STATISTICAL SELF-ORGANIZING MAP

A THESISIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
THE MIDDLE EAST TECHNICAL UNIVERSITY

BY

EMİN GERMEN

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY
IN
THE DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING

FEBRUARY 1999

Approval of the Graduate School of Natural and Applied Sciences

Prof. Dr. Tayfur ÖZTÜRK
Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Doctor of Philosophy.

Prof. Dr. Fatih CANATAN
Head of Department

This is to certify that we have read this thesis and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Doctor of Philosophy.

Prof. Dr. Semih BİLGEN
Supervisor

Examining Committee Members

Assoc. Prof. Dr. Melek YÜCEL (Chairperson)

Prof. Dr. Uğur HALICI

Assist. Prof. Dr. Işık AYBAY

Assist. Prof. Dr. Tolga ÇİLOĞLU

Prof. Dr. Semih BİLGEN

ABSTRACT

STATISTICAL SELF-ORGANIZING MAP

GERMEN, Emin

Ph.D., Department of Electrical and Electronics Engineering

Supervisor: Prof. Dr. Semih BİLGEN

February 1999, 92 pages

In this thesis, a new method has been proposed to calculate the learning rate and neighborhood function of Kohonen's Self Organizing Map methodology. The method is based on using statistical characteristics of both data and the net. This provides faster map convergence and better performance from the point of view of Average Quantization Error (AQE) compared to the conventional algorithms. Also the convergence proof of the proposed algorithm for one dimensional net and one dimensional data is given.

Keywords : Kohonen's Self-Organizing Map (SOM), Learning Rate, Neighborhood Function, Markov Process.

ÖZ

İSTATİSTİKSEL KENDİNDEN DÜZENLEMELİ HARİTALAR

GERMEN, Emin

Doktora, Elektrik ve Elektronik Mühendisliği Bölümü

Tez Yöneticisi: Prof. Dr. Semih BİLGEN

Şubat 1999, 92 sayfa

Bu tezde, Kohonen'in kendinden düzenlemeli harita algoritmasında öğrenme oranı ve komşuluk fonksiyonun hesaplanmasında yeni bir yöntem önerilmiştir. Yöntem, öğrenme oranı ve komşuluk fonksiyonun, gerek verinin, gerekse ağın istatistiksel yapısına göre değişimine göre şekillenmesi temeline dayanmaktadır. Önerilen yöntemin geleneksel yöntemlerle karşılaştırılması sonucu daha başarılı olduğu görülmüştür. Ayrıca tek boyutlu veri ve tek boyutlu harita için önerilen yöntemin doğru topografyaya yakınsadığı kanıtlanmıştır.

Anahtar Sözcükler : Kohonen Öz-düzenlemeli Harita, Öğrenme Oranı, Komşuluk Fonksiyonu, Markov süreci.

ACKNOWLEDGEMENT

I would like to express my sincere appreciation to Prof. Dr. Semih Bilgen for his expertly guidance and friendly encouragement throughout the research. Sincere thanks to Assoc. Prof. Dr. Melek Yücel for her assistance at the very beginning of the work. Also thanks for Prof. Dr. Passi Koikkalainen and Finnish government for supplying fruitful working atmosphere to formulate the basic idea of the work. To my wife, Gülriz, I thank her for her moral support and kind understanding of my frequent absences.

TABLE OF CONTENTS

LIST OF FIGURES

FIGURE

CHAPTER 1

INTRODUCTION

The appearance of digital computers in the midst of this century started to nourish technological improvement and caused the appearance of new interdisciplinary fields. The fast technological development in digital industry made come true the dreams of cognitive scientists in modeling the cognitive properties of the brain and the neuro-system of the body. The different areas of science, such as computer science, neuro-physiology and neuro-psychology, mathematics, statistics and biology, started to conglomerate the experiences to understand the nature of thinking. From the collaboration of those areas emerged a new subject called Artificial Neural Networks (ANN). While the theory of ANN is maturing, this was considered as a part of artificial intelligence. However nowadays the researchers who deal with the ANN theory or implementation are considered as members of an independent discipline.

The artificial neural network theory is divided into numerous sub disciplines according to the models that are used. Self-Organizing Map (SOM) is a neural network structure which allows unsupervised learning. After Kohonen has introduced the idea of SOM [Kohonen 1981] which combines the basic concepts of quantization theory and topological mapping structure of neurons in the brain, it has started to attract a great deal of attention. In a very short period of time like 15 years, thousands of applications of the SOM have been used in different areas of technology, social science, economics, and even in music. [Ritter and Schulten, 1997], [Kohonen 1988,c], [Lampinen and Oja, 1989], [Mononen et al., 1995], [Cosi et al., 1994] .

When Kohonen introduced the theory, he did not define an explicit learning rate and neighborhood function. He only stressed that those parameters have to have a decreasing nature with time. [Kohonen 1995]. In practice, they are chosen empirically according to the nature of the application. This thesis brings a novel contribution to the theory of SOM by introducing a method to determine the parameters of the SOM algorithm by considering the effects of the statistical changes of data and the response of the net to this data. These adaptive SOM parameters allow problem dependent topologies and fast map convergence. The convergence properties of the proposed method are investigated. By using Markov process, it is proven that this modified SOM for one-dimensional input data provides "correct" topology, which is described as "absorbing state" with probability one.

The remainder of this thesis is as follows. In Chapter 2, the general SOM theory is overviewed. The historical background of SOM and the methods which are used to strengthen the theoretical foundation of the idea will be given. This chapter also concentrates on the works done to prove the convergence of the map for some special data and the network topologies. Chapter 3 presents the new proposal for learning rate and the neighborhood function. The key point of this proposal is following the statistics of data and the net during training, and finding adaptive parameters, which provide data dependent solutions. In this chapter, in parallel with the development of the proposal, sample applications are presented to display its effectiveness. At this point, comparison of the proposed method and the commonly used learning rates and neighborhood function are presented. All software has been implemented on a Pentium 200 MMX computer using SOM_PAK package [Kohonen et al. 1995], and Microsoft Visual C++ 5.0 platform. In Chapter 4, the proof of convergence is given for the proposed method for one-dimensional net and one-dimensional data. In this proof, an approach based on the Markov process is used. Chapter 5 concludes the thesis with an overall summary and suggestions for further work.

CHAPTER 2

SELF-ORGANIZING MAP

Artificial Neural Networks (ANNs) constitute a very broad area of technology and find numerous applications in the wide spectrum of fields of science and technology. Research on the ANN theory has started in the 1940's and has accelerated after the 1970's due to the development of new learning algorithms, VLSI technology and parallel processing techniques [Freeman and Skapura,1991], [Haykin 1995]. With this rapid development, different neural network models, techniques and algorithms have been introduced. According to the learning rules, the networks can be categorized in three major groups.

- *Networks with supervised learning algorithms*: The essential point in the supervised learning algorithms is that the arrangement of the network parameters determined by an external teacher. This teacher can be thought as the desired

response of the net according to the given input signal. The difference between the actual response of the network and the desired response is called the error signal and the network parameters are adjusted according to this signal. Backpropagation [Rumelhart et. al 1986], LVQ Algorithm [Kohonen 1988 b], Adaline/Madaline [Widrow and Hoff 1960], Boltzmann Machine (BM) [Hinton et. al. 1984] are the examples of this category.

- *Networks with reinforcement learning algorithms:* In the reinforcement learning algorithms, the weights of the network are reinforced for properly performed applications and punished for poorly performed applications. The whole theory depends on maximizing a scalar performance index called a reinforcement signal throughout the trial and error process between the input-output mapping [Widrow et. al. 1973], [Sutton et. al. 1991]

- *Networks with unsupervised learning algorithms*: The major contribution of Kohonen Self Organizing Map is related with this category. Here, network elements compete with each other during the activation phase. Competition is realized through mutual lateral interactions and training continues in an unsupervised manner. The major examples of this category are SOM [Kohonen 1981] , ART1 [Carpenter and Grossberg 1986], ART2 [Carpenter and Grossberg 1987], BAM [Kosko 1987], Hopfield Memory, [Hopfield and Tank, 1986].

In this chapter we will concentrate on the SOM philosophy and study different techniques of improving the performance of the net, that have appeared in the literature. Also the theoretical background of the whole training process will be examined by checking different methods on convergence criteria of the map.

2.1 Self-Organizing Map

SOM is a neural network, which projects the higher dimensional input vector space onto one or two-dimensional array in a nonlinear fashion. Kohonen [Kohonen 1981] has introduced the basic algorithm of this theory in 1981. In the following years, the algorithm has matured [Kohonen 1988a], [Kohonen 1993a] [Kohonen 1993b], [Kohonen 1993c] and he has published two books containing the materials related with SOM and SOM dependent theory and applications [Kohonen 1988b], [Kohonen 1995]. Nowadays, numerous techniques are being proposed and published to increase the efficiency of SOM. [Koikkalainen 1993], [Koikkalainen 1995], [Fritzke 1992], [Bauer and Villman 1997]

The whole theory has been inspired by the results of biological research on retinotopic, somatosensory and cortical maps in the brain. The evidence obtained from experiments show that, some areas of brain tissue are organized according to the input signal. Different regions of the brain are dedicated to process some specific tasks, and in these parts of the brain, the topologically close neurons are connected with each other. The most impressive point of the localization process

is that, these areas (maps) are formed automatically and adaptively. The topological connection of the processing units (neurons) of SOM in one or two-dimensional array is the adaptation of this idea to ANN field. This was the first aspect of the SOM philosophy. The second important point was to find an answer to the question: "How do these topologically connected neurons organize adaptively and automatically according to the input signal?" The algorithms used in competitive learning theory and vector quantization (VQ) theory, helped to resolve this issue. The main idea of VQ is to find a code-book vector set which represents the input signal in a "best possible way", where the size of that vector set is much less than the actual signal vectors. The outcome of the VQ algorithm is a trained set of codebook vectors, which represent the input data distribution like SOM. However in SOM, the topographical settling of the neurons also contains information about the resemblance of individual data patterns that reside in the input data set.

In the initialization phase of the SOM algorithm, the vectors with the same dimension of the input data assigned to each neuron of a lattice of two-dimensional neuron structure are created randomly. The input vector is fully connected to every units of the lattice. During training, a competitive learning scheme is applied to each neuron and according to the activation strength, the winning neuron is selected as a best matching unit (BMU). This process has close similarities with Linde-Buzo-Gray algorithm that is one of the most widely used VQ algorithm. [Linde et al. 1980], [Buzo et al. 1980]. After the BMU is found, the

neurons inside the neighborhood of the BMU determined by the neighborhood function are updated.

2.1.1 Nomenclature

Here the basic notation which is used throughout during this thesis will be introduced.

N	: Dimension of the neuron and input training data,
i	: Neuron index, where $i=1, 2, \ldots, J$,
$\boldsymbol{M}_i(k)$	: The vector of the i^{th} neuron at iteration step k,
$m_{i,n}(k)$	: n^{th} coordinate* of $\boldsymbol{M}_i(k)$ where $n=1, 2, \ldots, N$, *(If a 1-D net is defined, the i^{th} neuron is also defined as m_i to differentiate it from a vector.)*
$x_{M_i}(k)$	: The topological x index of neuron i at iteration step k,
$y_{M_i}(k)$	: The topological y index of neuron i at iteration step k,
X	: Number of neurons in the x direction,
Y	: Number of neurons in the y direction,
J	: Total number of neurons in the net where $J = X \cdot Y$ in two-dimensional net, or $J = X$ in a one dimensional net
k	: The iteration step, where $k=1,2, \ldots, k_{max}$,

* In literature, some authors refer to these coordinates as weights [Haykin 1994] or weight-space coordinates [Freeman and Scapura 1991]. We have preferred to adhere to the simple term coordinate differentiating it from neuron indices.

$\Lambda(k)$: Training data presented at iteration step k, where $\Lambda^T = (\Lambda_1, \Lambda_2, \ldots \Lambda_N)$, in which, Λ_n represents the n^{th} coordinate of the data vector,

$M_c(k)$: The weight vector of the Best matching unit (BMU) (the winning neuron), at step k,

c : Best matching unit index,

$\alpha(k)$: Learning rate parameter,

$R(k)$: Radius around the BMU at iteration step k, It is used to calculate the width of Gaussian function around BMU,

$dd_{i,c}(k)$: Topographical distance between the BMU and M_i

$\beta(c, i, k)$: Neighborhood function.

2.1.2 Algorithm

The training algorithm steps are [Kohonen, 1995]:

1. Assign a random vector to each neuron.
2. For each training data, find the best matching unit (BMU) using Euclidean distance measure.

$$c = \arg\min_i \left[\sum_{n=1}^{N} \left(\Lambda_n(k) - m_{i,n}(k) \right)^2 \right] \qquad (2.1)$$

Where $\Lambda_n(k)$ is the n^{th} item of data sample vector at iteration k, and $m_{i,n}(k)$ is the location of the n^{th} item of the i^{th} neuron at time k.

3. Train the net iteratively by updating all neurons using $\alpha(k)$ and $\beta(c,i,k)$. where $\alpha(k)$ represents the learning rate and $\beta(k)$ is the neighborhood function.

$$M_i(k) = M_i(k-1) + \alpha(k) \cdot \beta(c,i,k) \cdot \left(\Lambda(k) - M_i(k-1)\right) \qquad (2.2)$$

where $0 < \alpha(k) < 1$ and $0 < \beta(c,i,k) < 1$

4. Repeat from *3* until $k=k_{max}$ See Figure 2.1

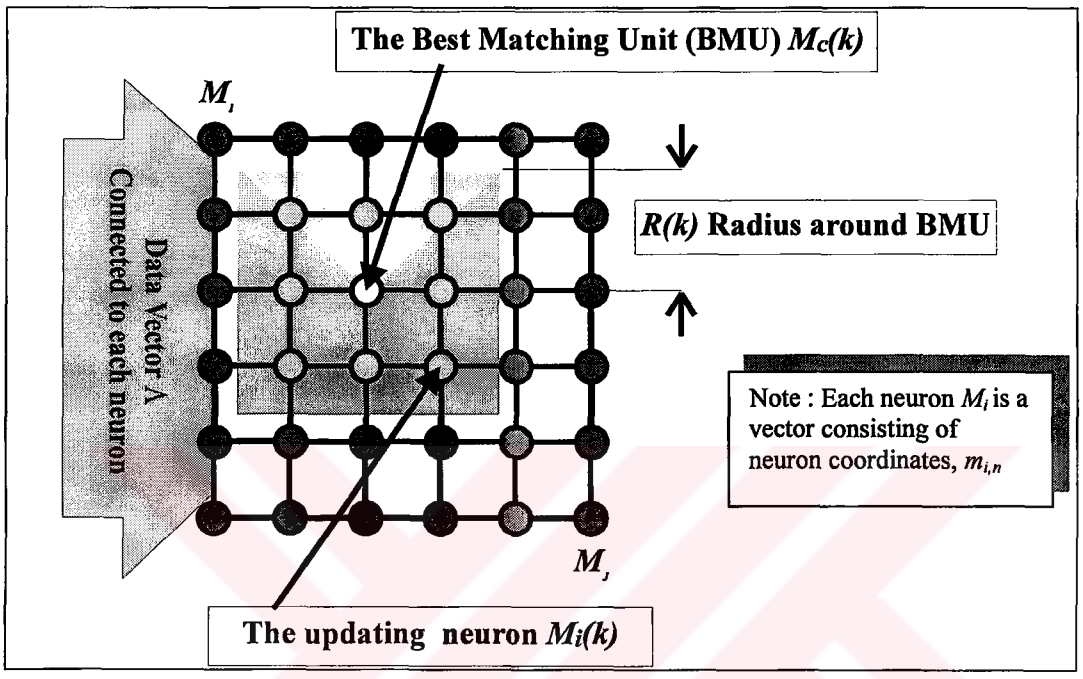


Figure 2.1 Two-dimensional neurons connected as a lattice and example of topological neighborhood around the Best Matching Unit (BMU)

2.1.3 Learning rate and neighborhood function

Performance of the whole training process and the quality of the resultant topology considering the input data distribution is closely related with the choice of the learning rate and the neighborhood function parameters. Learning rate is a scalar value and can be considered also as an adaptation gain. In recursive stochastic algorithms like SOM, this gain controls the stability of the system. [Ljung

1977]. The neighborhood function determines the adaptation strength of the neurons around the best matching unit. This function defines the stiffness of the elastic surface around the winning neuron during iteration. As it is mentioned in Chapter 1, there is no explicit definition for learning rate and neighborhood function, however they have to be decreased with time.

The most widely used learning rate parameters are given below:

- Exponentially decreasing learning rate [Kohonen 1990] (Exponential alpha):

$$\alpha(k) = \alpha_{inital}\left(\frac{\alpha_{final}}{\alpha_{initial}}\right)^{\frac{k}{k_{\max}}} \qquad (2.3)$$

Where $\alpha_{initial}$ and α_{final} values are determined empirically. The initial and final values of the learning rate are chosen as 1 and 0.01 respectively for all the experiments in this thesis.

- Inversely decreasing learning rate [Kohonen et al. 1995] (Inverse alpha):

$$\alpha(k) = \frac{C \cdot \alpha_{initial}}{C + k} \qquad (2.4)$$

where C is a problem dependent constant and a reasonable choice is $C = {k_{\max}}/{100} \cdot \alpha_{initial}$ is selected as 1.0 for the experiments in this thesis.

- Linearly decreasing learning rate [Kohonen et al 1995] (Linear alpha):

$$\alpha(k) = \alpha_{initial} \cdot \frac{k_{\max} - k}{k_{\max}} \qquad (2.5)$$

- The learning rate proposed by Mulier and Cherkassy [Mulier and Cherkassky 1995]. (Mulier alpha):

$$\alpha(k) = \frac{1}{p \cdot (k-1) + 1} \qquad (2.6)$$

where

$$p = \frac{1 - \frac{J}{k_{\max}}}{J - \frac{J}{k_{\max}}} \qquad (2.7)$$

Like learning rate, various neighborhood functions can be used according to the nature of the problem. The only constraint is that the neighborhood function be non-increasing around the BMU. The most common neighborhood function parameters are:

- Linear shrinking Gaussian neighborhood function [Kohonen et al 1995] (LinGauss neighborhood):

$$\beta(c,i,k) = e^{-\left(\frac{dd_{i,c}(k)^2}{2 \cdot R(k)^2}\right)} \qquad (2.8)$$

where

$$R(k) = 1 + (R_{initial} - 1) \cdot \frac{k_{\max} - k}{k_{\max}} \qquad (2.9)$$

and for two dimensional topology:

$$dd_{i,c}(k) = \sqrt{\left(x_{M_i}(k) - x_{M_c}(k)\right)^2 + \left(y_{M_i}(k) - y_{M_c}(k)\right)^2} \qquad (2.10)$$

$R_{initial}$ is chosen as $Max(X, Y)$.

- Exponential shrinking Gaussian neighborhood function [Cherkassky and Lari-Najafi 1991] (ExpGauss neighborhood):

$$\beta(c,i,k) = e^{-\left(\frac{dd_{i,c}(k)^2}{2 \cdot R(k)^2}\right)} \qquad (2.11)$$

where $$R(k) = R_{initial}\left(\frac{R_{final}}{R_{initial}}\right)^{\frac{k}{k_{\max}}} \qquad (2.12)$$

$R_{initial}$ = Max(X, Y) and R_{final} = 1.

- Bubble neighborhood function [Kohonen et al 1995] (Bubble neighborhood) :

$$\beta(c,i,k) = \begin{cases} 1, & if \ |c-i| \le s(k) \\ 0, & otherwise \end{cases} \qquad (2.13)$$

where *s(k)* is a linear or exponential decreasing function with iteration step *k*.

In general, when using rapidly decreasing learning rates like *exponential alpha* or *inverse alpha*, it is proposed that the learning process be divided into two phases. In the first phase, relatively large initial alpha values are used (e.g. α = 0.9, α = 1). This phase is considered as the time interval where the global ordering occurs. After this period, comes a second phase beginning with small initial alpha values (e.g. α = 0.1, α = 0.01) where fine-tuning of the neurons occurs [Kohonen 1990]. Experiments have shown that this training scheme with two phases improves training performance. This fact is studied in detail in [Mulier 1994]. He has stated that, "*the effects of the early presentations with a large neighborhood are nearly forgotten after self-organization is complete.*" The very striking result of this research is that only the last 20 percent of the input data has an effect on the formation of the final topology. Thus, it is not difficult to guess that the order of presentation has a great influence on the performance of the self-organizing phase. By introducing the learning rate in (2.6), Mulier and Chekassky have obtained equal

contribution of each input data on the formation of the final topological structure of neurons.

2.2 Analyses of the Self-Organizing Process

Kohonen [Kohonen 1982] was the first author who studied the mathematical background of the SOM theory. Here convergence properties of one-dimensional SOM are investigated. After SOM has started to gain practical importance, the theory has attracted the attention of many researchers and hence, numerous papers have been published on analyses of the theory. In general those contributions can be classified in the following groups:

- Statistical analyses of SOM,
- Analyses of SOM with energy functions,
- Ordering and convergence properties of SOM with Markov process.

The common point of all works analyzing the SOM is that, the ordering properties are investigated for one-dimensional input applied for one-dimensional net since the ordering of topology for one-dimensional net is known. The ordered form of one-dimensional net is stated as:

The organizing phase for one-dimensional network and one-dimensional input is the topographic formation of neurons having a monotonic increasing or

monotonic decreasing character. In a SOM where D=1 (dimension of neuron grid) and N=1 (dimensions of the neuron and input training data) this state can be shown as

$$\mathcal{S} = \mathcal{S}' \bigcup \mathcal{S}'', \text{ where}$$

$$\left.\begin{aligned} \mathcal{S}' &= \{m : m_1 > m_2 > \ldots > m_J\} \\ \mathcal{S}'' &= \{m : m_1 < m_2 < \ldots < m_J\} \end{aligned}\right\} \quad (2.14)$$

After this state is reached, the ordering remains same for any set of data.(On this, one can refer to [Kohonen 1988b], [Flanagan 1996], [Erwin et al. 1992a], [Erwin et al. 1992b] [Kohonen 1995].

2.2.1 Statistical Analyses of SOM

The SOM algorithm is considered as a stochastic process. Hence the idea of statistical analysis of SOM has attracted many researchers e.g. [Yin and Allinson 1993], [Yin and Allinson 1995] [Mulier 1994], [Mulier and Cherkassky 1995]. Since learning rate and neighborhood function parameters in SOM have no explicit optimum definition, those are quite sensible to data presented and the topology itself. In [Mulier 1994] and [Mulier 1995] the statistical analyses of learning rate parameter have been carried out. In those works, the statistical effects of contribution of each data on the final map have been investigated. This method can be described here as:

The updating equation (2.2) can be rewritten in noniterative form as:

$$\begin{aligned} M_j(k) = & \prod_{r=1}^{k}\left[1 - a_j\left(M_c(r), r\right)\right] \cdot M_j(0) \\ & + \prod_{r=2}^{k}\left[1 - a_j\left(M_c(r), r\right)\right] \cdot a_j\left(M_c(1), 1\right) \cdot \Lambda(1) \\ & \vdots \\ & + \prod_{r=n-1}^{k}\left[1 - a_j\left(M_c(r), r\right)\right] \cdot a_j\left(M_c(n), n\right) \cdot \Lambda(n) \\ & \vdots \\ & + a_j\left(M_c(k), k\right) \cdot \Lambda(k), \end{aligned} \qquad (2.15)$$

where $a_j()$ denotes the "*adaptation strength*" which combines the learning rate and the neighborhood functions.

Rewriting this equation gives:

$$M_j(k) = d_j(k) \cdot M_j(0) + \sum_{n=1}^{k} d_j(k,n) \cdot \Lambda(n) \qquad (2.16)$$

The $d_j(k,n)$ describes the contribution of data presented at time n on the position of neuron j at time k. Also $d_j(k)$ is the contribution factor of the initial value of the same neuron on iteration k. By keeping those values in a matrix in multiple experiments, the contribution factor of every presented data on each neuron in the net is calculated and studied. In those studies it is observed that if Exponential alpha and ExpGauss neighborhood is used, almost 80% of presented data has no influence on the final position of each neuron.

In order to find a possible learning rate function which provides equal contribution of each data item, the following condition has to be satisfied: [Mulier and Cherkassky 1995]

$$\sum_{j=1}^{J} d_j(k_{\max}, n) = \frac{J}{k_{\max}} \quad for\ n = 1, \ldots, k_{\max} \qquad (2.17)$$

By determining the initial and the final contributions and introducing the Eq. (2.17) into (2.15) the Mulier learning rate has been found analytically.

The statistical analyses are also studied in [Cherkassky et al. 1996] and criteria for comparing different adaptive methods including ANN from the point of view of statistics are proposed. Those criteria can easily be adapted to every method which aims to estimate an unknown function from a finite number of presented data.

2.2.2 Analyses of SOM with Energy Functions

The main idea of introducing the energy function to analyze the SOM is to represent the whole process as a system of energy equations and describe the whole system behavior in terms of global Lyapunov functions. According to this approach, (if it is possible), for each step of iteration, the global ordering can be analyzed by gradient-descent learning procedure. This kind of energy equations are used to study the convergence properties of Hopfield networks.

There are plenty of works considering this phenomenon [Tolat 1990], [Erwin et. al 1992a], [Erwin et. al 1992b], [Lo et. al], [Heskes 1996]. [Tolat 1990] suggests to define a set of energy functions for each unit in the map, instead of defining a general energy function. A very trivial small set consisting of three neurons connected in one dimension is used to define those energy functions. The energy of a neuron i is defined as:

$$e_i(k) = \int_{\Lambda \in P} \alpha(k) \cdot \beta(c,i,k)\left[J\left(\Lambda(k), m_i(k)\right)\right]^2 d\Lambda \qquad (2,18)$$

Where $\Lambda(k)$ represents an input pattern in the input pattern space P, and J is a function used to determine the similarity between input data and the neuron. In the work, euclidian distance is used as J. For the net with 3 neurons, the energy functions of possible 6 ordering of the weights have been derived and their minimal values are obtained. For example for the case of ordering $m_1 < m_2 < m_3$, the energy function for neuron i (m_i) is derived as:

$$e_i = \int_0^{(m_1+m_2)/2} c(i,1)(\Lambda - m_i)^2 dx + \int_{(m_1+m_2)/2}^{(m_2+m_3)/2} c(i,2)(\Lambda - m_i)^2 dx + \int_{(m_2+m_3)/2}^{1} c(i,3)(\Lambda - m_i)^2 dx \qquad (2.19)$$

where $c(i,j)$ represents the adaptation strength, result of the learning rate and the neighborhood function. For constant learning rate and bubble neighborhood function with bubble width 1, all possible energy functions are calculated and found that, ordered weights minimize the energy functions. Thus the weights are guaranteed to become ordered with minimum energy functions.

In [Erwin et. al 1992a] it is shown that, for highly disordered maps, energy functions explained in [Tolat 1990], are not valid and a more general set of energy functions is proposed for one dimensional net. In this proposal, in order to simplify the energy equations and to define the limits of influence regions of data on each neuron, new index values are appointed to the neurons such that, if the neuron coordinate has increased, the index is increased. That is if $m_i < m_j \longrightarrow i < j$. Where i, j defines the index and m_i, m_j defines the neuron coordinates or "weight values". The new indices can be converted to the old indices by means of a permutation function $\mathcal{P}(i)$ such that $m_{\mathcal{P}(i)} \equiv m_s$, meaning that the ordering m_i is obtained via permutation $(\mathcal{P}(x))'$ from an earlier ordering m_s. This permutation function is uniquely defined for each possible configuration of neuron coordinates. Here m_s is denoting the old symbol for neuron s.

Thus the average force acting on neuron m_i can be represented as:

$$V(m_i,k) = \alpha(k) \cdot \sum_{j=1}^{J} \tilde{\beta}(i,j,k) \cdot \int_{\Lambda \in \Omega(j)} (\Lambda - m_i) P(\Lambda)\, d\Lambda \qquad (2.20)$$

where $\tilde{\beta}(i,j,k) = \beta(\mathcal{P}(i),\mathcal{P}(j),k)$ and

Λ is integrated in a Voronoi[†] cell where those are defined as:

[†] A Voronoi cell of a unit x is consists of those points in data space which are closer to neuron m_x than to any other neuron m_y

$$\left.\begin{aligned}
&\Omega(1)=\left\{\Lambda \mid 0<\Lambda<\frac{1}{2}(m_1+m_2)\right\},\\
&\vdots\\
&\Omega(j)=\left\{\Lambda \mid \frac{1}{2}(m_{j-1}+m_j)<\Lambda<\frac{1}{2}(m_j+m_{j+1})\right\},\\
&\vdots\\
&\Omega(J)=\left\{\Lambda \mid \frac{1}{2}(m_{J-1}+m_J)<\Lambda<1\right\}
\end{aligned}\right\} \qquad (2.21)$$

Substituting Eq.(2.21) in Eq.(2.20) with constant probability density function, *P(x)* provides:

$$\begin{aligned}
V(m_i)=\sum_{j=2}^{J-1}\tilde{\beta}(i,j,k)&\left[\frac{(m_{j+1}^2-m_{j-1}^2)}{8}+\frac{m_j(m_{j+1}-m_{j-1})}{4}+\frac{m_j(m_{j-1}-m_{j+1})}{2}\right]\\
&+\tilde{\beta}(i,1,k)\left[\frac{(m_1+m_2)^2}{8}+\frac{m_i(m_1+m_2)}{2}\right]\\
&+\tilde{\beta}(i,J,k)\left[\frac{1}{2}-m_i-\frac{(m_J+m_{J-1})^2}{8}+\frac{m_i(m_J+m_{J-1})}{2}\right]
\end{aligned} \qquad (2.22)$$

Since the $V(m_i)$ is the average force on neuron m_s, the energy E[m] is calculated by solving the equation:

$$\frac{\partial E[m]}{\partial m_i}=-V_i(m) \qquad (2.23)$$

In [Erwin et. al 1992a], it is shown that there can not be an energy function E[u]. In [Tolat 1991] it has been suggested to use individual energy functions instead of using general energy function as :

$$\frac{\partial E_i[m]}{\partial m_i} = -V_i(m) \qquad (2.24)$$

In [Erwin et. al 1992a], it is explained that the individual energy functions require two terms for unit u_x as:

$$\begin{aligned} E_i[m] &= \tilde{E}_i[m] + X_i[m] \\ &= \alpha(k)\sum_{y=1}^{J}\tilde{\beta}(i,j,k)\int_{v\in\Omega(j)}\frac{1}{2}(\Lambda - m_i)^2 P(\Lambda)d\Lambda + \\ &\quad \frac{\alpha(k)}{48}\sum_{j=2}^{J}(m_j - m_{j-1})^3\left(1-\tilde{\beta}(j-1,j,k)\right) + P\left(\frac{1}{2}(m_j + m_{j-1})\right) \end{aligned} \qquad (2.25)$$

Here the $\tilde{E}_i[m]$ represents the energy function which [Tolat 1991] proposes and the second term contributes the correction of total energy due to changing the borders of Voronoi regions during an adaptation process. If the map is ordered, the second term has no domination on the energy, however when the map is not ordered, the second term has great influence on the energy function.

2.2.3 Ordering and convergence properties of SOM using Markov process.

The ordering and convergence properties of SOM have been studied using Markovian process in many articles [Kohonen 1982], [Kohonen 1988], [Kohonen 1995], [Ritter and Schulten, 1986], [Flanagan 1996], [Flanagan 1997], [Thiran and Hasler 1994]. Since the only well-described convergence criterion is defined on one dimensional SOM [Cotrell and Fort 1987], the proofs in all articles

are restricted to one dimensional input for the one-dimensional SOM. The Markovian process constitutes the basis of the general structure of the proofs.

Markov Random Process and Markov Chain Property :

The discrete valued Markov Random Process satisfies the Probability Mass Function (PMF) condition: [Stark and Woods 1994]

$$P_X(x_n \mid x_{n-1},\ldots,x_1;k_n,\ldots,k_1) = P_X(x_n \mid x_{n-1};k_n,k_{n-1}) \qquad (2.26)$$

for all $x_1,\ldots x_n$ and for all $k_1 < \ldots < k_n$ and for all integers $n > 0$.

The value of the process X at a given time k determines the conditional probabilities for future values of the process. The set of values of the process at any time k is called the *state of the process*. The conditional probabilities are thought of as transition probabilities between the states. If the states are finite and countable, then the process is called a *Markov chain*. The Markov property states that, the probability of entering a certain state depends only on the *last state* occupied.

If $X = \{X_1 \ldots, X_n\}$ is the set of all states in a Markov chain, then the subset of states $X' \subseteq X$ is said to be closed if no one-step transition is possible from any state in X' to any state in X'^c (the complement of the set X' in X). If X' consists of a single state, say E_i, then it is called an *absorbing state* [Kleinrock 1975 p.28].

For a SOM, in the case of a 1-D network and 1-D input space, a set of neuron coordinates (or configurations of the weights) $\{m_1, m_2, \ldots, m_J\}$ is considered as a state of the Markov process. All of the proofs which use the Markov process for analyzing the SOM, use the same general idea: If it is possible to find a set of input sequences $\zeta(k)$ having a nonzero probability, which forces the initially disordered neurons to an ordered configuration, which is an absorbing state (See Sec. 2.2), any infinite random sequence of inputs will bring the neurons to that absorbing state with probability 1 in a finite amount of time, regardless of the initial neuron configuration.

In [Kohonen 1988] convergence of 1-D SOM is investigated by analyzing the dynamic behavior of the expected values $E\{m_i\}$. In this work the neighborhood function is defined as :

$$\beta(c,i,k) = \begin{cases} 1, & if \; |c-i| = 1 \\ 0, & otherwise \end{cases} \qquad (2.27)$$

Since the neighborhood function in Eq. (2.27) effects only the 1-neighborhood of the BMU, in the adaptation process, at most 3 neurons, that is 5 inter-neuron distances will be modified [Kohonen 1988b p. 150]. Here also a disorder parameter D has been defined in order to measure the degree of disorder during training as:

$$D = \sum_{i=2}^{J} \left(|m_i - m_{i-1}| \right) - |m_J - m_1| \qquad (2.28)$$

If $D=0$ than, it is concluded that the neurons are ordered and D remains the same for the rest of the process. During training, the learning rate parameter α is non-increasing, and eventually decreases to 0.

The ordering theorem [Kohonen 1988b] states that: during SOM training process, D parameter has a decreasing nature for most of the time and finally reaches 0. In the proof of that theorem, all possible neuron orderings which can cause an increase in the value of D are considered. For example, one possible neuron order which can increase D is:

- $(m_1(k), m_2(k), m_3(k)) > m_J(k)$
- $|\Lambda(k)-m_1(k)|<|\Lambda(k)-m_2(k)|$ *and* $|\Lambda(k)-m_1(k)|<|\Lambda(k)-m_3(k)|$
- $\Lambda(k) < m_1(k)$

At time $k+1$ only two neurons m_1 and m_2 will be updated as :

$m_1(k+1) = \alpha(\Lambda(k)-m_1(k))$ *and* $m_2(k+1) = \alpha(\Lambda(k)-m_2(k))$

The $D(k+1)$ parameter has 4 possibilities according to Table 2.1 as :

Table 2.1 Neuron positioning possibilities at time k

Case	m_2-m_1	m_3-m_2
1	>0	>0
2	>0	<0
3	<0	>0
4	<0	<0

The change in D also has four possibilities according to the Table 2.1 as :

$$
\begin{aligned}
&dD_{Case1}/dt = 0 \\
&dD_{Case2}/dt = 2\alpha(\Lambda(k) - m_2(k)) < 0 \\
&dD_{Case3}/dt = 2\alpha(m_2(k) - m_1(k)) < 0 \\
&dD_{Case4}/dt = 2\alpha(\Lambda(k) - m_1(k)) > 0
\end{aligned}
\qquad (2.29)
$$

In Eq.(2.29) it is observed that in Case4, D increases however the others make it decrease or keep it unchanged. In [Kohonen 1988b] all possibilities are studied and it is concluded that : "There are more cases above in which D decreases than in which D stays constant or increases. When ordering starts to build up, one will observe that the cases in which D stays constant become more and more frequent, and finally the network reaches to the ordered state". A similar kind of analysis has been carried out for different neighborhood functions with bubble width 1 in [Kohonen 1995].

In [Flanagan 1996] a different approach has been used to analyze SOM with a more general neighborhood function. There, it is defined as:

$$
\begin{aligned}
&\beta(c,c,k) = 1 \\
&\beta(c,i,k) > \beta(c,j,k) \ , \ |c-i| < |c-j| \\
&\beta(c,i,k) > 0
\end{aligned}
\qquad (2.30)
$$

In Flanagan's work the input data set is restricted as $\Lambda \in [0,1]$. In his set two regions are defined at the borders as $[0,\varphi]$ and $[1-\varphi,1]$. Here $\varphi < 1/2$ and it is defined in terms of minimum learning rate and neighborhood function as:

$$
\varphi = \frac{\alpha_{\min} \cdot \phi}{1 + \alpha_{\min} \cdot (2\phi - \beta_{\min})} \qquad (2.31)
$$

where α_{min} : The minimum value of the learning rate function,

ϕ : The minimum difference between the two values of neighborhood function,

β_{min} : The minimum value of the neighborhood function.

The value φ has a property that if two neurons reside in $[0,\varphi]$ as $m_i, m_j \in [0,\varphi]$, and if there exist an input in $[1-\varphi,1]$ at time k then even if the condition $\beta(m_c, j, k) > \beta(m_c, i, k)$ holds, the next iteration yields $m_j(k+1) > m_i(k+1)$. Hence the neurons i and j become organized with respect to the neuron c. This constitutes the basic idea of the Flanagan's converge proof.

Flanagan's SOM convergence theorem is based on finding set of samples of ψ, with probability greater than zero, which will eventually place all neurons in an organized configuration (absorbing state of the process) regardless of the arbitrary initial state. The convergence theorem is expressed in three phases:

Phase 1. Define a set of samples which move all neurons into the region $[0,\varphi]$ with nonzero probability. For this phase, it is shown that, if the samples are continually drawn from this set, in a finite amount of time, all neurons enter into the region $[0,\varphi]$ as:

$$\pi_{x^o}\{\psi \in \Psi : \tau_\varphi \le T_\varphi\} \ge \delta_\varphi \qquad (2.32)$$

Whereby π_{x^0} is defined as a Markov process in time τ_φ, with initial state of x^0, and training data sequences ψ drawn from the input space Ψ. Here T_φ is the first entry time of all neurons into the region, with probability $\delta_\varphi > 0$.

Phase 2. Since all neurons are in region $[0,\varphi]$ then if the next data sample is drawn from the region $[1-\varphi,1]$ all neurons are organized according to the best matching unit m_c. If $c=1$ or $c=J$, then all weights are ordered which is the absorbing state for the whole process. If not, the state can be brought to the set Δ by drawing data samples from $[1-\varphi,1]$ continually where Δ is defined as:

$$\Delta = \Delta_1 \cup \Delta_2 \cup \Delta_3 \cup \Delta_4 \qquad (2.33)$$

In which:

$$\left.\begin{aligned} \Delta_1 &= \{\mathbf{m} : for\ J-1 > r \geq 1;\ m_1 < m_2 < \ldots m_{r-1} < m_r \leq 1-\varphi < m_{r+1}, \ldots m_J\} \\ \Delta_2 &= \{\mathbf{m} : for\ 2 < r \leq J;\ m_N < m_{N-1} < \ldots m_{r+1} < m_r \leq 1-\varphi < m_{r-1}, \ldots m_1\} \\ \Delta_3 &= \{\mathbf{m} : 1-\varphi < x_1 < x_2, x_3, \ldots, x_J\} \\ \Delta_4 &= \{\mathbf{m} : 1-\varphi < x_J < x_{J-1}, x_{J-2}, \ldots, x_1\} \end{aligned}\right\} \qquad (2.34)$$

As a Markov process in time τ_Λ, this phase can be defined as:

$$\pi_{x^0}\{\psi \in \Psi : T_\varphi < \tau_\Lambda \leq T_\Lambda\} \geq \delta_\Lambda \qquad (2.35)$$

Where $x^0 = X(0) \in [0,\varphi]$, $T_\Lambda < \infty$ and $\delta_\Lambda > 0$. T_Λ is the first entry time for all neurons into Δ.

Phase 3. After the second phase at the end of which the neurons are in the set Λ, if a data sample is drawn from the region $[0,\varphi]$ one more time, it is shown that the neurons will be fully ordered.

The main contribution of Flanagan's work is, extending the application of Markov process for proving the convergence of SOM by defining some critical regions in data space for more general learning rates and neighborhood functions. He has shown that if these critical regions can contain data with nonzero probability, the direction of convergence can be controlled and the phases to reach the absorbing state can be defined explicitly. In Chapter 4, our analysis convergence of the proposed method with a new learning rate and neighborhood function will follow a similar approach.

2.3 Other Analysis Methods of SOM

The resultant topologies of SOM contain information of the similarity of input data. The performance of topological formation at the end of training has been studied in several articles. [Bauer and Pawelzik 1992], [Martinetz and Schulten 1994], [Kaski and Lagus 1996], [Villmann et. all 1997], [Herrmann 1995], [Ypma and Duin 1997]. In those papers representation of high-dimensional data with one or two dimensional neuron topologies is studied. Stating a measure in order to compare different maps for the same set of data comprises the most important point of this task. In [Ypma and Duin 1997] and [Kaski and Lagus 1996], a criterion called

measure of goodness has been developed. Here for each data item Λ, the distance d(Λ) from Λ to the second nearest referenced vector $m_{c'(\Lambda)}$ is found. However during this process the path from the best representative neuron $m_{c(\Lambda)}$ is found and shortest path by passing through other neurons in the lattice is searched as:

$$d(\Lambda) = \| \Lambda - m_{c(\Lambda)} \| + \min_{i} \sum_{k=0}^{K_{c'(\Lambda),i}-1} \| \mathrm{m}_{\mathrm{I}_i(k)} - m_{I_i(k-1)} \| \quad (2.36)$$

where $I_i(k)$ represents the index of k^{th} unit on a path along the map grid from the unit $I_i(0)$ to the second nearest vector in input space $I_i\left(K_{c'(\Lambda),i}\right)$.

where $I_i(0) = c(\Lambda)$ and $\mathrm{I}_\mathrm{i}(K_{c'(\Lambda),i}) = c'(\Lambda)$

The goodness of the map is defined as: C=E[d(Λ)] that measures the continuity of the mapping and quantization error for dissimilar maps. Here E defines the average value. It is calculated over all input samples.

CHAPTER 3

THE NEW LEARNING RATE AND NEIGHBORHOOD FUNCTION PARAMETERS

The quality of the resultant topology in Kohonen's Self-Organizing Map for on-line learning is highly dependent on the characteristics of the learning rate and the neighborhood function, which are formulated heuristically before the training period. This is considered as the major weakness of the whole method [Mulier 1994], [Kangas 1994]. Although this theory has been studied for almost 15 years (see Chapter 2) there still are ambiguities in optimizing the parameters of the algorithm.

In this chapter, a new approach is introduced to define a new learning rate and the neighborhood function in SOM. This approach employs defining those parameters in a way that they are sensitive to the change in the statistical characteristics of both data and the response of the net to this data simultaneously [Germen and Bilgen 1997], [Germen and Bilgen 1998]. In this method, while finding the best matching unit (BMU), also the worst-matching unit (WMU) is calculated. The on-going calculations of standard deviation and the mean of WMU are used to redefine the learning rate and neighborhood function. These new adaptive SOM parameters allow problem dependent topologies and fast map convergence as well as sensitivity to possible changes in data statistics.

3.1 The motivation

The fundamental idea to improve the performance of SOM is keeping the whole SOM philosophy untouched, however finding a new approach whereby SOM can follow the statistics of data presented. In other words, input $\Lambda(k)$ is considered to have a stochastic nature, and the topological ordering of the neurons in the net has to respond this data using the statistical outline of training. In conventional SOM algorithms, the variations in data can not easily be tracked and adaptation of the topologies to the new incoming data having different statistical

nature is almost impossible. The reason of this phenomenon is that, the parameters of the algorithm are not adaptive and they are chosen empirically.

In this proposal the aim is to bring three concepts together. The first one is that the statistical nature of data has to effect the training parameters. The second one is, during training, the current network topological structure has to have a direct influence on the learning rate and neighborhood parameters. The last one is that the statistical variations in data during training must have an impact on the memorized patterns. If those three requirements are met, the proposed method which has an adaptive nature, will be more reliable than the conventional approaches in which a predefined adaptation strength is used. In conventional methods in which empirical parameters are used, subsequent changes in data can not alter the topologies. Hence the global ordering of neurons is affected mainly on the later data and early presentations don't have much effect on the neuron configuration although they have different statistical nature than the succeding ones. [Mulier and Cherkassky 1995]

In this chapter, the reasoning that led to the final version of the new learning rate and neighborhood functions will be presented step by step, displaying the intermediate stages before the final proposal is reached. The proposed learning rate is:

$$\alpha(k) = \begin{cases} 1, & if \dfrac{\nu(k)\cdot\theta(J,k)}{\sqrt{\kappa}} > 1 \\ \dfrac{\nu(k)\cdot\theta(J,k)}{\sqrt{\kappa}}, & otherwise \end{cases} \qquad (3.1)$$

and the novel neighborhood function is:

$$\beta(c,i,k) = e^{-\left(\frac{dd_{i,c}(k)^2}{2\cdot\Omega(k)^2}\right)} \qquad (3.2)$$

where

$$\Omega(k) = 1 + (\Omega_{initial} - 1)\cdot\alpha(k) \qquad (3.3)$$

Here $\Omega_{initial}$ is chosen as *Max (X, Y)* like in commonly used methods. This provides good elasticity of neurons in the very beginning of the training. The other terms are defined in Section 2.1.1.

As it can be noticed, neighborhood function $\beta(k)$ is a Gaussian function, however the width of the function is dependent on the learning rate parameter $\alpha(k)$. In other words, the learning rate has been embedded into the neighborhood function. So, a decrease in learning rate will cause a decrease in the width of the neighborhood function. The new learning rate and neighborhood

function depend on three parameters: $\nu(k)$ (standard deviation of worst matching unit), $\theta(J,k)$ (data impact parameter) and κ (retention parameter). Each of these three parameters has its own contribution to the formation of the topology during training. The following three subsections will describe those parameters and clarify the reasoning behind this formulation

3.2 Standard Deviation of Worst Matching Unit: $\upsilon(k)$

In order to delineate the proposal, an additional notation is described as:

w	: Worst matching unit index,
$M_w(k)$	: Worst Matching Unit (WMU) at iteration step k,
$dd_{Max}(k)$	: Euclidean distance between the training data and WMU,
$\upsilon(k)$	: The standard deviation in $dd_{Max}(k)$ parameter at iteration step k,
$\Omega(k)$	: Radius around the BMU.

SOM methodology provides two basic features to the user. The first one is data reduction by mapping each data in the input space to the most similar reference neuron in net, and the second one is monitoring the physical similarities of data by geometrical relations. Hence the roots of motivation for formulating a new learning rate and neighborhood function parameters are based on the idea that there

had to be statistical similarities between the data and the neurons. Considering this, a parameter is introduced to the learning rate (and consequently to the neighborhood function), which describes the data-topology relations, and changes according to the statistical variations during training. With this parameter, the empirical learning scheme can be converted to an adaptive method without modifying the nature of SOM. Here a parameter *v(k),* standard deviation of the Worst Matching Unit (WMU) has been introduced. In every iteration, the distance between the WMU and the data will contain information on which region data comes and in where the most distant neuron is. The "good" positioning of the neurons will reduce the fluctuations of distance of WMU and this idea is embedded into the learning rate. During training, while calculating the Best Matching Unit (BMU) in the algorithm, the WMU is calculated as:

$$w = \arg\max_{i} \left[\sum_{n=1}^{N} \left(\Lambda_n(k) - m_{i,n}(k) \right)^2 \right], \; where \; i = 1, 2, \ldots, J \qquad (3.4)$$

$dd_{Max}(k)$, the distance between the WMU and the current input vector is a significant measure for estimating topological orientation of neurons during training. It is expected that, when neurons settle to their final locations, the standard deviation (STD) of dd_{Max} shall be small. In the earlier stages of training, however, it will be large, causing easy modification of the whole topology.

Designating the worst matching neuron index as w, $dd_{Max}(k)$ is calculated as:

$$dd_{Max}(k) = \sqrt{\sum_{n=1}^{N} \left(\Lambda_n(k) - m_{w,n}(k)\right)^2} \qquad (3.5)$$

The Standard Deviation $v(k)$ for dd_{Max} is calculated as :

$$v(k) = \sqrt{\frac{1}{k-1}\sum_{j=1}^{k} \left(dd_{Max}(j) - \overline{dd_{Max}(k)}\right)^2} \qquad (3.6)$$

where

$$\overline{dd_{Max}(k)} = \frac{1}{k}\sum_{l=1}^{k} dd_{Max}(l) \qquad (3.7)$$

which is the mean of $dd_{Max}(k)$.

Without adding complexity to the whole process, the standard deviation of $dd_{Max}(k)$ is calculated in an on-line fashion as:

$$v(k) = \sqrt{\frac{1}{k-1}\left(\sum_{j=1}^{k}\left(dd_{Max}(j)\right)^2 - \frac{1}{k}\left(\sum_{j=1}^{k} dd_{Max}(j)\right)^2\right)} \text{ with } k \geq 2, \text{and } v(1) = 0 \quad (3.8)$$

By using only $v(k)$, the learning rate $\alpha(k)$ is defined at this point as:

$$\alpha(k) = \begin{cases} 1, & if \ \dfrac{v(k)}{\sqrt{k}} > 1 \\ \dfrac{v(k)}{\sqrt{k}}, & otherwise. \end{cases} \qquad (3.9)$$

The neighborhood function is calculated according to Eq.(3.2) and Eq. (3.3). The neuron updating scheme remains the same as in Eq.(2.2).

3.2.1 Experimental Results

In order to analyze the contribution of the proposed learning rate and neighborhood function to the SOM algorithm, two criteria have been taken into consideration. The first one is topological convergence, concerning the ordering of neurons, and the second one is the Average Quantization Error (AQE). To analyze the ordering performance, two dimensional input data is used to train the network. Earlier research suggest [Rojas 1996], [Ritter and Schulten 1986], [Erwin et. al 1992a], [Erwin et. al 1992b], [Heskes, 1996], [Flanagan, 1996], that the topological disordering of two dimensional map results in twists (butterflies), and removing them is considered as the transition from the unordered configuration to ordered one. In the following experiments, the map without twist is considered as the ordered map.

The AQE is another criterion to monitor the quality of learning. Lower AQE means better performance of the network for the presented data. The quantization error for an input vector is defined as :

$$d(\Lambda, \boldsymbol{M}_c) = \min_i \{d(\Lambda, \boldsymbol{M}_i)\} \qquad (3.10)$$

where *d()* is a distance measure. In the experiments Euclidean distance is used.

Effects of the proposed learning rate and neighborhood function can be seen in the following experiment. Two dimensional 5x5 neuron map is trained with two dimensional data. The neurons are connected in a rectangular lattice. The training set consisted of 10,000 samples with a normal distribution of Mean = 0, and Standard Deviation = 1. In all training experiments, normally distributed data have been used in line with the similar work in literature. Other distributions have also been tested with similar results but for the sake of brevity, those experiments have not been included in the thesis report. The training set is randomly sampled 10,000 times. Figure 3.1 shows the AQE with step size 100 for various different learning rate parameters, including the proposed one. Here "*Exponential Alpha*", "*Inverse Alpha*," and "*Mulier Alpha*" are mentioned in section 2.1.3. Finally, "*Alpha with STD of WMU*" denotes the learning rate function proposed in Eq. 3.9, above.

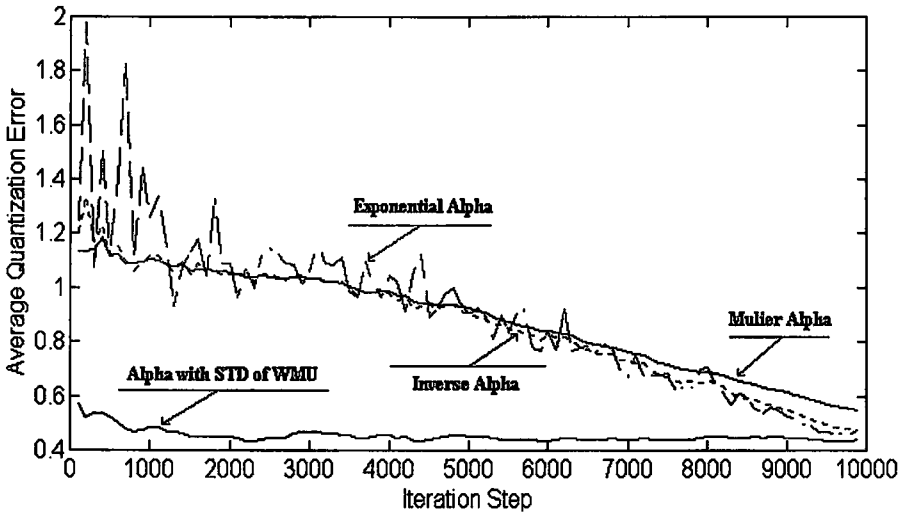


Figure 3.1 AQE for the 5x5 net with 10,000 iterations

In the experiment, topology formation in the early steps of training period demonstrates a remarkably fast convergence rate with the proposed neighborhood function and learning rate parameters compared to the others. Figures 3.2.1 through 3.2.4 show this fast converging phenomenon.

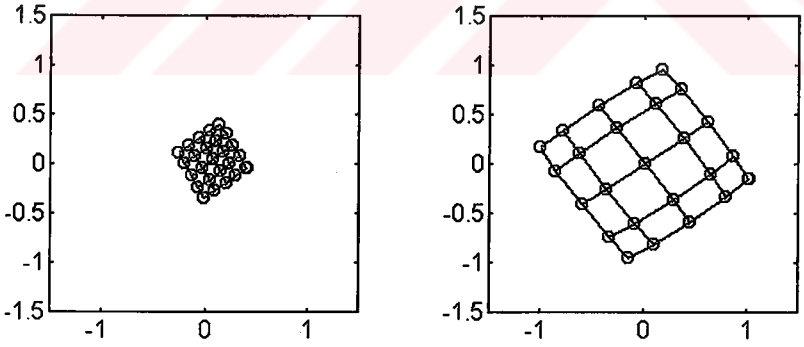


Fig. 3.2.1 Topology formation with *"Mulier Alpha"* learning rate at training steps 4000 and 9500

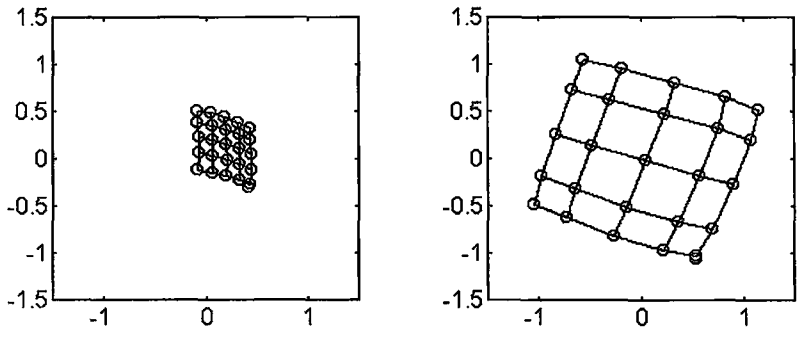


Fig. 3.2.2 Topology formation with *"Inverse Alpha"* learning rate at training steps 4000 and 9500

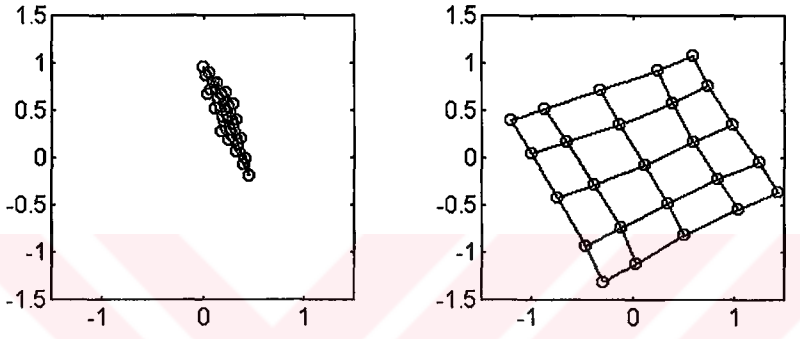


Fig. 3.2.3 Topology formation with *"Exponential Alpha"* learning rate at training steps 4000 and 9500

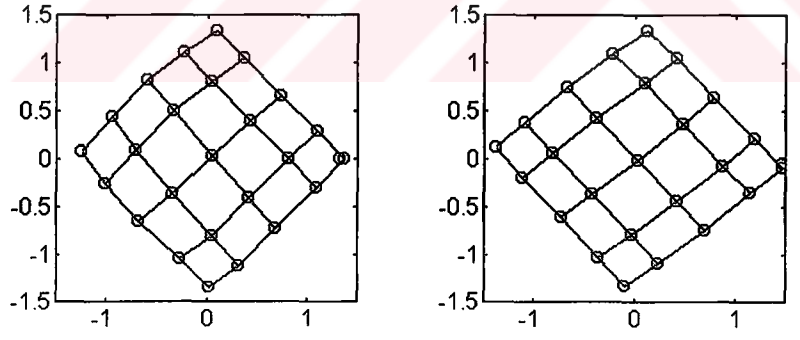


Fig. 3.2.4 Topology formation with Alpha with STD of WMU and the new neighborhood function at training steps 4000 and 9500

In the very early steps of the training period, it is observed that, with the proposed neighborhood function and learning rate, the network finds its perfect topology and there is no need for further training. Figure 3.3 shows the AQE of the same experiment. The only difference is that the number of training iterations is 20,000 instead of 10,000. It is clear that, the conventional learning rates and the neighborhood function are highly dependent on the maximum training size whereas the proposed one is not.

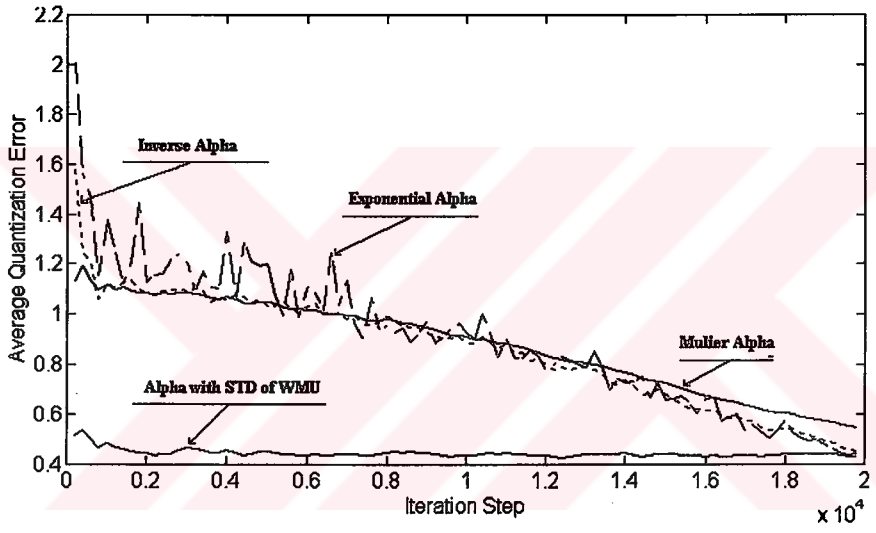


Fig. 3.3 AQE for the 5x5 net with 20,000 iterations

The previous experiments have been carried out for two dimensional data. When the dimensionality was increased, similar results have been obtained. In

Figure 3.4, the AQE results are shown for 5 dimensional input with an 8x8 net. 20,000 data is generated in normal distribution and the net is trained 50,000 times.

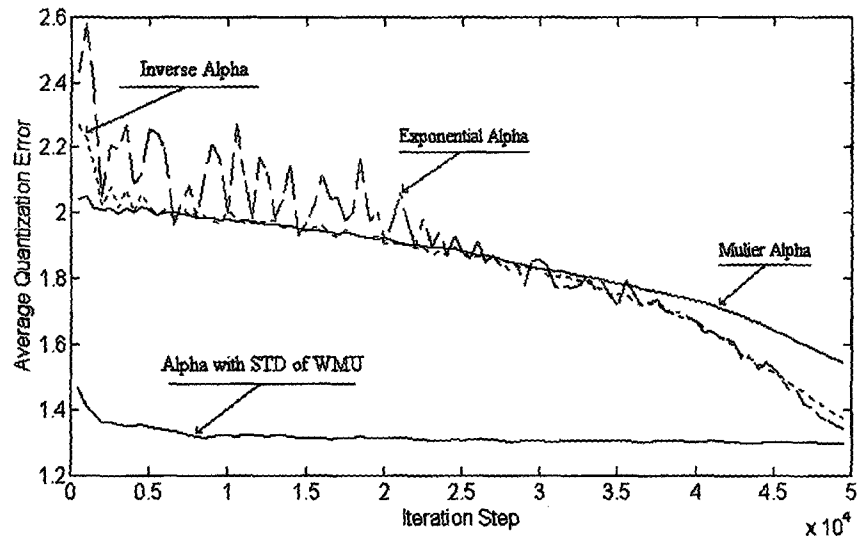


Fig. 3.4 AQE for the 8x8 net with 50,000 iterations

3.3 Data-impact parameter: $\theta(J,k)$

While calculating the learning rate parameter (α) by using *only* $\nu(k)/\sqrt{k}$, it has been observed that the size of the data variance and the size of the network had very strong influences on the network performances. For example, if the variance of data is small, then the globe of the neighborhood function is decreased suddenly and a possible meta-stable state reached at that instant remains stable until the end of training. Also, it is observed that, the size of the network affects the variance of $dd_{Max}(k)$, since the larger the number of neurons, the lower their

fluctuations to reach their stable points during training. Those conditions have to be taken into consideration for determining the new SOM parameters.

In order to obtain better performance, a data-impact parameter, $\theta(J,k)$, is introduced into the learning rate as in Eq. 3.11, and the optimum structure for $\theta(J,k)$ has been investigated.

$$\alpha(k) = \begin{cases} 1, & if \ \dfrac{\nu(k)\cdot\theta(J,k)}{\sqrt{k}} > 1 \\ \dfrac{\nu(k)\cdot\theta(J,k)}{\sqrt{k}}, & otherwise \end{cases} \qquad (3.11)$$

Since analytical determination of the optimum structure of the parameter would be overly cumbersome, an empirical study has been carried out.

In this study, numerous normally distributed data sets have been generated with different standard deviations and each data set has been used to train nets of different sizes. Also, for each training, we need to find the optimum $\theta(J,k)$ value. In order to do this, for each training, with the same data set and net, the experiment was carried out with a broad range of $\theta(J,k)$ values. The range was determined experimentally to include its optimum value. For each $\theta(J,k)$ value, the resultant Average Quantization Error (AQE) was recorded. $\theta(J,k)$ which provides the

smallest AQE is marked as the optimum value for that particular training. An important point observed during the experiments is that, in some situations, better AQE has been obtained with a topologically disordered neuron configuration. For those specific situations, the $\theta(J,k)$, which provides an ordered state in spite of higher AQE, has been marked as optimum $\theta(J,k)$. The algorithm for the experiments is:

1. Fix the size of the net
2. Generate normally distributed data with a mean m, and standard deviation (STD) x.
3. For this data set x, train the network with different values of θ and record the AQE for each training.
4. Find the minimum AQE.
5. If the minimum AQE also provides unfolded topology then mark θ as the best θ, otherwise, find the θ which provides unfolded topology with the closest AQE to minimum AQE
6. Do the steps 2-5 with data sets having different means and STD's
7. Do the steps 1-6 for nets having different number of neurons.

3.3.1 Optimum θ

The results of the experiments for one and two-dimensional nets are given in Figure 3.5. During the experiments it is observed that, the data mean has no effect on optimum value of $\theta(J,k)$ parameter whereas the standard deviation does affect it.

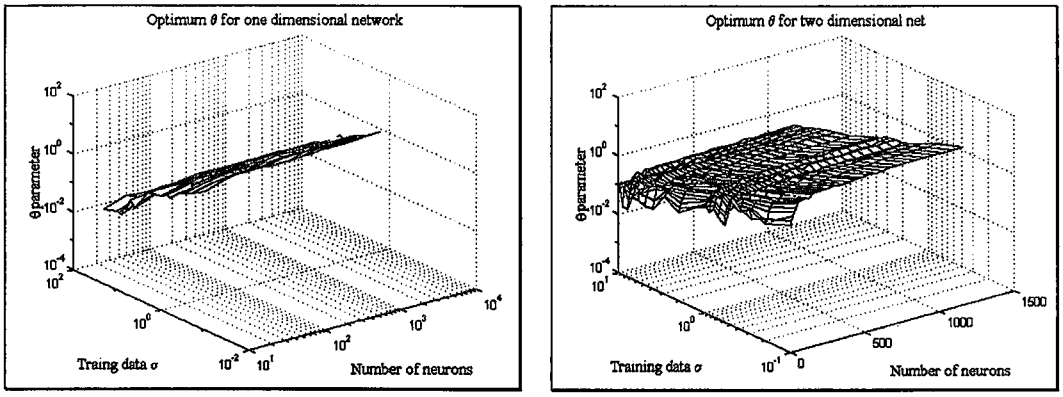


Figure 3.5 Experimentally determined optimum θ for one and two-dimensional nets for normal data

In Figure 3.5, it is clearly seen that the optimum $\theta(J,k)$ parameter varies inversely with the standard deviation of data. Also it was observed that, the size of network affects the optimum $\theta(J,k)$ in an exponentially decaying manner. The following expression for $\theta(J,k)$ was found to represent the experimentally determined optimum θ in the minimum least squares error sense:

$$\theta(J,k) = \frac{(2 - e^{-J/50} \cdot 2) \cdot 1.8}{\sigma(k)} \quad (3.12)$$

where $\sigma(k)$ = Standard Deviation of data at step k.

In Figure 3.6, the results of calculated optimum $\theta(J,k)$ for one and two-dimensional networks are given. The experimental and the calculated $\theta(J,k)$ values are observed to fairly resemble each other.

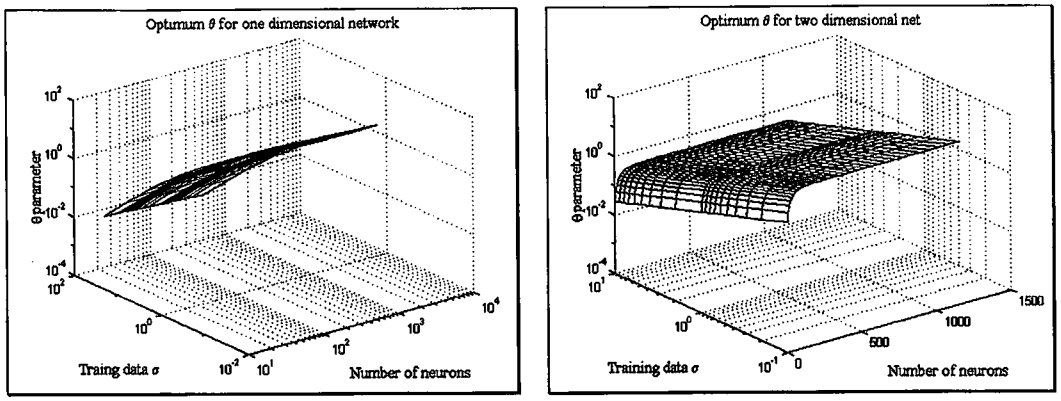


Figure 3.6 Calculated $\theta(J,k)$ for one and two-dimensional nets for normal data

3.3.2 Experimental Results

In order to investigate the influence of the $\theta(J,k)$ parameter on the performance increase, four different learning rate values are compared. Those are:

- The learning rate in Eq.3.9, without $\theta(J,k)$ parameter,
- The new learning rate with $\theta(J,k)$ parameter,
- Exponential Alpha,
- Inverse Alpha.

In the first experiment, two dimensional Gaussian distributed training data set was generated. This set consisted of 10,000 data samples having Mean = 0 and Standard Deviation = 0.4 and was used to train 25x25 net. The maximum

iteration step was chosen as 10,000 and each data in the training set was used only once. The AQE was calculated for step size 20 during the training period and plotted.

Figure 3.7 presents the AQE results of the experiments for those four different types of learning rates and neighborhood functions during training. The performance increase can be clearly seen for the proposed rates. Also it is observed that, by using the proposed learning rate and neighborhood function, AQE decreases at the early stages of the training process. The obvious superiority of the proposal can be observed by investigating the topological formation of the neurons during training.

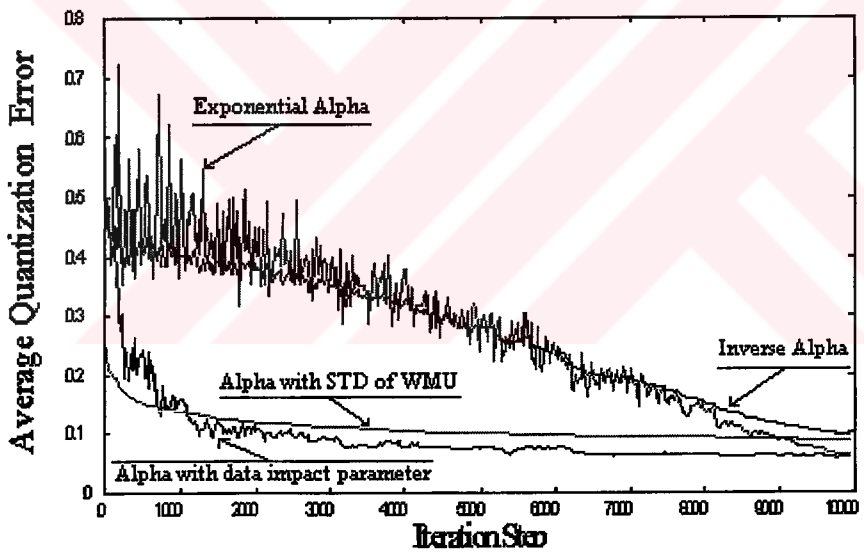
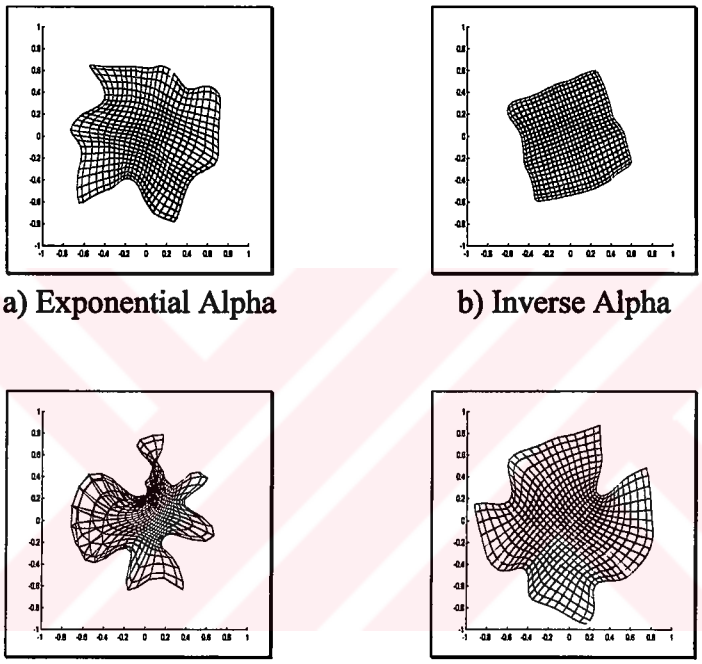


Figure 3.7 AQE for 25x25 net with 10,000 iterations using two dimensional Gaussian data with Mean = 0, Std = 0.4

Figure 3.8, illustrates the results of topological ordering for the experiment at the end of the iteration. It is clearly seen that the new method with $\theta(J,k)$ parameter yields a significant improvement on the behavior of the network. Also there is a remarkable phenomenon that, although a topologically disordered state has been obtained with novel parameters without $\theta(J,k)$, the AQE of this ordering is better than the AQE of training with *inverse alpha.*



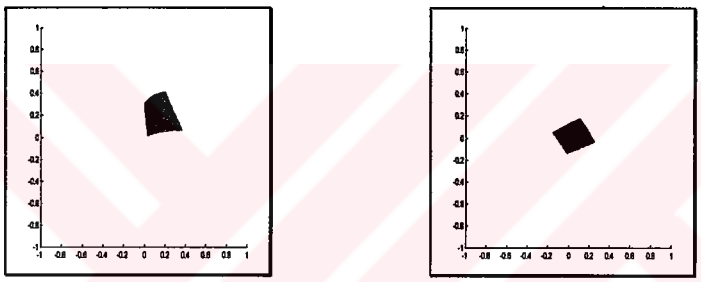
a) Exponential Alpha b) Inverse Alpha

c) Novel Alpha without $\theta(J,k)$ d) Novel Alpha with $\theta(J,k)$

Figure 3.8 The topological ordering of neurons at end of iteration step 10,000

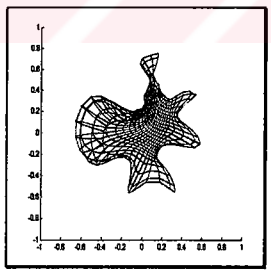
In the experiments conducted, topology formation in the early steps of the training period demonstrates an outstandingly fast convergence rate of the

proposed method. Figure 3.9 shows the topology formation for four different types of learning rate and neighborhood functions in the early stages of the training period. If we compare the results of the topologies in the early stages, we see that although the unfolded topological structure has been obtained with the *exponential alpha* and *inverse alpha*, the neurons are still condensed and need time to expand to find topology with low AQE. The network with the novel parameter without $\theta(J,k)$ has almost reached to the shape of the final topology, but folding on the edges prevents its reliability.

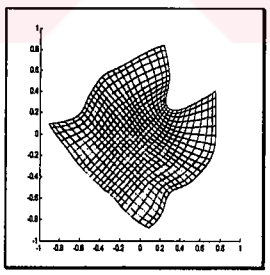


a) Exponential Alpha

b) Inverse Alpha



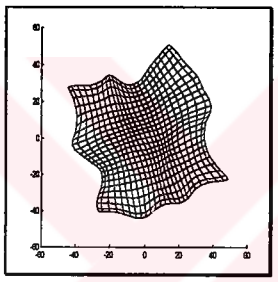
c) Novel Alpha without $\theta(J,k)$



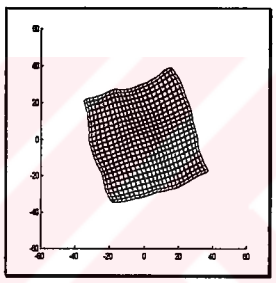
d) Novel Alpha with $\theta(J,k)$

Figure 3.9 The topological ordering of neurons at end of iteration step 3,000

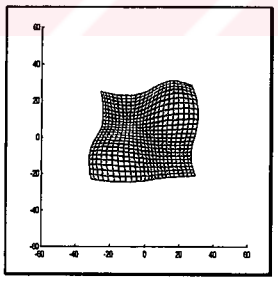
In order to show that the performance of the new method is independent of the variance of data, another experiment was carried out with the same type of neuron configuration (25x25) but with different data set having Standard Deviation of 25 and Mean of 0. The results, which show topological ordering at the end of the iteration and the map formation at the early stages of training, are given in Figure 3.10 and Figure 3.11. It is clearly seen that the value of the standard deviation of data has almost no effect on the topology formation when the proposed method is used although the performance of the net is quite dependent on data variation when other methods are applied.



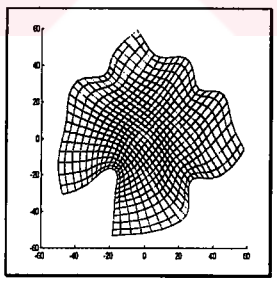
a) Exponential Alpha



b) Inverse Alpha

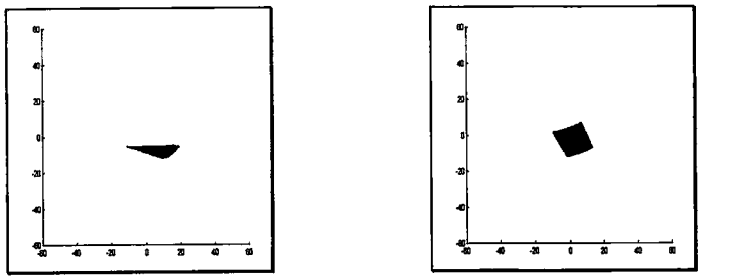


c) Novel Alpha without $\theta(J,k)$

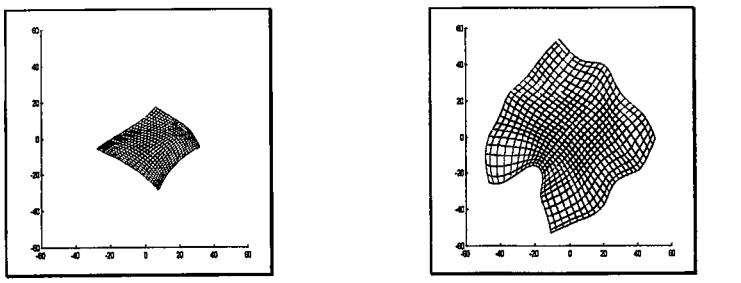


d) Novel Alpha with $\theta(J,k)$

Figure 3.10 The topological ordering of neurons at end of iteration step 10,000



a) Exponential Alpha b) Inverse Alpha



c) Novel Alpha without $\theta(J,k)$ d) Novel Alpha with $\theta(J,k)$

Figure 3.11 The topological ordering of neurons at end of iteration step 3,000

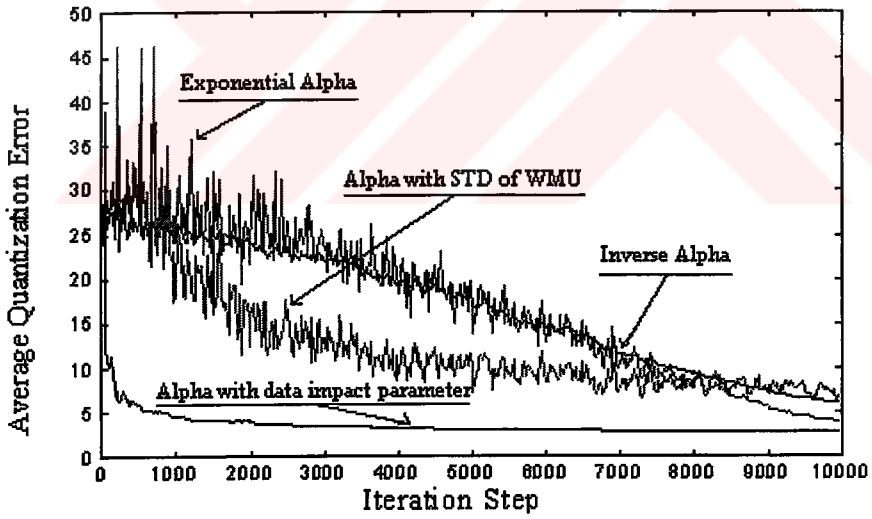


Figure 3.12 AQE for 25x25 net with 10,000 iterations training with two dimensional Gaussian data with Mean = 0, Std = 25

Figure 3.12 represents the AQE for different types of learning rates during training of the 25x25 net with data having Standard Deviation of 25. When we compare the results of the experiment with the previous one given in Figure 3.7, the effects of the $\theta(J,k)$ value can be seen clearly. The statistical characteristic of the data has no effect on training for both Exponential and Inverse learning rates. The performance of the novel learning rate without $\theta(J,k)$ is directly effected with the characteristic of data. Introducing $\theta(J,k)$ solves this problem and fast convergence to topology with small AQE is achieved.

3.3 Retention parameter: κ

The learning rate in Eq. 3.11 without a retention parameter κ, provides a very high learning rate (~1) and broad neighborhood function at the very beginning of training. This provides the global localization of the neurons when training starts. After this initial period of training where global ordering occurs, the learning rate decreases very rapidly and due to this drastic change, the width of the neighborhood function starts to shrink. This is the end of the first phase of training. The second phase in which the neurons converge to their asymptotic values can be referred as the refinement period. The final topology takes shape in this period. According to this phenomenon, the data presented at the early stages of the training phase have a

significant effect on the final topology. In other words, SOM memorizes the initial patterns, than refines the shape of topology. In order to clarify this, an experiment has been carried out. An artificial two-dimensional data set is generated to train two-dimensional 20x20 neurons net. This set consists of two regions as shown in Figure 3.13. The first one is generated as normally distributed with mean of $x = -2$ and $y = -2$. The STD of data is 3.0 in both dimensions. The first 10.000 data samples are chosen from this region. The second region is again generated as normally distributed however, with mean 5.0 for both x and y dimensions and having STD 3.0. The number of data samples from this region is 50.000.

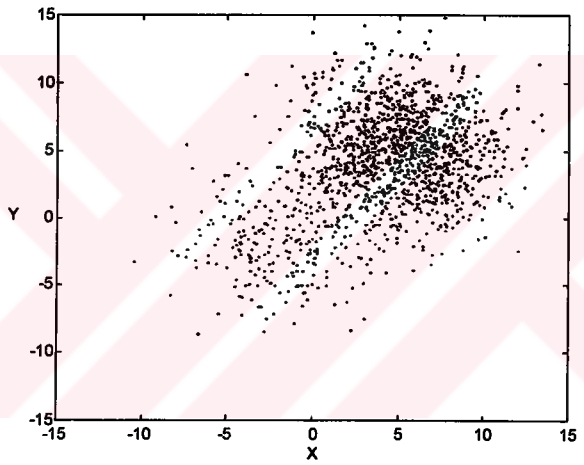


Figure 3.13 The training data set composed of two clusters

The data set is composed of 60.000 samples by concatenating those sets by taking the first 10.000 data from the first region and the remaining samples from the second one. Figure 3.14 shows the topological ordering of 20x20 neurons at

the training step 40.000 after applying the learning rate in Eq. 3.11. It is not difficult to estimate the contribution of data presented at the early stages. It is observed that the neurons have settled to the first region in the early stages of training and the topology has frozen. If the statistics of presented data changes after this moment, adaptation of the net to new data becomes difficult since only minor changes occur in learning rate and neighborhood function. However, if it is desired that the net be able to adapt itself towards the structure of the later data pattern, the algorithm needs some modification.

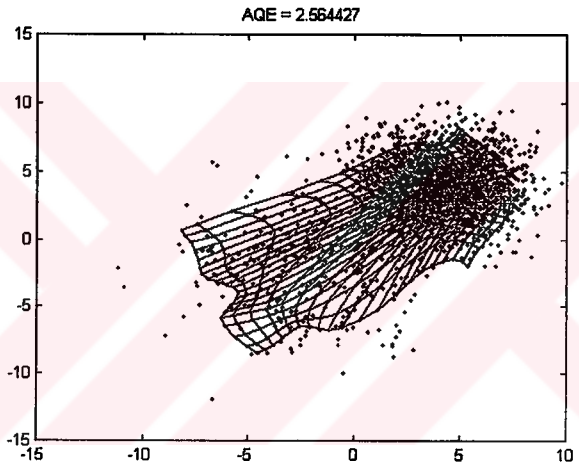


Figure 3.14 20x20 neuron topology after 40.000 training

The learning rate and the neighborhood function are the only parameters that define the whole SOM process. Since in our proposal, the neighborhood function is also based on the learning rate, to follow the statistical

changes in data during training, the learning rate has to change according to those variations. This requires ability to increase the learning rate parameter. The retention parameter κ has been associated with the iteration step parameter k in Eq. 3.11 The iteration step parameter k has monotonically increasing nature and causes the learning rate to diminish gradually.

The main idea of retention parameter κ is to follow the indeterministic and consistent variations in data. Those can be sensed always by tracing the mean of WMU distance. If there is a consistent change in data after a frozen topology, the mean of WMU distance (MWMU) will increase or decrease consistently until a settlement is reached. This settlement means a new frozen topology. In order to recognize a consistent variation in the MWMU during training, a sliding window algorithm is used:

First define Δ_k to represent the direction of change, if any, in the average difference between data vectors and WMU in a window of size q. That is, a window W consists of q such change directions:

$$W = \left(\Delta_{k-q}, \Delta_{k-q+1}, ..., \Delta_{k-1}\right) \quad (3.14)$$

in which

$$\Delta_i = \begin{cases} -1 & if \quad dd_{\max}(i) < \overline{dd_{\max}} \\ 0 & if \quad dd_{\max}(i) = \overline{dd_{\max}} \\ +1 & if \quad dd_{\max}(i) > \overline{dd_{\max}} \end{cases} \qquad (3.15)$$

where

$$\overline{dd_{\max}} = \frac{1}{i}\sum_{j=1}^{i} dd_{\max}(j). \qquad (3.16)$$

For each new data element, Λ(k), compute $dd_{max}(k)$ and compare it with current $\overline{dd_{\max}}$, generating the latest Δ_k and advancing the window by one. At each step, the algebraic sum of all change directions in the window is computed, and if the total positive or negative change is greater than a pre-determined threshold value, κ is decremented so that the α function can be re-sensitized to allow for adaptation to this change.

An additional criterion imposed on the modification of κ to allow only consistent shifts in data statistics to relocate settled neurons is, whether or not the number of data items which fit the consistency condition is larger than the number of data items that have caused the latest neuron settlement to occur.

The algorithm is presented below :

Initialization:

Fix the size (q) of sliding window (W) which keep the values of running MWMU.

```
For i = 0 to q
Begin
    W(i) = 0
End
Previous_Mean = 0
Threshold_Value = 0          /* T */
Change_Direction = 0
Current_Cluster_Size = 1
Previous_Cluster_Size = 0
Mean_Of_WMU = 0              /*Used to calculate MWMU*/
κ = 0
```

Training :

```
(At time step k)
Mean_Of_WMU = WMU(k)/Current_Cluster_Size + WMU(k-1) *
(Current_Cluster_Size-1) / Current_Cluster_Size
/*Mean of WMU is calculated with Current_Cluster_Size*/
If Mean_Of_WMU > Previous_Mean Then
Begin
    Change_Direction = Change_Direction + 1 – W((k+1) mod q)
    W(k mod q) = 1
End
Else If Mean_Of_WMU = Previous_Mean Then
    Begin
    Change_Direction = Change_Direction - 1 - W((k+1) mod q)
    W(k mod q) = 0
End
Else If Mean_Of_WMU < Previous_Mean Then
Begin
    Change_Direction = Change_Direction - 1 - W((k+1) mod q)
    W(k mod q) = -1
End

Previous_Mean = Mean_Of_WMU
If | (Change_Direction) | > Threshold_Value And (Current_Cluster_Size) >
Previous_Cluster_Size
```

$\kappa = \kappa - 1$
Else
$\kappa = \kappa + 1$
Current_Cluster_Size = Current_Cluster_Size + 1

If $\kappa = 0$ Then
Begin
$\kappa = 1$
For $i = 0$ to q
Begin
W(i) = 0
End
Current_Cluster_Size = 1
Previous_Cluster_Size = Previous_Cluster_Size + Current_Cluster_Size / 2
End

Use κ in Learning Rate as :

$$\alpha(k) = \begin{cases} 1, & if \ \frac{\nu(k) \cdot \theta(J,k)}{\sqrt{\kappa}} > 1 \\ \frac{\nu(k) \cdot \theta(J,k)}{\sqrt{\kappa}}, & otherwise \end{cases} \quad (3.13)$$

By using the above algorithm, the same data set of Figure 3.13 is used to train 20x20 neurons network. The map at step 40.000 is shown in Figure 3.15. For the two topologies of the Figure 3.14 and Figure 3.15, the Average Quantization Error (AQE) is calculated for 60.000 data samples. The AQE for the topology trained with retention parameter κ is 1.071447. However the result without the parameter κ is found as 2.564427. This demonstrates a significant improvement in performance.

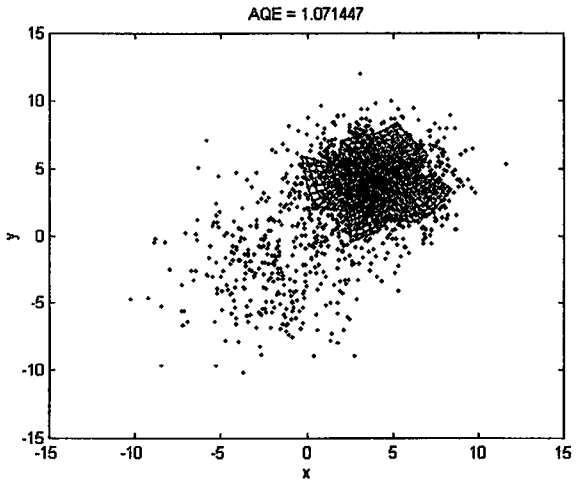


Figure 3.15 Topology of 20x20 neuron after 40.000 training using retention parameter κ

Figure 3.16 presents two graphs of the learning rate during training. The first one represents the training without using the retention parameter κ, and in the second one the retention parameter κ has been used in the learning rate function. The spike around step 22.000 corresponds to the detection of the consistent shift of data statistics from those of the first 10.000 points to the more crowded region. (See Figure 3.13)

Experiments using this algorithm, have also been conducted with data clustered in concave regions and in disconnected multiple sub-regions as well as data with consistent shifts between more than two regions, and successful map formation

has been observed in all cases. For the sake of the brevity, discussion of those experiments have been excluded from this thesis.

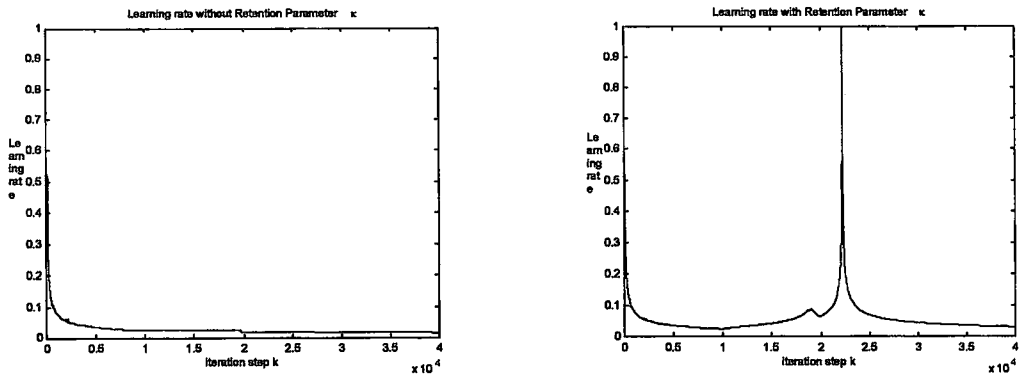


Figure 3.16 Learning rates during training a) Training without retention parameter κ b) Training with κ

CHAPTER 4

CONVERGENCE PROOF OF SOM WITH THE NEW LEARNING RATE AND NEIGHBORHOOD FUNCTION

While proposing or developing a method to improve the performance of the widely accepted NN algorithms, the general aim is to generate or use some ad-hoc approaches since the lack of pure mathematical background of the related disciplines. This chapter, concentrates on the problem of proving the convergence of the proposed learning algorithm in order to enforce the theoretical basis of the method. Also this work helps to understand the nature of the ordering process and effects of the proposed parameters on the shaping of the net.

As it is mentioned in Chapter 2, there are several methods to investigate to understand the nature of the training process by working on convergence criteria of SOM. One method is describing the orders of the neurons as

state vector X and alterations of those state vectors as Markov process. The process is defined on a probability space (Ψ, π) with probability measure π. A sample $\psi \in \Psi$ is a possible sequence of inputs where $\psi = (\Lambda(0), \Lambda(1), \Lambda(2)\ldots)$ which is applied to the neurons with initial state **X**(0). The aim of the proof is to describe the set ψ such that: for any initial configuration of neurons, the organized configuration of $\mathcal{S}$ (see 2.14) for M = D = 1 SOM can be reached with probability one in a finite amount of time τ. Because $\mathcal{S}$ is an absorbing state of the Markov process [see Section 2.2.3], if it is possible to reach this organized configuration, the state remains the same such that:

$$if \;\; X(\tau_{\mathcal{S}}) = \mathcal{S} \Rightarrow X(\tau_{\mathcal{S}} + T) = \mathcal{S} \;\; \forall T > 0 \quad (4.1)$$

The whole process is described as:

$$\pi_{X_o}\{\psi \in \Psi : \tau_{\mathcal{S}} \leq T_{\mathcal{S}}\} \geq \delta_{\mathcal{S}} \qquad (4.2)$$

where X_0 is the initial configuration of the neurons, $T_{\mathcal{S}} < \infty$, and $\delta_{\mathcal{S}} > 0$.

It should be noted that hereafter, the i^{th} neuron will be denoted as m_i as only 1-D net will be considered. (Cf. sec. 2.1.1).

Restriction :

The input data set Σ is continuous, one-dimensional, and bounded with [0, 1] as:

$$\Lambda(k) \in [0,1], \quad (4.3)$$

$$\int_a^b d(\Lambda) > \varepsilon \;\; \text{where } b > a, \; 0 \leq a < 1, \text{ and } 0 < b \leq 1, \; \varepsilon > 0. \quad (4.4)$$

LEMMA 1.

Consider the process defined by (3.1), (3.2) and (3.3) and for any subset $A \subset \Sigma$ which is bounded with A_l and A_h where $A_l < \Lambda(k) < A_h$. If after some time K during training, the data starts to be drawn continually from the region A, all neurons will be localized in region A in a finite time. That is:

$$A = \{\Lambda \in [0,1] : A_l \le \Lambda(k) \le A_h\} \text{ where } k > K \ \exists\ T_K \mid T_K < T_\phi < \infty \text{ such that:}$$

$$A_l \le m_i(T_\phi) \le A_h \text{ where } 1 \le i \le J \qquad (4.5)$$

PROOF:

$\forall k, \alpha(k) > 0, \beta(c,i,k) > 0$ then $|m_i(k) - m_i(k\text{-}1)| > \varepsilon$ where $1 \le i \le J$ and $\varepsilon > 0$. □

DEFINITION 1:

For any time K, the $\alpha_{\min}^K$ is defined as the *possible minimum learning rate* for the time K as:

$$\alpha_{\min}^K = \frac{\nu^K \cdot \left(2 - e^{-\frac{J}{50}} \cdot 2\right) \cdot 1.8}{\sigma_{\max}^K \cdot \sqrt{\kappa_{\max}^K}} \qquad (4.6)$$

where

$$\left.\begin{array}{c}\left\{v^K \mid v^K = v(K)\right\} \\ \left\{\sigma_{\max}^K \mid \sigma_{\max}^K > \sigma(K) \text{ for } t = 1,2,\ldots,K\right\} \\ \left\{\kappa_{\max}^K \mid \kappa_{\max}^K \geq \kappa(t) \text{ for } t = 1,2,\ldots,K\right\}\end{array}\right\} \quad (4.7)$$

LEMMA 2:

$$\sigma_{\max}^K \leq 0.5$$

PROOF :

The maximum std can be obtained if $\Lambda(1) = 1$, $\Lambda(2) = 0$.

$$\therefore std(\Lambda(2)) = \sqrt{\frac{1}{2}\left[(1-0.5)^2 + (0-0.5)^2\right]} = 0.5 \square \quad (4.8)$$

Applying Lemma 2 on (4.6) yields:

$$\alpha_{\min}^K = \frac{v^K \cdot \left(2 - e^{-\frac{J}{50}} \cdot 2\right) \cdot 1.8}{0.5 \cdot \sqrt{\kappa_{\max}^K}} \quad (4.9)$$

Fact : $\forall K, \kappa_{\max}^K \leq K$. (By Algorithm 3.1 pp. 54-55)

By applying the fact above:

$$\alpha_{\min}^K = \frac{v^K \cdot \left(2 - e^{-\frac{J}{50}} \cdot 2\right) \cdot 1.8}{0.5 \cdot \sqrt{K}} = \frac{v^K \cdot \left(2 - e^{-\frac{J}{50}} \cdot 2\right) \cdot 1.8 \cdot 2}{\sqrt{K}} \quad (4.10)$$

DEFINITION 2:

$\forall K$, and for $\alpha_{\min}^K$ as defined in (4.10), two different regions Φ_l^K and Φ_h^K are defined as:

$\Phi_l^K = \{\Lambda \in [0,1] : 0 \le \Lambda < \varphi_K\}$ and

$\Phi_h^K = \{\Lambda \in [0,1] : 1 - \varphi_K < \Lambda \le 1\}$ (4.11)

Also define φ_k as:

$$\varphi_K = \frac{\alpha_{\min}^K \cdot \left[\left(e^{-\frac{(J-2)^2}{2\cdot(1+(J-1)\cdot\alpha_{\min}^K)^2}} \right) - \left(e^{-\frac{(J-1)^2}{2\cdot(1+(J-1)\cdot\alpha_{\min}^K)^2}} \right) \right]}{1 + \alpha_{\min}^K \cdot \left[\left(e^{-\frac{(J-2)^2}{2\cdot(1+(J-1)\cdot\alpha_{\min}^K)^2}} \right) - 2 \cdot \left(e^{-\frac{(J-1)^2}{2\cdot(1+(J-1)\cdot\alpha_{\min}^K)^2}} \right) \right]} \quad (4.12)$$

Also let $\{ \exists\ m_i \mid\ m_i \notin \Phi_l^K \ \textit{and}\ m_i \notin \Phi_h^K \ \}$ *where* $1 \le i \le J$.

PROPOSITION 1:

If data sample at time $(K+\tau+1)$ is drawn such that $\Lambda(K+\tau+1) \in \Phi_h^K$ and if $\alpha(K+\tau) > \alpha_{\min}^K$ and if $0 \le m_i(K+\tau), m_j(K+\tau) < \varphi_K$ and $\mid c-i \mid > \mid c-j \mid$, then

$$m_i(K+\tau+1) < m_j(K+\tau+1) < m_c(K+\tau+1), \quad (4.13)$$

where c = winning neuron index. (See Figure 4.1)

That is: If m_i, m_j and m_c reside in region Φ_l^K and the next data is drawn from the region Φ_h^K with BMU = m_c, then the neurons m_i and m_j will be ordered with respect to m_c regardless of the previous values they have.

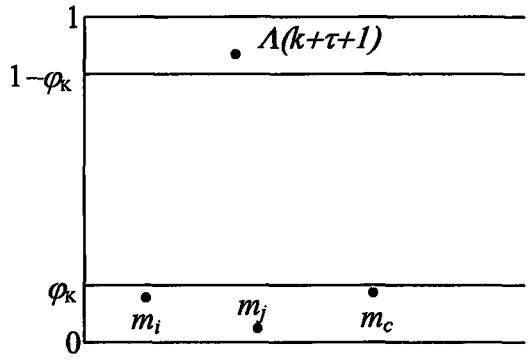


Figure 4.1 The regions defined in time K

Similarly if $m_i(K+\tau) \in \Phi_h^K$ and $\alpha(K+\tau) > \alpha_{\min}^K$ and if

$1-\varphi_K \le m_i(K+\tau), m_j(K+\tau) < 1$ *and* $\Lambda(K+\tau+1) \in \Phi_l^K$ and $|\,c-i\,| > |\,c-j\,|$,

then

$$m_i(K+\tau+1) > m_j(K+\tau+1) > m_c(K+\tau+1). \qquad (4.14)$$

Where c = winning neuron index.

PROOF:

The update equation for the two neurons are:

$$m_j(K+\tau+1) = m_j(K+\tau) + \alpha(K+\tau)\cdot\beta(c,j,(K+\tau))\cdot(\Lambda - m_j(K+\tau)) \quad (4.15)$$

$$m_i(K+\tau+1) = m_i(K+\tau) + \alpha(K+\tau)\cdot\beta(c,i,(K+\tau))\cdot(\Lambda - m_i(K+\tau)) \qquad (4.16)$$

After updating process, there are two possibilities. The first one is $m_i(K+\tau+1) > m_j(K+\tau+1)$ and the second one is $m_j(K+\tau+1) > m_i(K+\tau+1)$. The second possibility is the organized form of the i^{th} and the j^{th} neurons with respect to the c^{th} neuron.

Before the update process, the worst case conditions are:

- The distance between m_i and m_j is maximum. That is:

$$|m_i - m_j| \geq |m_i - m_l| \text{ and } |m_i - m_j| \geq |m_j - m_l| \text{ where } l \neq i \text{ and } l \neq j,\ l = 1,2,...J$$

$$\therefore |m_i - m_j| = \varphi_K \qquad (4.17)$$

- The distance between data Λ and the neurons m_i and m_j is at a minimum:

$$|\Lambda - m_i(K+\tau)| \leq |\Lambda' - m_i(K+\tau)| \text{ and } |\Lambda - m_j(K+\tau)| \leq |\Lambda' - m_j(K+\tau)|$$
$$\text{where } \Lambda \neq \Lambda' \in \Phi_h^K$$

$$\therefore \Lambda = 1 - \varphi_K \qquad (4.18)$$

- The difference between the index values of the neurons m_i and m_j and the winning neuron m_c are maximum:

$$|c-i| + |c-j| \geq |c-n| + |c-l| \quad \text{where} \quad n,l = 1, 2, ..., J$$

$$\therefore \text{if } i < j \Rightarrow |c-i| = J-1 \text{ and } |c-j| = J-2 \qquad (4.19)$$

Considering the worst case conditions described above, the threshold value of the learning rate parameter α that will force the neurons m_i and m_j to become ordered with respect to m_c has to satisfy the following equality:

$$m_i(K+\tau+1) = m_j(K+\tau+1). \qquad (4.20)$$

Thus

$$m_i(K+\tau)+\alpha(K+\tau)\cdot\beta(c,i,(K+\tau))\cdot(\Lambda-m_i(K+\tau))-$$

$$m_j(K+\tau)+\alpha(K+\tau)\cdot\beta(c,j,(K+\tau))\cdot(\Lambda-m_j(K+\tau))=0 \quad (4.21)$$

Rearranging the above equation and using the conditions described above yields:

$$\varphi_K+\alpha_{\min}^K\cdot\beta(c,i,(K+\tau)\cdot(1-2\varphi_K)=\alpha_{\min}^K\cdot\beta(c,j,(K+\tau)\cdot(1-\varphi_K) \qquad (4.22)$$

Substituting the neighborhood function explained in (4.22) results in:

$$\varphi_K+\alpha_{\min}^K\cdot e^{\frac{(J-1)^2}{2\cdot(1+(J-1)\cdot\alpha_{\min}^K)^2}}\cdot(1-2\varphi_K)=\alpha_{\min}^K\cdot e^{\frac{(J-2)^2}{2\cdot(1+(J-1)\cdot\alpha_{\min}^K)^2}}\cdot(1-\varphi_K) \quad (4.23)$$

thus:

$$\varphi_K+\alpha_{\min}^K\cdot\left(e^{\frac{(J-1)^2}{2\cdot(1+(J-1)\cdot\alpha_{\min}^K)^2}}\right)-2\cdot\varphi_K\cdot\alpha_{\min}^K\cdot\left(e^{\frac{(J-1)^2}{2\cdot(1+(J-1)\cdot\alpha_{\min}^K)^2}}\right)= \quad (4.24)$$

$$\alpha_{\min}^K\cdot\left(e^{-\frac{(J-2)^2}{2\cdot(1+(J-1)\cdot\alpha_{\min}^K)^2}}\right)-\varphi_K\cdot\alpha_{\min}^K\cdot\left(e^{-\frac{(J-2)^2}{2\cdot(1+(J-1)\cdot\alpha_{\min}^K)^2}}\right)$$

From the above equation we obtain:

$$\varphi_K = \frac{\alpha_{\min}^K \cdot \left[\left(e^{-\frac{(J-2)^2}{2\cdot\left(1+(J-1)\cdot\alpha_{\min}^K\right)^2}} \right) - \left(e^{-\frac{(J-1)^2}{2\cdot\left(1+(J-1)\cdot\alpha_{\min}^K\right)^2}} \right) \right]}{1+\alpha_{\min}^K \cdot \left[\left(e^{-\frac{(J-2)^2}{2\cdot\left(1+(J-1)\cdot\alpha_{\min}^K\right)^2}} \right) - 2\cdot \left(e^{-\frac{(J-1)^2}{2\cdot\left(1+(J-1)\cdot\alpha_{\min}^K\right)^2}} \right) \right]} \quad (4.25)$$

Hence for any $\alpha(k) > \alpha_{\min}^K$, neuron ordering shall be realized. □

PROPOSITION 2:

After any time k, if data is drawn from the region Φ_l^K continually,

$\exists\ \tau < \infty$ such that $m_i(K+\tau) \in \Phi_l^K$, $\forall i = 1, 2, \ldots, J$.

PROOF:

This result is the direct consequence of Lemma 1. □

THEOREM 1:

During the random sequence of input, there exists a time step K and corresponding $\alpha_{\min}^K$, and a consecutive input sequence which can attract the all neurons in region φ_l^K in time T_φ with learning rate $\alpha(\mathrm{T}_\varphi) > \alpha_{\min}^K$ with non-zero probability as:

$$\Lambda(t) \in \Sigma, \exists K \mid \pi_{X^o}\left\{ \psi \in \Psi : \tau_\kappa \le T_\varphi \right\} \ge \delta_\varphi \quad (4.26)$$

where : $\delta_\varphi \ge 0$, $\mathrm{T}_\varphi < \infty$, $m_i(T_\varphi) \in \left[0, \varphi_l^K\right]$, $\alpha(\mathrm{T}_\varphi) > \alpha_{\min}^K$, *and* $\mathrm{T}_\varphi > K$.

PROOF:

The proof of this theorem consists of defining two phases where in the first phase, the neurons enter into the region φ_l^K and in the second one the retention parameter κ decreases to 1. It is also proven that at the end of second phase, with a nonzero probability, the learning rate parameter takes on a value greater than the minimum learning rate $\alpha_{\min}^K$ necessary for convergence. In order to complete the proof, the following Proposition 3, Proposition 4 and Lemma 3 have to be considered.

PROPOSITION 3 (PHASE 1):

$\exists \delta_\gamma > 0$ *and* $\mathrm{T}_\gamma < \infty$ such that ,

$$\exists K \mid \pi_{X^o}\left\{ \psi \in \Psi : \tau_\gamma \le T_\gamma \right\} \ge \delta_\gamma \qquad (4.27)$$

where : $m_i(T_\gamma) \in \left[0, \varphi_l^K\right]$, *and* $\mathrm{T}_\gamma > K$

PROOF:

After time *K*, drawing the data continually from the region φ_l^K and applying the result of Proposition 2 and considering Restriction 1, the neurons will settle in region φ_l^K .□

PROPOSITION 4 (PHASE 2):

After time T_γ, drawing data continually from the region φ_l^K according to the Rule 1 defined below, and applying the result of Proposition 2 and considering Restriction 1, the neurons will stay in region φ_l^K and retention parameter κ will decrease to 1 at time T_φ.

That is, $\exists \delta_\varphi > 0$ *and* $T_\varphi < \infty$ such that,

$$\exists K \,|\, \pi_{X^T}\{\psi \in \Psi : \tau_\varphi \le T_\varphi\} \ge \delta_\varphi \qquad (4.28)$$

where : $m_i(T_\varphi) \in [0, \varphi_l^K]$, *and* $T_\varphi > K$, *and* $\kappa = 1$.

In this phase, we shall apply the following Rule 1 which will ensure that Lemma 3, below, will hold.

RULE 1:

According to the retention parameter algorithm (Algorithm 3.1), after any time ζ, if there is a permanent increase or decrease in MWMU (Mean of WMU distance) for $2 \cdot \zeta$ time duration, it is certain that the retention parameter, κ, will decrease to 1. This rule will be formulated as follows:

When $k = \zeta + 1$ apply data from the region $\left[0, \frac{\varphi_l^K}{2}\right]$,

When $k = \zeta+2$ apply data from the region $\left[0, \frac{\varphi_l^K}{4}\right]$,

$\vdots$

When $t = \zeta+\tau$ apply data from the region $\left[0, \frac{\varphi_l^K}{2\cdot\tau}\right]$.

Note that this rule ensures monotonic decrease of κ and Lemma 3 follows:

LEMMA 3:

$$\exists\, \delta_\varphi > 0 \text{ and } \mathrm{T}_\varphi < \infty \text{ such that},$$

$$\exists\, K \,|\, \pi_{X^\Gamma}\left\{ \psi \in \Psi : \tau_\varphi \le T_\varphi \right\} \ge \delta_\varphi \qquad (4.29)$$

where : $m_i(T_\varphi) \in [0, \varphi_l^K]$, *and* $\mathrm{T}_\varphi > K$, *and* $\kappa = 1$.

PROOF :

Direct result of Rule 1 and Proposition 4.

Thus, at the end of Phase 2 , at time T_φ, the learning rate parameter $\alpha(\mathrm{T}_\varphi)$ is calculated as :

$$\alpha(\mathrm{T}_\varphi) = \frac{\nu(\mathrm{T}_\varphi)\cdot\omega}{\sigma(\mathrm{T}_\varphi)\cdot 1} \qquad (4.30)$$

where $\omega = \left(2 - e^{-\frac{J}{50}}\cdot 2\right)\cdot 1.8$.

Now we can proceed with the proof of Theorem 1:

The proof of the theorem relies upon on showing $\alpha(T_\varphi) > \alpha_{\min}^K$. That is, using the definition of ν^K in (4.7), we must show that:

$$\frac{\nu(T_\varphi)\cdot\omega}{\sigma(T_\varphi)} > \frac{\nu^K\cdot\omega\cdot 2}{\sqrt{K}}. \quad (4.31)$$

Since by Lemma 2, $\sigma(T_\varphi) \le 0.5$, replacing $\sigma(T_\varphi)$ *with 0.5* , this will be implied if:

$$\frac{\nu(T_\varphi)\cdot\omega}{0.5} > \frac{\nu^K\cdot\omega\cdot 2}{\sqrt{K}} \quad (4.32)$$

This means, it will suffice to show that:

$$\nu(T_\varphi) > \frac{\nu^K}{\sqrt{K}} \quad (4.33)$$

LEMMA 4:

If sample standard deviation value of data x at time t_1 is calculated as $\sigma_{t_1}(x)$, after time t_2 where $t_2 > t_1$, the minimum value of sample standard deviation can be

$$\sqrt{\frac{\sigma_{t_1}(x)\cdot t_1}{t_2}}. \quad (4.34)$$

PROOF:

$$\text{Since } \sigma_{t_1}(x) = \sqrt{\frac{1}{t_1}\left[\sum_{i=1}^{t_1}\left(x(i)-\bar{x}\right)^2\right]} \qquad (4.35)$$

Fact:

$\sigma_{t_2}(x)$ *is minimum* if $x(t) = \bar{x}$ *for* $t_1 < t \le t_2$,

Thus,

$$\min \sigma_{t_2}(x) = \sqrt{\frac{1}{t_2}\left[\sum_{i=1}^{t_1}\left(x(i)-\bar{x}\right)^2 + \sum_{j=t_1+1}^{t_2}\left(x(j)-\bar{x}\right)^2\right]} \qquad (4.36)$$

It is obvious that the second term is equal to zero. Then :

$$\min \sigma_{t_2}(x) = \sqrt{\frac{1}{t_2}\cdot t_1 \cdot \sigma_{t_1}(x)} \quad \square \qquad (4.37)$$

By applying Lemma 4 , inequality (4.33) will follow if we can now show that:

$$\sqrt{\frac{\nu^K \cdot K}{\mathrm{T}_\varphi}} > \frac{\nu^K}{\sqrt{K}} \qquad (4.38)$$

That is:

$$\frac{\nu^K \cdot K}{\mathrm{T}_\varphi} > \frac{\left(\nu^K\right)^2}{K} \qquad (4.39)$$

or:

$$\frac{1}{\nu^K} > \frac{\mathrm{T}_\varphi}{K^2} \qquad (4.40)$$

We know (Lemma 2) that the maximum possible value of ν^K is 0.5.

Hence if it is possible to find a T_φ which satisfies:

$$T_\varphi < 2 \cdot K^2 \quad (4.41)$$

then the proof shall be complete.

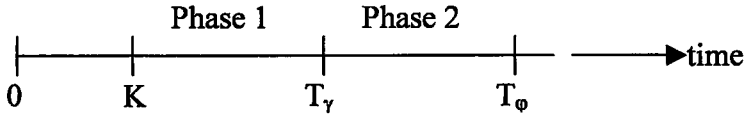


Figure 4.2 Two phases for the Theorem 1

The retention parameter algorithm ensures that, after time *t*, if there is continual increase or decrease in the mean value of the MWMU distance, that is in phase 2 (Figure 4.2), the retention parameter κ, will decrease to 1 after 2*t* training steps. Therefore:

$$Max(T_\varphi) = 2 \cdot (K + T_\gamma) \quad (4.42)$$

That is, any T_φ less than this maximum will definitely satisfy (4.41), if we have

$$T_\gamma < K^2 - K \quad (4.43)$$

for some *K*. But as by Lemma 1, T_φ is finite, such a *K* can always be found. This completes the proof of Theorem 1. □

With Theorem 1 it has been proven that for any initial configuration X(0), there exists a time step K for which $\alpha_{\min}^{K}$ and the regions Φ_l^K and Φ_h^K are defined. Now it will be shown that, if data comes from the region Φ_h^K at time $T_\varphi+1$, neurons will be partially or fully organized.

LEMMA 5:

After, the neurons are settled according to Theorem 1, in region $[0, \varphi_K]$, if $\Lambda(T_\varphi + 1) \in \Phi_h^K$ one of the following three configurations of neurons will result:

Configuration 1 : (Fully organized, descending)

$$\{\forall i, j \mid m_1 > m_2 > \ldots > m_i > m_j > \ldots > m_J\} \qquad (4.44)$$

where $i \neq j$ *and* $i < j$ *and* $i, j = 1, 2, \ldots J$

Configuration 2 : (Fully organized, ascending)

$$\{\forall i, j \mid m_1 < m_2 < \ldots < m_i < m_j < \ldots < m_J\} \qquad (4.45)$$

where $i \neq j$ *and* $i < j$ *and* $i, j = 1, 2, \ldots J$

Configuration 3 : (Partially organized)

$$\{\forall i, j, o, p, r \mid m_1 < m_2 < \ldots < m_i < m_j < \ldots < m_o > \ldots m_p > m_r > \ldots > m_{J-1} > m_J\}$$

where $i \neq j \neq o \neq p \neq r$ *and* $i, j, o, p, r = 1, 2, \ldots J$ *and* $i < j$ *and* $p < r$ (4.46)

PROOF :

By applying the results of Proposition 1 and Theorem 1, Configuration 1 can be obtained if $m_c(\mathrm{T}_\varphi + 1) = m_1(\mathrm{T}_\varphi + 1)$. In other words if the best matching unit (BMU) index is 1 at time $(\mathrm{T}_\varphi + 1)$ then the fully organized descending configuration is reached.

Similarly if $m_c(\mathrm{T}_\varphi + 1) = m_J(\mathrm{T}_\varphi + 1)$ then Configuration 2 can be obtained and this is the fully organized ascending configuration.

In *Configuration 3,* the neurons are partially organized and this configuration can occur if $m_c(\mathrm{T}_\varphi + 1) = m_o(\mathrm{T}_\varphi + 1)$ where $o \neq 1$ *and* $o \neq J$. □

If *Configuration 1* or *Configuration 2* were the only configurations then the proof of convergence of the proposed method can be obtained at this point. However due to the existence of *Configuration 3,* we need to consider the following final phase.

According to Lemma 5 :

$$\Lambda(t) \in \Sigma, \exists K \,|\, \pi_{X^o}\left\{ \psi \in \Psi : \tau_\kappa \leq T_{\varphi+1} \right\} \geq \delta_{\varphi+1} \qquad (4.47)$$

where : $\delta_{\varphi+1} \geq 0$, $\mathrm{T}_{\varphi+1} < \infty$, $X(T_{\varphi+1}) \in \Delta$ *and* $\mathrm{T}_\varphi > K$,

in which $\Delta = Configuration1 \cup Configuration2 \cup Configuration3$.

LEMMA 6:

If the neurons are partially ordered as:

$$\{\forall i,j,o,p,r \mid m_1(t) < m_2(t) < \ldots < m_i(t) < m_j(t) < \ldots < m_o(t) > \\ > \ldots m_p(t) > m_r(t) > \ldots > m_{J-1}(t) > m_J(t)\} \quad (4.48)$$

and if $\{\forall\ \Lambda(t+1) \in \Sigma \mid \Lambda(t+1) > m_o(t)\}$, the BMU will be $m_o(t+1)$ at time $t+1$, after updating process, the neurons ordering will not change as :

$$\{\forall i,j,o,p,r \mid m_1(t+1) < m_2(t+1) < \ldots < m_i(t+1) < m_j(t+1) < \ldots < m_o(t+1) > \\ > \ldots m_p(t+1) > m_r(t+1) > \ldots > m_{J-1}(t+1) > m_J(t+1)\} \quad (4.49)$$

PROOF :

Since neurons $m_1, \ldots m_i, mj, \ldots m_c$ are ordered as

$$m_1(t) < \ldots < m_i(t) < m_j(t) \ldots < m_o(t) \quad (4.50)$$

where $2 \le i < j < c$

and when the BMU is $m_o(t)$ at any time t, it is obvious that the ordering remains the same.

Similarly since the neurons $m_c, \ldots m_p, m_r, \ldots m_J$ are ordered as

$$m_c(t) > \ldots > m_p > m_r(t) > \ldots > m_J(t) \quad (4.51)$$

where $c < p < r < J$

and when the BMU is $m_o(t)$ at any time t, it is obvious that the ordering remains the same.

It is possible to define the partially ordered configuration in (4.48) by (4.50) and (4.51). Thus for any time t, if the BMU is $m_c(t)$, then the partially ordered configuration remains the same.□

THEOREM 2 :

If the neurons are partially organized at step $T_\varphi+1$ it is possible to find a $\alpha_{\min}^{T_{\varphi+1}}$ and $\Phi_l^{T_{\varphi+1}}$ and $\Phi_h^{T_{\varphi+1}}$. Also

$$\Lambda(t) \in \Sigma, \exists T_{\varphi+1} \mid \pi_{X^{\varphi+1}}\left(\{\psi \in \Psi : \tau_\kappa \le T_{\tilde{s}-1}\}\right) \ge \delta_{\tilde{s}-1} \quad (4.52)$$

where : $\delta_{\tilde{s}-1} \ge 0$, $T_{\tilde{s}-1} < \infty$, $m_i(T_{\tilde{s}-1}) \in \left[1-\varphi_h^{T_{\varphi+1}}1\right]$, $\alpha(T_{\tilde{s}-1}) > \alpha_{\min}^{T_{\varphi+1}}$, *and* $T_{\tilde{s}-1} > T_{\varphi+1}$

PROOF :

The proof of Theorem 2 is similar to the proof of Theorem 1. In order to proceed, it is necessary to apply the same steps, however the attraction region has to be changed as:

In Theorem 1 the region of interest was: $[0, \varphi_K]$

In Theorem 2 the region of interest was: $[1, 1-\varphi_{T_{\varphi+1}}]$.

By Lemma 6, the neurons reside in the region $[1,1-\varphi_{T_{\varphi+1}}]$ in a partially organized manner as:

$$\{\forall i,j,o,p,r \mid m_1(\mathcal{S}-1) < m_2(\mathcal{S}-1) < \ldots < m_i(\mathcal{S}-1) < m_j(\mathcal{S}-1) < \ldots < m_o(\mathcal{S}-1) > \\ > \ldots m_p(\mathcal{S}-1) > m_r(\mathcal{S}-1) > \ldots > m_{J-1}(\mathcal{S}-1) > m_J(\mathcal{S}-1)\} \quad (4.53)$$

where $1-\varphi_{T\varphi+1} \le m_1 \le 1$ and $1-\varphi_{T\varphi+1} \le m_J \le 1$.□

THEOREM 3 :

$$\pi_{X_o}\{\ \psi \in \Psi : \tau_{\mathcal{S}} \le T_{\mathcal{S}}\} \ge \delta_{\mathcal{S}} \qquad (4.49)$$

where $X(T_{\mathcal{S}}) \in \mathcal{S}$

PROOF:

If the neurons reside in the region $[1,1-\varphi_{T_{\varphi+1}}]$, than if a data item drawn from the region $[0,\varphi_{T_{\varphi+1}}]$, the best matching unit (BMU) index will be either 1 or *J*. This immediately forces the neurons to be ordered. This condition occurs with probability $\delta_{\mathcal{S}} > 0$. □

The neuron configurations within the successive phases considered in this proof of the convergence are displayed in Figure 4.3. Figure 4.3.a shows the initial random state X(0) with 10 neurons. After data are drawn repeatedly from the region Φ_l^K the neurons reside in [0, φ_k] as shown in Figure 4.3.b. In Figure 4.3.c it is shown that the data Λ(k+1) is drawn from the region $[1-\varphi_K]$ and the 10 neurons are

ordered partially. In Figure 4.3.d, the new regions $[0,\varphi_{T_{\varphi+1}}]$ and $[1-\varphi_{T_{\varphi+1}},1]$ are shown in which partially ordered neurons are settled. In Figure 4.3.e, it is shown that the data item $\Lambda(\delta)$ is drawn from the region $[0,\varphi_{T_{\varphi+1}}]$ and 10 neurons are fully ordered.

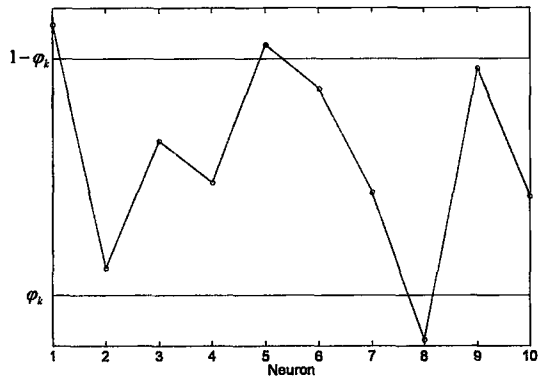


Figure 4.3.a

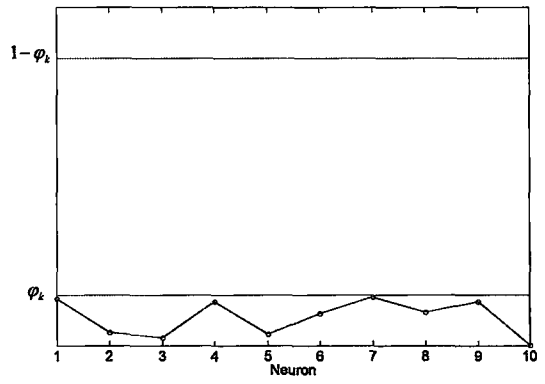


Figure 4.3.b

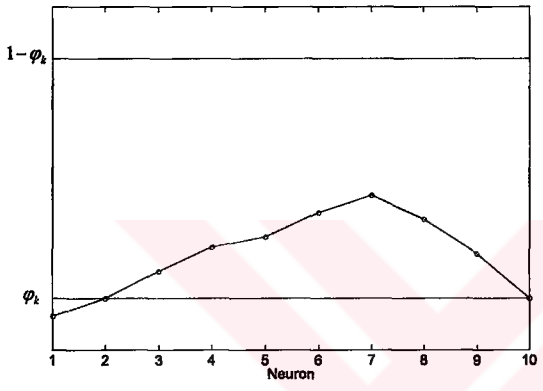


Figure 4.3.c

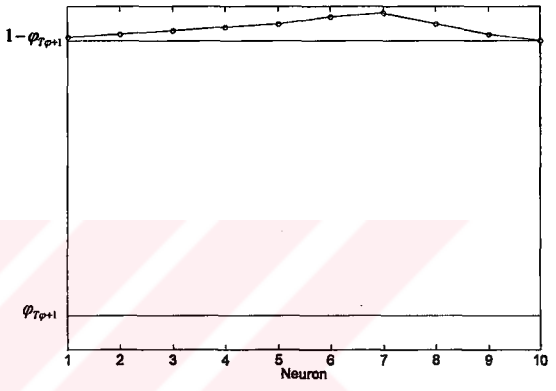


Figure 4.3.d

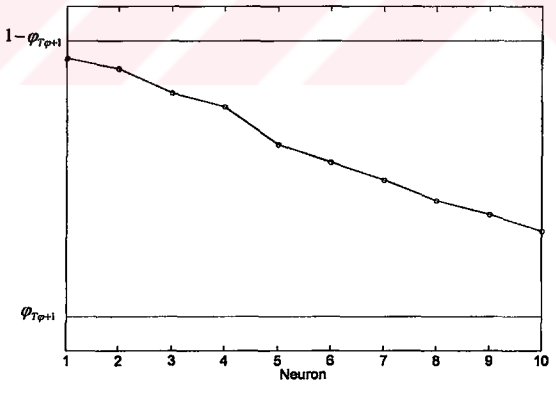


Figure 4.3.e

Figure 4.3. The steps of the convergence proof

CHAPTER 5

CONCLUSION

This thesis has presented a novel technique to incorporate a statistical approach to the classical SOM methodology with the aim of obtaining faster map convergence and sensitivity to data characteristics. This removes the need for an a priori definition of a maximum number of iterations in the training algorithm. The learning rate and neighborhood function are defined in terms of data and network statistics to achieve this adaptability.

In parallel with the formulation of the proposal, the developed methodology is applied to a number of examples and comparisons with earlier SOM implementations are presented. It is shown that both the average quantization error and the convergence time are much lower than those achieved with earlier methods.

Moreover, the proposed method avoids map convergence to "folded" topologies provided that data and map dimensionalities are comparable.

The proposed methodology relies on to define the learning rate and neighborhood function according to the statistics of the WMU (Worst Matching Unit). which is calculated at the same time of finding BMU (Best Matching Unit). The standard deviation of the distance of WMU is introduced into the learning rate parameter. Also the standard deviation of data and the continual alterations in the distance of WMU is used to define the learning rate. This parameter also is used to calculate the neighborhood function.

Convergence of the developed method is analyzed for one-dimensional data and map, and a proof based on a Markovian analysis is given.

While not explicitly discussed in the thesis, it should be noted that implementation of the method, based on earlier references [Germen and Bilgen 1997], [Germen and Bilgen 1998] have been favorably reported [Avkaroğulları and Çiloğlu 1998 a], [Avkaroğulları and Çiloğlu 1998 b]. Those authors have applied this proposed method on a problem of speech coding. In those works, to calculate the secondary excitation formulations for CELP (Code Excited Linear Predictive) coders, SOM technique has been used. In those papers it has been noted that, the

performance of the conventional SOM algorithms was not satisfactory however the proposed method gives quite satisfactory results.

An explicit stopping criterion has not been formulated within the scope of this thesis. Even though it has been shown that fast convergence to correct topologies will be established, a minor step forward, this may be considered as the subject of possible future work using method based on "goodness of map" [Kaski and Lagus 1996], [Ypma and Duin 1997] or average quantization error.

Another suggestion for future work would be to adapt this method to a batch SOM. The current work has focused only on the flow-through SOM methodology, whereas the batch methodology is generally accepted to provide a higher level of data representation. [Kohonen 1993c], [Kohonen 1995].

Similarly such a statistical approach may be applied on learning vector quantization (LVQ) problems. Also the general vector quantization (VQ) problem can be investigated under the light provided by this method since VQ method can be considered as the batch mode of the SOM algorithm with neighborhood 1 during the training process [Kohonen 1993c], [Kohonen 1995].

Naturally, use of the method in real-life SOM applications will provide much more realistic evaluations of its effectiveness and efficiency.

REFERENCES

Avkaroğulları, G., Çiloğlu T., 1998 a, Increasing Quality of Celp Coders by Source-Filter Interrelation Using Self Organising Maps, EUSIPCO (European Signal Processing Conference) 98, Rhodes Islands-Greece, September 8-11,1998, Vol. III, pp:1417-1420

Avkaroğulları, G., Çiloğlu T., 1998 b, Kaynak-Süzgeç İlişkisinin Kendinden Düzenlemeli Haritalarda Kullanılması ile CELP Kodlayıcıların Niteliğinin Artırılması, SIU '98, 6. Sinyal İşleme ve Uygulamaları Kurultayı, Kızılcahamam, Ankara, 28-30 Mayıs 1998, sayfa 25-30

Bauer, H., U., Pawelzik, K., R., 1992, Quantifying the Neighborhood Preservation Of Self-Organizing Feature Maps, IEEE, Trans. Neural Network, Vol.3, No. 4, pp. 570-579.

Bauer, H., U., Villmann, T.,1997, Growing a hypercubical output space in a self-organizing feature map, IEEE Transactions on Neural Networks, 8(2), pp. 218-226.

Buzo, A., Gray, A., Jr., Gray, R., Markel, J., 1980 Speech coding based upon vector quantization, IEEE Trans. Acoustics. Speech and Signal. Processing, Vol. ASSP 28, No. 5, pp 562-574.

Carpenter, G., Grossberg, S., 1986, Adaptive resonance theory: Stable self-organization of neural recognition codes in response to arbitrary lists of input patterns, 8th Annual Conference of the Cognitive Science Society, Hillsdale, NJ., pp. 45-62.

Carpenter, G., Grossberg, S., 1987, ART2: Self-organization of stable category recognition codes for analog input patterns, Applied Optics, Vol., 26, pp. 4919-4930.

Cherkassky, V., Lari-Najafi, 1991, Self organizing neural network for non-parametric regression analyses, Proc. INNJ, Paris, France, pp. 370-374.

Cherkassky, V., Gehring, D., Mulier, F., 1996, Comparison of Adaptive Methods for Function Estimation from Samples, IEEE, Trans., Neural Networks, Vol. 7, No. 4, pp, 969-984.

Cosi, P., Poli, G., Lauzzana, G.,1994, Timbre Classification by NN and Auditory Modelling, Proc. ICANN 94.

Erwin, E., Obermayer, K., Schulten, K., 1992 a, Self organizing maps: ordering convergence properties and energy functions, Biological Cybernetics, Vol. 67, pp. 47-55.

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

Flanagan, J., A., 1996, Self-organization in Kohonen's SOM, Neural Networks, Vol 9, No 7, pp 1185-1197.

Flanagan, J., A., 1997, Self-Organization in the One-Dimensional SOM with a Reduced Width Neighborhood, Proc. VSOM 97 Helsinki.

Freeman, James, A., Skapura, David, M., 1991, Neural Networks Algorithms, Applications and Programming Techniques, Addision-Wesley Publishing Company.

Fritzke, B., 1992, Growing Cell Structures a Self-Organizing Network in k Dimensions, Artificial Neural Networks, 2, Vol. II, pp. 1051-1056, North-Holland,

Germen, E., Bilgen, S., 1997, A Statistical Approach to Determine the Neighborhood Function and Learning Rate in Self-Organizing Maps, Proc. ICONIP'97, Dunedin New Zealand, Springer, pp.334-337 .

Germen, E., Bilgen, S., 1998, STATSOM, Statistical Self-Organizing Map, Proc. NC'98, Vien, Austria, pp. 117-122.

Herrman, M., 1995, Self-Organizing Feature Maps with Self-Organizing Neighborhood Widths, Proc. ICNN'95 Perth, http://www.gwdw.de/~mherrma/paper.htm.

Heskes, T., M., 1996, Transition timed in self-organizing maps, Biological Cybernetics, Vol. 75, pp. 49-57.

Hopfield, J., Tank, D., 1985, "Neural" computation of decissions in optimization problems, Biplogical Cybernetics, Vol.52, pp. 141-152.

Haykin, S., 1994, Neural Networks A Comprehensive Foundation, *Macmillan College Pub. Co.*, 1994.

Hinton, G., Ackley, D., Sejnowsky, T., 1984, Boltzmann Machines: Constraint satisfaction networks that learn, Carnegie-Melon University, Dept. of Computer Science Technical Report, CMU-CS-84-119.

Kangas, J., 1994, On The Analysis of Pattern Sequences by Self-Organizing Maps, Ph. D. Thesis, Helsinki University of Technology.

Kaski, S,. Lagus, K., 1996, Comparing Self-Organizing Maps, Proc. ICANN96, International Conference on Artificial Neural Networks, Lecture Notes in Computer Science, Vol. 1112, pp. 809-814.

Kleinrock, L., 1975, Queuing Systems, Wiley, 1975.

Kohonen, T., 1981, Automatic formation of topological maps of patterns in a self-organizing system, Proceedings of Second Scandinavian Conference on Image Analysis, Espoo, Finland, Suomen Hahmontunnistustutkimuksen Seura, pp 214-220.

Kohonen, T., 1982 a, Analyses of Simple Self-Organizing Process, Biological Cybernetics, Vol.44, pp. 135-140.

Kohonen, T., 1982 b, Self-Organized Formation of Topologically Correct Feature Maps, Biological Cybernetics, Vol.43, pp. 59-69.

Kohonen, T., 1988 a, An Introduction to Neural Computing, Neural Networks, Vol 1, pp. 3-16.

Kohonen, T., 1988 b, Self Organization and Associative Memory, Springer Verlag, Berlin.

Kohonen, T., 1988 c, The "Neural" Phonetic Typewriter, Computer, Vol 21, pp. 11-22.

Kohonen, T., 1990, The Self Organizing Map, Proceedings of IEEE, Vol. 78, No 9, pp 1464-1480.

Kohonen, T., 1993a, Generalizations of the Self-Organizing Map, Proc. 1993 International Joint Conference on Neural Networks, pp. 457-462.

Kohonen, T., 1993b, Physiological Interpretation of the Self-Organizing Map Algorithm, Neural Networks, 6, pp. 895-905.

Kohonen, T., 1993c, Things You Haven't Heard about the Self-Organizing Map, Proc. of 1993 IEEE International Conference on Neural Networks, San Francisco, California, USA, March 28 - April 1, 1993, pp. 1147-1156.

Kohonen, T., 1995, Self Organization Map, Springer Verlag, Berlin.

Kohonen, T., Hynninen, J., Kangas, J., Laaksonen, 1995, J., SOM_PAK Self-Organizing Map Program Package Ver 3.1 Helsinki University of Technology, http://www.cis.hut.fi/nnrc/nnrc-programs.html.

Koikkalainen, P., 1993, Fast Organization of the Self-Organizing Map, Proc. Symp. on Neural Networks in Finland, Finnish Artificial Intelligence Society pp. 51-62.

Koikkalainen, P., 1995, Fast Deterministic self-organizing maps, Proc. ICANN'95, Int. Conf. on Artificial Neural Networks, Vol. II, pp. 63-68.

Kosko, B., 1987, Adaptive Bidirectional Associative Memories, Applied Optics, 26, pp. 4947-4960.

Lampinen, J., Oja, E., 1989, Self-Organizing Maps For Spatial And Temporal AR Models, Proc. The 6'th Scandinavian Conference On Image Analysis, Oulu Finland, pp. 120-127.

Linde, Y., Buzo, A., and Gray, R..,M., 1980, An Algorithm for Vector Quantizer Design, IEEE Trans. Communication., Vol. 28, Pp84-95.

Ljung, L., 1977, Analyses of recursive stochastic algorithms, IEEE Transaction on Automatic Control, Vol. AC-22. No. 4, pp. 551-575.

Lo, Z., Yu, Y., 1993, Bavarian, B., Analyses the converging properties of topology preserving neural networks, IEEE Trans. Neural Networks, Vol. 4, No. 2, pp. 207-220.

Martinetz, T., Schulten, K., 1994, Topology Representing Networks, Neural Networks, Vol. 7, No. 3, pp. 507-522.

Mononen, J., Hakkinen, E., Koikkalainen, P., 1995, Customer Analysis Through Self-Organizing Map, Proc. ICANN'98, Paris France.

Mulier, F., M., Cherkassky V., 1995 Statistical Analyses of Self-organization, Neural Networks, Vol. 8, No. 5, pp. 717-727.

Mulier, F., M., 1994, Ph. D. Thesis, University of Minnesota.

Ritter, H, Schulten, K.,1986, On the stationary state of Kohonen's Self-Organizing Sensory Mappings, Biological Cybernetics, Vol. 60, pp. 59-71.

Ritter, H., Schulten, K.,1987, Extending Kohonen's Self-Organizing Mapping Algorithm to Learn Ballistic Movements, Neural Computers, Springer Verlag pp. 393-406.

Rojas, R., 1996, Neural Networks, A Systematic Approach, Springer.

Rumelhart, D., Hinton, G., Williams, R., 1986, Learning representations by backpropagating errors, Nature, 323, pp. 533-536.

Simpson, P., K., 1990, Artificial Neuron Systems, Foundations, Paradigms, Applications and Implementations, Pergamon Press, 1990.

Stark, H., Woods, J., W., 1994, Probability, Random Processes and Estimation Theory for Engineers, Prentice Hall Inc., Second Edition.

Sutton, R., S., Bartı, A., G., Williams, R., J., 1991, Reinforcement learning is direct adaptive optimal control, Proceedings of the American Control Confgerence, Boston, MA, pp. 2143-2146.

Thiran, P., Hasler, M.,1994, Self-organization of a one-dimensional Kohonen Network with quantized weights and inputs, Neural Networks, Vol. 7, No. 9, pp. 1427-1439.

Tolat, V., V., 1990, An analyses on Kohonen's self-organizing maps using a system of energy functions, Biological Cybernetics, Vol, 64, pp. 155-164.

Villmann, T., Der, R., Herrmann, M., Martinez, T. M., 1997, Topology Preservation in Self-Organizing Feature Maps: Exact Definition and Measurment, IEEE Trans. Neural Networks, Vol. 8, No. 2, pp. 256-266.

Widrow, B., Hoff, M., E, Jr., 1960, Adaptive switching circuits, IRE WESCON "Convention Record", pp. 96-104.

Widrow, B., Gupta, N., Maitra, S., 1973, Punish/reward: Learning with a critic in adaptive threshold systems, IEEE Trans. on Systems, Man and Cybernetics, SMC-5, pp. 455-465.

Yin, H., Allinson, N., M., 1993, Statistical Analysis and Treatment of Kohonen's Self-Organizing Map, Tech. Rep., Image Eng. Lab. University of York UK,.

Yin, H., Allinson, N., M., 1995, On the Distribution and Convergence of Feature Space in Self-Organizing Maps, Neural Computation, Vol. 7, No. 6, pp. 1178-1187.

Ypma, A., Duin, R.,P.,W., 1997, Novelty Detection Using Self-Organizing Maps, Proc. ICONIP'97 Dunedin New Zealand, Springer Verlag, pp. 1322-1326.

VITA

Emin Germen was born in Ankara March 15 1966, He has received his B.S and M. S. degrees, both in Electrical and Electronics Engineering from the Middle East Technical University in July 1987 and September 1991 respectively. He has worked in the same department between 1988-1996 as an assistant. Since 1996, he has been an instructor in Electrical and Electronics Engineering Department in Anadolu University. In 1995-1996 he has worked in a research laboratory in Lappeenranta in Finland on Tree-Structure Self-Organizing Map project with Prof. Dr. Pasi Koikkalainen.

PUBLICATIONS :

1. Germen, E., Bilgen, S., 1991, Melodinin Bilgisayar Yardımı ile Notasının Çıkartılması, Elektrik Mühendisliği 4. Ulusal Kongresi, pp, 645-648.
2. Germen, E. , Bilgen, S., 1992, Transcription and Recognition of One Part Melody With Computer, *Proc. ISCIS'8*, pp. 643-646.
3. Germen, E., Bilgen, S. 1992 A Microprocessor-Based System for Transcription and Processing of Single Part Music, Workshop Notes of 10th ECAI Conference. Vienna, Austria.

4. Germen, E., Bilgen, S., 1997, A Statistical Approach to Determine the Neighborhood Function and Learning Rate in Self-Organizing Maps, Proc. ICONIP'97, Dunedin New Zealand, Springer, pp.334-337.
5. Germen, E., Bilgen, S., 1998, STATSOM, Statistical Self-Organizing Map, Proc. NC'98, Vien, Austria, pp. 117-122.
6. Karul, C., Soyupak, S., Germen, E., 1997, A New Approach to Mathematical Water Quality Modeling in Reservoirs: Neural Networks, Proc. 3rd International Conference on Reservoir Limnology and Water Quality, Ceske Budejovice, Czeck Rep., August 11-15.
7. Karul, C. , Soyupak, S., Germen, E., 1998, "A Study on the Use of Neural Network Based Ecosystem Models as a Management Tool in Lakes", Proc. International Conference on Trophic Interactions in Shallow Freshwater and Brackish Lakes, 3-8 August,1998, Blossin Near Berlin.
8. Karul, C. , Soyupak, S., Germen, E., Beklidoğlu, M., Tuncer, A., 1998, Case Studies on the Use of Neural Networks Ineutrophication Modeling" Proc. First International Symposium Issues in Environmental Pollution (IEP'98), Denver, Colorado, August 23-26.
9. Karul, C., Soyupak, S., , Germen, E., 1998, A New Approach to Mathematical Water Quality Modeling in Reservoirs; Neural Networks, International Review of Hydrobiology, Journal, Vol,83, pp.689-696.