

Unsupervised Classification of Complex Clusters in Networks of Spiking Neurons

Sander M. Bohte¹, Joost N. Kok² and Han La Poutré¹

¹CWI, Kruislaan 413, 1090 GB Amsterdam, P.O. Box 94079, The Netherlands, {S.M.Bohte, Han.La.Poutré}@cwi.nl

²LIACS, Leiden University, 2300 RA Leiden, P.O. Box 9512, The Netherlands, joost@liacs.nl

Abstract

For unsupervised clustering in a network of spiking neurons we develop a temporal encoding of continuously valued data to obtain arbitrary clustering capacity and precision with an efficient use of neurons. Input variables are encoded independently in a population code by neurons with 1-dimensional graded and overlapping sensitivity profiles. Using a temporal Hebbian learning rule, the network architecture yields reliable clustering of high-dimensional multi-modal data. Additionally, multi-scale sensitivity to the input is achieved by using an appropriate choice of local activation functions. To correctly classify non-spherical clusters, we present a multi-layer version of the algorithm to perform a form of hierarchical clustering. Together with the addition of lateral excitatory connections in the hidden layer, this enables the correct classification of complex, interlocking clusters by synchronizing the neurons coding for parts of the same cluster. We show how synchronous spiking of neurons can emerge with a local Hebbian learning rule and can be exploited by subsequent RBF layers employing the same local learning rule. Neuronal synchrony thus naturally enhances the clustering capabilities of artificial spiking neural networks, much as has been suggested in neurobiology.

1 Introduction

As neurons in the brain produce all-or-none action potentials, the significance of the timing of these pulses has only recently been recognized as a means of neuronal information coding. As the biological evidence has been mounting (for a review, see [1]), it has been shown theoretically that such temporal coding enables powerful information processing by neurons [2], and that it could be instrumental in solving dynamic combinatorial problems [3]. These considerations have generated considerable interest in time-based artificial neural networks. Hopfield [4] presents a model of spiking neurons for discovering clusters in an input space akin to Radial Basis Functions. Natschläger & Ruf [5] present a learning algorithm for spiking neural networks that performs unsupervised clustering using spike-times as input. This model encodes the input patterns in the delays across its synapses, and is shown to reliably find centers of high-dimensional clusters but is limited in both cluster capacity as well as precision (discussed in detail in section 2).

In this paper we present an architecture that is able to deal with real-world data by enabling unsupervised clustering over multiple real-valued inputs and enhancing the precision and capacity of the network in a flexible and scalable manner, thus solving the problems associated with the network architecture described in [5]. Inspired by the local receptive fields of biological neurons, we encode continuous input variables by neurons with graded and overlapping sensitivity profiles. In addition, each input dimension of a high dimensional dataset is encoded independently, avoiding an exponential increase in the number of input neurons. This is shown to effectively enable the network to successfully classify a number of clustering problems at relatively low expense in terms of neurons. The multi-scale encoding allows for the reliable detection of clusters over a considerable and flexible range of spatial scales. This is especially significant for unsupervised classification tasks, where scale-information is a-priori unknown.

By extending the network to multiple layers, we show how the temporal aspect of spiking neurons can be exploited to enable the correct classification of non-globular or interlocking clusters. In a multi-layer RBF network, we demonstrate that the neurons in the first layer specialize on components of clusters. The

(near) synchrony of neurons coding for nearby components is then shown to be easily distinguishable by a downstream RBF layer resulting in a form of hierarchical clustering. The addition of lateral excitatory connections is then shown to enable such a multi-layer network to correctly classify complex, interlocking structures by synchronizing the neurons coding for parts of the same cluster. The additional lateral connections thus yield a low neuron count while increasing the complexity of classifiable clusters and still use only a local Hebbian learning rule.

Summarizing, we describe a network of spiking neurons that is capable of clustering high-dimensional data with arbitrary precision and cluster capacity. This is achieved by employing a 1-dimensional array of graded overlapping receptive fields for the encoding of each variable. The introduction of a multi-layer extension of the architecture is then shown to allow for the clustering of complex clusters.

2 Networks of delayed spiking neurons

Network Architecture. The network architecture consists of feedforward connected spiking neurons of the leaky integrate-and-fire type. As illustrated in figure 1A, connections in the network consists of a set of synapses k , each with weight w_{ij}^k and delay d^k . An input spike generates a set of weighted post-synaptic potentials (PSP's, a function modeling the impact of a spike on the target neuron as a function of time). The size of the synaptic weights determines the height of the PSP. At every time-step of the algorithm, the values of these PSP's are summed at the target neuron, and when the sum exceeds threshold θ , an outgoing spike is generated.

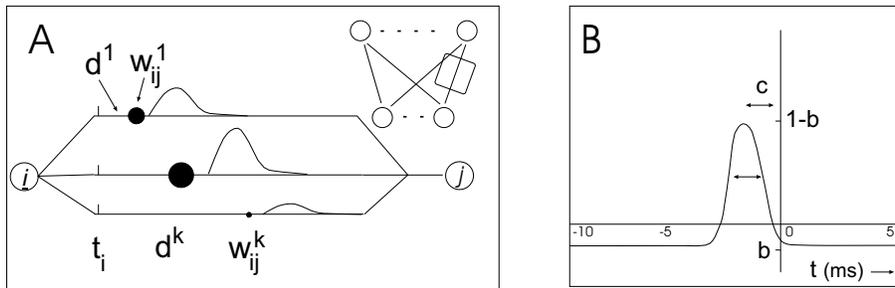


Figure 1: A) Network connectivity and a single connection composed of multiple delayed synapses. B) Graph of the learning function $L(\Delta t)$. Δt denotes the time-difference between the onset of a PSP at a synapse, and the time of the spike generated in the target neuron.

Input patterns can be encoded in the synaptic weights by delay-learning. After learning, the firing time of an output neuron reflects the distance of the evaluated pattern to its learned input pattern (thus realizing a kind of RBF neurons). For unsupervised learning, the weights between the source neurons and the neuron first to fire in the target layer (Winner-Take-All) are modified using a time-variant of the Hebbian learning rule. If the start of the PSP at a synapse slightly precedes an action potential in the target neuron, the weight of this synapse is increased, as it exerted significant influence on the spike-time via a relatively large contribution to the membrane potential. Earlier and later synapses are decreased in weight, reflecting their lesser impact on the neuron's spike time. For a weight with delay d_k from neuron i to neuron j we use

$$\Delta w_{ij}^k = \eta L(\Delta t) = \eta(1-b) \exp\left[\frac{(\Delta t - c)^2}{\beta^2}\right] + b, \quad (1)$$

after [5] (depicted in figure 1B). The weights are limited by a minimum and maximum value. An input to the network is coded by a pattern of firing times within a coding interval and each input neuron is required to fire once during this coding interval. In our experiments, we set a continuous coding interval of 0 – 9ms and a delay interval of 1 – 15ms in 1ms intervals.¹

¹Unless indicated otherwise, the parameter values for the learning function used in [5] are set to: $b = -0.2$, $c = -2.85$, $\beta = 1.67$, $\eta = 0.0025$ and $w_{max} = 2.75$. To model the (strictly excitatory) post-synaptic potentials, we used an α -function $\varepsilon(t) = t/\tau \exp(1-t/\tau)$, with $\tau = 3.0$ ms.

Previous Results and Open Questions. It has been shown that artificially constructed clusters of inputs firing within the encoding interval are correctly clustered in an unsupervised manner, also when the clusters were positioned in a subspace of the input space. The results on unsupervised clustering in [5] refer to a situation where a simple coding scheme is employed: each input neuron fires exactly once within a certain time interval $[0, T]$, the coding interval. A cluster is defined as a set of times $C = \{T_1, \dots, T_N\}$, with T_i designating the spike-time of input neuron i for cluster C . For such a setup it was found that the RBF neurons converged reliably to the centers of the clusters, also in the presence of noise and randomly spiking neurons.

In practice, several important problems arise. A first problem concerns the *coding* of input: following the aforementioned method [5], it proved impossible to cluster data containing significantly more clusters than input-dimensions. This problem is associated with the minimum width β of the learning function $L(\Delta t)$, leading to a fixed minimal spatial extend of a learned cluster, potentially (much) larger than the actual cluster size. In fact, for 2-dimensional input containing more than two clusters, the above algorithm failed in our experiments for a wide range of parameters. Furthermore, the finite width of the learning rule effectively inhibits the detection of multiple nearby clusters of smaller size (relative to the width of the learning function), requiring advance knowledge of the effective cluster-scale. Hence, to achieve practical applicability, it is necessary to develop an encoding that is scalable in terms of cluster capacity and precision and that is also efficient in terms of the number of input-neurons required. In this paper, we present improvements to the architecture that solve these problems.

3 Encoding continuous input variables in spike-times

To extend the encoding precision and clustering capacity, we investigate the use of multiple local receptive fields to distribute an input variable over multiple input neurons. Such a population code where input variables are encoded with graded and overlapping activation functions is a well-studied method for representing real-valued parameters (eg.: [6]). In these studies, the activation function of an input-neuron is modeled as a local receptive field that determines the firing rate. A translation of this paradigm into relative firing-times is straightforward: an optimally stimulated neuron fires at $t = 0$, whereas a value up to say $t = 9$ is assigned to less optimally stimulated neurons (depicted in figure 2).

For actually encoding high-dimensional data in the manner described above, a choice has to be made with respect to the dimensionality of the receptive-fields of the neurons. The least expensive encoding in terms of neurons is that where each input is encoded independently with 1-D receptive fields. Since we are interested in high-dimensional clustering, this encoding is highly desirable as it behaves linearly in terms of neurons required per dimension, and is also tunable by allowing for the encoding of some dimensions with higher precision without excessive neuron cost.

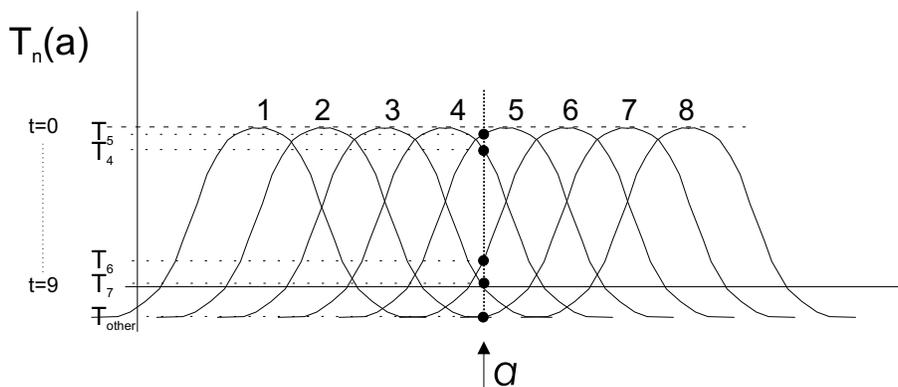


Figure 2: Encoding with overlapping Gaussian receptive fields. An input value a is translated into firing times for the input-neurons encoding this input-variable. The highest stimulated neuron (5), fires at a time close to 0, whereas less stimulated neurons, as for instance neuron 7, fire at increasingly later times.

4 Clustering with Receptive Fields

To study the enhanced capacity and precision provided by the encoding described, we examined the classification of a number of artificially generated problems, both for low- as well as for high-dimensional input. When clustering input consisting of two separately encoded variables, we found that a network consisting of 24 input neurons (each variable encoded by 12 neurons) was easily capable of correctly classifying 17 evenly distributed clusters, demonstrating a significant increase in the clustering capacity. Similarly good results were also obtained for less regularly spaced clusters (more elaborately described in [7]). It was also found that by decreasing the width of the receptive fields, increasingly narrowly separated clusters could be distinguished, as predicted by the results from [6].

Encoding input variables with local Gaussian receptive fields incorporates an inherent choice of scale by fixing the width of the receptive fields. Encoding with a mix of neurons with varying receptive field widths proved to significantly enhance the range of detectable scales, in experiments at least by up to an order of magnitude, and in general improved the clustering reliability. We thus achieve a significant amount of scale-adaptability in the classification. In an unsupervised setting, scale is typically not, or not well, known. The multi-scale adaptability thus adds a desirable property to the architecture.

Good results were also obtained classifying high-dimensional artificial input (10-d+), and the clustering of the 4-dimensional iris data-set, encoded in 4x8 input neurons, yielded 90 to 97% correct classification, comparable to k-nearest neighbor results. The results demonstrate that the encoding used preserves the desirable high-dimensional clustering properties (described in more detail in [7]).

5 Clustering in a multi-layer network

Hierarchical clustering. As noted in [5], a single layer spiking RBF network with more outputs than clusters was found to center multiple output-neurons on the same cluster and correct classification was only reliably achieved when the number of RBF neurons matched the number of clusters. We found the same effect, especially when clusters consisted of multiple components.

As it could be beneficial to find the individual components constituting clusters, we modified the Winner-Take-All learning rule to allow all neurons in a layer to fire, but only for the winning neuron the weights are modified. In the case of an excess of RBF neurons, learning will center the respective neurons either on the same cluster-component, or on different cluster-components. By interpreting the firing times of the output-neurons, we can then discriminate between clusters, as the neurons centered on components of the respective clusters will differ significantly in firing time. Different components of the same cluster can be identified in the same way, as they tend to fire relatively close to each other. Moreover, when RBF neurons are centered on the same cluster component, they will fire almost perfectly synchronous.

Such “interpretation” is also possible by adding a second RBF layer to the network. As shown in figure 3, a multi-layer RBF network can thus perform a kind of hierarchical clustering. Here, we trained a multi-layer network on data containing two clusters each consisting of two components. The winner-take-all classification found in the first RBF layer is shown in figure 3a, and clearly identifies the respective components of the two clusters. For such a configuration, the receptive field of any RBF neuron extends over the accompanying component. In this case, the evaluation of a single data point elicits a response from both neurons in the cluster, albeit one somewhat later than the other. The neurons centered on the other cluster are insufficiently stimulated to elicit a significant response. This disparate response is sufficient for the next layer to concatenate the neurons in the first layer into two clusters 3b. Thus, the addition of subsequent RBF layers comprising of fewer neurons to the network in effect results in a hierarchical clustering as nearby components are clustered together based on relative proximity (as expressed in their temporal distance).

In this hierarchical clustering example we tuned the number of neurons to the number of components and clusters. However, in the case of more neurons than components/clusters the same hierarchical clustering principle holds as multiple neurons centered on the same component are identifiable by their strong synchrony.

As indicated, without prior knowledge of the number of classes, the Winner-Take-All paradigm is prone to incorrect classification by spreading a cluster over multiple classes. The relative synchronization of nearby neurons is an important clue when extracting the classification from a layer, as well as an effective means of coding for further (hierarchical) neuronal processing. The simultaneous availability of such hierarchical

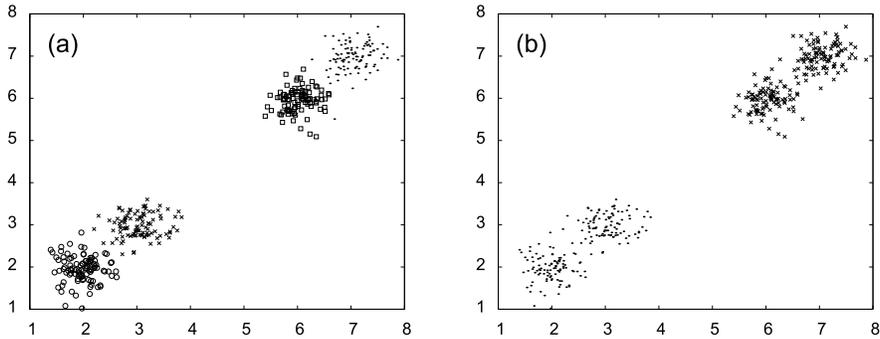


Figure 3: Hierarchical clustering in a 2 layer RBF network. A) Clustering in the first layer consisting of 4 RBF neurons. Each data-point is labeled with a marker designating the winning neuron (squares, circles crosses and dots). B) Clustering in the second layer, consisting of 2 RBF neurons. Again each data-point is labeled with a marker signifying the winning neuron. (crosses and dots).

clustering information could be useful for applications such as iterative cluster-visualization.

Complex clusters. Further exploitation of temporal synchrony via the addition of reciprocal excitatory lateral connections to the first RBF-layer can further enhance the clustering capabilities of the multi-layer RBF network. When modifying the lateral connections between the winner and the other neurons in the same layer with the temporal Hebbian learning rule, we create in effect a correlation layer. As only lateral connections between neurons that fire close to each other in time are augmented, ultimately neurons centered on the same cluster are synchronized due to the data points that lie “in between” neighboring neurons. These points elicit approximately the same time-response from the nearest neurons, strengthening their mutual connections. This process does not take place for neurons coding for different clusters, due to the relative lack of points “in between”. As the different sets of neurons associated with each structure synchronize their respective firing-times, the temporal response from the hidden layer enables a correct classification in the second layer.

Such lateral connections were implemented in the multi-layer network and modeled as the feedforward connections, albeit with only one delay bin (1 ms). The Winner-Take-All rule was adapted to modify the lateral connections from the winning neuron: its weights were modified using the learning function: $L(\Delta t) = e^{-\Delta t^2/b^2} \{(1 - c)e^{-\Delta t^2/\beta^2} + c\}$, with $b = 4.5$, $c = -0.2$, $\beta = 0.8$. During learning, the maximal allowed strength of the lateral connections was slowly increased from 0 to a value sufficiently strong to force connected neurons to fire. Experiments with these connections incorporated in the multi-layer network yielded the correct classification of complex, interlocking clusters (as is shown in fig 4). Thus, the addition of lateral excitatory connections enables the correct clustering of complex non-linear clusters in a multi-layer network, without requiring additional neurons.

6 Conclusions

Concluding, we investigated unsupervised clustering with a network of spiking neurons for continuously-valued multi-modal input, where we defined multi-modality as the independent encoding of the inputs. This has been shown to be efficient in terms of required neurons. To detect non-globular interlocking clusters we also used hierarchical clustering and exploited the synchronized firing of neurons coding for different parts of the same cluster when excitatory lateral connections were added. This combination added significantly to the clustering capabilities, while the number of neurons required remained small. Also, to the best of our knowledge, this is the first paper where it is demonstrated how a local Hebbian learning-rule can both induce and exploit synchronous neurons resulting in enhanced unsupervised clustering capabilities, much as theorized in neurobiology.

We note however that, as the clustering results of the Iris dataset indicate, high-dimensional clusters may

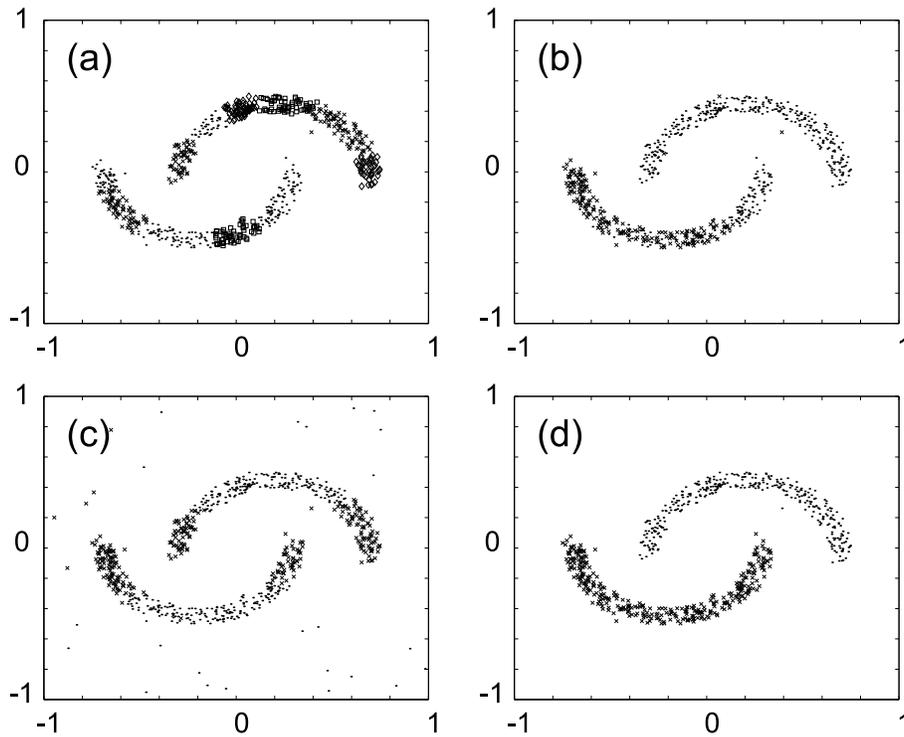


Figure 4: Clustering of two interlocking clusters in a multi-layer RBF network. (a) classification in the first layer: 11 outputs, the clusters are spread over 5 respectively 6 neurons (the respective classifications are denoted by different markers). (b) Clustering in the second layer with two RBF neurons and input from (a), without lateral connections. (c) Classification as obtained in a single-layer network. (d) Classification in the second layer, with lateral connections. Each input variable was encoded by 12 input neurons (3 broadly and 9 sharply tuned).

often be sufficiently Gaussian-like and thus for problems like unsupervised remote-sensing classification our multi-modal encoding approach could already be effective without the need to actually find more complicated clusters. Indeed, preliminary analysis of real-world remote-sensing data has yielded promising results.

References

- [1] W. Singer and C.M. Gray. Visual feature integration and the temporal correlation hypothesis. *Annu. Rev. Neurosci.*, 18:555–586, 1995.
- [2] W. Maass. Networks of spiking neurons: The third generation of neural network models. *Neural Networks*, 10(9):1659–1671, 1997.
- [3] C. von der Malsburg and W. Schneider. A neural cocktail-party processor. *Biological Cybernetics*, 54:29–40, 1986.
- [4] J.J. Hopfield. Pattern recognition computation using action potential timing for stimulus representation. *Nature*, 376:33–36, 1995.
- [5] T. Natschläger and B. Ruf. Spatial and temporal pattern analysis via spiking neurons. *Network: Computation in Neural Systems*, 9(3):319–332, 1998.
- [6] H.P. Snippe. Parameter extraction from population codes: A critical assessment. *Neural Computation*, 8(3):511–529, 1996.
- [7] S.M. Bohte, J.N. Kok, and J.A. La Poutré. Unsupervised classification in a layered network of spiking neurons. *CWI Technical Report*, in preparation, www.cwi.nl/~sbohte.