Proceedings of ESCI'2000, Kosice, Slovakia, Springer-Verlag, pp. 25-31

Self-Organizing Maps for Representing Structures

Igor Farkaš

Institute of Measurement Science, Slovak Academy of Sciences, Bratislava, Slovak Republic

Abstract. We propose a novel neural network model for representing data structures. The model consists of a hierarchy of Self-Organizing Maps (SOMs) equipped with leaky integrating units. Each of the maps is thus designed to represent sequences of data in a fashion resembling Barnsley's iterated function system. Each data structure is decomposed into a hierarchy of sequences where in all but the lowest levels a special symbol is substituted to represent corresponding subtrees. The advantage of this representation is that it is directly computable, and if neurally implemented using SOMs, it is computationally unexpensive. Preliminary simulations using simple symbolic tree structures demonstrate that obtained representations have the required property of systematic order.

1 Introduction

In most artificial intelligence tasks such as language processing and reasoning the need arises to represent data structures such as sequences and trees. Originally, these tasks were predominantly tackled using symbolic approaches, while neural networks were claimed to lack this representation power [1]. However, to face this critisism, during the last decade a number of architectures and algorithms were designed demonstrating the capability of connectionist approaches to generate structured representations as well. The best known examples include recursive autoassociative memory [2], tensor-product-based approaches [3,4] and systems using synchrony of firing to perform binding [5,6].

In this paper, we propose an alternative structure-representing model which incorporates the concept of *self-organization* (SR-SOM). It is based on a hierarchy of modified self-organizing maps with leaky integrating units. The key idea is that each data structure is decomposed (using some external parsing module) into a hierarchy of sequences each of which is represented separately in a map.

First we describe how symbolic sequences can be represented and learnt using the SOM. In recent years, there have appeared various attempts to enhance the standard, static map with ability to represent sequences of data. Existing approches can be classified into two groups.

The first group can be characterized as a *sequence mapping* by SOM, where every sequence of input patterns becomes represented by a subsequent firing of winners in the map. In proposed models (e.g. [7–9]), due to distributed representation of a sequence, the SOM units themselves do not become sequence recognizers, however.

The second group was introduced by Kohonen as *operator SOMs*, where each unit serves as a sequence processor (operator). In addition, neighboring units in the map are supposed to be sensitive to similar sequences (according to some similarity measure). In our approach, we employed *leaky integrator* SOM units, which were previously shown to be capable of topologically representing sequences [10,11]. The latter model [11] overcomes some limitations of the former (it works for longer sequences than the former) and was also employed in our model.

2 The model

Leaky integrating symbolic sequences with SOM formally corresponds to an Iterated Function System (IFS) introduced by Barnsley [12]. IFS is defined by a collection of affine contraction mappings

$$i(s) = \mu s + (1-\mu)x_i, \quad x_i \in \{0,1\}^N, \quad x_i \neq x_j \quad \text{for} \quad i \neq j \qquad (1)$$

functioning on a unit hypercube $X = \langle 0,1 \rangle^N$, with[1] $N = \lceil \log_2 M \rceil$, where M is the number of symbols in input alphabet, $s \in X$ and using contraction coefficient $\mu \in (0, .5\rangle$. The number of mappings is equal to M, the center $\{\frac{1}{2}\}^N$ of the hypercube is the starting point. IFS leads to spatial representation of symbolic sequences (for rigorous analysis published recently see [15]). IFS representation has a nice ordering property in that two sequences with common suffices are mapped close to each other in X. In Fig.1a, considering symbols $\mathtt{A} = [0,0], \mathtt{B} = [1,0], \mathtt{C} = [0,1]$ and $\mathtt{D} = [1,1]$, we show IFS representations of a few sequences.

Learning to represent sequences. Training SOM with leaky integrating units proceeds as follows. With each presentation of a symbol $\mathbf{x}$, every unit i in the map updates its input as

$$\mathbf{z}_i(t) = \mu\ \mathbf{z}_i(t-1) + (1-\mu)(\mathbf{x}(t) - \mathbf{w}_i(t)) \ ,$$

where $\mathbf{w}_i$ is the classical weight vector associated with unit i. At the beginning of each sequence, $\mathbf{z}_i = 0,\ \forall i$. Sequences are parsed sequentially, symbol by symbol. As in standard SOM, units compete for the current symbol and the winner c satisfies the condition

$$\mathbf{z}_c(t) = \min_i \{ \|\mathbf{z}_i(t)\| \} \ ,$$

where $\|.\|$ denotes the Euclidean norm. Weight adaptation in winner's neighborhood is based on the rule

$$\mathbf{w}_i(t+1) = \mathbf{w}_i(t) + \alpha(t) h_{ic}(t) \mathbf{z}_i(t) \ ,$$

[1] for $x \in \Re$, $\lceil x \rceil$ denotes the smallest integer y, such that $y \geq x$

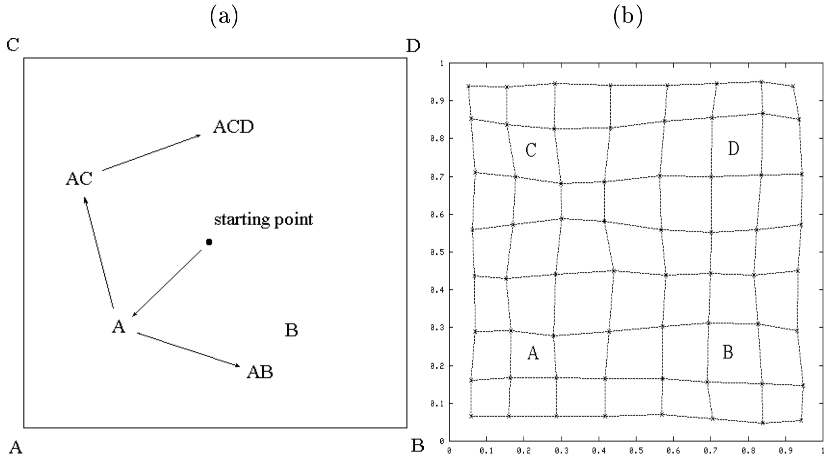

Fig. 1. (a) Examples of IFS sequence representations from the symbol set $\{\mathtt{A}, \mathtt{B}, \mathtt{C}, \mathtt{D}\}$ and for coefficient $\mu = 0.5$. (b) Ordered SOM with 8×8 leaky integrator units trained on all sequences with length ≤ 3, presented with equal probability. Symbol `A` shown denotes the surrounding area of 4×4 units representing all sequences having suffix `A`; similarly for other suffices. Due to fractal structure of this arrangement, the analogical subdivision applies to lower levels as well.

where parameters $\alpha(t)$ and $h_{ic}(t)$ have standard meanings of the decreasing learning rate and neighborhood function, respectively. As the SOM attempts to approximate the distribution of training data, we can obtain a nicely ordered map of sequence representations, such as the one shown in Fig.1b. It was trained on sequences with limited length, from 4-symbol set, where each symbol was selected with equal probability.

Note that due to linearity of encoding, the underlying sequence reconstruction is straightforward. Given the weight $\mathbf{w}(\tau)$, we repeatedly apply the formula inverse to eq. 1 until we end in the center of X or in its vicinity. The extracted symbol $\mathbf{x}_c$ in each iteration τ must satisfy the equation $x_c = \arg\min_l\{\|\mathbf{w}(\tau) - \mathbf{x}_l\|\}$.

Representing structure. Using the hierarchy of SOMs, one map per level, we can decompose each structure into a hierachy of sequences. At each level of this hierarchy, we can have one or more sequences to be represented in SOM. The difference between level 1 and all higher levels is that whereas in level 1 the sequences can contain only symbols belonging to input alphabet, in higher levels the sequences contain a symbol Δ which serves as substitution of a lower subtree (e.g., structure `A(AB)` becomes decomposed to `AB` in level 1 and `A`Δ in level 2).

During decomposition, it happens that two or more sequences need to be simultaneously represented in one map. To have the "multi-sequence" represention capability, the represention is taken as a global activity of the

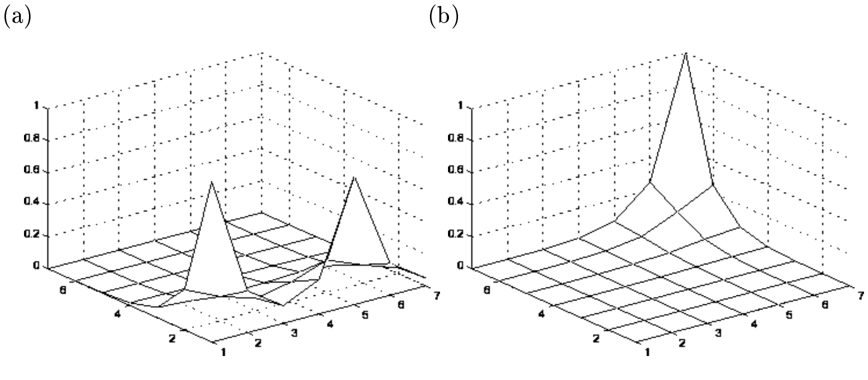

Fig. 2. Example of structure representation using SR-SOM. Structure (AB)(AA) is decomposed into sequences AB $\oplus$ AA in level 1 (figure a) and to ΔΔ in level 2 (figure b). Both maps have 7×7 units. For explanation see the text.

map. This choice is justified by observation that if topographic order exists in terms of units' positions in the map, it remains preserved if the output is taken as a global map activity having gaussian shape with a peak centered around the winner [13]. For that purpose, every unit has the RBF-like input-output function of the form

$$y_i = \exp(-\|\mathbf{r}_c - \mathbf{r}_i\|^2/\sigma) \ ,$$

where $\mathbf{r}_j$ is the position of unit j in the map, and σ determines the sharpness of the Gaussian surface. The overall SOM output is thus taken as vector $\mathbf{Y} = [y_1, y_2, ..., y_K]$, with K denoting the number of units. In order two distinguish between two structures containing the same sequences but in different order, such as (AB)(AA) and (AA)(AB), the corresponding peaks are combined in one SOM in leaky integrating fashion. This means that the earlier the sequence occurs in the structure, the lower the height of its peak will be ($y_i \leftarrow ky_i, k < 1,\ \forall i$). Fig.2 shows an example of representing a structure (AB)(AA). SOM in level 1 (SOM_1) has two peaks: one corresponding to the sub-sequence AA (left), the other for AB (smaller peak on the right, $k = 0.8$). SOM_2 in figure b has one peak corresponding to the sequence ΔΔ, where we considered $\Delta \equiv \mathrm{D} = [1, 1]$ for convenience.

For structure representations it is important that they themselves must be structured to allow structure-sensitive processing [14]. To check this property of SR-SOM representations, we have generated all 2-symbol sequences having one of the templates {(xx)x,x(xx),(xx)(xx)}. Each structure was represented by a 7×1 SOM_1 and a 7×7 SOM_2 as a concatenated vector $\mathbf{Y} = [\mathbf{Y}_1, \mathbf{Y}_2]$. SOM_1 had only one input (because there are only 2 symbols), whereas SOM_2 needed to have two inputs (because we have symbols A, B and Δ). From these representations, the hierarchical cluster diagram was calculated and plotted

```
    (AA)A  ----------|
    (BA)A  ----------|-|
    (AB)A  ----------| |-----------------------|
    (BB)A  ----------|-|                       |
    A(AA)  ----------|                         |
    A(BA)  ----------|-|                       |----------------------
    A(AB)  ----------| |-------------------|   |                      |
    A(BB)  ----------|-|                   |   |                      |
    B(AA)  ----------|                     |   |                      |
    B(BA)  ----------|-|                   |---|                      |
    B(AB)  ----------| |----------------|  |                          |
    B(BB)  ----------|-|                |  |                          |
    (AA)B  ----------|                  |--|                          |
    (BA)B  ----------|-|                |                             |
    (AB)B  ----------| |----------------|                             |
    (BB)B  ----------|-|                                              |
(AA) (BA)  --|                                                        |
(BA) (AA)  --|-----------|                                            |
(BA) (BA)  --------------|---|                                        |
(AB) (BA)  --|               |---|                                    |
(BA) (AB)  --|-------------| |   |                                    |
(BA) (BB)  --|             |-|   |                                    |
(BB) (BA)  --|-------------|     |---|                                |
(AA) (AB)  --|                   |   |                                |
(AB) (AA)  --|-------------|     |   |                                |
(AA) (BB)  --|             |-|   |   |                                |
(BB) (AA)  --|-------------| |---|   |--------------------------------
(AA) (AA)  ------------------|       |
(AB) (BB)  --|                       |
(BB) (AB)  --|-----------|-------|   |
(AB) (AB)  --------------|       |---|
(BB) (BB)  ----------------------|
```

Fig. 3. A hierarchical cluster diagram of SR-SOM representations of simple sequences contaning two symbols. Distances fall in range from 0.3 to 3.2.

to show the similarities.[2] It can be seen from Fig.3 that obtained representations indeed show a nice property that similar structures are similar also in terms of their SR-SOM representations $\mathbf{Y}(s_i)$.

3 Discussion

Preliminary experiments show that at least in case of simple structures consisting of a few symbols, the obtained SR-SOM representations appear to be structured. The similarity between two structures is induced by the property of IFS in which the longer common suffix of the two sequences, the more similar are their representations. [3] It can also be observed in the dendrogram (Fig.3), where e.g. structure (BA)(AA) is more similar to (BA)(BA) than to (BA)(AB). The advantage of these representations is their computational efficiency (to the contrary with RAAM) and transparency. The computational efficiency practically equals that of a SOM, because all training consists in learning subsequences, which result from structure decomposition.

Regarding the type of representation, SR-SOM appears to possess features of both classical, symbolic systems as well as connectionist systems. On the system level, the representation is *spatially concatenative* (using terminology in [17]), because individual map representations are concatenated together. At the same time, the representation at a particular level is maintained by a SOM which produces *distributed representations* associated with two inherent dimensions in the data. In our setting, the neurons are considered to be

[2] i.e., for two sequences s_1, s_2 their difference is taken as $\|\mathbf{Y}(s_1) - \mathbf{Y}(s_2)\|$

[3] This differs from Kohonen's symbol maps which employ "spatially well-balanced" similarity measure between strings [16].

overlapping *value units*, which enable simultaneous representation of multiple values in one map.

What needs to be further tested is how this approach scales with larger symbol set as well as more complex structures (both in terms of "height" and "width"). The core of SR-SOM applicability to more complex structures resides in our assumption that if we have more than 4 symbols in input alphabet (which increases the dimensionality of input space), in real tasks not all sequences appear in the data. This implies that mapping these sequences onto a 2-dimensional SOM while preserving local relationships may be feasible.

Acknowledgement: The work has been supported in part by VEGA grant No. 2/5088/2000.

References

1. J.A. Fodor and Z.W. Pylyshyn. Connectionism and cognitive architecture: A critical analysis. *Cognition*, 28:3–71, 1988.
2. J.B. Pollack. Recursive distributed representations. *Artificial Intelligence*, 46(1-2):77–105, 1990.
3. P. Smolensky. Tensor product variable binding and the representation of symbolic structures in connectionist systems. *Artif. Intelligence*, 46:159–216, 1990.
4. T. Plate. Holographic reduced representations. *IEEE Transactions on Neural Networks*, 6(3):623–641, 1995.
5. L. Shastri and V. Ajjanagadde. From simple associations to systematic reasoning. *Behavioral nad Brain Sciences*, 16(3):417–494, 1993.
6. J.E. Hummel and K.J. Holyoak. Distributed representations of structure: a theory of analogical access and mapping. *Psych. Review*, 104(3):427–466, 1997.
7. D.J. James and R. Miikkulainen. Sardnet: a self-organizing feature map for sequences. In *Advances in Neural Information Processing Systems 7*, 1995.
8. K. Kopecz. Unsupervised learning of sequences on maps with lateral connectivity. In *ICANN'95*, pages 431–436, Paris, France, 1995.
9. J. Göppert and W. Rosenstiel. Dynamic extensions of self-organizing maps. In *Proceedings of ICANN'94*, pages 330–333, 1994.
10. G.J. Chappell and J.G. Taylor. The temporal Kohonen map. *Neural Networks*, 6:441–445, 1993.
11. T. Koskela et al. Time series prediction using recurrent SOM with local linear models. *Int. Journal of Knowledge-Based Intell. Eng. Systems*, 2:60–68, 1998.
12. M. Barnsley. *Fractals Everywhere.* Academic Press, 1988.
13. I. Farkaš. Invariance of gaussian-vector mapping using a self-organizing map. *Neural Network World*, 7(2):153–159, 1997.
14. L. Niklasson and M. Bodén. Representing structure and structured representations in connectionist networks. In A. Browne, editor, *Neural Network Perspectives on Cognition and Adaptive Robotics*, pages 20–50. IOP Press, 1997.
15. P. Tiňo. Spatial representation of symbolic sequences through iterated function systems. *IEEE Trans. on Systems, Man and Cybernetics*, 29(4):386-392, 1999.
16. T. Kohonen. Self-organizing maps of symbol strings. *Technical report A42*, Helsinki University of Technology, Espoo, Finland, 1996.
17. N.E. Sharkey Connectionist representation techniques. *Artificial Intelligence Review*, 5:143-167, 1990.