

## Interpolating self-organising map (iSOM)

H. Yin and N.M. Allinson

A new learning algorithm is presented for enhancing the scale or structure of an already trained self-organising map (SOM) without the need to re-use the original training data. Alternative methods for the insertion of these additional interpolating neurons, while still preserving the learnt topology, are presented together with two illustrative examples of the algorithm in operation.

**Introduction:** The self-organising map (SOM) [2, 3] is an unsupervised learning algorithm that clusters and projects potentially high dimensional input data onto a discrete neural grid or map of usually reduced dimensions. It is often applied with a predetermined size and structure as the correct size is rarely known *a priori*. Several methods for adapting the map's size or structure during training have been developed, such as the neural gas method [3], the growing grid method [4], and the growing SOM [5], as variants of the standard SOM. More nodes or grid rows and columns are added to the map 'online' according to the nodes' firing frequencies or local map distortions. Thus the map's size or structure changes with the training process so adapting to the input space. Some of these methods, however, do not preserve the existing topology of the original map, as the structure of the map grows arbitrarily with the input data. This may not affect some applications such as pattern classification, but will make data visualisation almost impossible. Also, the inserted map nodes of these algorithms, if trained concurrently with the original map, require all training data in order to permit convergence of the whole map.

There are situations where more nodes or higher resolution (hence larger map) are needed when the map is already, or almost, fully trained; and previous training data may no longer be available (e.g. in real-time video compression). Re-training of a map of a different size by the standard SOM algorithm is almost impossible. This type of problem is often encountered in practice. For example, after an SOM is trained, it is found that the map is too small or under-defined, as the appropriate size was not known at the beginning. A larger SOM could be retrained from scratch, if the training data were still available. However, the trained SOM can be interpolated based only on the weights of the learnt map, in order to obtain an approximate larger and finer map. None of above growing algorithms is suitable for this post-training adaptation. This interpolation can also be regarded as a scaling process, which serves to provide a basic blueprint for the map. That is, nodes of original map can be used as large units to capture the data's cluster centres, while the interpolated nodes, filling the spaces between the older nodes, act as fine gradations. This is useful in data visualisation, that is instead of clustering to a few centres, the sample points are mapped onto finer nodes surrounding these centres. So the intra-cluster relations can be approximately measured or displayed. In this Letter a simple learning algorithm, which does not use numerical methods (e.g. polynomials), for both linear and nonlinear interpolations in iSOMs is presented. Two examples of this new approach are provided.

**Interpolating SOM (iSOM) and algorithms:** Assume that the initial size of the map is  $N_1 \times N_2 \times \dots \times N_k$  of  $k$  dimensions.  $\Omega$  is the index set of the map. After training (i.e. the first training phase), the map is found to be too small or have too low a resolution. Then a larger and finer map can be formed by adding more nodes between the original nodes without the need to train a new larger map again from scratch. Assume that we wish to add  $q$  columns and rows between existing columns and rows, respectively (unequal number of columns and rows can also be used), equivalently,  $q^k$  ( $= q \times q \times \dots \times q$ ) nodes to each hypercube of the original map and  $q$  nodes on each edge (vertex) of the hypercube. The new map will be of size  $[N_1 + (N_1 - 1) * q] \times [N_2 + (N_2 - 1) * q] \times \dots \times [N_k + (N_k - 1) * q]$ . An algorithm, similar to the standard SOM algorithm, can be used to train these new nodes but from the trained map weights only; i.e.

$$\mathbf{w}_c(t+1) = \mathbf{w}_c(t) + \alpha(t) * \eta(t, c, \mathbf{v}) (\mathbf{W}_v(t) - \mathbf{w}_c(t))$$

$$c \in \{\Omega' - \Omega\} \quad (1)$$

where  $\Omega'$  is the index set of the new map after the insertion of new nodes.  $\mathbf{w}_c(t)$  is the weight vector of the inserted node  $c$ ,  $\mathbf{W}_v(t)$  is

the weight of a randomly picked node from original trained map, but  $v$  is the index according to its position in the new map,  $\alpha(t)$  is the learning rate (as used in the SOM or standard stochastic algorithm), and  $\eta(t, c, v)$  is the neighbourhood function, which has a pre-specified and fixed form and effectively restricts the learning to a small neighbourhood only.

In the second training phase, as in the ordinary SOM, the initial weights of the inserted nodes can be randomly chosen. However, to speed up convergence, they can be set to the medium values (with or without a small random dispersion) of the hypercubes that they are inserted into. At each learning step, a node from the original map is chosen randomly, and its weights are input  $\mathbf{W}_v(t)$ . Then its neighbouring new nodes, i.e. the inserted nodes in the hypercubes between  $v$  and its all nearest neighbouring old nodes, are updated using the above rule. These learning rules are similar to those of the SOM but with inputs replaced by weights of the initially trained nodes, which are not subject to the above updating. The original topology obtained by the first training process will be preserved in this new map.

Two types of interpolations can be anticipated, namely linear and nonlinear. In the former attempts are made to linearly (or uniformly) place the added nodes in the hypercubes which would be most useful for visualisation. In the latter attempts are made to add these new nodes in as smooth a manner as possible, in the hope that the new map  $\Omega'$  will be, or at least close to, a larger map if it had been trained from the outset.

From an analysis developed for self-organising mixture networks [6, 7], we know that if a linear neighbourhood function, which is maximum, i.e. one, at the winning node and linearly decreases to zero at its nearest old neighbouring nodes, is used then a uniform distribution among the inserted nodes within this area will be imposed, so a linear interpolation can be achieved. Such a neighbourhood function is

$$\eta(t, c, v) = \frac{(q+1 - |c-v|)}{q+1} \quad (2)$$

$$\eta(t, c, \mathbf{v}) = \frac{(q+1 - |c_x - v_x|)}{q+1} \cdot \frac{(q+1 - |c_y - v_y|)}{q+1} \quad (3)$$

$$\eta(t, c, \mathbf{v}) = \prod_{i=1}^k \frac{(q+1 - |c_i - v_i|)}{q+1} \quad (4)$$

for 1D, 2D, and  $k$ D mappings, respectively. In the above definitions, uniform insertion (i.e. the same number  $q$  for each hypercube) is assumed. Extension to non-uniform cases is straightforward.

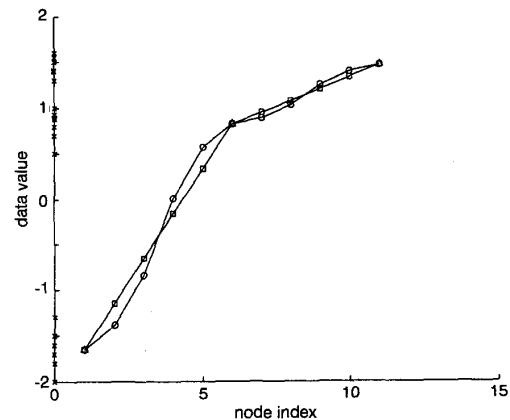
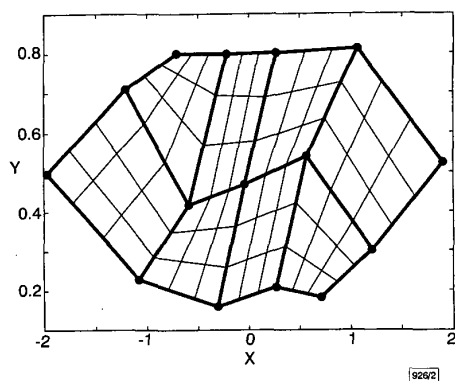


Fig. 1 1D linear and nonlinear interpolating SOMs

△ three-node SOM  
□ linear SOM  
○ nonlinear iSOM  
× original data

For nonlinear interpolation (i.e. distribution among inserted nodes is not uniform) Gaussian neighbourhood functions can be

used with a width proportional to half of the inter-neuron index distance (for the old map). For example in 1D, it can be  $\exp(-(c - \nu)^2/2\sigma^2)$ , where  $\sigma$  should be between  $(q+1)/4$  and  $(q+1)$ . In this case the aim is to use new nodes to approximate the large map as if it were trained from the beginning.



**Fig. 2 2D linear interpolating SOM**  
 ● — original trained 5 × 3 SOM  
 — interpolated nodes iSOM is 13 × 7 in size

**Experiments and results:** In the first example, a three-node SOM chain was trained on the data set  $\{-2, -1.8, -1.7, -1.6, -1.5, -1.3, 0.5, 0.7, 0.8, 0.88, 0.9, 0.93, 1.0, 1.3, 1.4, 1.41, 1.5, 1.55, 1.6\}$ . The result, after 8000 iterations, is shown in Fig. 1 as triangles. Four new nodes ( $q = 4$ ) have been inserted in each of the two intervals by the iSOM algorithm for both linear and nonlinear interpolations. The enhanced chains, after 3000 iterations, are shown in Fig. 1. This very simple example indicates the successful inclusion of additional neurons and shows the slight differences between linear and nonlinear interpolations.

In the second example, a 5×3 SOM initially learnt from an input source, where the  $x$  component follows the  $N(0, 1)$  normal distribution and the  $y$  dimension is uniformly distributed between  $[0, 1]$ . Fig. 2 shows the map after 8000 iterations. From this map, the iSOM learns to linearly insert two more columns and rows ( $q = 2$ ) between existing ones and increases the map to 13 × 7 in size. The result shown is after 5000 learning steps from the old map.

**Conclusions:** We have proposed a post-training method for the interpolation of a trained SOM, so that the map can be enlarged without retraining from scratch or the need to use the original data. The algorithm is simple but efficient in magnifying a learnt map without losing its previously formed topology. The iSOM is also suitable for 'online' growing if all nodes learn from the input data after new nodes are inserted.

The SOM has been used for data visualisation as an alternative tool to Sammon mapping [2]. An SOM is trained on the data, which are then projected onto the map using U-matrix colours to reflect inter-neuron or inter-sample distances. In the iSOM, interpolated nodes can act as guides to a trained map as they will be distributed between original nodes in the trained map. Data points are then mapped onto inserted and original nodes (rather than just original nodes). So the interrelationships between sample points can be visualised more precisely in a Euclidean-distance sense, which will be more visually appealing than the U-matrix colour scheme.

© IEE 1999  
 Electronics Letters Online No: 19991149  
 DOI: 10.1049/el:19991149

H. Yin and N.M. Allinson (Department of Electrical Engineering and Electronics, UMIST, PO Box 88, Manchester M60 1QD, United Kingdom)

**References**

- 1 KOHONEN, T.: 'Self-organised formation of topologically correct feature maps', *Biol. Cybernetics*, 1982, 43, pp. 56–69
- 2 KOHONEN, T.: 'Self-organising maps' (Springer, Berlin, 1997)

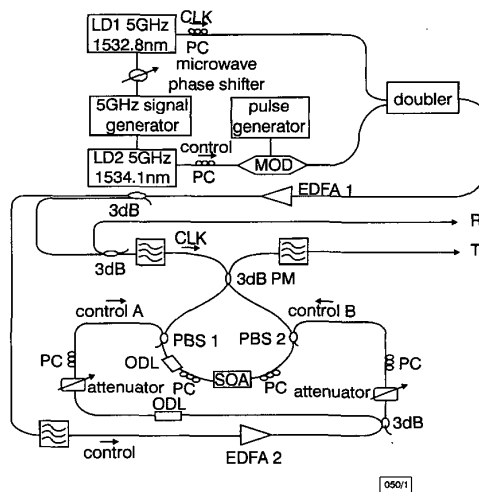
- 3 MARTINETZ, T.M., and SCHULTEN, K.J.: 'A "neural-gas" network learns topologies' in KOHONEN, T., *et al.* (Eds.): 'Artificial neural networks' (1991), pp. 397–402
- 4 FRITZKE, B.: 'Growing grid—a self-organising network with constant neighbourhood range and adaptation strength', *Neural Process. Lett.*, 1995, 2, (5), pp. 9–13
- 5 BAUER, H.-U., and VILLMANN, T.: 'Growing a hypercubical output space in a self-organising feature map', *IEEE Trans.*, 1997, NN-8, (2), pp. 218–226
- 6 YIN, H., and ALLINSON, N.M.: 'Bayesian learning for self-organising maps', *Electron. Lett.*, 1997, 33, (4), pp. 304–305
- 7 YIN, H., and ALLINSON, N.M.: 'A self-organising mixture network for density modelling'. 1998 IEEE Int. Conf. Neural Networks, Alaska, USA, 1998, Vol. 3, pp. 2277–2281

**10Gbit/s all-optical Boolean XOR with SOA fibre Sagnac gate**

T. Houbavlis, K. Zoiros, A. Hatziefremidis, H. Avramopoulos, L. Occhi, G. Guekos, S. Hansmann, H. Burkhard and R. Dall'Ara

All-optical Boolean XOR logic is demonstrated with a three terminal fibre Sagnac interferometer employing a semiconductor optical amplifier. Full duty cycle operation at 10GHz is shown and low pattern dependence has been achieved on a pseudo-data pattern at 10Gbit/s with low switching energy.

**Introduction:** Ultra-high speed optical time division multiplexing (OTDM) multiaccess networks are being developed to satisfy the increasing bandwidth demand due to the massive use of the Internet and multimedia. Users will access these networks at nodes interfacing the high-speed electronics to the ultra-high-speed optical data bus by performing on-the-fly a set of all-optical processing operations. Boolean XOR operation is particularly important for OTDM network functions including address and header recognition as well as data encoding and encryption. All-optical Boolean XOR operation has been demonstrated using the nonlinear fibre loop mirror at 10Gbit/s [1] and more recently at 100Gbit/s [2]. All-optical logic operations have also been performed using the nonlinearity of semiconductor optical amplifiers (SOAs) [3–5] up to 100Gbit/s [6], but so far the demonstration of XOR operation has been restricted to 1GHz [7] and more recently to 5 GHz with a full duty cycle control signal [8]. Given that semiconductor-based switching devices possess the practical advantages of lower switching energy and lower latency, the demonstration of XOR operation at higher rates is particularly important for all-optical applications requiring feedback. In this Letter we report full duty cycle and pseudo-data control pattern XOR operation of an SOA assisted Sagnac interferometer gate at 10GHz and 10Gbit/s, respectively, with low switching energies and a low pattern effect on the switched pulses.



**Fig. 1 Experimental setup**