Dynamic Cluster Formation using Populations of Spiking Neurons

Ammar Belatreche, Rakesh Paul
School of Computing and Intelligent Systems, Intelligent Systems Research Centre
University of Ulster
Londonderry, United Kingdom
a.belatreche@ulster.ac.uk

Abstract— **This paper introduces a novel neuro-dynamic system for adaptive online clustering using populations of spiking neurons and spike-timing dependent plasticity (STDP). Real-valued data samples are temporally encoded into spike events, used by biological neurons to encode information and communicate with one another, and clusters are represented by spiking neuron populations of varying size. The number of clusters is unknown a priori and clusters are learned in an online fashion where each data sample is provided only once. The coincidence detection capability of spiking neurons is utilized for data clustering and clusters are dynamically formed. The structure of the spiking neural network is constantly adjusted through adding and pruning of neuron populations. Besides, the number of neurons within each population constantly adapts as new data arrives. STDP is employed to adjust the strength of synaptic connections and enhance the selectivity of each population to its corresponding group of data. Preliminary experiments were carried out on synthetic and selected benchmark datasets to evaluate the performance of the proposed system. Promising results were obtained, which indicate the viability of spike-based population coding for online data clustering.**

Keywords- Spiking Neurons, Unsupervised Learning, Online Clustering, Population Coding, STDP, Spike Response Model

I. INTRODUCTION

Spiking neural networks continue to gain increased popularity due to their biological plausibility and their suitability in dealing with spatio-temporal data. Unlike traditional neuron models based on discrete or continuous activation functions, spiking neurons use spike events (action potentials) to encode information and communicate with each other [1]. They can operate as a coincidence detector a neuron fires only when it receives coinciding input spikes [2]. Spiking neurons are classified as the third generation of neural networks owing to their ability to effectively capture the rich dynamics of real biological neurons, hence increasing the level of realism in neural simulations. Recent developments in computational neuroscience have resulted in a basic understanding of information representation and processing in biological systems and the nervous system in particular. One of the remarkable characteristics of these systems is the ability to process large amounts of information efficiently and with apparent ease. Studies of the computational principles at the neuron level have resulted in a variety of neural models which have been used as a building block for different neural architectures, implemented both in software and hardware, with the aim of understanding the underlying mechanisms of computing and learning in neural assemblies. Also, a number of biologically plausible and non-plausible learning rules were put forward [4]. To date, however, with a few exceptions these efforts have found limited success in applying spiking neural networks to solving real-world problems due to the lack of efficient and scalable learning algorithms for these biologically plausible neuron models. Also, most of the existing supervised and unsupervised learning algorithms for spiking neural networks use fixed network structures often determined by trial and error (with the exception of [9]). In addition, the free parameters of these fixed structures are learned in an offline fashion where the learning procedure repeatedly sweeps through a set of training samples until an optimal (or near optimal) configuration is found. However, offline learning cannot cope well with changing environments and large scale problems as new classes of data cannot be incorporated without retraining on previously learned samples. Many applications require online or incremental learning where data samples are provided one- at-a-time and no a priori information about the data distribution is available.

This paper presents an adaptive spiking neural network (SNN) architecture for online clustering using populations of spiking neurons and spike-timing dependent plasticity (STDP). While several studies emphasize the importance of individual spike times in information coding and processing, compelling evidence suggests that more information about how neurons encode input stimulus features can be gained from analysing the activity of neuronal populations [3][5- 8]. Real-valued data samples are temporally encoded into spike events and dynamically formed clusters are represented by spiking neuron populations of varying size. The proposed approach exploits the coincidence detection property of spiking neurons for data clustering. The number of clusters is unknown a priori and clusters are learned in an online fashion where each data sample is provided only once. The structure of the spiking neural network is constantly adjusted through adding and pruning of neuron populations. In addition, the number of neurons within each population is constantly adapted as new data arrives. STDP is employed to adjust the strength of synaptic connections and enhance the selectivity of each population to its corresponding group of data.

The remainder of this paper is organised as follows: Section II presents background information about spiking neurons and a brief review of related work. Section II describes the proposed neuronal population-based online clustering system in detail. Performance evaluation of the proposed system through a set of preliminary experiments carried out on synthetic and selected benchmark datasets and a discussion of the obtained results are provided in Section IV. Finally, Section V concludes the paper and discusses future potential extensions for future work.

II. Background and Related Work

Spiking neurons communicate through short electrical pulses called action potentials or spikes. A spiking neuron model describes the dynamics of the membrane potential of a biological neuron. In response to an input spike pattern, a spiking neuron may produce an output spike train depending on the neuron current state and the strength of its synaptic connections to its predecessor neurons. The internal state of a neuron j is described by its membrane potential u_j. The neuron is at its resting potential in the absence of any input spikes. Each incoming spike time will alter the neuron membrane potential u_j resulting in post-synaptic potential (PSP). An output spike is triggered or fired when the neuron membrane potential u_j reaches a certain threshold θ as illustrated in Fig 1. This study is based on the Spike Response Model (SRM) [10-11], a mathematically tractable model of the integrate and fire class of neurons. Mathematically, the time course of an SRM neuron response to an incoming spike is described by the PSP function ε [11] (see Fig 1.(b) and (d)). The membrane potential of SRM-based neuron is given as the weighted sum of the PSP functions $\varepsilon(t, t_i)$ as described in equation (1) where w_{ij} is the connection weight between neurons i and j, Γ_j is the set of presynaptic neurons to neuron j which receives a set of spikes fired at times t_i, $i \epsilon \Gamma_j$. A typical PSP function is described in equation (2) where τ is the time constant

$$u_j\,(t) = \sum_{i \in \Gamma_j} w_{ij}\,\varepsilon(t-t_i) \qquad (1)$$

$$\varepsilon(t) = (t/\tau\,)\,e^{\,1-(t/\tau)} \qquad (2)$$

A neuronal circuit can produce different behaviors by modifying the number of synaptic connections, by modulating the excitability properties of individual neurons or simply by modifying the strength or efficacy of synaptic transmission at pre-existing synapses. This ability of the synaptic connections to undergo efficacy alteration is termed as synaptic plasticity. Learning rules account for experience-dependent development of synaptic plasticity during various combinations of presynaptic and postsynaptic activity. Spike-timing dependent plasticity (STDP) is a temporal learning rule based on the time difference between pre- and post-synaptic spikes. Applying STDP to spiking neurons will enhance their coincidence detection capabilities, as the firing time will be adjusted according to the input spikes. STDP is a synaptic specific Hebbian form of plasticity. However, STDP can regulate firing-rate instabilities that plague purely Hebbian models by keeping the synaptic strengths bounded within a certain range [12]. The learning model for this study is based on the STDP rule proposed in [13] with bounded synaptic strengths. The amount of synaptic modification arising from a single pair of pre and post synaptic spikes separated by time Δt is governed by a function F(Δt) as described in equation (3).

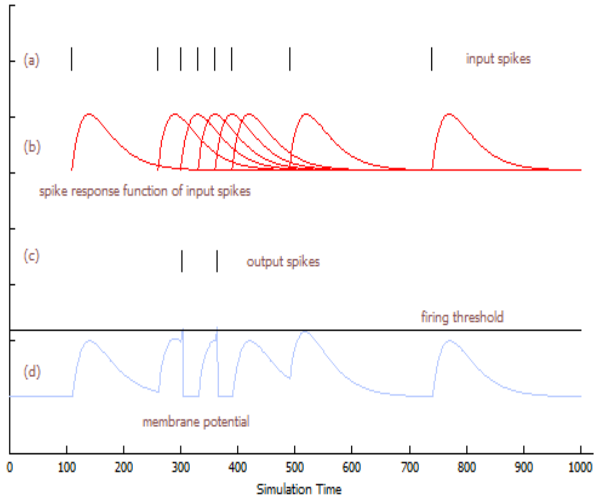

Figure 1: Dynamics of spiking neuron a) Input spike times to the spiking neuron. The weights are set to unity. b) Spike response functions of the input spikes. τ = 3. c) Output spike time for the given inputs. d) Spiking neuron internal state represented by the membrane potential u. An output spike is generated whenever the membrane potential exceeds the threshold θ =1.2.

$$F(\Delta t) = \begin{cases} A_{+}\exp(\Delta t/\tau_{+}) & \text{if } \Delta t < 0 \\ -A\text{-}\exp(-\Delta t/\tau_{-}) & \text{if } \Delta t \geq 0 \end{cases} \qquad (3)$$

The parameters τ_+ and τ_- determine the ranges of pre-to-postsynaptic inter spike intervals over which synaptic modification occur. A_+ and A_- are positive terms that determine the maximum amounts of synaptic modification when Δt is close to zero.

A number of results in experimental neurophysiology suggest existence of correlations between firing times of neurons. Neurons may partially synchronize their firing times in response to a particular visual or auditory stimulus. Without perfect correlation in firing times of different neurons, relevant information can be conveyed using statistically significant correlation making this coding mechanism robust to noise [14]. Population coding is a key mechanism for neural variability where different neurons in a population may produce different postsynaptic spike trains. The collective activity of neuronal populations can provide more information about how neurons encode input stimulus features than by a single neuron. For instance, the response of a single neuron may not be sufficiently reliable to distinguish between different objects in the visual cortex. However, this uncertainty can be resolved by considering the responses of other neurons and coordinating their firing patterns to capture salient features of a particular stimulus [14]. Quiroga and Panzeri present a detailed discussion of the importance of population analysis in further understanding how neurons encode and process information [14].

Very few attempts were made on unsupervised clustering using spiking neurons. In [15], Natschläger and Ruf showed that spiking neurons receiving temporally encoded inputs can compute radial basis functions (RBFs) by storing the relevant information in their delays. This idea was expanded in [11] where Bohte et al. demonstrated how spiking neural networks based on spike-time coding and Hebbian learning can successfully perform unsupervised clustering on real-world data. They also developed a temporal encoding of continuously valued data to obtain adjustable clustering capacity and precision by encoding input variables in a population code by neurons with graded and overlapping sensitivity profiles. In [2], Pham and Charles discuss how a self-organizing delay adaptation spiking neural network model can be used for clustering control chart patterns. However, none of these unsupervised clustering algorithms are online. They also have a fixed architecture with predefined number of neurons in the hidden and output layers. To find the best architecture for a particular clustering problem, these algorithms require offline evaluation of diverse configurations of hidden and output layers. The proposed population based system addresses these limitations and introduce a novel adaptive SNN architecture for dynamic online clustering.

III. POPULATION BASED UNSUPERVISED CLUSTERING

This section describes the design and implementation of spiking neuron population-based online clustering. The basic architecture is shown in Fig 2. A population of spiking neurons represents each cluster. Neurons within a population receive input as spike times. To compute with a spiking neuron, the inputs are encoded as spike times within a given input time window. Output spikes are generated depending on the input spike times and the parameters of a neuron. The neuron is simulated over a time window and the time of first spike is taken as the output. The continuous time window of simulation is divided into discrete intervals dt. Spike response functions for the input spikes are evaluated at each of these discrete time intervals. The weighted sum of spike response functions of each input spike defines the membrane potential of a neuron at a given time t. The real-valued input x_i is scaled down to a value between [0, 1]. An input value x_i Є [0, 1] is represented by a spike at step t_i and is computed by equation (4) where t_{input_window} is the activity window when inputs are presented to a spiking neuron. dt is the time interval between two consecutive steps as introduced for clustering in [2].

$$t_i = \left\lfloor \frac{(1-x_i)\ t_{input\_window}}{dt} \right\rfloor \qquad (4)$$

Let $n_1, n_2 \ldots n_m$ be a set of spiking neurons in a population P_n. The centre of the population $[c_1, c_2 \ldots c_n]$ is given by $[t_1, t_2 \ldots t_n]$, the time to first spike of the neurons $n_1, n_2 \ldots n_m$ for inputs from a given cluster. The size of a population depends on the variations in the inputs from a cluster centre. Some clusters can be encoded by a population as small as one neuron, while other populations may require more neurons to capture all salient features of a cluster.

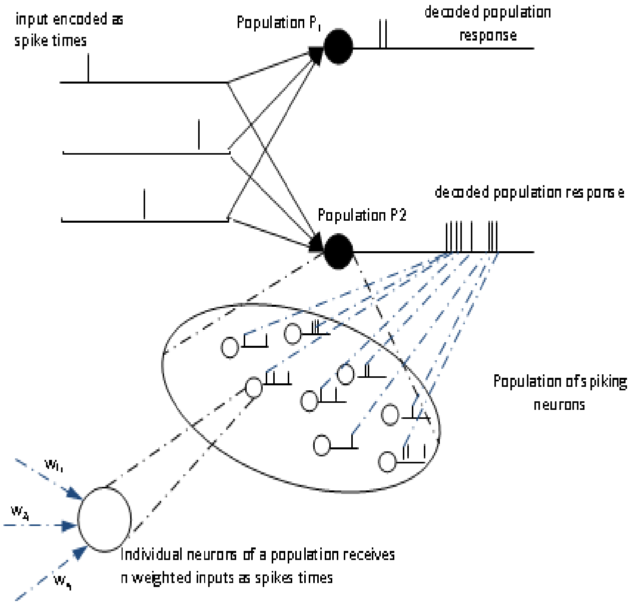
input encoded as spike times
Population P1
decoded population response
Population P2
decoded population response
Population of spiking neurons
w1
w2
wn
Individual neurons of a population receives n weighted inputs as spikes times

Figure 2: Basic architecture of population based unsupervised clustering using spiking neurons.

Clustering begins with an empty population. The population response coefficient $0 < \gamma_{min} \leq 1$ defines the minimum level of population response required to identify an input, e.g. $\gamma_{min} = 0.7$ means at least 70% of neurons within a population should respond to a given input to indicate that the input belongs to the cluster represented by that population. γ_{min} controls when to create a new population. For a given input X, let $t_1, t_2 \ldots t_n$ be the firing times of neurons $n_1, n_2 \ldots n_m$ in a population P_n. Let σ_{max} be the maximum distance allowed from the center of a population. The distance σ between firing times t_n and centre c_n is calculated as the standard deviation between $[t_n, c_n]$. The population response γ is calculated by identifying the number of neurons that fire within the maximum distance σ_{max} divided by the total number of neurons in that population. For instance, if a population has 4 neurons and 3 of them fire within the allowed distance, then γ is 0.75. The basic steps involved in population based clustering are given below:

- *Encode real-valued inputs as spike times.*
- *Define the initial population size and the minimum population response coefficient γ_{min}.*

- *On receiving an input, the responsiveness of all existing populations is evaluated. Identify populations that show a response greater than or equal to the γ_{min}. If there are no such populations create a new population assuming the current input is from a new cluster.*
- *Apply STDP based learning to populations that show responsiveness equal or above γ_{min}.*
- *Perform pruning of neurons within a population when they no longer contribute to clustering.*
- *Merge populations when they have similar centres.*

If the population response is below γ_{min}, a new population is created with specified population size. Initial population centre is given by the time to first spike of the neurons within a population for the current input based on which a new population is created. However, before creating a new population, an attempt is made to increase the population response of the existing population above γ_{min}. Adding new neurons to an existing population that shows maximum response to a given input does this. The maximum number of neurons added to an existing population is limited to the default size of a new population. The newly added neurons should fire within the maximum distance from the existing center. The default population size is one.

A. Learning in a Population

Input patterns are encoded in the synaptic weights by local STDP based learning. After learning, the firing time of a neuron reflects the distance of the evaluated pattern to its learned input pattern. The learning is carried online when the population response exceeds γ_{min}. The population centre is updated after each learning cycle to reflect the variations in firing. The updated population centre after learning is the mean of centre $[c_1, c_2 \ldots c_n]$ and firing time $[t_1, t_2 \ldots t_n]$ for the current input. STDP based synaptic weight update is given by equation (5) where $F(\Delta t)$ is given by equation (3), η is the learning rate and $[W_{min}, W_{max}]$ defines lower and upper bounds for the synaptic weights.

$$w_{ij}' = \begin{cases} \min [W_{max}, (w_{ij} + \eta . F(\Delta t))] & \text{if } \Delta t < 0 \\ \max [W_{min}, (w_{ij} + \eta . F(\Delta t))] & \text{if } \Delta t \geq 0 \end{cases} \quad (5)$$

B. Merging Populations

Another important aspect of population based clustering is the merging of populations whose centers are near each other. In the current architecture, populations are merged when at least one neuron from each population fire within the minimum distance i.e. populations with $\gamma > 0$. Populations are merged to the population with maximum response to a given input. The major steps involved in population merging are described below:

- *Let $P = P_1, P_2 \ldots P_n$ be the populations to be merged.*
- *Identify the Population P_{max} showing maximum response to a given input.*
- *Let $n = n_1, n_2 \ldots n_m$ be the neurons from populations P_i Є $[P, P_i \neq P_{max}]$. Let $t_1, t_2 \ldots t_m$ be the firing of neurons $n_1, n_2 \ldots n_m$ for a given in put X. A neuron n_i Є n is added to population P_{max} if t_i firing time is within the maximum deviation from the centre of P_{max}*

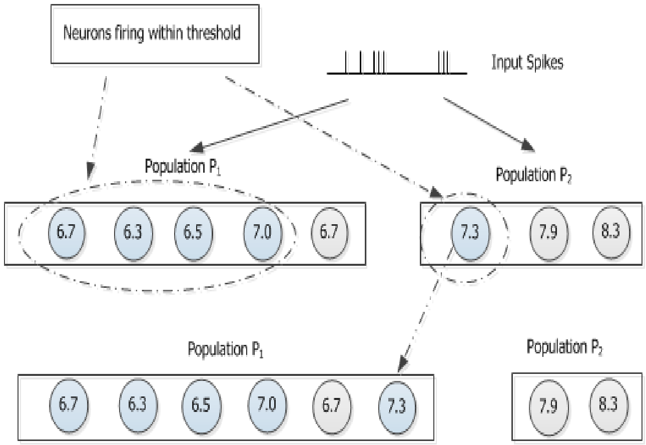

Figure 3: Merging of two populations.

During merging, neurons from populations that fire closer to neurons in P_{max} are merged with P_{max}. In Fig. 3 a neuron from population P_2 (neuron with firing time 7.3) gets added to population P_1 after the merging process. The threshold for merging is taken as 1 millisecond (a maximum standard deviation $\sigma_{max} \leq 0.8$ means the difference between the firing times of two neurons is equal to or below 1 millisecond). The neuron with firing time 7.3 from P_2 is moved to P_1 as they fire together for the given input and the standard deviation of firing time 7.3 from the firing times of all neurons in P_1 is either less than or equal to the threshold. P_2 gets fully merged to P_1 when all neurons in P_2 fire within the merging threshold.

C. Pruning within a Population

Pruning of neurons within a population is performed to remove unresponsive neurons**.** Pruning within a population removes neurons that do not contribute to clustering of a given input and it is done when the population response is above γ_{min}. Pruning within a population involves the following steps:

- *Let $n = n_1, n_2 \ldots n_m$ be the neurons of a population P with $\gamma \geq \gamma_{min}$.*
- *Let $n' = n_1, n_2 \ldots n_j$ be a subset of n with firing times above the maximum standard deviation from P centre.*
- *Neuron n_i Є n' will be removed from P, if there exist another neuron n_j Є n, $n_{j'} \notin n'$, with similar firing times t.*

Fig. 4 illustrates the pruning process. Population P has two neurons with firing times [6.7 7.0] milliseconds. There exist another two neurons with the same firing times. However, they don’t respond to the given input. Hence they are pruned from the population. Pruning within a population will increase population response γ as unresponsive neurons are removed from the population. It will also reduce the execution time as the number of neurons is reduced.

D. *Interpreting the output and cluster mapping*

Cluster mapping is done based on the output of population defined by population response γ which indicates the responsiveness of a population to a given input. The population with maximum γ is taken as the respective cluster. If there is more than one population equal to γ then the larger population is taken as the cluster. In previous studies clustering is done by identifying a winning neuron by interpreting the firing times of neurons in the output layer. Population based clustering depends on identifying a winning population from various competing populations.

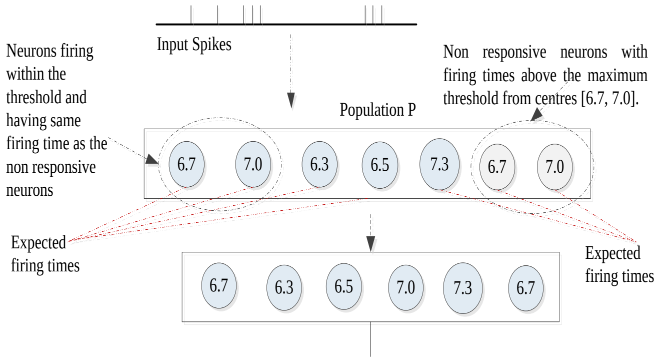
Input Spikes
Neurons firing within the threshold and having same firing time as the non responsive neurons
Non responsive neurons with firing times above the maximum threshold from centres [6.7, 7.0].
Population P
6.7
7.0
6.3
6.5
7.3
6.7
7.0
Expected firing times
Expected firing times
6.7
6.3
6.5
7.0
7.3
6.7

Figure 4: Pruning of neurons within a Population

IV. RESULTS

In order to evaluate the viability of the proposed algorithm, initial experiments were performed on a linearly and a non-linearly easily separable synthetic 2D datasets. Fig. 5(a) shows a scatter plot of the easily linearly separable synthetic dataset. The proposed system clustered both data sets successfully with 100% accuracy. After successful clustering of the first two trivial synthetic datasets the proposed system was then demonstrated on two selected benchmark datasets, namely the four-dimensional Iris dataset and the 9-dimensional Wisconsin Breast Cancer dataset. The parameter values for the spiking neurons were taken from [5] and the STDP parameters were adopted from [7]. The obtained clustering accuracies for these datasets are summarized in Table 1. The neuron membrane potential threshold θ along with the maximum merging standard deviation σ_{max} are two important parameters that control the evolution of populations. For all experiments θ was selected between 0.8 and 1.2. A 10 millisecond input window was used for all clustering experiments. A value of 0.8 was used for σ.

Ten-fold cross validation of Iris dataset achieved a mean accuracy of 90.3 %. The number of clusters varied between 3 and 4 depending on the samples fed during online training. At times the number of clusters was also reduced to 2 producing less accurate separation between the less easily separable clusters of the Iris data set. Ten-fold cross validation of the Wisconsin Breast Cancer dataset produced a mean accuracy of 87.9%. For most datasets 50% of the data samples were used for online training except for the Wisconsin Breast Cancer dataset where only 30% were used.

Figure 5(b) shows two different population centers obtained during the clustering of the Iris dataset. It is evident that the number of neurons and their firing times can be very different and depends on the order in which data samples were presented to the clustering system. Figure 6(a) shows the evolution of the number of populations representing clusters for the Iris dataset. The number of populations (hence the number of clusters) starts with one. As data samples from other clusters are fed to the system, new populations are created to characterise them. Fig. 6(a) shows how the population size fluctuates as populations are added and pruned through merging of populations with close centers which might represent similar clusters. For the Iris dataset, the number of populations varied between 2 and 3 for the first 30 samples, then kept fluctuating between 3 and 4 thereafter until settling on 3 populations (after seeing around 60 samples from Iris dataset). On the other hand, Fig. 6(b) shows the evolution of neurons within one of the populations created during clustering of the Iris dataset. The number of neurons increases from 1 to 7 as new data samples are fed to the system and neurons are grown and pruned (note how the population size dropped twice from 5 neurons to 4 neurons after the 30th and the 50th data samples were presented). The centre of the population is represented by the firing times [6.7, 6.5, 6.4, 6.3, 6.5, 6.4, 6.3] milliseconds.

TABLE I CLUSTERING ACCURACY OBTAINED USING POPULATION BASED SNN FOR VARIOUS DATASETS.

Dataset	Clusters	Populations	Samples	Accuracy
Linearly-separable synthetic data	2	2	100/200	100%
Non-linearly separable synthetic data	3	3	150/300	100%
Iris	3	3/4	75/150	90.3%
Breast Cancer Wisconsin (Diagnostic)	2	3/4	200/683	87.9%

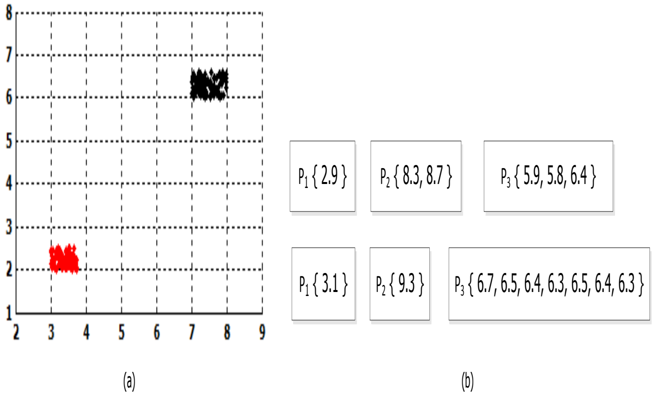

Figure 5: a) Scatter plot of 2 cluster easily separable synthetic data.
b) Population centres for Fisher's Iris dataset during two different iterations.

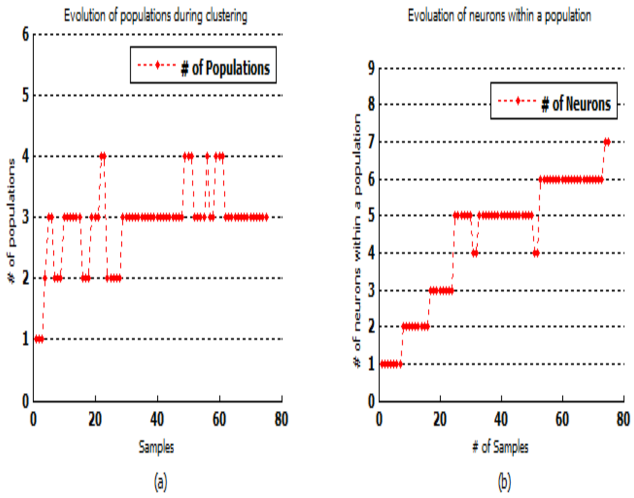

Figure 6: a) Evolution of populations representing clusters of Iris dataset. b) Evolution of neurons within one of the populations created during Iris clustering

The maximum accuracy achieved is comparable to conventional and other existing offline spiking neuron based models. However, the proposed spiking neuron population-based online clustering training used a single pass only with a limited number of data samples. The obtained results demonstrate the effectiveness of the proposed clustering system and indicate the viability of using spiking neuron populations for dynamic formation of clusters in an online manner with adaptive structure of the spiking neural network. This work is ongoing and more experiments on high dimensional and very large scale datasets are planned in order to further test the scalability of this online clustering system.

V. Conclusion and Future work

This paper introduced a novel adaptive spiking neural network for online clustering of temporally encoded input patterns using populations of spiking neurons. Clustering is achieved through dynamic formation of populations of spiking neurons through adding and pruning. STDP-based learning rule was utilized to adapt the synaptic weights and enhance the selectivity. Data samples were successfully clustered using a single pass only and very good accuracy was achieved even when a only a limited number of data samples are used. The obtained results demonstrate the effectiveness of the proposed clustering system and indicate the viability of spiking neuron populations online clustering

Potential extensions of this work include the exploration of alternative input encoding schemes and spike trains similarity metrics which may further enhance the performance of the proposed population-based online clustering. Also, the use of spike trains instead for single spikes is another option for input patterns encoding. The incorporation of an appropriate feature selection strategy in the proposed architecture will further enhance the suitability of the proposed clustering system to process high dimensional datasets and add to the versatility of the system.

References

[1] I. B. Levitan, and L. K. Kaczmarek, “The Neuron: Cell and Molecular Biology”, 3rd ed. Oxford University Press, 2002.

[2] D.T. Pham, M.S. Packianather, E.Y.A. Charles, "A self-organising spiking neural network trained using delay adaptation," Industrial Electronics, 2007. ISIE 2007. IEEE International Symposium on , vol., no., pp.3441-3446, 4-7 June 2007

[3] B.B. Averbeck, P.E. Latham, and A. Pouget, “Neural correlations, population coding and computation”. Nat Rev Neurosci. 2006;7:358–366.

[4] J. Wang; A. Belatreche, L. Maguire, M. McGinnity, Online versus offline learning for spiking neural networks: A review and new strategies, 2010 IEEE 9th International Conference on Cybernetic Intelligent Systems (CIS), doi: 10.1109/UKRICIS.2010.5898113

[5] Deneve S, Latham PE, Pouget A (1999) Reading population codes: a neural implementation of ideal observers. Nat Neurosci 2: 740–745.

[6] Pouget A, Dayan P, Zemel R (2000) Information processing with population codes. Nat Rev Neurosci 1: 125–132.

[7] Jazayeri M, Movshon JA (2006) Optimal representation of sensory information by neural populations. Nat Neuroscience 9: 690–696

[8] Boerlin M, Dene`ve S (2011) Spike-Based Population Coding and Working Memory. PLoS Comput Biol 7(2).

S. G Wysoski , L. Benuskova , N. Kasabov, On-line learning with structural adaptation in a network of spiking neurons for visual pattern recognition, ICANN, 2006.

[9] W. Grestner, Spiking Neurons. In: Pulsed Neural Networks, W. Maass and C. Bishops, eds., MIT Press (Cambridge), 1999, pp.3-53

[10] S.M. Bohte, H. La Poutre, J.N. Kok, "Unsupervised clustering with spiking neurons by sparse temporal coding and multilayer RBF networks," Neural Networks, IEEE Transactions on , vol.13, no.2, pp.426-435, Mar 2002

[11] L.F. Abbott, and S.B. Nelson, “*Synaptic plasticity: Taming the beast*.”, Nat. Neurosci. vol.3, pp.1178 -1183, 2000

[12] S. Song, K. D. Miller and L. F. Abbott, “Competitive Hebbian learning through spike-timing-dependent synaptic plasticity”. Nature Neurosci. vol.3, no.9, pp.919−926, 2000

[13] W. Maass, “On the relevance of time in neural computation and learning.”, Theoretical Computer Science, 261, pp.157–178, 2001 (extended version of ALT’97, in LNAI 1316:364-384).

[14] R. Quian Quiroga and S. Panzeri, “Extracting information from neuronal populations: information theory and decoding approaches.” Nature Rev. Neurosci, vol.10, pp.173–185, 2009

[15] T. Natschläger, and B. Ruf, “Spatial and temporal pattern analysis via spiking neurons”, Network: Comp. Neural Syst., vol.9, pp. 319–332, 1998