ESANN'2003 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 23-25 April 2003, d-side publi., ISBN 2-930307-03-X, pp. 27-32

Unsupervised Recursive Sequence Processing

Marc Strickert, Barbara Hammer
Dept. of Math./Comp. Science, University of Osnabrück, Germany
e-mail: {marc,hammer}@informatik.uni-osnabrueck.de

Abstract. We propose a self organizing map (SOM) for sequences by extending standard SOM by two features, the recursive update of Sperduti [7] and the hyperbolic neighborhood of Ritter [5]. While the former integrates the currently presented item and recent map activations, the latter allows representation of temporally possibly exponentially growing sequence diversification. Discrete and real-valued sequences can be processed efficiently with this method as demonstrated in three experiments.

1 Introduction

Unsupervised clustering by using the Kohonen SOM [4] is a standard tool for the exploration of high dimensional data and its topological structure: typically, a two dimensional Euclidean grid of neurons is adapted to the data space, more or less preserving the data topology. If temporal or spatial data are dealt with such as time series, language data, or DNA strings, sequences of a priori unrestricted length constitute a natural domain for data analysis and classification. Unfortunately, the temporal scope is unknown in most cases, and therefore fixed vector dimensions, as used for standard SOM, cannot be applied. Several extensions of SOM to sequences have been proposed: time-window techniques or representation by statistical features allow processing with standard methods [4]. Thereby, information might get lost and adaptation of the metric or grid is advisable [5, 6, 8]. Various approaches extend SOM by recurrent dynamics: the temporal Kohonen map (TKM) and the recurrent SOM (RSOM) incorporate the biologically plausible dynamics of leaky integrators [1, 9]. The recursive SOM (recSOM) and the SOM for structured data (SOMSD) are based on a richer representation of the respective time context, the activation profile of the entire map or the index of the most recent winner, respectively [2, 10]. A general framework for these dynamics has been proposed in [3].

We will here focus on the compact and flexible representation of time context given by the winning location of the map for the previously presented sequence element as proposed in [2]. Since this approach heavily relies on an adequate grid topology, we extend this approach to general, possibly non-Euclidean, triangular grid structures. In particular, we combine a hyperbolic grid and a last-winner-in-grid reference to process three types of sequences: the Mackey-Glass series, binary automata, and a benchmark Reber grammar.

2 Unsupervised processing of sequences

Given a sequence $s = (s_1, \ldots, s_t)$ with entries in $\mathbb{R}^n$, the TKM computes the distance of s from neuron n_j labeled with $w_j \in \mathbb{R}^n$ as leaky integration

ESANN'2003 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 23-25 April 2003, d-side publi., ISBN 2-930307-03-X, pp. 27-32

$d_{\mathrm{TKM}}(s, n_j) = \sum_{i=0}^{t-1} \eta(1-\eta)^i \|s_{t-i} - w_j\|^2$ where $\eta \in (0,1)$ [1]. The RSOM first integrates the directions and afterwards computes the distance [9]. In both cases, one can analytically compute weights with optimum response to a given sequence s: $w = \sum_{i=0}^{t-1}(1-\eta)^i s_{t-i} / \sum_{i=0}^{t-1}(1-\eta)^i$. This explains the encoding scheme of TKM and RSOM. For example, in case of binary sequences with $\eta \leq 0.5$, the representation of fixed length sequences corresponds to a fractal encoding in $[0,1]$, e.g. sequences of length three are represented by 8 consecutive points encoding 000, 001, 010, ..., 111. This encoding, although reasonable, points out the limitation of TKM and RSOM: representations for sequences of different length are mixed; e.g. 0 gives optimum response to sequences with entries only 0 and arbitrary length. Furthermore, the maximum length of sequences which can be represented does not depend on the size of the grid, but on the range of w, i.e. the range of sequence entries.

The RecSOM uses a more detailed representation of context [10]: each neuron n_j has got a weight $w_j \in \mathbb{R}^n$ representing the recent sequence entry and a vector $c_j \in \mathbb{R}^N$, N denoting the number of neurons, which represents the contextual map activation of all neurons in the previous time step. Distance d is recursively computed by $d_{\mathrm{RecSOM}}((s_1, \ldots, s_t), n_j) = \eta_1 \|s_t - w_j\|^2 + \eta_2 \|C_{\mathrm{RecSOM}}(s_1, \ldots, s_{t-1}) - c_j\|^2$ where $\eta_1, \eta_2 > 0$. Thereby, $C_{\mathrm{RecSOM}}(s) = (\exp(-d_{\mathrm{RecSOM}}(s, n_i)))_{i=1}^N$ constitutes the context. Since the entire map activation is considered, sequences of any fixed length can be stored by a sufficient number N of neurons; the associated context dimensionality of N, however, makes this approach computationally quite costly. The SOMSD method represents temporal context by the winner index in the previous time step, i.e. in addition to a weight $w_j \in \mathbb{R}^n$ neuron n_j uses a value $c_j \in \mathbb{R}^l$, l denoting the dimensionality of the Euclidean lattice [2]. Distance d is computed by $d_{\mathrm{SOMSD}}((s_1, \ldots, s_t), n_j) = \eta_1 \|s_t - w_j\|^2 + \eta_2 \|C_{\mathrm{SOMSD}}(s_1, \ldots, s_{t-1}) - c_j\|^2$ where $C_{\mathrm{SOMSD}}(s)$ equals the index of neuron n_j with smallest $d_{\mathrm{SOMSD}}(s, n_j)$. Due to the compressed context information, this approach is very efficient by exploiting the neighborhood structure of the neurons with its topological ordering property. Since for sequential data, like for words over an alphabet $\{a_1, \ldots, a_M\}$, the neighborhood size is an exponential function of the word length, an Euclidean target grid with inherent power law neighborhood growth is not suited for topology preserving representation. We therefore extend this approach to arbitrary triangular grid structures. Using a grid with > 6 neighbors per neuron, the graph becomes part of the 2-dimensional hyperbolic plane, where exponential neighborhood growth is possible. Hence, adequate data representation can be expected, as demonstrated in [5] for a non-recursive text processing SOM.

3 SOM for sequences (SOM-S)

Standard SOMs operate on a rectangular neuron grid embedded in a real-vector space. More flexibility for the topological setup can be obtained by describing the grid in terms of a graph: neural connections are realized by assigning each neuron a set of direct neighbors. We have implemented a grid generator with circular triangle meshing around a center neuron and given neighborhood degree n. Figure 1 shows a small map with 7 neighbors for the inner neurons. For ≥ 7 neighbors, an exponential neighborhood increase can be observed and

ESANN'2003 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 23-25 April 2003, d-side publi., ISBN 2-930307-03-X, pp. 27-32

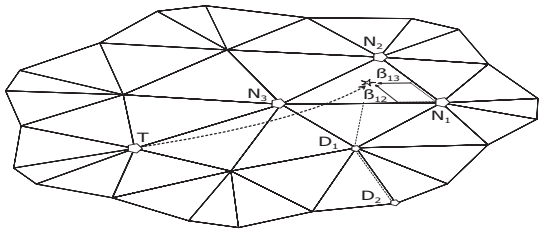

Figure 1: Hyperbolic self organizing map with context.

a faultless embedding into the Euclidean plane is not possible, but still local projections in terms of a fish eye magnification focus can be obtained.

The recursive nature of the shown map is explained exemplarily for neuron T in Fig. 1. T is equipped with a weight $w \in \mathbb{R}^n$ and a context c given by a location within a triangle of neurons N_1, N_2, N_3, corner expressed affinities by means of adjustable linear combination parameters $\mathsf{ß}_{12}$ and $\mathsf{ß}_{13}$. Distance d of a sequence from T is recursively computed by $d_{\text{SOM-S}}((s_1, \ldots, s_t), \mathsf{T}) = \eta \|s_t - w\|^2 + (1-\eta) g(C_{\text{SOM-S}}(s_1, \ldots, s_{t-1}), c)$. $C_{\text{SOM-S}}(s)$ is the index of the neuron n_j in the grid with smallest $d_{\text{SOM-S}}(s, n_j)$. g denotes the grid distance of the winner from the triangular position $c_j = (\mathsf{N}_1, \mathsf{N}_2, \mathsf{N}_3, \mathsf{ß}_{12}, \mathsf{ß}_{13})$ given by the shortest possible path in the mesh structure. Thereby, grid distances between neighboring neurons possess unit length, and within the triangle $\mathsf{N}_1, \mathsf{N}_2, \mathsf{N}_3$ the metric structure is approximated by the Euclidean distance. The range of g is implicitly normalized by dividing by the maximum grid distance.

Training is carried out by presenting a pattern $s = (s_1, \ldots, s_t)$, the determination of the winner k, and the update of weight and context in Hebbian style applied to all neurons in the breadth first search graph around the winning neuron, according to their grid distances. Hence, weight w_j is updated for sequence entry s_i by $\Delta w_j = \gamma h_{jk}(s_i - w_j)$. The learning rate γ is typically exponentially decreased from 0.1 to 0.005 during training; h_{jk} describes the influence of the winner k to the current neuron j as a decreasing function of the grid distance. Context update is analogous: the current context, expressed in terms of neuron triangle corners and coordinates, is moved by a γ-fraction of the shortest path's distance along such path towards the previous winner location, adapting $\mathsf{ß}_{12/13}$, and exchanging $\mathsf{N}_{1/2/3}$ if necessary. In case of planar circular grids with two possible ways of equal length the midway is taken. This explains why the update path for the current context towards D_2 is the dotted line via D_1 in Fig. 1. Grid distances and edges are initially stored in a matrix, thus providing fast shortest path calculations. The parameter η controls the balance between pattern and context influence; since at the beginning nothing is known about the temporal structure, this parameter starts at 1, resulting in the standard non-contextual SOM. It is decreased during training to an application dependent value that mediates between the externally presented pattern and the internally gained model about historic contexts.

4 Experiments

Mackey-Glass Series: The first learning task is the dynamics of the real-valued chaotic Mackey-Glass time series $\frac{\mathrm{d}x}{\mathrm{d}\tau} = bx(\tau) + \frac{ax(\tau-d)}{1+x(\tau-d)^{10}}$ using

ESANN'2003 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 23-25 April 2003, d-side publi., ISBN 2-930307-03-X, pp. 27-32

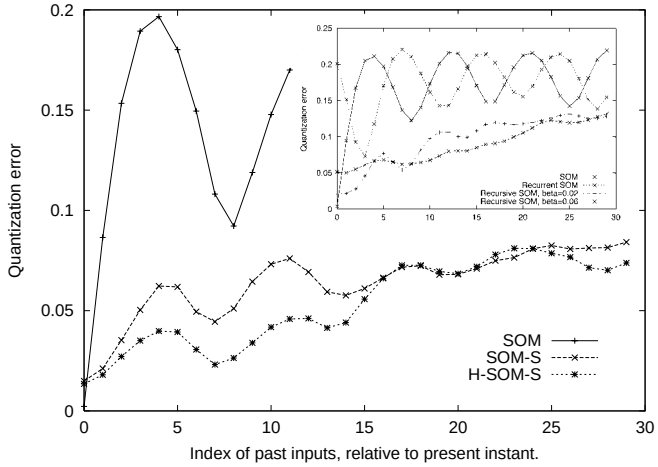

Figure 2: Temporal quantization errors of different model setups for the Mackey-Glass series. The inset in the upper right shows the results from [10].

$a = 0.2$, $b = -0.1, d = 17$. The same setup is given in [10] and allows a comparison of results.[1] Three types of maps with 100 neurons have been trained: a 6-neighbor map without context constituting circular standard SOM, a map with 6 neighbors and context consideration (SOM-S), and a 7-neighbor map with context utilization providing hyperbolic grid (H-SOM-S). Each run has been computed multiple with $1.5 * 10^5$ presentations starting at random positions within the Mackey-Glass series and step size $\Delta t = 3$; the neuron weights have been initialized white within [0.6, 1.4]. η has been decreased from 1 to 0.97. Figure 2 shows the quantization error for the above setups expressed by the average standard deviation of the given sequence and the mean unit receptive field for 29 time steps into the past. Similar to Voegtlin's results shown in the inset, we observe large cyclic oscillations driven by the periodicity of the training series for standard SOM. SOM-S without and with hyperbolic grid (H-SOM-S) yield better and less oscillating results. These are comparable or even better than the results of RecSOM while requiring less computing power.

Binary Automata: The second experiment is also inspired by Voegtlin to learn a $\mathsf{0/1}$-sequence generated by a binary automaton with $P(\mathsf{0}|\mathsf{1}) = 0.4$ and $P(\mathsf{1}|\mathsf{0}) = 0.3$. The specialization of a neuron is given as the longest sequence that leads to unambiguous winner selection. Fig. 3 shows the specialization of a trained H-SOM-S and its good correspondence to the 100 most characteristic sequences of the probabilistic automaton. Training has been performed with $3 * 10^6$ presentations, increasing the context influence η exponentially from 0% to 6%. Putting more emphasis on the context results in a smaller number of active neurons with longer specialization strings that cover small parts of the input space only. In addition to the examination of neuron specialization, the whole map representation can be characterized by comparing the input symbol transition statistics with the learned context-neuron relations. While the

[1] We would like to thank T.Voegtlin for providing data for comparison.

ESANN'2003 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 23-25 April 2003, d-side publi., ISBN 2-930307-03-X, pp. 27-32

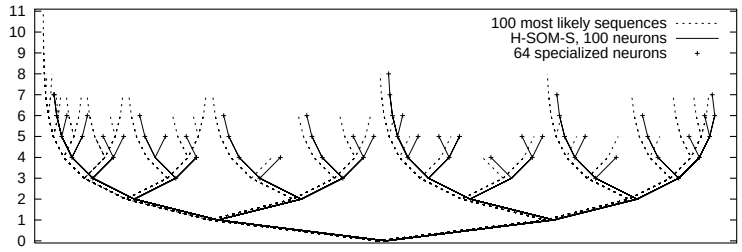

Figure 3: Receptive fields of a H-SOM-S compared to the most probable subsequences of the binary automaton. Left hand branches denote 0, right is 1.

current symbol is coded by the winning neuron weight, the previous symbol is represented by the linear combination of the winner's context triangle neurons' weights. The obtained pairs are clearly expressed in the trained map and only few neurons contain values in an indeterminate interval $[\frac{1}{3}, \frac{2}{3}]$. Results for the reconstruction of three automata can be found in table 1. The left column indicates the number of expressed neurons and the total number of neurons in the map. Note that the automata can be well re-obtained from the maps.

Reber Grammar: In a third experiment we have used more structured symbolic sequences as generated by the Reber grammar depicted in Fig. 4. The 7 symbols have been encoded in a 6-dimensional Euclidean space. For training and testing we used concatenations of randomly generated words until sequences of $3 * 10^6$ and 10^6 input vectors, respectively, were available. The map contains 617 neurons on an hyperbolic grid. A number of 338 neurons developed a specialization for Reber strings with an average length of 7.23 characters. Specializations separate into strict sectors on the circular grid, ordered in a topological way by the last character. In agreement with the grammar, the letter T takes the largest sector on the map. This ordering preference could not be found with Euclidean grids, for which polymorphic patches emerged. Similar to the binary automata learning tasks, we analyzed the map representation by the data reconstruction, backtracking all possible context sequences of each neuron up to length 3. Only 118 of all 343 possible trigrams are realized. In a ranked table the most likely strings cover all 33 Reber trigrams, and there is a leap in the probability plot from entry number 33 (TSS, valid) to 34 (XSX, invalid) emphasizing the learned Reber characteristic. The correlation of Reber trigram probabilities and the relative frequencies found in the map is 0.75.

Type	$P(0)$	$P(1)$	$P(0\|0)$	$P(1\|0)$	$P(0\|1)$	$P(1\|1)$
Automaton 1	$4/7 \approx 0.571$	$3/7 \approx 0.429$	0.7	0.3	0.4	0.6
Map (98/100)	0.571	0.429	0.732	0.268	0.366	0.634
Automaton 2	$2/7 \approx 0.286$	$5/7 \approx 0.714$	0.8	0.2	0.08	0.92
Map (138/141)	0.297	0.703	0.75	0.25	0.12	0.88
Automaton 3	0.5	0.5	0.5	0.5	0.5	0.5
Map (138/141)	0.507	0.493	0.508	0.492	0.529	0.471

Table 1: Results for binary automata extraction.

ESANN'2003 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 23-25 April 2003, d-side publi., ISBN 2-930307-03-X, pp. 27-32

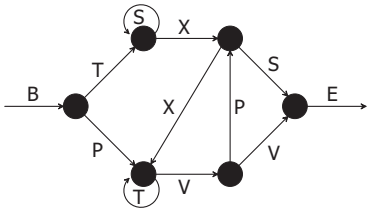

Figure 4: State graph of the Reber grammar.

5 Conclusion

We have presented a self organizing map with neural back-reference to its recently active sites and flexible topological structure of the neuron grid. This context model can be interpreted as the development of long range synaptic connections, leading to more specialized map regions. By analyzing the trained net, regularities within the input sequences could be reconstructed from the learned neuron-context relations, as shown in different experiments. Our model can be generalized to more sophisticated grid structures than the presented circular triangular graphs with a fixed number of neighbors for the inner neurons: only the context must be expressible in terms of affiliation to k (here 3) context neurons, and for adaptation, the processability of the shortest path within the graph is required. Graph traversals around the winner neurons can be reduced to local regions by defining a radius dependent threshold below which further updates can be neglected. Thus, together with the recursive properties of the training dynamic, 3-dimensional structures with exponential branching could be realized as approximation to simplistic brain models.

References

[1] G. Chappell and J. Taylor. The temporal Kohonen map. *Neural Networks*, 6:441–445, 1993.

[2] M. Hagenbuchner, A. C. Tsoi, and A. Sperduti. A supervised self-organising map for structured data. In N. Allinson, H. Yin, L. Allinson, and J. Slack, editors, *Advances in Self-Organising Maps*, 21–28. Springer, 2001.

[3] B. Hammer, A. Micheli, and A. Sperduti. A general framework for unsupervised processing of structured data. In M. Verleysen, editor, *European Symposium on Artificial Neural Networks'2002*, 389–394. D Facto, 2002.

[4] T. Kohonen. *Self-Organizing Maps.* Springer-Verlag, Berlin, 2001.

[5] J. Ontrup and H. Ritter. Text categorization and semantic browsing with self-organizing maps on non-euclidean spaces. In L. D. Raedt and A. Siebes, editors, *Proceedings of PKDD-01*, 338–349. Springer, 2001.

[6] J. Sinkkonen and S. Kaski. Clustering based on conditional distributions in an auxiliary space. *Neural Computation*, 14:217–239, 2002.

[7] A. Sperduti. Neural networks for adaptive processing of structured data. In *Proc. ICANN 2001*, 5–12. Springer, 2001.

[8] M. Strickert, T. Bojer, and B. Hammer. Generalized relevance LVQ for time series. In *Proc. ICANN'2001*, 677–638. Springer, 2001.

[9] M. Varsta, J. del R. Milan, and J. Heikkonen. A recurrent self-organizing map for temporal sequence processing. In *Proc. ICANN'97*, 421–426. Springer, 1997.

[10] T. Voegtlin. Recursive self-organizing maps. *Neur.Netw.*, 15(8-9):979–991, 2002.