Proceedings of International Joint Conference on Neural Networks, Atlanta, Georgia, USA, June 14-19, 2009

Growing Mechanisms and Cluster Identification with TurSOM

Derek Beaton, Iren Valova, Dan MacLean

***Abstract*—TurSOM [1] is a novel self-organizing map algorithm with the capability of connection reorganization, not just neuron reorganization. This behavior facilitates the ability to map distinct patterns in a given input space. Multiple networks exist, and operate independently. This work presents an application driven approach, based on the theoretical and empirical work of previous TurSOM experiments. TurSOM is a highly robust algorithm, designed to eliminate the need for post processing methods of cluster identification using SOM algorithms.**

One of the applications TurSOM is suitable for, but obviously not limited to, is image segmentation, as it is demonstrated in this work.

I. Introduction

TurSOM is a novel SOM variant algorithm that is designed for cluster identification and pattern recognition during execution, rather than utilize post-processing methods. TurSOM is a unique update to the traditional one-dimensional self-organizing map, introduced by Kohonen in the early 1980s. TurSOM is a one-dimensional network utilizing connection reorganization, which facilitates multiple one-dimensional networks in a single input space. Previous work on TurSOM was primarily theoretical and driven by experiments – to demonstrate the effectiveness of connection reorganization, and the use of multiple networks in a single input space. This work presents application-driven features and discussions on new methods introduced to TurSOM, which include network growing and initialization techniques.

II. Background

TurSOM draws its inspiration from the works of Alan Turing's Unorganized Machines (TUM) [2], Grossberg and Carpenter's Adaptive Resonance Theory (ART) [3, 4], and Kohonen's Self-organizing feature Maps (SOM) [5]. TurSOM is an unsupervised SOM algorithm that adheres to the criteria of competitive learning. TurSOM introduces new, and more robust methods of self-organization by allowing connections to reorganize, a la TUM. One of the major contributions of TurSOM to the field is the elimination of post-processing techniques that are used for severing SOM networks to determine clusters, or classes. Small SOMs can be viewed as *k*-means or non-linear PCA, however, large networks may have many neurons within the same topological space, making it necessary to group neurons, and effectively, the input they represent, into classes.

This paper introduces new features to TurSOM, adding to the robustness and effectiveness as a clustering or image segmentation method. The primary focus on TurSOM that is discussed here is network growth, and, implicitly, network pruning.

One of the major contributors to the field of SOMs, and, most importantly, growing SOMs, is provided by Fritzke. Fritzke introduced several growing algorithms and variants, including growing neural gas (GNG), growing cells (GC), and growing grids (GG). Since Fritzke's models, several other models have been introduced, including SOINN (ESOINN) and ParaSOM.

The primary difference between TurSOM's growing mechanisms and all other growing algorithms, is that TurSOM begins as a one-dimensional network of a static size, which, after one-dimensional convergence, grows all one-dimensional chains in the input space into two-dimensional square grids. The grid size is $n \times n$, where n is the size of the one-dimensional chain, post convergence. The majority of well-known growing algorithms exhibit a growing behavior during the entire execution of the algorithm. Some algorithms, such as ParaSOM, remain one-dimensional, whereas others are two-dimensional during execution. Yet, others are considered *k*-dimensional, like ESOINN, GNG, GC, because they form *k*-simplexes, where k is the number of connections for the neuron with the most connections in the network.

III. Growing Networks

A. Growing Neural Gas, Cells, and Grids

Growing grids (GG), growing cells (GC) and growing neural gas (GNG) are three architectures introduced by Fritzke [6-11], in the 1990's that are significant variants of Kohonen's SOM. Through GG, GC, and GNG, a growing methodology is introduced to SOM, which, in its classic version is a static size network. Fritzke's new algorithms start with a minimal amount of neurons, and grow neurons during execution. Every neuron in GG, GC, and GNG architectures includes an error parameter. It determines

Manuscript received January 4, 2009

Derek Beaton is a research associate with the James J Kaput Center for Research and Innovation in Mathematics Education, University of Massachusetts Dartmouth, 200 Mill Rd., Suite 150B, Fairhaven, MA 02719 (u_dbeaton@umassd.edu).

Iren Valova is with the department of Computer and Information Science, University of Massachusett Dartmouth, 285 Old Westport Rd, North Dartmouth, MA 02747, (corresponding author, phone 508 999 8502; fax 508 999 9144, ivalova@umassd.edu).

Daniel MacLean has graduated with an MS degree from Computer and Information Science Department, University of Massachusett Dartmouth, 285 Old Westport Rd, North Dartmouth, MA 02747, (dmaclean82@gmail.com).

which neuron is selected to have a new neuron (or set of neurons) inserted nearby.

1) Growing Cells

In GC, the dimensionality of the network is always *k*, where *k* is the maximum number of connections for a neuron in the network. Neurons may have a different number of neighbors. Neurons and their connections produce *k*-simplex structures.

Age of the network determines when a new neuron is introduced. At a user-determined integer multiple of the intended number of iterations of the GC network, the neuron with the highest error value is selected. The longest edge from this neuron to its furthest neighbor is replaced by two edges, and a new neuron is inserted exactly in the center of these two neurons.

2) Growing Grids

Similar to GC, neurons in this network have an age parameter. GG utilizes rectangular grid pattern, where each neuron has at most four neighbors. This is maintained through the duration of execution.

Each time a best matching unit (BMU) is selected for an input, a winner counter is incremented for that neuron. At given intervals, similar to that of GC, the neuron with the highest winning value is selected. The furthest direct topological neighbor from the selected neuron is also found. If these two neurons are in the same row of neurons, a column will be added between the two – thus affecting neurons in other rows. If the selected neurons are in the same column, then a new row will be added between them – thus affecting neurons in other columns.

3) Growing Neural Gas

Growing neural gas (GNG) is another growing SOM-based algorithm proposed and implemented by Fritzke [11,12]. GNG operates similarly to GC, with the exception that connections (edges, as Fritzke refers to them), and not neurons, are given an age parameter. GNG has the capability of adding and removing connections. However, these connections are added or removed based on neuron behavior and selection. GNG can achieve a "multiple network" configuration, by deleting edges that become too old.

B. ESOINN

The enhanced self-organizing incremental neural network (ESOINN) is a fairly recent architecture introduced by Furao et al [13], as an improvement to SOINN. ESOINN architecture can be characterized as an enhancement of Fritzke's GNG architecture. During execution, ESOINN, and similarly SOINN, add or remove connections based on criteria of density and age of nodes and connections. After a convergence of the network(s), ESOINN enters a growing phase where new neurons can be introduced, if necessary, based on new input being presented.

ESOINN achieves this behavior by finding the best matching unit, and the second best matching unit (2BMU), when presented with input.

One of the primary features that sets ESOINN appart is incremental learning. ESOINN adapts to data without disregarding previous knowledge.

Connection building and removal in ESOINN is dependent on density information provided by the nodes in the network. This behavior occurs when the BMU and 2BMU are either in the same subclass, or if one of the nodes is a newly grown node, introduced by growing mechanisms. In the final stages of the network, nodes are reclassified for optimal class identification.

ESOINN has addressed issues of machine vision and clustering in complex patterns such as hand-written digits, feature vectors of faces and overlapping Gaussian distributions.

C. ParaSOM

ParaSOM [14] is a unique architecture that utilizes growing mechanisms and parallelism. Another unique feature of ParaSOM is that neurons have a cover region, which is a Gaussian degrading responsibility for all neurons under the cover region. The most important feature of ParaSOM is the parallelism. ParaSOM still adheres to competitive learning, but in this architecture, every neuron wins a subset of input that is different from all other neurons. Minimal overlap does occur, where the cover region is weak (tail ends of Gaussian distribution). Kohonen's original algorithm is inherently prepared for parallelization of neuron computation – just as the brain works. Neurons work in a parallel distributed processing method. Parallelism is of significant concern, and very beneficial to ANN algorithms, as it provides a truer model of the brain, and computation can time can be significantly increased.

IV. TurSOM

TurSOM implements two methods of self-organization:

- Neurons organize with respect to input
- Connections organize with respect to neurons

Neurons in TurSOM behave just as they do in traditional models of the SOM network. Neurons move toward input based on competitive selection, and influence their neighbors within a given region. Neurons also adhere to a learning rate, to decrease the amount of movement as the network adapts to input. Connections still play the traditional role of determining neuron neighbor influence and effect. However, connections now have the ability to reorganize. Reorganization entails:

- Disconnection
- Reconnection
- Switching neighbors

The employment of connection reorganization has empirically shown that convergence of the network(s) occurs in significantly fewer iterations than in traditional models; hundreds or thousands of iterations compared to tens or hundreds of thousands of iterations. TurSOM also does not suffer a drawback that currently has no immediate during-execution fix for traditional SOM algorithms: network tangling. One-dimensional and two-dimensional SOM networks are susceptible to connections tangling and crossing. One-dimensional networks can eventually unravel, but this is not guaranteed with two-dimensional networks.

The initial form of TurSOM is a one-dimensional network that reorganizes in order to converge in a Peano-like curve – that is – no tangling of the network.

A. Neurons

TurSOM introduces new mechanisms to the traditional model, so that it may perform in the way described. Neurons now have neuron responsibility regions (NRR) defined by the following:

If attribute dimensionality is even, NRR is calculated by:

$$r_e = \left[\rho/e\right]^{1/\delta} \tag{1}$$

where e is:

$$e = \left(\frac{\delta}{2}!\right)^{-1} \times \pi^{\frac{\delta}{2}} \tag{2}$$

If attribute dimensionality is odd, NRR is calculated by:

$$r_o = \left[\rho/o\right]^{1/\delta} \tag{3}$$

where o is:

$$o = \left(\frac{2^{\delta} \times \frac{\delta-1}{2}!}{\delta!}\right) \times \pi^{\frac{\delta-1}{2}} \tag{4}$$

Where δ represents the number of dimensions, and ρ represents the number of theoretical inputs a neuron is responsible for. NRR exists for neurons that do not have two neighbors. This facilitates a "feeler" for a neuron with one connection to recognize another neuron with only one connection.

B. Connections

Connections determine survivability based on a modified version of the standard outlier formula, called the connection learning rate (CLR):

$$CLR = Q_3 + (i \times (Q_3 - Q_1)) \tag{5}$$

The CLR, similar to a neuron learning rate, is a moving value. Neuron learning rates decrease as networks progress, to slow the movement of neurons, whereas the CLR increases, to slow the removal of connections between neurons. Connections also observe connections that are in close proximity. This connection-connection observation is employed to switch connections between neurons, in the event that two non-neighbor neurons are topologically closer than two neighboring neurons. Essentially, connections monitor and adapt the organization of the network, and neurons are still, traditionally responsible for representation of data in the network. Connections in TurSOM are the key components to highly effective self-organization.

Connection behaviors are monitored by a mechanism referred to as the gap junction (henceforth "GJ"). For a more detailed explanation of the features, and algorithmic steps of TurSOM, please see the complementary paper to this one [19].

Figure 1 presents an example of one-dimensional TurSOM displaying its natural disconnecting behavior for a rigid form of a benchmark pattern, the double spiral. Figure 1 demonstrates TurSOM mid-execution, while sub-networks are still adapting to the input space, and to one another.

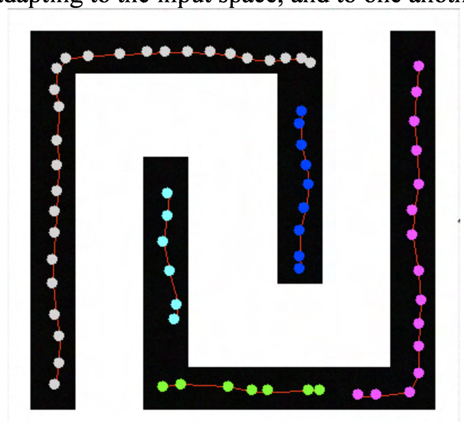


Figure 1: A mid-execution screenshot of TurSOM displaying four separate networks in a single input space, with two distinct patterns.

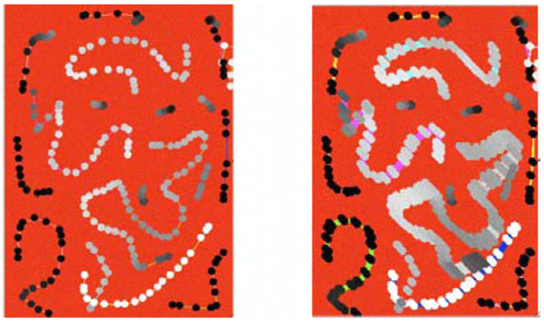
V. Growing Mechanisms

The growing mechanism of TurSOM is entirely dependent on its one-dimensional network, at the time of convergence. As the neuron learning and connection learning rates update during execution, one-dimensional TurSOM should reach an optimal mapping. At convergence (presumed optimal mapping), TurSOM would have found distinct patterns in input space. All sub-networks at this point would be subject to a new mechanism called spontaneous growth (SG). SG transforms all one-dimensional sub-networks into two-dimensional square grids. Two methods of neuron introduction can occur at this point:

- new row and column neurons are introduced with the exact same vectors as the previous chain or
- new row and column vectors are introduced with very small incremental change for only one dimension for either rows or columns

This produces a dense square grid network either completely overlapping, or very close. Effectively, the neurons in this case are being positioned in a tangle-free method. The examples of SG in these experiments are incremental positive change. This can be noted in Figures 2a and 2b, as the grids are positioned relative to the chains.

Figure 2 displays multiple networks in a single input space at the time of convergence for TurSOM one iteration prior to SG (2a) and one iteration post SG (2b).



(a) (b)

Figure 2: a) Input space with one-dimensional chains; b) One iteration after spontaneous growth.

VI. PRUNING

Often accompanying growing mechanisms in SOMs are methods of pruning. Overgrowing, or overfitting (as well as underfitting), leads to poor representation of input. When growing new neurons, we want to ensure that we do not grow too many.

Pruning methods for TurSOM are simple, and occur prior to SG. Empirically, pruning was deemed unnecessary, due to the initialization methods used for TurSOM. TurSOM is initialized based on positioning the neurons along a Hilbert curve [15, 16, 17].

The pruning mechanism of TurSOM occurs between one-dimensional convergence, and two-dimensional growth. Similar to Fritzke's models, neurons maintain an age value. However, a major difference is that no error value is calculated. The age value is considered a survivability value – similar to connection maintenance during one-dimensional execution – and is used to determine the best neurons to move from one-dimensional generation to a two-dimensional generation (survival of the fittest), based on the number of times it was selected as the best representation of input. All neuron ages in a sub-network are averaged, and all neurons whose age falls below the average are removed prior to growth.

VII. INITIALIZATION TECHNIQUES

The initialization techniques used here are traditional methods that include random vector initialization, and techniques that have been developed and explored in our previous SOM work, specifically designed for one-dimensional SOMs: Hilbert curve initialization [15, 16, 17]. Hilbert curves are a subset of Peano curves. They are space filling curves, and they are used for one-dimensional SOM networks to fill possible input vector space.

VIII. EXPERIMENTS

A. *Gray Scale Portrait*

The following experiment is performed on a gray scale image (314 x 400 pixels), of Hollywood actor, Christian Bale [18]. The image shown in Figure 3 is interpreted as a three-dimensional input space (x, y and pixel intensity). Initially, a one-dimensional TurSOM is used to find distinct patterns in the entire input space.

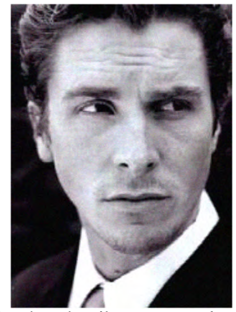


Figure 3: Thumbnail representation of input

Figures 4 - 10a demonstrate the progress TurSOM goes through, including its growing mechanisms.

Hilbert initialization is used here, with a modification: the z, or intensity attribute, is on a degrading scale from black to white (256 gray-scale). TurSOM also has the capability to delay connection reorganization. It is not required to begin when the entire network begins to adapt. The immediate and delayed stages of connection reorganization are referred to as neuron adaptation phase (NAP) and connection adaptation phase (CAP). NAP should be considered the traditional SOM behavior, whilst CAP is a new phase that allows connections to reorganize. CAP may occur simultaneous with NAP – it effectively means that connections are reorganizing from initialization, rather than a delayed effect of letting neurons "settle". Figures 5 and 6 display the delay of connection reorganization.

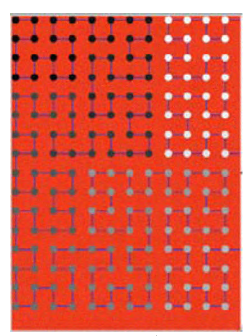


Figure 4: 0 Iterations. A depth degrading Hilbert curve is being used.

Figure 5: 250 Iterations; NAP with training

Figure 6: 750 Iterations; CAP initialization

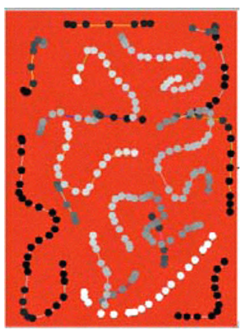


Figure 7: 1000 Iterations; Separate networks find distinct patterns and image segmentation occurs

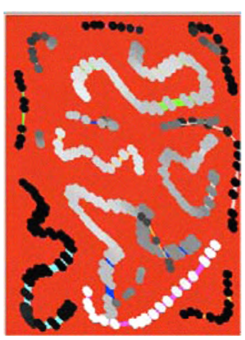


Figure 8: 4000 Iterations; Network growth begins; 15 networks exist.

TurSOM breaks apart a single network into multiple networks that are representing distinctly different patterns. The behavior of each sub-network is independent from all other networks, and is like a localized one-dimensional SOM. However, if a GJ deems two connections to be topologically close, in which case, two localized networks will combine to form a single, larger network.

Figure 7 displays the effective final positions of sub-networks in TurSOM. Little change occurs between these iterations and the first iteration of spontaneous growth, which is demonstrated in Figure 8.

Figure 9 displays the two-dimensional network adaptation, after just a mere 500 iterations after SG occurred. The original image is nearly restored, accounting for loss of "resolution" because there are less neurons than pixels. No network rejoining occurs here, because connection reorganization stops as soon as SG occurs.

Figure 9: 4500 Iterations; Grid networks become responsible for large areas of similar input

The original image is nearly "restored" in just 5500 iterations as shown in Figure 10a. The original in Figure 10b is repeated for comparative purposes. Distinct networks exist for the jacket/background on either side of the face; a network for the white collar; separate networks for the eye regions; and even a network to represent shadow and facial hair on the right side of the image.

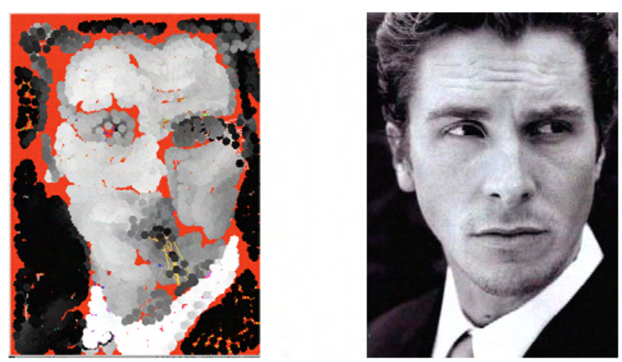


Figure 10a and 10b: 5500 Iterations; Final representation of with 15 distinct networks & the original image.

B. *Structural MRI*

The intent of this experiment is to demonstrate segmentation of structural MRI images. Structural MRI can provide information about areas of the brain that may experience some effect, such as degeneration. Using volumetric information from MRI slices can provide this kind of information. This experiment presents segmentation of MRI with TurSOM. Here, the same slice (number 9 of 16) is selected for two different subjects, to further exemplify the applicability of TurSOM. Figure 11 presents the first MRI.

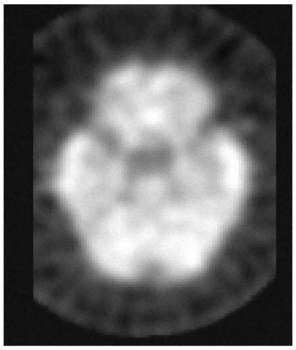


Figure 11: Subject one. This slice is slice (9) in a 16-slice MRI.

The images are 128 by 128 pixels, with 128 gray scale. However, a threshold value has been set to remove background and noise pixels. This threshold value was set at 6. This removes all the black box pixels around brain slice image. One hundred neurons are used, with random vector initialization.

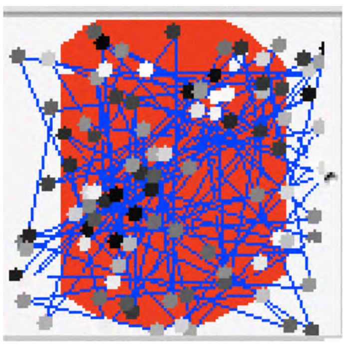


Figure 12: Patient 2 at iteration 0.

Figure 12 shows TurSOM at iteration 0, with random initialization. Figure 13 shows TurSOM at iteration 750, just before the single network breaks into multiple networks.

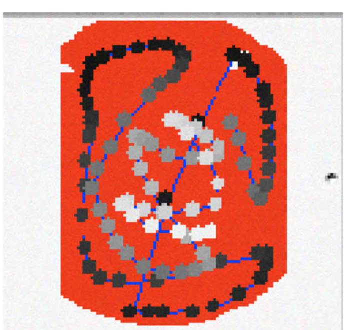


Figure 13: Patient 2 at iteration 750.

Figure 14 shows TurSOM with multiple networks just before spontaneous growth (SG). Figure 15 demonstrates TurSOM with SG 50 iterations after two-dimensional networks appear.

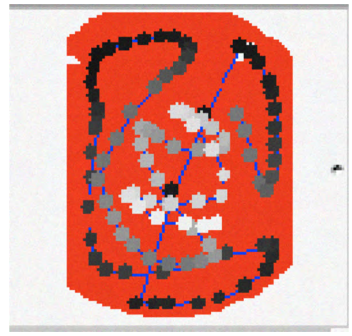


Figure 14: Patient 2 at iteration 3950, which is 50 iterations before spontaneous growth.

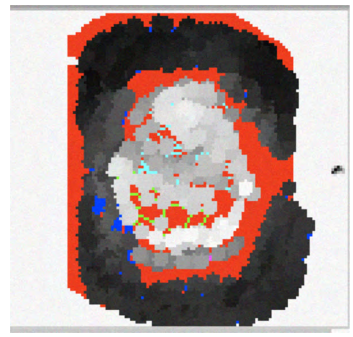


Figure 15: SG after 50 iterations

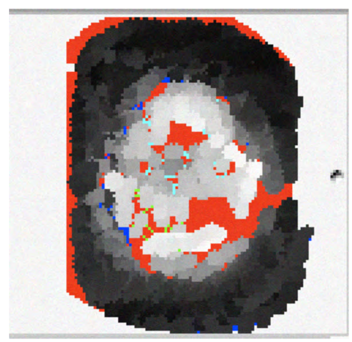


Figure 16: The final mapping of TurSOM with SG. 5 networks exist.

Figure 16 demonstrates the final mapping of multiple networks for patient 2. Note that a single network encompasses the dark gray-black region around the outside, where as separate networks on the interior handle white to light gray scale sections. Figures 17 - 22 represent TurSOM processing a second subject's MRI. The same slice (9) out of the 16 slices is used.

The same experimental process is used here for subject 1. The process is documented in Figure 18 at iteration 0, through iteration 750 (Figure 19), followed by multiple networks at 3950 iterations in Figure 20; Figure 21 shows the status of the network 50 iterations after the occurrence of spontaneous growth. Figure 22 illustrates the converged map with 5 networks with two of them covering the outside darker regions and the remaining 3 covering the white to light gray regions on the inside.

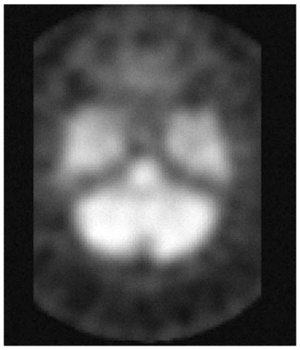


Figure 17: Subject 2, slice 9 of 16

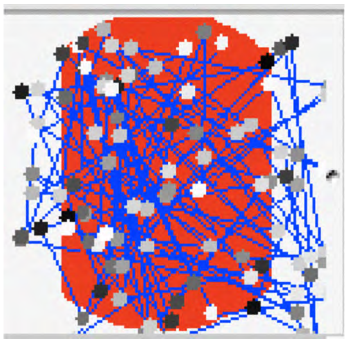


Figure 18: Random initialization with 100 neurons


Figure 19: 750 Iterations into execution, just before the single network breaks into multiple networks.

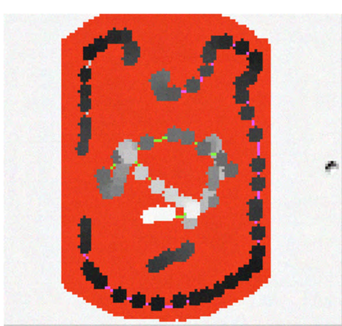


Figure 20: Iteration 3950, multiple networks exist. This is 50 iterations before spontaneous growth.

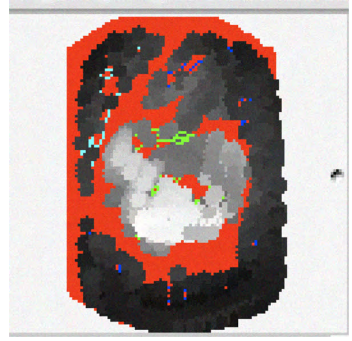


Figure 21: 50 iterations after SG.

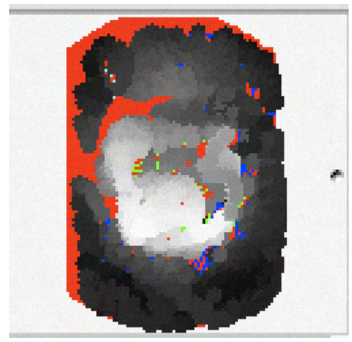


Figure 22: Final mapping of multiple networks (5 networks) at 5000 iterations. Two networks cover the outside darker regions, three networks cover the interior.

IX. Conclusions

TurSOM is a powerful, robust and efficient variant of the SOM algorithm. Its theoretical basis is demonstrated to be applicable to a field as complex as image segmentation.

Though TurSOM, or any SOM for that matter, has a much lower resolution than the majority of image segmentation algorithms, SOMs are not often used for 1:1 pixel to neuron ratios. SOMs have the main intent of data representation. One neuron often represents many inputs, in this case, pixels.

Though connection building and deleting is not a new concept for self-organizing neural networks (ESOINN, GNG), the method presented with TurSOM is unique, in that connections are active entities, separate from the neurons and assume responsibility for final neuron organization.

The growing mechanisms applied to TurSOM, inspired by methods created by Fritzke, enhance the standard one-dimensional capability of TurSOM. TurSOM in its one-dimensional form tends to find the center of density of distinct patterns in input space, rapidly.

The rapid execution and completion of one-dimensional TurSOM make it analogous to a secondary initialization stage, finding optimal center-of-mass based placement for small, string networks, which allow grid networks to accurately represent, without topological deformation, the underlying information.

References

[1] D. Beaton, "Bridging Turing Unorganized Machines and Self-organizing Maps for Cognitive Replication", UMass Dartmouth MS Thesis, 2008

[2] D.C. Ince (ed.). *Collected Works of A.M. Turing – Mechanical Intelligence*, Vol. 3/4 Elsevier Science Publishing, 1992.

[3] S. Grossberg. "Competitive Learning: From Interactive Activation to Adaptive Resonance", *Cognitive Science*, Vol. 11, pp 23-63, 1987.

[4] G. Carpenter, S. Grossberg. "ART 3: Hierarchical Search Using Chemical Transmitters in Self-organizing Pattern Recognition Architectures," *Neural Networks,* Vol. 3, pp. 129-152, 1990.

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

[6] B. Fritzke. "Growing cell structures -- a self-organizing network in k dimensions," *Artificial Neural Networks II* , I. Aleksander & J. Taylor, eds., North-Holland, Amsterdam, pp. 1051—1056, 1992.

[7] B. Fritzke. "Growing Cell Structures – A Self-organizing Network for Unsupervised and Supervised Learning," *Neural Networks*, Vol. 7 pp. 1441-1460, 1993.

[8] B. Fritzke. "Supervised Learning with Growing Cell Structures," *Advances in Neural Information Processing Systems*, Vol. 6, 1994.

[9] B. Fritzke. "Kohonen Feature Maps and Growing Cell Structures – a Performance Comparison," *Advances in Neural Information Processing Systems*, Vol. 5, 1993.

[10] B. Fritzke. "Growing Grid – a self-organizing network with constant neighborhood range and adaptation strength," *Neural Processing Letters*, Vol 2 No. 5 9-13 1995.

[11] B. Fritzke. "A Growing Neural Gas Network Learns Topologies," *Adv. In Neural Information Processing Systems*, Vol. 7 pp. 625-632, 1994.

[12] http://www.neuroinformatik.ruhr-uni-bochum.de/ini/VDM/research/gsn/DemoGNG/GNG.html

[13] S. Furao, T. Ogura, O. Hasegawa. "An Enhanced Self-Organizing Incremental Neural Network for Online Unsupervised Learning," *Neural Networks*, Vol. 20 No. 8, pp. 893-903, 2007.

[14] I. Valova, D. Szer, et al. "A parallel growing architecture for self-organizing maps with unsupervized learning," *Neurocomputing*, Vol. 68 pp. 177-195, 2005.

[15] I.Valova, D.Beaton, A.Buer, D.MacLean, "Fractal Initialization for High-Quality Mapping with Self-Organizing Maps," *Journal of Neural Computing and Applications (in review), Springer.*

[16] I. Valova, D. Beaton, D. MacLean, "Role of Initialization in SOM Networks - Study of Self-Similar Curve Topologies", *Proceedings International Conference on Artificial Neural Networks in Engineering (ANNIE), Vol.18, pp 681-688, 2008.*

[17] D.Beaton, I.Valova, D.MacLean, "CQoCO: a Measure for Comparative Quality of Coverage and Organization for Self-Organizing Maps," *Neurocomputing (in review), Elesevier.*

[18] http://sweb.cz/carreen/christian%20bale.jpg

[19] D. Beaton, I. Valova, D. MacLean, "TurSOM: A Turing Inspired Self-organizing Map," accepted to Proceedings of IJCNN, 2009.