the connection weights between value and object neurons according to formula (3) and (4).

To develop such an associative neuronal structure, only one browse through the training data is necessary. When for a given training data, the construction process of the APNN is finished, the network is ready for input stimulations via receptors to classify, multi-label classify input data, recognize training samples, or to determine the most similar objects (here training patterns) to the given inputs.

Several construction steps of development of the APNN for Iris data are illustrated in Fig. 3. The upper A-view of the network structure of the APNN represents the first training sample [Setosa, 5.1, 3.5, 1.4, 0.2]. When the receptors and neurons of this network represent the first training sample, the receptors representing minima and maxima are automatically added for each attribute. The adequate receptors of minima and maxima quickly activate connected neurons when a new or existing minimum or maximum for any attribute is presented. The way how these receptors work were described in [12].

The B-view in Fig. 3 presents the network structure after addition of the second training sample [Setosa, 4.9, 3.0, 1.4, 0.2] where we can observe aggregations of values 1.4 and 0.2 represented by the same receptors and value neurons as for the first training sample. Moreover, we can see neighbor connections (2) between value neurons 4.9 and 5.1 as well as between value neurons 3.0 and 3.5. The C-view in Fig. 3 presents the structure created for the first three training samples, and the D-view is the structure developed for the first six training samples. We can observe still more aggregations of values which define various training patterns, e.g. the value 0.2 is connected to five object neurons, and the value 1.4 has a connection with three object neurons.

The final structure developed for all Iris training samples is presented in Figs. 1 and 4. As we can notice, 150 values of all training samples for each attribute are represented by a much smaller number of receptors and value neurons in sensory fields representing these values. Namely, we got 35 value neurons for the representation of leaf-length, 23 value neurons for the representation of leaf-width, 43 value neurons for the representation of petal-length, 22 value neurons for the representation of petal-width plus two extra neurons representing minimum and maximum for each attribute.

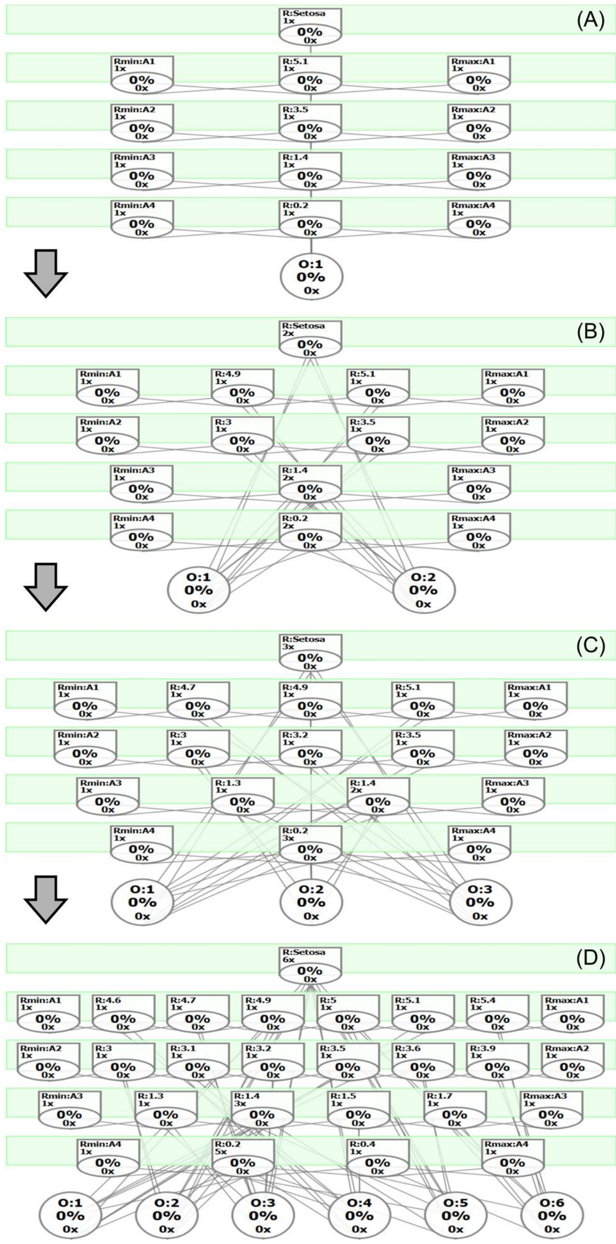

Fig. 3. The first steps of the development process of an Associative Pulsing Neural Network for the training samples from the Iris dataset [32].

IV. Classification Results for APNNs

The second generation of non-linear neurons usually returns the answer of the neural network in the form of the numbers in the range of [0,1] or [-1,1]. Pulsing neurons, as well as real neurons, work differently, and the results can be read from the different numbers of pulses counted for individual neurons during some period of time and from the speed of their first activation after the input stimuli [8], [10], [16], [19]. In biology, the number of pulses sent to the muscles decides about the contraction force or speed of movement [16], [19]. Generally, the APN which spikes the first and the most frequently in comparison to other APNs is the most associated with the given input stimulation called an associative context. On this basis, we can define results of classification as well. Some examples of classification are shown in Fig. 1 for the first validation patterns of the Setosa class, and in Fig. 4 where we can see the intermediate result of classification of the validation sample of the Virginica class. In both cases, the neurons representing the winning classes are more frequently pulsing than the neurons representing the other classes. In Fig. 1, the neuron Setosa pulses 3 times, while the neurons Versicolor and Virginica do not pulse at all. In the Fig. 4, the neuron Virginica pulses 7 times, while the neuron Versicolor only 3 times, and the neuron Setosa does not pulse at all. When classifying we stimulate the APNN with some input data, we wait for the most frequently pulsing APN representing one of the defined class. To get a reliable answer, we have to wait until neurons representing expected classes will not differentiate in the number of pulses for the given associative input context. Sometimes, such a differentiation is not possible even after a long period of stimulation as described in Example 2. It means that the associative input context is equally similar to more than a single class and we get an ambiguous or multi-label classification result. The stimulation process which leads to differentiation of the numbers of pulses for class label neurons can take different time due to the correlation of training data and the similarity with the classified input pattern. Thanks to the fast algorithms controlling the behavior and internal parallel processes in the APNNs described in [9], [10] and [12] we can get results quite quickly as shown for the exemplary training data in Table I. Each of the exemplary training datasets of Table I was used for developing ten APNNs using 10-folds cross-validation with a proportional selection of training and validation data from all classes. This adaptation strategy allowed to validate all training data and compute average validation errors.

Example 1

In this paper, APNNs was used to develop classifiers for classical benchmark training datasets from UCI ML Repository [32] to show how this kind of neural networks can adapt to classification tasks and what construction, adaptation and validation time and what quality of generalization we can achieve. Thus, we used 10-folds cross-validation to enable easy comparisons of the achieved classification results (Tab. I) to the results of other kinds of neural networks and learning algorithms obtained by others [28], [29], [30], [31].

Fig. 4. presents the intermediate results of stimulation of the tested APNN networks by one of the validating pattern [?, 7.4, 2.8, 6.1, 1.9] without a fixed class (marked as "?"). In the second row of receptors, two receptors representing the neighbor values 7.3 and 7.6 to the not represented value 7.4 are stimulated with the strength 0.97 and 0.94 appropriately (1) to approximate the value 7.4. In all other cases, i.e. for values 2.8, 6.1, 1.9, the appropriate receptors already exist in the network, so these receptors are stimulated with the full strength equal to 1.0 as follows from (1). All these receptors are continuously stimulated in time until the APNN will be able to distinguish between the numbers of pulses of the defined class neurons, here neurons

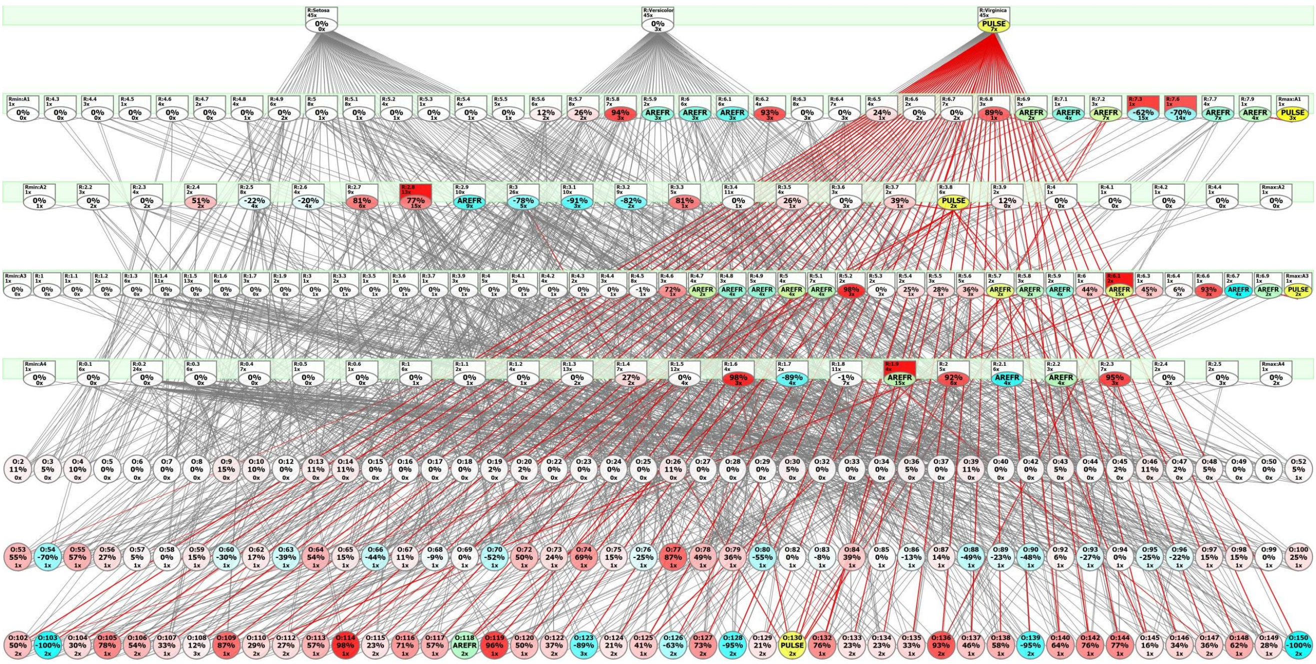

Fig. 4. The APNN structure developed for Iris training data [32] during the cross-validation process for the validation input pattern [7.4, 2.8, 6.1, 1.9] of the Virginica class where the class label was not presented but the number of pulses (7 times) of the value neuron representing this class is the bigger than the number of pulses achieved for the Versicolor (3 pulses) or Setosa (0 pulses) classes, so the predicted class by the APNN is validated correctly in the presented case. The above-given input values (7.4, 2.8, 6.1, 1.9) affect receptors (red rectangles), which continuously stimulate connected value neurons (ellipses below them). The stimulated value neurons start pulsing after some time. Their pulses stimulate connected object neurons and connected neighbor value neurons representing the closest smaller and bigger values of the same attributes. The stimulated value neurons and object neurons also start pulsing after some time, and in turn, start stimulation of other connected neurons. The most frequently pulsing neurons represent the most associated values, the most similar objects, and the most probable class of the input values.

representing Setosa, Versicolor and Virginica classes. If the difference is big enough, the APNN stops stimulating receptors, waits until all neurons return to their resting states, and the next validation sample is taken to stimulate the network again. In the end, the most stimulated class neuron is compared to the class of the validation sample class to determine whether the classification is correct or not. If two or more class neurons have the same numbers of pulses all the time, the stimulation is stopped, and the ambiguous result of classification is treated as incorrect.

TABLE I. COMPARISON OF CLASSIFICATION RESULTS

Datasets	APNN	The Best Classifiers	Time *
Iris (p=1, q=1)	99.47%	98,45% = 99,8% (MLP; RBFN; PNN; NaiveBayes; ROC Area) [29]	6.02 s
Wine (p=1, q=1)	98.48%	97,8% - 98,9% (kNN, Manhattan, auto k=1-10; IncNet, Gauss; SSV opt prune) [28]	6.33 s
Cars Evaluation (p=1, q=1)	97.24%	93,22% - 93,51% (Decision Trees; NaiveBayesian; MLP [30]	41:12.76 s
Breast Cancer Wisconsin (p=1, q=1)	95.58%	97,0% - 97,5% (Naïve MFT; SVM Gauss, C=1, s=0.1; NB + kernel est) [27][28]	10:21.01 s

* Total time of the creation, adaptation, and 10-folds CV of APNNs

Table I presents the comparison of classification results obtained for the APNNs and other the best classifiers used for the training data mentioned above [32] and 10-folds cross-validation.

V. MULTI-LABEL CLASSIFICATION RESULTS FOR APNNS

The majority of computational intelligence methods [26], [26] are constructed only for classification tasks or multi-class classification, but they are unable to perform multi-label classification. The presented Associative Pulsing Neural Networks make no difference between various classification and multi-label classification tasks. They do not need to use any special adaptation routines to switch between various class labels because the APNNs do not use supervised learning which requires specifying class labels before starting a training process. The presented solution develops the APNN and adapts its weights always in the same way. When finished, the network can be stimulated by any combination of input data, and the network finds the most associated neurons which in this case represent the most similar training patterns and their defining values of all attributes. Some of these attributes can define desired classes and class labels.

Example 2

Sometimes, some results of stimulation of the APNN for Iris data are indecisive, and we cannot get clear answers about classification or recognition. For instance, in the input vector [?, 6.0, ?, 4.9, 1.4], the desired class attribute and the leaf-width attribute were not given (marked as “?”). In this case, the APNN does not clearly determine which object or class neuron is the most associated with the given inputs, because we get a very similar number of pulses of several object neurons and two value neurons representing classes Versicolor and Virginica. The most associated neurons are those which pulse the most frequently, so we can easily find out which training patterns represented by the object neurons are the most similar and which values represented by the value neurons are the most associated with this input vector. In this experiment, the APNN was constructed for all Iris patterns and used for multi-label classification. Only three input values [?, 6.0, ?, 4.9, 1.4] were used to stimulate tree receptors: R6.0 for the attribute “leaf length”, R4.9 for the attribute “petal length”, and R1.4 for the attribute “petal width”. The APNN classified these inputs as Versicolor and Virginica with the same strength (the numbers of pulses were 29x for both these classes). It also determined the most probable values for the missing attribute “leaf width” as 3.0 (24 pulses), 2.8 (21 pulses), 2.9 (20 pulses), 2.7 (18 pulses), 3.1 (17 pulses), 3.2 (17 pulses), 2.5 (15 pulses), 2.6 (15 pulses), 3.3 (14 pulses), 3.4 (13 pulses), or 2.2 (12 pulses), which is correct when comparing with the training patterns. The analysis of the number of pulses of object neurons points to the following training samples as the most similar to the presented inputs: [Versicolor, 6.1, 2.9, 4.7, 1.4] (10 pulses), [Versicolor, 6.3, 2.5, 4.9, 1.5] (10 pulses), [Versicolor, 6.0, 2.9, 4.5, 1.5] (10 pulses), [Versicolor, 6.1, 3.0, 4.6, 1.4] (10 pulses), [Virginica, 6.0, 2.2, 5.0, 1.5] (10 pulses), [Virginica, 6.1, 3.0, 4.9, 1.8] (10 pulses), [Virginica, 6.0, 3.0, 4.8, 1.8] (10 pulses). This variety of the similar training samples of two classes also explains why the network could not decide between the classification of these inputs as the Versicolor or Virginica class.

It is important that the result of multi-label and multi-class classification is determined not by a defined number of the closest training patterns, but it depends on a variable number of the most similar training patterns with different strength dependent on their similarity. As can be verified, the presented results are correct and consistent with intuition.

VI. CONCLUSIONS AND FINAL REMARKS

This paper presented the use of Associative Pulsing Neurons and Associative Pulsing Neural Networks to the classical and multi-label classification tasks. Besides this, APNNs were also used for multi-class and multi-label classification and determination of similar objects and illustrated with the training samples as explained in Example 2. The versatility of these networks together with quick construction and adaptation time with automatic associative self-organization, and good generalization properties (Table I) make them usable to various computational intelligence tasks. Moreover, the presented neuron model and neural networks have many similar properties as biological neuronal structures. People learn various patterns without establishing the use of them or defining their classes or names at the beginning. When necessary, we are able to cluster, classify, multi-label classify things all around on the previously learned patterns as well as quickly point out which are the most similar. The APNNs have similar abilities to those people have, which distinguishes them from other neural network techniques.

This paper was also focused on explaining how biological neurons can communicate and represent input patterns as well as output responses. The presented Associative Pulsing Neurons application to various classification tasks have shown that each neuron can represent various time-spread combinations of input stimuli. The neurons charged to the activation thresholds can influence connected neurons and also charge them to their activation thresholds. It means that these neurons communicate by the number and rate of pulses, and only if this rate, number, and strength are big enough in comparison to the rate of the relaxation process, they can successfully recall represented associations between neurons. The comparison of the number of pulses gives us information about the strength of associations between various neurons for the given input stimuli representing an associative input context. It is also important that such neurons can represent any set of time-spread combinations of input stimuli which make them pulsing. It means that they can represent static as well as sequential input patterns. In addition, they can also be used to represent secondary and more abstract patterns in various deep architectures as was already presented in [10] where relational databases were transformed into the special kind of APNN networks.

In the future research, APNs and APNNs will also be adapted used to sequential data and various knowledge exploration [18], [24], artificial intelligence [22], [23] and cognition tasks [2], [17] and for construction of deep hierarchical graph structures [7]. Another future research will address the transformation of relational databases into deep associative graphs [1], [10] to draw conclusions in another way than it is contemporary performed using data mining and knowledge exploration algorithms [18]. We would like to combine the presented associative systems with external and internal stimuli which will motivate [21] the system to learn some specific actions more preferable than others. The presented classification systems will also expand the semantic and episodic memories [8] constructed on the basis of associations between training data.

ACKNOWLEDGMENT

This work was supported by AGH 11.11.120.612 and the grant from the National Science Centre, Poland DEC-2016/21/B/ST7/02220.

REFERENCES

[1] Basawaraj, Janusz A. Starzyk, and A. Horzyk, "Lumped Mini-Column Associative Knowledge Graphs," IEEE Xplore, In: 2017 IEEE Symposium Series on Computational Intelligence, pp. 347-354, 2017.

[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, pp. 87–90, 2003.

[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. Snaider, "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, J.A. Starzyk, J. Graham, "Integration of Semantic and Episodic Memories," IEEE Transactions on Neural Networks and Learning Systems, Vol. 28, Issue 12, pp. 3084 - 3095, Dec. 2017, DOI: 10.1109/TNNLS.2017.2728203.

[9] A. Horzyk, "Neurons Can Sort Data Efficiently," In: Rutkowski L., Korytkowski M., Scherer R., Tadeusiewicz R., Zadeh L., Zurada J. (eds), Artificial Intelligence and Soft Computing, Best Paper Award at the ICAISC 2017, Springer-Verlag, LNCS, Vol. 10245, pp. 64-74, 2017, DOI: 10.1007/978-3-319-59063-9_6.

[10] A. Horzyk, "Deep Associative Semantic Neural Graphs for Knowledge Representation and Fast Data Exploration," Proc. of KEOD 2017, SCITEPRESS Digital Library, pp. 67-79, 2017.

[11] A. Horzyk, J.A. Starzyk and Basawaraj, "Emergent creativity in declarative memories," IEEE Xplore, In: 2016 IEEE Symposium Series on Computational Intelligence, Athens, Greece.ISBN 978-1-5090-4239-5, DOI: 10.1109/SSCI.2016.7850029.

[12] A. Horzyk and J.A. Starzyk, "Fast Neural Network Adaptation with Associative Pulsing Neurons," IEEE Xplore, In: 2017 IEEE Symposium Series on Computational Intelligence, pp. 339-346, 2017.

[13] A. Horzyk, "Innovative types and abilities of neural networks based on associative mechanisms and a new associative model of neurons," Proc. of ICAISC 2015, Springer-Verlag, LNAI 9119, pp. 26-38, 2015, DOI 10.1007/978-3-319-19324-3_3.

[14] A. Horzyk, Artificial Associative Systems and Associative Artificial Intelligence, Academic Publishing House EXIT, Warsaw (2013).

[15] E. Izhikevich, "Neural excitability, spiking, and bursting," Int. J. Bifurcat. Chaos, 10:1171–1266, 2000.

[16] J.W. Kalat, Biological grounds of psychology. 10th ed., Wadsworth Publishing, 2008.

[17] J.E. Laird, "Extending the Soar Cognitive Architecture," in Proc. of the First Conference on AGI, Memphis, Tenn pp. 224–235, 2008,.

[18] D.T. Larose, Discovering knowledge from data. Introduction to Data Mining. PWN, Warsaw, 2006.

[19] A. Longstaff, Neurobiology. PWN, Warsaw, 2006.

[20] W. Maass, "Networks of spiking neurons: The third generation of neural network models," Neural Networks, Vol. 10, Issue 9, Elsevier, pp. 1659–1671, 1997.

[21] J. A. Starzyk, J. Graham, A. Horzyk, "Trust in Motivated Learning Agents," IEEE Xplore, In: 2016 IEEE Symposium Series on Computational Intelligence, Athens, Greece ISBN 978-1-5090-4239-5, DOI: 10.1109/SSCI.2016.7850027, 2016.

[22] R. Tadeusiewicz, "New Trends in Neurocybernetics," Computer Methods in Materials Science, 10 (1): 1–7, 2010.

[23] J. Read, B. Pfahringer, G. Holmes, E. Frank, Classifier Chains for Multi-label Classification. Machine Learning Journal. Springer. Vol. 85(3), 2011.

[24] M.L. Zhang, Z.H. Zhou, "Multi-label neural networks with applications to functional genomics and text categorization (PDF)," IEEE Transactions on Knowledge and Data Engineering, Vol. 18, pp. 1338–1351 2006.

[25] E.M. Izhikevich, "Simple Model of Spiking Neurons," IEEE Transactions on Neural Networks, Vol. 14, No. 6, pp. 1569–1572, 2003.

[26] Rutkowski, L. Techniques and Methods of Artificial Intelligence. PWN, Warsaw, 2012.

[27] R. Tadeusiewicz, Introduction to intelligent systems, Fault Diagnosis. Models, Artificial Intelligence, Applications, CRC Press, Boca Raton, FL, 2011.

[28] K.P. Bennett, J. Blue, A Support Vector Machine Approach to Decision Trees, R.P.I Math Report No. 97-100, Rensselaer Polytechnic Institute, Troy, NY, 1997.

[29] https://www.fizyka.umk.pl/~duch/projects/projects/datasets.html

[30] K. Patel, J. Vala, J. Pandya, Comparison of various classification algorithms on iris datasets using WEKA, Int. Journal of Advanced Engineering and Research Development (IJAERD), Vol. 1 Issue 1, 2014.

[31] J. Awwalu, A. Ghazvini, and A. Abu Bakar, Performance Comparison of Data Mining Algorithms: A Case Study on Car Evaluation Dataset, Int. Jour. of Computer Trends and Technology (IJCTT) – Vol. 13, No 2, 2014.

[32] UCI ML Repository, http://archive.ics.uci.edu/ml/datasets/Iris, last accessed 2018/04/14.