ESANN'2004 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 28-30 April 2004, d-side publi., ISBN 2-930307-04-8, pp. 39-44

Self-Organizing Context Learning

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

Abstract. This work is designed to contribute to a deeper understanding of the recently proposed Merging SOM (MSOM). Its context model aims at the representation of sequences, an important subclass of structured data [7]. In this work, we revisit the model with a focus on its fractal context encoding and the convergence of its recursive dynamic. Experiments with artificial and real world data support our findings and demonstrate the power of the MSOM model.

1 Introduction

Recursive data processing is a challenging task for dealing with graph structured data, or sequences as a special case. The natural domain of sequential data is given by series of temporally or spatially connected observations – DNA chains and articulatory time series are two examples. Usually, a vector representation exists or can be found for individual sequence entries. Kohonen's self-organizing map is a well known method for projecting high dimensional data to a low dimensional grid, enabling analysis and visualization [3]. The neural gas (NG) algorithm yields data representations based on a small number of prototypes in the data space providing a minimum quantization error [4]. However, temporal or spatial contexts within a series are usually taken into consideration in terms of data windows. These windows are constructed as a serialized concatenation of a fixed number of vectors from the input stream causing problems of loss of information, curse of dimensionality, and usually inappropriate metrics. The latter can be partially accounted for by adaptive metrics [5, 6].

Recently, increasing interest in unsupervised recurrent selforganizing networks can be observed, that directly deal with sequential data. Prominent methods are the temporal Kohonen map (TKM), the recurrent self-organizing map (RSOM), recursive SOM (RecSOM), and SOM for structured data (SOMSD) [1, 2, 10, 11]. Comparisons of these methods with respect to accuracy and efficiency have already been presented in [7, 8]. However, little is known about which formalism underlies their storage of temporal information. Especially for unsupervised approaches, a thorough understanding of the emerging representation is needed for their valid interpretation. The focus of this contribution is a theoretical and experimental investigation of a very recent, efficient, and promising approach, the Merge SOM (MSOM) [7]; the temporal context of which combines the currently presented pattern with the sequence past in an intuitive way. We will show that the MSOM model learns a fractal encoding of recursive data, and thus MSOM follows up a successful technique, which is well-established in supervised learning tasks [9].

ESANN'2004 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 28-30 April 2004, d-side publi., ISBN 2-930307-04-8, pp. 39-44

2 The Merge Context

Temporal processing requires a directional context for taking historic influence on the present sequence state into account. Three basic types of context models can be found in the literature for self-organized learning. The context given by TKM and RSOM is implicitly expressed in the data space by summing up the exponentially weighted historic errors of a neuron and the presented sequence element [1, 10]. The RecSOM context has an explicit back reference to all the neuron activations in the previous step [11]; these are stored as a vector in each neuron in addition to their pattern representing weight vector. Since this type of context is computationally very demanding for large networks and is also subject to random activations, a compact version of this refers to the previously winning neuron only, as realized by SOMSD [2]. The back reference is implemented as a pointer to the location of the winner in a regular, usually two dimensional SOM grid. Still, the explicit SOMSD context contains a major drawback: the dependence on a regular neuron indexing scheme. Therefore the following Merging SOM (MSOM) context model has been developed.

In general, the merge context refers to a fusion of two properties characterizing the previous winner: the weight and the context of the last winner neuron are merged by a weighted linear combination. During MSOM training, this context descriptor is calculated online and it is the target for the context vector c_i of a neuron i. Target means that the vector tuple $(w_i, c_i) \in \mathbb{R}^{2n}$ of neuron i is adapted into the direction of the current pattern and context according to Hebb learning.

Definition of the Merge Context
The winner is the best matching neuron j, for which the recursively computed distance

$$\tilde{d}_j(a_t) = (1-\alpha) \cdot |a_t - w_j|^2 + \alpha \cdot |C_t - c_j|^2$$

to the current sequence entry a_t and the context descriptor C_t is minimum. Both contributions are balanced by the parameter α. The context descriptor

$$C_t = (1-\beta) \cdot w_{I(t-1)} + \beta \cdot c_{I(t-1)}$$

is the linear combination of the properties of winner $I(t-1)$ in the last time step. A typical merging value for $\beta \geq 0$ is 0.5.

The Merge Context for known Architectures
An integration of the merge context into self-organizing networks like the neural gas model [4], Kohonen's self-organizing map or the learning vector quantization model [3] is easily possible. Here, we will focus on a combination of the context model with neural gas that we will call merging neural gas (MNG).

After presentation of sequence element a_t, for each neuron j its rank $k = \text{rnk}(j)$ is computed, providing the information that k neurons are closer to a_t than neuron j is. The update amount is an exponential function of the rank:

$$\begin{aligned} \Delta w_j &= \eta_1 \cdot \exp(-\text{rnk}(j)/\sigma) \cdot (a_t - w_j) \\ \Delta c_j &= \eta_2 \cdot \exp(-\text{rnk}(j)/\sigma) \cdot (C_t - c_j) \end{aligned}$$

The context descriptor C_t has to be updated to date during training by keeping track of the respective last winner. In experiments, the learning rates where set to identical

ESANN'2004 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 28-30 April 2004, d-side publi., ISBN 2-930307-04-8, pp. 39-44

values $\eta_1 = \eta_2 = \eta$. The neighborhood influence σ decreases exponentially during training to obtain neuron specialization. The initial contribution of the context term to the distance computation and thus to the ranking order is chosen low by setting the weight/context balance parameter α to a small positive value. Since the weight representations become more reliable during training, it is worth to gradually also pay more attention to the context that refers to them. In the following, after an initial weight specialization, α has been successively steered to a final value that maximizes the neuron activation entropy; in other words, the highest possible number of neurons shall be on average identically active at the end of training. Thinking in terms of hierarchical neural activation cascades this heuristic is not optimal, when a small number of often visited root states are branching out to states of decreasing probability. However, the α-control strategy proved to be very suitable in our experiments.

Properties of the Merge Context

In order to shed light on the convergence properties of the merge context, we determine the optimum encoding given by context and weight vectors. The best adaptation of neuron j is the one for which w_j and c_j yield $\tilde{d}_j(a_t) = 0$:

$$\tilde{d}_j(a_t) = (1-\alpha) \cdot |a_t - w_j|^2 + \alpha \cdot |(1-\beta) \cdot w_{I(t-1)} + \beta \cdot c_{I(t-1)} - c_j|^2$$

Both squared summands can be considered separately. The left one trivially becomes the minimum 0 for $w_{opt(t)} = w_j = a_t$. Then, the right one expands to

$$\begin{aligned} c_{opt(t)} &= (1-\beta) \cdot a_{t-1} + \beta \cdot c_{I(t-1)} \\ &= (1-\beta) \cdot a_{t-1} + \beta \cdot ((1-\beta) \cdot a_{t-2} + \ldots + \beta \cdot ((1-\beta) \cdot a_1 + 0)) \\ &= \sum_{j=1}^{t-1} (1-\beta) \cdot \beta^{j-1} a_{t-j} \end{aligned}$$

by induction with the zero context assumption that is associated with the beginning at a_1. Now, focusing on the convergence of a neuron that is specialized on a particular sequence element within its unique context, we obtain asymptotically stable fixed points of the training update dynamic. The analysis of iterative weight updates compared to the target vector yields:

$$|w_{I(t)} + \eta \cdot (a_t - w_{I(t)}) - a_t| = (1-\eta) \cdot |w_{I(t)} - a_t| \quad \Rightarrow \quad w_{I(t)} \to a_t$$

Which is an exponential convergence because of $\eta \in (0,1)$. Analogously,

$$|c_{I(t)} + \eta \cdot \big((1-\beta) \cdot w_{I(t-1)} + \beta \cdot c_{I(t-1)} - c_{I(t)}\big) - c_{opt(t)}| \;\Rightarrow\; c_{I(t)} \to c_{opt(t)}$$

describes the context convergence if we can show that

$$(1-\beta) \cdot w_{I(t-1)} + \beta \cdot c_{I(t-1)} \to c_{opt(t)} \,.$$

With $w_{I(t-1)} \to a_{t-1}$ and by induction of $c_{I(t-1)} \to c_{opt(t-1)}$ with $c_{I(1)} := 0$:

$$\begin{aligned} (1-\beta) \cdot a_{t-1} + \beta \cdot c_{opt(t-1)} &= (1-\beta) \cdot a_{t-1} + \beta \cdot \sum_{j=2}^{t-1} (1-\beta) \cdot \beta^{j-2} a_{t-j} \\ = \sum_{j=1}^{t-1} (1-\beta) \cdot \beta^{j-1} a_{t-j} &= c_{opt(t)} \end{aligned}$$

ESANN'2004 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 28-30 April 2004, d-side publi., ISBN 2-930307-04-8, pp. 39-44

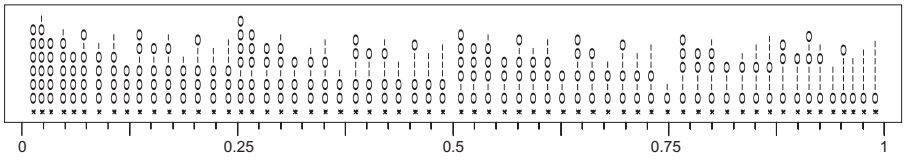

Figure 1: Context associated with the current symbol **0** of a binary sequence.

This sum for c_{opt} denotes a fractal encoding of the context vector in the weight space, which is known to be a very compact and efficient representation [9]. In experiments, we can observe the emergence of a Cantor set like non-overlapping fractal context for a merging parameter of $\beta = 0.5$. The spreading dynamic of the zero initialized context in the weight space is self-organizing with respect to the density of the contextual input. Since the context is a function of the previous winner's weight and context, the adaptation is a moving target problem; therefore, it is generally a good policy to have either a faster weight than context update, or to put more influence on the pattern matching than on the context matching by choosing $\alpha < 0.5$.

3 Experiments

Exemplary Context development
Figure 1 displays the experimental context space resulting from the MNG training of 128 neurons for a random binary sequence containing independently drawn symbols **0** and **1** with $P(\mathbf{0}) = P(\mathbf{1}) = 1/2$. The plot is reduced to the 63 non-idle neurons that represent the current symbol **0**, that can be found as the lower line of zeroes. The context fills the input space in $(0, 1)$ with an almost equidistant spacing after 10^6 symbol presentations. The stacking of symbol lines indicates the reference to the further past. Remarkably, the longest sequences which the neurons can still uniquely discriminate, are arranged, as stated in the theory section, in a Cantor like way in the context space.

Representation of the Reber grammar
This experiment refers to sequences generated by the Reber automaton depicted in figure 3. The 7 symbols have been encoded in a 6-dimensional Euclidean space. For training and testing we concatenated randomly generated Reber words and produced sequences of $3 * 10^6$ and 10^6 input vectors, respectively. The number of neurons is 617, the merge parameter is $\beta = 0.5$, the starting neighborhood size is $\sigma = 617$, and the context vector is initialized to $\mathbf{0} \in \mathbb{R}^6$, the center of gravity of the embedded symbols. The learnings rate is $\eta = 0.03$; after training, the adaptive parameters are $\sigma = 0.5$ and $\alpha = 0.43$. Finally, the context information stored within the ensemble of neurons has been analyzed. The average length of Reber strings from the test sequence leading to unambiguous winner selection is 8.90187, whereby a number of 428 neurons develop a distinct specialization on Reber words. The reference results of a hyperbolic SOMSD with 617 neurons are an average string length of 7.23, and number of 338 active neurons [8]. In addition to this test sequence driven statistics, a network internal backtracking has been

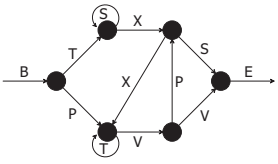

Figure 2: Reber graph.

ESANN'2004 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 28-30 April 2004, d-side publi., ISBN 2-930307-04-8, pp. 39-44

performed: on average 67.9 neurons referred to as the best matching contexts of each single neuron represent the same symbol before a different symbol is encountered; this is a very strong support for a high context consistency and a proper precedence learning. Therefore, a backtracking scheme has been used to collect the strings for each neuron, composed of symbols represented by the recursively visited best matching predecessors. The string assembly stops at the first revisit of a neuron: words with an average length of 13.78 are produced, most of them with valid Reber grammar. The longest word `TVPXTTVVEBTSXXTVPSEBPVPXTVVEBPVVEB` corresponds, as for most other neurons, perfectly to the training set driven neuron specialization `TVPXTTVVEBTSXXTVPSEBPVPXTVVE`.

Speaker identification by a posteriori MNG labeling
This experiment processes speaker data from the UCI repository[1]. Recordings from 9 speakers of the Japanese vowel 'ae' are given as sequences of 12 dimensional frequency based vectors. Each utterance comprises a number between 7 and 29 temporally connected vectors. In the training set 30 articulations are available for each speaker; in the test set there is a total of 370 utterances. Each articulation has its own temporal structure; since between different utterances there is no temporal connection, a neuron d is added to represent the no-context-available-state by $w_d = c_d$ in the data center, being the default previous winner for an utterance start. After unsupervised MNG training of 150 neurons without speaker identity, each neuron is assigned a 9 bin histogram containing activation frequencies for the speakers from the training set. For each articulation sequence in the test set, the accumulated majority vote over the bins of activated neurons is calculated to identify the speaker. The resulting histograms are very specific for the speakers. Applying the a posteriori labels, there is no error on the training set and an error of only 2.7% on the test set, which is much better than the reference error of 5.9% coming with the data set. Using 1000 neurons, the error decreases to only 1.6%.

4 Conclusions

We have investigated the merge context model for temporally or spatially connected sequential data. Context is obtained for self-organizing training architectures like SOM, NG, or LVQ by referring back to the winner compactly described by a linear combination of the represented weight and context. This recursive context definition leads to an efficient fractal encoding, which is the fixed point of the dynamics of self-organizing methods. Since the context emerges well during training, a preprocessing by data partitioning or fractal encoding is not necessary for methods implementing the merge context. As a consequence of self-organization, the capacity of the context representation grows with the number of neurons. A maximum utilization of neurons is forced by maximizing the network entropy by adjusting the context influence parameter α. Since recurrent neural networks cover the supervised learning tasks pretty well, the main applications of the merge context model should be found for unlabeled data. Nevertheless, an experiment of speaker recognition with a posteriori labeling indicates that we can trust in the found context representations and that there is much potential for processing labeled data, too.

[1] `http://kdd.ics.uci.edu/databases/JapaneseVowels/JapaneseVowels.html`

ESANN'2004 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 28-30 April 2004, d-side publi., ISBN 2-930307-04-8, pp. 39-44

The transfer of the merge context to the standard SOM is straight forward, only that the neuron neighborhood function is not defined by the ranking but by the neural grid neighborhood. It should be kept in mind that for regular grids the SOMSD approach might be more efficient. For supervised scenarios, preliminary results show that after unsupervised MNG training a fine tuning with a modified merge context LVQ is possible. Winner selection is done by taking the neuron j with smallest distance $\tilde{d}_j(a_t)$. In the case that the neuron is of the desired class, the update degenerates to neural gas without neighbors. In the other case the weight vector w_j is adapted an η-step into opposite direction of a_t, but the context c_j must still be adopted towards C_t, because the context is the same in both cases.

As future work, more general metrics than the squared Euclidean distance are considered and the interplay of the context influence and the context update strength is investigated in more detail. Another, even more challenging research is connected with the transfer of the presented sequential context to processing graph structures, which will be the next step of our work.

References

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

[2] M. Hagenbuchner, A. Sperduti, and A. Tsoi. A self-organizing map for adaptive processing of structured data. *IEEE Transactions on Neural Networks*, 14(3):491–505, 2003.

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

[4] T. Martinetz, S. Berkovich, and K. Schulten. "Neural-gas" network for vector quantization and its application to time-series prediction. *IEEE Transactions on Neural Networks*, 4(4):558–569, 1993.

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

[6] M. Strickert, T. Bojer, and B. Hammer. Generalized relevance LVQ for time series. In G. Dorffner, H. Bischof, and K. Hornik, editors, *Proceedings of the International Conference on Artificial Neural Networks (ICANN)*, pages 677–683. Springer, 2001.

[7] M. Strickert and B. Hammer. Neural Gas for Sequences. In T. Yamakawa, editor, *Proceedings of the Workshop on Self-Organizing Networks (WSOM)*, pages 53–58, Kyushu Institute of Technology, 2003.

[8] M. Strickert and B. Hammer. Unsupervised recursive sequence processing. In M. Verleysen, editor, *European Symposium on Artificial Neural Networks (ESANN)*, pages 433–439. D-side Publications, 2003.

[9] P. Tino and G. Dorffner. Predicting the future of discrete sequences from fractal representations of the past. *Machine Learning*, 45(2):187–217, 2001.

[10] M. Varsta, J. del R. Milan, and J. Heikkonen. A recurrent self-organizing map for temporal sequence processing. In *Proc. ICANN'97, 7th International Conference on Artificial Neural Networks*, volume 1327 of *Lecture Notes in Computer Science*, pages 421–426. Springer, Berlin, 1997.

[11] T. Voegtlin. Recursive self-organizing maps. *Neural Networks*, 15(8-9):979–991, 2002.