SOFT SELF-ORGANIZING MAP

BY

JOHN PUI-FAI SUM

SUPERVISED BY

DR. LAI-WAN CHAN

A THESIS

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF MASTER OF PHILOSOPHY

DIVISION OF COMPUTER SCIENCE

THE CHINESE UNIVERSITY OF HONG HONG

1 JUNE 1995

Acknowledgement

I would like to express my gratitude to Dr. Lai-wan Chan for her supervision and advice throughout these years. Furthermore, I also want to thank Dr. Lei Xu and Dr. Irwin King for their valuable comments on the earlier version of this thesis.

I am indebted to Dr. Peter Tam and Dr. Chi-Sing Leung for their valuable suggestions. In particular, the experiments carried out in Chapter 5 is suggested by Dr. Chi-sing Leung.

Finally, I would like to thank Shirley for her spiritual support during these years and her proof reading of the early manuscript of the thesis.

Abstract

This thesis presents two algorithms for soft Self-Organizing Map (SSOM) called $SSOM_1$ and $SSOM_2$. Their constructions are motivated by the ordering property manifested by Self-Organizing Map (SOM) and the nature of soft competition manifested by Maximum Likelihood Competitive Learning (MLCL) and Fuzzy Competitive Learning (FCL). By studying the relationship between the algorithm of competitive learning and SOM, a mechanism for ordered map formation is proposed. Based on the introduction of such mechanism, the idea of neighborhood interaction, the algorithms of MLCL and FCL are modified to form SOM-like algorithms: $SSOM_1$ and $SSOM_2$.

These algorithms ($SSOM_1$ and $SSOM_2$) in addition with SOM are then applied to solve two problems including (i) uncovering the neighborhood amongst different vowels and (ii)minimizing the channel noise effect for vowel data transmission. It is found that $SSOM_1$ is not feasible to construct the cluster relationship while SOM and $SSOM_2$ can construct such relationship based on a simple heuristic labeling scheme. In the problem of vowel data transmission, it is experimented that the performance of $SSOM_1$ and $SSOM_2$, in the sense of quantization error and the changes of quantization error with respect to channel noise variance, are comparable to that of using SOM.

Moreover, this thesis provides some results on the convergence analysis on the three algorithms discussed. In particular, the proof on the convergence of the one dimensional SOM will be proven. It is shown that the convergence can be locally almost sure even if the neighborhood size is not finite. Furthermore, if the input data is uniformly distributed, an energy function can be defined. Equivalent the energy function to Lyapunov function, the convergence of SOM is proven to be globally almost sure.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

The principal objective of this thesis is to construct two algorithms called Soft Self-Organizing Map I ($SSOM_1$) and Soft Self-Organizing Map II ($SSOM_2$). They can generate soft-competition-based neighborhood preserved map. In sequel, the algorithms are applied to uncover the relationship amongst different vowels and minimize the channel noise effect in vowel data transmission. In this chapter, we will mainly present the behind motivation of this thesis and the contributions of this thesis.

Four sections are included in this chapter. In the first section, the motivation will be presented. Then, the methodology of the construction of such SSOM algorithm will be discussed in section two with remarks on certain similar approaches. To clarify the contribution of the thesis, section three concisely lists out all the new results obtained including new algorithms, possible applications and theoretical supplement. Finally, we outline the thesis in section four.

1.1 Motivation

Self-Organizing Map (SOM) is an unsupervised learning algorithm resembling the structure and learning of sensory maps in the mammalian brain due to its manifestation of neighborhood preserved map and its vector quantization ability. It was proposed early by Willshaw and Malsburg in a somewhat different structure [48, 49]. Lately, Kohonen proposed the current algorithm and applied it to engineering problems such as speech recognition [20] and the construction of semantic map [37]. Recently, SOM has even been applied to many other areas including motor control [38], traveling salesman problem [1], channel noise reduction [25] and etc. Nowlan and Jou proposed two soft competition[1] algorithms: Maximum Likelihood Competitive Learning (MLCL) [33] and Fuzzy Competitive Learning (FCL) [16]. The former one represents an algorithm of neural network while the latter one represents an algorithm of fuzzy clustering[2]. Although MLCL and FCL are from two rather different areas, their mechanism share one common property: the boundaries of the resultant clusters are "soft" in nature.

The advantage of soft boundary can be illustrated by some of recent papers. In [33], Nowlan applied MLCL to classify vowel data and demonstrated that the correct classification ratio attended by MLCL was higher than that from SOM. In [50] Yair et.al. applied this algorithm[3] to generate codebook for Gauss-Markov data. Similarly, in [7], Chung and Lee applied FCL to vowel data classification and found that the performance of FCL in vowel data classification was better than competitive learning and Learning Vector Quantization. Therefore, it is speculated that the good results indicated in these papers are essentially due to the soft boundary nature of the clustering algorithms utilized. However, as we mentioned previously, there are some problems which cannot be solved by soft competition, for instance, construction of semantic map for syntactic analysis [37] and the minimization of channel noise effect in image transmission [25] as they needs topological preserved map.

As both soft competition algorithms and SOM have their own contributions, it is interesting to ask whether we can merge them together or not. Thus the resultant

[1] Here and after, each time we use the term soft competition, we actually refer to MLCL and FCL.

[2] It is remarked that the reason why FCL is being considered in this thesis is not due to its soft boundary nature.

[3] In [50], they called MLCL as soft competition.

algorithm can possess all the good features from both soft competition and SOM. In the rest of the thesis, the merging algorithm will be presented.

1.2 Idea of SSOM

To accomplished such merging algorithm, there are two possible approaches. The first approach is to extend the algorithms of MLCL and FCL to SOM-like algorithms by introducing the process of neighborhood interaction. The method is as follows: We first examine the mechanism of SOM and competitive learning and figure out how competitive learning can be extended to SOM. Based on this cue, we extend the algorithms of MLCL and FCL in such a way as that of competitive learning do[4].

The second approach is to extend the mechanism of SOM by modifying the competition mechanism to soft competition. The method is as follows: We first figure out the competition mechanism in the algorithm of SOM. Then we replace this competition algorithm by soft competition algorithm and keep the neighborhood interaction mechanism unchange.

Though these two approaches seem distinct, their resultant methodologies are the same: to merge the algorithm of SOM with soft competition, strictly speaking, to merge the algorithm of SOM with MLCL and FCL.

1.3 Other Approaches

In the recent years, many researchers have proposed different approaches to merge topological order and soft competition. Durbin et.al. [10] proposed a model called Elastic network to solve TSP. Once the training is finished, elastic net manifests

[4] In fact, there is an alternative method to realize such idea. It is came from Elastic net. The method can be treated as a regularization problem since the objective function consists of a term corresponding to the objective function of MLCL and another term corresponding to the distance between neighborhood neurons. Along the same line, we can extend any other clustering algorithm in the same way as long as objective function for the clustering algorithm exists.

topological ordered property. Pal et.al. [34] proposed a classifier which used Self-Organizing Map (SOM) as part of the network structure. The proposed classifier was similar to counter-propagation network proposed by Hecht-Nielsen [14]. Mitra et.al. [30] proposed a self-organizing fuzzy classifier (SOFC) which is similar to the algorithm of SOM. SOFC is responsible to classify the batch of data into clusters and using the topological ordering property to reveal the neighborhood structure amongst clusters. Pham et.al. [35] and Vuorimaa [46] defined a SOM-based training procedure to obtain the cluster centers. Then, based on the result obtained after the first phase, they construct the fuzzy sets.

Pal et.al. and Mitra et.al. increase the number of fan-in by a factor of three in order to incorporate the concept of linguistic variable. Although the structure can be revealed using their approaches, the structure cannot reflect to the original input data space. The algorithms of Pham et.al. and Vuorimaa are too heuristic. In such case, they are difficult to be analyzed and evaluated. Amongst all, only elastic net does not suffer from their problems. Besides, Yullie has provided a vigorous analysis on the statistic-mechanical property to elastic net. However, the usefulness of this method is in so far restricted to one-dimensional map.

1.4 Contribution of the Thesis

According to the previous discussion, hopefully, SSOM should be able to reveal the neighborhood relationship amongst clusters (due to soft competition). Furthermore, SSOM can be applied to data transmission with the aid of ordering property. Fortunately, it does. In addition with other supplementary results, below lists the contribution of the thesis:

1. **Development of SSOM**

   (a) $SSOM_1$ and $SSOM_2$ are constructed.

   (b) Ordering property of both models are demonstrated by simulation.

   (c) Convergence property of both models are proven using the technique of stochastic approximation and perturbation method.

2. **Theoretical supplement to SOM**

   (a) Following the approach of Bouton and Pages [4], the convergence of 1D SOM is proven to be almost sure even if the neighborhood set size is not finite.

   (b) Applying Krasovskii method [18], an energy function is constructed for the 1D SOM when the input data distribution is uniform. Hence, the convergence of 1D SOM under such conditions is globally almost sure.

3. **Theoretical supplement to FCL and CL**

   (a) Applying the technique of stochastic approximation, the sufficient condition ensuring the convergence of FCL is globally almost sure is derived.

4. **Application of SSOM**

   (a) SSOM is applied to construct the neighborhood relationships amongst clusters.

   (b) SSOM is applied to data transmission.

1.5 Outline of Thesis

This thesis is organized into seven chapters and four Appendices. This chapter presents the motivation and the basic ideas of SSOM. In chapter two, the mechanism and properties of SOM will be presented. The mechanisms of competitive learning, soft competitive learning and algorithms of soft SOM will be elucidated in chapter three. The ordering property of SSOM is demonstrated by several simulation examples. Then, SSOM is applied to solve two problems. In chapter four, SSOM is applied to reveal the relationship amongst clusters. In chapter five, SSOM is combined with quadrature amplitude modulation (QAM) scheme to transmit vowel data. Preliminary theoretical study on the convergence of SSOM, SOM, CL and FCL will be presented in chapter six. Then conclusion follows in chapter seven. Appendix A provides the proof of Corollary 1 and 2, which are stated in Chapter 2. Appendix B discusses different sense of neighborhood which can help to understand the definition of SSOM. Appendix C includes some figures supplemented to the results discussed in chapter four. Appendix D gives a brief review on QAM.

Chapter 2

Self-Organizing Map

This chapter reviews the model of SOM. The convergence and ordering properties will be discussed based on several illustrative examples. Mathematical discussion on the convergence property will be presented in Chapter 7 and Appendix A[1]. After an introduction given in the first section. The mechanism of SOM will briefly be described in section two. Then, in section three, a some examples are provided to illustrate the behavior and the properties of SOM. The purpose of these three sections is to visualize the mechanism of SOM. Section four and section five summarize current results on the convergence, ordering and cost function of SOM. Finally, a conclusion will be presented in section six.

2.1 Introduction

As evidence from neural science [17], human brain exhibits topological ordered map in a number of place in the cerebral cortex such as retinotopic map in the visual cortex, somatotopic neural map in the somatosensory cortex, tonotopic neural map and motor map. According to the property of topological ordering, researchers have proposed many different models to mimic such neural maps. Willshaw and Von der

[1] In Chapter 7, the local convergence proof on higher dimensional map is shown. In Appendix A, the convergence of one dimensional map is elucidated.

Malsburg first proposed a model of retinotopic map [48], [49] and demonstrated the ordering property through simulations. Since then, many other models have also been proposed to accomplish such MAP [19] [27] and [45]. Self-Organizing Map (SOM) is one of the simpliest model and widely applied[2]. However, there are limitations in the application of SOM: there is no *complete* analysis on the convergence and ordering properties of SOM. Besides, in so far, there is no energy function has been proven to be its cost function. Therefore, it becomes not so possible to evaluate the performance of SOM analytically based on the criteria mean square error. The following sections will be devoted to the description of the model of SOM and its properties. Certainly, not all of them, codevector density for instance, will be discussed due to the scope of this thesis. Only some of the crucial property related to the development of Soft Self-Organizing Map will be introduced.

2.2 Algorithm of SOM

Generally, SOM is a two layered neural network. Each of the nodes in the input layer receives input signal and transmits to the second layer through weights (synapses). Suppose there are s input nodes, we denote the input data by $x = (x_i, x_2, \ldots, x_s)^T \in R^s$ and the values of weights connecting input to ith output node by $v_i = (v_{i1}, v_{i2}, \ldots, v_{is})^T \in R^s$. Each of the node in the second layer collects all the signal fed from the first layer and output a signal, say y_i. Consider there are c neurons at the output layer. They are indexed by $1, 2, .., c$. Once the x is fed to the network, each of the output nodes will give out a signal either one or zero depended on the Euclidean distance between x and v_i:

$$y_i = \begin{cases} 1 & \text{if } \|x - v_i\| \leq \|x - v_j\| \text{ for all } i \neq j. \\ 0 & \text{otherwise.} \end{cases}$$

[2] See [38] and the reference listed.

Unless the data is equal distance from two weights vector, there is one and only one output node will give out one. So, this mechanism is also called *winner-take-all.* The output one node will be called the winner.

While the SOM is in learning, the weight vectors will be modified according to these output values. The learning mechanism can be summarized in the following four steps [21]:

Step 1 *Select randomly one sample, x, from the stationary sample space, $f(x)$.*

Step 2 *Evaluate the winning output neuron, Ith output node, by evaluating* $\|x - v_I\| = \min_i \|x - v_i\|$

Step 3 *Modify the weight vectors by*

$$v_i(t+1) = \begin{cases} v_i(t) + \alpha(t)A_i(t)[x(t) - v_i(t)] & \text{if } i \in N_I \\ v_i(t) & \text{otherwise} \end{cases} \tag{2.1}$$

Step 4 *Goto Step 1.*

Here N_I in (2.1) is a set which defines the weight vectors to be updated. $A_i(t)$ is a scalar function of time and $|i - I|$. It determines the relative update step size for the ith weight vector. $\alpha(t)$ is the updating step size. Usually, A_i is a decreasing function of $|i - I|$ and $\alpha(t)$ decreases to zero as $t \to \infty$. Adopted from [51], N_I is called the *neighborhood interacting set* (NIS). and $A_i(t)$ is called the *neighborhood interacting function.*

Example 1 *Consider that there are six output nodes. Their corresponding weight vectors are denoted by $v_1, v_2, v_3, v_4, v_5, v_6$. We can define the NIS and NIF as the following: $N_1 = \{1,2\}, N_2 = \{1,2,3\}, N_3 = \{2,3,4\}, N_4 = \{3,4,5\}, N_5 = \{4,5,6\}, N_6 = \{5,6\}$ and $A_i = 1$.*

Example 2 *Consider the output nodes are arranged as a two dimensional mesh. The winner node is denoted by IJ. The NIS can be defined as that*

$N_{IJ} =$

$I-1, J-1$	$I-1, J$	$I-1, J+1$
$I, J-1$	I, J	$I, J+1$
$I+1, J-1$	$I+1, J$	$I+1, J+1$

and the NIF can be defined as that

$$A_i(t) = \begin{cases} \beta_0(t) & \text{if } i = I \\ \beta_1(t) & \text{if } i \in N_I - \{I\}. \end{cases} \tag{2.2}$$

If the map is three by three, the NIF can be written by

$\beta_0(t)$	$\beta_1(t)$	0
$\beta_1(t)$	$\beta_1(t)$	0
0	0	0

$\beta_1(t)$	$\beta_0(t)$	$\beta_1(t)$
$\beta_1(t)$	$\beta_1(t)$	$\beta_1(t)$
0	0	0

0	$\beta_1(t)$	$\beta_0(t)$
0	$\beta_1(t)$	$\beta_1(t)$
0	0	0

$\beta_1(t)$	$\beta_1(t)$	0
$\beta_0(t)$	$\beta_1(t)$	0
$\beta_1(t)$	$\beta_1(t)$	0

$\beta_1(t)$	$\beta_1(t)$	$\beta_1(t)$
$\beta_1(t)$	$\beta_0(t)$	$\beta_1(t)$
$\beta_1(t)$	$\beta_1(t)$	$\beta_1(t)$

0	$\beta_1(t)$	$\beta_1(t)$
0	$\beta_1(t)$	$\beta_0(t)$
0	$\beta_1(t)$	$\beta_1(t)$

0	0	0
$\beta_1(t)$	$\beta_1(t)$	0
$\beta_0(t)$	$\beta_1(t)$	0

0	0	0
$\beta_1(t)$	$\beta_1(t)$	$\beta_1(t)$
$\beta_1(t)$	$\beta_0(t)$	$\beta_1(t)$

0	0	0
0	$\beta_1(t)$	$\beta_1(t)$
0	$\beta_1(t)$	$\beta_0(t)$

Ignoring the boundary nodes, the inner node is surrounded by eight neighboring nodes. Therefore, this type of NIS is called eight-neighbor type.

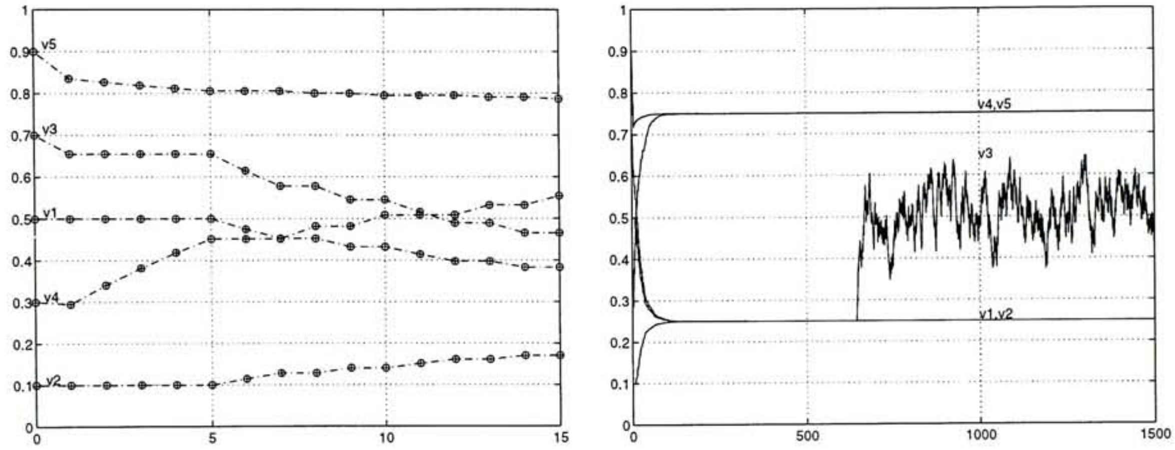
2.3 Illustrative Example

For clarity, here gives a simple example to illustrate the mechanism of SOM learning and to describe its properties numerically and graphically. The SOM is constituted by one input node and five output nodes. The weights are denoted by v_1, v_2, v_3, v_4 and v_5. The input sample set consists of two elements $\{0.25, 0.75\}$. The probability mass function is given by $f(0.25) = f(0.75) = 0.5$. Initially, the weight values are set as following: $v_1(0) = 0.5$, $v_2(0) = 0.1$, $v_3(0) = 0.7$, $v_4(0) = 0.3$ and $v_5(0) = 0.9$. The step size $\alpha(t) = 0.1$. The NIF, $A_i = 1$ and the NIS are defined as $\{1, 2\}$, $\{1, 2, 3\}$, $\{2, 3, 4\}$, $\{3, 4, 5\}$ and $\{4, 5\}$ respectively. The values of v_is in the first fifteen iterations are tabulated in Table 2.1. The first column indicates the number of iterations. The second column indicates the winning node at the corresponding step. The input to the SOM is shown in the third column. The values of v_is are indicated from the 4th column to 8th column. In the first iteration, the element $x = 0.25$ is selected. At that time, $v_4(0)$ is the closest weight vector. Hence, node four is the winner node. In sequel, v_3, v_4 and v_5 are changing according to Step 3. Graphically, these update manifests two phenomena simultaneously: (i)the input x attracts the winner and its neighborhood to move towards itself and (ii)the winner and its neighborhood are getting closer. Figure(2.1a) plots the values of v_is in the first fifteen iterations. As v_4 and v_5 are neighbor, they tend to getting closer. Similar situation happens to v_1, v_2 and v_3.

	Winner	Input	v_1	v_2	v_3	v_4	v_5
0	-	-	0.5000	0.1000	0.7000	0.3000	0.9000
1	4	0.2500	0.5000	0.1000	0.6550	0.2950	0.8350
2	5	0.7500	0.5000	0.1000	0.6550	0.3405	0.8265
3	5	0.7500	0.5000	0.1000	0.6550	0.3814	0.8188
4	5	0.7500	0.5000	0.1000	0.6550	0.4183	0.8120
5	5	0.7500	0.5000	0.1000	0.6550	0.4515	0.8058
6	2	0.2500	0.4750	0.1150	0.6145	0.4515	0.8058
7	2	0.2500	0.4525	0.1285	0.5780	0.4515	0.8058
8	5	0.7500	0.4525	0.1285	0.5780	0.4813	0.8002
9	2	0.2500	0.4322	0.1406	0.5452	0.4813	0.8002
10	5	0.7500	0.4322	0.1406	0.5452	0.5082	0.7952
11	2	0.2500	0.4140	0.1516	0.5157	0.5082	0.7952
12	2	0.2500	0.3976	0.1614	0.4891	0.5082	0.7952
13	5	0.7500	0.3976	0.1614	0.4891	0.5324	0.7907
14	2	0.2500	0.3829	0.1703	0.4652	0.5324	0.7907
15	5	0.7500	0.3829	0.1703	0.4652	0.5541	0.7866

Table 2.1: The changing of the v_is value in the first fifteen iterations.

Repeating the steps for several hundreds of iterations, v_4 and v_5 merge together at 0.75. The v_1, v_2 and v_3 merge together at the value 0.25. As a result, topological order is formed since $v_1 = v_2 = v_3 < v_4 = v_5$. However, when the learning steps are



(a) (b)

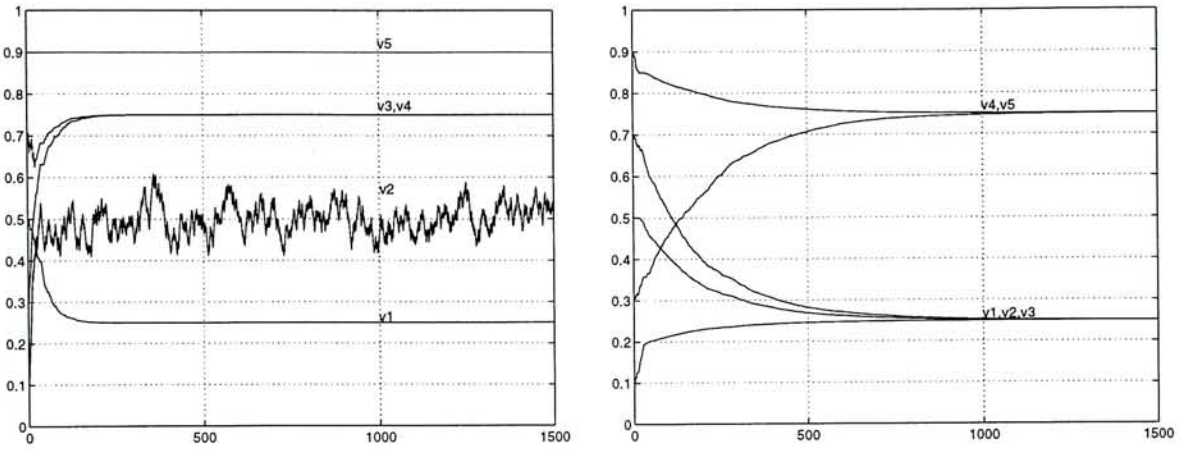
Figure 2.1: The changing of the weight values for the illustrative example. The vertical axis indicates the value of the weight while the horizontal axis indicates the number of training step. In (a) and (b), $\alpha(t)$ is set to a constant value 0.1. The weight values at the first fifteen iterations are shown in (a). (b) shows the changing of weight values within 1500 iterations.

repeated until the 646th iteration, it is found v_3 gets out from the value 0.25 and increases to about 0.5, Figure(2.1b).

If the value of α is changed to a smaller value, the resultant v_is are difference. Figure(2.2) shows the cases when $\alpha = 0.05$ and $\alpha = 0.01$ respectively. It is found that ordering cease in the former case while the ordering preservation is manifested in the latter case.

In all three cases, ordering property is manifested. However, it is not always the case. For instance, Figure(2.3) shows one case where ordering property is ceased. Here, NIS and NIF are defined in the same way as above. The data set is the same. But the initial condition is different. Figure(2.3a) shows the case when $\alpha(t) = 0.01$. Figure(2.3b) shows the case when $\alpha(t) = 0.1$.

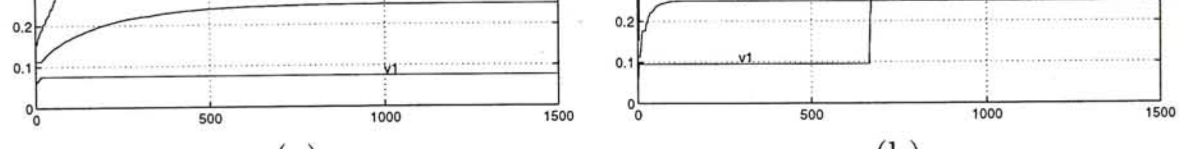
According to these examples, four observations can be noted: (1)If the network parameters and initial conditions are set appropriately, topological map can be formed. (2)SOM can converge to a stationary state. (3)There are more than one stationary state that SOM can be reach. (4)Larger step size can enhance the formation of



(a)

(b)

Figure 2.2: The changing of the weight values for the illustrative example. The vertical axis indicates the value of the weight while the horizontal axis indicates the number of training step. In (a) $\alpha(t)$ is set to a constant value 0.05. In (b), $\alpha(t) = 0.01$.



(a)

(b)

Figure 2.3: The changing of the weight values for the illustrative example. The vertical axis indicates the value of the weight while the horizontal axis indicates the number of training step. In (a) $\alpha(t)$ is set to a constant value 0.01. In (b), $\alpha(t) = 0.1$. Noting that the initial conditions of v_is are different.

topological map.

2.4 Property of SOM

2.4.1 Convergence property

Although the mechanism of SOM is very simple, there is no proof on the convergence property except on certain simple cases [4] [8] [11] [21] [22] [28] [36] [42] [51].

Ritter and Schulten treated it as a Markovian process. They derived Fokker-Planck approximated equation for SOM and arrived with an approximated equation for the mean average update [36]. Hence they showed that SOM can converge to stationary state.

Theorem 1 (Ritter and Schulten [36]) *Suppose that $\beta_0(t) = \beta_1(t) = 1$ for all t and V^* be an asymptotic equilibrium state, then the necessary and sufficient conditions for the local convergence of SOM are: (i) $\lim_{t\to\infty} \int_0^t \alpha(s)ds = \infty$ and (ii) $\lim_{t\to\infty} \alpha(t) = 0$.*

□□□

Others applied the so-called Gladyshev Theorem to show that the convergence of $1D$ or $2D$ SOM is locally almost sure if the input distribution is uniform [8] [22] [28] [51].

Recently, Bouton et.al. [4] proved that the convergence of SOM, under nonuniform distribution, is almost sure.

Theorem 2 (Bouton and Pages [4] [8]) *Consider 1-D SOM which $\beta_0(t) = \beta_1(t) = 1$ for all t, the convergence is locally almost sure if either*

- *the distribution of the data is uniform, i.e. $f(x) = 1$*
- *the distribution of the data is logconcave or*

- *the distribution of the data is loglinear.*

And $\alpha(t)$ should satisfies that $\sum_{t=0}^{\infty}\alpha(t)=\infty$ and $\sum_{t=0}^{\infty}\alpha^2(t)<\infty$.

□□□

Following the same idea as Bouton et.al. we can extend the convergence proof to the case that the NIF is decreasing outward and the size of NIS is any large. As the proof is lengthy, the corollary is stated below while the proof is presented in Appendix.

Corollary 1 *Consider 1-D SOM which $\beta_0 \geq \ldots \geq \beta_l \geq 0$ for all t, where $l > 1$ the convergence is almost sure if $\alpha(t)$ satisfies the conditions of Theorem 2 and the input distribution is either uniform, logconcave or loglinear. Furthermore, if the input distribution is uniform, the convergence is globally almost sure.*

□

2.4.2 Topological Order

One promising property of SOM is that SOM can organize to an ordered map which reveals the intrinsic relationship amongst the training clusters. It has been demonstrated by hte illustrative example. However, the analysis on this property, in so far, is restricted to one-dimensional map. One critical reason is due to the lacking of formal defintion of "order" in higher dimensional map.

2.4.3 Objective Function of SOM

Besides the lacking of formal definition of "order", SOM is suffered from the lack of objective function. So that, on one hand, the performance of SOM is hard to compared with other algorithms analytically. On the other hand, it reveals one reason why the convergence proof of SOM is not yet completed. Anyway, there are two special cases under which we can define the objective function for SOM.

Firstly, suppose the $\beta_0(t) = 1$ and $\beta_1(t) \to 0$ for all $k \neq 0$, the objective function of self organizing map can be defined by

$$J = \sum_{k=1}^{c} \sum_{x \in \Omega_k} \|x - v_k\|^2 f(x)dx. \tag{2.3}$$

It is just the same as the objective function of LBG [26] or competitive learning [15], if the NIF is decreasing to zero. That is to say, one special case of SOM can be treated as LSE-based algorithm. However, when the NIF is not decreasing, it is difficult to construct such an objective function. Tolat [42] tried to define neuronal energy function, which is tried to generalize the proof to high dimension. However, the energy functions are not the true energy functions. So, it cannot reflect the true mechanism of SOM. Recently, Erwin et.al. [11] claimed that the global objective function for SOM does not exist.

Secondly, consider an one dimensional SOM. If (a) the input data is scalar and the distribution is uniform and (b) $\beta_0 \geq \ldots \geq \beta_l \geq 0$ where $l > 1$, then

$$J = \sum_{i=1}^{c} h_i(v_1, \ldots, v_c)^2 \tag{2.4}$$

is the object function, where

$$\begin{aligned} h_1(v_1, \ldots, v_c) &= \int_{\Omega_1} (x - v_1) f(x)dx + \int_{\Omega_2} (x - v_1) f(x)dx + \ldots \\ &+ \int_{\Omega_{l+1}} (x - v_1) f(x)dx \end{aligned} \tag{2.5}$$

$$\begin{aligned} h_2(v_1, \ldots, v_c) &= \int_{\Omega_1} (x - v_2) f(x)dx + \int_{\Omega_2} (x - v_2) f(x)dx + \ldots \\ &+ \int_{\Omega_{l+2}} (x - v_2) f(x)dx \end{aligned} \tag{2.6}$$

$$\ldots$$

$$h_i(v_1, \ldots, v_c) = \int_{\Omega_{i-l}} (x - v_i) f(x)dx + \ldots + \int_{\Omega_{i-1}} (x - v_i) f(x)dx \tag{2.7}$$

$$
\begin{aligned}
&+ \int_{\Omega_i}(x-v_i)f(x)dx + \int_{\Omega_{i+1}}(x-v_i)f(x)dx \\
&+ \ldots + \int_{\Omega_{i+l}}(x-v_i)f(x)dx
\end{aligned}
$$

$$\ldots$$

$$h_c(v_1,\ldots,v_c) = \int_{\Omega_{c-l}}(x-v_c)f(x)dx + \ldots + \int_{\Omega_c}(x-v_c)f(x)dx, \qquad (2.8)$$

where $\Omega_i = \{x|\|x-v_i\| \leq \min_{k\neq i}\|x-v_k\|\}$. The derivation of such objective function is in the proof of the case 1 of corollary 1. The idea of proof is based on Krasovskii method [18]. The derivation of such objective function is given in Appendix A. As energy function can be defined, the convergence of SOM defined in this case can be proven to be almost sure.

2.5 Conclusion

In this chapter, we have briefly reviewed the model of SOM. Its network structure and its learning mechanism. To clarify the mechanism, a simple example is given. Moreover, these examples illustrate the ordering and convergence behavior of SOM. Some theoretical results on these issues are discussed as well. In summary, SOM is a simple neural network model and it resembles the map property of cerebral cortex in our brain. However, it suffers from the lack of completed analytical proof on each of its properties including convergence and ordering.

Even though, we have added on some new results on both the convergence proof and objective function for SOM, there are lot of work have to be done to accomplish a complete theoretical analysis and to explore the topological ordering property to other models.

Chapter 3

Algorithms for Soft Self-Organizing Map

This chapter presents two algorithms for Soft Self Organizing Map and demonstrates their ordering properties using three simulation examples. In the first section, the mechanisms of competitive learning, Maximum Likelihood Competitive Learning and Fuzzy Competitive Learning will be described. Then, the relationship between the algorithms of SOM and simple competitive learning will be discussed in section three. It aims at of indicating a cue explaining why SOM can generate neighborhood preserved map but competitive learning cannot. Using this cue, MLCL and FCL are extended to two soft algorithms of SOM: $SSOM_1$ and $SSOM_2$. These algorithms will be presented in section four. In section five, simulation results are provided to illustrate the ordering properties of both algorithms. Hence a conclusion will be presented in section six.

3.1 Competitive Learning and Soft Competitive Learning

The proposing of competitive learning can be traced back to the time when Frank Rosenblatt invented Perceptron. In [39], Rosenblatt proposed a class of Perceptron model to explain the information processing and storage in our brain. One model called γ-perceptron is exactly the modern time competitive learning algorithm [15]. Briefly, the mechanism of competitive learning can be described using Figure(3.1).

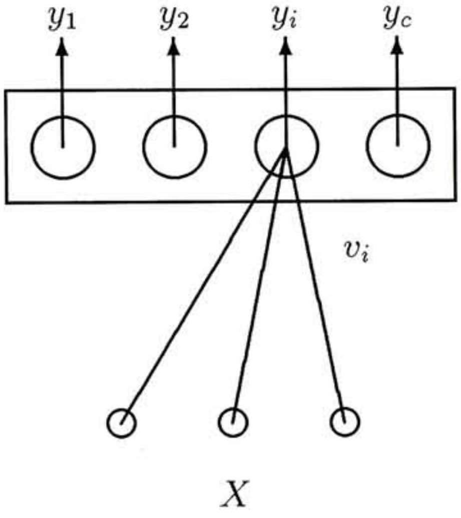


Figure 3.1: Structure of competitive learning.

Suppose that there are c neurons in the output layer. The output of the neurons are denoted by $y_1, \ldots, y_c$. Each neurons receives signal from the input layer. The response of the neuron is defined by

$$y_i(t) = \begin{cases} 1 & \text{if } \|x(t) - v_i(t)\| \leq \min_{k \neq i} \|x(t) - v_k(t)\| \\ 0 & \text{otherwise.} \end{cases} \tag{3.1}$$

The learning of the weights, v_i is defined by

$$v_i(t+1) = v_i(t) + \alpha(t) y_i(t)(x(t) - v_i(t)), \tag{3.2}$$

where $\alpha(t)$ satisfies the conditions: $\int_0^\infty \alpha(t)dt = \infty$ and $\int_0^\infty \alpha^2(t)dt < \infty$.

Suppose that the pdf of x is denoted by $f(x)$, the objective function of competitive learning is given by

$$J = \sum_{i=1}^{c} \int_{\Omega_i} \|x - v_i\|^2 f(x)dx, \tag{3.3}$$

where $\Omega_i = \{x : \|x - v_i\| < \min_{k \neq i} \|x - v_k\|\}$. Hence, competitive learning is just the on-line complement of LBG algorithm [26].

In competitive learning, it can conceive that the response of neuron is a characteristic function indicating the degree of winning of that neuron in the competition. If the neuron is winner, its degree will be one. If it is loser, the degree will be zero. In another words, each neuron can only either be winner or loser. The decision whether the neuron is winner or not is hard decision and the competition is called hard decision.

Instead of defining the competition in a hard way, Nowlan and Jou recently proposed algorithms which are incorporated with the concept of soft competition:

- $$v_i(t+1) = v_i(t) + \alpha(t)y_i(x(t) - v_i(t)), \tag{3.4}$$

  where

  $$y_i(x, v_i, v_2, \ldots, v_c) = \frac{\exp(-\|x - v_i\|^2/t)}{\sum_{k=1}^{c} \exp(-\|x - v_k\|^2/t)} \tag{3.5}$$

  and $t > 0$;

- $$v_i(t+1) = v_i(t) + \alpha(t)y_i^m(x - v_i(t)) \tag{3.6}$$

  where

  $$y_i(x, v_1, \ldots, v_c) = \left[\sum_{k=1}^{c} \left(\frac{\|x - v_i\|^2}{\|x - v_k\|^2} \right)^{1/(m-1)} \right]^{-1} \tag{3.7}$$

  and $m > 1$.

The former algorithm is Nowlan's maximum likelihood competitive learning while the latter one is Jou's fuzzy competitive learning. Their algorithm share one common feature. There is no absolute winner or loser. Each neuron is winner. The value of the characteristic function is not binary but any value between zero and one, i.e. $y_i \in [0,1]$. Essentially, their algorithms can be written in the following general form:

$$y_i = y_i(x, v_i, v_2, \ldots, v_c),$$

$$v_i(t+1) = v_i(t) + \alpha(t)F(y_i(t))(x(t) - v_i(t)), \tag{3.8}$$

where $F(y_i(t))$ is a monotone increasing function defined on $[0,1]$ and $\alpha(t)$ satisfies the conditions, $\int_0^\infty \alpha(t)dt = \infty$ and $\int_0^\infty \alpha^2(t)dt < \infty$. Then, in case of Nowlan's algorithm, $F(y_i) = y_i$. In case of Jou's algorithm, $F(y_i) = y_i^m$.

Intuitively, as their algorithms are different, both algorithms minimize different objective function except at the limiting case when $m = 1^+$ and $t = 0^+$. In case of Nowlan's algorithm, the objective function is defined by

$$J_t = -\int \log\left[\sum_{i=1}^{c} \exp\left(-\|x - v_i\|^2/t\right)\right] f(x)dx. \tag{3.9}$$

In case of Jou's algorithm, the objective function is defined by

$$J_m = \sum_{i=1}^{c} \int \left[\sum_{k=1}^{c} \left(\frac{\|x - v_i\|^2}{\|x - v_k\|^2}\right)^{\frac{1}{m-1}}\right]^{-m} \|x - v_i\|^2 f(x)dx. \tag{3.10}$$

3.2 How does SOM generate ordered map?

In order to modify MLCL and FCL we need to understand how SOM generates ordered map. In sequel, we may find out some cues so that we can modify the algorithms of Nowlan and Jou in such a way. Ignoring the conditions of $\alpha(t)$, the algorithms of SOM and CL are given as follows: Without loss of generality, we consider the SOM a one dimensional map and suppose that $x, v_i \in R^n$, $y_i \in R$. For SOM,

$$v_i(t+1) = \begin{cases} v_i(t) + \alpha(t)[x(t) - v_i(t)] & \text{if } i \in N_I \\ v_i(t) & otherwise \end{cases} \tag{3.11}$$

and for CL,

$$v_i(t+1) = \begin{cases} v_i(t) + \alpha(t)[x(t) - v_i(t)] & \text{if } i = I \\ v_i(t) & otherwise, \end{cases} \tag{3.12}$$

where I is defined by $I = arg\min_i\{\|x(t)-v_i(t)\|\}$, N_I is the neighborhood interacting set. For example, $N_1 = \{1,2\}$, $N_2 = \{1,2,3\}$, $N_3 = \{2,3,4\}$ and so on. If we define $y_i(t)$ as that

$$y_i(t) = \begin{cases} 1 & \text{if } \|x(t) - v_i(t)\| \le \min_{k\neq i} \|x(t) - v_k(t)\| \\ 0 & \text{otherwise,} \end{cases} \tag{3.13}$$

we can re-formulate the mechanism of SOM and CL in the following way:

$$v_i(t+1) = v_i(t) + \alpha(t)[x(t) - v_i(t)]z_i(t), \tag{3.14}$$

where $z_i(t) = \sum_{k\in N_i} y_k(t)$.

Thus if $N_i = \{i\}$, (3.14) reduces to CL. If $N_i = \{i-1, i, i+1\}$, then (3.14) reduces to SOM. Therefore, we can define a three-layered network to mimic SOM, Figure(3.2), with the first two layers constituting the competitive learning network. Between the y_i layer and z_i layer, the associated weights are not fully connected but partially connected. The value of each weight connection is one. Certainly, it does not mean that SOM is a three-layered network.

If g_{ij} denotes the value of the weight connecting y_i and z_j, $g_{ij} = 1$ if $|i-j| \le 1$ and zero otherwise, then the cue that makes SOM generate ordered map can be conceived as the existence of the associated weights, the g_{ij}s, connecting y-layer and z-layer. Using this cue, we can imagine that the ordering map can still be generated if the lower layer is replaced by MLCL or FCL instead. This is the idea that will be elucidated in the rest of the paper.

It is worthy to note that although our discussion concern solely on the one dimensional map and the value of the weights associating y-z layers is one, the principle still holds for the case when the map is a higher dimensional map and the value of the weights are decreasing outward, i.e. $g_{ii} \ge g_{i,i\pm1} \ge g_{i,i\pm2}$ and so on.

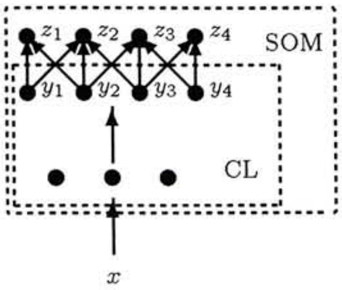


Figure 3.2: Network structure of SOM.

3.3 Algorithms of Soft SOM

According to our preliminary analysis, it is found that the formation of topological map is due to the existence of association between the y-layer and z-layer. That is, from the definition of g_{ij}, we can extend the algorithms of MLCL and FCL to $SSOM_1$ and $SSOM_2$. Again, without loss of generality, we assume that the $SSOM_1$ and $SSOM_2$ are one dimensional map[1]. Assuming that $G(t) = (g_{ij})_{c \times c}$ is a toeplitz matrix satisfying the condition $g_{1i} \geq g_{1j}$ for all $i \leq j$, their learning algorithms can be stated as shown below:

SSOM 1 *For all $i = 1, 2, \ldots, c$,*

$$v_i(t+1) = v_i(t) + \alpha(t)[x(t) - v_i(t)]z_i(t), \tag{3.15}$$

where $z_i(t) = \sum_{k \in N_i} g_{ik} y_k(t)$ and

$$y_i = \frac{\exp(-\|x - v_i\|^2/\tau)}{\sum_{k=1}^{c} \exp(-\|x - v_k\|^2/\tau)} \tag{3.16}$$

for all $\tau > 0$.

SSOM 2 *For all $i = 1, 2, \ldots, c$,*

$$v_i(t+1) = v_i(t) + \alpha(t)[x(t) - v_i(t)]z_i(t), \tag{3.17}$$

[1] Note that the index of v_i, y_i and z_i will be changed to (i, j) in case the map is defined as a two dimensional mesh and and $N_{(i,j)} = \{(i,j), (i \pm 1, j \pm 1), (i, j \pm 1), (i \pm 1, j)\}$

where $z_i(t) = \sum_{k \in N_i} g_{ik} y_k^m(t)$ *and*

$$y_i(x, v_1, \ldots, v_c) = \left[\sum_{k=1}^{c} \left(\frac{\|x - v_i\|^2}{\|x - v_k\|^2} \right)^{1/(m-1)} \right]^{-1} \tag{3.18}$$

for all $m > 1$.

In equilibrium, the cluster centers v_is will be given by $v_i = \frac{\sum_x x z_i(x)}{\sum_x z_i(x)}$. In case of $SSOM_1$, v_i is given by

$$v_i = \frac{\sum_x \sum_{k \in N_i} x g_{ik} y_k(x)}{\sum_x \sum_{k \in N_i} g_{ik} y_k(x)},$$

for all $i = 1, 2, \ldots, c$. In case of $SSOM_2$, v_i is given by

$$v_i = \frac{\sum_x \sum_{k \in N_i} x g_{ik} y_k^m(x)}{\sum_x \sum_{k \in N_i} g_{ik} y_k^m(x)},$$

for all $i = 1, 2, \ldots, c$. Either $SSOM_1$ or $SSOM_2$ can be treated as a generalized model of CL and SOM. It can transform to any one of them by modifying the parameter τ (or m) and reducing the size of neighborhood to singleton, i.e. $N_i = \{i\}$. For clarity, we assume that $g_{ij} = 1$ if $|i - j| \leq 1$. Without loss of generality, we discuss how $SSOM_1$ can change to other algorithms. If $\tau > 0$ and $N_i = \{i\}$, it reduces to MLCL. If $\tau = 0^+$ and $N_i = \{i\}$, it reduces to competitive learning. If $\tau = 0^+$ and $N_i = \{i-1, i, i+1\}$, it reduces to SOM. Similar discussion on the relationship between $SSOM_2$ and other algorithms can follow the same way.

As in the case of SOM, a complete theoretical analysis on the above algorithms, $SSOM_1$ and $SSOM_2$, is very difficult. Therefore the topological ordering property can only be demonstrated by the simulation results as given in the next section.

3.4 Simulation Results

The first example demonstrates the capability of SSOM in handling one dimensional data. The matrix $G(t)$ is defined as: $G(t) = (g_{ij}(t))_{c\times c}$, where

$$g_{ij}(t) = \begin{cases} 1 & \text{if } i = j. \\ \beta(t) & \text{if } |i-j| = 1. \\ 0 & \text{otherwise.} \end{cases} \tag{3.19}$$

while in the second example, two dimensional data is handled. The matrix $G = (g_{ij,rs})cc \times cc$ is defined as follows:

$$g_{ij,rs} = \begin{cases} 1 & \text{if } ij = rs. \\ \beta(t) & \forall rs \in N_{ij}\backslash\{ij\}. \\ 0 & \text{otherwise.} \end{cases} \tag{3.20}$$

3.4.1 One dimensional map under uniform distribution

In this example, the SSOM consists of five weight vectors. The input data is uniformly distributed on $[0,1]$. Initially, the weight vectors are in random position within $[0,1]$. For all $t > 0$, $\beta(t)$ in equation (3.19) is set to be a constant. $\alpha(t)$ is set to be 0.01. As the results of $SSOM_1$ and $SSOM_2$ are similar except that the training time is different, only the results obtained by algorithm $SSOM_1$ are displayed in Figure(3.3). From the figures, we can make the following observations: (i)As $\beta(t)$ decreases, the spread of $|v_1(\infty) - v_5(\infty)|$ will increase. (ii)When $\beta(t) \geq 0.5$, it is possible to obtain ordered map within two thousand times of iteration even the initial map is not in order. (iii)As $\beta(t)$ decreases, the time to reach ordering becomes longer. (iv)As $\beta(t)$ decreases, the fluctuation of v_is also decreases. (v)When $\beta(t) < 0.5$, no ordering map can be obtained within two thousand times of iteration.

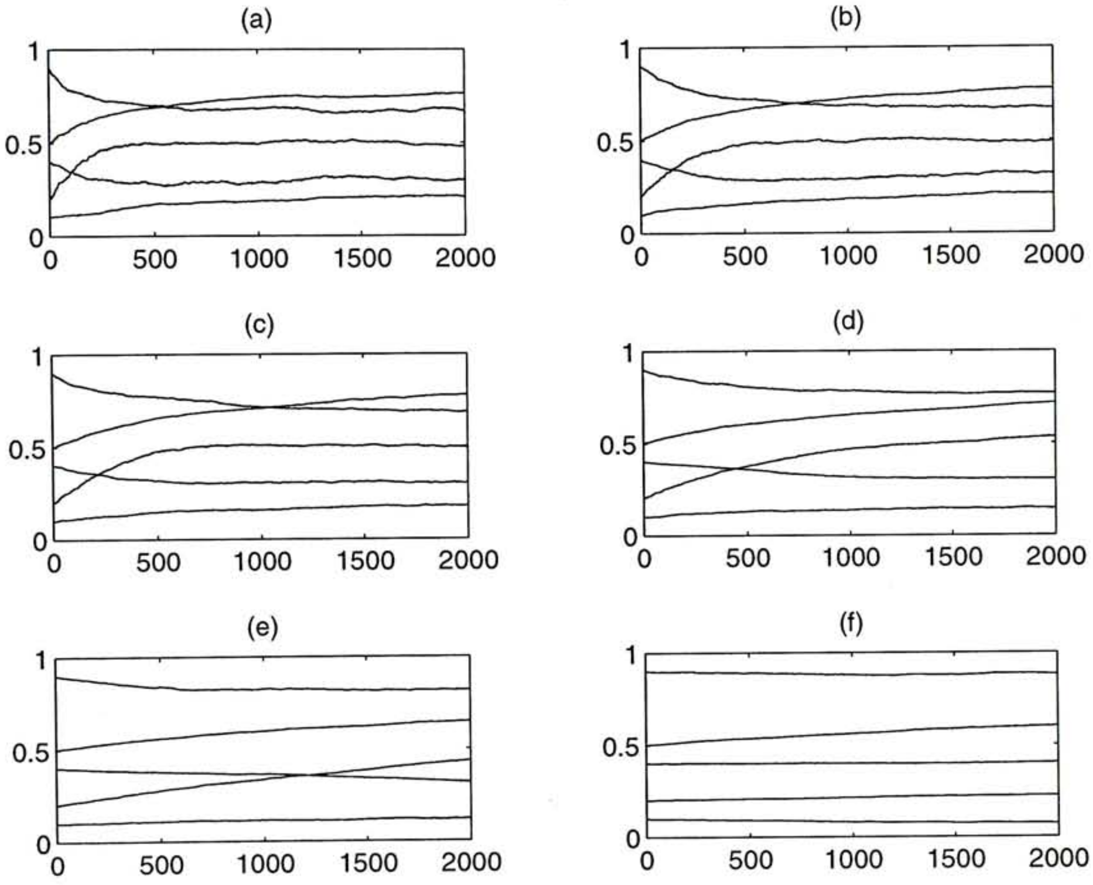


Figure 3.3: The plots of change of v_i of the $SSOM_1$ under uniform distribution. $\beta(t)$ is set to be a constant during each run. The horizontal axis indicates the number of iteration while the vertical axis indicates the value of each of v_i. (a) $\beta(t) = 1$, (b) $\beta(t) = 0.75$, (c) $\beta(t) = 0.5$, (d) $\beta(t) = 0.25$, (e) $\beta(t) = 0.1$ and (f) $\beta(t) = 0$. The value of v_is are set to be: $v_1(0) = 0.5$, $v_2(0) = 0.9$, $v_3(0) = 0.2$, $v_4(0) = 0.4$ and $v_5(0) = 0.1$.

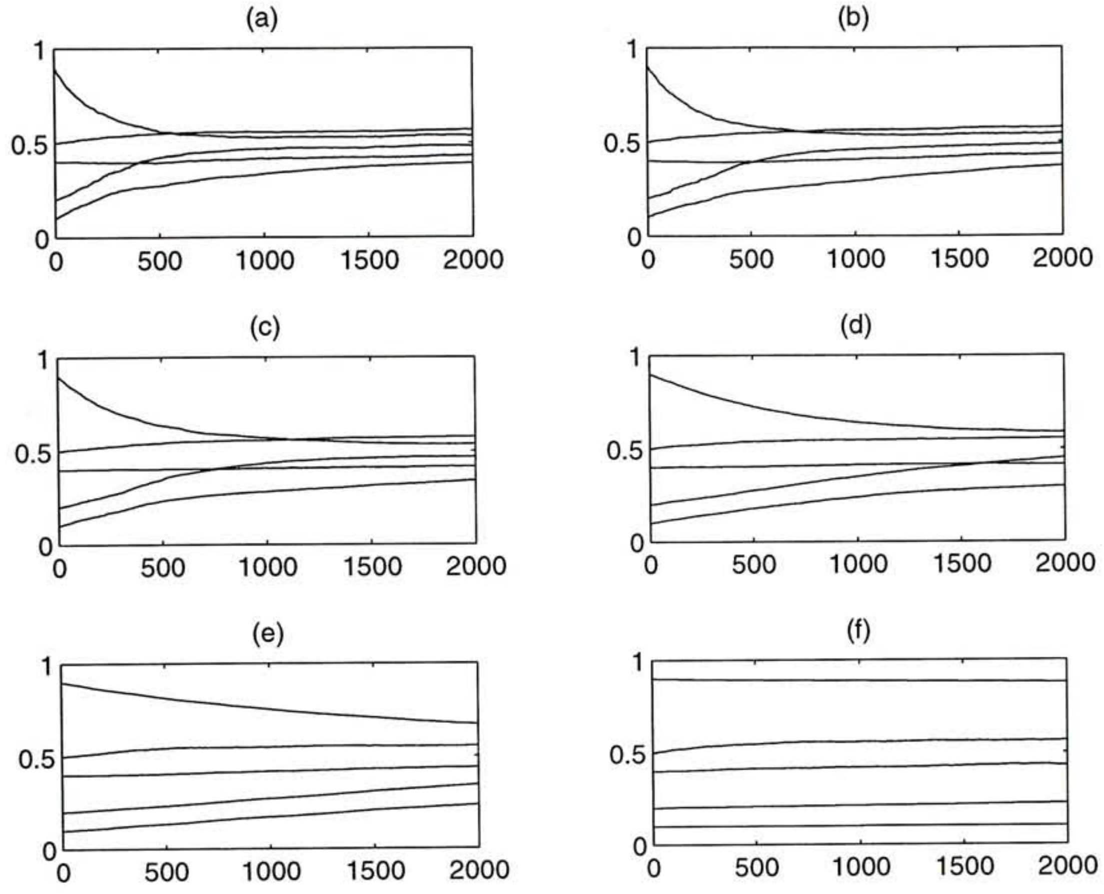


Figure 3.4: The plots of change of v_i of the $SSOM_1$ under Gaussian distribution, where mean is 0.5 and variance is 0.1. $\beta(t)$ is set to be a constant during each run. (a) $\beta(t) = 1$, (b) $\beta(t) = 0.75$, (c) $\beta(t) = 0.5$, (d) $\beta(t) = 0.25$, (e) $\beta(t) = 0.1$ and (f) $\beta(t) = 0$.The value of v_is are set to be: $v_1(0) = 0.5$, $v_2(0) = 0.9$, $v_3(0) = 0.2$, $v_4(0) = 0.4$ and $v_5(0) = 0.1$. The horizontal axis is corresponding to the number of iteration while the vertical axis is corresponding to the weight value.

3.4.2 One dimensional map under Gaussian distribution

In this example, the setting of the parameters are the same as last example except that the input data is in Gaussian distribution. The mean of the distribution is 0.5 while the variance is 0.1. The results are shown in Figure(3.4). It is observed that when $\beta > 0.5$, $SSOM_1$ can converge to an ordered map within 2000 time of iterations. Besides, the weight values get closer if β is set larger. Worthy noting that, when $\beta = 0$, the resultant weight values is similar to the first simulation example.

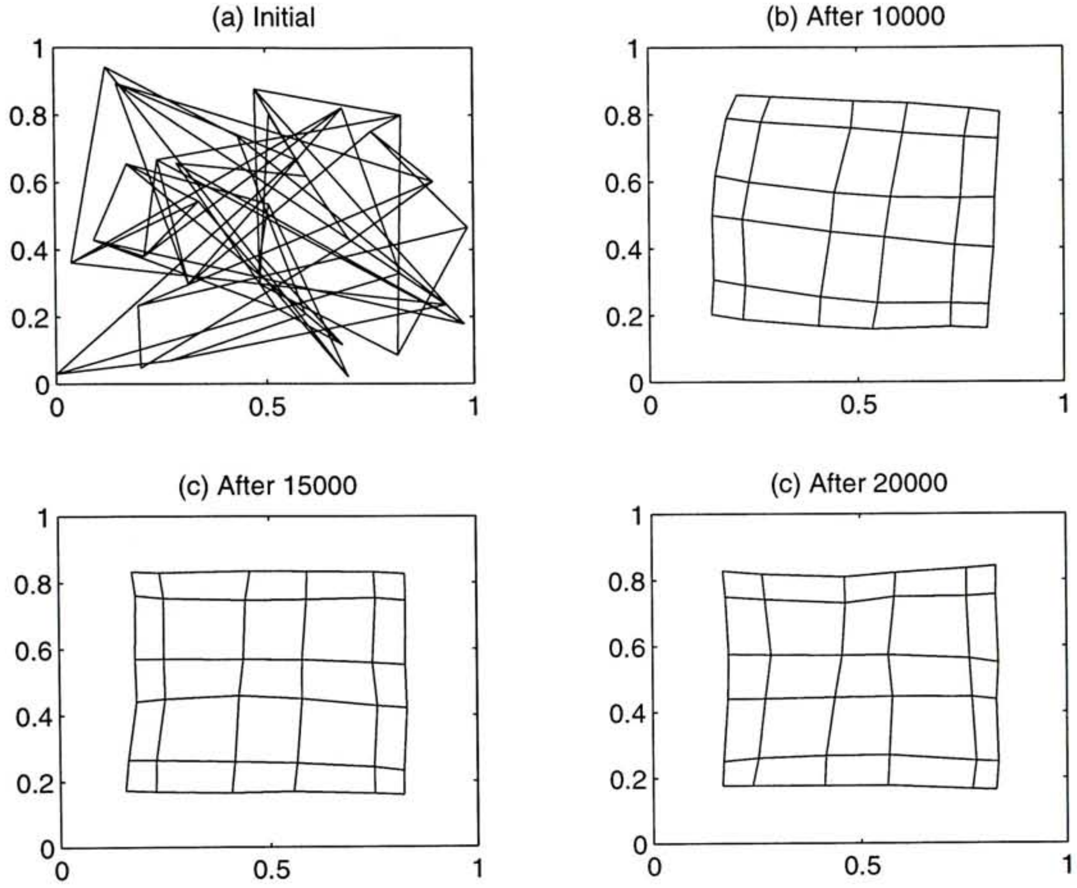


Figure 3.5: The evolution of the $SSOM_1$ under uniform distribution. (a) The map is randomly initialized and $\tau = 0.02$. (b) and (c) indicate the map formed after 10^4 and 1.5×10^4 iteration respectively. The ordered map is formed after 2×10^4 iteration and remained unchanged afterward.

3.4.3 Two dimensional map in a unit square

In this example, a 6 by 6 SSOM are initialized randomly inside a unit square. The input data is distributed uniformly inside the unit square. 8-neighbor type is used. $\beta(t)$ in (3.20) is set to be 1 and $\alpha(t)$ is 0.05. For the case of $SSOM_1$, the ordered map can be formed after 20000 iteration. Figure(3.5) displays the map structures at the 0, 10000, 15000 and 20000 iterations. For the case of $SSOM_2$, the results are displayed in Figure(3.6). The map structures formed at 0, 20000, 60000 and 100000 are shown.

Two observations can be noted from the above experiment: (i)The convergence rate of $SSOM_1$ is faster than that of $SSOM_2$. From the experiment, the time consumed by $SSOM_2$ is 5 times the time consumed by $SSOM_1$. (ii)The size of the map

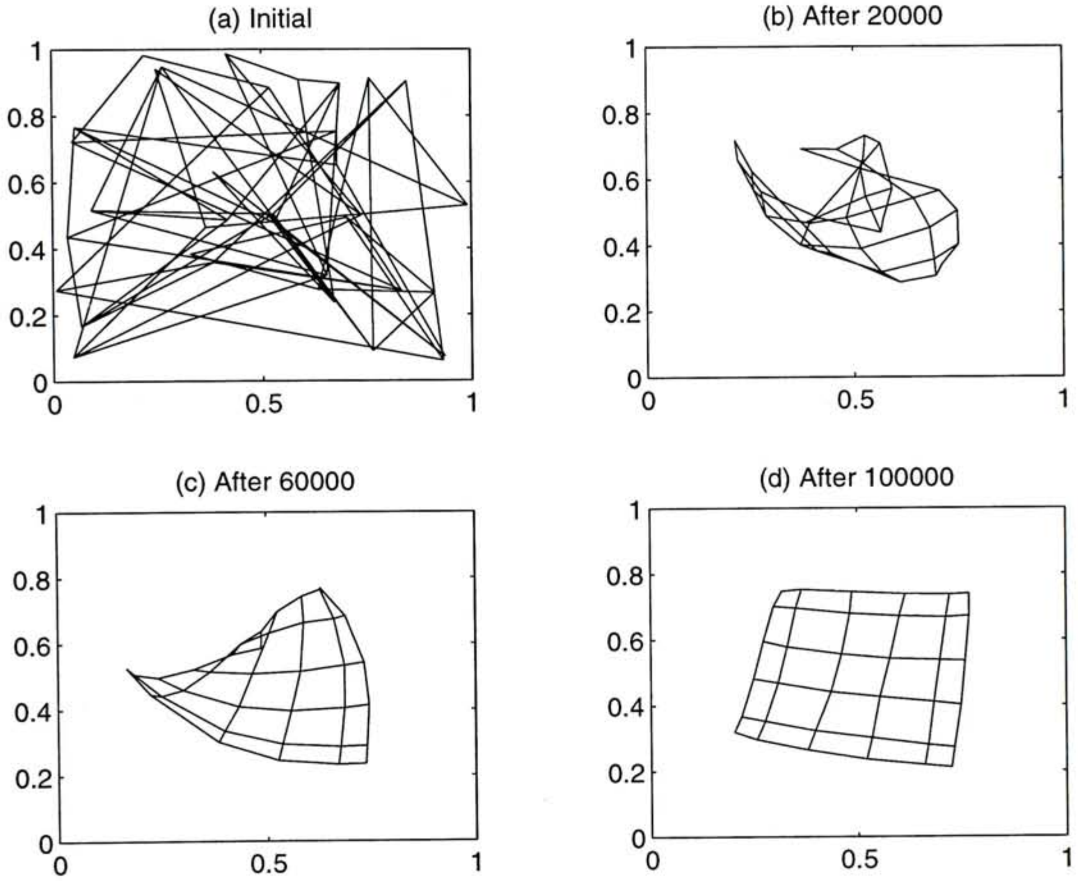


Figure 3.6: The evolution of the $SSOM_2$ under uniform distribution. The neighborhood used is the eight neighbor type. Initially, the map is randomly initialized, (a). (b) and (c) indicate the resultant shape of the map after 2×10^4 and 6×10^4 iteration respectively. The ordered map is formed after 10×10^4 iteration. And the shape of the map remains unchanged afterward.

generated by $SSOM_1$ is larger than the one generated by $SSOM_2$.

3.5 Conclusion

In summary, this chapter presents two algorithms for Soft Self Organizing Map and demonstrates their ordering properties using three simulation examples. They are inspired by the ordering property manifested by Self-Organizing Map (SOM) and the soft competition nature of Maximum Likelihood Competitive Learning (MLCL) and Fuzzy Competitive Learning (FCL). By studying the mathematical formulations of competitive learning and SOM, a cue for the formation of neighborhood preserved map is suggested. Using this cue, MLCL and FCL are used to derived two SOM-like algorithms: $SSOM_1$ and $SSOM_2$. Simulation results based on one dimension and two dimension data are provided to illustrate their ordering properties.

Although $SSOM_1$ and $SSOM_2$ show ordering properties, there is still difficulty in using $SSOM_1$. It is found that there is no simple method to tune the value of τ. We have carried out a number of simulations with different values of τ. It was found that $SSOM_1$ cannot generate topological map once τ is greater than 0.15. In case of $SSOM_2$, we have set $m = 1.5$ and $m = 3$. It was found that topological map still can be generated. Furthermore, there is no analytical proof for the ordering and convergence of both algorithms except that the neighborhood set os reduced to singleton, i.e. MLCL and FCL respectively.

Chapter 4

Application to Uncover Vowel Relationship

In the last chapter, we have derived two soft versions for Self-Organizing Map (SOM), called $SSOM_1$ and $SSOM_2$, and demonstrated its ordering property through simulations. Due to the ordering property manifested by $SSOM_1$ and $SSOM_2$, $SSOM_1$ and SSO_2 are likely applied to reveal the topological structure of a set of data, strictly the vowel data. In our preliminary study, it is found that not both $SSOM_1$ and $SSOM_2$ can uncover the neighborhood relationship among different classes of vowel provided by Peterson and Barney[1]. The algorithm of $SSOM_1$ is not feasible to do so. So, in this chapter, our principal concern is akin to the implementation of $SSOM_2$ to uncover the vowel relationship. The following sections are devoted to present the detail of this application. In the first section, the experimental set up including the network structure, training procedure and relationship construction scheme will be elucidated in section two. Then the results of the experiment will be given in section three. In which, the reason why $SSOM_1$ is not appropriate to be implemrnted will be explained. In section four, the conclusion will be presented.

[1] It is a benchmark database which is located in the UCI machine learning repository.

4.1 Experiment Set Up

Peterson-Barney vowel database is used as the training set. The database consists of a digitized version of the first and second formant frequencies of ten vowels for multiple male and female speakers. As in [33] and [31], the first and second formant frequencies are used for the experiments. It is remarked that our principal concern is to uncover the vowel relationship.

4.1.1 Network structure

The structure of the network is similar to the counter-propagation-network (CPN) [14]. Figure(4.1) shows the network structure for this experiment. It consists of three layers: input, hidden and output. There are two units in the input layer, hundred units in the hidden layer and ten units in the output layer. The hidden units are constructed as a ten by ten 2D mesh. The input-hidden weights, v_is, are determined by the algorithm of $SSOM_2$ (or SOM), i.e. equation (3.17). The hidden-output weights are determined by the method of minimum square error which will be elucidated shortly.

4.1.2 Training procedure

The training of this network is divided into two phases: (i) input-hidden weight evaluation and (ii) hidden-output weight evaluation. Denote $x \in R^2$ be the input data, $z_{(i,j)} \in R$ be the output of ijth hidden unit and $v_{(i,j)} \in R^2$ be the input-hidden weight. In the first phase, the value of $v_{(i,j)}$ are evaluated based on the algorithm of $SSOM_2$ (or SOM). For simplicity, $\alpha(t) = 0.1$ and $N_{(i,j)} = \{(i,j), (i \pm 1, j \pm 1), (i, j \pm 1), (i \pm 1, j)\}$ at the first 152000 iterations. After that, $\alpha(t) = 0.01$ and $N_{(i,j)} = \{(i,j)\}$ for another 152000 iterations. Once the first phase training is finished, the vowel data have been partitioned into hundred clusters.

Then $N_{(i,j)} = \{(i,j)\}$ and the values of v_{ij} are frozen. (At that time, if a vowel

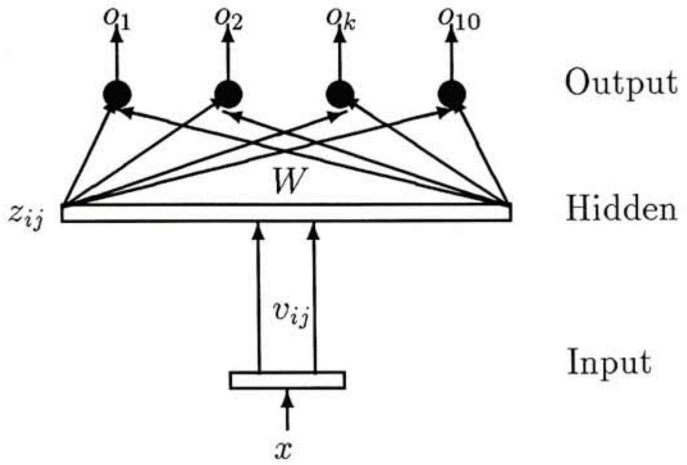


Figure 4.1: The network for the experiment. It is a three layer network with two input nodes, one hundred hidden node and ten output node. The input-hidden weights, v_{ij}s, are determined by using either SOM or $SSOM_2$ while the hidden-output weights, W, are determined by using the method of minimum square error. Here, $i, j \in \{1, 2, \ldots, 10\}$.

data x is fed to the input, each of the hidden units will output a value, z_{ij}. In case of $SSOM_2$, $z_{ij}(x) \in [0, 1]$. In case of SOM, $z_{ij}(x) \in \{0, 1\}$. In either case, we can observed that $\sum_{i,j} z_{ij}(x) = 1$.) Suppose that the total number of output node, c, is ten, we denote $O = (o_1 \ldots o_c)^T \in R^{10}$ be the output of the network, W be the hidden-output weight matrix, $Z = (z_{11} z_{12} \ldots z_{cc})^T$ and $O^1 = (10 \ldots 0)^T$, $O^2 = (01 \ldots 0)^T$, ..., $O^{10} = (00 \ldots 1)^T$ be the output of network corresponding to the ten classes. In the second phase, each of vowel data is fed to input, the values of z_{ij}s are evaluated and output from the hidden layer to the output layer. The output of the ten output units, o_ks, are compared with the target values. The error is backpropagated to modify the hidden-output weights. Suppose that the square error contributed by data x is given by $e_x = \|O^x - WZ(x)\|^2$, thus the total square error is $\sum_x \|O^x - WZ(x)\|^2$. Therefore, using the method of minimum square error, W can be determined using the following formulae:

$$W = \left(\sum_x Z^T(x) Z(x)\right)^{-1} \left(\sum_x Z^T(x) O^x\right).$$

4.1.3 Relationship Construction Scheme

Using the aforementioned network structure and training steps, we obtain a vowel classifier. Based on this classifier, the following heuristic steps are taken to reveal the vowel relationship: (i)The hidden-output weight is set to be one if it is larger than a threshold. Otherwise, it is set to be zero. (ii)The hidden unit is assigned to be class i if the weight connecting it to the ith output node is one. (iii)If there is a hidden unit which is assigned to more than one class, set threshold to a larger value and repeat the first two steps. If each hidden unit is assigned to at most one class, then go to next step. (iv)Class i and class j are neighborhood in the data space if they are neighborhood in the organizing map.

4.2 Results

Figure(4.2) and (4.3) display the clustering results of $SSOM_2$ and SOM after the first phase training. The input data are normalized. The circles are corresponding to the location of the cluster centers while the edges connecting circles indicate the neighborhood structure of the hidden units. In order to label the hidden units, we need to know the hidden-output weight values.

4.2.1 Hidden-unit labeling for $SSOM_2$

Figure(4.4) shows the mesh plot of the weight values connecting the hidden units to the first output unit[2]. It is found that large weight values are usually localized in a small region. The same property exists in the weights connecting the hidden units to other output units. Then following the hidden-unit labeling scheme (step(i) and step(ii) in the relationship construction scheme), the labeling $\mathcal{L}_{SSOM2}$ of the hidden

[2] For the rest of the other mesh plots are shown in Appendix C.

units are:

$$\mathcal{L}_{SSOM2} = \begin{bmatrix} 9 & 9 & 9 & - & - & - & - & 1 & 1 & 1 \\ - & - & 9 & - & - & 10 & 2 & 2 & - & 1 \\ 7 & 8 & 8 & - & - & - & - & 2 & 2 & 1 \\ 7 & 7 & 8 & 8 & - & 10 & 3 & - & 2 & - \\ 7 & 7 & 5 & 10 & - & - & - & - & 2 & 1 \\ - & - & 5 & - & - & - & 3 & 3 & 1 & 1 \\ - & - & - & - & 4 & 4 & - & 3 & 2 & 1 \\ 6 & 6 & - & 5 & - & 4 & - & 3 & - & 2 \\ - & - & - & 5 & 5 & 4 & 4 & - & 3 & - \\ 6 & 6 & 6 & 5 & - & 4 & 4 & 4 & 3 & - \end{bmatrix}. \tag{4.1}$$

Here the threshold is set to be one. The interpretation of this matrix is as following. Consider the element in the third column forth row, the value is 8. This means that the cluster represented by $v_{(3,4)}$ is belongs to the 8th vowel. If the element is a *dash*, it means that the corresponding cluster is unclassified.

4.2.2 Hidden-unit labeling for SOM

Figure(4.5) shows the weight values connecting the hidden units to the first output unit. Similar to the case of $SSOM_2$, it is found that the weight values which are greater than zeros are usually localized in a small region. Then following the same hidden-unit labeling scheme and the threshold is set to be 0.7, the labeling $\mathcal{L}_{SOM}$ of the hidden units are:

$$\mathcal{L}_{SOM} = \begin{bmatrix} 6 & 6 & 6 & - & - & 7 & 7 & 7 & - & 9 \\ 6 & 6 & 6 & - & - & - & 8 & 8 & 9 & 9 \\ - & - & - & 5 & 5 & - & 8 & 8 & - & 9 \\ 5 & 5 & - & - & - & 10 & - & - & - & 9 \\ 5 & - & 4 & 4 & 10 & 10 & 10 & - & - & - \\ - & - & 4 & 4 & - & - & - & 2 & 2 & - \\ - & 4 & 4 & 4 & 3 & - & 2 & 2 & 2 & 1 \\ 4 & 4 & - & - & 3 & - & - & 2 & 1 & 1 \\ 4 & 4 & 3 & 3 & 3 & 3 & 2 & 2 & 1 & 1 \\ 4 & 4 & 3 & 3 & - & - & 2 & 2 & 1 & 1 \end{bmatrix}. \tag{4.2}$$

Similarly, the value of each of the element indicates the class of the corresponding cluster belongs to. Based on the labeling matrix obtained previously, we can obtain the relational matrix $\mathcal{R}_{SSOM2}$ and $\mathcal{R}_{SOM}$ as following:

$$\mathcal{R}_{SSOM2} = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \end{bmatrix},$$

$$\mathcal{R}_{SOM} = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 \end{bmatrix}.$$

These matrix indicate the neighborhood relationship between different vowels. If the value of the ijth element is equal to one, then i class and j class are neighborhood. If the value is zero, then they are not neighborhood.

4.3 Conclusion

In summary, this chapter has presented a simple method of uncovering the relationship amongst clusters of vowel data. Furthermore, as the ordered map (the hidden layer) is predefined as a lower dimensional mesh, the relationship obtained manifests a lower dimension relationship between cluster. Although our principal concern is not to construct a vowel classifier with very high classification rate, it is worthwhile to point out that the rate of correct classification of $SSOM_2$ and SOM are 0.76 and 0.76 respectively which are far below the performance of using MLCL as indicated in [33]. The reason for this poor performance is remained to be investigated. Finally, it should note that the idea expressed in this chapter can actually be applied to those cluster based fuzzy model identification techniques [41, 5]. In particular, if the clustering technique is FCL (or FCM [2]) based, $SSOM_2$ can directly applied. In case other clustering technique is implemented, equation (3.17) needed to be modified

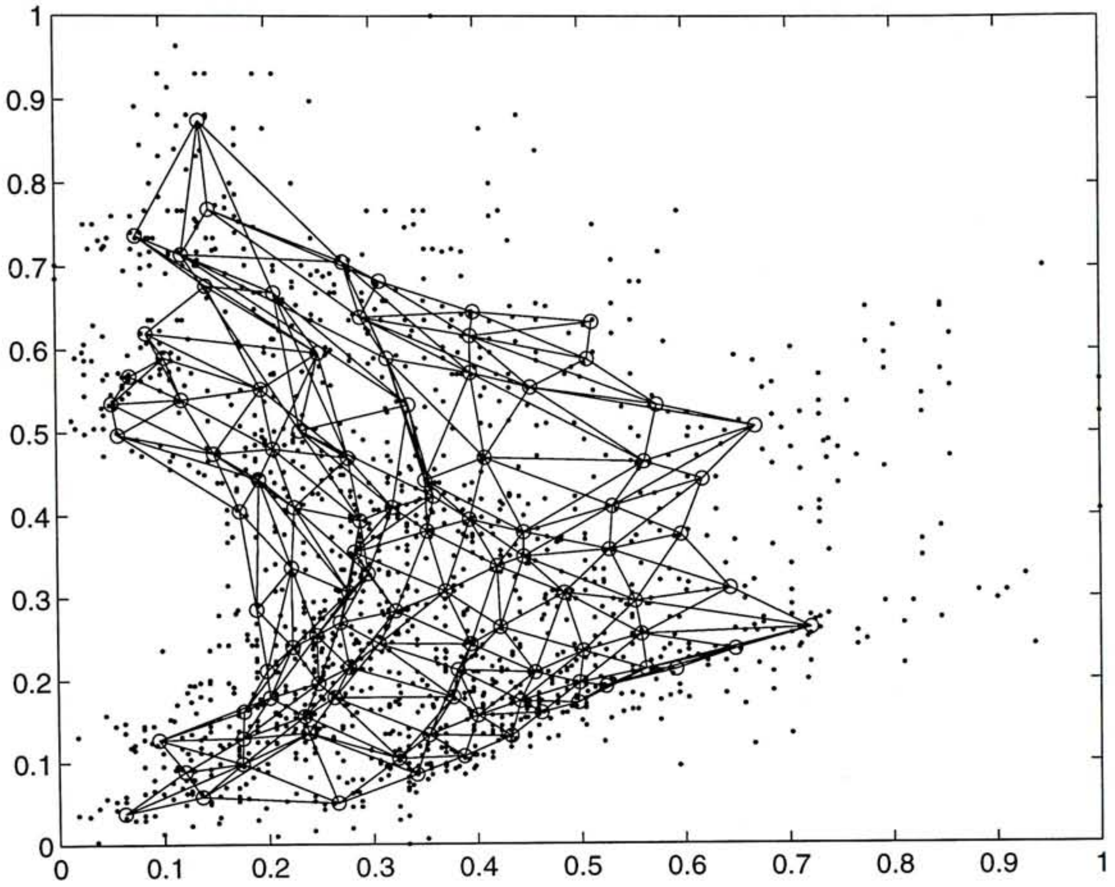


Figure 4.2: The resultant of $SSOM_2$ after training. The circles are corresponding to the location of v_{ij}. The edges connecting two circles indicate that the corresponding nodes are neighborhood in the hidden layer. Observed that there are some edges crossing over. So, the resultant map is not a nice topological map.

accordingly.

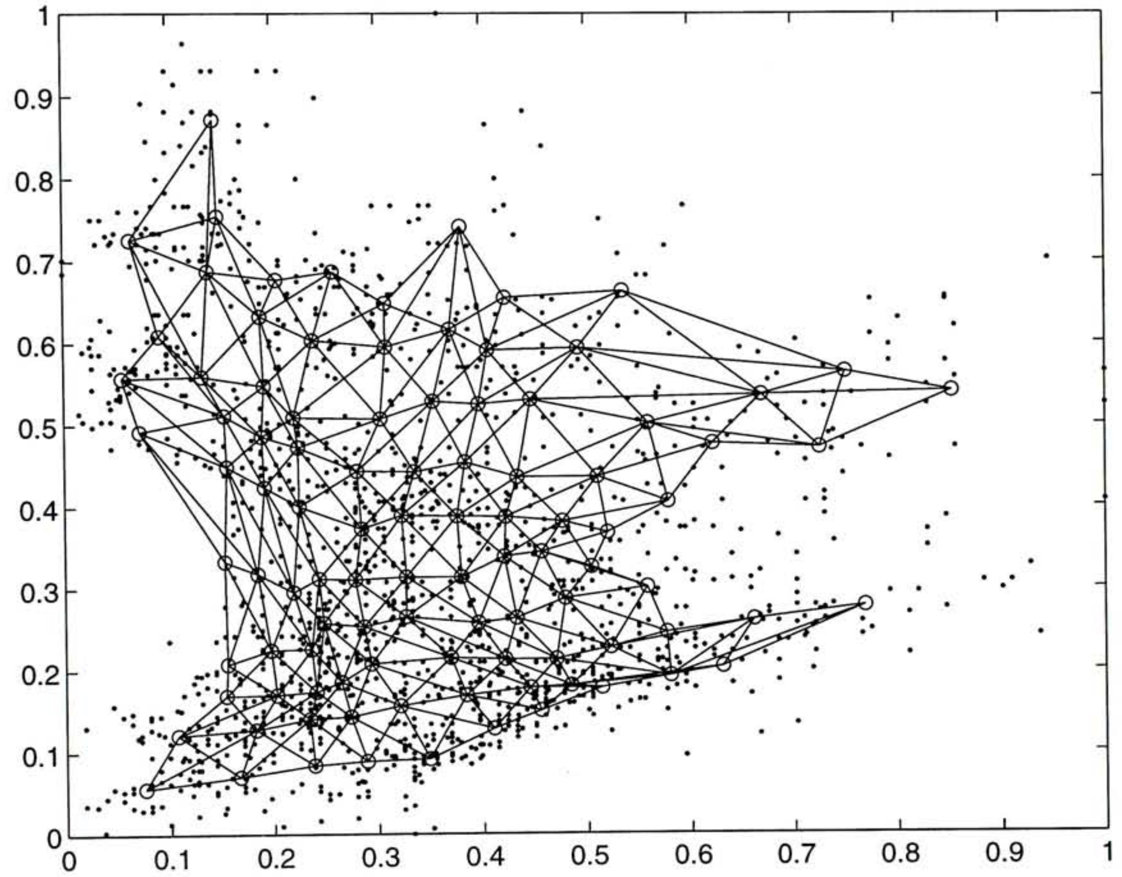


Figure 4.3: The resultant of SOM after training. The circles are corresponding to the location of v_{ij}. The edge connecting two circles indicate that the corresponding nodes are neighborhood in the hidden layer. Observed that there is no edges crossing over in the resultant map.

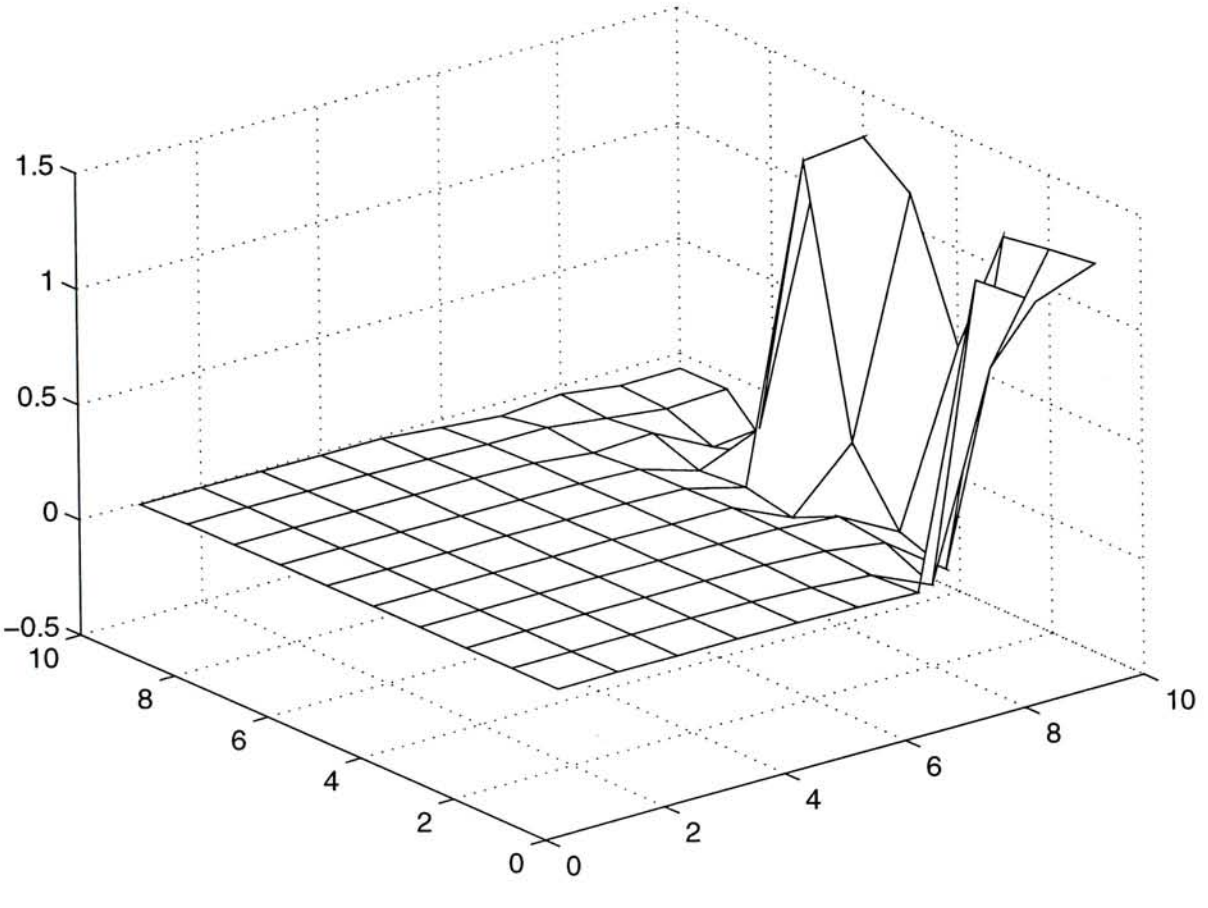


Figure 4.4: The mesh plot of the weights connection the hundred hidden unit to the first output using $SSOM_2$ as input-hidden layer.

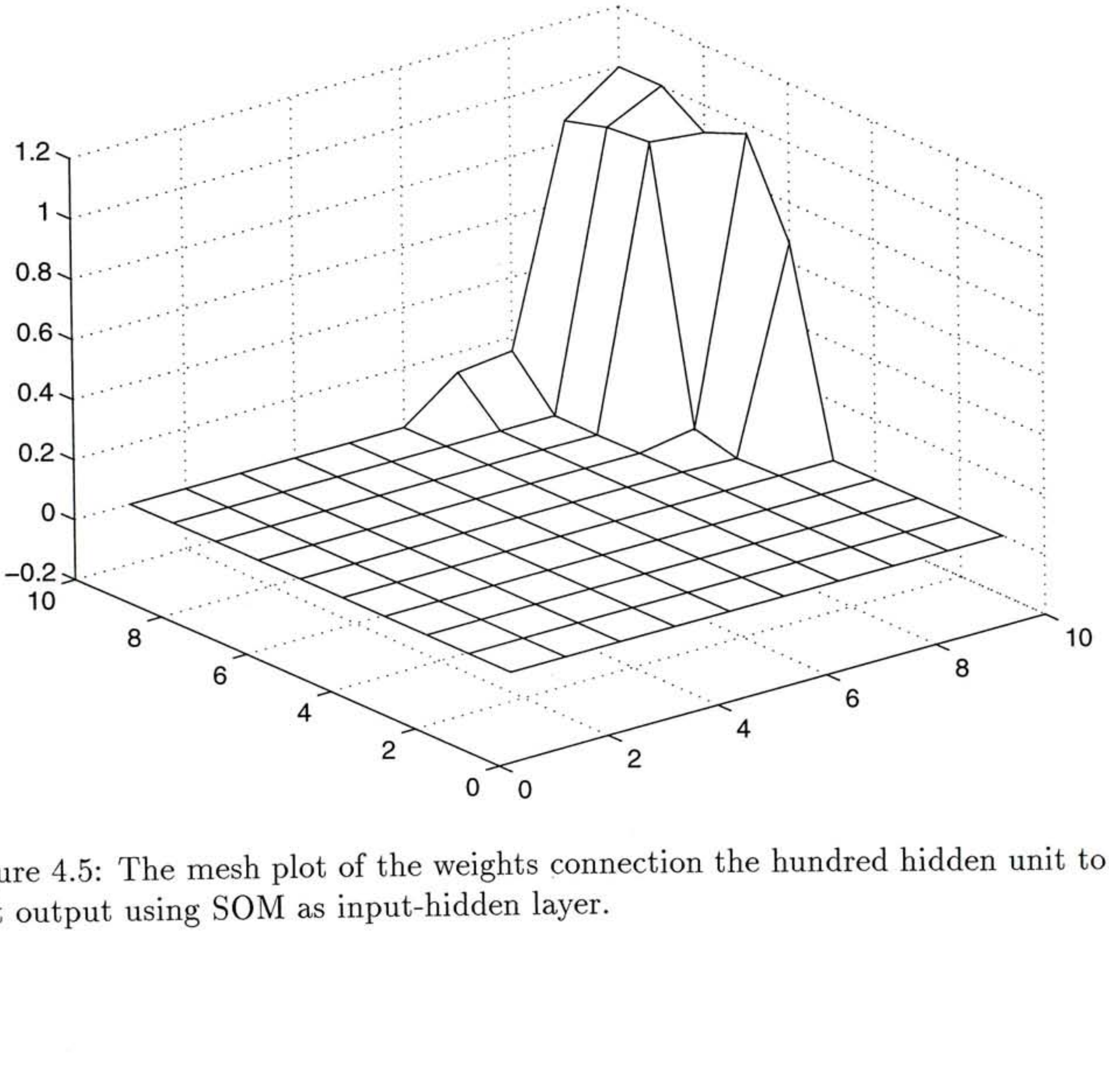


Figure 4.5: The mesh plot of the weights connection the hundred hidden unit to the first output using SOM as input-hidden layer.

Chapter 5

Application to vowel data transmission

In this chapter, we further apply these algorithms in the transmission of vowel data under a noisy channel. The quantizer codebook is generated by using $SSOM_1$ and $SSOM_2$ and embedded into modulation system. The overall system performance of using $SSOM_1$ and $SSOM_2$ in transmitting Peterson-Barney vowel data are compared with that of using SOM, in the sense of data reconstruction error. Simulation results indicate that (i)in higher channel noise level, the reconstruction error committed by using $SSOM_1$ is smaller; (ii) while in lower channel noise level, the error committed by using SOM will be smaller. In the next section, the data transmission problem and the motivation of using soft self-organizing map in generating codebook will be elucidated. Then the simulation and the main results will be described in section two and three. The conclusion will be presented in section four.

5.1 Introduction

To build a transmission system, the following steps are usually undertaken to transmit and receive a batch of data [40]: (*Quantization and Encoding*) A vector quantizer is

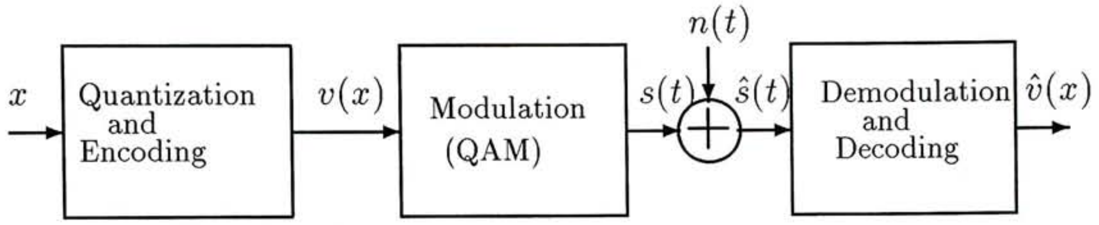


Figure 5.1: The block diagram of a simple transmission system.

designed to divide the data space into a number of partitions, $\mathcal{P}_1, \ldots, \mathcal{P}_c$. Each of these partitions is represented by a representative vector, $v_1, \ldots, v_c$. Once a data x is going to be sent, the quantizer will classify x into one of these partitions based on nearest neighbor scheme: If v_i is the closest representative vector to x, $x \in \mathcal{P}_i$ and the data x is encoded by symbol i. (*Modulation*) The code of the data x is passed to the modulator. The modulator then generates and transmits a waveform $s(t)$ to the communication channel, usually noisy. (*Demodulation and Decoding*) In the destination side, the receiver demodulates the contaminated signal $s(t) + n(t)$ and gives out a code. Based on this code, the approximation of data x, $\hat{v}(x)$ is reconstructed. Figure(5.1) shows the block diagram of a simple transmission system.

In traditional approach, the design of the vector quantizer is independent of the design of the modulator in such transmission system. Recently, Leung [25] has put them as a whole for consideration: the vector quantizer is generated by using Self-Organizing Map (SOM) while the modulator is designed based on quadrature amplitude modulation (QAM). In this chapter, we follow the same idea as suggested by him to design the transmission system. Apart from using SOM to train the quantizer, we apply two algorithms of soft SOM to build such quantizer. And their performance are evaluated.

The reason why we follow Leung's idea can be explained as following. Consider that the QAM is a 16-ary with waveforms defined by $s_{ij}(t) = a_i cos\omega_c t + a_j sin\omega_c t$, where $a_i = 1.5 - (i-1)$ and ω_c is the carrier frequency. The quantizer consists of sixteen codevectors, $\{v_{11}, v_{12}, \ldots, v_{44}\}$, which are generated by using SOM. While this quantizer is implemented in QAM, the following codevector-waveform assignment (CWA) is defined: $v_{ij} \mapsto s_{ij}$. Usually, we also denote s_{ij} as a two dimensional vector in the signal space: $s_{ij} = (a_i, a_j)$. In sequel, if v_{ij} and v_{rs} are neighborhood in the sense of SOM, s_{ij} and s_{rs} are neighborhood in the signal space. Consequently, if x is classified as ijth partition and being transmitted by the waveform $s_{ij} = (a_i, a_j)$ to a noisy channel, the reconstructed codevector at the destination will probably be neighborhood of $\hat{v}_{ij}$ according to the fact that the received waveform should not be too far from s_{ij}. As a result, the distance between x and $\hat{v}_{ij}$ should not be large. Hence the reconstruction error will be smaller. On the contrary, in case SOM is not applied, 's_{ij} and s_{rs} are neighborhood' could not imply 'v_{ij} and v_{rs} are neighborhood'. So, the error contributed by $\|x - \hat{v}_{ij}\|$ will probably be large. Therefore, the total reconstruction error using non-map type training algorithm, Frequency sensitive competitive learning (FSCL) for instance, will be large [25].

Although Leung has shown promising result, we would like to further reduce the reconstruction error by substituting the algorithms of $SSOM_1$ and $SSOM_2$ to SOM in the generation of codebook for QAM. This substitution is motivated by three current results : In [50], Yair et.al. showed that better codebook can be accomplished by using soft competition (i.e. Maximum Likelihood Competitive Learning (MLCL)). In [33], Nowlan applied MLCL to classify vowel data and found that the misclassification rate is far below than using other neural network approach including SOM. Similar result was claimed by Chung and Lee [7]. They applied Fuzzy Competitive Learning (FCL) to classify vowel data and found that the missclassification rate is lower than that of LVQ and FSCL. Based on their results, it is reasonable to induce that using the soft boundary technique (MLCL or FCL) in clustering or clustering based classification may achieve better performance. In conclusion, the combination of soft competition mechanism with topological ordering mechanism may improve the performance of data transmission under noise channel.

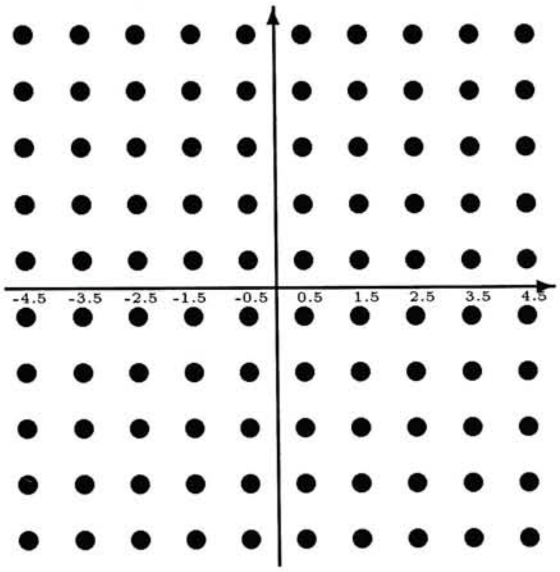


Figure 5.2: The location of the waveforms are defined regularly in the signal space. Each solid circle represents one waveform.

5.2 Simulation

5.2.1 Setup

As $SSOM_1$, $SSOM_2$ and SOM can cluster data in an ordered fashion, three different codebooks are generated by using $SSOM_1$, $SSOM_2$ and SOM respectively. Half of the database is used for training. Each of these codebooks contains 100 codevectors. Each organized map is defined as 10 by 10 mesh. We denote the codevectors by $v_{11}, v_{12}, \ldots, v_{10,10}$. The QAM is designed as 100-ary. The waveforms are defined regularly in the signal space: $\forall i, j = 1, 2, \ldots, 10, s_{ij}(t) = a_i cos\omega_c t + a_j sin\omega_c t$, where $a_i = 4.5 - (i - 1)$. The CWA is simply defined by associating s_{ij} to v_{ij}. For instance, to transmit the codevector v_{23}, the waveform $3.5cos\omega_c t + 2.5sin\omega_c t$ will be sent, Figure(5.2) shows the locations of the waveforms in the signal space.

5.2.2 Noise model and demodulation scheme

Without loss of generality, the channel noise is modeled as a two-dimensional additive white Gaussian noise (AWGN), i.e. $n = (n_1, n_2)$, where $E\{n_1(t)\} = E\{n_2(t)\} = 0$ and $E\{n_1^2(t)\} = E\{n_2^2(t)\} = \sigma^2$. Here σ is the standard deviation of the channel noise in each dimension. When a data x is transmitted to the destination, the following steps will be simulated: (i)$x \mapsto v_{ij}(x)$ (x is quantized to one of the codevectors in the codebook.), (ii)$v_{ij} \mapsto s_{ij}(t)$ (The corresponding waveform will be transmitted.), (iii)$s_{ij} \mapsto s = s_{ij} + (n_1, n_2)$ (AWGN is added to the transmitted waveform.) and (iv)$s \mapsto \hat{v}_{ij}$ (The received waveform is demodulated[1].).

5.2.3 Performance index

To evaluate the performance of the three algorithms in data transmission, the following performance index is considered:

$$E_1 = \frac{1}{N}\sum_{i=1}^{N} \|\hat{v}(x_i) - x_i\|^2,$$

where N is the total number of training data. With reference to Figure(5.1), $\hat{v}(x_i)$ is the output signal after demodulation and decoding. x_i is the input to the quantizer and $v(x_i)$ is the input to the modulator. The former index measures the mean square reconstruction error. Smaller the valuer of E_1, the better the transmission system.

5.2.4 Control experiment: random coding scheme

In order to demonstrate the advantage of topological order. A control experiment is carried out. The set up is the same as above except that the CWA is defined arbitrary. We call it random coding scheme (RC). In such case, the neighborhood preservation property is ceased. If the quantized vector v_{23} and v_{24} are fed to the

[1] Remind that s_{rs} can be written as (a_r, a_s). The demodulation scheme is defined as following: If $\|s - \hat{s}_{ij}\| \leq \|s - s_{rs}\|$ for all $r, s = 1, 2, \ldots, 10$, the corresponding $\hat{v}_{ij}$ will be treated as the reconstruction of x.

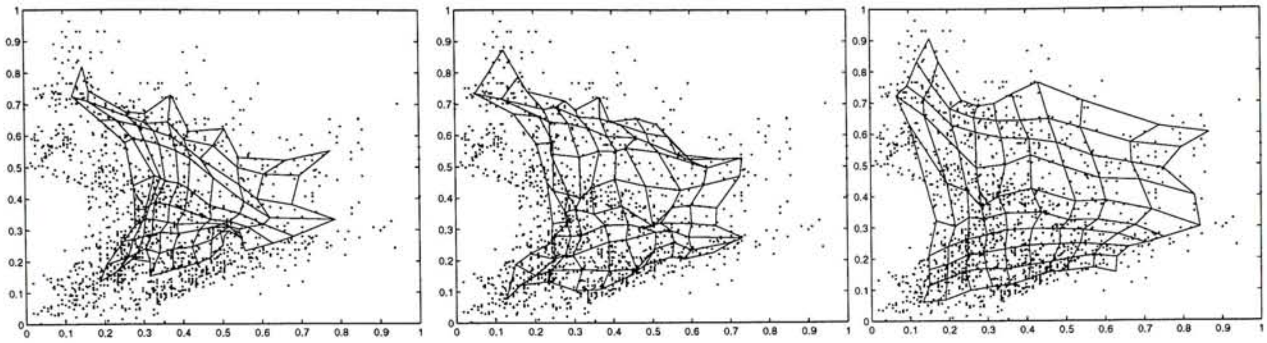


Figure 5.3: The resultant maps of $SSOM_1$(left), $SSOM_2$(middle) and SOM(right) after training.

encoder consecutively, the outcome will no more be $(2,3)$ and $(2,4)$. Instead, they may be $(5,9)$ and $(1,10)$ which are depended on definition of the one to one corresponding.

5.3 Results

To clarify the discussion, the resultant codebooks generated by $SSOM_1$, $SSOM_2$ and SOM are displayed in Figure(5.3). Remark that the maps obtained are different from those displayed in the last chapter since the size of the training set is just in size compared with the experiment carried in the last chapter. The locations of the small circles are corresponding to the locations of the codevectors. The edges indicate the neighborhood relationship between codevectors, which are defined a priori. The dots are corresponding to the locations of the vowel data. We set the standard deviation of the channel to 29 different values: $0.0, 0.1, 0.2, \ldots 2.0$ *and* $: 3, 4, \ldots 10$. Twenty-nine sets of experiments are then carried out. The resultant E_1 are plotted against the channel noise standard deviation in Figure(5.4). The numerical data of tabulated in the following tables, Table 5.1 to Table 5.2.

5.3.1 Null channel noise ($\sigma = 0$)

While the channel noise is null, the reconstruction error is purely quantization error. It is found that the quantization error committed by $SSOM_1$ is the largest: $E_1 = 0.0479$.

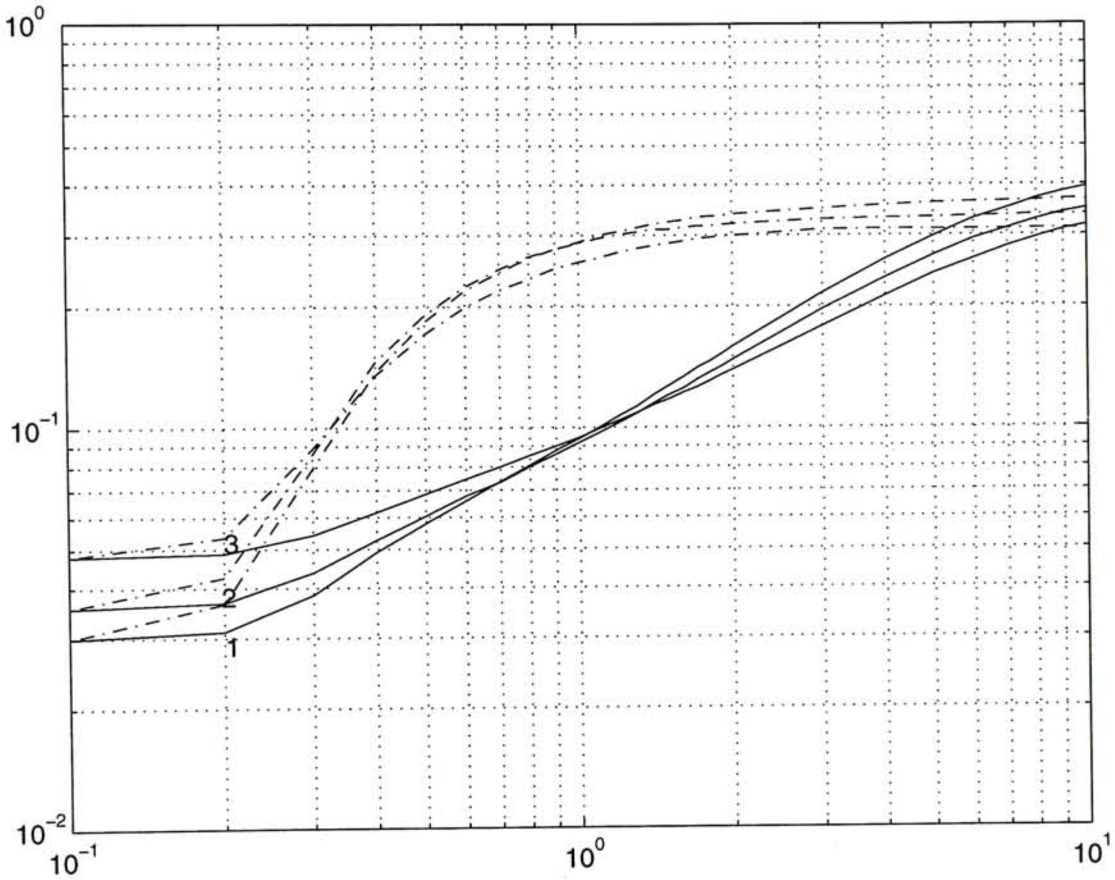


Figure 5.4: The performance of transmission system against channel noise. The horizontal axis is corresponding to channel noise variance while the veritcal axis is corresponding to the average mean square error, E_1. The results obtained by using SOM are indicated by line (1). The results obtained by using $SSOM_1$ are indicated by line (3). The results obtained by using $SSOM_2$ are indicated by line (2). The solid lines are corresponding to the case when the codevector waveform assignment follows neighborhood preservation scheme. While the dash lines are corresponding to the case when the codevector waveform assignment is random.

The quantization error generated by using SOM is the smallest, $E_1 = 0.03$.

5.3.2 Small channel noise $(0 < \sigma \leq 1)$

For the case that ordered map is implemented as the quantizer, Figure(5.4), it is experimented that the *reconstruction error of the system is dominated by the quantization error* when the channel noise level is low, i.e. $\sigma \leq 1$. For the case that the ordering topology is ceased the situation is the similar but the reconstruction error is dominated by the quantization error only when $\sigma \leq 0.35$. In summary, when the channel noise is small, the order of the reconstruction error E_1 is that

$$E_1(SOM) < E_1(SSOM_2) < E_1(SSOM_1) < E_1(RC),$$

when RC stands for random coding scheme.

5.3.3 Large channel noise $(1 \leq \sigma \leq 7)$

When the standard deviation of the channel noise is between 1 to roughly 7, the performance of ordered map is still better than random coding:

$$E_1(SSOM_1) < E_1(SSOM_2) < E_1(SOM) < E_1(RC).$$

5.3.4 Very large channel noise $(\sigma > 7)$

When the channel noise is very large, it is found that the mean square error obtained by all three algorithms are similar. While their results are compared with random coding scheme, it is found that the error obtained are larger. A possible reason for this aspect is due to the locations of the codevectors. However, very large channel noise seems to be infeasible in real situation. This results are just for reference.

5.4 Conclusion

In summary, we have presented one important application of the Soft SOM in this chapter. It is to implement Soft SOM as the quantizer in the data communication system. Combine the QAM modulation technique, it is demonstrated that the reconstruction error can largely be reduced when the channel noise level is low. Besides, it is found that overall system performance is less noise sensitive. For example, if we set $E_1 = 0.1$ as a reference limit, it is found that the channel noise tolerated by using random code technique is less than 0.3. In case of ordered map technique, it increases to a value larger than one. Moreover, when the channel noise is very large, i.e. larger than 7, it is observed that all three ordering technique cannot help to reduce the reconstruction error due to channel noise.

S.D.	SOM	$SSOM_2$	$SSOM_1$
0.0000	0.0300	0.0357	0.0479
0.1000	0.0300	0.0357	0.0479
0.2000	0.0312	0.0368	0.0487
0.3000	0.0384	0.0438	0.0542
0.4000	0.0491	0.0527	0.0617
0.5000	0.0579	0.0604	0.0685
0.6000	0.0658	0.0678	0.0746
0.7000	0.0736	0.0733	0.0800
0.8000	0.0807	0.0800	0.0853
0.9000	0.0878	0.0859	0.0899
1.0000	0.0940	0.0918	0.0945
1.1000	0.1005	0.0976	0.0998
1.2000	0.1073	0.1032	0.1045
1.3000	0.1133	0.1089	0.1091
1.4000	0.1207	0.1151	0.1140
1.5000	0.1274	0.1207	0.1178
1.6000	0.1338	0.1253	0.1224
1.7000	0.1409	0.1317	0.1260
1.8000	0.1461	0.1371	0.1305
1.9000	0.1524	0.1420	0.1354
2.0000	0.1591	0.1475	0.1392
3.0000	0.2156	0.1965	0.1784
4.0000	0.2614	0.2353	0.2121
5.0000	0.2981	0.2673	0.2409
6.0000	0.3292	0.2934	0.2617
7.0000	0.3491	0.3113	0.2809
8.0000	0.3695	0.3277	0.2957
9.0000	0.3835	0.3402	0.3086
10.0000	0.3945	0.3487	0.3167

Table 5.1: The performance of transmission system. The results are obtained by using $SSOM_1$, $SSOM_2$ and SOM. The performance index E_1.

S.D.	SOM	$SSOM_2$	$SSOM_1$
0.0000	0.0300	0.0357	0.0479
0.1000	0.0300	0.0357	0.0479
0.2000	0.0366	0.0425	0.0534
0.3000	0.0794	0.0861	0.0888
0.4000	0.1373	0.1460	0.1351
0.5000	0.1833	0.1905	0.1710
0.6000	0.2191	0.2240	0.1994
0.7000	0.2431	0.2473	0.2197
0.8000	0.2638	0.2659	0.2339
0.9000	0.2776	0.2775	0.2486
1.0000	0.2888	0.2868	0.2575
1.1000	0.3005	0.2957	0.2644
1.2000	0.3069	0.3018	0.2728
1.3000	0.3138	0.3060	0.2784
1.4000	0.3198	0.3107	0.2832
1.5000	0.3242	0.3107	0.2869
1.6000	0.3272	0.3143	0.2904
1.7000	0.3327	0.3163	0.2949
1.8000	0.3345	0.3165	0.2963
1.9000	0.3351	0.3190	0.2986
2.0000	0.3384	0.3206	0.2996
3.0000	0.3499	0.3287	0.3101
4.0000	0.3564	0.3315	0.3113
5.0000	0.3605	0.3323	0.3105
6.0000	0.3624	0.3340	0.3102
7.0000	0.3636	0.3353	0.3106
8.0000	0.3652	0.3376	0.3111
9.0000	0.3679	0.3362	0.3115
10.0000	0.3673	0.3379	0.3114

Table 5.2: The performance of transmission system in the sense of mean square error. The results are obtained by using random coding technique. The performance index E_1.

Chapter 6

Convergence Analysis

This chapter discusses certain theoretical results on the convergence of the competitive learning, soft competitive learning, SOM and Soft SOM. The technique is based on the application of Kushner-Clark Lemma and the Lyapunov indirect method. The first section states the Kushner-Clark Lemma. The convergence conditions of Jou's algorithm are presented in second section. For our best knowledge, no researcher has proven these conditions yet. In section 3, we extend the result obtained in section 2 to provide an alternative proof on the convergence of competitive learning. Section 4 presents one of the main results of this thesis: the convergence of SSOM. Based on the same approach as section 3, we extend the result in section 4 to prove the convergence of SOM which is presented in section 5.

6.1 Kushner and Clark Lemma

The following Lemma is adopted from [23]. For the sake of application, some irrelevant terms and conditions are ignored in order to simplify the convergence proof. To see the completed version of the Lemma and its proof, please refer to [23].

Lemma 1 (Theorem 2.3.1 of [23]) *Let $\{M_t\}$ be given by*

$$M_{t+1} = M_t + \alpha_t h(M_t) + \alpha_t \epsilon_t.$$

And assume that

KC1 *$h(.)$ is continuous R^n valued function on R^n.*

KC2 *α_t is a sequence of positive real numbers such that $\sum_t \alpha_t = \infty$ and $\sum_t \alpha_t^2 < \infty$.*

KC3 *$\{\epsilon_t\}$ is a sequence of R^r valued random variables and such that for each $\delta > 0$*

$$\lim_{t \to \infty} P\left\{ \sup_{m \geq t} \mid \sum_{i=t}^{m} \alpha_i \epsilon_i \mid \geq \delta \right\} = 0,$$

Suppose that $\{M_t\}$ is bounded with probability one. Let M_0 be a locally asymptotically stable solution to

$$\dot{M} = h(M),$$

with domain of attraction $DA(M_0)$. Then if $A \subset DA(M_0)$ such that $M_t \in A$, we have $M_t \to M_0$ as $n \to \infty$.

□ □ □

6.2 Condition for the Convergence of Jou's Algorithm

Theorem 3 *The convergence of Jou's fuzzy competitive learning algorithm, (3.6) and (3.7), is almost sure.*

(Proof) We assume that all v_i are bounded. Consider equations: (3.6) and (3.7), the associated different equation is that

$$\frac{\partial J_m}{\partial v_i} = -2E[y_i^m(x)(x - v_i(t))]. \tag{6.1}$$

And it is just the case that

$$\frac{dv_i}{dt} = -\frac{\partial J_m}{\partial v_i}.$$

This is a descent algorithm which $\frac{dJ_m}{dt} \leq 0$. According to Lyapunov indirect method, J_m is the Lyapunov function and there exists $D_s = \{V | \forall\ V \in [0,1]^n, \frac{\partial J_m}{\partial V} = 0\}$, where $V = (v_1, v_2, \ldots, v_c)^T$.

Next, it is going to checked that (3.6) satisfies condition KC1 to KC3 of Kushner and Clark Lemma. Denote $h(V) = (h_1(V), h_2(V), \ldots, h_c(V))^T$, where

$$h_i(V) = -2E[y_i^m(x)(x - v_i(t))]. \tag{6.2}$$

Moreover, let us denote $\epsilon = (\epsilon_1, \epsilon_2, \ldots, \epsilon_c)^T$, where

$$\epsilon_i(V, x) = -2y_i^m(x)(x - v_i(t)) - h_i(V). \tag{6.3}$$

Since we can fix the $\alpha(t)$ according to A.2, for instance $\sum_k \alpha_k^2 < \infty$ and $\sum_{k=1}^{\infty} \alpha_k = \infty$, and then check that $h(V)$ is continuous function. The major proof is A.3. From (6.3), we can define a stochastic process $\{S_\pi\}$ by

$$\sum_{i=n}^{\pi} \alpha_i \epsilon(V, x(i)).$$

Obviously, $E[S_{\pi+1}] = S_\pi$. $\{S_\pi\}$ is a Martingale Process. Based on Martingale Inequality [9],

$$P\left\{\sup_{\pi \geq n} |\sum_{i=n}^{\pi} \alpha(i)\epsilon(V, x(i))| \geq \varepsilon\right\} \leq \lim_{\pi \to \infty} \frac{E|S_\pi|^2}{\varepsilon^2},$$

for all $\varepsilon > 0$. As $h(V)$ and M are bounded, $\epsilon(V, x)$ is bounded. Hence,

$$P\left\{\sup_{\pi \geq n} |\sum_{i=n}^{\pi} \alpha(i)\epsilon(V, x(i))| \geq \varepsilon\right\} \leq k_3 \frac{\sum_{i=n}^{\infty} {\alpha_i}^2}{\varepsilon^2},$$

where k_3 is a constant. Moreover, $\sum_\pi \alpha_\pi^2 < \infty$ implies that

$$\lim_{n \to \infty} \sum_{k=n}^{\infty} \alpha_k^2 = 0.$$

Therefore,

$$\lim_{n \to \infty} P\left\{\sup_{\pi \geq n} |\sum_{i=n}^{\pi} \alpha(i)\epsilon(V, x(i))| \geq \varepsilon\right\} \leq \lim_{n \to \infty} k_3 \frac{\sum_{i=n}^{\infty} \alpha_i^2}{\varepsilon^2}.$$

implies that

$$\lim_{n \to \infty} P\left\{\sup_{\pi \geq n} |\sum_{i=n}^{\pi} \alpha(i)\epsilon(V, x(i))| \geq \varepsilon\right\} = 0.$$

It satisfies A.3. According to Kushner and Clark Lemma, it can be concluded that the convergence of FCL is almost sure. Besides, $\lim_{n\to\infty} V(n) \in D_s$, where D_s is the stable invariant set.

□

Corollary 2 *The convergence condition for the Jou's fuzzy competitive learning algorithm is that*

- $\sum_k \alpha_k^2 < \infty$ *and*
- $\sum_{k=1}^{\infty} \alpha_k = \infty$.

In [6] and [16], the authors did not prove the convergence of FCL is almost sure. Hence, the condition on the step size is not provided. They only set the step size, $\alpha(t)$, equal to a small constant which cannot guarantee that the convergence is almost sure.

6.3 Alternative Proof on the Convergence of Competitive Learning

Using the above theorems, it is possible to extend the result to prove that the convergence of competitive learning is almost sure. The mechanism of competitive learning is stated as (2.1) by setting $N_I = \{I\}$. In such case,

$$\frac{dV(t)}{dt} = g(V(t)), \tag{6.4}$$

where $g_i(V(t)) = \int_{\Omega_i}(x - v_i)f(x)dx$. From (6.2),

$$\lim_{m\to 1^+} h(V(t); m) = g(V(t)). \tag{6.5}$$

Theorem 4 *Once the assumption of Lemma 1 are satisfied, the convergence of (6.4) is almost sure.*

(Proof) In the same approach as Lemma 1, the only need to prove is to show that (6.4) is stable. So what we need to prove is that

$$\lim_{m\to 1^+} J_m(V)$$

is the Lyapunov function of (6.4). According to (6.5),

$$\lim_{\Delta t\to 0}\lim_{m\to 1^+} V_h(t+\Delta t) = \lim_{\Delta t\to 0} V_g(t+\Delta t),$$

where $V_h(s)$ and $V_g(s)$ are the solutions of Jou's algorithm and CL with initial conditions $V_h(t) = V_g(t)$. Moreover, as $J_m(V)$ is continuos for all $m > 1$,

$$\lim_{\Delta t\to 0}\lim_{m\to 1^+} J_m(V_h(t+\Delta t)) = \lim_{\Delta t\to 0} J_m(V_g(t+\Delta t)),$$

and

$$\lim_{\Delta t\to 0}\lim_{m\to 1^+} J_m(V_h(t+\Delta t)) = \lim_{m\to 1^+}\lim_{\Delta t\to 0} J_m(V_h(t+\Delta t))$$

Hence

$$\lim_{\Delta t\to 0}\lim_{m\to 1^+} \left[J_m(V_g(t+\Delta t) - J_m(V_g(t))\right] = \lim_{m\to 1^+}\lim_{\Delta t\to 0} \left[J_m(V_h(t+\Delta t) - J_m(V_h(t))\right].$$

So that,

$$\begin{aligned}\lim_{\Delta t\to 0} \frac{\lim_{m\to 1^+} \left[J_m(V_g(t+\Delta t) - J_m(V_g(t))\right]}{\Delta t} &= \lim_{m\to 1^+}\lim_{\Delta t\to 0} \frac{\left[J_m(V_h(t+\Delta t) - J_m(V_h(t))\right]}{\Delta t} \\ &< 0. \qquad (6.6)\end{aligned}$$

Hence

$$\frac{d}{dt}\lim_{m\to 1^+} J_m(V_g(t)) < 0.$$

The system (6.4) is stable and the convergence of competitive learning is almost sure. And the proof is completed.

□

6.4 Convergence of Soft SOM

The proof of convergence of soft SOM is based on the application of Kushner-Clark Lemma (Lemma 1) and the following Lemma.

Lemma 2 *Consider a stable gradient system,*

$$-\frac{\partial J}{\partial V} = \frac{d}{dt}V(t) = h(V(t)), \tag{6.7}$$

where J is the lyapunov function of the system. The perturbated system is given by

$$\frac{d}{dt}V(t) = h(V(t)) + \beta(t)P(V(t)). \tag{6.8}$$

If (i) $\lim_{t\to\infty}\beta(t) = 0$ for all $t > t_0$, and (ii) $\|P(V(t))\| < \infty$ then (6.8) can converge to stable state.

(Proof) Let $e(t)$ be the difference of the stable system and the perturbated system, $e(t)$ is given by

$$\|e(t)\| = \|\int_{t_0}^{t} \beta(s)P(V(s))ds\|.$$

Since $\beta(t) > 0$ and $P(V(t))$ is bounded,

$$\|e(t)\| < M\int_{t_0}^{t}\beta(s)ds$$

where $M = max\|P(V(t))\|$.

$$\begin{aligned}\frac{dJ}{dt} &= \frac{\partial J}{\partial V}\frac{dV}{dt} \\ &= -h^T(V)\left(h(V) + \beta(t)P(V)\right) \\ &= -h^T(V)h(V) - \beta(t)h^T(V)P(V)\end{aligned} \tag{6.9}$$

as $t \to \infty$, $\beta(t) \to zero$. Hence $\frac{dJ}{dt} < 0$. As (6.7) is stable, (6.8) is also a stable system as t_0 is sufficient large. And the proof is completed.

□ □□

Now, the mechanism of $SSOM_2$ can be defined as following:

$$v_i(t+1) = v_i(t) + \alpha(t)\left[y_i^m(x; V(t)) + \beta(t) \sum_{k \in N_i \setminus \{i\}} y_k^m(x; V(t))\right](x - v_i(t)), \quad (6.10)$$

where $V(t) = (v_1(t), \dots, v_c(t))^T$, $\alpha(t)$ satisfies the assumptions of Lemma 1 and $\beta(t)$ satisfies the assumptions of 2.

Corollary 3 *The convergence of $SSOM_2$ (6.10) is almost sure if $\alpha(t)$ and $\beta(t)$ satisfy the assumptions of Lemma 1 and Lemma 2 respectively.*

(Proof) Denote

$$h(V(t)) = (h_1(V(t)), \dots, h_c(V(t)))^T$$

and

$$P(V(t)) = (p_1(V(t)), \dots, p_c(V(t)))^T,$$

where

$$h_i(V(t)) = y_i^m(x; V(t))(x - v_i(t))$$

and

$$p_i(V(t)) = \sum_{k \in N_i \setminus \{i\}} y_k^m(x; V(t))(x - v_i(t)).$$

Based on Lemma 2,

$$\frac{d}{dt}V(t) = h(V(t)) + \beta(t)P(V(t))$$

is a stable system as t is sufficiently large. Then using Lemma 1, $SSOM_1$ converges almost sure. The proof is completed.

□

In the same manner, if $y_i(x)$ is defined as the way in Nowlan's MLCL, the mechanism $SSOM_1$ can be written as that

$$v_i(t+1) = v_i(t) + \alpha(t)\left[y_i(x; V(t)) + \beta(t) \sum_{k \in N_i \setminus \{i\}} y_k(x; V(t))\right](x - v_i(t)), \quad (6.11)$$

the convergence is again almost sure.

Corollary 4 *The convergence of $SSOM_1$ (6.11) is almost sure if $\alpha(t)$ and $\beta(t)$ satisfy the assumptions of 2 and Lemma 1 respectively.*

□

6.5 Convergence of SOM

As $m \to 1^+$, (6.10) reduces to the algorithm of SOM. Hence, the convergence of SOM can be proven using the same approach as in the proof of competitive learning. Without loss of generality, the convergence property of SOM is stated below without proof.

Corollary 5 *The convergence of SOM, where $\alpha(t)$ and $\beta(t)$ satisfy the assumptions of Lemma 2 and Lemma 1 respectively, is almost sure.*

□

Chapter 7

Conclusion

We have presented a softing version of SOM and demonstrated the ordering property through a number of simulations and applications. As SSOM is an extension of SOM and soft competition (MLCL and FCL), the algorithms of SOM, MLCL and FCL were studied. (Note that the motive of using FCL is due to its soft competition nature but not its fuzzy background.) Based on the relationship between SOM and competitive learning, a simple scheme of modification of MLCL and FCL is proposed to form SOM-like MLCL ($SSOM_1$) and SOM-like FCL ($SSOM_2$).

The application of SSOM in cluster analysis has been presented. It is shown that $SSOM_1$is unable to reveal the intrinsic structure of a batch of data although MLCL itself is outperformed in vowel data classification. On the other hand, although SOM is poorly performed in vowel data classification, SOM manifests two advantages: (i) The values of hidden-output weight are between zero and one. (ii) Localization effect is shown in the hidden-output weights which can help to sketch the neighborhood relationship amongst clusters of labeled data.

SSOM is also applied to reduce the channel noise effect in data communication. The method of the combination of topological map and QAM in data transmission has been proposed by C.S.Leung [24] [25]. Here, we follow the steps Leung proposed. It is found that $SSOM_1$ is better noise tolerated than $SSOM_2$ and SOM when the channel noise level is high. When the channel noise level is low, SOM is the best. Note that the performance of SOM, $SSOM_1$ and $SSOM_2$ rely on the shapes of the organized maps. The results reported in this thesis is just an example.

Apart from the development of the model SSOM, this thesis provides certain theoretical results supplementary to the model of SOM. In the Appendix A, we follow the approach of Bouton and Pages [4] to prove that the convergence of *1D SOM* is almost sure even if the neighborhood set size is not finite. Besides, an energy function is constructed for the 1D SOM when the input data distribution is uniform. Hence, the convergence of *1D SOM* is globally almost sure if the input data distribution is uniform.

The convergence of higher dimension SOM, SSOM, FCL and CL has been discussed in chapter six. Denote $\alpha(k)$ to be the step size, we proved that if $\sum_{k=1}^{\infty} \alpha(k) = \infty$ and $\sum_{k=1}^{\infty} \alpha^2(k) < \infty$, then the convergence of FCL and CL are globally almost sure. In case of SOM and SSOM, we prove that the convergence are locally almost sure.

7.1 Limitations of SSOM

There are three limitations in the application of SSOM in clustering:

- **Universal Approximation** In Chapter four, although the capability of SSOM in revealing the topological relationship has been demonstrated, an assumption on the approximation capability has been made. We assume that the hybrid network shown in Figure(4.1) is universal approximators. In fact, this assumption can only be valid for the cases when $SSOM_1$ and SOM are implemented as the input-hidden layer. For $SSOM_1$, the hybrid network behaves the same as Radial Basis Function net. The output is the summation of radial basis function (3.5). Hence the universal approximation property is guarantee [47]. For SOM, as the output of each of the hidden unit is rectangular, so it is possible to prove the universal approximation property using the same technique as in [47]. As a result, both $SSOM_1$ and SOM implemented hybrid networks are universal approximators. However, in so far, there is no proof on the universal approximation property of using (3.7) as basis function, for the best knowledge of the author. The conclusion that we made in Chapter four is therefore based on the assumption that the hybrid network, shown in Figure(4.1), implemented by $SSOM_2$ can be an universal approximator.

- **Ordering Property** In this thesis, we have not presented any theoretical proof on the ordering property of SSOM and SOM. The ordering property is solely demonstrated by simulation examples and application examples. Be aware that all the examples shown are in the dimension of two.

- **Computational Speed** As the competition mechanism is soft, the time consumed for building a map is much longer than using the convention winner-take-all rule. The reason can be conceived as following. Suppose the map size is $n \times n$ and eight-neighbor is implemented. In case of SOM, ignoring the time for finding the winner and the identification of neighborhood, the number of node to be updated is nine. Assume that the NIF is a step function with value one, no more mathematical operation. In case of SSOM, the number of node to be updated is $n \times n$. For each of the node, eight addition-operation are required. As a result, $8n^2$ extra addition-operation are needed which make the training time for SSOM is much longer than SOM.

7.2 Further Research

According to the last section, it is obvious that one possible further research is to do a bit theoretical analysis on the property of SSOM. Besides, using SSOM in construction of relational matrix, we have to define the training in two-phases fashion. The hidden-output weights are determined after the clustering is finished. It comes out a problem when the number of data is very large. In this case, an on-line training seems to be necessary.

Appendix A

Proof of Corollary 1

Essentially, this Appendix is devoted to the proof of Corollary 1. The proof of Corollary is in fact an implication of Corollary 1. The proof of Corollary 1 is divided into three sections, from the second section to the forth section according to the distribution discussed: uniform, logconcave and loglinear. When the input data distribution is uniform, an energy function can be constructed by using Krasovskii method to show that the convergence is global. The extension on the recent results are listed following for clarity:

- Extension of Bouton-Pages results [4] [8] on the convergence of 1-D Map to the case when the size of neighborhood is any large (Corollary 1).
- Simplifying the proof of Bouton-Pages Theorem [4] on the log-concave input distribution case by introducing the Trushkin Lemma [43] (Corollary 1).
- Extension of Bouton-Pages [4] result on uniform input distribution by construction an objective function using Krasovskii method [18]. Hence, the global convergence of SOM can be guaranteed.

This Appendix is composed of four sections. In the first section, the mean update of the SOM mechanism is derived. Following the *same* technique as in Bouton-Pages paper [4], the convergence proof on the uniform, logconcave and loglinear input distribution cases are presented in section two. In section two, two new results are also presented as well: (i) the size of neighborhood can be any large and (ii) the convergence of one dimensional SOM is globally almost sure. The case of logconcave input distribution and loglinear distribution are proven in section three and four respectively.

A.1 Mean Average Update

Since input x is a random variable, the updating of $V(t)$ is indeed a stochastic recursive algorithm. Suppose that the distribution of x is $f(x)$, the mean average update is that

$$E[V(t+1)] = V(t) + \alpha(t)\int_0^1 \Lambda(x,V)[xu - V(t)]f(x)dx,$$

where

$$\Lambda(x,M) = \sum_{k=1}^{c} \Phi_k(x(t))\Lambda_k.$$

Here $\Phi_k(x(t))$ is an indicator function defined as

$$\Phi_k(x(t)) = \begin{cases} 1 & \text{if } \|x(t) - v_k(t)\| = min_i\|x(t) - v_i(t)\| \\ 0 & \text{otherwise} \end{cases}.$$

Since $[0,1] = \bigcup_{k=1}^{c} \Omega_k$,

$$E[V(t+1)] = V(t) + \alpha(t)\sum_{k=1}^{c}\int_{\Omega_k} \Lambda_k[xu - V(t)]f(x)dx, \tag{A.1}$$

where $f(x)$ is the probability density function of x. And the algorithm of SOM can also be rewritten as

$$V(t+1) = V(t) + \alpha(t)[h(V(t)) - \epsilon(V(t), x(t))], \tag{A.2}$$

where

$$h(V) = \sum_{k=1}^{c}\int_{\Omega_k} \Lambda_k[xu - V(t)]f(x)dx \tag{A.3}$$

and

$$\epsilon(V(t), x(t)) = \Lambda(x, V)[x(t)u - V(t)] - \sum_{k=1}^{c} \int_{\Omega_k} \Lambda_k[xu - V(t)]f(x)dx. \tag{A.4}$$

Furthermore, denote $h(V) = (h_1(V), h_2(V), \ldots, h_c(V))^T$ where

$$\begin{aligned} h_1(V) &= \beta_0 \int_{\Omega_1} (x - v_1)f(x)dx + \beta_1 \int_{\Omega_2} (x - v_1)f(x)dx + \ldots \\ &+ \beta_l \int_{\Omega_{l+1}} (x - v_1)f(x)dx \end{aligned} \tag{A.5}$$

$$\begin{aligned} h_2(V) &= \beta_1 \int_{\Omega_1} (x - v_2)f(x)dx + \beta_0 \int_{\Omega_2} (x - v_2)f(x)dx + \ldots \\ &+ \beta_l \int_{\Omega_{l+2}} (x - v_2)f(x)dx \end{aligned} \tag{A.6}$$

$$\ldots$$

$$\begin{aligned} h_i(V) &= \beta_l \int_{\Omega_{i-l}} (x - v_i)f(x)dx + \ldots + \beta_1 \int_{\Omega_{i-1}} (x - v_i)f(x)dx \\ &+ \beta_0 \int_{\Omega_i} (x - v_i)f(x)dx + \beta_1 \int_{\Omega_{i+1}} (x - v_i)f(x)dx \\ &+ \ldots + \beta_l \int_{\Omega_{i+l}} (x - v_i)f(x)dx \end{aligned} \tag{A.7}$$

$$\ldots$$

$$\begin{aligned} h_c(V) &= \beta_l \int_{\Omega_{c-l}} (x - v_c)f(x)dx + \beta_{l-1} \int_{\Omega_{c-2}} (x - v_c)f(x)dx + \ldots \\ &+ \beta_0 \int_{\Omega_c} (x - v_c)f(x)dx. \end{aligned} \tag{A.8}$$

Hence $h(V)$ can be rewritten as

$$h(V) = (\beta_0 - \beta_1)h^{(0)}(V) + (\beta_1 - \beta_2)h^{(1)}(V) + \ldots + (\beta_{l-1} - \beta_l)h^{(l-1)}(V) + \beta_l h^{(l)}(V), \tag{A.9}$$

where $h^{(k)}(V) = (h_1^{(k)}(V), h_2^{(k)}(V), \ldots, h_c^{(k)}(V))^T$ for all $1 \leq k \leq l$, and

$$h_i{}^{(k)}(V) = \begin{cases} \int_0^{\frac{v_{i+k}+v_{i+k+1}}{2}} (x - v_i) f(x) dx & \forall 1 \leq i < k+1 \\ \\ \int_{\frac{v_{i-k}+v_{i-k-1}}{2}}^{\frac{v_{i+k}+v_{i+k+1}}{2}} (x - v_i) f(x) dx & \forall k+1 \leq i \leq c-k-1 \\ \\ \int_{\frac{v_{i-k}+v_{i-k-1}}{2}}^{1} (x - v_i) f(x) dx & \forall c-k-1 < i \leq c \end{cases} . \tag{A.10}$$

We obtain the associated differential equation

$$\frac{d}{dt} V = h(V). \tag{A.11}$$

for SOM algorithm and the invariant set $D_C = \{V | \forall V \in D_A, h(V) = 0\}$.

A.2 Case 1: Uniform Distribution

Substitute $f(x) = 1$ into (A.10), we get

$$h^{(0)}(V) = \left(\frac{1}{8}\right) \begin{bmatrix} (v_1 + v_2)(v_2 - 3v_1) \\ (v_1 + v_3)(v_1 - 2v_2 + v_3) \\ (v_2 + v_4)(v_2 - 2v_3 + v_4) \\ \cdots \\ (v_{c-2} + v_c)(v_{c-2} - 2v_{c-1} + v_c) \\ (v_{c-1} + v_c)(2 + v_{c-1} - 3v_c) \end{bmatrix}. \tag{A.12}$$

When $k = 1, 2, \ldots, l$,

$$h_i^{(k)}(V) = \left(\frac{1}{8}\right) (v_{i+k} + v_{i+k+1} - 4v_i)(v_{i+k} + v_{i+k+1}) \tag{A.13}$$

for $1 \leq i \leq k$;

$$\begin{aligned} h_i^{(k)}(V) &= \left(\frac{1}{8}\right) (v_{i+k} + v_{i+k+1} - v_{i-k-1} - v_{i-k}) \\ &\times \ (v_{i+k} + v_{i+k+1} - 4v_i + v_{i-k-1} + v_{i-k}) \end{aligned} \tag{A.14}$$

for $k+1 \leq i \leq c-k-1$ and

$$h_i^{(k)}(V) = \left(\frac{1}{8}\right)(2 - v_{i-k-1} - v_{i-k})(2 - 4v_i + v_{i-k-1} + v_{i-k}) \tag{A.15}$$

for $c-k \leq i \leq n$. Recall that

$$h(V) = \sum_{k=1}^{l}(\beta_{k-1} - \beta_k)h^{(k)}(V).$$

Taking partial derivative of (A.13) to (A.15), it can no difficult to check that $\frac{\partial h^{(k)}(V)}{\partial V}$ *is negative definite* whenever $\beta_i \geq \beta_{i+1}$ for all $0 \leq i < l$ and the fact that

$$\begin{array}{ll} \forall 1 \leq i \leq k & q_{i+k+1} - q_{i-k} > q_{i+k+1} - v_i \\ \forall k+1 \leq i \leq c-k-1 & q_{i+k+1} - q_{i-k} = (q_{i+k+1} - v_i) + (v_i - q_{i-k}) \\ \forall c-k \leq i \leq c & q_{i+k+1} - q_{i-k} > (v_i - q_{i-k}), \end{array} \tag{A.16}$$

for all $0 \leq k \leq l$. Hence $\frac{\partial h(V)}{\partial V}$ is strictly negative definite whenever $l > 0$. Let us construct an scalar function $J(V) = h^T(V)h(V)$. Obviously, it is greater than zero and $J(V) = 0$ when $h(V) = 0$. Taking the derivative of $J(V)$ with respect to t,

$$\frac{d}{dt}J(V) = h^T(V)\left[\left(\frac{\partial h(V)}{\partial V}\right)^T + \left(\frac{\partial h(V)}{\partial V}\right)\right]h(V) \tag{A.17}$$

Since $\frac{\partial h(V)}{\partial V}$ is strictly negative definite, $\frac{d}{dt}J(V) \leq 0$ and equality holds if and only if $h(V) = 0$. Therefore it can be concluded that $J(V)$ is a Lyapunov function for (A.11). In other word, (A.11) is a gradient system minimizing $J(V)$ when $f(x) = 1$. Similar to case 1, there exists $D_s \subset D_C$ such that $\lim_{t\to\infty} V(t) = V^*$ where $V^* \in D_s \subset D_C$. Furthermore, D_s is asymptotically stable in large. Using the same argument as Bouton and Page, the convergence of SOM is globally almost sure in the sense of Kushner and Clark. And the proof is completed.

□

The construction of $J(V)$ is actually based on Krasovskii method [18]. Other approachs to the proof on this case have been done by a number of researchers [4] [8] [21] [28] but all of them only show that the stability is local. Moreover, they assume that $\beta_i = 1$ for all $i = 0, 1, \ldots, l-1$ except [28]. In [28], they prove only that $\beta_0 > \beta_1$ and $\beta_i = 0$ for all $i \geq 2$.

A.3 Case 2: Logconcave Distribution

To visualize the proof, we consider a smaller size map where $c = 5$. Hence,

$$h(V) = (h_1(V), h_2(V), h_3(V), h_4(V), h_5(V))^T, \tag{A.18}$$

where

$$h_i(V) = \int_{\Omega_i} (x - v_i) f(x) dx, \tag{A.19}$$

for all $i = 1, 2, 3, 4, 5$. Recall that $\Omega_1 = \left[0, \frac{v_1+v_2}{2}\right), \Omega_k = \left[\frac{v_{k-1}+v_k}{2}, \frac{v_k+v_{k+1}}{2}\right)$ for all $2 \le k \le c-1$ and $\Omega_c = \left[\frac{v_{c-1}+v_c}{2}, 1\right]$. Hence

$$h_1 = h_1(v_1, v_2, v_3),$$

$$h_2 = h_2(v_2, v_3, v_4),$$

$$h_3 = h_3(v_1, v_2, v_3, v_4, v_5),$$

$$h_4 = h_4(v_2, v_3, v_4),$$

and

$$h_5 = h_5(v_3, v_4, v_5).$$

Taking the partial derivative of equation(A.18) with respect to V, we get the Jacobian matrix

$$\begin{aligned} \frac{\partial h}{\partial V} &= \left(\frac{\partial h_i}{\partial v_j}\right)_{5\times 5} \\ &= \begin{bmatrix} \frac{\partial h_1}{\partial v_1} & \frac{\partial h_1}{\partial v_2} & \frac{\partial h_1}{\partial v_3} & 0 & 0 \\ 0 & \frac{\partial h_2}{\partial v_2} & \frac{\partial h_2}{\partial v_3} & \frac{\partial h_2}{\partial v_4} & 0 \\ \frac{\partial h_3}{\partial v_1} & \frac{\partial h_3}{\partial v_2} & \frac{\partial h_3}{\partial v_3} & \frac{\partial h_3}{\partial v_4} & \frac{\partial h_3}{\partial v_5} \\ 0 & \frac{\partial h_4}{\partial v_2} & \frac{\partial h_4}{\partial v_3} & \frac{\partial h_4}{\partial v_4} & 0 \\ 0 & 0 & \frac{\partial h_5}{\partial v_3} & \frac{\partial h_5}{\partial v_4} & \frac{\partial h_5}{\partial v_5} \end{bmatrix} \end{aligned} \tag{A.20}$$

where

$$\frac{\partial h_1}{\partial v_1} = -\int_0^{(v_2+v_3)/2} p(x)\, dx$$

$$\frac{\partial h_1}{\partial v_2} = \frac{\partial h_1}{\partial v_3} = \frac{1}{2}\left(\frac{v_2+v_3}{2} - v_1\right) p\left(\frac{v_2+v_3}{2}\right)$$

$$\frac{\partial h_2}{\partial v_2} = -\int_0^{(v_3+v_4)/2} p(x)\, dx$$

$$\frac{\partial h_2}{\partial v_3} = \frac{\partial h_2}{\partial v_4} = \frac{1}{2}\left(\frac{v_3+v_4}{2} - v_2\right) p\left(\frac{v_3+v_4}{2}\right)$$

$$\frac{\partial h_3}{\partial v_1} = \frac{\partial h_3}{\partial v_2} = \frac{1}{2}\left(\frac{v_1+v_2}{2} - v_3\right) p\left(\frac{v_1+v_2}{2}\right)$$

$$\frac{\partial h_3}{\partial v_3} = -\int_{(v_1+v_2)/2}^{(v_4+v_5)/2} p(x)\, dx$$

$$\frac{\partial h_3}{\partial v_4} = \frac{\partial h_3}{\partial v_5} = \frac{1}{2}\left(\frac{v_4+v_5}{2} - v_3\right) p\left(\frac{v_4+v_5}{2}\right)$$

$$\frac{\partial h_4}{\partial v_2} = \frac{\partial h_4}{\partial v_3} = \frac{1}{2}\left(\frac{v_2+v_3}{2} - v_4\right) p\left(\frac{v_2+v_3}{2}\right)$$

$$\frac{\partial h_4}{\partial v_4} = -\int_{(v_2+v_3)/2}^{1} p(x)\, dx$$

$$\frac{\partial h_5}{\partial v_3} = \frac{\partial h_5}{\partial v_4} = \frac{1}{2}\left(\frac{v_3+v_4}{2} - v_5\right) p\left(\frac{v_3+v_4}{2}\right)$$

$$\frac{\partial h_5}{\partial v_5} = -\int_{(v_3+v_4)/2}^{1} p(x)\, dx$$

To justify the stability of the equilibrium points, we put M_0 into the Jacobian matrix at those equilibrium points, $\frac{\partial h}{\partial V}$. According to Trushkin Lemma (Theorem 5), it can be easily shown that $\frac{\partial h}{\partial V}$ is strictly diagonal dominant matrix with negative diagonal elements. Based on the Gerschgorin's Theorem (see section 7.3 of [12]), it can be shown that all the eigenvalues of $\frac{\partial h}{\partial V}$ are strictly in the negative complex plane. Therefore, it can conclude that all the equilibrium points are asymptotically stable by using Lyapunov linearization method [18]. Hence the proof is completed.

□

Without loss of generality, the proof can be extended to N_I in any size by using the following Trushkin Lemma [43]:

Theorem 5 (Trushkin Lemma[43]) *If a continuous function $f(x)$ is defined on a closed interval $[a, b]$, where either $-\infty \leq a < b < +\infty$ or $-\infty < a < b \leq +\infty$, $f(x) > 0$ for every $x \in (a, b)$, $f(-\infty) = f(+\infty) = 0$ and*

$$Z_0 = \int_a^b f(x)dx < +\infty,$$

$$Z_1 = \int_a^b xf(x)dx < +\infty,$$

then if $\log f(x)$ is a concave function on the interval (a, b) then

$$Z_0 > f(a)(Z_1/Z_0 - a) + f(b)(b - Z_1/Z_0).$$

A.4 Case 3: Loglinear Distribution

While the distribution is loglinear, only local stability is achieved. The proof is accomplished by substitution

$$f(x) = c_0 e^{sx}$$

into equation (A.11), where $c_0 = \left[\int_0^1 \exp(sx)dx\right]^{-1}$ and $s \neq 0$. For simplicity, we only prove the case that $c = 5$ and $l = 1$. However, the proof can easily be extended to whatever $c > 0$ and $l > 1$. Before analysis the behavior of (A.11) for loglinear distribution, let us deduce several equations which are useful for the proof. First of all, consider that

$$h(a, b, v) = c_o \int_a^b (x - v)\exp(sx)dx. \tag{A.21}$$

Note that (A.21) is h_i if we put $a = q_{i-k}$, $b = q_{i+k+1}$ and $v = v_i$. Differentiate (A.21) with respect to v, a and b,

$$\frac{\partial h}{\partial v} = \frac{c_o}{s}\left(\exp(sa) - \exp(sb)\right),$$

$$\frac{\partial h}{\partial a} = -c_o(a - v)\exp(sa),$$

and

$$\frac{\partial h}{\partial b} = c_o(b - v)\exp(sb).$$

Integrating (A.21) by part and set $h(a, b, v) = 0$, the solution, v, is given by

$$v = \frac{b\exp(sb) - a\exp(sa)}{\exp(sb) - \exp(sa)} - \frac{1}{s}. \tag{A.22}$$

With the above equalities, we can proceed to the proof. In order to illustrate clearly the step of proof, we set $c = 5$ and $0 < v_1(0) < v_2(0) < v_3(0) < v_4(0) < v_5(0) < 1$,

$$\begin{aligned}
\dot{v}_1 &= c_0 \int_0^{q_2} (x - v_1)\exp(sx)dx; \\
\dot{v}_2 &= c_0 \int_0^{q_3} (x - v_2)\exp(sx)dx; \\
\dot{v}_3 &= c_0 \int_{q_1}^{q_4} (x - v_3)\exp(sx)dx; \\
\dot{v}_4 &= c_0 \int_{q_2}^{1} (x - v_4)\exp(sx)dx; \\
\dot{v}_5 &= c_0 \int_{q_3}^{1} (x - v_5)\exp(sx)dx.
\end{aligned}$$

The Jacobian matrix at the equilibrium point, V_0, is that

$$\frac{\partial h}{\partial V}|_{V=V_0} = \left[\begin{array}{ccccc}
\frac{\partial h_1}{\partial v_1} & \frac{\partial h_1}{\partial v_2} & \frac{\partial h_1}{\partial v_3} & 0 & 0 \\
0 & \frac{\partial h_2}{\partial v_2} & \frac{\partial h_2}{\partial v_3} & \frac{\partial h_2}{\partial v_4} & 0 \\
\frac{\partial h_3}{\partial v_1} & \frac{\partial h_3}{\partial v_2} & \frac{\partial h_3}{\partial v_3} & \frac{\partial h_3}{\partial v_4} & \frac{\partial h_3}{\partial v_5} \\
0 & \frac{\partial h_4}{\partial v_2} & \frac{\partial h_4}{\partial v_3} & \frac{\partial h_4}{\partial v_4} & 0 \\
0 & 0 & \frac{\partial h_5}{\partial v_3} & \frac{\partial h_5}{\partial v_4} & \frac{\partial h_5}{\partial v_5}
\end{array}\right]_{v=v_0}$$

where

$$\begin{aligned}
\frac{\partial h_1}{\partial v_1} &= \frac{\exp(sq_0) - \exp(sq_2)}{s}, \\
\frac{\partial h_1}{\partial v_2} = \frac{\partial h_1}{\partial v_3} &= \frac{1}{2}(q_2 - v_1)\exp(sq_2), \\
\frac{\partial h_2}{\partial v_2} &= \frac{\exp(sq_0) - \exp(sq_3)}{s},
\end{aligned}$$

$$\frac{\partial h_2}{\partial v_3} = \frac{\partial h_2}{\partial v_4} = \frac{1}{2}(q_3 - v_2)\exp(sq_3),$$
$$\frac{\partial h_3}{\partial v_1} = \frac{\partial h_3}{\partial v_2} = -\frac{1}{2}(q_1 - v_3)\exp(sq_1),$$
$$\frac{\partial h_3}{\partial v_3} = \frac{\exp(sq_1) - \exp(sq_4)}{s},$$
$$\frac{\partial h_3}{\partial v_4} = \frac{\partial h_3}{\partial v_5} = \frac{1}{2}(q_4 - v_2)\exp(sq_4),$$
$$\frac{\partial h_4}{\partial v_2} = \frac{\partial h_4}{\partial v_3} = -\frac{1}{2}(q_2 - v_4)\exp(sq_2),$$
$$\frac{\partial h_4}{\partial v_4} = \frac{\exp(sq_2) - \exp(sq_5)}{s},$$
$$\frac{\partial h_5}{\partial v_3} = \frac{\partial h_5}{\partial v_4} = -\frac{1}{2}(q_3 - v_5)\exp(sq_3),$$
$$\frac{\partial h_5}{\partial v_5} = \frac{\exp(sq_3) - \exp(sq_5)}{s}.$$

According to Theorem 1, $0 < v_1(t) < v_2(t) < v_3(t) < v_4(t) < v_5(t) < 1$ for all $t \geq 0$. As $s \neq 0$, it is found that $\frac{\partial h_i}{\partial v_i} < 0$ for all $1 \leq i \leq 5$ and $\frac{\partial h_i}{\partial v_j} \geq 0$, where $i \neq j$. In particular, the matrix is a band matrix which looks like

$$\begin{bmatrix} - & + & + & 0 & 0 \\ 0 & - & + & + & 0 \\ + & + & - & + & + \\ 0 & + & + & - & 0 \\ 0 & 0 & + & + & - \end{bmatrix}.$$

Where '-' denotes the element is negative while '+' denotes a positive element. It can then show that the sum of each row is negative. It is stated as following lemma.

Lemma 5: *If $f(x)$ is loglinear, then $\sum_{j=1}^{5} \frac{\partial h_i}{\partial v_j} < 0$ for all $1 \leq i \leq 5$.*

(Proof) Before proceed to the proof of Lemma 5, we need the following three Lemma.

Lemma A1: *$e^y - 1 - y \geq 0$ for all $y \in (-\infty, +\infty)$. Equality holds if and only if $y = 0$.*

(Proof) Set $f(y) = e^y - 1 - y$, the derivative of $f(y)$ with respect to y is that $\frac{df}{dy} = e^y - 1$. Since $\frac{df}{dy} = 0$ if and only if $y = 0$. As $\frac{df}{dy} > 0$, $f(0)$ is a local minimum. Hence it is global minimum. As a result, $f(y) \geq 0$ for all y and $f(y) = 0$ if and only if $y = 0$. Hence the proof is completed.

□

Lemma A2: *For all $k \neq 0$ and $a \in (0, 1]$,*

$$g_1(a) = \frac{1 - 2e^{ka}}{k} + \frac{a}{e^{ka} - 1} < 0.$$

(Proof) Rewrite g_1, we obtain that

$$g_1(a) = \frac{(1 - 2e^{ka})(e^{ka} - 1) + ka}{k(e^{ka} - 1)}.$$

First we consider the case $k > 0$. Under this case, $k(e^{ka} - 1) > 0$ for all $a \in (0, 1]$. According to Lemma A1,

$$\begin{aligned}(1 - 2e^{ka})(e^{ka} - 1) + ka &= -(e^{ka} - 1)^2 - e^{ka}(e^{ka} - 1) + ka \\ &< -(e^{ka} - 1)^2 - e^{ka}ka + ka \\ &= -(e^{ka} - 1)^2 - (e^{ka} - 1)ka \\ &= -(e^{ka} - 1)(e^{ka} - 1 + ka) \\ &< 0.\end{aligned} \tag{A.23}$$

Hence $g_1(a) < 0$ if $k > 0$. Next we consider $k < 0$. Similar, we get that $k(e^{ka} - 1) > 0$ for all $a \in (0, 1]$. Again, according to Lemma A1,

$$\begin{aligned}(1 - 2e^{ka})(e^{ka} - 1) + ka &< (1 - 2e^{ka})(e^{ka} - 1) + (e^{ka} - 1) \\ &= 2(1 - e^{ka})(e^{ka} - 1) \\ &< 0,\end{aligned} \tag{A.24}$$

for all $a \neq 0$. Therefore $g_1(a) < 0$ for all $a \in (0, 1]$. The proof is completed.

□

Lemma A3: *For all $k \neq 0$ and $0 < a < b < 1$,*

$$g_2(a,b) = \frac{(a-b)e^{k(a+b)}}{e^{kb}-e^{ka}} - \frac{e^{kb}}{k} + \frac{2e^{ka}}{k} < 0.$$

(Proof) Similar to the proof of Lemma A2, we consider two cases, $k > 0$ and $k < 0$. After manipulation on $g_2(a,b)$, we get that

$$\begin{aligned} g_2(a,b) &= \frac{-k(b-a)e^{k(a+b)} + e^{ka}(e^{kb}-e^{ka})}{k(e^{kb}-e^{ka})} - \frac{e^{kb}-e^{ka}}{k} \\ &= \frac{-e^{k(a+b)}}{k(e^{kb}-e^{ka})}\left\{e^{k(a-b)} - 1 + k(b-a)\right\} - \frac{e^{kb}-e^{ka}}{k}. \end{aligned} \tag{A.25}$$

As $b > a > 0$ and from Lemma A1, $g_2(a,b) < 0$ for all $k > 0$. Next, we consider the case that $k < 0$. When $k < 0$, $e^{k(a+b)} > 0$. $k(e^{kb} - e^{ka}) > 0$ and $\frac{e^{kb}-e^{ka}}{k}$ whenever $b > a$. According to Lemma A1, $e^{k(a-b)} - 1 + k(b-a) > 0$. Therefore, $g_2(a,b) < 0$ when $k > 0$. As a result, $g_2(a,b) < 0$ for all $k \neq 0$ and the proof is completed.

□

(Proof of Lemma 5) Adding all the elements within each row, and put the value of $v_1, \ldots, v_5$ derived from (A.22), we get the following equalities.

$$\begin{aligned} \sum_{j=1}^{5} \frac{\partial h_1}{\partial v_j} &= \frac{e^{sq_0} - e^{sq_2}}{s} - (q_2 - v_1)e^{sq_2} \\ &= \frac{1 - 2e^{s_q 2}}{s} + \frac{q_2}{e^{sq_2} - 1} \end{aligned} \tag{A.26}$$

$$\begin{aligned} \sum_{j=1}^{5} \frac{\partial h_2}{\partial v_j} &= \frac{e^{sq_0} - e^{sq_3}}{s} - (q_3 - v_2)e^{sq_3} \\ &= \frac{1 - 2e^{sq_3}}{s} + \frac{q_3}{e^{sq_3} - 1} \end{aligned} \tag{A.27}$$

$$\begin{aligned}\sum_{j=1}^{5}\frac{\partial h_3}{\partial v_j} &= \frac{e^{sq_1}-e^{sq_4}}{s}+(q_1-v_3)e^{sq_1}-(q_4-v_3)e^{sq_4} \\ &= \frac{2}{s}(e^{sq_1}-e^{sq_4}) \qquad \text{(A.28)}\end{aligned}$$

$$\begin{aligned}\sum_{j=1}^{5}\frac{\partial h_4}{\partial v_j} &= \frac{e^{sq_2}-e^{sq_5}}{s}+(q_2-v_4)e^{sq_2} \\ &= \frac{(q_2-q_5)e^{s(q_5+q_2)}}{e^{sq_5}-e^{sq_2}}-\frac{e^{sq_5}}{s}+\frac{2e^{sq_2}}{s} \qquad \text{(A.29)}\end{aligned}$$

$$\begin{aligned}\sum_{j=1}^{5}\frac{\partial h_5}{\partial v_j} &= \frac{e^{sq_3}-e^{sq_5}}{s}+(q_3-v_5)e^{sq_3} \\ &= \frac{(q_3-q_5)e^{s(q_5+q_3)}}{e^{sq_5}-e^{sq_3}}-\frac{e^{sq_5}}{s}+\frac{2e^{sq_3}}{s} \qquad \text{(A.30)}\end{aligned}$$

According to Lemma A2 and A3,

$$\sum_{j=1}^{5}\frac{\partial h_1}{\partial v_j}<0,$$

$$\sum_{j=1}^{5}\frac{\partial h_2}{\partial v_j}<0,$$

$$\sum_{j=1}^{5}\frac{\partial h_4}{\partial v_j}<0,$$

and

$$\sum_{j=1}^{5}\frac{\partial h_5}{\partial v_j}<0.$$

Moreover, $(e^{sq_1}-e^{sq_4})>0$ if $s<0$ and $(e^{sq_1}-e^{sq_4})<0$ if $s>0$. $\sum_{j=1}^{5}\frac{\partial h_3}{\partial v_j}<0$. Hence,

$$\sum_{j=1}^{5}\frac{\partial h_i}{\partial v_j}<0$$

for all $1\le i\le 5$ if $f(x)$ is loglinear and the proof is completed.

The general case of Lemma 5 is stated as following lemma.

Lemma 5': *If $f(x)$ is loglinear, then $\sum_{j=1}^{c}\frac{\partial h_i}{\partial v_j}<0$ for all $1\le i\le c$.*

In addition to the fact that the diagonal elements of the matrix are negative, it is concluded that all the eigenvalues of Jacobian matrix $\frac{\partial h(V)}{\partial V}$ are located in the negative half plane. So, the equilibrium of (A.11) is again asymptotically stable. Hence the proof of Corollary 1 is completed.

Appendix B

Different Senses of neighborhood

Recently, many researchers have tried to explore the idea of neighborhood interaction to other clustering algorithm. As a result, they brought out different definitions of neighborhood and neighborhood interacting functions (NIF) other than Kohonen's original definition. As the model discussed in this thesis applies the concept of neighborhood interaction, it is necessary to clarify what sense of neighborhood is being applied. In general, the senses of neighborhood can be divided into two classes: static and dynamic.

B.1 Static neighborhood: Kohonen's sense

It is the simplest sense of neighborhood which is defined by T.Kohonen for his SOM model. This definition can be stated as follows. For simplicity, only 1D Map will be considered but it does not loss the generality.

Definition 1 (Kohonen [21]) *The neighborhood interacting set (NIS) is defined as* $N_I = \{I-1, I, I+1\}$ *for* I *is not at the boundary. While* $I = 1$, $N_I = \{1, 2\}$. *While* $I = N$, $N_I = \{N-1, N\}$. *The NIF,* $\beta_i(k)$ *is defined as follows:*

$$\beta_i(k) = \begin{cases} 1 & \textit{if } |i-k| \leq 1 \\ 0 & \textit{otherwise.} \end{cases} \tag{B.1}$$

Apart from fixing the function as a step function, Kohonen also defined the NIF in Gaussian shape:

Definition 2 (Kohonen [21]) *The NIS is defined as Definition 1 but the NIF, $\beta_i(k)$ is defined by $\beta_i(k) = \exp(-(i-k)^2)$.*

In both of the above definitions, their common feature is that their definitions on the neighborhood interacting set are independent of the Euclidean distance between the input vector x and the weight vector.

B.2 Dynamic neighborhood

Using above definitions, some researchers find that that static neighborhood is not flexible enough to form a good data manifold for some special type of input data set such as sphere data. Therefore some researchers attempted to define neighborhood in a dynamic sense [3] [29] [32]. In these cases, the neighborhood set of a SOM are updated after a number of training cycles.

B.2.1 Mou-Yeung Definition

In [32], the neighborhood set is update after every predetermined number of iterations. The neighborhood relationship is constructed using the following definition:

Definition 3 (Mou and Yeung [32]) *Node i and node j are neighborhood if and only if $\|v_i - v_j\|^2 < \|v_i - v_k\|^2 + \|v_k - v_j\|^2$, for all $k \neq i, j$. And the NIF $\beta_i(k)$ is defined by*

$$\beta_i(k) = \begin{cases} 1 & i \text{ and } k \text{ are neighbor} \\ 0 & \text{otherwise.} \end{cases} \tag{B.2}$$

Instead of defining the neighborhood in term of the distance amongst v_is,

B.2.2 Martinetz et al. Definition

Bezdek et al. [3] and Martinetz et al. [29] defined the neighborhood set in term of difference between the location of input x and the location of weight vectors v_is. The idea is that. Once a data vector x is presented, the winner is the one closest to x, say node I. Then the first neighborhood of node I is the second closest to x. Then the second neighborhood of node I is the third closest and etc. Under such circumstance, each neighbor of I will be marked with a value called neighborhood ranking value. The definition of neighborhood ranking is defined as following:

Definition 4 (Neural Gas [29]) *Consider that v_{π_1} be the vector being closest to x, then the neighborhood set is defined as $\{v_{\pi_1}, v_{\pi_2}, \ldots, v_{\pi_c}\}$, where v_{π_2} is the second closest to x. The neighborhood ranking value is defined as a function of π_i, $\beta_i(\pi_i(x, V)) = e^{-\pi_i/\lambda}$.*

As π_i is a function of x and V, β_i is also a function of x and V. Besides, $\beta_i < \beta_j$ if $i > j$. If $\|x - v_i\| \to \infty$, β_i is still non-zero.

B.2.3 Tsao-Bezdek-Pal Definition

Similarly, Tsao et.al. [44] defined the neighborhood sense in the same manner:

Definition 5 (Tsao-Bezdek-Pal [44]) *Consider that v_{π_1} be the vector being closest to x, then the neighborhood set is defined as $\{v_{\pi_1}, v_{\pi_2}, \ldots, v_{\pi_c}\}$. The NIF is defined as a function of π_i,*

$$\beta_i(\pi_k) = \begin{cases} 1 & \textit{If } k = 1. \\ \xi & \textit{otherwise}, \end{cases} \tag{B.3}$$

where ξ is a number smaller than 1.

In order to visualize the similarity and difference among all these definitions of neighborhood, let us have a simple example.

B.3 Example

Suppose that the a 1D map consists of six nodes, Figure B.1. The input data x is indicated by the black solid circle while v_is are indicated by hollow circles. In this example, v_4 is obviously the closest to x. hence, no matter using which definition, node 4 is the winner. However, based on different senses of neighborhood, the neighborhood set of node 4 are different.

In the sense of Kohonen, $N_4 = \{3, 4, 5\}$. In the sense of Mou-Yeung, $N_4 = \{2, 4, 5\}$. In the sense of Neural Gas and Tsao-Bezdek-Pal, $N_4 = \{4, 2, 5, 6, 1, 3\}$, where the location reflects the ranking. But, due to their different in the definitions of NIF, the values of $\beta_i(.)$ are different. The values are listed on the table.

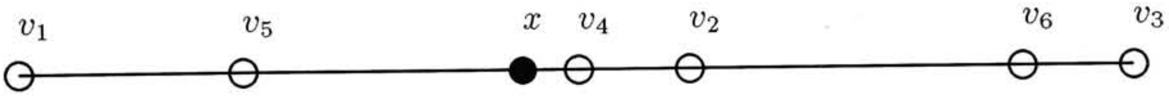


Figure B.1: An example of 1D Map.

Definition	I	β_1	β_2	β_3	β_4	β_5	β_6
Def. 1	4	0	0	1	1	1	0
Def. 2	4	e^{-9}	e^{-4}	e^{-1}	1	e^{-1}	e^{-4}
Def. 3	4	0	1	0	1	1	0
Def. 4	4	e^{-4}	e^{-1}	e^{-5}	1	e^{-2}	e^{-3}
Def. 5	4	ξ	ξ	ξ	1	ξ	ξ

Table B.1: Definition of NIF $\beta_i(x)$ when x is presented. ξ is a small number.

To contrast the different between different definitions, the neighborhood interacting values are shown in Figure(B.2).

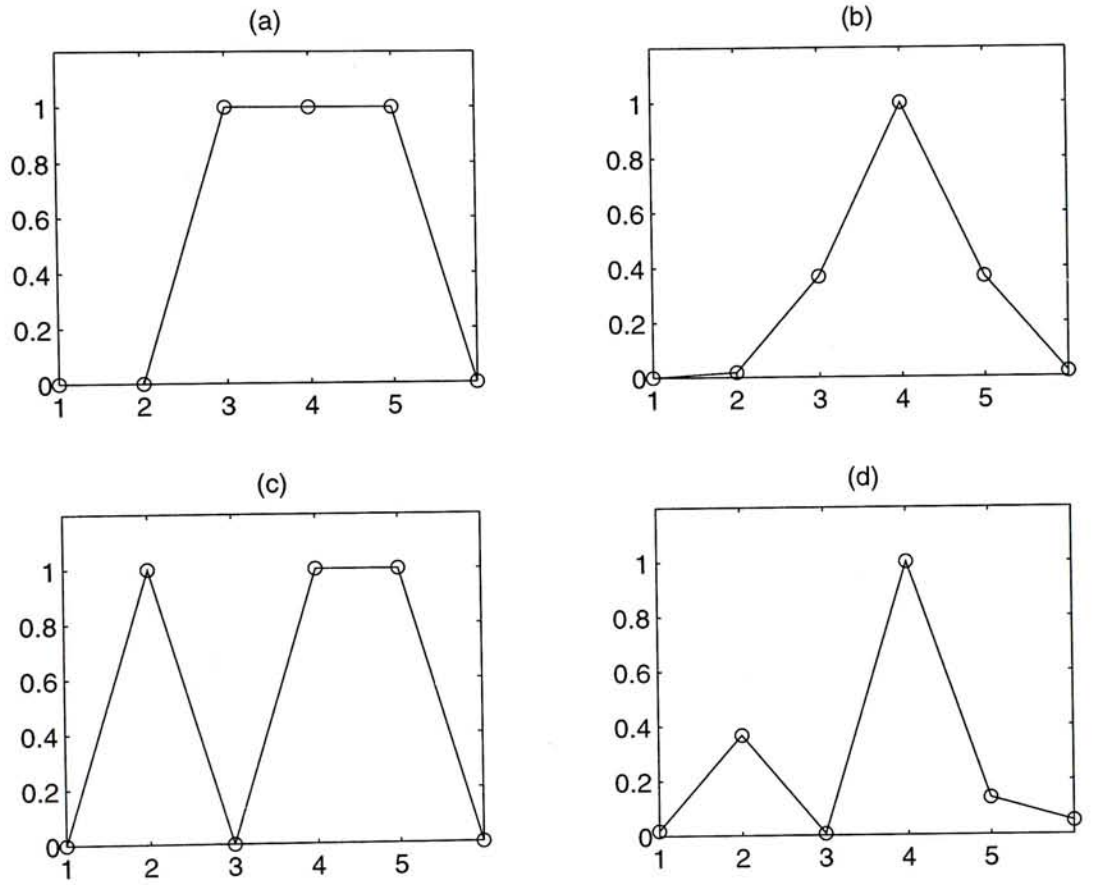


Figure B.2: The neighborhood interacting functions defined by (a) uniform function, (b) Gaussian function, (c) Mou-Yeung definition and (d) Neural gas definition.

B.4 Discussion

Actually, there are many ways to combine the concept of neighborhood interaction to the algorithms of Maximum Likelihood Competitive Learning and Fuzzy Competitive Learning. Figure B.3 indicates some possible extensions. However, not all of them are considered in this thesis in order to implement the softing version of SOM.

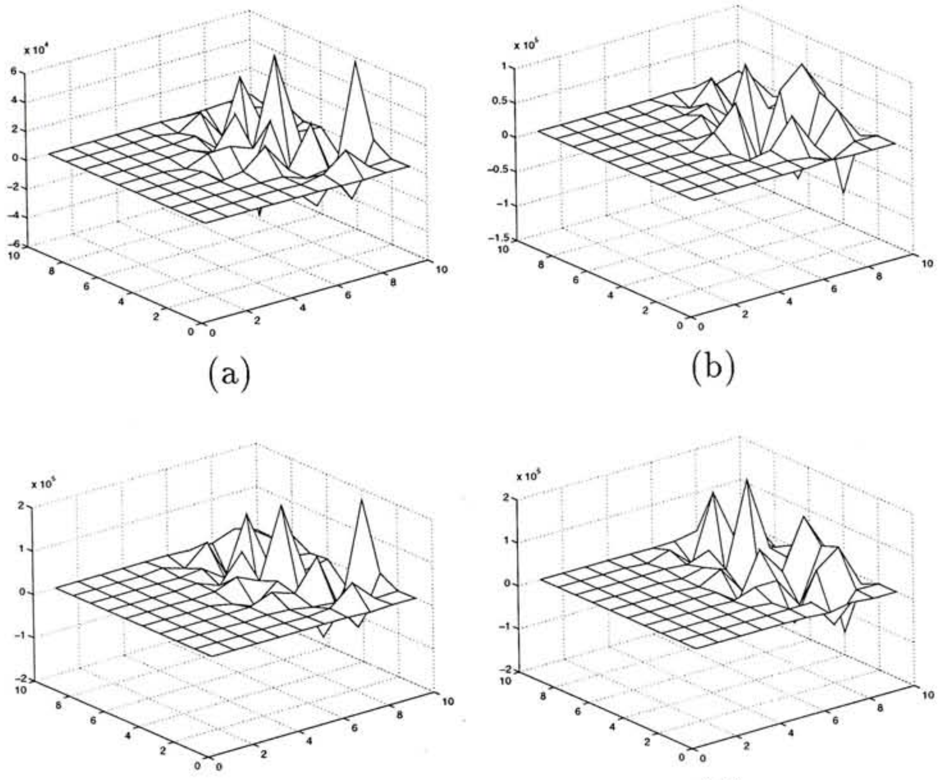
	Def.1	Def.2	Def.3	Def.4	$\{I\}$
Hard	SOM	SOM	Mou-Yeung	Neural Gas	CL
Bezdek	$SSOM$	$SSOM$			FCL
RBF	$SSOM$	$SSOM$			MLCL

Figure B.3: The suggested combinations.

The reasons can be explained as following. The senses of Mou-Yeung and Neural Gas are not considered since both senses of neighborhood are defined in a dynamic way so that in some cases the global information of neighborhood may be lost. For example, under Mou-Yeung's definition, the map has to be constructed time after time. There will be a serious problem when the data set consists of two isolated clusters which are separated far apart: The map will be separated into two. In this case, it cannot identify the neighborhood relationship between clusters. Under Neural Gas's definition, the situation is even worst since the neighborhood relationship is totally lost. Besides, the compuatational cost on the construction of neighborhood sets for each of the neuron is also very high. So, in the design of , we define the sense of neighborhood as one of the Kohonen's definition, Def.1.

Appendix C

Supplementary to Chapter 4



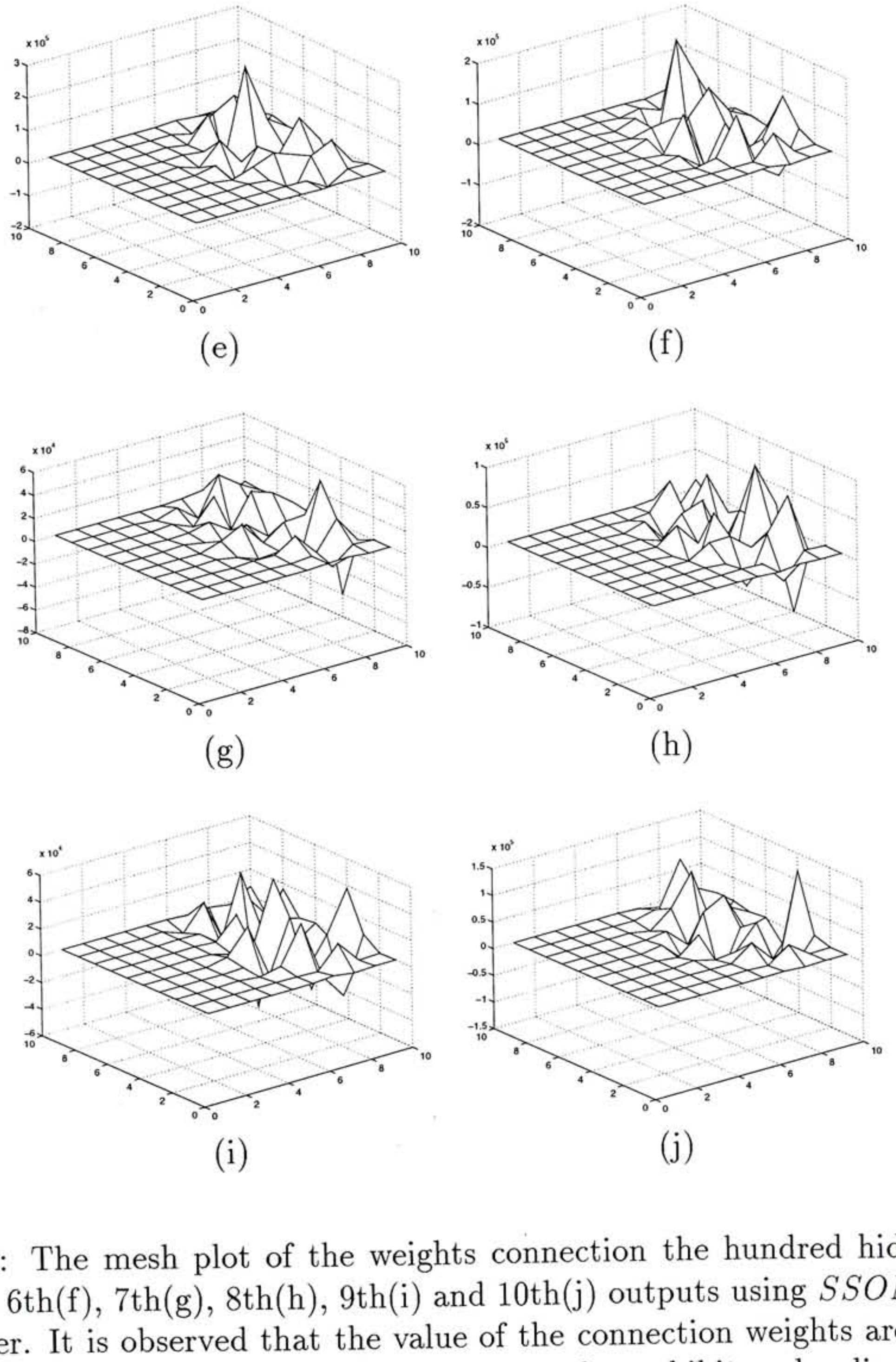
(a)

(b)

(c)

(d)

Figure C.1: The mesh plot of the weights connection the hundred hidden unit to the 1st(a), 2nd(b), 3rd(c) and 4th(d) outputs using $SSOM_1$ as input-hidden layer. It is observed that the value of the connection weights are very large, in an order of magnitude of five. Besides, the weights exhibit no localization effect.



(e)

(f)

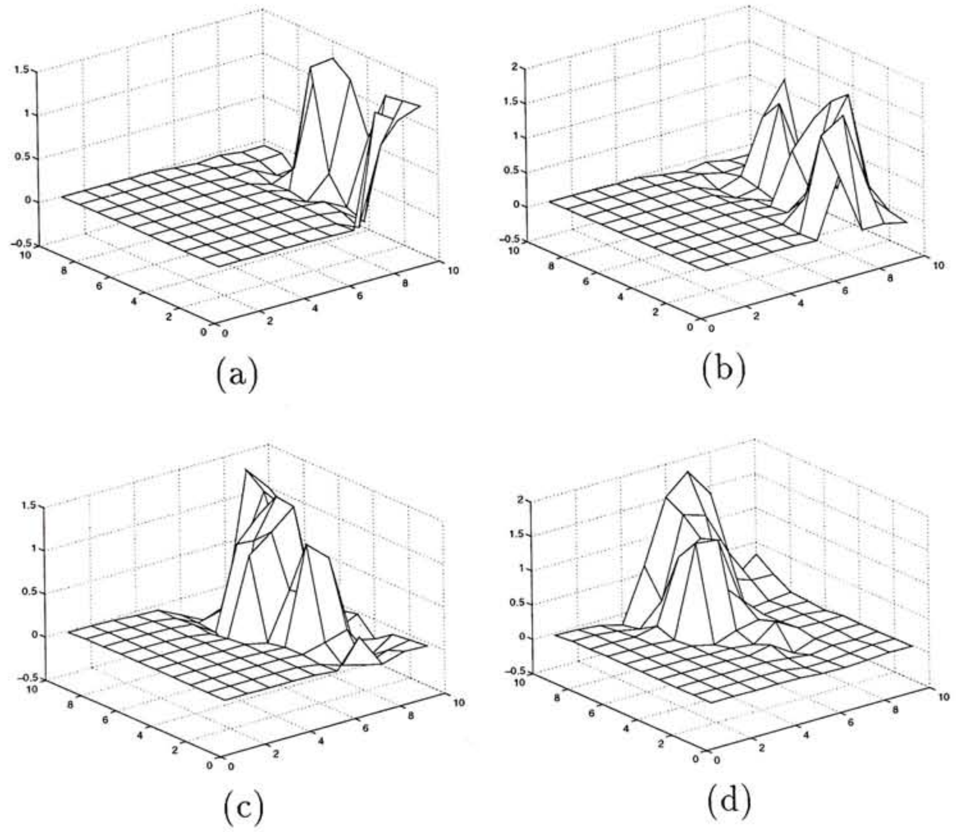
(g)

(h)

(i)

(j)

Figure C.2: The mesh plot of the weights connection the hundred hidden unit to the 5th(e), 6th(f), 7th(g), 8th(h), 9th(i) and 10th(j) outputs using $SSOM_1$ as input-hidden layer. It is observed that the value of the connection weights are very large, in an order of magnitude of five. Besides, the weights exhibit no localization effect.



(a) (b) (c) (d)

Figure C.3: The mesh plot of the weights connection the hundred hidden unit to the 1st(a), 2nd(b), 3rd(c) and 4th(d) outputs using $SSOM_2$ as input-hidden layer. It is observed that the weights with large value are localized.

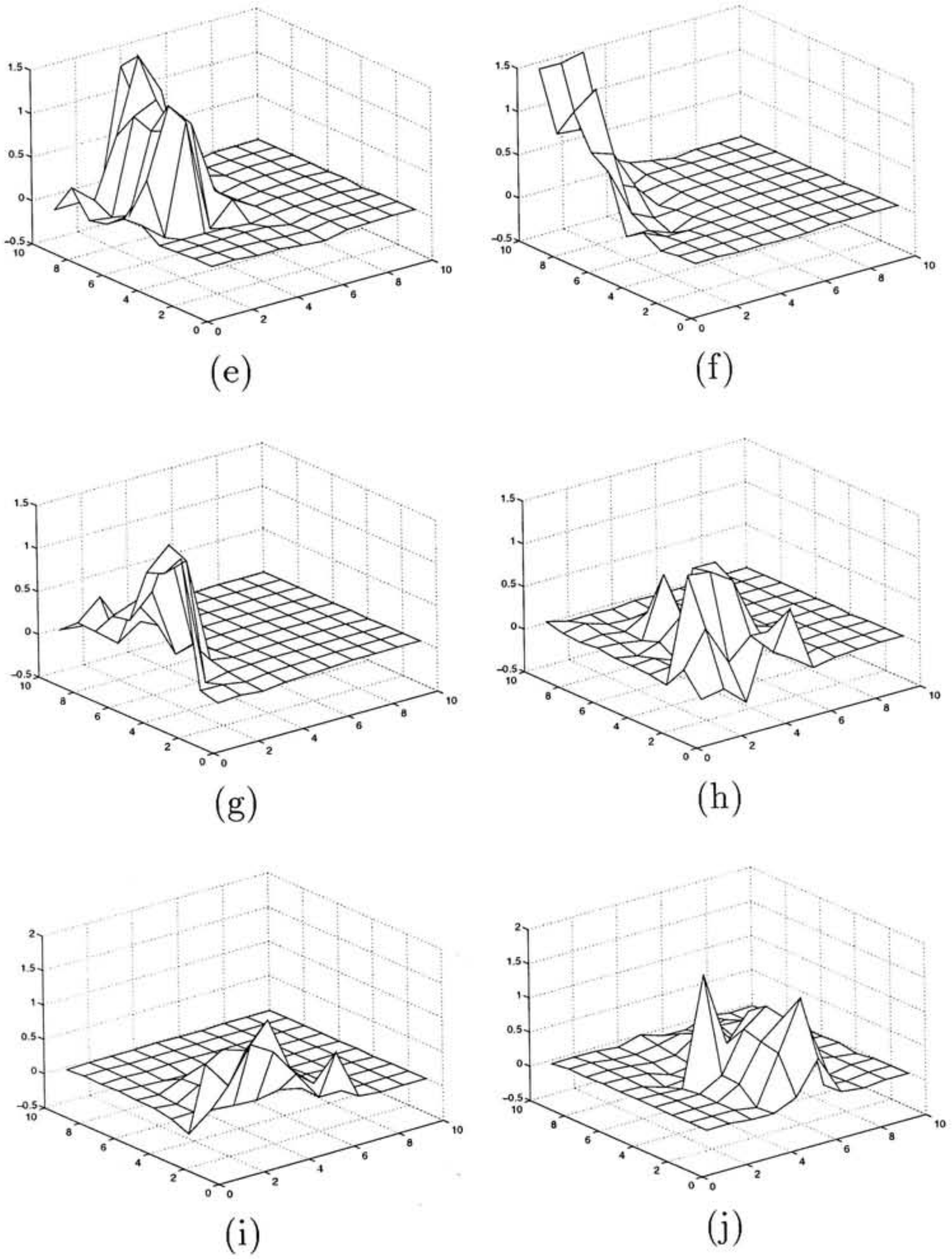
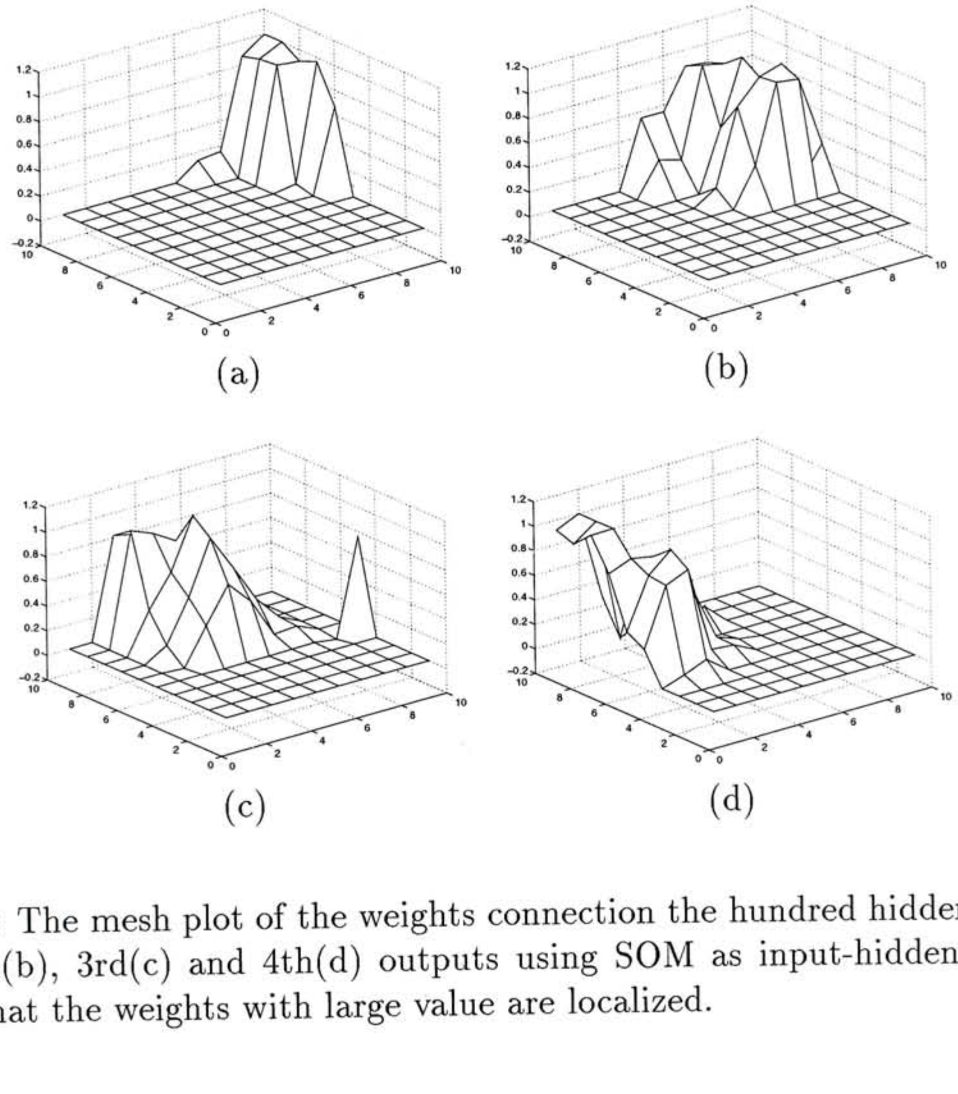

(e) (f) (g) (h) (i) (j)

Figure C.4: The mesh plot of the weights connection the hundred hidden unit to the 5th(e), 6th(f), 7th(g), 8th(h), 9th(i) and 10th(j) outputs using $SSOM_2$ as input-hidden layer. It is observed that the weights with large value are localized.



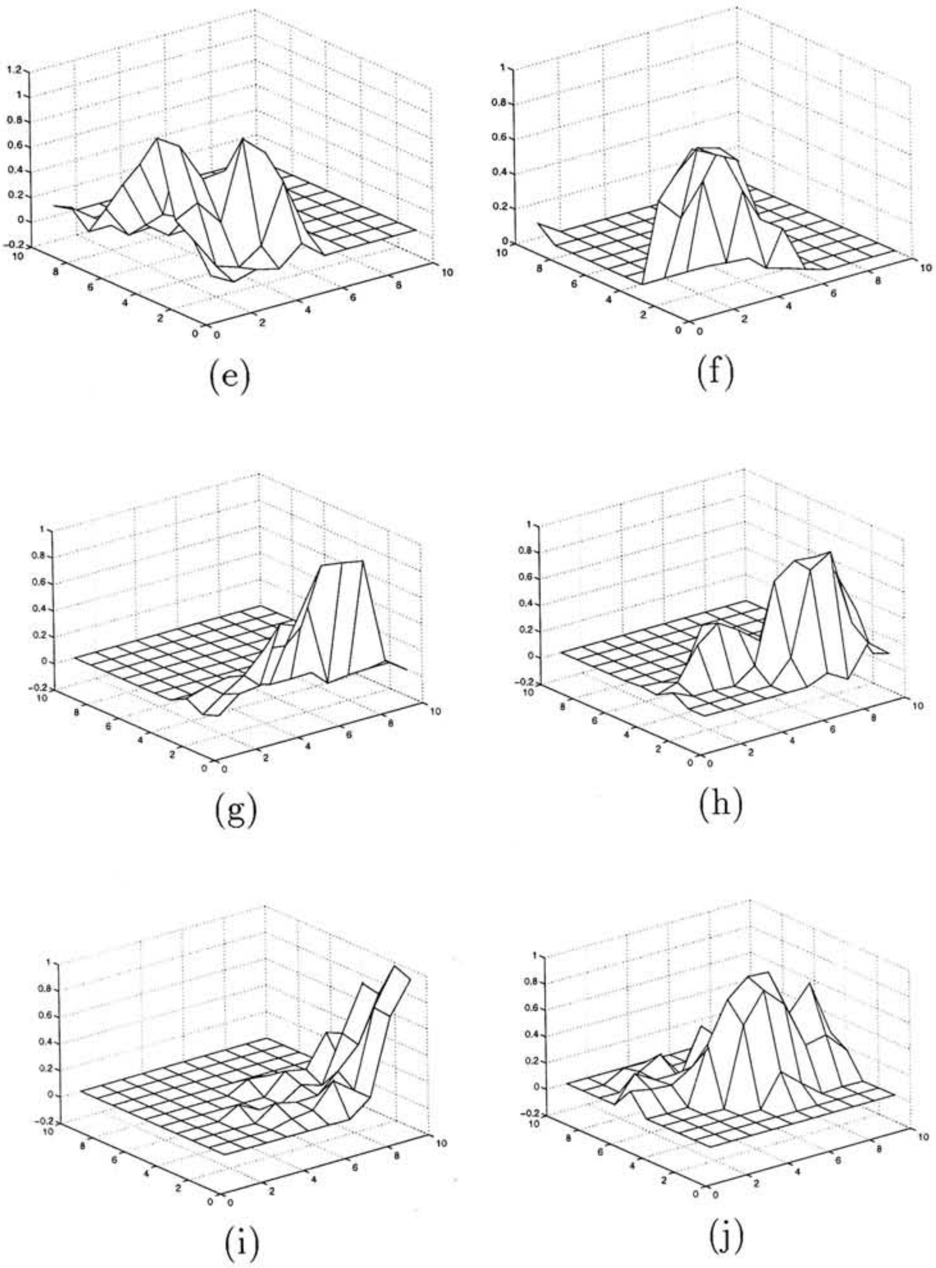
(a)

(b)

(c)

(d)

Figure C.5: The mesh plot of the weights connection the hundred hidden unit to the 1st(a), 2nd(b), 3rd(c) and 4th(d) outputs using SOM as input-hidden layer. It is observed that the weights with large value are localized.



(e) (f)

(g) (h)

(i) (j)

Figure C.6: The mesh plot of the weights connection the hundred hidden unit to the 5th(e), 6th(f), 7th(g), 8th(h), 9th(i) and 10th(j) outputs using SOM as input-hidden layer. It is observed that the weights with large value are localized.

Appendix D

Quadrature Amplitude Modulation

Quadrature amplitude modulation (QAM) is one of the technique for multisymbol transmission. In one of the simulated experiment, this modulation technique is combined with the clustering algorithm proposed – the Soft Self Organizing Map – to demonstrate the gain due to topological order. Here we give only a very brief introduction to QAM. For further detail, please refers to [40].

D.1 Amplitude Modulation

QAM is essentially an extension of amplitude modulation (AM) which is used in radio broadcasting. The principle can be described as following. Suppose that the broadcast station would like to transmit a speech signal, say $s(t)$, to the audience. Based on amplitude modulation technique, the broadcast signal will be formed by multiplying the speech signal with a carrier wave, say $c(t) = sin(w_c t)$, in the way as that $s(t)c(t)$. In the receiver side, this speech signal is reconstructed by using the relation $s(t) = \int_T s(t)c(t)c(t)dt$. Figure(D.1) show a simple example. The speech signal is low frequency sinusoidal function. It is modulated through a high frequency carrier wave. The resultant broadcast signal is shown in Figure(D.1c).

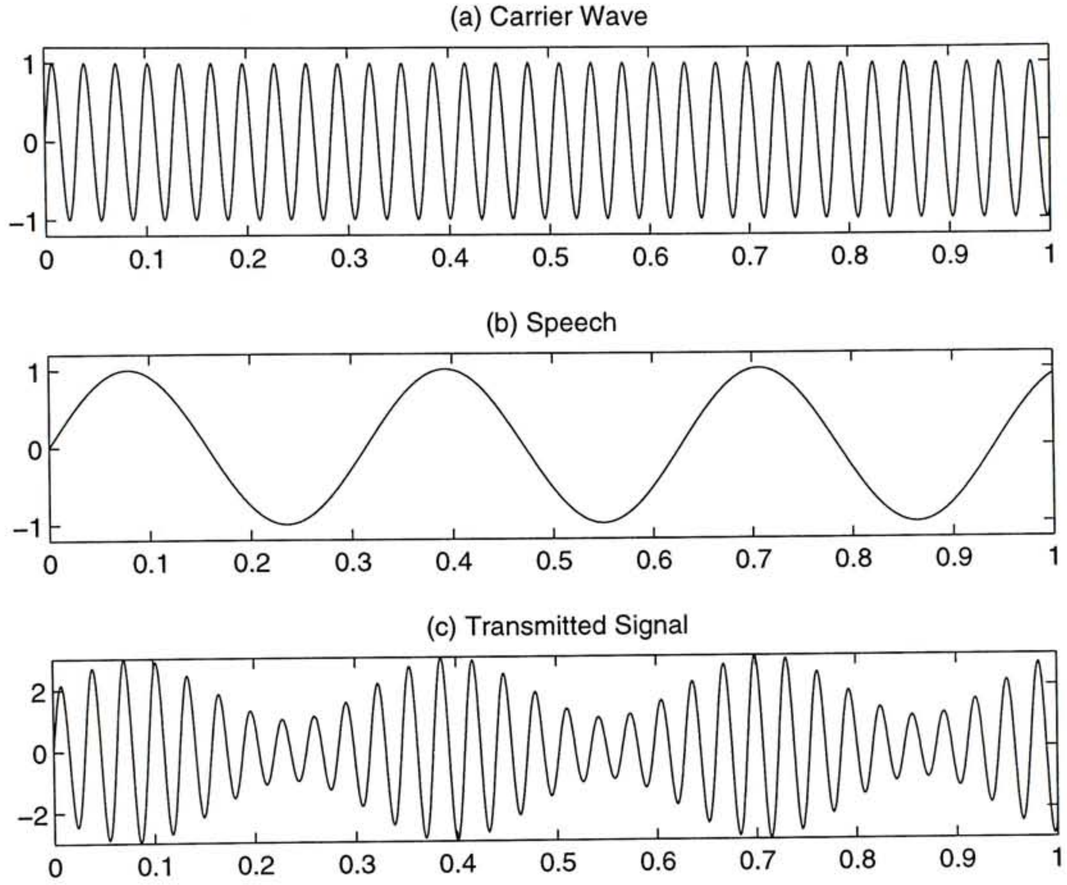


Figure D.1: A simple example showing the idea of amplitude modulation: (a) the carrier wave $c(t) = sin(200t)$, (b) the speech signal $s(t) = sin(20t)$ and (c) the broadcast signal $(2 + sin(20t))sin(200t)$.

D.2 QAM

QAM extends the idea of amplitude modulation and provides a simple modulation method for the transmission of digital signal. Imagine that a sequence of binary signal, say 00101101, is going to be sent out. We can treat this sequence in the same way as speech. Then using the amplitude modulation technique to form the broadcast signal. Suppose the carrier wave is also $c(t) = sin(200t)$, the modulation steps can be described as following:

- $\{0\,0\,1\,0\,1\,1\,0\,1\} \rightarrow s(t) = \{-1\ -1\,1\ -1\,1\,1\ -1\,1\}$.,

- the amplitude of the broadcast signal is $\{1\ 1\ 3\ 1\ 3\ 3\ 1\ 3\}$.

Obviously, it is a Bi-level amplitude modulation since each digit can represent two possible cases only. Figure(D.2) shows the waveforms of the corresponding signals during modulation.

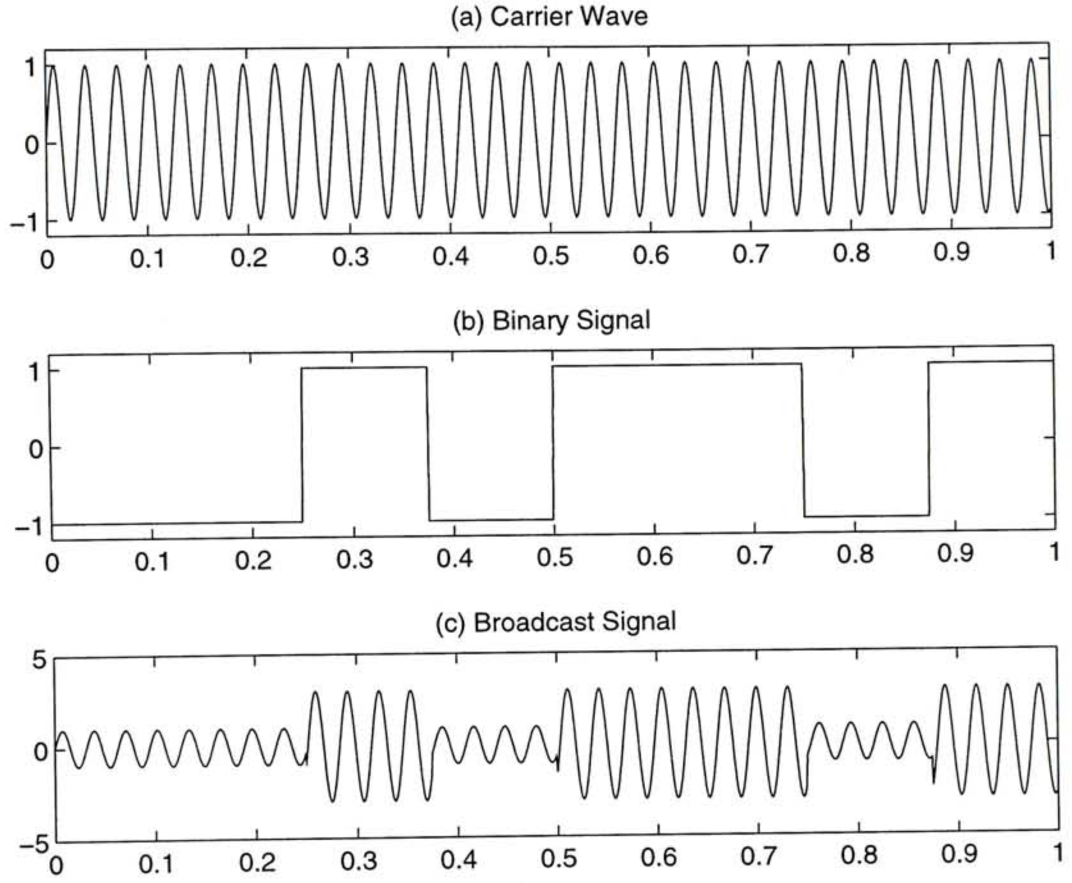


Figure D.2: Transmission of sequence of binary signal using amplitude modulation: (a) the carrier wave $c(t) = sin(200t)$, (b) the binary signal and (c) the broadcast signal.

Instead of modulating the signal bit by bit, it is possible to modulate the signal two-bit by two-bit. In the above example, the transmission sequence becomes $\{00\ 10\ 11\ 01\}$. As the possible event generated by two-bits are four, 00, 01, 10 and 11 respectively, two alternative schemes can be applied. The first one is Four-level scheme. According to the following encoding scheme:

- $00 \rightarrow -1.5$,

- $01 \rightarrow -0.5$,

- 10 → +0.5,
- 11 → +1.5,

we obtain the waveforms as Figure(D.3).

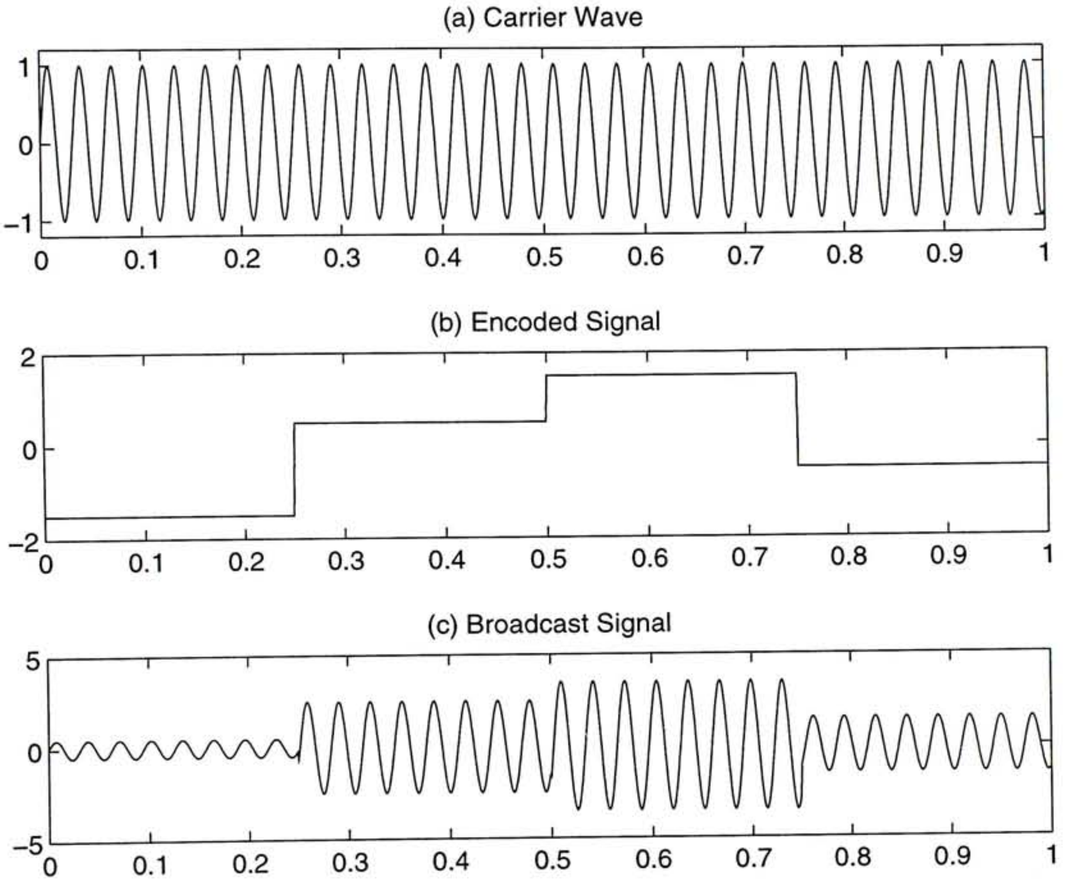


Figure D.3: Transmission of sequence of binary signal using amplitude modulation: (a) the carrier wave $c(t) = sin(200t)$, (b) the encoded signal and (c) the broadcast signal.

An alternative is to modulate the signal based on QAM. Figure(D.4) shows the structure of a QAM transmitter.

Instead of using single sinusoidal wave as the carrier wave, the carrier wave of QAM is composed of two orthogonal sinusoidal waves, $cosw_ct$ and $sinw_ct$. The broadcast signal is then the superposition of two modulated sinusoidal waves, $a_icosw_ct+b_isinw_ct$. Similar to 4-level method, we assign each of the four possible signal combinations by an encoding scheme:

- 00 → $(a_1 = -1, b_1 = -1)$,

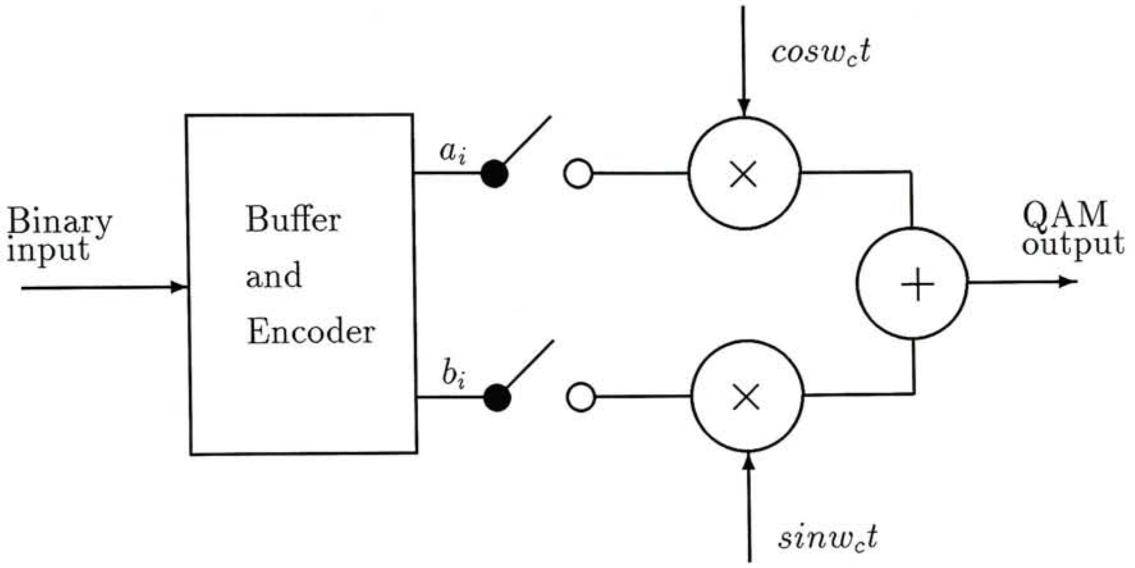


Figure D.4: Simple diagram of a QAM transmitter.

- 01 → $(a_1 = -1, b_1 = +1)$,
- 10 → $(a_1 = +1, b_1 = -1)$,
- 11 → $(a_1 = +1, b_1 = +1)$.

Figure(D.5) shows the corresponding waveforms.

It is useful to represent encoded signal in a two-dimensional diagram by locating the various points (a_i, b_i). The signal points are said to be represent a signal constellation. Figure(D.6) shows the signal constellations of the above QAM.

Suppose that a_i and b_i can be assigned to be either one of $\{-1.5\ -0.5\ +0.5\ +1.5\}$, we can design a 16-symbol QAM constellation, Figure(D.7).

As the locations of the waveforms are in regular mesh, it is possible to assign these waveforms to a organizing map which is defined in the same mesh structure. Figure(D.8) shows the idea of this assignment. This assignment method is the way that we apply to vowel data transmission.

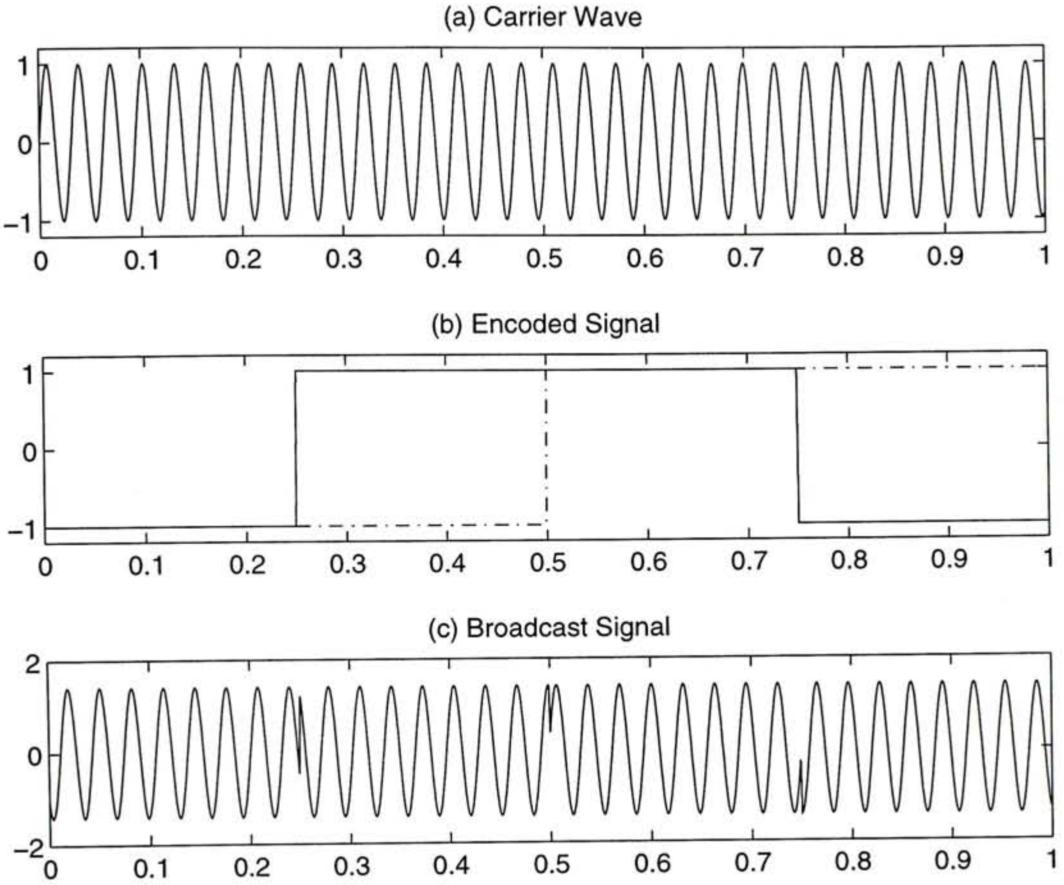


Figure D.5: Transmission of sequence of binary signal using QAM: (a) the carrier wave $c(t) = sin(200t)$, (b) the encoded signal (solid line is the signal of $a(t)$ while the dash-dot line is the signal of $b(t)$ and (c) the broadcast signal.).

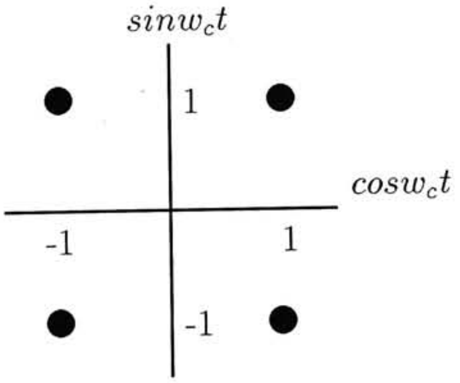


Figure D.6: The signal constellation of the above QAM scheme.

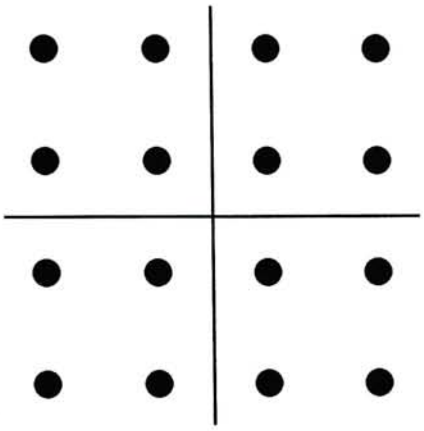


Figure D.7: The signal constellation of the 16-symbol QAM scheme.

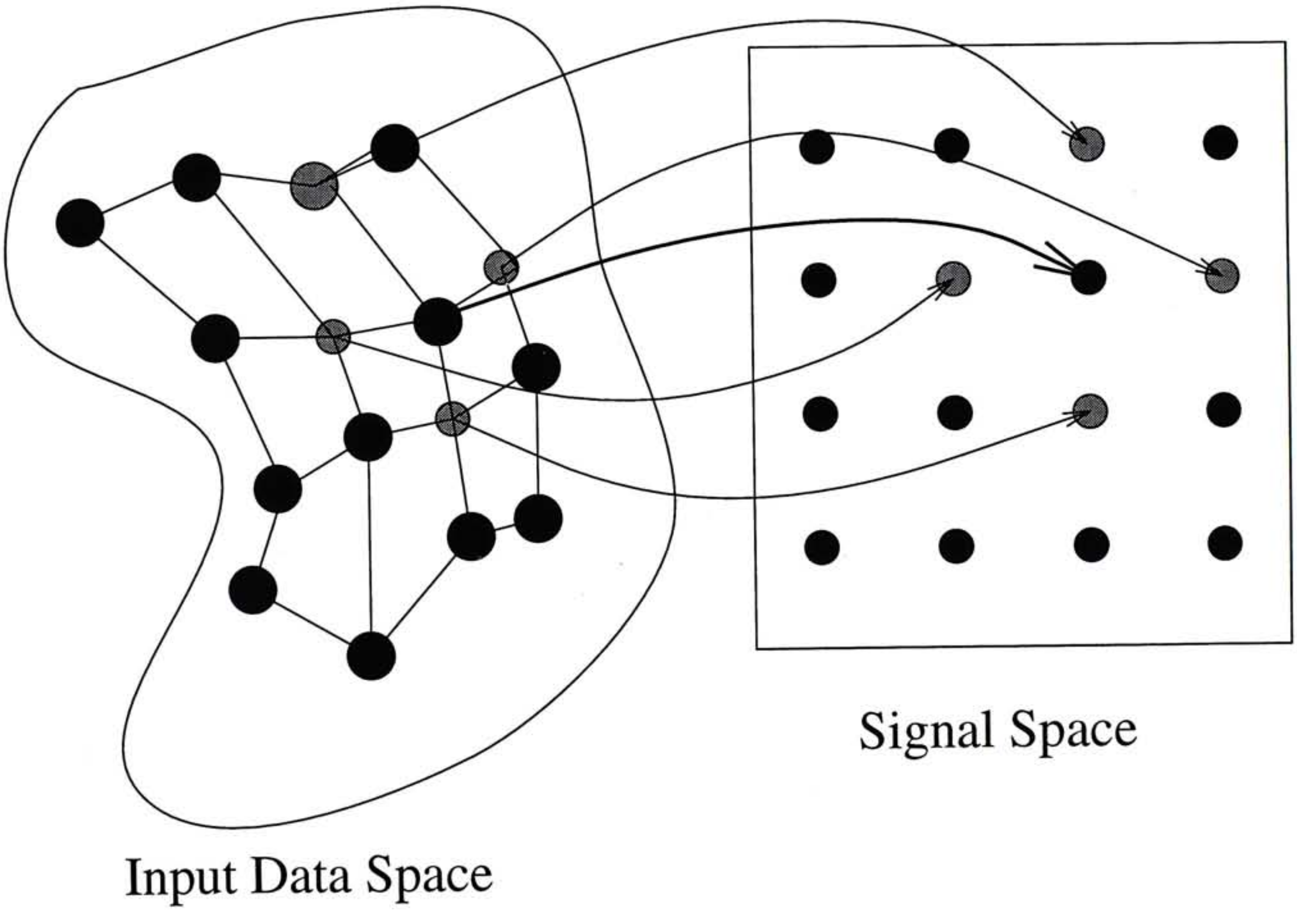


Figure D.8: The idea of codevector waveform assignment.

Bibliography

[1] B.Angeniol et.al., Self-Organizing Feature Maps and the Traveling Salesman Problem, *Neural Networks*, Vol.1, 289-293, 1988.

[2] J.Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms.* NY:Plenum,1981.

[3] J.Bezdek, E.Tsao and N.Pal, Fuzzy Kohonen clustering networks, , *Proceeding of First IEEE Conf. on Fuzzy Systems*, 1035-1043, 1992.

[4] C.Bouton and G.Pages, Self-organizing and a.s. convergence of one-dimensional Kohonen algorithm with non-uniform distributed stimuli, *Stochastic Processes and their Application*, VOL.47, 249-274, 1993.

[5] S.Chiu, Fuzzy Model Identification Based on Cluster Estimation. *Journal of Intelligence and Fuzzy Systems*, Vol.2, 267-278, 1994.

[6] F.Chung and T.Lee, Fuzzy Learning Vector Quantization. *Proceedings of 1993 IJCNN'Nagoya*, 2739-2742, 1993.

[7] F.Chung and T.Lee, Fuzzy Competitive Learning. *Neural Networks*, Vol.7, No.3. 539-551, 1994.

[8] M.Cottrell, J.C.Fort and G.Pages, Two or three things that we know about the Kohonen algorithm, *Technical report, Samos/Universite'e Paris*, 1994.

[9] J.L.Doob, *Stochastic Processes*, John Wiley, N.Y. 1953.

[10] R.Durbin, R.Szeliski and A.Yuille, An Analysis of the Elastic Net Approach to Traveling Salesman Problem, *Neural Computation*, Vol.1, 348-358, 1989.

[11] E.Erwin, K.Obermayer and K.Schulten, "Self-organizing maps: ordering, convergence properties and energy functions", *Biological Cybernetics*, Vol.67, 47-55, 1992.

[12] J.L.Goldberg, *Matrix Theory with Applications.* McGraw-Hill,Inc. 1992.

[13] S.Grossberg, Competitive Learning: from interactivation to adaptive resonance. *Cognitive Science*, VOL.11, 23-63, 1987.

[14] R.Hecht-Neilsen, Applications of Counterpropagation Networks, *Network Networks*, Vol.1, 131-139, 1988.

[15] R.Hecht-Neilsen, Neurocomputing. Addison Wesley. 1990.

[16] C.C.Jou, Fuzzy CounterPropagation Networks, Presented in the *Workshop on the Future Directions of Fuzzy Theory and Systems*, organized by faculty of engineering, CUHK, 27 Oct, 1993.

[17] E.Kandel, J.Schwartz and T.Jessell, *Principles of Neural Science.* 3rd edition. Prentice-Hall. 1991.

[18] H.Khalil, *Nonlinear System*, MacMillan, 1992

[19] T. Kohonen, Self-Organized Formation of Topologically Correct Feature Maps. *Biological Cybernetics*, 43:59-69, 1982.

[20] T.Kohonen, The "neural" phonetic typewriter. *Computer*, Vol.21, 11-22, 1988.

[21] T. Kohonen, *Self-Organization and Associative Memory.* Springer-Verlag, 3rd Ed., 1989. (Chap 5)

[22] T.Kohonen, Self-Organizing Maps: Optimization Approaches , in *Artificial Neural Networks* edited by T.Kohonen et.al., Elsevier Science Publishers B.V. (North-Holland), 1991.

[23] H.J.Kushner and D.S.Clark, *Stochastic Approximation for Constrained and Unconstrained Systems*, Springer Verlag, 1978.

[24] C.S.Leung, Trellis Coded Kohonen Map, in *Proceedings of ICONIP'94 Seoul*, 1994.

[25] C.S.Leung, Kohonen Map: Combined Vector Quantization and Modulation, in *Proceedings of ICONIP'94 Seoul*, 1994.

[26] Y.Linde, A.Buzo and R.Gray, An Algorithm for Vector Quantizer Design, *IEEE Transactions on Communications*, Vol.COM-28. NO.1, 84-95, 1980.

[27] R.Linsker, How to Generate Ordered Maps by Maximizing the Mutual Infornation between Input and Output Signals, *Neural Computation*, 1, 402-411, 1989.

[28] Z.-P. Lo, Y.Yu and B.Bavarian, "Analysis of the convergence properties of Topology Preserving Neural Networks," *IEEE Transactions on Neural Network*, Vol. 4, No. 2, pp.207 – 220, 1993.

[29] T.Martinetz, S.Berkovich and K.Schulten, Neural-Gas Network for Vector Quantization and its Application to Time-Series Prediction, *IEEE Transactions on Neural Networks*, Vol.4(4), 558-569, 1993.

[30] S.Mitra and S.K.Pal, Self-Organizing Neural Network As A Fuzzy Classifier, *IEEE Transactions on Systems, Man and Cybernetics*, Vol.24, No.3, 385-399, 1994

[31] J.Moody and C.Darken, Fast Learning in Networks of Locally-Tuned Processing Units. *Neural Computation*, 1,281-294, 1989.

[32] K.Mou and D.Yeung, Grbriel Networks: Self-Organizing Neural Networks for Adaptive Vector Quantization, *Proceeding of ISSIPNN'94, 13-16 April 1994, Hong Kong*, 658-661, 1994.

[33] S.Nowlan, Maximum Likelihood Competitive Learning, *Advance in Neural Information Processing Systems*, 574-582, 1990.

[34] S.K.Pal and S.Mitra, Multilayer Perceptron, Fuzzy Sets, and Classification, *IEEE Transaction on Neural Network*, VOL.3(5), 683-697, 1992.

[35] D.T.Pham and E.J.Bayro-Corrochano, Self-Organizing Neural-Network-Based Pattern Clustering Method with Fuzzy Outputs, *Pattern Recognition*, Vol.27, No.8, 1103-1110, 1994.

[36] H.Ritter, K.Schulten, "Convergence Properties of Kohonen's Topology Conserving Map: Fluctuations, Stability and Dimension Selection," *Biological Cybernetics*, Vol.60, 59-71, 1988.

[37] H.Ritter and T.Kohonen. Self-Organizing semantic maps. *Biological Cybernetics.* **61**, 241-254. 1989.

[38] H.Ritter, T.Martinetz and K.Schulten, *Neural Computation and Self-Organizing Maps: An Introduction.* Addison Wesley, 1992.

[39] F.Rosenblatt, The Perceptron: The Probabilistic Model for Information Storage and Organization in the Brain. *Psychological Review*, 65, 386-408, 1958.

[40] M.Schwartz, *Information Transmission, Modulation, and Noise.* 4th edition. McGraw Hill. 1990.

[41] M.Sugeno and T.Yasukawa, A Fuzzy logic based approach to qualitative modeling, *IEEE Transaction on Fuzzy Systems*, Vol.1(1), 1-24, 1993.

[42] V.V.Tolat, "An analysis of Kohonen's self organizing maps using a system of energy functions", *Biological Cybernetics*, Vol.64, 155-164, 1990.

[43] A.Trushkin, Sufficient Conditions for Uniqueness of a Locally Optimal Quantizer for a Class of Convex Error Weighting Function, *IEEE Transactions on Information Theory*, VOL.IT-28, NO.2, MARCH, 1982.

[44] E.Tsao, J.Bezdek and N.Pal, Fuzzy Kohonen Clustering Networks, *Pattern Recognition*, Vol.27(5), 757-764, 1994.

[45] G.A. van Velzen, Topological maps on Ising spin networks, *Physica A*, 185, 439-443. 1992.

[46] P.Vuorimaa, Use of the Fuzzy Self-Organizing Map in Pattern Recognition, *Proceeding of FUZZ-IEEE'94*, 798-801, 1994.

[47] L.Wang, J.Mendel, Fuzzy Basis Functions, Universal Approximation, and Orthogonal Least-Squares Learning, *IEEE Transaction on Neural Network*, VOL.3(5), 807-814, 1992.

[48] D. Willshaw and C. von der Malsburg, "How patterned neural connections can be set up by self-organization," *Proceedings of Royal Society of Landon.* B-194, 431-445, 1976.

[49] D. Willshaw and C. von der Malsburg, "A marker induction for the establishment of order neural mapping: Its application to the retino-tectal problem," *Philosophical Transactions of Royal Society of London*, 287, B1021, 203-243. 1979.

[50] E.Yair, K.Zeger and A.Gersho, Competitive Learning and Soft Competition for Vector Quantizer Design. *IEEE Transactions on Signal Processing*, Vol.40, No.2, 294-309, 1992.

[51] H.Yang and T.S.Dillon, "Convergence and Self-Organizing Neural Algorithms," *Neural Networks*, Vol. 5, pp.485 – 493, 1992.