Time Topology for the Self-Organizing Map

Panu Somervuo

Helsinki University of Technology
Neural Networks Research Centre
P.O. Box 5400 FIN-02015 HUT
Finland
panu.somervuo@hut.fi

Abstract

Time information of the input data is used for evaluating the goodness of the Self-Organizing Map to store and represent temporal feature vector sequences. A new node neighborhood is defined for the map which takes the temporal order of the input samples into account. A connection is created between those two map nodes which are the best-matching units for two successive input samples in time. This results in the time-topology preserving network.

1. Introduction

The Self-Organizing Map (SOM) [3, 5] is an unsupervised artificial neural network which defines a nonlinear transform from the input space to the set of nodes in the output space. Each node is associated with a model in the input space. Through an unsupervised learning process, these models become specially tuned and organized according to input patterns. The learning algorithm which leads to self-organization can be simplified into two steps [3, 5]: for each input sample, 1) the best-matching unit of the map is found by using the chosen similarity measure, and 2) the model of this unit as well as the models of its topological neighbors are adapted towards the input sample. Updating of the reference models can be done incrementally after each input sample or in a batch process [5].

In this work the time information of the input samples is taken into account when constructing the connections between map nodes. The reference models associated with the SOM nodes are first trained in the usual way, treating the input samples as static, separate vectors and defining the node neighborhood on the regular map grid. Once the map has been trained, old node connections are removed and new connections are created according to the best-matching unit trajectories corresponding to the temporal input sample sequences. SOM training can then be continued by using the new node connections as a neighborhood when adapting the reference models. Node connections represent signal paths in the input space and two input items which occur close to each other in time are mapped to neighboring map nodes. This *time-topology* preserving network is able to store and preserve temporal relations of the input items.

2. Map lattice

Neighborhood is an essential part of the SOM. It can be defined as a closeness of the map units in the output space. If that is a vector space, each map unit is provided with a position there. Depending on these positions, the lattice is then either regular or irregular. Although the word lattice may imply some kind of regularity, in this paper it is used to denote any kind of node arrangement.

Instead of defining the output space in the vector space, another possibility is to consider only connections between the map units. The SOM network can be described as a graph, where vertices denote the map units and edges denote the adjacency between them. In this case the familiar Euclidean distance, like any other vector-space distance, cannot be used for measuring the relative positions of the nodes in the network. Then the graph distance can be used, which also satisfies the properties of a true metric.

The choice of the node connections, and thus the neighborhood, affects highly the capability of the SOM to preserve the topology of the input in the mapping. Perfect topology preservation requires that adjacent input items are mapped to adjacent (or identical) map nodes. Since the map can be divided into nodes and their connections, and one node is associated with a reference model representing the local input space only, topology preservation is a demand for the network as an entity. The node connections play the key role at this.

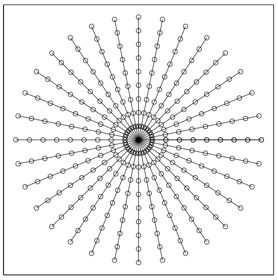


Figure 1: Input data for the first time topology experiment. Sequences of two-dimensional feature vectors proceed from the origo to the unit circle. Input samples are depicted by dots and successive input samples in time are connected by lines.

3. Time topology

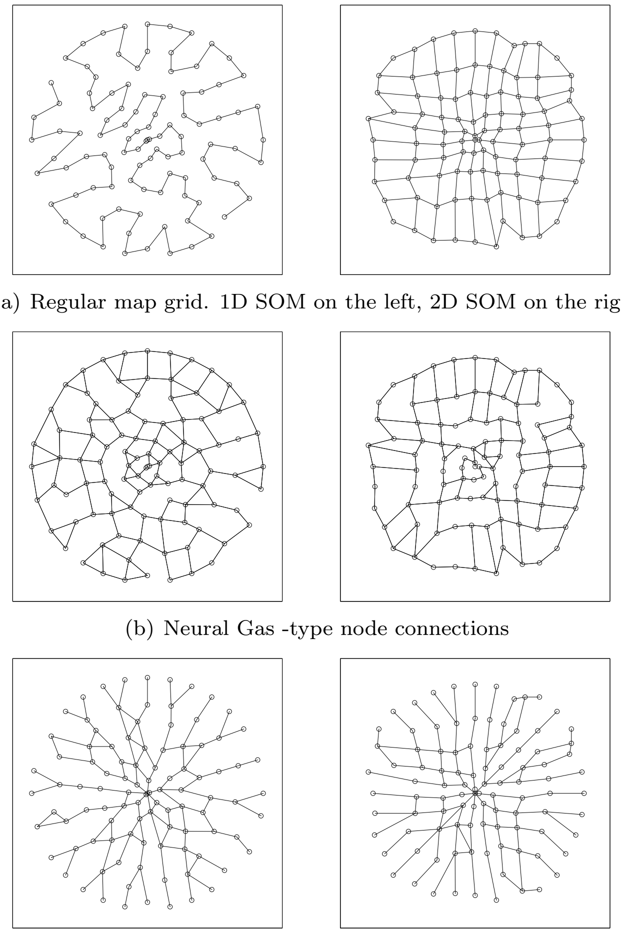
Usually the map lattice is a regular grid where a symmetric neighborhood function is defined. Other map topologies which have been experimented include the minimum spanning tree [1] and the Neural Gas network [7]. The main idea in the current work is to consider the time information of the input when forming the node connections and defining the neighborhood. A straightforward way to do this is to connect those two nodes which are the best-matching units for two consecutive input samples in time. Since any two nodes can be connected independently of their Euclidean distance on the regular map lattice, the new connections may provide "worm-holes" to the original map lattice space.

Different types of node connections are illustrated in Figure 2.

4. Experiments with speech data

The SOM with the time topology was experimented with speech data. 15 male speakers and 5 female speakers had each uttered four times the vocabulary of 22 Finnish command words. Feature sequences were computed from these 1760 utterances. They consisted of 10-dimensional cepstrum vectors which were computed from 16 ms time windows with 8 ms time spans.

One experiment consisted of training the SOM and creating the time topology. The average quantization error and the word recognition error was then computed using a separate test set. Different types of node connections were used for the comparison of the time-topology network. In each experiment, the test set consisted of 88 feature sequences from one speaker and the training set consisted of 1672 feature sequences from the rest of the speakers. The tests were repeated 20 times, each



(a) Regular map grid. 1D SOM on the left, 2D SOM on the right

(b) Neural Gas -type node connections

(c) Time topology

Figure 2: Three types of map node connections. Input data consist of sequences of two-dimensional feature vectors proceeding from the origo to the unit circle as shown in Figure 1. One- and two-dimensional SOMs with 100 nodes were constructed using this data (a). The prototype vectors are depicted by dots and the neighborhood connections are depicted by line segments. Figure (b) shows the connections created between the nodes which are the two best-matching units for each single input sample. Reference models of the map nodes were taken from the SOMs in (a). Figures on the bottom row (c) represent the connections created between the best-matching units of two successive input items in time. This gives a representation of temporal signal paths in the feature space. The reference models were the same as in the upper maps. Networks in (c) resemble clearly best the original input data.

time having a different speaker in the test set. All results are thus averages of 20 test runs and altogether 1760 test sequences.

When computing the quantization error and the word recognition error, test sequences were encoded into map node sequences so that each feature vector sequence was projected on the map as an entity. The map node trajectory corresponding to the input sequence was computed using dynamic time-warping [9]. The node sequence which had the minimum cumulative sum of the squared vector distances between the model vector of the node and the feature vector of the input sequence and which formed a connected path in the network was the resulting sequence. In every node transition in the sequence, one feature vector of the input sequence was expended and a transition from one node to another was allowed only if there existed a connection between them. This approach resembles the Viterbi search [8], the only difference was that instead of state and transition probabilities, a quantization error between the reference vectors and the input feature vectors was used.

Distances along the map lattice have earlier been used for evaluating the goodness of the map in [6] and [2]. But in those works, the path on the map has been computed between two best-matching units for one input vector, not for the whole input vector sequence.

4.1. SOM training with the regular map lattice

Reference models of the SOM were 10-dimensional feature vectors. Eight different initial map lattices were experimented. These were 1-, 2-, 3-, and 4-dimensional hypercubes, and for each of them, two different map sizes were experimented. Initialization of the reference vectors was done according to the principal components of the feature vectors in the training set. The largest eigenvalues of the covariance matrix of the training set determined the ratio of the side lengths of the map lattice. The reference vectors were initialized according to the lattice coordinates of the map nodes so that each component of the lattice coordinate vector referred to one eigenvector of the covariance matrix. These eigenvectors corresponded to the largest eigenvalues. The mean of the training vectors was then added to the reference vectors in order to move the center of the initial map to the data mean. The side lengths of the map lattice had to be integer numbers and the total number of the nodes was limited to be either 120 or 420. Since the largest eigenvalues of all 20 different training sets were almost equal, the sizes of the map lattices used in the experiments were fixed. These are shown in Table 1.

Table 1: Map lattice sizes and Batch-Map parameters used in the experiments. Lattices are 1-, 2-, 3-, and 4-dimensional hypercubes with 120 and 420 nodes.

number of nodes in map lattice	kernel width of Gaussian neighborhood	batch rounds
120	60 . . . 1	60
12×10 = 120	6 . . . 1	20
6×5×4 = 120	3 . . . 1	20
5×4×3×2 = 120	2 . . . 1	20
420	100 . . . 1	100
28×15 = 420	10 . . . 1	20
10×7×6 = 420	5 . . . 1	20
7×5×4×3 = 420	3 . . . 1	20

Map training was done using the Batch-Map algorithm [5]. In this phase of the experiment the neighborhood of the nodes was defined on the regular map grid. A Gaussian neighborhood function was used with slowly decreasing kernel width in order to preserve the map orderliness as well as possible. The initial map was ordered due to the initialization procedure described above.

After map training, old node connections were removed and new connections were created. Four different methods were now experimented:

1. Node connections according to the regular map lattice. In the N-dimensional hypercube each node which is not on the edge of the lattice has $2N$ neighboring units inside the radius of one grid unit length; two neighbors for each lattice dimension.

2. Neural Gas -type connections. Those two nodes are connected which are the best- and second-best-matching units to one input sample. This gives the node topology approximating the data manifold of the static input items.

3. Time topology. A connection is created between two nodes which are the best-matching units for two successive input samples in time.

4. All nodes are connected to each other. This network forms a complete graph.

The number of the node connections in the network using the four different methods described above is shown in Table 2. One connection is a directed one, and therefore all symmetric connections are counted as two. Only

Table 2: Average number of node connections in the network. Row "time2" corresponds to the experiment where the time topology was used as a node neighborhood when adapting the reference vectors. Other rows correspond to experiments where the SOM was trained using the regular map lattice and node connections were recreated only for encoding the input sequence to the map node sequence.

SOM with 120 nodes				
node connections	1D	2D	3D	4D
radius 1	358	556	692	772
radius $\sqrt{2}$		952		
Neural Gas	2229	1779	1799	1856
time1	5968	5952	6140	6055
all	14400	14400	14400	14400
time2	5353	5411	5420	5443

SOM with 420 nodes				
node connections	1D	2D	3D	4D
radius 1	1258	2014	2596	3002
radius $\sqrt{2}$		3526		
Neural Gas	7441	6836	7447	7729
time1	23944	23606	23374	22765
all	176400	176400	176400	176400
time2	21504	21574	21502	21510

in time-topology networks, the connections were not symmetric. In all types of the node connection, there was a self-connection from each node to itself. This enables the time-warping of the input sequence when it is encoded to the node sequence.

Quantization errors of the test sequences were used to investigate the goodness of different map topologies. Each sum of the squared vector distances was divided by the length of the input sequence and these quantization errors were then averaged over all test sequences. The results are shown in Table 3. Since the prototype vectors were fixed before changing the node connections in each experiment, the results using different topologies can be easily compared. Some comparisons can also be made between different map lattices. The width of the Gaussian neighborhood kernel was 1.0 at the end of the training for all map lattices in the experiments. It can be expected that the quantization error increases as a function of the lattice dimension if the number of the map nodes is kept the same and transitions from one node to all others are allowed. This is because the map becomes more stiff when the lattice dimension grows; the number of the map nodes inside a constant neighborhood radius increases. But if the quantization error is computed along the node connections on the regular map lattice and if the lattice dimension is too low for the input, although the map is flexible, the quantization error of the input sequence may be large if there are rapid transitions between successive input vectors. In a higher-dimensional map, the number of connections between any two nodes on the map lattice is smaller which increases the tolerance to the rapid changes in the input sequence, but the reference vectors are then not necessarily spread to the whole input space due to the stiffness of the map.

Table 3: Average quantization error of test sequences computed along the node connections.

SOM with 120 nodes				
node connections	1D	2D	3D	4D
radius 1	1165.2	547.1	488.7	536.4
radius $\sqrt{2}$		495.4		
Neural Gas	418.7	437.3	471.2	525.3
time1	412.6	427.4	459.8	515.8
all	412.4	427.1	459.5	515.5
time2	330.6	332.2	332.0	331.9

SOM with 420 nodes				
node connections	1D	2D	3D	4D
radius 1	1196.6	724.2	424.9	424.7
radius $\sqrt{2}$		628.8		
Neural Gas	319.3	353.1	374.0	403.5
time1	305.0	337.3	353.8	385.8
all	303.3	335.5	351.9	383.8
time2	264.5	264.8	265.5	266.6

The best lattice of the regular hypercubes seems to be three-dimensional, see the row "radius 1" in Table 3. However, although almost all phonemes of Finnish were represented in the current speech material, only a fraction of the phoneme transitions were represented. Therefore, for a larger speech material, a higher-dimensional map lattice could be better.

The quantization error of the input sequences computed along the map topology gives information about the capability of the map to represent temporal data sequences. But if the prototype vectors are kept fixed, and only the node connections are recreated, it is clear that the lowest quantization error is achieved when each node has connections to all other nodes in the network. However, this does not give information about the signal paths or the topology of the input data sequences.

If the ability of the map to represent signal paths is of interest, the same node connections which were used in the SOM training should be used when computing the quantization error. Then the time topology is the best choice. Neural Gas -type connections do not necessarily give continuous connections to the whole input sequence because they give the topology only for the static input data manifold.

4.2. Network for speech recognition

Aforementioned quantization error of the sequences does not necessarily give easily the information of the capability of the network to store and represent sequences. This is because those results can be compared only relatively. Therefore the performance of the network was investigated in the speech recognition task. The prototype vectors of the aforementioned node sequences were matched against the reference templates of spoken words. Dynamic time-warping was used to compute the distance between sequences [9]. From the speech-recognition point of view, the original unquantized input sequences could have been used now, but the main idea in this test was to investigate the capability of the network to store and represent sequences.

One reference template was used for each word class. It was a classwise median sequence of the training set. A median sequence is a sequence with the smallest sum of distances to other sequences in the set [4].

Recognition results using the prototype vectors of the node sequences are shown in Table 4. For comparison, the unquantized test sequences were also matched against the reference templates. Then the average recognition rate was 96.2 per cent.

Another experiment was carried out in parallel with the previous test. After Batch-Map training, the map nodes were labeled using phonetically pre-segmented training data. Each map node received a probabilistic label vector. The dimension of the label vector was 19 since there were altogether 19 different phonemes in the words of the vocabulary. The value of each label vector component of one node was determined by the number of the times that node was the best-matching unit to the data vector from the corresponding phoneme class. All vectors were then normalized to be unit vectors.

Encoding of the input feature sequence to the node sequence was done as before, i.e., finding the map node trajectory in the feature space along the node connections, but after that the probability label vectors of the nodes were used in matching the map node sequence against the reference templates. Again, one reference template was used for each word class. That reference template represented ideal phoneme sequence, i.e., in each vector of the reference sequence there was only one nonzero component corresponding to the correct phoneme in the sequence and the rest of the components were zeros. The lengths of the phoneme segments in the reference templates were average lengths of the phoneme segments in the training set. The recognition results of this test are shown in Table 5.

Table 4: Speech recognition with 10-dimensional prototype vectors as features. Number of correct words per cent.

SOM with 120 nodes				
node connections	1D	2D	3D	4D
radius 1	33.5	87.3	91.1	86.5
radius $\sqrt{2}$		89.7		
Neural Gas	94.2	91.4	91.5	86.6
time1	94.7	91.2	91.2	86.8
all	94.7	91.2	91.2	86.7
time2	95.9	95.7	95.9	95.8

SOM with 420 nodes				
node connections	1D	2D	3D	4D
radius 1	29.3	78.5	93.6	91.5
radius $\sqrt{2}$		85.9		
Neural Gas	95.8	95.0	94.1	91.8
time1	96.1	95.2	94.4	91.8
all	96.0	95.2	94.3	91.8
time2	96.0	96.1	96.1	96.4

The motivation for this experiment was to test how well the network is able to produce quasiphoneme sequences. Since the network allows decoding of arbitrary feature sequences, word recognition with unlimited vocabulary can be performed in such phonetic languages as Finnish. The conversion from the node sequence to the symbol sequence can utilize similar techniques as has been used in [10].

4.3. SOM training using the time topology

In the previous experiments, the prototype vectors associated with the SOM nodes were trained by using the neighborhood on the regular map lattice after which the prototype vectors were fixed. Now their training was continued so that in each batch round the updating neighborhood was newly defined according to the time topology. The Batch-Map algorithm was performed five rounds with the constant neighborhood radius. The neighborhood function which was 1 to the

Table 5: Speech recognition with 19-dimensional probabilistic class vectors as features. Number of correct words per cent.

SOM with 120 nodes				
node connections	1D	2D	3D	4D
radius 1	31.9	70.3	76.2	75.9
radius $\sqrt{2}$		74.9		
Neural Gas	78.2	78.3	80.1	76.8
time1	79.1	78.9	79.8	77.3
all	79.0	78.8	79.8	77.3
time2	84.2	83.1	83.8	83.6

SOM with 420 nodes				
node connections	1D	2D	3D	4D
radius 1	28.2	55.3	82.0	81.2
radius $\sqrt{2}$		64.9		
Neural Gas	85.2	83.9	85.5	82.6
time1	86.3	84.3	86.6	82.8
all	85.7	84.3	86.8	83.4
time2	88.7	89.5	88.6	88.6

best-matching unit itself, $e^{-1/2}$ to its time-topological neighbor, and zero otherwise, was weighted according to the strength of the node connection. This was determined by the number of the input sample pairs corresponding to each node connection. The final neighborhood function of each node was then normalized so that it summed to one.

The number of the node connections, quantization error, and recognition results are shown in Tables 2, 3, 4, and 5, corresponding to the row "time2".

The results using the time-topological node neighborhoods when adapting the reference vectors of the SOM are mutually very similar for different initial map lattices (1D, 2D, 3D, and 4D-hypercubes). Quantization error of the test sequences decreased considerably compared to the results of the previous experiments. This is very satisfactory and shows the effect of the proposed method. It is interesting to find also that the number of the node connections decreased in the network during training. This can be seen by investigating the rows "time1" and "time2" in Table 2. This concentration of the node connections and represented signal paths is an interesting emergent feature of the network.

When comparing the results of the speech recognition experiments it should be noted that the network was trained for representing the sequences, not for discriminating different classes. However, the results are better than those of the previous experiments and very close to the results of using unquantized input sequences. Nevertheless, supervised discriminative training could also be applied to the network with the time topology.

5. Conclusions

In this work, the Self-Organizing Map was provided with the time topology. Construction of such a map was here done in two steps, first training the SOM in the usual way, considering the input samples as static vectors and defining the node neighborhood on the regular map lattice. In the second phase old node connections were removed and new connections were created according to the time information of the input samples. The learning was then continued with the new node topology and neighborhood. Creating the time topology requires to seek the best-matching unit for each input sample. The two nodes which are the best-matching units for two successive input samples in time are connected. The resulting network gives a representation of temporal signal paths in the input space. This was experimented with the feature vector sequences computed from speech.

6. References

[1] J. Kangas, T. Kohonen, and J. Laaksonen, "Variants of Self-Organizing Maps", *IEEE Trans. Neural Networks*, vol. 1, no. 1, pp. 93-99, 1990.

[2] S. Kaski and K. Lagus, "Comparing self-organizing maps", *Proc. ICANN'96*, pp. 809-814, 1996.

[3] T. Kohonen, "Self-organized formation of topologically correct feature maps", *Biological Cybernetics*, vol. 43, pp. 59-69, 1982.

[4] T. Kohonen, "Median strings", *Pattern Recognition Letters*, vol. 3, pp. 309-313, 1985.

[5] T. Kohonen, *Self-Organizing Maps.* Springer, Berlin, 1995.

[6] M. Kraaijveld, J. Mao, and A. Jain, "A Non-Linear Projection Method Based on Kohonen's Topology Preserving Maps", *Proc. 11ICPR*, pp. 41-45, 1992.

[7] T. Martinetz and K. Schulten, "A Neural-Gas Network Learns Topologies", In: T. Kohonen et al. (Eds.), *Artificial neural networks*, vol. I, pp. 397-402, Elsevier, Amsterdam, 1991.

[8] L. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition", *Proc. IEEE*, vol. 77, no. 2, pp. 257-286, February 1989.

[9] H. Sakoe and S. Chiba, "Dynamic programming algorithm optimization for spoken word recognition" *IEEE ASSP*, vol. 26, no. 1, pp. 43-49, 1978.

[10] K. Torkkola, J. Kangas, P. Utela, S. Kaski, M. Kokkonen, M. Kurimo, and T. Kohonen, "Status report of the Finnish phonetic typewriter project", In: T. Kohonen et al. (Eds.), *Artificial Neural Networks*, vol. I, pp. 771-776, Elsevier, Amsterdam, 1991.