Proceedings of the First International Conference on Machine Learning and Cybernetics, Beijing, 4-5 November 2002

ACHIEVING SELF-ORGANIZATION BY LATERAL INHIBITION

BIN TANG, MICHAEL SHEPHERD

Faculty of Computer Science, Dalhousie University, Halifax, Nova Scotia, Canada B3H 1W5
E-MAIL: btang@cs.dal.ca, shepherd@cs.dal.ca

Abstract:

A new model based on self-organization by lateral inhibition (SOLI) is proposed for self-organizing networks. This model combines many of the good features of previous models while overcoming many of the drawbacks. Experiments on this new model indicate that SOLI is well suited for unsupervised learning tasks, such clustering, has the potential to preserve topology, and can be used for novelty detection. It is computationally efficient with O (n) time complexity and is not sensitive to the initial network parameters.

Keywords:
Self-organizing networks; Lateral inhibition; Unsupervised learning; SOM

1 Introduction

Many computational models have been proposed to simulate the topological ordering property of the cortical area of the human brain [1, 2, 3]. These models have all demonstrated that global topographic ordering can be achieved by local cooperation and competition among neurons in the network. Perhaps the most popular such model is Kohonen's self-organizing feature map (SOM). The SOM is particularly popular when applied to real life applications. A more complete review of its technical applications can be found in [4].

Kohonen's SOM was motivated originally by Hebbian learning. In Hebbian learning, the external stimuli are the major sources of excitation of the neurons. Due to different degrees of excitation, neurons cooperate and compete through lateral connections. In standard SOM, lateral connections are approximated by a Gaussian shaped topological neighborhood function, which causes the excitation to decrease monotonically with the distance from the most excited neuron. Lateral inhibition is not explicitly implemented in the standard SOM.

SOM is one of the best unsupervised neural network models. It is a powerful visualization tool that can project complex relationships in a high dimensional input space onto a 2D grid while "preserving" the topological relationship to the best degree. The self-organization process of SOM is fully input driven in an automatic fashion. After training, the neurons (usually set on 1D or 2D grids) are topologically ordered, such that neurons that are close in neighborhood on the grids are close in the input space.

But SOM does have its own drawbacks. The topology preserving property is questionable when applied to high dimensional data. It cannot guarantee that a neighborhood relationship in the input space will be maintained on the final map. This topology preserving property of SOM has been rigorously investigated [5, 6]. To apply SOM properly, many network parameters have to be predetermined, such as, the size and shape of the network, the total number of trainin4g epochs and the initial learning rate. The initial values for such parameters will affect the quality of the final map, but optimal values may not be known in advance. Topological defects (kinks) may occasionally form by some random initialization of the network, which may not be unwrapped during the limited training time [7]. Another criticism of SOM is the long training time, which is required to warrantee the final convergence and topological ordering. Though there have been many techniques invented to speed up the training [8], fundamentally, the slow cooling procedure has to stay for the final convergence.

To overcome these problems, many models have been proposed [5, 6, 9, 10, 11, 12, 13, 14]. Some models do not require that the predetermined mapping space be 1D or 2D. Some models have a fixed number of nodes while others allow the number of nodes to grow. Some [13, 14] model the lateral connections (including lateral inhibition) more faithfully to the biological reality than does the original SOM. Though not very heavily used within the machine learning and data mining communities, lateral inhibition mechanisms are indispensable in understanding the complexity of our visual cortex and simulating its functionality artificially [13, 14, 15].

This research explores the potential of explicitly applying lateral inhibition to form a self-organizing network. We demonstrate that our model, which is part of a more comprehensive architecture, has the potential to overcome many drawbacks of other related models. In this paper, we concentrate only on the training of receptive fields of the neurons in the network using a more biologically plausible neighborhood function. Different from models presented in [13, 14], our model can be applied to arbitrarily high dimensions. To test the efficacy of our algorithm, we have applied it to clustering problem using specially designed

0-7803-7508-4/02/$17.00 ©2002 IEEE

artificial data sets. Section 2 briefly reviews the computational models previously used for self-organizing networks, and identifies their common features. This naturally leads to the understanding of our model described in Section 3. In section 4, we describe the experiments. Section 5 presents the experimental results and discusses the implications. In Section 6, we conclude and discuss future work.

2 Review of other models

There have been many self-organizing network models proposed. Most of the models belong to the soft competitive learning class of models (a.k.a. "winner take most") [16]. During learning, they allow the winning neuron to cooperate with its local neighbor neurons while competing with non-neighbors. These models can be categorized into networks with or without fixed dimensionality [16]. Networks with fixed dimensionality usually allocate neurons on a 1D or 2D grid. These grids can be different shapes, such as rectangular or triangular. Examples of such networks are Kohonen's SOM map, growing grids, and growing cell structures (GCS) [10]. The training of these networks, whose neighborhoods have a preset topology, involves only the receptive fields of the neurons. Thus, they all share the common problem of topology preservation with high dimensionality data.

For networks without fixed dimensionality, the neighborhood of each neuron is formed dynamically during the training process. Topology preservation is a natural property of this type of network [12]. Examples of such networks are neural gas (NG) and topology representing network (TRN) and growing neural gas (GNG) [5, 9, 12]. The training of such networks includes training both the receptive fields of the neurons and the strength of the lateral connections between neurons. The lateral connections are used to dynamically determine the neighborhood for each neuron.

Comparing all these models, we notice that the fundamental difference among these models is how the neighborhood function is defined. In the following, we summarize the models while ignoring some finer details, which can be found elsewhere [16].

1. Initialize the set Nodes containing N neurons u_i, Nodes={ u_1, u_2, ..., u_N,}. Each u_i is represented by a receptive field weight vector $w_{Ui} \in R^n$, w_{Ui} has the same dimension as the input space. w_{Ui} is initialized with some small random value.
2. Present an input pattern x at random according to its probability distribution p(x) to the network.
3. Find the best matching unit (BMU) u_b, such that it satisfy $b(x)=\text{argmin}_{U \in Nodes} \text{dist}(x, w_U)$.
4. Determine the proper form of neighborhood function, $h_{UiUb}(x, t)$,
   - Case1. For conventional SOM and growing grids [11], $h_{UiUb}(x, t)=\exp(-\text{dist}_{L1}(u_i-u_b)^2/2\sigma^2)$, $\text{dist}_{L1}(.)$ is the Manhattan distance function. The span of the neighborhood, σ, is slowly shrinking with time.
   - Case 2. For NG and TRN [6, 9], an ascending rank of Euclidean distance from input x to all units u_i is determined. Each unit is associated with a unique rank $k_i(x, t)$ for input x at time t. $h_{UiUb}(x, t)=\exp(-(k_i-k_b)/\sigma)$, k_b is the ranking of BMU which is 1 in this case. σ is slowly shrinking with time.
   - Case 3. For GCS and GNG models [10, 12], the neighborhood is determined by the existence of lateral connections between units. $h_{UiUb}(x, t)$ is 1 if u_i is the second nearest unit to x, otherwise 0.
5. Weight update of w_{Ui} by $\Delta w_{Ui}=lr* h_{UiUb}(x, t)*(x-w_{Ui})$. The learning rate, lr, usually decreases monotonically with time. For case 3 of above, authors differentiate lr for BMU and its direct neighbors, i.e., allowing BMU to have larger lr than those of its direct neighbors.
6. The data set is presented repeatedly to the network until maximum epoch is reached.

For all these models, they all require a slow decreasing of lr or σ or both. Their time complexities can be estimated as O(epochs)*O(n), n is the size of the data. Often O(epochs)=O(n), therefore the overall time complexity is $O(n^2)$.

Ideally, the neighborhood function, H, should be defined dynamically through lateral connections, and its shape should be a Mexican hat [17]. In case 3, H is constructed dynamically but in discrete form (0 or 1), which is not ideal.

3 The SOLI model

Our model adopts the strategy of non-fixed dimension similar to TRN and GNG for topology preserving purposes. We define our neighborhood function to be a Mexican hat function. A Mexican hat function can be implemented in many ways [17]. In the computer vision community, the Mexican hat function is implemented by difference-of-Gaussians (DoGs) and has been used intensively for feature extraction, forming topographic maps [15, 18].

In our implementation, we define the neighborhood function to be, $h_{Ui}(x, u_i)=\exp(-\text{dist}(x, u_i)^2/2\sigma^2)-\exp(-\text{dist}(x, u_i)^2/2(\lambda\sigma)^2))/\lambda$. λ is a scalar factor, together with σ controlling the range between excitation and inhibition. $\lambda \in R$, and $\lambda > 1$. With λ approaching 1, the inhibition is concentrated tightly around the excitation area. When λ takes a larger value, the inhibition is spread out more globally. For extremely large values of λ, the global inhibition is approximately uniform outside the excitation area. Figure 1 shows the general effects of λ.

The first two steps for training of the SOLI model are the same as in other models, as describe above. Training of our model takes the following steps:

1. Initialize the set Nodes containing N neurons u_i, Nodes={ u_1, u_2, ..., u_N,}. Each u_i is represented by a

receptive field weight vector $w_{Ui} \in R^n$, w_{Ui} has the same dimension as the input space. w_{Ui} is initialized with some small random value.

2. Present an input pattern x at random according to its probability distribution $p(x)$ to the network.
3. For each unit, calculate its distance to x in Euclidean norm.
4. Calculate the neighborhood function, $h_{Ui}(x, u_i)$ as above.
5. Weight update of w_{Ui} by $\Delta w_{Ui}=lr^* h_{Ui}(x, u_i)^*(x-w_{Ui})$, $0<lr<1$.

Continue training the network until converge, i.e, relative weight changes compare to those in previous epoch is reduced to very small. No predetermined maximum epochs is required.

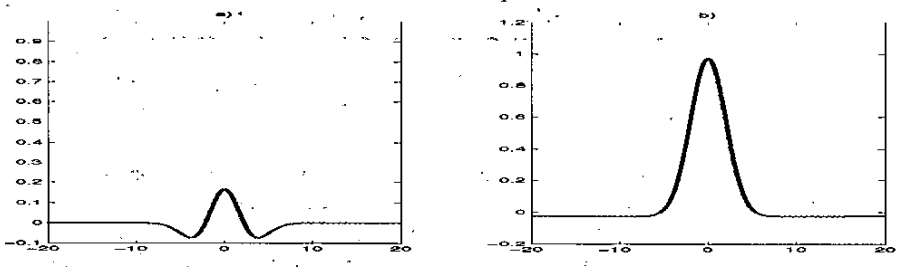

Fig.1. The Mexican hat function. σ is taken as 2 for both graphs while for the graph on the left λ is 1.2 and for the graph on the right λ is 40.

4 Experimental setup

In this paper, we concentrate only on a clustering problem using artificial data. Any robust clustering algorithm should be able to handle data sets with different shapes and different density distributions. Therefore, for these experiments, we designed artificial data sets with different shapes. We also designed data clusters with different density distributions and density levels. For these experiments, we constrained ourselves to at most 3D data. Table 1 tabulates the feature of the data set and the network parameters used for training.

Table 1. Data description and network training parameters

	Set1	Set2	Set3	Set4	Set5	Set6	Set7
|data|	3500	4200	3534	3534	4000	6000	4000
|Nodes|	300	300	225	225	400	400	250
lr	0.5	0.5	0.5	0.5	0.5	0.5	0.5
σ	0.25	0.15	0.05	0.05	0.15	0.15	0.5
λ	2	2	2	2	2	2	2

5 Results and discussion

We presented each data set to the network for training. After network convergence, we expect a well-differentiated activation behavior among neurons. We expect the most activated neurons to be located in the data dense areas and those neurons having the lowest activations to be in data sparse areas. In the following figures, the original data is plotted as grey area. The neurons are plotted as '+' (plus), 'o' (circle), ' ' (square), and '*' (star) in the order of ascending activation level.

5.1 Analysis of Results

All of the resulting maps are in Figures 2 through 8 in the Appendix to this paper. Data sets 1 and 2 (Figures 2 and 3) are designed to test the applicability of the algorithm to data with arbitrarily complex shapes. It is obvious that the most activated neurons are located in the data dense areas. Their physical locations naturally follow the data distribution, thus the arbitrary orientations and shapes.

Data sets 3 and 4 are designed to demonstrate the sensitivity of the SOLI algorithm to different density distributions. For set 3 (Figure 4), the neurons are more or less uniformly distributed in the data area, following the uniform distribution of the original data set. Set 4 data follows a Gaussian distribution and shows that the neuron activations declined with the distance further away from the Gaussian center (Figure 5). Neurons with distinguishably high activations are located in the Gaussian center.

Data sets 5 and 6 were designed to test the sensitivity of SOLI to different density levels. Set 5 is composed of two identical Gaussians with the same density. Figure 6 shows that the most activated neurons converge to the two Gaussian centers with the same activation level. Set 6 is composed of 2 Gaussians with a density ratio 1:5. Figure 7 shows 3 distinguished levels of neuronal activation. The

two highest activated levels of neurons are located at the centers of the two Gaussians, with activation ratio approximating 1:5. Clearly, the new algorithm is able to identify different density levels.

We tested our algorithm on a higher dimensional data set, set 7, a 3D ring band. Figure 8 illustrates that our algorithm can be easily extended to higher dimensional data set with its full capability.

From Table 2, we see that convergence of our algorithm is linear in the number of epochs. Thus, the time complexity of our algorithm can be estimated as O(n), where n is the size of the data. This is an improvement over other models.

Table 2. Network training results

	Set1	Set2	Set3	Set4	Set5	Set6	Set7
|Epochs|	50	5	10	50	15	25	5
CpuTime(s)	299.9	32.2	56.5	307.8	163.3	339.5	47.0
Effective Nodes	109	160	176	162	354	260	178

We notice that our network often starts with a large number of nodes but, after training, only some of the nodes have been used to represent the data distribution. Among these remaining nodes, many converge onto even fewer nodes (i.e., onto a few physical locations in the input space). This is generally true for all our data sets, as can be seen by comparing the number of nodes in Table 1 and the number of effective nodes of Table 2 for each data set respectively. This implies that, by allowing neurons to merge during training, the size of the network can be reduced dynamically, resulting in less computation. This also indicates that the initial size of the network is not a crucial factor for the quality of the final network, as long as the initial network has enough nodes.

Similar to TRN and GNG, our network has no fixed dimensionality. Though at this moment, we have not yet included the training of the strength of lateral connections, our network naturally shares the topology preserving property of these networks.

We observe that, after training, there is an activation dead zone (a "white" rim on the figures around the data dense area), which clearly separates activated neurons from those non-activated ones. This "dead zone" (an activation gap) draws a natural boundary between data dense areas and data sparse areas. This property of the new algorithm can be applied for novelty detection purposes.

6 Conclusions and future work

By combining promising properties of different models of self-organizing networks, our SOLI model is able to overcome many drawbacks of other models. As illustrated on the clustering problem, SOLI is well suited for unsupervised learning applications. Our model is computationally efficient with time complexity of O(n), a great improvement over previous models with time complexities of $O(n^2)$. SOLI belongs to the class of networks without fixed dimension, and thus has the potential to preserve topology. Our model can be extended easily to a more dynamic version with network size evolving during the training process. Our model can also be used for novelty detection.

Unlike similar models, the performance of our model is not heavily affected by the initial network parameter settings, with the exception σ. For our model, the only parameter that is influential is σ and has to be chosen with care. In the future, we will try to automatically configure its value in a self-correcting manner. We will conduct more experiments on some benchmark data, and compare its performance with those of other models. We will add training of lateral connections to the current model. The correlation and de-correlation of neuron activity will be explored in searching for more abstract forms of knowledge representation. Eventually, the whole model will be applied to text mining tasks.

While these are only initial results, we are very encouraged by the performance of the SOLI model.

Acknowledgements

We would like to thank Thomas Trappenberg and Malcolm Heywood for their useful comments and suggestions.

References

[1] S.-I. Amari, Togographic organization of nerve fields, Bulletin of Mathematical Biology, vol 42, pp. 339-364. 1980.

[2] D. J. Willshaw and C. von der Malsburg, How patterned neural connections can be set up by self-organization, Proceedings of the Royal Society of London, Series B 194, pp. 431-445, 1976.

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

[4] T. Kohonen, Self-Organizing Maps, Springer Series in Information Sciences, Vol 30. Springer, Berlin, Heidelberg, New York, Third Edition. 2001.

[5] T. Martinetz and K. Schulten, Topology representing networks, Neural Networks, Vol 7, no. 3, pp. 507-522, 1994.

[6] T. Villmann, R. Der, M. Herrmann and T. M. Martinetz, Topology preservation in self-organizing feature maps: extract definition and measurement, IEEE Trans. Neural Networks, Vol 8, no. 2, pp. 256-266, 1997.

[7] E. Erwin, K. Obermayer, and K. Schulten, Self-organizing maps: Stationary states, metastability and convergence rate, Biological Cybernetics, Vol 67, pp. 35-45, 1992.

[8] T. Kohonen, S. Kaski, K. Lagus, J. Salojärvi, J. Honkela, V. Paatero and A. Saarela, Self organization of a massive text document collection, IEEE Transactions on Neural Networks, Special Issue on Neural Networks for Data Mining and Knowledge Discovery, Vol 11, no. 3, pp. 574-585. 2000.

[9] T. M. Martinetz, S. G. Berkovich and K. Schulten, "Neural –Gas" network for vector quantization and its application to time-series prediction. IEEE Trans. Neural Networks. Vol. 4. No, 4, pp. 558-569, 1993.

[10] B. Fritzke, Growing cell structures - a self-organizing network for unsupervised and supervised learning, Neural Networks, Vol. 7, No. 9, pp. 1441-1460,1994.

[11] B. Fritzke, Growing Grid - a self-organizing network with constant neighborhood range and adaptation strength, Neural Processing Letters, Vol.2, No. 5, pp. 9-13, (1995).

[12] B. Fritzke, A growing neural gas network learns topologies, in Advances in Neural Information Proceesing System 7, G. Tesauro, D. S. Touretzky, and T. K. Leen Eds. Cambridge, MA: MITPress, 1995, pp. 625-632, 1995.

[13] R. Miikkulainen, Self-Organizing Process Based on Lateral Inhibition and Synaptic Resource Redistribution. In T.Kohonen, K. Makisara, Olli Simula and Jari Kangas Eds. Proceedings of the International Conference on Artificial Neural Networks (ICANN-91, Espoo, Finland), pp. 415-420. New York: Elsevier, 1991.

[14] J. Sirosh and R. Miikkulainen, Topographic Receptive Fields and Patterned Lateral Interaction in a Self-Organizing Model of the Primary Visual Cortex. Neural Computation, Vol 9, pp. 577-594, 1997.

[15] M. Pötzsch and C. von der Malsburg, Self-organization of networks in the visual system, In G. Dorffner, K. Möller, G. Paaß, and S. Vogel, Eds, Konnektionismus und Neuronale Netze, vol 272 of GMD-Studien, pp. 299-315, Münster/Westf., 1995.

[16] B. Fritzke, Some competitive learning methods, Draft Doc., http://www.neurolin-formatik.ruhr-unibochum.de/ini/VDM/research/gsn/DemoGNG, 1998.

[17] S. Haykin, Neural networks, a comprehensive foundation. Macmillan College Publishing Company, New York, pp. 404-408. 1994.

[18] L. Itti, C. Koch, A Comparison of Feature Combination Strategies for Saliency-Based Visual Attention Systems. SPIE Human Vision and Electronic Imaging IV (HVEI'99), San Jose, CA, Vol, 3644, pp. 373-382, 1999.

Appendix

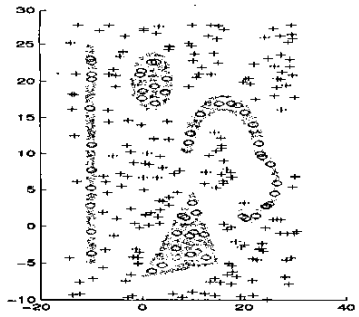

Fig.2. Set 1 – arbitrary shapes.

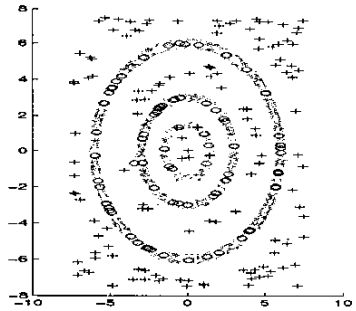

Fig.3. Set 2 - 3 rings with different radii.

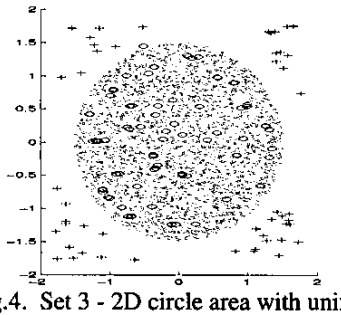

Fig.4. Set 3 - 2D circle area with uniform distribution.

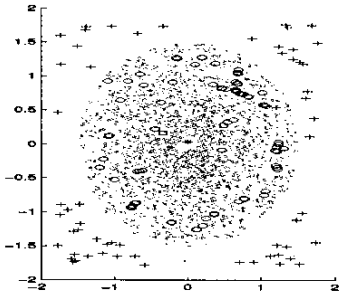

Fig.5. Set 4 - circle with Gaussian distribution.

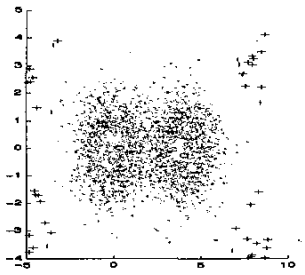

Fig.6. Set 5 - Gaussians with same density.

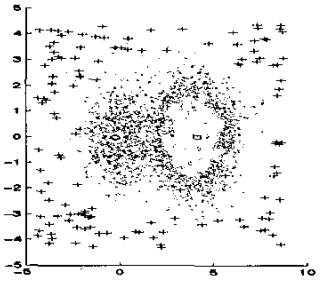

Fig. 7. Set 6 - two Gaussians with density ratio of 1:5.

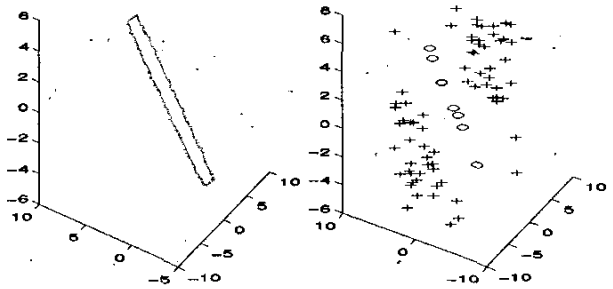

Fig.8. Set 7. Left - side view of 3D ring data. Right - Neuron plot with same view angle. Neurons are located along the ring.