Learning and data clustering with an RBF-based spiking neuron network

Natacha Gueorguieva\textsuperscript{a}, Iren Valova\textsuperscript{b} & Georgi Georgiev\textsuperscript{c}

\textsuperscript{a} Computer Science, CSI/City University of New York, 2800 Victory Boulevard, Staten Island, NY 10314, USA

\textsuperscript{b} Computer Science, University of Massachusetts Dartmouth, 285 Old Westport Road, N. Dartmouth, MA 02747, USA

\textsuperscript{c} Computer Science, University of Wisconsin Oshkosh, 800 Algoma Boulevard, Oshkosh, WI 54901, USA

Published online: 20 Feb 2007.

To cite this article: Natacha Gueorguieva, Iren Valova & Georgi Georgiev (2006) Learning and data clustering with an RBF-based spiking neuron network, Journal of Experimental & Theoretical Artificial Intelligence, 18:01, 73-86, DOI: 10.1080/09528130600552888

To link to this article: http://dx.doi.org/10.1080/09528130600552888
Learning and data clustering with an RBF-based spiking neuron network

NATACHA GUEORGUIEVA*†, IREN VALOVA‡ and GEORGI GEORGIEV§

†Computer Science, CSI/City University of New York, 2800 Victory Boulevard, Staten Island, NY 10314, USA
‡Computer Science, University of Massachusetts Dartmouth, 285 Old Westport Road, N. Dartmouth, MA 02747, USA
§Computer Science, University of Wisconsin Oshkosh, 800 Algoma Boulevard, Oshkosh, WI 54901, USA

(Received 16 September 2004; in final form 22 December 2005)

A spiking neuron is a simplified model of the biological neuron as the input, output, and internal representation of information based on the relative timing of individual spikes, and is closely related to the biological network. We extend the learning algorithms with spiking neurons developed by earlier workers. These algorithms explicitly concerned a single pair of pre- and postsynaptic spikes and cannot be applied to situations involving multiple spikes arriving at the same synapse. The aim of the algorithm presented here is to achieve synaptic plasticity by using relative timing between single pre- and postsynaptic spikes and therefore to improve the performance on large datasets. The learning algorithm is based on spike timing-dependent synaptic plasticity, which uses exact spike timing to optimize the information stream through the neural network as well as to enforce the competition between neurons during unsupervised Hebbian learning. We demonstrate the performance of the proposed spiking neuron model and learning algorithm on clustering and provide a comparative analysis with other state-of-the-art approaches.

Keywords: Spiking neuron; Neural networks; Learning algorithm; Radial basis function; Data clustering
1. Introduction

Analysis of large datasets is required in many applications and scientific research areas. This becomes difficult when the dataspace is high dimensional, i.e. when the number of features that characterize each point is very large. Solutions to such problems often rely exclusively on the distances between the data points. This is the basis of many clustering methods that use these distances as a dissimilarity measure. The basic assumption is that points within a cluster are closer to one another than to points in other clusters, which leads to a search for groups of points where the distances within each group cluster are much smaller than the distances between points in different clusters.

Spiking neuron networks (SNNs), in which input, output and internal representation of information are based on the relative timing of individual spikes, are closely related to the biological network. The theoretical results show that coding with the timing of single spikes allows simulation of powerful neuronal information processing. Spiking neurons receiving temporally encoded inputs can compute radial basis functions (RBFs) by storing the relevant information in their delays, which can be determined using the time difference between the pre- and postsynaptic spike (Maas 1998). Theoretically, SNNs can perform very powerful computations with precisely timed spikes. Simulations with SNNs show that they are computationally more effective than networks composed of threshold gates or sigmoid gates. However, learning algorithms for spiking neural networks are still lacking.

Hopfield’s (1995) approach to discovering clusters in an input space based on RBFs with spiking neurons was based on neurobiological findings (Haberley 1985, O’Keefe and Reece 1993). Evidence of neurobiological processes which resemble the approaches described in this research has been reported by (O’Keefe and Reece 1993, Markram and Tsodyks 1996). The Hopfield model for RBFs with delays uses the fact that the encoding type of SNNs is temporal, i.e. SNN models conform to the analysis of time-structured data. Therefore an RBF neuron can be used to encode a particular input spike pattern with delays. If an input pattern is similar to the respective encoded spike pattern (centre of the RBF neuron), the delays balance out the differences between the firing times of the input neurons and the RBF neurons.

A learning mechanism for a single RBF neuron based on the choice of appropriate delays to encode the centre of an RBF neuron has been introduced (Maas 1996, Bohte et al. 2000). The proposed learning rule includes a mechanism for increasing the efficacy of synapses with corresponding suitable delays and decreasing the efficacy of other synapses.

Hopfield’s idea was extended by Natschläger and Ruf (1998) who proposed a learning algorithm that performs unsupervised clustering in spiking neural networks using spike times as input. Their model encodes the input patterns in the delays across its synapses. It is reliable in finding the centres of high-dimensional clusters but shows some limitations in both cluster capacity and precision.

Here we extend the above approach by taking into consideration not only the firing of a neuron, the firing time, and delays, but also multiple delays between the input and output neurons based on the assumption that there are several paths with different delays between each input and each output neuron. We show how, on the basis of these ideas, networks of such RBF neurons can be constructed and trained to divide the input space into several clusters, where the selection of the proper delays
depends on the difference between pre- and postsynaptic firing time. Using the evidence that the temporal coding of biological systems relies on the timing of single spikes (Gerstner et al. 1996, Maas 1997, Bohte et al. 2000), we develop a learning algorithm for spiking neurons realizing multiple layers with RBF based on temporal spike-time coding. Because of the relative novelty of this research area, there are few applications of SNNs to real data. Therefore, we apply our model to the iris problem and compare the results with those achieved using other approaches.

2. Spiking neurons: a mathematical model

Our neural network model is based on the spike response model (SRM) introduced by Gerstner (1995). It includes synaptic delays determined by the difference between the presynaptic firing time and the time when the postsynaptic potential starts rising. To reflect the dynamics of spiking neurons we use a class of functions known as $\alpha$-functions as the spike response (activation) function. It is modelled by

$$
\epsilon(\delta) = \begin{cases} 
\frac{\delta}{\tau} \exp(1 - \frac{\delta}{\tau}), & \text{if } \delta > 0 \\
0, & \text{if } \delta < 0
\end{cases}
$$

where $\delta$ is the time that has passed since the onset of the spike’s effect and $\tau$ is the membrane potential decay time constant which determines the rise time to reach the peak of the postsynaptic reaction as well as its decay time.

The state of a spiking neuron $n_i$ is described by an internal state function $u_i(t)$ which models the membrane potential of neuron $n_i$ at time $t$. An incoming presynaptic spike alters the neuron membrane voltage and suddenly increases its value. Whenever $u_i(t)$ crosses the threshold $\theta$, neuron $n_i$ fires, generating an action potential (spike). Let us denote the set of such firing times for neuron $n_i$ (the times when the membrane potential crosses the threshold) by $F_i = \{t^1_i, t^2_i, t^3_i, \ldots\}$ where $t^{(q)}_i$ is one of the neuron’s firing times (figure 1). Therefore

$$
F_i = \{t^1_i, t^2_i, t^3_i, \ldots\} = \{t^{(q)}_i, 1 \leq q \leq l\} = \{t|u_i(t) = \theta\}.
$$

After the spike, the neuron undergoes an absolute refractory period in which no other spike can be generated by $n_i$ (figure 2). Then, for a longer interval (denoted the

Figure 1. Spike response model: two postsynaptic potentials generated by presynaptic neurons with some delay.
relative refractory period) following the absolute refractory period, it is more difficult for neuron \( n_i \) to generate an action potential. The relative refractory period has a duration of up to tens of milliseconds. Mathematically, the above process is represented by the negative contribution of function \( h_i(t - t_i^{(q)}) \) to the membrane potential \( u_i(t) \) in equation (2) below.

Neuron \( n_i \) receives inputs from the presynaptic neurons \( n_j \in G_i \) where \( G_i = \{ j | j \text{ presynaptic to } i \} \). A presynaptic single spike generated by neuron \( n_j \) at time \( t_j^{(q)} \) increases or decreases the membrane potential \( u_i(t) \) of neuron \( n_i \) for \( t > t_j^{(q)} \) by the amount of the kernel \( w_{ij} \varepsilon_{ij}(t - t_j^{(q)}) \) in equation (2) below. The latter is referred to as the postsynaptic potential (PSP) and models how the arrival of a single weighted spike sent by the source neuron \( n_j \) changes the membrane potential of the target neuron \( n_i \) as a function of the time since impact. The synaptic weight (efficacy) \( w_{ij} \) represents the strength of the connection between the neurons \( n_i \) and \( n_j \). PSP can be either positive (excitatory), denoted EPSP, or negative (inhibitory), denoted IPSP. Whereas spikes are very similar, the PSPs differ in size.

The total membrane potential \( u_i(t) \) is modelled as a linear superposition of the presynaptic potentials represented by the kernel \( \varepsilon_{ij}(t - (t_j^{(q)} + d_{ij})) \):

\[
u_i(t) = \sum_{j \in G_i} \sum_{\ell_j^{(q)} \in F_j} w_{ij} \varepsilon_{ij} \left( t - (t_j^{(q)} + d_{ij}) \right) + \sum_{t_j^{(q)} \in F_i} h_i \left( t - t_j^{(q)} \right)
\] (2)

where \( t_j^{(q)} \) is one of the presynaptic firing times, \( d_{ij} \) models the delay from the occurrence of a spike in neuron \( n_j \in G_i \) to the onset of its effect on neuron \( n_i \), \( (t_j^{(q)} + d_{ij}) \) is the time when the spike sent from neuron \( n_j \in G_i \) at firing time \( t_j^{(q)} \) starts to affect neuron \( n_i \), and \( G_i \) is the set of presynaptic neurons sending outputs to neuron \( n_i \).

Therefore the membrane potential \( u_i(t) \) can be represented as follows:

\[
u_i(t|t_i^{\text{last\_fire}}) = h_i(t - t_i^{\text{last\_fire}}) + u_i(t - 1)
\] (3)

where \( t_i^{\text{last\_fire}} \) is the last firing time of neuron \( n_i \) and \( h_i(t - t_i^{\text{last\_fire}}) \) is the negative contribution to the membrane potential \( u_i(t) \) of neuron \( n_i \) after the spike. Thus \( t_i^{\text{last\_fire}} \) can be obtained from the equation

\[
 t_i^{\text{last\_fire}} = \max \left\{ t_i^{(q)} | t_i^{(q)} > t \right\}.
\] (4)
the effect of the last firing $t^\text{last-fire}_i$ on the membrane potential.

3. Neural network topology of RBF with spiking neurons

We demonstrate how the system of leaky integrate-and-fire neurons can be used to perform a clustering task, relying on the fact that the neuron fires if its excitatory input is strong enough to push the postsynaptic potential above its threshold (Gerstner 1995, Delorme et al. 1999, Guedalia et al. 1999). The $n$-dimensional input patterns $X^{(j)} = (x^{(j)}_1, x^{(j)}_2, \ldots, x^{(j)}_m)$ are encoded with input spiking neurons $n_1, n_2, \ldots, n_m$, which are connected to the next RBF layer of output spiking neurons $v_1, v_2, \ldots, v_m$ (figure 3). For the purposes of clustering, the temporal coding of data assumes that each input neuron emits exactly one spike when the pattern is presented to the neural network; for example, the spike from $n_i$ is delayed by $d_i$ ms. Thus, the three-dimensional input ($3, 7, 1$) will be coded by $n_1$ firing after $3$ ms, $n_2$ firing after $7$ ms, and $n_3$ firing after $1$ ms.

RBFs are symmetric around a centre. Therefore the centre of an RBF neuron $C_i$ is associated with $m$-dimensional vector $Z_i = \{z_{i1}, z_{i2}, \ldots, z_{im}\}$, where $z_{ij} = d_{ij} - \min\{d_{ij} | 1 \leq i \leq m\}$. This is achieved by having only one active (non-zero) synapse among the synapses coming from input neuron $n_i$. The active synapse is the one whose delay is equal to $z_{ij}$. Formally, $w^{(d)}_{ij} = w_{\max}$ when $d = z_{ij}$ and $w^{(d)}_{ij} = 0$ for all other $d$. The delays between spikes in an input pattern will be balanced out by the delays in the synapses causing the output $v_j$ to react simultaneously to all incoming spikes.

![Figure 3. Neural network topology of an RBF with spiking neurons.](image-url)
When comparing the reaction of $v_j$ with an input pattern which is at its centre and an input pattern which is slightly off its centre, we can see that the first pattern causes a higher peak in the membrane potential of $v_j$ and that the membrane potential crosses the threshold. When a pattern is far from the centre, the peak will be lower than the threshold and no spike will be generated. The neuron realizes an RBF which is symmetric about the centre defined above and whose input is an $m$-dimensional pattern and whose output is the time until it emits a spike.

Figure 4 shows the membrane potential resulting from four 15-dimensional input patterns. The solid curve is the response to a pattern directly in the centre of the RBF, the dotted line is the response to an input which is slightly off centre, and the broken and chain curves are responses to patterns which are farther away. If the threshold is set to 10, the fastest response occurs with the centred pattern (solid curve); the response which fails to reach the threshold (therefore the neuron does not fire) occurs with the pattern furthest from the centre (broken line). Furthermore, we can see that the response to the centred pattern has the highest peak of the responses which cross the threshold.

We now briefly discuss the RBF SNN topology based on multiple synapses and constant delays as developed by Mattia and Del Giudice (2000). It is based on the following assumptions: each input neuron $n_i$ has one synaptic connection to each output RBF neuron $v_j$ with weight $w_{ij}$ and delay $d_{ij}$, and each delay $d_{ij}$ can be represented by the time difference between presynaptic firing and the activation of the synapse between the two neurons. Therefore there is a set of $k$ independent synapses with weights $W = \{w_{ij}^{(1)}, w_{ij}^{(2)}, \ldots, w_{ij}^{(k)}\}$ and delays $d_{ij}$ of to 1 ms, 2 ms, 3 ms, etc., respectively, which correspond to each pair comprising an input neuron $n_i$ and

![Figure 4. Membrane potential as a result of four 15-dimensional objects.](image-url)
an output neuron \(v_j\). The membrane potential \(u_i(t)\) for this case of multiple synapses and constant delays is given by

\[
u_i(t) = \sum_{j \in G_i} \sum_{i_j \in F_j} w_{ij}^{(d)} \varepsilon_{ij}(t - (t_j^{(q)} + d)). \tag{5}
\]

The inhibition of synapses between all of the RBF neurons (activated when they fire) implements a winner-take-all mechanism which allows only one RBF neuron to respond to each pattern (Mattia and Del Giudice 2000).

The Hopfield approach (Hopfield 1999) was extended by Natschläger and Ruf (1998) who proposed unsupervised clustering of spiking neural networks using spike times as input. They used a simple coding scheme where each input neuron \(n_i\) fires exactly once at time \(t_i\) within a given coding time interval \([0, T]\). The firing times represent the input vector \(X^{(j)} = (x_1^{(j)}, x_2^{(j)}, \ldots, x_m^{(j)})\) with \(x_i = \max\{t_i|1 \leq i \leq m\} - t_i\).

If, for some input vector \(X^{(j)}\), the difference \(\|X^{(j)} - Z_j\|\) is small enough to make \(v_j\) fire, \(X^{(j)}\) is close to the centre \(Z_j\) of an RBF neuron. Therefore the RBF neuron whose centre is closest to \(X^{(j)}\) will fire first. Thus a set of such RBF neurons can be used to separate inputs into various clusters (Hopfield 1995, Natschläger and Ruf 1998).

In our model, the RBF neurons have more than one active synapse from each input neuron, where \(Z_i = \{z_1, z_2, \ldots, z_m\}\) is defined by the synaptic weights of the neuron and has adaptive delays. In addition, the algorithm includes the establishment of multiple delays between the input and the output neuron layers. Thus each coordinate of \(Z_i\) is a weighted average of the delays from the matching input coordinates. The centre of such a neuron is (as in the case of one active synapse) a mirror image of \(Z_i\) and the response time to input patterns, and is symmetric around that centre.

4. Learning algorithm for clustering with RBF using spiking neurons

The proposed learning algorithm uses spike-timing-dependent synaptic plasticity (STDP) to adapt the synaptic weights and reward inputs which correlate well with the neuron’s output timing. STDP is a form of competitive Hebbian learning based on the exact spike timing information. Hebbian plasticity is a type of unsupervised learning which is useful for clustering input data but less appropriate for designing supervised pattern classifiers (Song et al. 2000).

Let us assume that we have a cluster \(K\) of input vectors \(X\) in the input space. The proposed training algorithm will cause neuron \(v_j\) to activate one delay for each input coordinate. As a result, we receive the cluster centre \(Z_j\) which minimizes \(\sum_{X \in K} \|X - Z_j\|\). In order to achieve this, the weights of the spiking neurons could be shifted during learning so that they recognize RBFs whose centres are the cluster centre.

The learning rule is applied to the synaptic weights of the neuron that fires when the input is presented to the network. The weights are changed in such a way that the neuron’s centre moves closer to the input pattern following the rule that synapses
which contributed to the neuron’s firing are strengthened and synapses which did not contribute are weakened. We distinguish two types of synapses:

- those that started affecting the postsynaptic neuron slightly before the neuron actually crossed the threshold contribute to the neuron’s firing
- those that affected the membrane potential much earlier or much later do not contribute to the neuron’s firing.

All RBF neurons should converge to the centre of some cluster. This can be done by the following procedure. Initially all weights are set to random values so that no RBF neuron can fire until it has received at least one spike from every neuron $n_i, 1 \leq i \leq m$. The synaptic plasticity window (figure 5) used in the learning algorithm is different for synapses with different strengths. Let us denote the window length by $t_{winL}$. Once an RBF neuron fires, the spike propagates backwards to its synapses where the weight change is given by

$$
\Delta w_{ij}^{(k)} = \begin{cases} 
A \Delta t_{ij} \exp(-\Delta t_{ij}), & \text{if } \Delta t_{ij} > 0; \\
-AB \Delta t_{ij} \exp(-\Delta t_{ij}), & \text{if } \Delta t_{ij} < 0,
\end{cases}
$$

where $\Delta w_{ij}^{(k)}$ is the synapse weight (strength) between input neuron $i$ and output neuron $j$; $\Delta t_{ij}$ is the normalized relative timing between the pre- and postsynaptic spikes, $A$, $B$, and $t_{winL}$ are parameters, $t_{preSyn}^i = t_i + d^{(k)}$ and $t_{postSyn}^j = t_j + d_{back}$ where $t_i$ is the time since neuron $i$ fired, $t_j$ is the time since neuron $j$ fired, $d^{(k)}$ is the synaptic delay, and $d_{back}$ shows the time used by the postsynaptic spike to propagate backwards.

![Figure 5. The learning window $A = e; B = 0.6; t_{winL} = 100 ms$.](image-url)
to the synapses. We use a set of delays $D = \{d^{(1)}, d^{(2)}, d^{(3)}, \ldots \}$ which each input neuron $i$ can set to each output neuron $j$ during training. The numerator on the right-hand side of equation (7) determines the time that elapses between the occurrence of a spike in neuron $j$ and the occurrence of a spike in neuron $j$.

The weight changes in equation (6) can be presented as follows:

$$w_{ij}^{\text{new}} = \begin{cases} w_{ij}^{\text{old}} + \eta \Delta w_{ij}(1 - w_{ij}^{\text{old}}) & \text{if } \Delta w_{ij} > 0; \\ w_{ij}^{\text{old}} + \eta \Delta w_{ij} w_{ij}^{\text{old}} & \text{if } \Delta w_{ij} < 0. \end{cases}$$

(8)

where $\eta$ is a learning rate parameter.

Equations (6) and (8) show that if the presynaptic spike precedes the postsynaptic spike, the weights will be increased. Otherwise, they will be reduced.

5. Simulations with artificial data sets

In order to study how our RBF SNN behaves in a high-dimensional space we perform six sets of simulations based on artificial data for clustering. These simulations are organized with the following common parameters: the length of the coding interval is set to 10 ms and the delay interval is 15 ms such that the available delays are 1 ms, 2 ms, $\ldots$, 16 ms. Every 50 ms a new learning cycle is started by presenting a new vector $X$ to the neural network; there is no overlap between the clusters.

In the first set of experiments (figure 6), the number of RBFs is equal to the number of clusters. After 30 learning cycles for each cluster, each RBF spiking
neuron converges to one cluster. We also observe that as the absolute refractory period increases, the postsynaptic response for each spike decreases on average. Further, increased continuous input to the presynaptic neuron is required to trigger a synaptic discharge (a spike). Also, after an RBF neuron converges to a cluster, it starts to fire significantly earlier when a sample from this cluster is presented.

In the next four sets of experiments (figures 7 and 8), we establish the network behaviour when the number of clusters is different from the number of RBF neurons.

Figure 7. The number of clusters is less than the number of RBFs.
In the case of more clusters than RBF neurons, some neurons converge to the mean of a subset of clusters (figure 7). In the case of fewer clusters than RBF neurons (figure 8), RBF neurons converge on the same cluster while others do not take part in the learning process and so do not converge on any cluster. We observed that over a 1s interval of generating spikes, the postsynaptic response is highest for the first few spikes, but then dramatically decreases for the remaining spikes.

Figure 8. The number of clusters exceeds those of RBFs.
After the first few initial spikes in each set of experiments, the electric potential becomes larger for the firing of the other spikes and the amplitude of the postsynaptic response decreases as the absolute refractory period for each set is increased.

6. Simulations with iris data

We implemented our RBF neural network based on spiking neurons to the iris dataset consisting of 150 data points belonging to three groups of the same size. There are three varieties of iris: *Iris setosa*, *Iris versicolor*, and *Iris virginica* (Michie et al. 1994). The original problem was to classify a new iris flower into one of these three types based on four attributes (petal and sepal length and width). The input groups for our experiments are three species of iris where each of the data points is described by four features: petal-length, petal-width, sepal-length, and sepal-width. By using the proposed RBF neural network based on spiking neurons we were able to classify 89% of the data points correctly, which is comparable with other powerful algorithms. The results of our clustering algorithm based on RBF with spiking neurons are shown in figure 9. Each peak corresponds to the activation of one of the three clusters defined by the three groups of input neurons.

The performance of the spiking neuron algorithm for clustering presented here was compared with those of other clustering algorithms. Blatt et al. (1997), who developed an algorithm based on an analogy of data points with magnetic spins, reported a 17% error in the solution of the iris data clustering problem. The success rate of expectation maximization principal components analysis (EMPCA) is 86.10% (Rosipal and Girolami 2000), and that of the neural network for clustering TEST is 82.10% (Soo and Song 1999). The best results reported so far were obtained using the subtractive clustering method FCM (93.54%) (Yager and Filev 1994), in comparison with the fuzzy C-means clustering method (FCM) (82.10%), and maximum likelihood principal components analysis (MLPCA) (83.40%) (Wentzell et al. 1997). However, FCM is computationally expensive, and the amount of computation required increases rapidly with the increasing dimensionality of the data.

Figure 9. The total spiking activity for Iris data set.
7. Conclusions

We have presented a novel model of a spiking neuron network, which is based on dynamics of the synapse and implements an efficient delay mechanism. We have demonstrated that spiking neural networks encoding information in the timing of single spikes are capable of computing and learning clusters from large datasets. The performance of the model was tested and a comparative analysis was performed. The results demonstrate the efficiency of the learning algorithm.

The data clustering based on the neural network algorithm with RBF and neurons of the leaky integrate-and-fire type showed the following.

- The RBF neurons converged very reliably to the centres of the clusters even in the presence of noise.
- If clusters were added or removed during the learning process, the RBF neurons were able to reconfigure dynamically which is of particular importance for biological systems.
- As the absolute refractory period increased, more input potential was needed for the next synaptic discharge to occur.

Acknowledgements

This work was partially supported by a CREW/NSF grant, a CSI/City University of New York President Award Grant, and a University of Wisconsin Oshkosh Faculty Development Grant.

References


