

GST Networks: Learning Emergent Spatiotemporal Correlations¹

Chaitanya Tumuluri Chilukuri K. Mohan Alok N. Choudhary

tumuluri@cat.syr.edu

ckmohan@syr.edu

choudhar@cat.syr.edu

121 Link Hall

2-120 CST

121 Link Hall

Syracuse University, Syracuse, NY 13244

Abstract

This paper presents two novel (GIST and GEST) networks, which combine unsupervised feature-extraction and Hebbian learning, for tracking emergent correlations in the evolution of spatiotemporal distributions. The networks were successfully tested on the challenging Data Mapping problem, using an execution driven simulation of their implementation in hardware.

1 Introduction

Several important and difficult problems require tracking emergent spatio-temporal correlations among useful features of time-varying data distributions, where it may not be possible to obtain *a priori* analytical characterization of the processes generating the data. Data Mapping is one such problem where disjoint sets of data must be mapped to processors in a multiprocessor system to minimize off-processor data accesses. This requires (a) modeling emergent distribution of data references and (b) tracking correlations in the distribution of references. We extend Fritzke's unsupervised Growing Cell Structures (GCS) network [1] for these purposes to produce two new networks models: the GCS Instantaneous Spatio-Temporal (GIST) network and the GCS Epochal Spatio-Temporal (GEST) network, together referred to as the GST networks. We demonstrate their success in learning the spatiotemporal dynamics on the challenging Data Mapping problem.

2 GST Spatial Mapping Dynamics

The GST network is a 2-dimensional network of neurons forming planar maps, whose topology always consist of triangles. Figure 1 illustrates the network and highlights the various connectivities among neurons. The lateral weight W_{ij} , connecting neuron j to neuron i , records emergent temporal correlations in their activations. A Signal Receptivity Meter (SRM) records the number of hits to each neuron over an epoch and is an enhancement over the Signal Frequency Counter (SFC) of the GCS network.

The GST spatial dynamics position neurons at cluster centers in the input Signal Vector Space (SVS) during both the training and the production modes of operation. The network uses training

¹This research was supported in part by an NSF Young Investigator Award CCR-9357840.

data to approximately map time-varying distributions in the offline training mode. The map is refined online using actual incoming data during the production mode. Network performance parameters, evaluated at epochal checkpoints, are used to decide if the network needs offline retraining. Both modes involve:

1. Input vector quantization and subsequent input weight (W^I in Figure 1) adaptations as in Kohonen's SOM [2] with the difference that the adaptation rates are fixed (i.e. non-stochastic).
2. The SRMs of the winner neuron *and* all its neighbors are adapted as opposed to the GCS dynamics which adapt the SFC of only the winner neuron.

The GST architectural dynamics define the structural adaptations in the network which only occur under conditions wherein the network-wide average quantization error changes less than 10% across successive epochs. Architectural adaptations under such conditions of positional equilibrium involve:

1. Addition of a neuron, as a topological neighbor of the neuron with the highest average quantization error, to increase map resolution and decrease network-wide average quantization error.
2. Deletion of *one* superfluous neuron with the least relative SRM value, $\tau_i / \sum_j (\tau_j)$, (and any neighbors not part of at least one topological triangle [1]), to coarsen resolution in areas of low input probability.

3 GST Temporal Mapping Dynamics

The production mode temporal dynamics of the GIST and GEST networks involve: **(a)** different lateral weight update schemes but **(b)** the same computations of temporal correlations in neuronal activations. The computed temporal correlations are used to obtain the total probability of activation, P_i , of neuron i . P_i is termed the *Signal Utility* of inputs mapped to this neuron and measures the usefulness of this data-cluster in the temporal evolution of the SVS distribution. The total probability, P_i is given by:

$$P_i = \sum_j (P_{i|j}) \times \hat{p}_j \quad (1)$$

$P_{i|j}$ is the conditional probability of correlations in the evolution of cluster i with respect to the evolution of cluster j and is given by: $P_{i|j} = W_{ij} / \sum_k W_{kj}$. The *a priori* probability of inputs quantized by neuron j is: $\hat{p}_j = \tau_j / \sum_k \tau_k$, where τ_j is the SRM of neuron j .

The GIST lateral weight, W_{ij} , records the frequency with which first neuron j and then neuron i are activated (i.e. first-order correlations). These weights are used in computing the $P_{(i|j)}$ values which yield the various P_i .

The GEST lateral weights record higher-order correlations using an adaptation of the Drive Reinforcement Theory (DRT) learning equation [3]. The correlation interval, $\Lambda = \sum_k \tau_k / |\bar{\tau} - \sigma_\tau|$, represents the number of presentations after which weight updates are performed. $\bar{\tau}$ is the network-wide average SRM value and σ_τ is the standard deviation. The lateral weight update rule at the end of the interval $[t - \Lambda, t]$ is:

$$\Delta W_{ij}(t) = \Delta P_i(t) \times \sum_n (|W_{ij}(t - n\Lambda)| \times \Delta \hat{p}_j(t - n\Lambda))$$

4 GST Application: Problem Formulation

The allocation of computations on data to processors, in data mapping, requires data-dependency information. The problem of modeling temporally emergent dependencies is solved by determining temporally emergent probabilities of *correlations* in processor references to data elements.

The test case involves mapping data in an ecological simulation, “WaTor,” of sharks and minnows in a toroidal ocean [4]. WaTor exhibits rapidly varying localities among slowly varying dataset sizes. The simulation parameters include a 200×75 ocean grid with 5000 sharks and 1563 minnows. The simulation runs for 10 iterations using 4, 8, 16 and 24 processors. Each application processor’s task partition consists of minnows distributed in the toroidal ocean mesh. Remote references would be minimized if each processor’s partition consists of neighboring minnows as WaTor computations require near-neighbor interactions. Hence, execution locality is measured by the average Euclidean distance (in the ocean) separating minnows within each processor’s task partition.

5 The Solution: Simulation Model and Application

Memory is abstracted as consisting of indivisible and contiguous **A**tomic **M**emory **U**nits (AMUs). Among descriptors associated with each AMU, the *AMU Utility* records the probability of local references to the AMU. Ongoing local computation induces changes in AMU descriptors producing non-stationary SVS distributions at each processor. For example, the *AMU Utility* varies due to increases (decreases) in the frequency of references to various AMUs as they are inducted (expelled) from the local working-set.

A GST (either GIST or GEST) network is used at each processor to capture spatial maps of AMU distributions in the local SVS from the processor reference streams. The *Signal Utility* of the neuron quantizing an AMU is assigned its *AMU Utility*. Offline training sets for networks are

GST Activities	Execution Speed
Training Vector Generation (T_{Vect})	7 KSR1 cycles
Snoop Vector Generation (T_{SVect})	62 KSR1 cycles
Input Vector Quantization (T_{Quant})	10 KSR1 cycles
GIST Lateral Weights Update (T_{IM})	34 KSR1 cycles
GEST Lateral Weights Update (T_{EM})	$(n^2 + 30n)$ KSR1 cycles
Neuron Addition (T_{Add})	100 KSR1 cycles
Neuron Deletion (T_{Del})	200 KSR1 cycles

Table 1: GST Simulation Timing Parameters. n in the T_{EM} timing refers to the network size.

derived from past processor references and the current contents of local memory. In this context, \hat{p}_j provides an *a priori* frequency based measure of the probability that AMU-cluster j contains AMUs of the current local working set. Also, $P_{(i|j)}$ can be viewed as the conditional probability that references to AMUs in cluster i are correlated with references to AMUs in cluster j . Thus, P_i is the probability of the presence of references belonging to cluster i in the reference stream.

The simulations assume an on-chip GST coprocessor unit per processor in the multiprocessor system. The execution driven simulation, implemented on the KSR1 [5], simulates concurrent execution of the GST coprocessor and the WaTor application. The essential timings for GST activities are given in Table 1, and have been derived from simulations and results in [6, 5, 7].

6 Simulation Results

The experimental results show that the GST networks successfully track the varying locality characteristics online and produce increasingly better execution locality.

- GST Sampling Efficiency:** References from the co-executing application processor are discarded during training and so the sampling efficiency, $[\text{No. of Refs. Processed}]/[\text{Total No. of Refs.}]$, evaluates the online tracking ability. The sampling efficiency ranges between **90%** to **100%**, indicating short offline training phases as in Figure 2 for the GEST network. The GIST sampling efficiency (not shown due to space limitations) was higher in comparison with the GEST network (primarily because $T_{EM} > T_{IM}$ usually), but suffered more high-frequency variations.
- WaTor Application Performance:** The execution timings in Table 2 show scalable WaTor performance and indicate that Minnow updates were responsible for 75% to 90% of the total execution time. The GIST locality measure plotted in Figure 3 shows increasingly better execution locality wherein the inter-minnow distances decrease from an initial high of 75 to level off at about 40. Hence, processor partitions achieved increasing contiguity

Run Size	Minnow Loop		Shark Loop	
	GIST Run	GEST Run	GIST Run	GEST Run
4 Processors	23.695952 sec	21.768551 sec	4.679540 sec	3.528160 sec
8 Processors	13.460341 sec	13.248189 sec	2.312860 sec	2.040482 sec
16 Processors	8.473986 sec	6.6604046 sec	1.409917 sec	1.273765 sec
24 Processors	6.034877 sec	4.6708921 sec	1.140522 sec	1.208808 sec

Table 2: WaTor Execution Timings (using GIST and GEST networks)

and compactness *in spite of the increasing dataset sizes*. This is also evident in the GEST network runs plotted in Figure 4 which shows an almost monotonic decrease in the inter-task distances.

7 Conclusions

The simulation results (in Section 6) show that both the GIST and GEST networks converge to stable maps of emergent non-stationary distributions without undue loss in their sampling efficiencies. Higher-order correlations are useful for smoothing high-frequency variations, and extract long-term low-frequency correlations in neuronal activations. Hence, the GEST network is better at tracking the long-range evolution of non-stationary distributions as seen in the almost monotonic improvements in locality in the GEST network runs (Figure 4). However, GEST network sampling efficiency was lower in comparison with that of the the GIST network.

References

- [1] B. M. Fritzke, “Growing Cell Structures - A Self-organizing Network for Unsupervised and Supervised Learning,” *Neural Networks*, vol. 7, no. 9, pp. 1441–1460, 1993.
- [2] T. Kohonen, *Self-Organization and Associative Memory*. Heidelberg, Germany: Springer-Verlag, third ed., 1989.
- [3] M. Caudill, “Neural Networks Primer, Part VII,” *AI Expert*, pp. 55–58, May 1989.
- [4] A. K. Dewdney, “Computer Recreations,” *Scientific American*, pp. 14–22, December 1984.
- [5] Kendall Square Research, *KSR/Series Principles of Operation*, March 1994.
- [6] R. H. Saavedra, R. S. Gaines, and M. J. Carlton, “Micro Benchmark Analysis of the KSR1,” in *Proc. of SUPERCOMPUTING ’93*, pp. 202–213, September 1993.
- [7] H. Speckmann, P. Thole, and W. Rosenstiel, “A COprocessor for KOhonen’s Selforganizing Map (COKOS),” in *Procs. 1993 International Joint Conference on Neural Networks*, pp. 1951–1954, 1993.

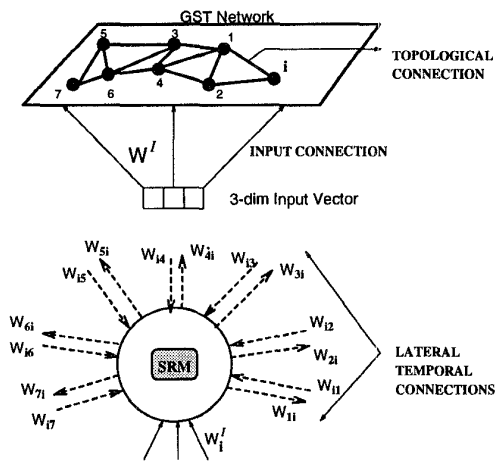


Figure 1: GIST and GEST Networks: Topology and Connectivities

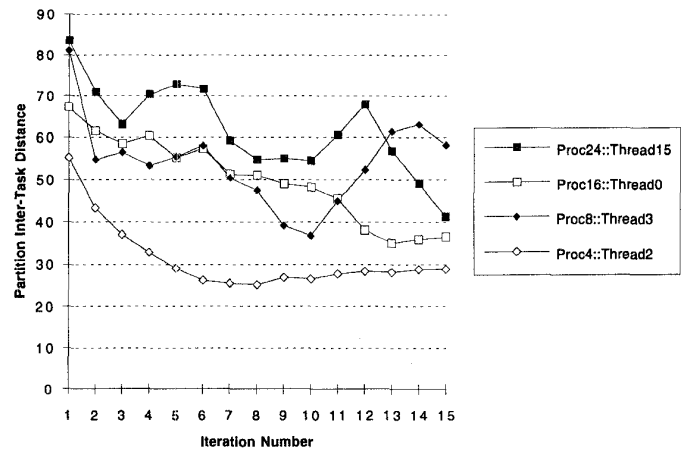


Figure 3: WaTor Locality Measure using GIST networks(15 iteration run) :: Thread Intra-Task Distances
Proc x ::Thread y \Rightarrow Locality Characteristics of Thread# y in an x processor run

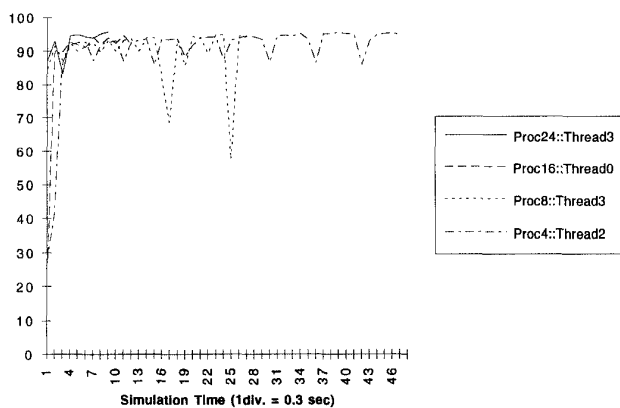


Figure 2: WaTor (10 iteration run) :: GEST Network Sampling Efficiency
Proc x ::Thread y \Rightarrow GEST network for Application Thread# y in an x processor run

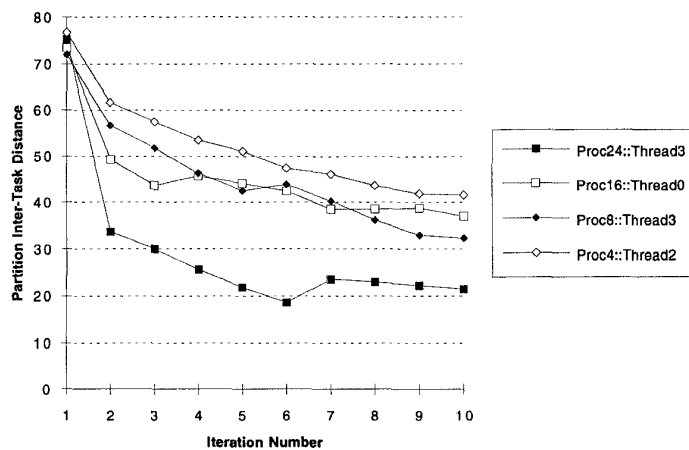


Figure 4: WaTor Locality Measure using GEST networks(10 iteration run) :: Thread Intra-Task Distances
Proc x ::Thread y \Rightarrow Locality Characteristics of Thread# y in an x processor run