

โครงข่ายประสาทเทียมแบบปรับตัวเองภายในแบบมีผู้สอน  
โดยใช้ฟังก์ชันความเป็นสมาชิกของเซตของความคลุมเครือ

SUPERVISED SELF ORGANIZING MAP  
USING FUZZY MEMBERSHIP FUNCTION

จิราพร เกียรติวุฒิมอร์  
JIRAPORN KIATWUTHAMORN

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของเอกสารที่มหาวิทยาลัยสุรนารีปริณิญาวิทยาศาสตรมหาบัณฑิต

สาขาวิชาเทคโนโลยีสารสนเทศ

บัณฑิตวิทยาลัย

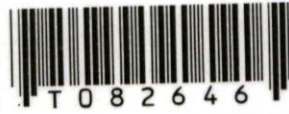
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ. 2551

KMITL-2008-IT-M-001-277

โครงข่ายประสาทเทียมแบบปรับตัวเองภายในแบบมีผู้สอน  
โดยใช้ฟังก์ชันความเป็นสมาชิกของเซตของความคลุมเครือ

SUPERVISED SELF ORGANIZING MAP  
USING FUZZY MEMBERSHIP FUNCTION



จิราพร เกียรติวุฒิมอร์

JIRAPORN KIATWUTHIAMORN

เลขหมู่.....  
เลขทะเบียน..... 82646  
วัน,เดือน,ปี..... 21 ก.ค. 2551



วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต

สาขาวิชาเทคโนโลยีสารสนเทศ

บัณฑิตวิทยาลัย

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ.2551

KMITL-2008-IT-M-001-277

**SUPERVISED SELF ORGANIZING MAP  
USING FUZZY MEMBERSHIP FUNCTION**

**JIRAPORN KIATWUTHIAMORN**

**A THESIS SUBMITTED IN PARTIAL FULFILLMENT  
OF THE REQUIREMENT FOR THE DEGREE OF  
MASTER OF SCIENCE PROGRAM IN INFORMATION TECHNOLOGY  
SCHOOL OF GRADUATE STUDIES  
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG**

**2008**

**KMITL-2008-IT-M-001-277**

**COPYRIGHT 2008**

**SCHOOL OF GRADUATE STUDIES**

**KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG**

หัวข้อวิทยานิพนธ์

โครงข่ายประสาทเทียมแบบปรับตัวเองภายในแบบมีผู้สอนโดยใช้ฟังก์ชันความเป็นสมาชิกของเซตของความคลุมเครือ

นักศึกษา

นางสาวจิราพร เกียรติวุฒิมอม

รหัสประจำตัว

47066128

ปริญญา

วิทยาศาสตรมหาบัณฑิต

สาขาวิชา

เทคโนโลยีสารสนเทศ

พ.ศ.

2551

อาจารย์ที่ปรึกษาวิทยานิพนธ์

รศ.ดร. อาริต ธรรมโน

### บทคัดย่อ

ในงานวิจัยนี้มีจุดมุ่งหมายที่จะพัฒนาแบบจำลองการจัดหมวดหมู่ข้อมูลแบบใหม่ โดยมีการนำโครงข่ายประสาทเทียมแบบการปรับตัวเองภายใน (Self Organizing Map) ซึ่งเป็นแบบจำลองทางด้านการจัดกลุ่มข้อมูล (Clustering) มาประยุกต์ใช้ร่วมกับแนวคิดต่างๆ เพื่อให้สามารถใช้กับการจัดหมวดหมู่ข้อมูล (Classification) ได้ ในงานวิจัยนี้ได้นำหลักการเรื่องค่าความเป็นสมาชิกของเซตของความคลุมเครือเข้ามาใช้ในการหาโหนดที่ชนะ ในขั้นตอนการแข่งขัน และในขั้นตอนการเรียนรู้ ได้นำกลไกการเรียนรู้แบบมีผู้สอน (Supervised Learning) เข้ามาประยุกต์ใช้พร้อมทั้งมีกระบวนการที่สามารถเพิ่มลดจำนวนโหนดในชั้นเอาต์พุตได้

ซึ่งจากแนวคิดข้างต้นทำให้ได้มาซึ่งแบบจำลองการจัดหมวดหมู่ข้อมูลแบบใหม่ที่มีประสิทธิภาพในการทำนายกลุ่มของข้อมูลทำให้มีความถูกต้องในการทำนายที่ดีมากยิ่งขึ้น

<b>Thesis Title</b>	Supervised Self Organizing Map Using Fuzzy Membership Function
<b>Student</b>	Miss Jiraporn Kaitwuttiamorn
<b>Student ID.</b>	47066128
<b>Degree</b>	Master of Science
<b>Program</b>	Information Technology
<b>Year</b>	2008
<b>Thesis Advisor</b>	Assoc. Prof. Dr. Arit Thammano

### **ABSTRACT**

This research aims to develop a new classification model by applying various concepts to Self Organizing Map (SOM), which is typically used for clustering data, in order to make it be able to solve classification task. In competition step, we use fuzzy membership function concept to find a winner node. Moreover, by applying supervised learning mechanism and increase-decrease node procedure, the efficiency of the proposed model is improved.

The above-mentioned new classification model is able to predict data class with more accuracy than the compared methods.

## กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงได้เป็นอย่างดีด้วยคำแนะนำ คำปรึกษา และความอนุเคราะห์ จาก รศ.ดร. อาริต ธรรมโน ซึ่งเป็นอาจารย์ผู้ควบคุมวิทยานิพนธ์ ด้วยอาจารย์เป็นผู้ที่ให้ความสนใจ แรงผลักดัน หัวใจ และติดตามความก้าวหน้าการทำวิทยานิพนธ์อย่างใกล้ชิด ตลอดจนให้คำปรึกษา คำแนะนำที่เป็นประโยชน์อย่างยิ่งในการทำวิทยานิพนธ์แก่ข้าพเจ้า ข้าพเจ้ารู้สึกซาบซึ้ง และ ขอขอบพระคุณเป็นอย่างสูง

ขอขอบพระคุณครู อาจารย์ทุกท่านที่ได้ประสิทธิ์ประสาทวิชา ความรู้ให้แก่ข้าพเจ้าตั้งแต่ออดีจนถึงปัจจุบัน

ขอขอบคุณเพื่อน พี่ และน้องๆ ในคณะเทคโนโลยีสารสนเทศ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบังทุกคนที่ให้คำแนะนำต่างๆ และคอยให้กำลังใจเสมอมา

สุดท้ายนี้ข้าพเจ้าขอขอบพระคุณ บิดา มารดา และครอบครัวของข้าพเจ้าที่เป็นกำลังใจ และให้การสนับสนุนในทุกเรื่องๆ ทำให้ข้าพเจ้าสามารถทำวิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงด้วยดี

คุณค่า และประโยชน์อันพึงมาจากวิทยานิพนธ์ฉบับนี้ ข้าพเจ้าขอบอบแด่ผู้มีพระคุณทุกท่าน ได้รับการสนับสนุนทุนการทำวิทยานิพนธ์จากบัณฑิตวิทยาลัย

จิราพร เกียรติวุฒิมร

# สารบัญ

	หน้า
บทคัดย่อภาษาไทย.....	I
บทคัดย่อภาษาอังกฤษ.....	II
กิตติกรรมประกาศ.....	III
สารบัญ.....	IV
สารบัญตาราง.....	VII
สารบัญรูป.....	VIII
บทที่ 1 บทนำ.....	1
1.1 ความเป็นมาและความสำคัญของปัญหา.....	1
1.2 ความมุ่งหมายและวัตถุประสงค์ของการศึกษา.....	1
1.3 ขอบเขตการวิจัย.....	1
1.4 ขั้นตอนของการศึกษา.....	2
1.5 ประโยชน์ที่คาดว่าจะได้รับ.....	2
1.6 โครงสร้างของวิทยานิพนธ์.....	2
บทที่ 2 หลักการและทฤษฎีที่เกี่ยวข้อง.....	4
2.1 การจัดหมวดหมู่ข้อมูล.....	4
2.2 โครงข่ายประสาทเทียมแบบการปรับตัวเองภายใน (SOM).....	5
2.2.1 ขั้นตอนการทำงานของโครงข่ายประสาทเทียมแบบการปรับตัวเองภายใน.....	6
2.2.1.1 ขั้นตอนการเตรียมการ.....	6
2.2.1.2 ขั้นตอนการแข่งขัน.....	6
2.2.1.3 ขั้นตอนการปรับปรุงค่าน้ำหนัก.....	7
2.2.1.4 ขั้นทำซ้ำ.....	7
2.3 ตรรกศาสตร์แห่งความคลุมเครือ (Fuzzy Logic).....	8
2.3.1 เซตของความคลุมเครือ (Fuzzy Set).....	8
2.4 โครงข่ายประสาทเทียมแบบซิมพลิไฟด์ฟัชชีอาร์ทแมพ (SFAM).....	12
2.4.1 กระบวนการทำงานของซิมพลิไฟด์ฟัชชีอาร์ทแมพ.....	13
2.4.1.1 กระบวนการเตรียมข้อมูล.....	14
2.4.1.2 กระบวนการเรียนรู้.....	14

## สารบัญ (ต่อ)

	หน้า
2.5 บทความงานวิจัยที่เกี่ยวข้อง .....	16
2.5.1 Hierarchical Overlapped SOM's for Pattern Classification .....	16
2.5.2 Concurrent Self-Organizing Maps for Pattern Classification.....	17
2.5.3 A supervised training algorithm for self-organizing maps for structure .....	19
2.5.4 Semi-supervised Learning of Dynamic Self-Organising Maps.....	22
2.5.5 A Comparison between a Modified Counter Propagation and an Extended Self-Organizing Map in Remotely Sensed Data Classification .....	25
บทที่ 3 Supervised SOM Using Fuzzy Membership Function.....	28
3.1 สถาปัตยกรรมของ Supervised SOM Using Fuzzy Membership Function.....	28
3.2 กระบวนการเรียนรู้ของ Supervised SOM Using Fuzzy Membership Function .....	29
3.2.1 ขั้นตอนเตรียมการ .....	29
3.2.2 ขั้นตอนการเรียนรู้แบบจำลอง .....	30
3.2.2.1 ขั้นตอนการแข่งขัน.....	30
3.2.2.2 ขั้นตอนการปรับปรุงค่าน้ำหนัก.....	31
3.2.2.3 ขั้นตอนการวนซ้ำ .....	34
3.2.2.4 ขั้นตอนการลบ .....	34
3.2.2.5 ขั้นตอนการวนซ้ำ .....	34
3.2.3 ขั้นตอนการทดสอบแบบจำลอง.....	34
3.3 ตัวอย่างการทำงาน .....	38
3.4 การเปรียบเทียบหลักการทำงานของแบบจำลอง.....	50
บทที่ 4 ผลการทดสอบการจัดหมวดหมู่ของข้อมูลโดยใช้ Supervised Self Organizing Map Using Fuzzy Membership Function.....	53
4.1 ข้อมูลที่ใช้ในการทดลอง.....	53
4.1.1 ข้อมูลทดสอบ .....	53
4.1.2 ข้อมูลมาตรฐาน .....	54

## สารบัญ (ต่อ)

	หน้า
4.2 การกำหนดพารามิเตอร์.....	57
4.3 ผลการทดลอง .....	58
4.3.1 ข้อมูลทดสอบ .....	58
4.3.2 ข้อมูลมาตรฐาน .....	61
4.4 สรุปผลการทดลอง.....	64
4.4.1 สรุปผลการทดลองของชุดข้อมูลทดสอบ .....	64
4.4.2 สรุปผลการทดลองของชุดข้อมูลมาตรฐาน.....	65
บทที่ 5 สรุปผลการวิจัยและข้อเสนอแนะ .....	71
5.1สรุปผลงานวิจัย.....	71
5.2 ปัญหาที่พบในงานวิจัย.....	71
5.3 แนวทางการพัฒนาในอนาคต .....	72
บรรณานุกรม .....	73
ภาคผนวก.....	75
ภาคผนวก ก ผลงานวิจัยที่ได้รับการตีพิมพ์เผยแพร่ .....	76
ประวัติผู้เขียน.....	81

# สารบัญตาราง

ตารางที่	หน้า
2.1 แสดงระดับความเป็นสมาชิกแบบ Crisp Set และ Fuzzy Set.....	11
3.1 แสดงค่าความกว้างของ $\sigma$ ทางซ้ายของแต่ละคลัสเตอร์ในแต่ละคุณลักษณะ.....	45
3.2 แสดงค่าความกว้างของ $\sigma$ ทางขวาของแต่ละคลัสเตอร์ในแต่ละคุณลักษณะ.....	45
3.3 แสดงชุดของอินพุตกับคลัสเตอร์ที่ชนะ.....	47
3.4 แสดงค่าความเป็นสมาชิกของแต่ละคลัสเตอร์.....	47
3.5 แสดงค่าความกว้างของ $\sigma$ ทางซ้ายของแต่ละคลัสเตอร์ในแต่ละคุณลักษณะ.....	48
3.6 แสดงค่าความกว้างของ $\sigma$ ทางขวาของแต่ละคลัสเตอร์ในแต่ละคุณลักษณะ.....	49
3.7 แสดงผลการทดลองเปรียบเทียบการใช้เกณฑ์การตัดสินใจโหนดชนะที่ต่างกัน.....	52
4.1 การแบ่งข้อมูลของชุดข้อมูล Flower1, Flower2 และ Sawtooth.....	54
4.2 การแบ่งข้อมูลของชุดข้อมูล Fan.....	54
4.3 การแบ่งข้อมูลของชุดข้อมูล Iris.....	55
4.4 การแบ่งข้อมูลของชุดข้อมูล Wine.....	55
4.5 การแบ่งข้อมูลของชุดข้อมูล Ionosphere.....	56
4.6 การแบ่งข้อมูลของชุดข้อมูล Haberman.....	57
4.7 การแบ่งข้อมูลของชุดข้อมูล Heart-Statlog.....	57
4.8 ผลการทดลองของข้อมูล Flower1.....	58
4.9 ผลการทดลองของข้อมูล Flower2.....	59
4.10 ผลการทดลองของข้อมูล Sawtooth.....	60
4.11 ผลการทดลองของข้อมูล Fan.....	60
4.12 ผลการทดลองของข้อมูล Iris.....	61
4.13 ผลการทดลองของข้อมูล Wine.....	62
4.14 ผลการทดลองของข้อมูล Ionosphere.....	62
4.15 ผลการทดลองของข้อมูล Haberman.....	63
4.16 ผลการทดลองของข้อมูล Heart-Statlog.....	64

# สารบัญรูป

รูปที่	หน้า
2.1 ลักษณะของโครงข่ายประสาทเทียมแบบการปรับตัวเองภายใน	5
2.2 ลักษณะการปรับค่าน้ำหนักของนิวรอนเพื่อนบ้าน	6
2.3 แสดงขอบเขตของตรรกศาสตร์แบบเดิมและตรรกศาสตร์แห่งความคลุมเครือ	8
2.4 แสดงลักษณะฟังก์ชันความเป็นสมาชิกแบบสามเหลี่ยม	9
2.5 แสดงลักษณะฟังก์ชันความเป็นสมาชิกแบบสี่เหลี่ยมคางหมู	10
2.6 แสดงลักษณะฟังก์ชันความเป็นสมาชิกแบบเกาส์เซียน	10
2.7 แสดงการใช้ฟังก์ชันเขตเข้ามาแบ่งเขตของคนตัวสูง	12
2.8 ลักษณะ โครงสร้างของ โครงข่ายประสาทเทียมแบบซิมพลิไฟด์ฟuzzyอาร์ทแมพ	13
2.9 ลักษณะ โครงสร้างของ Hierarchical Overlapped SOM	16
2.10 ขั้นตอนการเรียนรู้ของ Concurrent Self-Organizing Maps	18
2.11 ขั้นตอนการจัดหมวดหมู่ของ Concurrent Self-Organizing Maps	18
2.12 โครงสร้างกราฟและการเก็บข้อมูลของกราฟในรูปแบบตาราง	19
2.13 ลักษณะ โครงสร้างของ Supervised GSOM	22
2.14 ลักษณะ โครงสร้างของ Extended Self-Organizing Maps	25
3.1 แสดงสถาปัตยกรรมของ Supervised SOM Using Fuzzy Membership Function	29
3.2 แสดงโครงสร้างการทำงานของ Supervised SOM Using Fuzzy Membership Function	36
3.3 แสดงโครงสร้างการทำงานของ การปรับค่าน้ำหนักกรณีที่ 1	37
3.4 แสดงโครงสร้างการทำงานของ การปรับค่าน้ำหนักกรณีที่ 2	37
3.5 แสดงรูปแบบของอินพุตที่ใช้เป็นข้อมูลทดลองโดยเครื่องหมาย $\Delta$ แทนคลาส 1 และเครื่องหมาย $\times$ แทนคลาส 2	38
3.6 แสดงค่าน้ำหนักที่ได้หลังจากการเรียนรู้โดยเครื่องหมาย $\Delta$ แทนคลาส 1 และเครื่องหมาย $\times$ แทนคลาส 2	49
4.1 ลักษณะของชุดข้อมูลทดสอบ	53
4.2 Scatter Plot ของข้อมูลที่ใช้ในการทดสอบแบบจำลองที่นำเสนอของชุดข้อมูล Flower 1	67
4.3 Scatter Plot ของข้อมูลที่ใช้ในการทดสอบ LVQ ของชุดข้อมูล Flower 1	67
4.4 Scatter Plot ของข้อมูลที่ใช้ในการทดสอบแบบจำลองที่นำเสนอของชุดข้อมูล Flower 2	68
4.5 Scatter Plot ของข้อมูลที่ใช้ในการทดสอบ LVQ ของชุดข้อมูล Flower 2	68
4.6 Scatter Plot ของข้อมูลที่ใช้ในการทดสอบแบบจำลองที่นำเสนอของชุดข้อมูล Sawtooth	69

## สารบัญรูป(ต่อ)

รูปที่	หน้า
4.7 Scatter Plot ของข้อมูลที่ใช้ในการทดสอบ LVQ ของชุดข้อมูล Sawtooth.....	69
4.8 Scatter Plot ของข้อมูลที่ใช้ในการทดสอบแบบจำลองที่นำเสนอของชุดข้อมูล Fan .....	70
4.9 Scatter Plot ของข้อมูลที่ใช้ในการทดสอบ LVQ ของชุดข้อมูล Fan.....	70

# บทที่ 1

## บทนำ

### 1.1 ความเป็นมาและความสำคัญของปัญหา

การจัดหมวดหมู่ข้อมูล (Classification) คือ กระบวนการค้นหาเขตของแบบจำลองโดยพิจารณาจากคุณสมบัติต่างๆ ของข้อมูลนั้น เพื่อนำแบบจำลองที่ได้ไปใช้ในการทำนายกลุ่มของข้อมูล ในปัจจุบันการจัดหมวดหมู่ข้อมูลโดยใช้โครงข่ายประสาทเทียม (Neural Network) มีอยู่หลายวิธี ซึ่งแต่ละวิธีมีลักษณะเฉพาะของตัวเอง เช่น ลักษณะทางสถาปัตยกรรมของโครงข่ายหลักการที่ใช้ในการคำนวณ และกลไกในการเรียนรู้ ซึ่งแต่ละวิธีให้ประสิทธิภาพในการทำนายกลุ่มของข้อมูลได้ถูกต้องระดับหนึ่ง ในงานวิจัยนี้มีจุดมุ่งหมายที่จะพัฒนาแบบจำลองการจัดหมวดหมู่ข้อมูลแบบใหม่ โดยมีการนำโครงข่ายประสาทเทียมแบบการปรับตัวเองภายใน (Self Organizing Map) ซึ่งเป็นแบบจำลองทางด้านการจัดกลุ่มข้อมูล (Clustering) ที่มีหลักการทำงานที่ไม่ซับซ้อนเข้าใจง่าย และสามารถนำไปประยุกต์ใช้ในงานต่างๆ ได้อย่างกว้างขวางมาปรับให้มีความสามารถในการจัดหมวดหมู่ข้อมูลได้ ซึ่งจากแนวคิดข้างต้นทำให้ได้มาซึ่งแบบจำลองการจัดหมวดหมู่ข้อมูลแบบใหม่ที่มีประสิทธิภาพในการทำนายกลุ่มของข้อมูล ทำให้มีความถูกต้องในการทำนายที่ดีมากยิ่งขึ้น

### 1.2 ความมุ่งหมายและวัตถุประสงค์ของการศึกษา

1. เพื่อพัฒนาโครงข่ายประสาทเทียมแบบการปรับตัวเองภายในให้สามารถที่จะนำมาใช้จัดหมวดหมู่ข้อมูลได้
2. เพื่อพัฒนาแบบจำลองในการจัดหมวดหมู่ข้อมูลที่มีหลักการเรียนรู้ที่ง่ายต่อความเข้าใจ มีโครงสร้างที่ไม่ซับซ้อน และมีประสิทธิภาพในการจัดหมวดหมู่ของข้อมูลใกล้เคียงกับมนุษย์
3. เพื่อเป็นแนวทางในการศึกษา และพัฒนาความรู้ด้านปัญญาประดิษฐ์ และด้านอื่นๆ ที่เกี่ยวข้อง

### 1.3 ขอบเขตการวิจัย

1. การวิจัยนี้เป็นการทดลองแบบ off-line
2. แบบจำลองใช้ได้กับข้อมูลที่เป็นตัวเลขเท่านั้น

3. การวัดประสิทธิภาพของแบบจำลองโดยใช้อัตราส่วนความถูกต้องในการจัดกลุ่มข้อมูล

#### 1.4 ขั้นตอนของการศึกษา

1. ศึกษาแบบจำลองพื้นฐานทางด้านการจัดหมวดหมู่ของข้อมูล
2. ศึกษาโครงข่ายประสาทเทียมแบบการปรับตัวเองภายใน
3. ศึกษาโครงข่ายประสาทเทียมแบบการปรับตัวเองภายในที่ใช้กระบวนการเรียนรู้แบบมีผู้สอน (Supervised learning)
4. ศึกษาโครงข่ายประสาทเทียมแบบซิมพลิไฟด์ฟัซซีอาร์ทแมพ (Simplified Fuzzy ARTMAP) เพื่อศึกษาถึงกระบวนการทำงานของโครงข่ายประสาทเทียมที่มีการเรียนรู้แบบมีผู้สอน
5. ศึกษา และทดลองปรับแนวคิดทั้งหมดมาใช้ร่วมกันด้วยวิธีต่างๆ
6. วิเคราะห์และสรุปผลเพื่อเลือกหาแนวทางที่เหมาะสมในการสร้างแบบจำลองใหม่ที่ประยุกต์เอาแนวคิดทั้งหมดมาใช้ร่วมกัน
7. สร้างแบบจำลองใหม่ตามแนวทางที่ได้
8. แก้ไขปรับปรุงแบบจำลองให้มีความเหมาะสมยิ่งขึ้น
9. ทดลองและทดสอบแบบจำลองใหม่กับข้อมูลรูปแบบต่างๆ
10. สรุปผลการทดลอง

#### 1.5 ประโยชน์ที่คาดว่าจะได้รับ

1. ทราบถึงบทความงานวิจัยต่างๆ ที่เกี่ยวข้องกับงานวิจัยครั้งนี้
2. สามารถพัฒนาแบบจำลองโดยใช้โครงข่ายประสาทเทียมแบบการปรับตัวเองภายใน
3. เป็นพื้นฐานในการเรียนรู้การจัดหมวดหมู่ของข้อมูลโดยใช้โครงข่ายประสาทเทียมรูปแบบต่างๆ เพื่อใช้ในการพัฒนาต่อไปในอนาคต

#### 1.6 โครงสร้างของวิทยานิพนธ์

วิทยานิพนธ์ฉบับนี้ประกอบด้วยรายละเอียดของเนื้อหาเป็นบทต่างๆ ดังนี้  
**บทที่ 1** บทนำ กล่าวถึงความเป็นมาของงานวิจัย ความมุ่งหมายและวัตถุประสงค์ของการศึกษา ขอบเขตการวิจัย ขั้นตอนของการศึกษา และประโยชน์ที่คาดว่าจะได้รับ

**บทที่ 2** กล่าวถึงทฤษฎีพื้นฐานที่ใช้ในการวิจัย ทฤษฎีโครงข่ายประสาทเทียมแบบการปรับตัวเองภายใน (Self Organizing Map) ทฤษฎีโครงข่ายประสาทเทียมแบบซิมพลิไฟด์ฟัซซีอาร์ทแมพ (Simplified Fuzzy ARTMAP) และตรรกศาสตร์แห่งความคลุมเครือ (Fuzzy Logic)

**บทที่ 3** กล่าวถึงทฤษฎี Supervised SOM Using Fuzzy Membership Function และตัวอย่างการทำงานของอัลกอริทึมอย่างละเอียด

**บทที่ 4** กล่าวถึงการทดลองและผลการทดลอง

**บทที่ 5** สรุปผลการวิจัยและข้อเสนอแนะ บทนี้จะสรุปภาพรวมทั้งหมดที่ได้จากการวิจัย ปัญหาที่พบจากการวิจัย พร้อมทั้งได้เสนอแนวทางในการทำวิจัยต่อไป

## บทที่ 2

# หลักการและทฤษฎีที่เกี่ยวข้อง

### 2.1 การจัดหมวดหมู่ข้อมูล (Classification)

การจัดหมวดหมู่ข้อมูล คือ กระบวนการค้นหาเขตของแบบจำลองเพื่อนำแบบจำลองที่ได้ไปใช้ในการทำนายกลุ่มของข้อมูลโดยพิจารณาจากคุณสมบัติต่างๆ ของข้อมูล [1] ซึ่งแต่ละข้อมูลจะถูกกำหนดกลุ่มของข้อมูลไว้ก่อนหน้าแล้ว กระบวนการจัดหมวดหมู่ข้อมูลประกอบด้วย 2 ขั้นตอน ดังนี้

1) การสร้างแบบจำลองพยากรณ์ (Model Construction) คือ การสร้างแบบจำลองจากกลุ่มข้อมูลทดลอง (Training data) โดยทำการจัดกลุ่มข้อมูลทดลองที่มีความคล้ายคลึงกันไว้ในกลุ่มเดียวกัน โดยใช้การเรียนรู้แบบมีผู้สอน (Supervised learning) ซึ่งแบบจำลองจะทำการเรียนรู้จากค่าเป้าหมาย (Target output) ที่ถูกกำหนดไว้แล้ว และในขั้นตอนนี้สามารถทำการปรับเปลี่ยนแบบจำลองได้จนกว่าจะเป็นที่พอใจ

2) การทดสอบความถูกต้องของแบบจำลอง (Model Evaluation) คือ การนำแบบจำลองที่ได้จากข้อ 1) ไปใช้จัดหมวดหมู่ข้อมูลทดสอบ (Testing data) โดยข้อมูลทดสอบเป็นข้อมูลที่ไม่ถูกนำไปใช้ในขั้นตอนการสร้างแบบจำลอง ถ้าความถูกต้องของแบบจำลองมากพอเป็นที่ยอมรับได้แบบจำลองจะถูกนำไปใช้ในการจัดหมวดหมู่ต่อไป

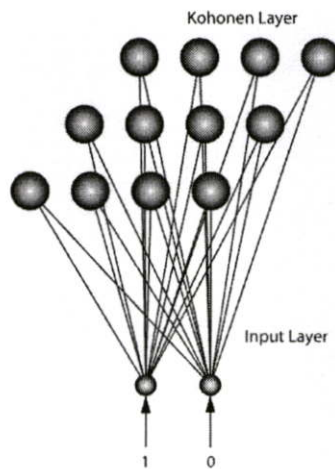
ในปัจจุบันมีการนำเทคนิคต่างๆ มาใช้เพื่อให้ได้แบบจำลองในการจัดหมวดหมู่ข้อมูลที่มีประสิทธิภาพโดยเทคนิคหนึ่งที่มีการนำมาใช้กันอย่างแพร่หลาย คือ การจัดหมวดหมู่ข้อมูลโดยใช้โครงข่ายประสาทเทียม (Neural Network) ซึ่งเป็นรูปแบบการประมวลผลที่ได้แนวคิดพื้นฐานมาจากการทำงานของระบบประสาทภายในสมองมนุษย์ [4] เช่น การรู้จำ และการจำแนกรูปแบบต่างๆ ซึ่งการที่โครงข่ายประสาทเทียมจะสามารถทำงานเหล่านี้ได้ โครงข่ายประสาทเทียมจะต้องผ่านขั้นตอนการเรียนรู้ ซึ่งประเภทของการเรียนรู้แบ่งได้เป็น 2 ประเภท [1] ดังนี้

1) การเรียนรู้โดยมีผู้สอน (Supervised learning) การเรียนรู้วิธีนี้จะต้องทราบผลลัพธ์ที่แท้จริงของข้อมูลที่นำมาเรียนรู้ก่อน จากนั้นนำผลลัพธ์ที่ได้จากระบบมาเปรียบเทียบกับผลลัพธ์ที่แท้จริงแล้วทำการปรับค่าน้ำหนักจนได้ผลลัพธ์ที่ใกล้เคียง หรือเหมือนกับผลลัพธ์ที่แท้จริงมากที่สุด

2) การเรียนรู้แบบไม่มีผู้สอน (Unsupervised learning) การเรียนรู้วิธีนี้ข้อมูลที่นำมาใช้จะไม่ถูกกำหนดหมวดหมู่ของข้อมูลมาก่อนหน้า เมื่อข้อมูลถูกส่งเข้ามาในแบบจำลอง แบบจำลองจะทำหน้าที่แบ่งกลุ่มข้อมูลโดยจัดข้อมูลที่มีลักษณะใกล้เคียงกันให้อยู่ในกลุ่มเดียวกัน

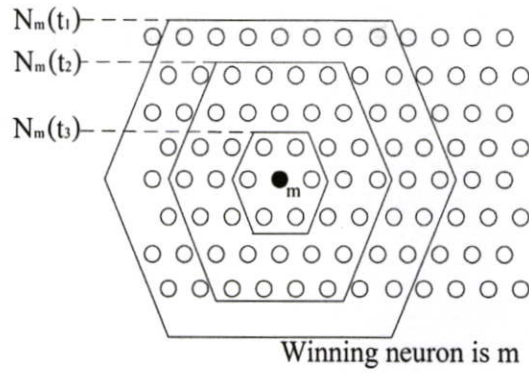
## 2.2 โครงข่ายประสาทเทียมแบบการปรับตัวเองภายใน (Self Organizing Map : SOM)

โครงข่ายประสาทเทียมแบบการปรับตัวเองภายใน ถูกนำเสนอโดย Kohonen [6] โดย SOM เป็นแบบจำลองทางด้านการจัดกลุ่มข้อมูล (clustering) ที่ได้รับความนิยมในการนำไปประยุกต์ใช้ในงานต่างๆ อย่างแพร่หลาย แบบจำลองของ SOM มีลักษณะเป็นโครงข่ายที่มีจำนวน 2 ชั้น คือ ชั้นอินพุต (Input layer) ที่ทำหน้าที่รับอินพุตเข้ามาแล้วส่งต่อไปยังชั้นเอาต์พุตทันทีโดยไม่มีการประมวลผลใดๆ และชั้นเอาต์พุต (Output layer) หรือ Kohonen layer ที่ทำหน้าที่ในส่วนของ การประมวลผล ลักษณะของโครงข่ายประสาทเทียมแบบการปรับตัวเองภายใน (SOM) แสดงได้ดังรูปที่ 2.1



รูปที่ 2.1 ลักษณะของโครงข่ายประสาทเทียมแบบการปรับตัวเองภายใน[6]

โครงข่ายประสาทเทียมแบบการปรับตัวเองภายใน เป็นการเรียนรู้แบบไม่มีผู้สอน (Unsupervised Learning) โดยเมื่อมีอินพุตเข้ามาในกระบวนการเรียนรู้ ชั้นอินพุตจะทำหน้าที่รับอินพุตเข้ามาแล้วส่งต่อไปยังชั้นเอาต์พุตทันทีโดยไม่มีการประมวลผลใดๆ ในชั้นเอาต์พุตโครงข่ายจะถูกกระตุ้น นิวรอนแต่ละตัวในชั้นเอาต์พุตจะมีการแข่งขันกัน ซึ่งนิวรอนที่ตอบสนองต่อการกระตุ้นที่ดีที่สุดจะเป็นผู้ที่ชนะ จากนั้นนิวรอนที่ชนะจะทำการปรับค่าน้ำหนัก (weight) เพื่อให้เข้าใกล้อินพุตมากขึ้น โดยนิวรอนที่ชนะจะถูกปรับค่าน้ำหนักมากกว่านิวรอนที่อยู่รอบๆ นิวรอนที่ชนะส่วนนิวรอนที่อยู่ไกลออกไปจากนิวรอนที่ชนะการปรับค่าน้ำหนักก็จะลดน้อยลงจนกระทั่งไม่มีการปรับค่าน้ำหนัก โดยลักษณะการปรับลดค่าน้ำหนักอาจปรับลดตามรูปแบบฟังก์ชันลด (decreasing function) ซึ่งฟังก์ชันลดจะทำการลดค่าน้ำหนักลงเมื่อรอบเวลาการทำงานเพิ่มขึ้นในที่นี้จะปรับลดตามฟังก์ชันรูประฆังคว่ำ (bell curve function) โดยลักษณะการปรับค่าน้ำหนักของนิวรอนที่อยู่รอบๆ นิวรอนที่ชนะ หรือนิวรอนเพื่อนบ้าน (neighborhood) เมื่อเวลาผ่านไปแสดงได้ดังรูปที่ 2.2



รูปที่ 2.2 ลักษณะการปรับค่าน้ำหนักของนิวรอนเพื่อนบ้าน ณ เวลา  $t$  ใดๆ ;  $t_1 < t_2 < t_3$  [6]

## 2.2.1 ขั้นตอนการทำงานของโครงข่ายประสาทเทียมแบบการปรับตัวเองภายใน

### 2.2.1.1 ขั้นเตรียมการ

ทำการกำหนดค่าพารามิเตอร์ต่างๆ ที่จะใช้ในการเรียนรู้แบบจำลองให้กับโครงข่าย เช่น ค่าน้ำหนักของโครงข่ายซึ่งกำหนดโดยการสุ่ม, ค่าอัตราการเรียนรู้ ( $\alpha$ ) ณ เวลา  $t$  ใดๆ ซึ่งกำหนดให้มิก่าบวก โดยทั่วไปจะอยู่ในช่วง 0 ถึง 1, จำนวนนิวรอนในชั้นเอาต์พุต และขนาดความกว้างของ Gaussian function ณ เวลา  $t$  ใดๆ

### 2.2.1.2 ขั้นตอนการแข่งขัน

เมื่อชั้นอินพุต (Input layer) รับอินพุตเข้ามาจะส่งต่อไปยังชั้นเอาต์พุตเมื่อโครงข่ายถูกกระตุ้นนิวรอนแต่ละตัวในชั้นเอาต์พุตจะทำการแข่งขันกัน โดยใช้ยูคลิเดียนดิสเทนซ์ (Euclidean distance) ดังสมการที่ 2.1 แล้วค่านิวรอนตัวไหนมีการตอบสนองการกระตุ้นได้ดีที่สุด (นิวรอนที่มีระยะทางน้อยที่สุด) นิวรอนตัวนั้นจะเป็นผู้ชนะ

$$\|x - w_m\| = \min_i \{\|x - w_i\|\} \quad (2.1)$$

โดย	$x$	คือ	รูปแบบอินพุต
	$i$	คือ	จำนวนนิวรอนในชั้นเอาต์พุต
	$m$	คือ	นิวรอนที่ชนะ

### 2.2.1.3 ขั้นตอนการปรับปรุงค่าน้ำหนัก

Kohonen ได้เสนอวิธีในการปรับค่าน้ำหนักไว้ คือ ถ้า  $m$  คือ นิวรอนตัวที่ชนะแล้ว  $m$  จะได้รับการปรับค่าน้ำหนักมากที่สุด และนิวรอนที่อยู่บริเวณรอบๆ นิวรอนตัวที่ชนะ (neighborhood) ก็ควรถูกปรับค่าน้ำหนักด้วย โดยให้นิวรอนที่อยู่บริเวณรอบๆ ปรับค่าน้ำหนักน้อยกว่านิวรอนตัวที่ชนะ นิวรอนตัวที่ถัดออกไปอีกก็ปรับค่าน้ำหนักน้อยลงไปอีกจนถึงจุดที่ไม่มีการปรับค่าน้ำหนักใดๆ ซึ่งการปรับค่าน้ำหนัก ณ เวลา  $t$  ใดๆ สามารถคำนวณได้โดยใช้สมการที่ 2.2 โดย  $\Delta w_i(t)$  คือ ค่าน้ำหนักที่เกิดการเปลี่ยนแปลง ซึ่งสามารถคำนวณได้โดยใช้สมการที่ 2.3 และ 2.4 โดยอัตราการลดลงของขนาดความกว้างของนิวรอนเพื่อนบ้าน ( $\sigma$ ) และอัตราการเรียนรู้ ( $\alpha$ ) จะลดลงสัมพันธ์กับจำนวนรอบที่เพิ่มขึ้นเมื่อเวลาผ่านไป

$$w_i(t+1) = w_i(t) + \Delta w_i(t) \quad (2.2)$$

$$\Delta w_i(t) = \alpha(N_i, t)[x(t) - w_i(t)] \quad (2.3)$$

$$\alpha(N_i, t) = \alpha(t) \exp[-\|r_i - r_m\|^2 / \sigma^2(t)] \quad (2.4)$$

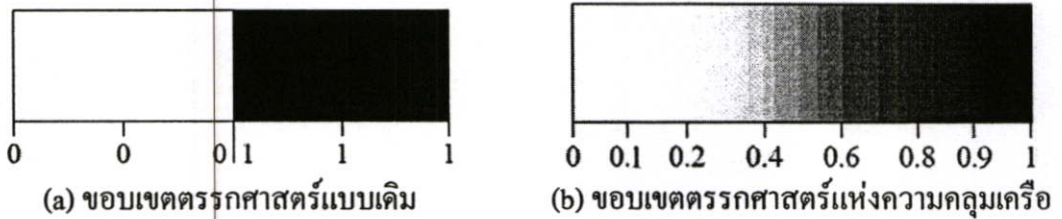
โดย	$w_i(t+1)$	คือ	ค่าน้ำหนักใหม่ที่ได้จากการปรับค่า
	$\Delta w_i(t)$	คือ	ค่าน้ำหนักที่เปลี่ยนแปลงไปตาม Neighborhood function
	$\alpha(N_i, t)$	คือ	อัตราการปรับ Neighborhood function ที่เวลา $t$ ใดๆ
	$r_m$	คือ	ตำแหน่งของนิวรอนที่ชนะ
	$r_i$	คือ	ตำแหน่งของนิวรอนรอบนิวรอนที่ชนะ
	$\sigma(t)$	คือ	ขนาดความกว้างของ Gaussian function
	$\alpha(t)$	คือ	อัตราการเรียนรู้ที่เวลา $t$ ใดๆ
	$t$	คือ	รอบการทำงานปัจจุบัน

### 2.2.1.4 ขั้นทำซ้ำ

วนกลับไปทำขั้นตอนที่ 2.2.1.3 ซ้ำจนกระทั่งค่าน้ำหนักไม่มีการเปลี่ยนแปลงหรือมีการเปลี่ยนแปลงน้อยมาก

## 2.3 ตรรกศาสตร์แห่งความคลุมเครือ (Fuzzy Logic)

ตรรกศาสตร์แห่งความคลุมเครือ เป็นหลักการที่เลียนแบบการใช้เหตุผล และวิถีคิดของมนุษย์ [7] จากหลักตรรกศาสตร์แบบเดิม (Boolean logic) ค่าความเป็นจริงมีแค่จริงหรือเท็จ หรือ 0 กับ 1 แต่ตรรกศาสตร์แห่งความคลุมเครือจะให้ค่าความเป็นจริงในลักษณะ multi-value คือ ค่าความจริงอยู่ระหว่าง 0 ถึง 1 ขอบเขตของตรรกศาสตร์แบบเดิม และตรรกศาสตร์แห่งความคลุมเครือ แสดงได้ดังรูปที่ 2.3



รูปที่ 2.3 ขอบเขตของตรรกศาสตร์แบบเดิมและตรรกศาสตร์แห่งความคลุมเครือ [7]

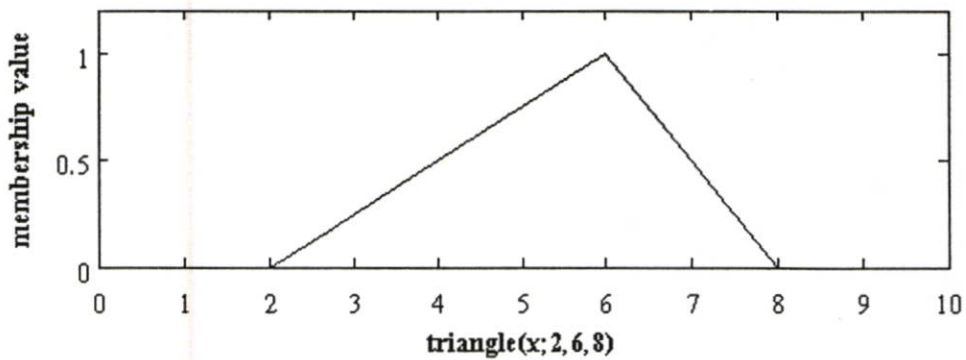
### 2.3.1 เซตของความคลุมเครือ (Fuzzy Set)

เซตของความคลุมเครือ คือ ฟังก์ชันหรือเซตที่อธิบายความน่าจะเป็นของเหตุการณ์ที่มีลักษณะคลุมเครือไม่ชัดเจน (Vagueness) หรือมีความไม่แน่นอน (Uncertainty) [3] เซตของความคลุมเครือเกิดขึ้นเนื่องจากข้อจำกัดของ classical set หรือ crisp set ที่ระดับความเป็นสมาชิกจะมีได้เพียง 2 ระดับ คือ 0 กับ 1 หรือเพียงแค่เป็นสมาชิกหรือไม่เป็นสมาชิก ขอบเขตการเปลี่ยนแปลงค่าความเป็นสมาชิกของเซตของความคลุมเครือมีลักษณะเป็นค่าที่ต่อเนื่อง และค่าความเป็นสมาชิกของเซตของความคลุมเครือมีค่าอยู่ระหว่าง 0 ถึง 1 ซึ่งสามารถนำมาประยุกต์ใช้ในโลกของความจริงที่งานหรือเหตุการณ์ต่างๆ ที่อาจมีความคลุมเครือหรือขัดแย้งในตัวเอง เช่น เซตของอายุคนอาจแบ่งเป็น วัยทารก วัยเด็ก วัยรุ่น วัยกลางคน วัยชรา ซึ่งแต่ละช่วงวัยอาจใช้เกณฑ์การแบ่งที่แตกต่างกัน บางเกณฑ์อาจจัดว่าช่วงวัยรุ่นคือช่วงอายุ 12-18 ปี ในขณะที่อีกเกณฑ์บอกว่าวัยรุ่นคือช่วงอายุ 18-20 ปี ซึ่งเป็นเกณฑ์ที่มีความขัดแย้งกัน การกำหนดค่าความเป็นสมาชิกของเซตของความคลุมเครืออาจใช้ฟังก์ชันความเป็นสมาชิก (membership function) ในการกำหนดระดับความเป็นสมาชิกของตัวแปรที่ต้องการใช้งาน การเลือกฟังก์ชันความเป็นสมาชิกควรเลือกให้เหมาะสม และครอบคลุมกับข้อมูลที่รับเข้ามา โดยตัวอย่างของฟังก์ชันความเป็นสมาชิกมีดังนี้ [2]

1. ฟังก์ชันสามเหลี่ยม (Triangular membership function) ประกอบด้วยพารามิเตอร์ 3 ตัว คือ  $a$ ,  $b$  และ  $c$  โดยถ้า  $x$  คือข้อมูลใดๆ เราสามารถหาค่าความเป็นสมาชิกของ  $x$  ได้โดยใช้สมการที่ 2.5

$$\text{triangle}(x : a, b, c) = \begin{cases} 0 & x < a \\ (x-a)/(b-a) & a \leq x \leq b \\ (c-x)/(c-b) & b \leq x \leq c \\ 0 & x > c \end{cases} \quad (2.5)$$

โดย  $a$  คือ จุดขอบล่างด้านซ้ายของสามเหลี่ยม  
 $c$  คือ จุดขอบล่างด้านขวาของสามเหลี่ยม  
 $b$  คือ จุดยอดของสามเหลี่ยม



รูปที่ 2.4 ลักษณะฟังก์ชันความเป็นสมาชิกแบบสามเหลี่ยม

2. ฟังก์ชันสี่เหลี่ยมคางหมู (Trapezoidal membership function) ประกอบด้วยพารามิเตอร์ 4 ตัว คือ  $a$ ,  $b$ ,  $c$  และ  $d$  โดยถ้า  $x$  คือข้อมูลใดๆ เราสามารถหาค่าความเป็นสมาชิกของ  $x$  ได้โดยใช้สมการที่ 2.6

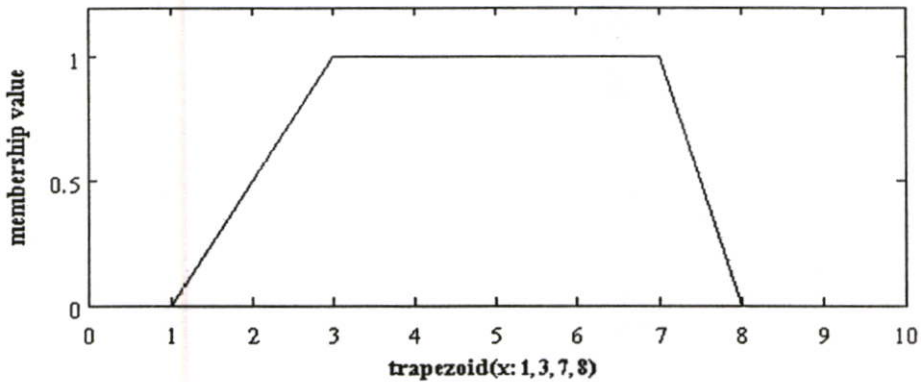
$$\text{trapezoid}(x : a, b, c, d) = \begin{cases} 0 & x < a \\ (x-a)/(b-a) & a \leq x < b \\ 1 & b \leq x < c \\ (d-x)/(d-c) & c \leq x < d \\ 0 & x \geq d \end{cases} \quad (2.6)$$

- โดย
- $a$  คือ จุดขอบด้านล่างด้านซ้ายของรูปสี่เหลี่ยมคางหมู
  - $b$  คือ จุดขอบด้านบนด้านซ้ายของรูปสี่เหลี่ยมคางหมู
  - $c$  คือ จุดขอบด้านบนด้านขวาของรูปสี่เหลี่ยมคางหมู
  - $d$  คือ จุดขอบด้านล่างด้านขวาของรูปสี่เหลี่ยมคางหมู

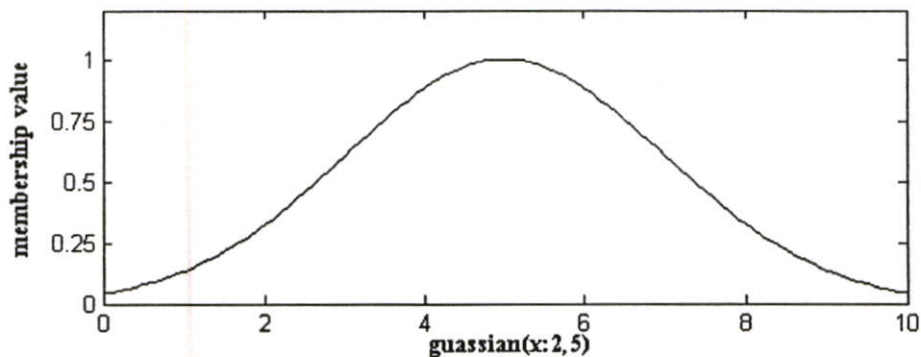
3. ฟังก์ชันเกาส์เซียน (Gaussian membership function) ประกอบด้วยพารามิเตอร์ 2 ตัวคือ  $m$  และ  $\sigma$  ถ้า  $x$  คือข้อมูลใดๆ เราสามารถหาค่าความเป็นสมาชิกของ  $x$  ได้โดยใช้สมการที่ 2.7

$$\text{gaussian}(x : m, \sigma) = \exp\left(-\frac{(x - m)^2}{\sigma^2}\right) \quad (2.7)$$

- โดย
- $m$  คือ ค่าเฉลี่ยของข้อมูล
  - $\sigma$  คือ ค่าเบี่ยงเบนมาตรฐานของข้อมูล



รูปที่ 2.5 ลักษณะฟังก์ชันความเป็นสมาชิกแบบสี่เหลี่ยมคางหมู



รูปที่ 2.6 ลักษณะฟังก์ชันความเป็นสมาชิกแบบเกาส์เซียน

### ตัวอย่างการใช้ฟuzzyเซตแบ่งคนตัวสูง

สมมติกำหนดให้ S แทนเซตของคนตัวสูง ถ้าบุคคลที่มีความสูงมากกว่า 180 เซนติเมตร [2] จะถือว่าเป็นคนสูง ดังสมการที่ 2.8

$$S = \{x \mid x > 180\} \quad (2.8)$$

โดย  $x$  คือ ค่าความสูงของคนแต่ละคน

จากการกำหนดลักษณะของคนสูงแบบนี้ถ้าเป็นการแบ่งแบบ classical set คนที่มีความสูง 180 เซนติเมตรจะจัดว่าเป็นคนตัวไม่สูง แต่คนที่มีความสูง 181 เซนติเมตรจะจัดว่าเป็นคนตัวสูงทั้งที่มีความสูงต่างกันเพียง 1 เซนติเมตรเท่านั้น

เราจะใช้ฟuzzyเซตเข้ามาแบ่งเซตของคนตัวสูงโดยกำหนดฟังก์ชันความเป็นสมาชิกดังสมการที่ 2.9

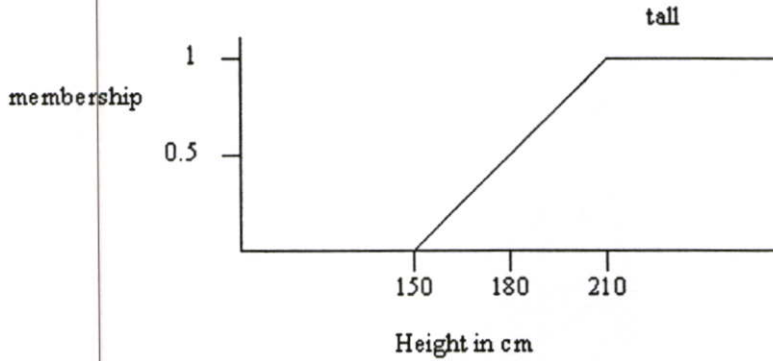
$$tall(x;150,210) = \begin{cases} 0 & x < a \\ (x-a)/(b-a) & a \leq x \leq b \\ 1 & x > b \end{cases} \quad (2.9)$$

โดย  $x$  คือ ข้อมูลใดๆ ที่เราต้องการหาค่าความเป็นสมาชิก

ถ้า  $x$  คือค่าความสูงของคนแต่ละคนแล้วจะสามารถหาค่าความเป็นสมาชิกที่ความสูงระดับต่างๆ ได้ดังแสดงในตาราง 2.1

ตารางที่ 2.1 แสดงระดับความเป็นสมาชิกแบบ Crisp Set และ Fuzzy Set

Name	Height(cm)	Degree of membership	
		Crisp	Fuzzy
a	208	1	0.9667
b	205	1	0.9167
c	198	1	0.8
d	181	1	0.5167
e	179	0	0.4833
f	172	0	0.3667
g	167	0	0.2833
h	158	0	0.133
i	155	0	0.0833
j	152	0	0.033



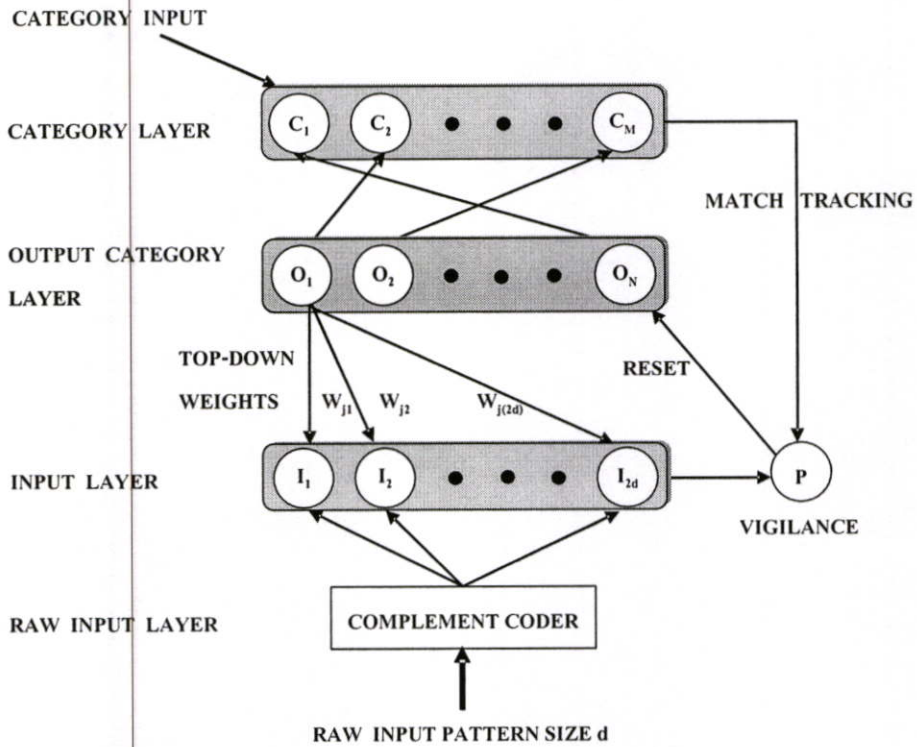
รูปที่ 2.7 การใช้ฟังก์ชันเซตเข้ามาแบ่งเซตของคนตัวสูง [3]

เนื่องจากฟังก์ชันเซตมีการใช้ฟังก์ชันความเป็นสมาชิกที่เป็นค่าต่อเนื่อง โดยขอบเขตจะค่อยๆ เปลี่ยนแปลงจาก 0 เป็น 1 ทำให้คนที่มีความสูงต่างกันไม่มาก เช่น 179 กับ 181 ยังคงเป็นสมาชิกของเซตของคนตัวสูงทั้งคู่ แต่จะมีค่าความเป็นสมาชิกในระดับที่ต่างกัน คือ คนสูง 179 จะมีค่าความเป็นสมาชิกที่ 0.48 ซึ่งน้อยกว่าคนที่สูง 181 ซึ่งมีค่าความเป็นสมาชิกที่ 0.51

#### 2.4 โครงข่ายประสาทเทียมแบบซิมพลิไฟด์ฟัซซีอาร์ทแมพ (Simplified Fuzzy ARTMAP: SFAM)

โครงข่ายประสาทเทียมแบบซิมพลิไฟด์ฟัซซีอาร์ทแมพถูกพัฒนาโดย Tom Kasuba ในปี 1993 [5] สถาปัตยกรรมของโครงข่ายประสาทเทียมแบบซิมพลิไฟด์ฟัซซีอาร์ทแมพประกอบด้วย 3 ชั้น คือชั้นอินพุต (Input Layer) ชั้นเอาต์พุต (Output Category Layer) และชั้นแคททิกอรี (Category Layer) ลักษณะโครงสร้างของโครงข่ายประสาทเทียมแบบซิมพลิไฟด์ฟัซซีอาร์ทแมพแสดงได้ดังรูปที่ 2.7

ลักษณะของอินพุตที่จะเข้าสู่โครงข่ายจะต้องเป็นจำนวนจริงใดๆ มีค่าอยู่ระหว่าง 0 ถึง 1 โดยอินพุตจะถูกส่งผ่านขั้นตอนคอมพลิเมนต์โคคิง (complement coding) ซึ่งจะทำการเพิ่มขนาดของอินพุตให้เป็นสองเท่าของขนาดอินพุตจริง หลังจากนั้นอินพุตที่ผ่านการเพิ่มขนาดแล้ว (I) จะถูกส่งไปยังชั้นอินพุต (Input Layer) โดยทุกโหนดในชั้นอินพุตจะทำการเชื่อมต่อกับโหนดในชั้นเอาต์พุต (Output Category Layer) ผ่านทางค่าน้ำหนัก  $w_{ji}$  ซึ่งเป็นค่าน้ำหนักที่เชื่อมจากชั้นเอาต์พุต โหนดที่  $j$  มายังชั้นอินพุต โหนดที่  $i$  โดยเรียกชื่อของค่าน้ำหนักนี้ว่าค่าน้ำหนักแบบบนลงล่าง (Top-down weights) ชั้นถัดไปคือชั้นแคททิกอรี (Category Layer) ซึ่งเป็นชั้นที่บอกถึงประเภทของกลุ่มข้อมูลที่โครงข่ายต้องเรียนรู้จำนวนทั้งหมด  $M$  ชนิด โดยแต่ละโหนดในชั้นเอาต์พุตจะชี้ไปยังตำแหน่งในชั้นแคททิกอรีได้เพียงตำแหน่งเดียวเท่านั้น



รูปที่ 2.8 ลักษณะ โครงสร้างของโครงข่ายประสาทเทียมแบบซิมพลิไฟด์ฟัซซีอาร์ทแมพ [5]

กลไกที่มีบทบาทสำคัญที่ทำให้เกิดกระบวนการเรียนรู้ คือ ค่าวิจิลแลนซ์ (vigilance) ซึ่งแทนด้วยสัญลักษณ์  $\rho$  (rho) วิจิลแลนซ์จะทำหน้าที่เป็นตัวควบคุมการจัดกลุ่มของเอาต์พุตโหนดว่ารูปแบบอินพุตจะต้องมีความคล้ายกันมากน้อยเพียงใดจึงจะสามารถจัดให้อยู่ในกลุ่มของเอาต์พุตเดียวกันได้ โดยถ้าวิจิลแลนซ์มีค่าสูงหมายความว่ารูปแบบอินพุตและเอาต์พุตโหนดจะต้องมีความเหมือนกันมากจึงจะรวมกลุ่มกันได้ และหากวิจิลแลนซ์มีค่าต่ำหมายความว่ารูปแบบอินพุตและเอาต์พุตโหนดที่มีความเหมือนกันน้อยก็สามารถรวมกลุ่มกันได้ ทั้งนี้ถ้าค่าวิจิลแลนซ์ยิ่งสูงก็จะทำให้จำนวนโหนดในชั้นเอาต์พุตมีมากขึ้นตามไปด้วย โดยทั่วไปผู้ใช้จะเป็นผู้กำหนดค่าวิจิลแลนซ์พื้นฐาน (baseline vigilance) โดยวิจิลแลนซ์จะมีค่าอยู่ในช่วง 0 ถึง 1

ขั้นตอนการทำแมตช์แทรคคิง (match tracking) เป็นขั้นตอนที่โครงข่ายทำการปรับค่าวิจิลแลนซ์ในระหว่างการเรียนรู้ ขั้นตอนนี้จะถูกทำงานเมื่อการจำแนกชนิดข้อมูลในช่วงการเรียนรู้โครงข่ายมีความผิดพลาด โดยจะทำการเพิ่มค่าวิจิลแลนซ์ขึ้นจากค่าวิจิลแลนซ์พื้นฐาน

#### 2.4.1 กระบวนการทำงานของซิมพลิไฟด์ฟัซซีอาร์ทแมพ

### 2.4.1.1 กระบวนการเตรียมข้อมูล

ลักษณะอินพุตในซิมพลิไฟด์พีชชีอาร์ทแมพจะมีค่าอยู่ในช่วง 0 ถึง 1 และอินพุตที่เข้ามาจะถูกเพิ่มขนาดด้วยการทำคอมพลิเมนต์โคคิง (complement coding) โดยถ้าอินพุตเวกเตอร์  $a$  มีจำนวนคุณลักษณะ (feature) ทั้งหมด  $d$  จำนวนคุณลักษณะแล้ว เวกเตอร์ที่ทำคอมพลิเมนต์ ( $\bar{a}$ ) สามารถหาได้จากสมการที่ 2.10

$$\bar{a} = 1 - a \quad (2.10)$$

อินพุตเวกเตอร์ที่ผ่านการทำคอมพลิเมนต์โคคิงแล้วจะมีขนาดเป็น 2 เท่า ( $2d$ ) ดังนั้นสามารถแทนค่าของอินพุตที่ผ่านการทำคอมพลิเมนต์โคคิงแล้วได้ดังสมการที่ 2.11

$$I = (a, \bar{a}) = (a_1, \dots, a_d, \bar{a}_1, \dots, \bar{a}_d) \quad (2.11)$$

### 2.4.1.2 กระบวนการเรียนรู้

โครงข่ายซิมพลิไฟด์พีชชีอาร์ทแมพจะใช้ฟังก์ชันกระตุ้น (activation function) ในการกระตุ้นโหนดในชั้นเอาต์พุตให้มีการตอบสนองต่ออินพุตที่ถูกส่งเข้ามา โดยฟังก์ชันกระตุ้นสามารถคำนวณได้โดยใช้สมการที่ 2.12

$$T_j(I) = \frac{|I \wedge W_j|}{\alpha + |W_j|} \quad (2.12)$$

โดย	$T_j$	คือ	ค่าการกระตุ้นของโหนดในชั้นเอาต์พุตลำดับที่ $j$
	$W_j$	คือ	ค่าน้ำหนักของโหนดในชั้นเอาต์พุตลำดับที่ $j$
	$I$	คือ	อินพุตที่ผ่านการทำคอมพลิเมนต์โคคิง
	$\alpha$	คือ	ค่าคงที่ที่มีค่าต่ำๆ โดยทั่วไปจะถูกกำหนดให้มีค่าประมาณ 0.0000001

เมื่อทุกโหนดในชั้นเอาต์พุตถูกกระตุ้นแล้วระบบจะทำการเลือกเอาต์พุตโหนดที่มีค่า  $T_j$  สูงที่สุดให้เป็นโหนดที่ชนะดังแสดงในสมการที่ 2.13

$$winner = \arg_j \max(T_j) \quad (2.13)$$

หากมีโหนดที่มีค่า  $T_j$  สูงที่สุดมากกว่า 1 โหนด ระบบจะทำการเลือกเอาที่พุดโหนดที่มีค่า  $j$  น้อยที่สุด (เอาที่พุดโหนดที่ถูกสร้างในระบบก่อนโหนดอื่นๆ) ให้เป็นโหนดที่ชนะ

ทำการตรวจสอบความเหมือนของอินพุตเวกเตอร์กับโหนดที่ชนะ โดยใช้แมตซ์ฟังก์ชันดังสมการที่ 2.14 ซึ่งเป็นการนำอินพุตที่ผ่านการทำคอมพลิเมนต์โคคดิงแล้วมาเปรียบเทียบกับค่าน้ำหนักของโหนดที่ชนะ

$$\frac{|I \wedge W_j|}{d} \geq \rho \quad (2.14)$$

ทำการเปรียบเทียบค่าแมตซ์ฟังก์ชันที่ได้กับค่าวิจิลแลนซ์

ก. ถ้าค่าแมตซ์ฟังก์ชันมากกว่าค่าวิจิลแลนซ์ แสดงว่าเกิดสภาวะเรโซแนนซ์ (resonance) ซึ่งหมายถึงเอาที่พุดโหนดที่ชนะนั้นคือเพียงพอที่จะเป็นตัวแทนของอินพุต  $I$  ได้ ให้ทำการเปรียบเทียบกลุ่มของอินพุตกับกลุ่มของโหนดที่ชนะ

- ถ้ากลุ่มของอินพุต  $I$  เป็นกลุ่มเดียวกับเอาที่พุดโหนดที่ชนะแล้ว ค่าน้ำหนักของเอาที่พุดโหนดที่ชนะจะถูกปรับค่าตามสมการที่ 2.15 เพื่อทำการเรียนรู้อินพุตตัวใหม่

$$W_j^{new} = \beta(I \wedge W_j^{old}) + (1 - \beta)W_j^{old} \quad (2.15)$$

โดย	$W_j^{new}$	คือ	ค่าน้ำหนักใหม่ของโหนดลำดับที่ $j$
	$W_j^{old}$	คือ	ค่าน้ำหนักเก่าของโหนดลำดับที่ $j$
	$j$	คือ	ลำดับของโหนดเอาที่พุด
	$I$	คือ	อินพุตที่ผ่านการทำคอมพลิเมนต์โคคดิง
	$\beta$	คือ	ค่าอัตราการเรียนรู้จะมีค่าอยู่ระหว่าง 0 ถึง 1

ค่า  $\beta$  ที่มีค่ามากการเรียนรู้ก็จะรวดเร็วมากขึ้น ในการปรับค่าน้ำหนักสามารถทำให้ง่ายขึ้นได้ถ้ากำหนดให้อัตราการเรียนรู้มีค่าเท่ากับ 1 (fast learning) สมการสำหรับปรับค่าน้ำหนักสามารถเขียนได้ดังสมการที่ 2.16

$$W_j^{new} = (I \wedge W_j^{old}) \quad (2.16)$$

- ถ้ากลุ่มอินพุต  $I$  ต่างกับกลุ่มของเอาที่พุดโหนดที่ชนะ แสดงว่าเกิดแคทิกอรีมิสแมตซ์ (category mismatch) ค่าน้ำหนักจะไม่ถูกเปลี่ยนแปลง โดยจะสังเกตเห็นว่าเมื่อมีอินพุตตัวใดตัวหนึ่งเข้ามาทำการเรียนรู้ในระบบจะมีเอาที่พุดโหนดเพียงโหนดเดียวเท่านั้นที่จะถูกปรับค่า

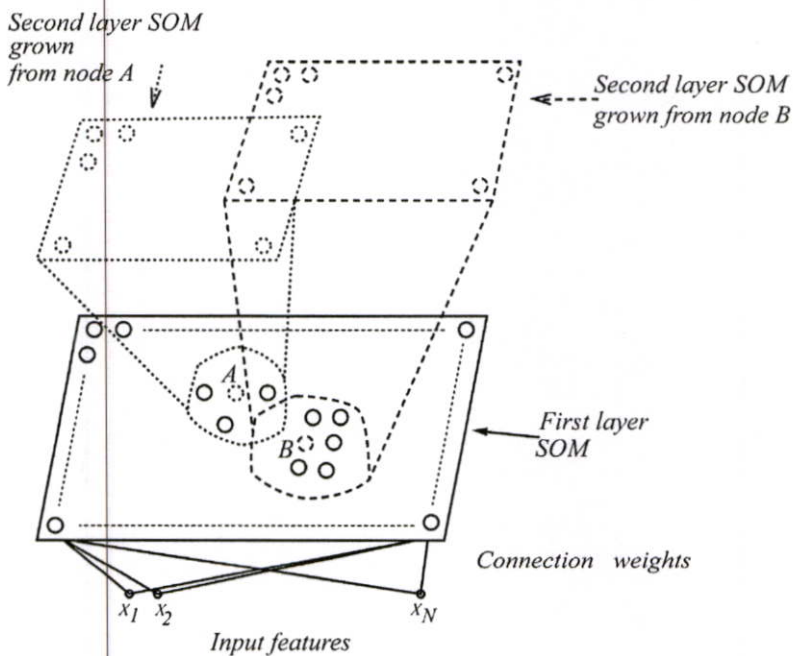
ข. ถ้าค่าแมทริกซ์ฟังก์ชันน้อยกว่าค่าวิจิเลนซ์ จะทำการเพิ่มค่าวิจิเลนซ์ขึ้นเล็กน้อยเพื่อไม่ให้เอาต์พุตโหนดตัวนั้นเป็นผู้ชนะ และพิจารณาค่า  $T$  ของโหนดที่มีค่ามากที่สุดถัดไป

หากพิจารณาจนครบแล้วไม่มีค่า  $T$  ใดๆ ที่ผ่านเงื่อนไขตามสมการที่ 2.14 หรือกลุ่มของเอาต์พุตโหนดที่ชนะกับกลุ่มของอินพุต  $I$  เป็นคนละกลุ่มกัน หรือในกรณีที่เป็นการอินพุตตัวแรกจะทำการเพิ่มเอาต์พุตโหนดใหม่ โดยกลุ่มของเอาต์พุตโหนดที่เพิ่มมานั้นจะถูกกำหนดให้เป็นกลุ่มเดียวกับกลุ่มของอินพุต  $I$  และมีค่าน้ำหนักเท่ากับค่าอินพุต  $I$  ที่ผ่านการทำคอมพลิเมนต์โคคิติงแล้ว

## 2.5 บทความงานวิจัยที่เกี่ยวข้อง

### 2.5.1 Hierarchical Overlapped SOM's for Pattern Classification [9]

งานวิจัยนี้ถูกคิดค้นโดย Markus P.N. Suganthan ในปี 1999 โดยงานวิจัยนี้เป็นการเสนอแนวทางการจัดหมวดหมู่ของข้อมูลจากการรวมโครงข่ายประสาทเทียมแบบการปรับตัวเองภายในแบบลำดับชั้นที่ซ้อนทับกันอยู่ โครงข่ายที่ซ้อนทับกันได้มาจากการผ่านกระบวนการเรียนรู้แบบไม่มีผู้สอนและมีผู้สอนแล้วนำโครงข่ายที่ซ้อนทับกันมาผ่านการตัดแต่ง (pruned) โดยการรวม (merging) หรือการลบนิรอรอน (removing) เพื่อให้ได้ผลการจัดหมวดหมู่ของข้อมูลที่ดี



รูปที่ 2.9 ลักษณะ โครงสร้างของ Hierarchical Overlapped SOM [9]

ขั้นตอนการทำงาน

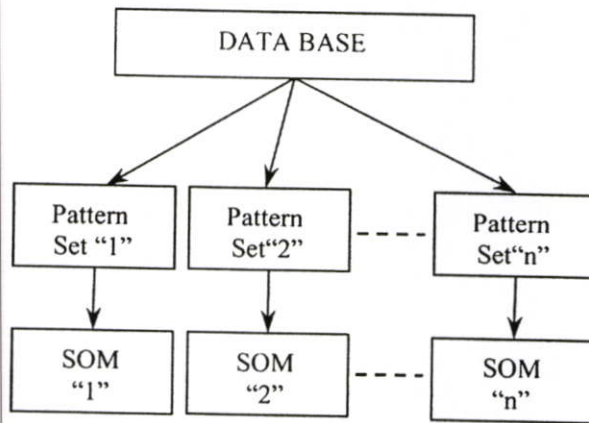
- 1) ทำการจัดกลุ่มข้อมูลโดยใช้กระบวนการเรียนรู้แบบไม่มีผู้สอนของโครงข่ายประสาทเทียมแบบการปรับตัวเองภายใน (Unsupervised SOM Algorithm)
  - 2) พิจารณาการจัดกลุ่มของข้อมูลจากแผนผังของโหนดในชั้นเอาท์พุท
  - 3) นำกลุ่มที่ได้จากข้อ 1 มาผ่านการเรียนรู้ที่ละกลุ่มโดยใช้กระบวนการเรียนรู้แบบมีผู้สอนของ LVQ 2 ทำให้ได้โครงข่ายของแต่ละกลุ่มที่มีการจัดกลุ่มที่ละเอียดขึ้น
  - 4) นำโครงข่ายที่ได้จากการเรียนรู้ที่มีการซ้อนทับกันอยู่มากัดแต่ง (pruned) โดยใช้การรวม (merging) หรือการลบ (removing) นีวรอน
- การรวมนีวรอนพิจารณาจากนีวรอนที่อยู่ติดกันว่ายู่กลุ่มเดียวกันหรือไม่ ถ้าอยู่จะทำการรวมนีวรอน
- การลบนีวรอนพิจารณาจากการเป็นตัวแทนของข้อมูล ถ้านีวรอนตัวไหนมีข้อมูลเป็นสมาชิกน้อยกว่าค่าที่กำหนดไว้ (กำหนดจำนวนสมาชิกเท่ากับ 3) ถือว่าเป็นตัวแทนที่ไม่ดีจะทำการลบนีวรอนนั้นทิ้งไป
- 5) นำโครงข่ายที่ผ่านการตัดแต่งมาผ่านการเรียนรู้โดยใช้กระบวนการเรียนรู้แบบมีผู้สอนของ LVQ 2 อีกครั้ง
  - 6) ทำซ้ำตั้งแต่ข้อ 1 จนครบตามเงื่อนไขที่กำหนด

งานวิจัยนี้ได้ทำการทดลองกับชุดข้อมูล NIST Special Database 19 ของ The National Institute of Standards and Technology โดยเปรียบเทียบผลการทดลองที่ได้กับงานวิจัยเรื่อง Neural-Network Classifiers for Recognizing Totally Unconstrained Handwritten Numeral ของ S.B.Cho, เรื่อง Handwritten Digit Recognition using Two-layer Self-Organizing Maps ของ J.Wu และเรื่อง Off-line Handwritten Numeral Recognition by Perturbation Method ของ T.M.Ha พบว่างานวิจัยนี้มีความถูกต้องมากกว่างานของ S.B.Cho 2.38% และมากกว่างานของ J.Wu 1.13% แต่เมื่อเทียบกับงานของ T.M.Ha งานวิจัยนี้มีความถูกต้องน้อยกว่า 0.91%

**2.5.2 Concurrent Self-Organizing Maps for Pattern Classification [10]**

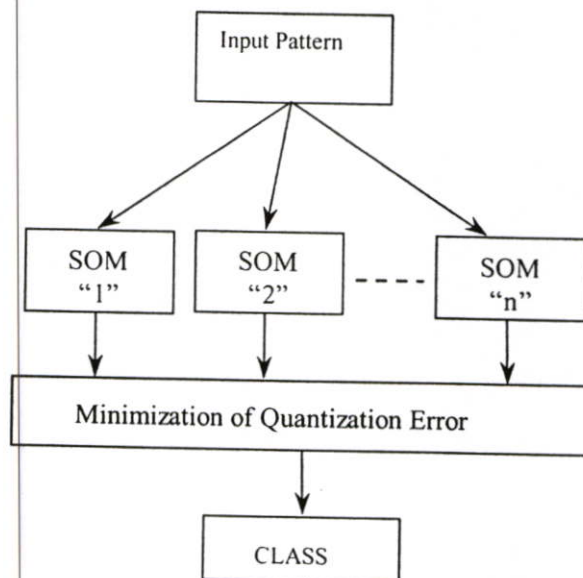
งานวิจัยนี้ถูกคิดค้นโดย Markus Victor-Emil NEAGOE และ Armand-Dragos ROPOT ในปี 2002 โดยงานวิจัยนี้เป็นการเสนอแนวคิดการแทนโครงข่ายประสาทเทียมแบบการปรับตัวเองภายในด้วยกลุ่มของโครงข่ายประสาทเทียมแบบการปรับตัวเองภายในแบบพิเศษ โดยภายในระบบประกอบด้วยกลุ่มของโครงข่ายประสาทเทียมแบบการปรับตัวเองภายในแบบพิเศษจำนวนเท่ากับกลุ่มของข้อมูล (class) ความแตกต่างของโครงข่ายประสาทเทียมแบบการปรับตัวเองภายในแบบพิเศษอยู่ที่แต่ละโครงข่ายจะเรียนรู้กับชุดข้อมูลเฉพาะที่ถูกแบ่งกลุ่มไว้แล้ว (ข้อมูลเฉพาะคือข้อมูล

จาก 1 กลุ่มเท่านั้น) เมื่อผ่านการเรียนรู้แต่ละโครงข่ายจะได้ผลการเรียนรู้ที่ดีที่สุดสำหรับกลุ่ม 1 กลุ่ม ขั้นตอนการทำงานของ Concurrent Self-Organizing Maps แสดงดังรูปที่ 2.10



รูปที่ 2.10 ขั้นตอนการเรียนรู้ของ Concurrent Self-Organizing Maps [10]

ในขั้นตอนการจัดหมวดหมู่ของข้อมูล เมื่อผ่านการเรียนรู้โครงข่ายจะประกอบด้วยกลุ่มของโครงข่ายประสาทเทียมแบบการปรับตัวเองภายในที่เหมาะสมกับแต่ละกลุ่มข้อมูลเมื่อมีข้อมูลเข้ามาโครงข่ายจะใช้ Minimization of Quantization Error เป็นเกณฑ์ตัดสินหากกลุ่มให้กับข้อมูลที่เข้ามาโดยพิจารณาจากค่าความแตกต่างที่น้อยที่สุดของค่าข้อมูลเข้ากับค่าที่ได้จากแต่ละโครงข่าย ซึ่งจะแสดงถึงกลุ่ม (class) ที่เกี่ยวข้อง ขั้นตอนการจัดหมวดหมู่ของ Concurrent Self-Organizing Maps แสดงดังรูปที่ 2.11



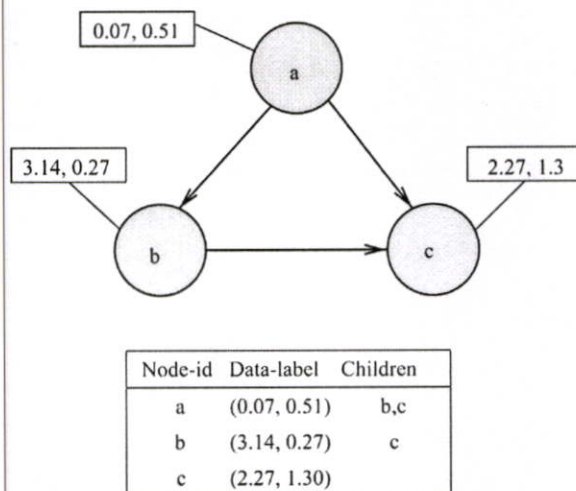
รูปที่ 2.11 ขั้นตอนการจัดหมวดหมู่ของ Concurrent Self-Organizing Maps [10]

งานวิจัยนี้ถูกนำไปประยุกต์ใช้กับการรู้จำใบหน้าของคน โดยทดลองกับชุดข้อมูลภาพใบหน้าในฐานข้อมูล ORL ซึ่งประกอบด้วยภาพใบหน้าของบุคคลจำนวน 40 คน คนละ 10 ภาพรวมทั้งสิ้น 400 ภาพ เป็นภาพระดับสีเทา 8 บิต ขนาด  $91 \times 112$  พิกเซลภาพมีการให้แสงในตำแหน่งที่ต่างกัน และมีการแสดงออกทางใบหน้าที่หลากหลาย ได้แก่ ภาพที่ลืมตา/หลับตา ยิ้ม/ไม่ยิ้ม นอกจากนี้ยังมีรายละเอียดของภาพที่แตกต่างกัน ได้แก่ ใส่แว่น/ไม่ใส่แว่น โดยเปรียบเทียบผลการทดลองที่ได้กับโครงข่ายประสาทเทียมแบบการปรับตัวเองภายในที่ใช้หลักดั้งเดิมในการจัดหมวดหมู่และโครงข่ายประสาทเทียมแบบการปรับตัวเองภายในที่ใช้หลัก k-NN ในการจัดหมวดหมู่พบว่างานวิจัยนี้มีความถูกต้องมากกว่า 20% และ 7.5% ตามลำดับ

### 2.5.3 A supervised training algorithm for self-organizing maps for structure [11]

งานวิจัยนี้ถูกคิดค้นโดย Markus Hagenbuchner และ Ah Chung Tsoi ในปี 2005 โดยงานวิจัยนี้เป็นการนำแนวคิดการกำหนดกลุ่ม (class) ให้กับข้อมูลไว้ล่วงหน้าก่อนนำไปใช้ในกระบวนการเรียนรู้ และมีการใช้ rejection term ในการปฏิเสธกรณีโหนดที่ถูกเลือกมีกลุ่มต่างจากข้อมูลที่เข้ามาเรียนรู้ ในกรณีที่ข้อมูลเข้ามาเรียนรู้ไม่สมบูรณ์หรือไม่มีกลุ่มแบบจำลองจะเปลี่ยนไปใช้การเรียนรู้แบบไม่มีผู้สอน

งานวิจัยนี้ถูกนำไปประยุกต์ใช้กับโครงสร้างข้อมูลประเภทกราฟที่มีข้อมูลเก็บอยู่ โดยเราจะใช้ข้อมูลอื่นๆ นอกเหนือจากข้อมูลของอินพุตเพื่อประกอบการเรียนรู้ แต่ละโหนดข้อมูลที่นำมาใช้ในการเรียนรู้เกิดจากการเชื่อมต่อกันของข้อมูลสัญลักษณ์ (data label  $l$ ) ขนาด  $p$  มิติกับเซตของตำแหน่งของลูกแต่ละตัวของโหนดนั้น ( $c$ ) ขนาด  $qo$  มิติ ( $q$  คือมิติของโครงข่ายกำหนดไว้ที่ 2,  $o$  คือจำนวนลูกมากที่สุดที่มีได้ของกราฟ) ดังนั้นข้อมูลที่ได้จากการเชื่อมต้อมีขนาด  $n$  มิติ  $n = (p + 2o)$



รูปที่ 2.12 แสดงโครงสร้างกราฟและการเก็บข้อมูลของกราฟในรูปแบบตาราง [11]

จากรูปที่ 2.12 เราจะนำข้อมูลในตารางที่ได้จากโครงสร้างประเภทกราฟซึ่งแสดงถึงความสัมพันธ์ระหว่างข้อมูลมาใช้ประกอบกับข้อมูลอินพุตเพื่อช่วยในการเรียนรู้ในแต่ละแถวจะแสดงถึงโหนดในกราฟ เช่นแถวแรกของตารางเก็บข้อมูลต่างๆ ของโหนด a (ฟิลด์ Node-id คือชื่อของโหนด, ฟิลด์ Data-label เก็บข้อมูลที่ตั้ง, ฟิลด์ Children เก็บข้อมูลลูกของโหนด a )

ขั้นตอนการทำงานของแบบจำลองประกอบด้วย 2 ส่วน

1. ขั้นตอนการแข่งขัน เมื่อโครงข่ายถูกกระตุ้นโหนดที่ตอบสนองต่อการกระตุ้นได้ดีที่สุดคือโหนดที่ชนะซึ่งหาได้จากสมการที่ 2.17

$$r = \operatorname{argmin}_i \|(x - m_i)\Lambda\| \quad (2.17)$$

โดย	$x$	คือ	รูปแบบอินพุต
	$i$	คือ	จำนวนนิวรอนในชั้นเอาต์พุต
	$m$	คือ	นิวรอนในชั้นเอาต์พุต
	$\Lambda$	คือ	เมตริกซ์diagonal ขนาด $n \times n$ มิติ

โดยส่วนของ Euclidean distance ในสมการที่ 2.17 ได้จากสมการที่ 2.18

$$d = \sqrt{\mu_1 \sum_{i=1}^p (l_i - m_i)^2 + \mu_2 \sum_{j=1}^{2o} (c_j - m_{p+j})^2} \quad (2.18)$$

โดย	$\mu_1$	คือ	เซตของข้อมูลบนเส้นทแยงมุมของเมตริกซ์
	$\mu_2$	คือ	เซตของข้อมูลที่ไม่อยู่บนเส้นทแยงมุมของเมตริกซ์
	$l_i$	คือ	ข้อมูลสัญลักษณ์ตำแหน่งที่ $i$ ของอินพุต
	$m_i$	คือ	ข้อมูลสัญลักษณ์ตำแหน่งที่ $i$ ของนิวรอนในชั้นเอาต์พุต
	$c_j$	คือ	เซตของตำแหน่งของโหนดที่ชนะตำแหน่งที่ $i$ ของอินพุต
	$m_i$	คือ	เซตของตำแหน่งของโหนดที่ชนะตำแหน่งที่ $i$ ของนิวรอนในชั้น เอาต์พุต

โหนดที่มีระยะทางน้อยที่สุดคือโหนดที่ชนะแล้วจึงทำการปรับค่าน้ำหนักในชั้นต่อไป

2. ขั้นตอนการปรับค่าน้ำหนัก การปรับค่าน้ำหนักจะพิจารณาจากคลาสของโหนดที่ชนะกับข้อมูลที่เข้ามาเรียนรู้โดยใช้สมการที่ 2.19 โดยค่าน้ำหนักที่เปลี่ยนแปลงไปสามารถคำนวณได้โดยใช้สมการที่ 2.20

$$m_{ij}^{new} = m_{ij}^{old} + \Delta m_{ij} \quad (2.19)$$

$$\Delta m_{ij} = \begin{cases} -\epsilon \alpha(t) f(\Delta_{jr}) h(x_i, m_{ij}); & \text{if } x \text{ and } m_j \text{ are in different classes} \\ \alpha(t) f(\Delta_{jr}) (x_i - m_{ij}) & ; \quad \text{else} \end{cases} \quad (2.20)$$

โดย  $m_j$  คือ โหนดที่ชนะ  
 $\alpha(t)$  คือ อัตราการเรียนรู้ซึ่งลดลงสู่ 0 เมื่อเวลาผ่านไป  
 $f(\Delta_{jr})$  คือ neighbourhood function  
 $t$  คือ จำนวนรอบการทำงาน  
 $\epsilon$  คือ rejection factor ซึ่งมีอิทธิพลต่อ rejection term  $h(.)$

วัตถุประสงค์ของ rejection term คือย้าย  $m_j$  และเพื่อนบ้านของมันออกจากข้อมูลที่เข้ามาถ้าคลาสของ  $m_j$  ต่างจากข้อมูลที่เข้ามา โดย rejection term กำหนดตามสมการที่ 2.21

$$h(x_i, m_{ij}) = \text{sgn}(x_i - m_{ij}) \left( \frac{\sigma_i}{\sigma_i + |x_i - m_{ij}|} \right) \quad (2.21)$$

โดย  $\text{sgn}(\cdot)$  คือ signum function  
 $\sigma_i$  คือ ค่าความเบี่ยงเบนสามารถหาได้จากสมการที่ 2.22

$$\sigma_i = \sqrt{\frac{\sum_{l=1}^N (x_{li} - \bar{x}_i)^2}{N}} \quad (2.22)$$

$$\bar{x}_i = \frac{1}{N} \sum_{l=1}^N x_{li}$$

โดย  $N$  คือ จำนวนข้อมูลทั้งหมดที่เรียนรู้

ถ้า neighborhood function สามารถพิจารณาได้จาก Gaussian function ตามสมการที่ 2.23

$$f(\Delta_{jr}) = \exp\left(-\frac{\|l_i - l_r\|^2}{2\sigma(t)^2}\right) \quad (2.23)$$

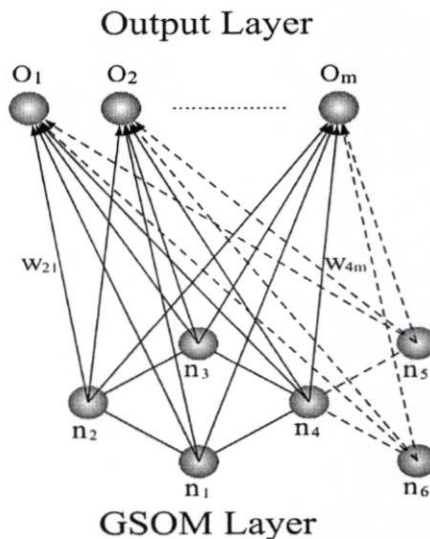
โดย  $\sigma(t)$  คือ ขอบเขตของเพื่อนบ้านลดลงตาม Gaussian function  
 $l_i$  คือ ตำแหน่งของนิวรอนที่อยู่รอบนิวรอนที่ชนะ  
 $l_r$  คือ ตำแหน่งของนิวรอนที่ชนะ

งานวิจัยนี้ได้ทำการทดลองกับชุดข้อมูลรูปสัญลักษณ์ที่จัดทำโดย Document Processing Group, Center for Automation Research, University of Maryland ซึ่งข้อมูลที่ใช้เรียนรู้มีจำนวน 300 ตัวอย่าง ใน 39 คลาสโดยทำการเปรียบเทียบผลการทดลองกับการเรียนรู้แบบไม่มีผู้สอนพบว่างานวิจัยนี้ได้ผลความถูกต้องมากกว่า 2.9%

#### 2.5.4 Semi-supervised Learning of Dynamic Self-Organising Maps [12]

งานวิจัยนี้ถูกคิดค้นโดย Arthur Hsu และ Saman K.Halgamuge ในปี 2006 โดยงานวิจัยนี้เป็นการเสนอแนวคิดการรวม Growing cell structures ที่นำเสนอโดย Fritzke กับกระบวนการเรียนรู้ของ Growing Self-Organising Map เพื่อสามารถทำการเรียนรู้ในกรณีที่ข้อมูลเข้าไม่สมบูรณ์ได้ลักษณะโครงสร้างของ Semi-supervised learning layer มีลักษณะเหมือนกับโครงสร้างของ Supervised GSOM ที่แสดงในรูปที่ 2.13 คำนำหน้าจากชั้น GSOM ไปยังชั้นเอาต์พุตใช้การปรับตามหลักการเรียนรู้แบบกึ่งมีผู้สอน (Semi-supervised learning)

โดยงานวิจัยนี้คาดว่าจะสามารถให้ความเชื่อมั่นในการทำนายที่ดีในกรณีที่ข้อมูลที่มีอยู่ไม่สมบูรณ์



รูปที่ 2.13 ลักษณะโครงสร้างของ Supervised GSOM [12]

ก่อนการเรียนรู้จะต้องทำการกำหนดค่าให้ตัวแปรควบคุมการโตของโครงข่าย Spread Factor (SF) ซึ่งมีค่าอยู่ระหว่างช่วง  $[0,1]$  โดย SF จะควบคุมค่า Growth Threshold (GT) ตามสมการที่ 2.24 ค่า GT คือค่าความผิดพลาดสะสมสูงสุดที่โหนดสามารถมีได้

$$GT = -D \times \ln(SF) \quad (2.24)$$

โดย  $D$  คือ จำนวนมิติข้อมูล  
ขั้นตอนการเรียนรู้

1. สุ่มเลือกข้อมูลเข้าสู่ชั้น GSOM
2. จำนวนหาโหนดที่ชนะคือโหนดที่มีค่าความเหมือนมากที่สุดโดยใช้ยูคลิเดียนคิสแทนซ์ (Euclidean distance) ในกรณีที่มีบางส่วนของข้อมูลหายไปเวลาคำนวณค่าความเหมือน เราจะไม่สนใจส่วนที่หาย
3. หาระดับการกระตุ้นของข้อมูลเข้ากับแต่ละโหนด  $c$  ในชั้น GSOM ( $o_c(I)$ ) สามารถคำนวณได้จากสมการที่ 2.25

$$o_c(I) = \exp\left(-\frac{\|I - w_c\|^2}{\sigma_c^2}\right) \quad (2.25)$$

โดย  $I$  คือ ข้อมูลเข้า  
 $w_c$  คือ ค่าน้ำหนักของนิวรอนโหนด  $c$   
 $\sigma_c$  คือ ขนาดความกว้างของ gaussian function

จากนั้นคำนวณหาคะแนนของแต่ละกลุ่ม  $m$  ในชั้นเอ้าท์พุท ( $O_m$ ) ตามสมการที่ 2.26

$$O_m = \sum_{c=1}^A w_{cm} o_c \quad (2.26)$$

โดย  $A$  คือ ขนาดของโครงข่าย GSOM ณ เวลานั้นๆ  
 $w_{cm}$  คือ ค่าน้ำหนักของนิวรอนโหนด  $c$  ที่เชื่อมไปเอ้าท์พุทโหนด  $m$

การทำนายกลุ่มของข้อมูลจะเลือกจากกลุ่มที่มีค่า  $O_m$  ที่มากที่สุด ในกรณีที่มีบางส่วนของข้อมูลหายไปเวลาคำนวณค่าคะแนนเราจะไม่สนใจส่วนที่หายไป

4. จำนวนหาค่าความผิดพลาดสะสม ณ เวลาใดๆ ของโหนดที่ชนะโดยใช้สมการที่ 2.27 โดย  $\Delta E_{win}$  คือ ค่าความผิดพลาดสะสมที่เปลี่ยนแปลงไป สามารถคำนวณโดยใช้สมการที่ 2.28

$$E_{win}(t+1) = E_{win}(t) + \Delta E_{win} \quad (2.27)$$

$$\Delta E_{win} = \begin{cases} 0 & \text{Case 1 : has label and correct prediction} \\ \sqrt{D}/2 & \text{Case 2 : has label and wrong prediction} \\ \|I - w_{win}\| & \text{Case 3 : missing label} \end{cases} \quad (2.28)$$

โดย  $E_{win}(t+1)$  คือ ค่าความผิดพลาดสะสมใหม่ที่ได้จากการปรับค่า  
 $t$  คือ รอบการทำงานปัจจุบัน

5. ถ้าค่าความผิดพลาดสะสมเกินกว่าค่า Growth Threshold จะทำการสร้างโหนดใหม่ในชั้น GSOM และสร้างการเชื่อมจากโหนดใหม่ดังแสดงในรูปที่ 2.13 โหนด  $n_5$  กับ  $n_6$  และให้ค่าน้ำหนักของเพื่อนบ้านกับโหนดใหม่

6. ปรับค่าน้ำหนักโหนดเพื่อนบ้านโดยปรับตามหลักการเรียนรู้แบบไม่มีผู้สอน

7. ปรับค่า  $\sigma_c$  ของโหนดในชั้น GSOM

8. ปรับลดค่าอัตราการเรียนรู้  $\alpha$  และรัศมีเพื่อนบ้านของโครงข่าย

9. ปรับค่าน้ำหนักที่เชื่อมต่อไปยังชั้นเอาต์พุตของโหนดที่ชนะโดยใช้สมการที่ 2.29

โดยค่าน้ำหนักที่เปลี่ยนแปลงไป  $\Delta w_{cm}$  สามารถคำนวณได้โดยใช้สมการที่ 2.30

$$w_{cm}^{new} = w_{cm}^{old} + \Delta w_{cm} \quad (2.29)$$

$$\Delta w_{cm} = \eta (s_m - o_m) D_c \quad (2.30)$$

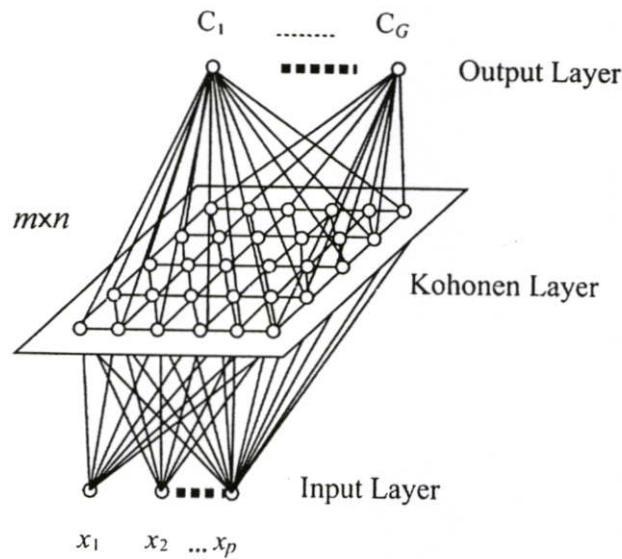
โดย  $\eta$  คือ ค่าอัตราการเรียนรู้

$s_m$  คือ ระดับการกระตุ้นจากกลุ่ม  $m$

งานวิจัยนี้ได้ทำการทดลองกับชุดข้อมูล Iris โดยเปรียบเทียบผลการทดลองกับการเรียนรู้แบบมีผู้สอนที่ใช้ข้อมูลที่สมบูรณ์ 100 เปอร์เซ็นต์ในการเรียนรู้ ส่วนงานวิจัยทำการเรียนรู้กับข้อมูล Iris ที่สร้างขึ้นจากข้อมูลที่มีความไม่สมบูรณ์ 20 เปอร์เซ็นต์โดยมีลักษณะเป็นข้อมูลที่มีค่าไม่ครบ (missing attribute values) และไม่มีกลุ่ม (missing data label) จากการทดลองพบว่างานวิจัยให้ค่าความถูกต้องน้อยกว่าเพียง 4 %

### 2.5.5 A Comparison between a Modified Counter Propagation and an Extended Self-Organizing Map in Remotely Sensed Data Classification [13]

งานวิจัยนี้ถูกคิดค้นโดย Markus Hagenbuchner และ Ah Chung Tsoi ในปี 2005 โดยงานวิจัยนี้เป็นการนำแนวคิดการเรียนรู้แบบมีผู้สอนมาใช้กับโครงข่ายประสาทเทียมแบบการปรับตัวเองภายในเพื่อให้ได้ผลการจัดหมวดหมู่ของข้อมูลที่ได้จากการสำรวจระยะไกลที่ดีที่สุดและสามารถนำไปประยุกต์ใช้กับข้อมูลชุดอื่นได้ ลักษณะของโครงข่ายมี 3 ชั้นดังแสดงในรูปที่ 2.14



รูปที่ 2.14 ลักษณะโครงสร้างของ Extended Self-Organizing Maps [13]

ในกระบวนการเรียนรู้การปรับค่าน้ำหนักจากชั้นอินพุตไปชั้น Kohonen จะปรับให้สอดคล้องกับอินพุต ส่วนการปรับค่าน้ำหนักจากชั้น Kohonen ไปชั้นเอาต์พุตจะปรับให้สอดคล้องกับอินพุตและค่าเป้าหมายที่คาดหวังของมัน

ขั้นตอนการเรียนรู้

1. สุ่มค่าน้ำหนักให้โครงข่ายค่าน้ำหนักของชั้นอินพุตกับชั้น Kohonen อยู่ในช่วง (0,1) และค่าน้ำหนักของชั้น Kohonen ไปชั้นเอาต์พุตกำหนดให้เป็น 0
2. กำหนดค่าพารามิเตอร์ต่างๆ ที่ใช้ในการเรียนรู้
3. สุ่มเลือกข้อมูลเข้าพร้อมค่าเป้าหมายที่คาดหวังของมันเข้าสู่โครงข่าย
4. หานิวรอนที่มีความเหมือนกับข้อมูลเข้ามากที่สุดโดยใช้สมการที่ 2.31 แล้วถ่ายทอดค่าของข้อมูลเข้าไปยังชั้นเอาต์พุต

$$winner = \min_{1 \leq l \leq L} \sum_{j=1}^P (x_j - w_{lj})^2 \quad (2.31)$$

โดย	$x$	คือ	ข้อมูลเข้าขนาด $P$ มิติ
	$j$	คือ	ลำดับของคุณลักษณะ
	$l$	คือ	ลำดับของนิวรอนในชั้น Kohonen
	$w_l$	คือ	ค่าน้ำหนักของนิวรอนที่ $l$

#### 5. ปรับค่าน้ำหนักของแต่ละชั้น

– การปรับค่าน้ำหนักจากชั้นอินพุตไปยังชั้น Kohonen ทำการปรับค่าน้ำหนักของโหนดที่ชนะและโหนดเพื่อนบ้านตามสมการที่ 2.32

$$w_{lj}(t+1) = w_{lj}(t) + \alpha_1(t) [x_j - w_{lj}(t)] \exp\left(-\frac{d_{bl}^2}{2r^2(t)}\right) \quad (j=1, 2, \dots, P) \quad (2.32)$$

โดย	$w_{lj}(t+1)$	คือ	ค่าน้ำหนักใหม่ที่ได้จากการปรับ
	$t$	คือ	รอบการทำงานปัจจุบัน
	$\alpha_1(t)$	คือ	อัตราการเรียนรู้ซึ่งสอดคล้องกับจำนวนรอบการทำงาน
	$r(t)$	คือ	ขนาดของเพื่อนบ้าน
	$d_{bl}$	คือ	ระยะทางระหว่างนิวรอน $b$ และ $l$ ในชั้น Kohonen

– การปรับค่าน้ำหนักจากชั้น Kohonen ไปยังชั้นเอาต์พุต ทำการปรับค่าน้ำหนักโหนดที่ชนะและเพื่อนบ้านตามสมการที่ 2.33

$$v_{ii}(t+1) = v_{ii}(t) + \alpha_2(t) [y_i - v_{ii}(t)] \exp\left(-\frac{d_{bl}^2}{2r^2(t)}\right) \quad (i=1, 2, \dots, G) \quad (2.33)$$

โดย	$v_{ii}(t+1)$	คือ	ค่าน้ำหนักใหม่ที่ได้จากการปรับ
	$y_i$	คือ	ค่าเป้าหมายที่คาดหวังไว้
	$\alpha_2(t)$	คือ	อัตราการเรียนรู้
	$r(t)$	คือ	ขนาดของเพื่อนบ้าน
	$G$	คือ	จำนวนกลุ่มทั้งหมดของข้อมูล

งานวิจัยนี้ได้ทำการทดลองกับชุดข้อมูลภาพถ่ายจากความเทียมของเมือง Atlanta (GA, US) โดยเปรียบเทียบผลการทดลองที่ได้กับแบบจำลองที่ชื่อว่า modified CPN จากการทดลองหลายชุดที่ใช้พารามิเตอร์ต่างกันไปพบว่า Extended Self-Organizing Map ให้ผลการทำนายโดยภาพรวมที่ดีกว่า modified CPN ในช่วง 1 ถึง 30%

## บทที่ 3

### Supervised SOM Using Fuzzy Membership Function

Supervised SOM Using Fuzzy Membership Function เป็นโครงข่ายประสาทเทียมที่ปรับปรุงมาจากโครงข่ายประสาทเทียมแบบการปรับตัวเองภายใน (Self Organizing Map) โดย Supervised SOM Using Fuzzy Membership Function ได้นำหลักการเรื่องค่าความเป็นสมาชิกของเซตของความคลุมเครือเข้ามาใช้ในการหาโหนดที่ชนะในขั้นตอนการแข่งขัน และในขั้นตอนการเรียนรู้ได้นำกลไกการเรียนรู้แบบมีผู้สอน (Supervised Learning) เข้ามาประยุกต์ใช้พร้อมทั้งมีกระบวนการที่สามารถเพิ่มลดจำนวนโหนดในชั้นเอาต์พุตได้ ทำให้สามารถนำหลักการทำงานของโครงข่ายประสาทเทียมแบบการปรับตัวเองภายในมาประยุกต์ใช้กับการจัดหมวดหมู่ข้อมูลได้

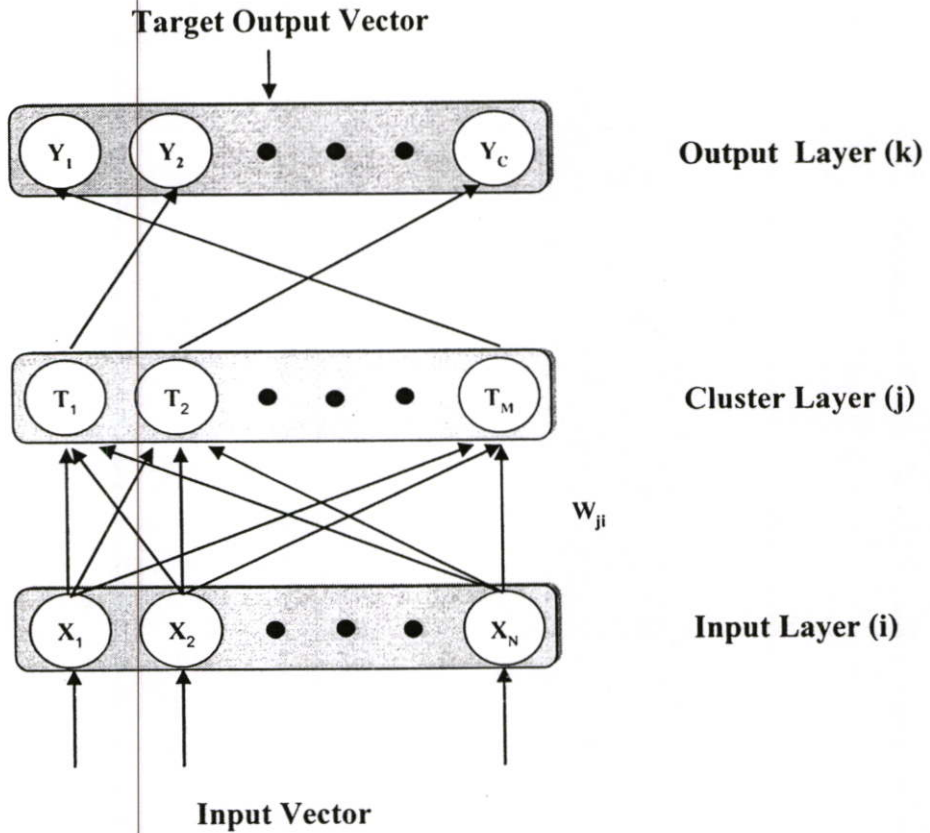
#### 3.1 สถาปัตยกรรมของ Supervised SOM Using Fuzzy Membership Function

Supervised SOM Using Fuzzy Membership Function เป็นโครงข่ายประสาทเทียมที่ประกอบด้วย 3 ระดับชั้น คือ ชั้นอินพุต (Input Layer) ชั้นคลัสเตอร์ (Cluster Layer) และชั้นเอาต์พุต (Output Layer) ซึ่งแต่ละชั้นมีลักษณะดังนี้

ชั้นที่ 1 ชั้นอินพุต (Input Layer) ทำหน้าที่รับอินพุตขนาด  $N$  มิติ

ชั้นที่ 2 ชั้นคลัสเตอร์ (Cluster Layer) แต่ละโหนดในชั้นเอาต์พุตจะเชื่อมต่อแบบทั่วถึงกับทุกโหนดในชั้นอินพุตผ่านทางค่าน้ำหนัก ( $w_{ji}$ ) โดยจำนวนโหนดในชั้นนี้สามารถเพิ่มและลดได้ในระหว่างกระบวนการเรียนรู้

ชั้นที่ 3 ชั้นเอาต์พุต (Output Layer) แต่ละโหนดในชั้นนี้จะเป็นตัวแทนที่แสดงถึงประเภทหรือชนิด (class) ของข้อมูลที่เราต้องการเรียนรู้ โดยจำนวนโหนดจะเท่ากับจำนวนประเภทหรือชนิดของข้อมูล เนื่องจากกระบวนการนี้เป็นการเรียนรู้แบบมีผู้สอน ดังนั้นคำตอบหรือชนิดของรูปแบบอินพุต (target output) จะถูกส่งเข้าไปในโครงข่ายพร้อมกับคุณลักษณะ (feature) ของอินพุต



รูปที่ 3.1 สถาปัตยกรรมของ Supervised SOM Using Fuzzy Membership Function

### 3.2 กระบวนการเรียนรู้ของ Supervised SOM Using Fuzzy Membership Function

#### 3.2.1 ขั้นตอนเตรียมการ

- ทำการกำหนดค่าพารามิเตอร์ต่างๆ ที่ใช้ในการเรียนรู้แบบจำลอง ดังนี้
  - 1) จำนวนรอบในการเรียนรู้
  - 2) จำนวนคลัสเตอร์เริ่มต้นของแต่ละคลาส
  - 3) ค่า threshold การเพิ่มคลัสเตอร์ ( $th$ )
  - 4) ค่า threshold การลบคลัสเตอร์ ( $thde$ )
  - 5) ค่าเริ่มต้นของการหาค่าอัตราการเรียนรู้ ( $u_0$ )
  - 6) ค่าเริ่มต้นของการหาค่าขนาดความกว้างของ Gaussian function ( $v_0$ )
- ทำการกำหนดค่าน้ำหนักของโครงข่ายโดยทำการสุ่มจากอินพุตของแต่ละคลาส
- ทำการคำนวณหาค่าความกว้างของ Gaussian function เริ่มต้นของแต่ละคลัสเตอร์ ( $\sigma_1, \sigma_r$ ) โดยหา(1/3)ของระยะทางที่สั้นที่สุดของ 2 คลัสเตอร์ใดๆ ที่อยู่คนละคลาสกัน

### 3.2.2 ขั้นตอนการเรียนรู้แบบจำลอง

โครงสร้างการทำงานแบ่งเป็น 3 ขั้นตอน ดังนี้

#### 3.2.2.1 ขั้นตอนการแข่งขัน

เมื่อชั้นอินพุต (Input layer) รับอินพุตเข้ามาจะทำการส่งอินพุตต่อไปยังชั้นคลัสเตอร์ (Cluster layer) เมื่อโครงข่ายถูกกระตุ้น โหนดทุกโหนดในชั้นคลัสเตอร์จะทำการแข่งขันกันเพื่อหาโหนดที่ชนะโดยทำการหาค่าความเป็นสมาชิก (membership) ของอินพุตกับโหนดแต่ละโหนด ซึ่งค่าความเป็นสมาชิกคำนวณได้จากสมการที่ 3.1 และ 3.2

$$T_j(X) = \frac{\sum_{i=1}^N \mu_j(x_i)}{N} \quad (3.1)$$

$$\mu_j(x_i) = \begin{cases} \exp\left[-\frac{(x_i - w_{ji})^2}{2\sigma_{ji}^2}\right] & \text{if } x_i < w_{ji} \\ 1 & \text{if } x_i = w_{ji} \\ \exp\left[-\frac{(x_i - w_{ji})^2}{2\sigma_{jr}^2}\right] & \text{if } x_i > w_{ji} \end{cases} \quad (3.2)$$

โดยที่ $T_j(X)$	คือ	ค่าความเป็นสมาชิกของอินพุตกับโหนดลำดับที่ $j$ ในชั้นคลัสเตอร์
$\mu_j(x_i)$	คือ	ระดับความเป็นสมาชิกของอินพุตเมื่อใช้ฟังก์ชันความเป็นสมาชิกแบบ Gaussian
$i$	คือ	ลำดับของคุณลักษณะ
$j$	คือ	ลำดับของโหนดในชั้นคลัสเตอร์
$N$	คือ	จำนวนคุณลักษณะ
$x_i$	คือ	อินพุตคุณลักษณะที่ $i$
$w_{ji}$	คือ	ค่าน้ำหนักของโหนดลำดับที่ $j$ ไปยังอินพุตคุณลักษณะที่ $i$
$\sigma_i$	คือ	ค่าความกว้างของ Gaussian ทางซ้าย
$\sigma_r$	คือ	ค่าความกว้างของ Gaussian ทางขวา

ทำการหาโหนดที่ชนะโดยโหนดที่ชนะ คือโหนดที่มีค่าความเป็นสมาชิกของอินพุตกับโหนดสูงที่สุด เมื่อเทียบกับค่าความเป็นสมาชิกของโหนดอื่นๆ ดังสมการที่ 3.3

$$winner = \arg_j \max(T_j(X)) \quad (3.3)$$

### 3.2.2.2 ขั้นตอนการปรับปรุงค่าน้ำหนัก แบ่งเป็น 2 กรณีดังนี้

กรณีที่ 1 ถ้าคลาสของโหนดที่ชนะ = คลาสของอินพุต

ทำการปรับค่าน้ำหนักของโหนดที่ชนะและโหนดที่อยู่คลาสเดียวกับโหนดที่ชนะ ณ เวลา  $t$  ใดๆ โดยใช้สมการที่ 3.4 ถึง 3.6

$$w_j(t+1) = w_j(t) + \Delta w_j(t) \quad (3.4)$$

$$\Delta w_j(t) = \alpha(N_i, t)(x_i(t) - w_j(t)) \quad (3.5)$$

$$\alpha(N_i, t) = \alpha(t) \exp(-\|r_i - r_m\| / \sigma^2(t)) \quad (3.6)$$

โดยที่	$w_j(t+1)$	คือ	ค่าน้ำหนักใหม่ที่ได้จากการปรับค่า
	$w_j(t)$	คือ	ค่าน้ำหนักเก่า
	$\Delta w_j(t)$	คือ	ค่าน้ำหนักที่เปลี่ยนแปลงตาม Neighborhood function
	$\alpha(N_i, t)$	คือ	อัตราการปรับ Neighborhood function ที่เวลา $t$ ใดๆ
	$r_m$	คือ	ตำแหน่งของ โหนดที่ชนะ
	$r_i$	คือ	ตำแหน่งของ โหนดรอบ โหนดที่ชนะ
	$t$	คือ	รอบการทำงานปัจจุบัน
	$\alpha(t)$	คือ	อัตราการเรียนรู้ที่เวลา $t$ ใดๆ หาได้จากสมการที่ 3.7
	$\sigma(t)$	คือ	ขนาดความกว้างของ Gaussian function ที่เวลา $t$ ใดๆ หาได้จากสมการที่ 3.8

$$\alpha(t) = \exp^{-(u_0 + 0.007(t-1))} \quad (3.7)$$

โดย  $u_0$  คือ ค่าเริ่มต้นของการหาค่าอัตราการเรียนรู้

$$\sigma(t) = \exp^{-(v_0 + 0.004(t-1))} \quad (3.8)$$

โดย  $v_0$  คือ ค่าเริ่มต้นของการหาค่าขนาดความกว้างของ Gaussian function

โดยอัตราการลดลงของขนาดความกว้างของ โหนดเพื่อนบ้าน ( $\sigma$ ) และอัตราการเรียนรู้ ( $\alpha$ ) จะลดลงสัมพันธ์กับจำนวนรอบที่เพิ่มขึ้นเมื่อเวลาผ่านไปตามฟังก์ชันที่กำหนดไว้ ส่วนโหนดอื่นๆ ที่อยู่คนละคลาสกับโหนดที่ชนะจะไม่ถูกปรับค่าน้ำหนัก

**กรณีที่ 2** ถ้าคลาสของโหนดที่ชนะ  $\neq$  คลาสของอินพุต ให้ทำตามขั้นตอนดังต่อไปนี้

1) หาโหนดใกล้เคียง (nearby cluster) ในที่นี้คือโหนดที่อยู่ใกล้ที่สุดและอยู่คลาสดียวกับอินพุต [8]

2) หาค่า  $memwin$  และ  $memnear$  ในที่นี้คือค่าความเป็นสมาชิกของอินพุตกับโหนดที่ชนะ และค่าความเป็นสมาชิกของอินพุตกับโหนดใกล้เคียง

3) ตรวจสอบการเพิ่มโหนดในชั้นคลัสเตอร์โดยพิจารณาจากเงื่อนไขต่อไปนี้

- ถ้า  $(memwin - memnear) \geq th$

ทำการสร้างโหนดในชั้นคลัสเตอร์เพิ่มโดยให้ค่าน้ำหนักของโหนดที่สร้างใหม่มีค่าเท่ากับอินพุตที่เข้ามา

- ถ้า  $(memwin - memnear) < th$

ทำการปรับค่าน้ำหนักในแต่ละคุณลักษณะของโหนดที่ชนะและโหนดใกล้เคียง โดยแต่ละคุณลักษณะของโหนดที่ชนะและโหนดใกล้เคียงจะถูกตรวจสอบค่าความเป็นสมาชิก จากการตรวจสอบแบ่งได้เป็น 4 กรณีดังนี้

**กรณีที่ 2.1** ถ้า  $memwin$  และ  $memnear \leq 0.01$  (เกิน  $3\sigma$ ) แล้วทำการปรับค่าน้ำหนักของโหนดทั้ง 2 ดังนี้

- ปรับค่าน้ำหนักของโหนดที่ชนะโดยใช้สมการที่ 3.9

$$w_{jk}^{w-new} = w_{jk}^{w-old} - \alpha(x_{ik} - w_{jk}^{w-old}) \quad (3.9)$$

- ปรับค่าน้ำหนักของโหนดใกล้เคียงโดยใช้สมการที่ 3.10

$$w_{jk}^{n-new} = w_{jk}^{n-old} + \alpha(x_{ik} - w_{jk}^{n-old}) \quad (3.10)$$

โดย	$\alpha$	คือ	อัตราการเรียนรู้
	$w_{jk}^{w-old}$	คือ	ค่าน้ำหนักเดิมของโหนดที่ชนะคุณลักษณะที่ k
	$w_{jk}^{w-new}$	คือ	ค่าน้ำหนักใหม่ของโหนดที่ชนะคุณลักษณะที่ k
	$w_{jk}^{n-old}$	คือ	ค่าน้ำหนักเดิมของโหนดใกล้เคียงคุณลักษณะที่ k
	$w_{jk}^{n-new}$	คือ	ค่าน้ำหนักใหม่ของโหนดใกล้เคียงคุณลักษณะที่ k

กรณีที่ 2.2 ถ้า ( $memwin$  และ  $memnear > 0.01$ ) และ ( $memwin > memnear$ ) แล้วทำการปรับค่า  $\sigma$  ของโหนดทั้ง 2 ดังนี้

- ปรับค่า  $\sigma$  ของโหนดที่ชนะโดยใช้สมการที่ 3.11

$$\sigma_{w\_new} = \sigma_{w\_old} - (0.05 \times ga) \quad (3.11)$$

- ปรับค่า  $\sigma$  ของโหนดใกล้เคียงโดยใช้สมการที่ 3.12

$$\sigma_{n\_new} = \sigma_{n\_old} + (0.05 \times ga) \quad (3.12)$$

โดย	$ga$	คือ	ค่า $\sigma$ เริ่มต้นของแต่ละโหนด
	$\sigma_{n\_new}$	คือ	ค่า $\sigma$ ใหม่ของโหนดใกล้เคียง
	$\sigma_{n\_old}$	คือ	ค่า $\sigma$ เก่าของโหนดใกล้เคียง
	$\sigma_{w\_new}$	คือ	ค่า $\sigma$ ใหม่ของโหนดที่ชนะ
	$\sigma_{w\_old}$	คือ	ค่า $\sigma$ เก่าของโหนดที่ชนะ

กรณีที่ 2.3 ถ้า ( $memwin > 0.01$ ) และ ( $memnear < 0.01$ ) แล้วทำการปรับค่าน้ำหนักของโหนดทั้ง 2 ดังนี้

- ปรับค่า  $\sigma$  ของโหนดที่ชนะโดยใช้สมการที่ 3.13

$$\sigma_{w\_new} = \sigma_{w\_old} - (0.05 \times ga) \quad (3.13)$$

- ปรับค่าน้ำหนักของโหนดใกล้เคียงโดยใช้สมการที่ 3.14

$$w_{jk}^{n\_new} = w_{jk}^{n\_old} + \alpha(x_{ik} - w_{jk}^{n\_old}) \quad (3.14)$$

กรณีที่ 2.4 ถ้า ( $memwin < memnear$ ) แล้วจะไม่มีค่าปรับค่าใดๆ

3.2.2.3 ขั้นตอนการวนซ้ำ ทำซ้ำตั้งแต่ขั้นตอนที่ 3.3.2.1 จนครบทุกอินพุตของข้อมูล  
ทดลอง (training data)

3.2.2.4 ขั้นตอนการลบ

ตรวจสอบการลบคลัสเตอร์โดยทำการคำนวณสัดส่วนค่า Entropy ของแต่ละคลัสเตอร์  $H_j$  โดยใช้สมการที่ 3.15 แล้วนำค่าที่ได้ไปเปรียบเทียบกับค่า threshold การลบคลัสเตอร์ ( $thde$ ) โดยถ้า  $H_j > thde$  แล้วคลัสเตอร์นั้นจะถูกลบ

$$H_j = \frac{N_j}{N} \times E_j \quad (3.15)$$

$$E_j = - \sum_i p_{ij} \log(p_{ij}) \quad (3.16)$$

โดย	$H_j$	คือ	สัดส่วนค่า Entropy ของคลัสเตอร์ $j$ โดยภาพรวม
	$N_j$	คือ	จำนวนอินพุตที่อยู่ในคลัสเตอร์ $j$
	$N$	คือ	จำนวนอินพุตทั้งหมด
	$E_j$	คือ	ค่า Entropy ของคลัสเตอร์ $j$ คำนวณได้จากสมการที่ 3.16
	$p_{ij}$	คือ	สัดส่วนค่าความเป็นสมาชิกของนิวรอนกลุ่ม (class) $i$ ในคลัสเตอร์ที่ $j$

3.2.2.5 ขั้นตอนการวนซ้ำ ทำซ้ำตั้งแต่ขั้นที่ 3.3.2.1 จนครบตามจำนวนรอบที่กำหนดไว้

3.2.3 ขั้นตอนการทดสอบแบบจำลอง

นำค่าน้ำหนักและค่าความกว้างของ Gaussian ที่ได้จากขั้นตอนการเรียนรู้แบบจำลองมาใช้ในการทดสอบความถูกต้องของแบบจำลองซึ่งมีขั้นตอนดังนี้

1. รับค่าอินพุตเข้ามาทีละ 1 รูปแบบ
2. ขั้นตอนการแข่งขัน

นิวรอนแต่ละตัวในชั้นคลัสเตอร์จะทำการแข่งขันกันโดยใช้ค่าความเป็นสมาชิกระหว่างอินพุตกับนิวรอน (Membership) เป็นเกณฑ์ในการแข่งขัน โดยค่าความเป็นสมาชิกสามารถคำนวณ

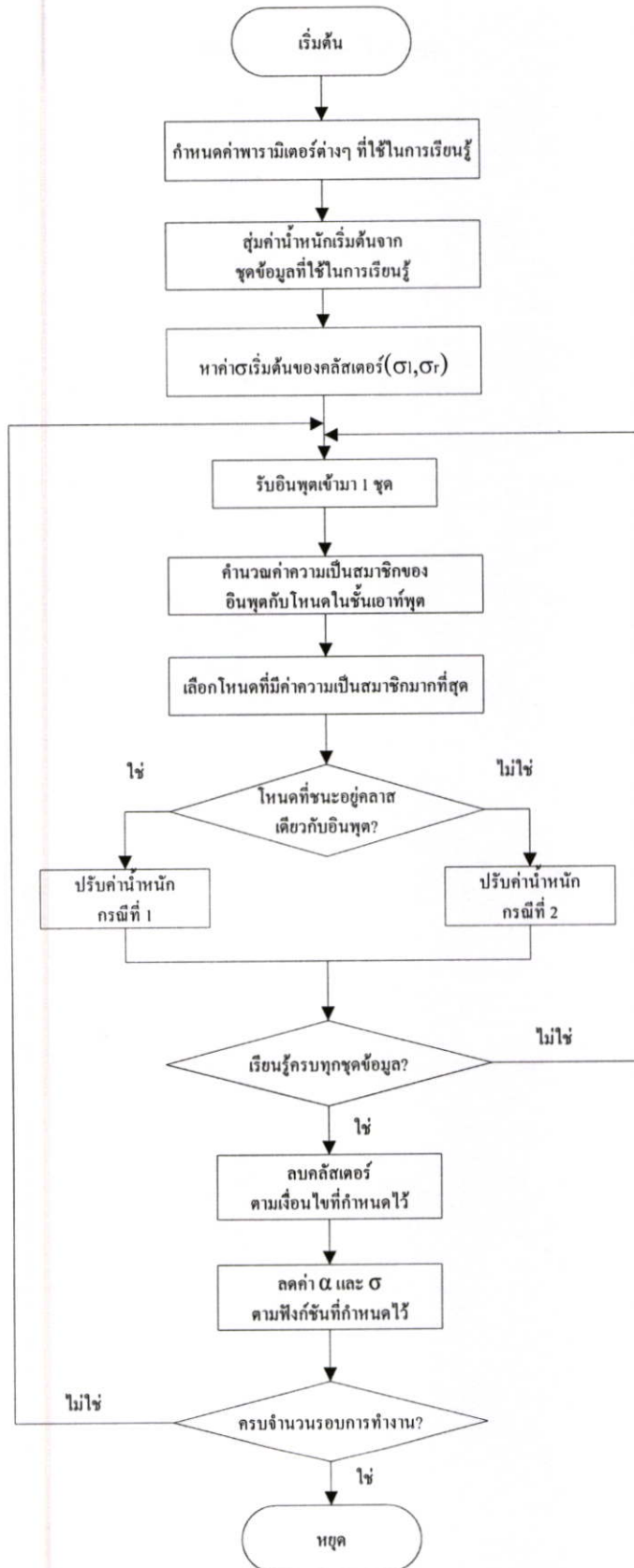
ได้โดยใช้สมการที่ 3.1 และ 3.2 นิเวรอนที่มีค่าความเป็นสมาชิกระหว่างอินพุตกับนิเวรอนมากที่สุดจะเป็นนิเวรอนที่ชนะดังสมการที่ 3.3

3. เปรียบเทียบคลาสของนิเวรอนที่ชนะกับคลาส ของอินพุตที่เข้ามาว่าตรงกันหรือไม่
4. วนกลับไปทำข้อ 3.3.3.1 จนครบทุกรูปแบบอินพุตของข้อมูลทดสอบ (testing data)
5. คำนวณค่าเปอร์เซ็นต์ความถูกต้องโดยใช้สมการที่ 3.17

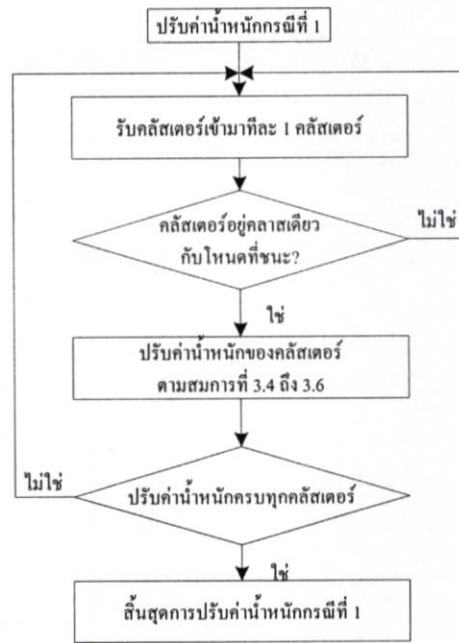
$$error = \frac{(100 \times wrong)}{total} \quad (3.17)$$

โดยที่ *error* คือ เปอร์เซนต์ความผิดพลาด  
*wrong* คือ จำนวนข้อมูลที่ทายผิด  
*total* คือ จำนวนข้อมูลทดสอบ (Testing data)

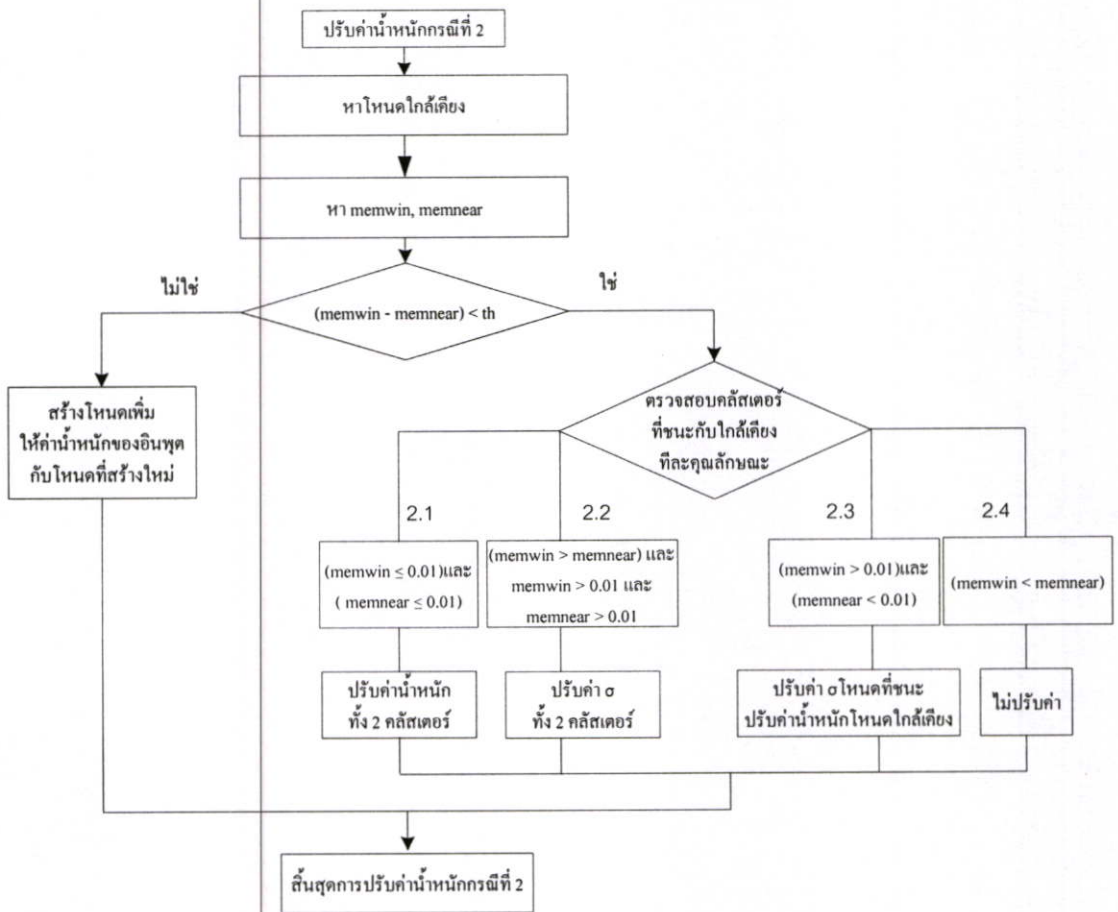
กระบวนการทำงานของอัลกอริธึม Supervised SOM Using Fuzzy Membership Function แสดงได้ดังรูปที่ 3.2



รูปที่ 3.2 โครงสร้างการทำงานของ Supervised SOM Using Fuzzy Membership Function



รูปที่ 3.3 โครงสร้างการทำงานของการทำงานการปรับค่าน้ำหนักกรณีที่ 1

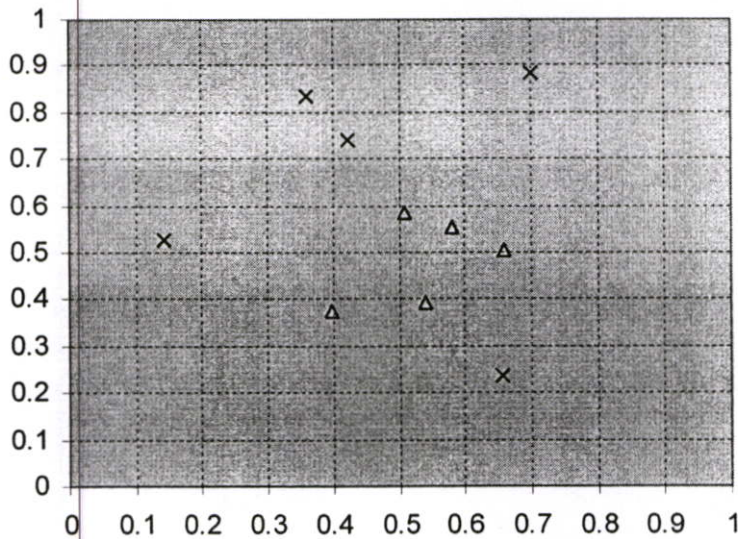


รูปที่ 3.4 โครงสร้างการทำงานของการทำงานการปรับค่าน้ำหนักกรณีที่ 2

### 3.3 ตัวอย่างการทำงาน

- ข้อมูลที่ใช้มีรูปแบบอินพุตจำนวน 10 รูปแบบ
- จำนวนคลาส 2 คลาส คือ คลาส 1 และ คลาส 2
- จำนวนคุณลักษณะ 2 คุณลักษณะ

1.	0.3981	0.375	คลาส 1
2.	0.6597	0.5046	คลาส 1
3.	0.5093	0.5856	คลาส 1
4.	0.5394	0.3935	คลาส 1
5.	0.581	0.5556	คลาส 1
6.	0.1412	0.5278	คลาส 2
7.	0.3588	0.8356	คลาส 2
8.	0.6991	0.8843	คลาส 2
9.	0.4236	0.7407	คลาส 2
10.	0.6551	0.2384	คลาส 2



รูปที่ 3.5 รูปแบบของอินพุตที่ใช้เป็นข้อมูลทดลอง โดยเครื่องหมาย  $\Delta$  แทนคลาส 1 และเครื่องหมาย  $\times$  แทนคลาส 2

1. กำหนดค่าพารามิเตอร์ต่างๆ ที่ใช้ดังนี้

จำนวนรอบในการเรียนรู้กำหนดให้ = 900 รอบ

จำนวนคลัสเตอร์เริ่มต้นของแต่ละคลาส = 1

ค่า threshold การเพิ่มคลัสเตอร์ ( $th$ ) = 0.25

ค่า threshold การลบคลัสเตอร์ ( $thde$ ) = 0.01

ค่าเริ่มต้นของการหาค่าอัตราการเรียนรู้ ( $u_0$ ) = 2

ค่าเริ่มต้นของการหาค่าขนาดความกว้างของ Gaussian function ( $v_0$ ) = 2.3

## 2. ทำการสุ่มค่าน้ำหนักเริ่มต้นดังนี้

1. 0.5093      0.5856    คลาส 1

2. 0.4236      0.7407    คลาส 2

## 3. ทำการคำนวณหาค่าความกว้างของ Gaussian function เริ่มต้นของแต่ละคลัสเตอร์ ( $\sigma_l, \sigma_r$ ) = 0.0591

## 4. ขั้นตอนการเรียนรู้

ทำการคำนวณหาค่าความเป็นสมาชิกของอินพุตกับคลัสเตอร์แต่ละคลัสเตอร์โดยใช้สมการที่ 3.1 และ 3.2

รูปแบบอินพุตที่ 1      0.3981 0.375    คลาส 1

$$\eta_{11} = \frac{\left( \exp\left[ \frac{-1}{2} \left( \frac{0.3981 - 0.5093}{0.0591} \right)^2 \right] + \exp\left[ \frac{-1}{2} \left( \frac{0.375 - 0.5856}{0.0591} \right)^2 \right] \right)}{2}$$

$$\eta_{11} = (0.17 + 0.0017)/2 = 0.0859$$

$$\eta_{12} = \frac{\left( \exp\left[ \frac{-1}{2} \left( \frac{0.3981 - 0.4236}{0.0591} \right)^2 \right] + \exp\left[ \frac{-1}{2} \left( \frac{0.375 - 0.7407}{0.0591} \right)^2 \right] \right)}{2}$$

$$\eta_{12} = (0.911 + 0)/2 = 0.455$$

คลัสเตอร์ที่ชนะคือ คลัสเตอร์ที่ 2 ซึ่งอยู่ในคลาส 2 ดังนั้น ทำนายผิด จึงทำการปรับค่าน้ำหนักโดยใช้กรณีที่ 2 ตามขั้นตอนดังนี้

1) หาโหนดใกล้เคียง (nearby cluster) ได้เป็น โหนดที่ 1

2) หาค่า  $memwin$  และ  $memnear$

$$memwin = 0.455$$

$$memnear = 0.0859$$

## 3) ตรวจสอบการเพิ่มโหนดในชั้นเอาต์พุตโดยพิจารณาจาก

$(memwin - memnear) = 0.3691$  ซึ่งมากกว่า 0.25 ดังนั้นทำการสร้างโหนดในชั้นเอาต์พุตเพิ่มจะได้

1. 0.5093 0.5856 คลาส 1
2. 0.4236 0.7407 คลาส 2
3. 0.3981 0.375 คลาส 1

รูปแบบอินพุตที่ 2                      0.6597 0.5046                      คลาส 1

$$\eta_{21} = (0.0391 + 0.3905) / 2 = \mathbf{0.2148}$$

$$\eta_{22} = (0.00034 + 0.00034) / 2 = 0.0003$$

$$\eta_{23} = (0.00005 + 0.0901) / 2 = 0.0451$$

คลัสเตอร์ที่ชนะคือ คลัสเตอร์ที่ 1 ซึ่งอยู่ในคลาส 1 ดังนั้น ทำนายถูก จึงทำการปรับค่าน้ำหนักโดยใช้กรณีที่ 1 ทำการปรับค่าน้ำหนักของโหนดที่ชนะและโหนดที่อยู่คลาสเดียวกับโหนดที่ชนะ ณ เวลา  $t$  ใดๆ โดยใช้สมการที่ 3.4 ถึง 3.6 จะได้

1. 0.5865 0.5440 คลาส 1
2. 0.4236 0.7407 คลาส 2
3. 0.3981 0.3750 คลาส 1

รูปแบบอินพุตที่ 3                      0.5093 0.5856                      คลาส 1

$$\eta_{31} = (0.4255 + 0.7805) / 2 = \mathbf{0.6030}$$

$$\eta_{32} = (0.3491 + 0.0318) / 2 = 0.1904$$

$$\eta_{33} = (0.17 + 0.0717) / 2 = 0.0859$$

คลัสเตอร์ที่ชนะคือ คลัสเตอร์ที่ 1 ซึ่งอยู่ในคลาส 1 ดังนั้น ทำนายถูก จึงทำการปรับค่าน้ำหนักโดยใช้กรณีที่ 1 ทำการปรับค่าน้ำหนักของโหนดที่ชนะและโหนดที่อยู่คลาสเดียวกับโหนดที่ชนะ ณ เวลา  $t$  ใดๆ โดยใช้สมการที่ 3.4 ถึง 3.6 จะได้

1. 0.5468 0.5654 คลาส 1
2. 0.4236 0.7407 คลาส 2
3. 0.3981 0.3750 คลาส 1

รูปแบบอินพุตที่ 4                      0.5394 0.3935                      คลาส 1

$$\eta_{41} = (0.992 + 0.0145) / 2 = 0.5033$$

$$\eta_{42} = (0.1464 + 0) / 2 = 0.0732$$

$$\eta_{43} = (0.057 + 0.952) / 2 = \mathbf{0.5047}$$

คลัสเตอร์ที่ชนะคือ คลัสเตอร์ที่ 3 ซึ่งอยู่ในคลาส 1 ดังนั้น ทำนายถูก จึงทำการปรับค่าน้ำหนักโดยใช้กรณีที่ 1 ทำการปรับค่าน้ำหนักของโหนดที่ชนะและโหนดที่อยู่คลาสเดียวกับโหนดที่ชนะ ณ เวลา  $t$  ใดๆ โดยใช้สมการที่ 3.4 ถึง 3.6 จะได้

1. 0.5468 0.5654 คลาส 1
2. 0.4236 0.7407 คลาส 2
3. 0.4706 0.3845 คลาส 1

รูปแบบอินพุตที่ 5                      0.581    0.5556                      คลาส 1

$$\eta_{51} = (0.8463 + 0.9864) / 2 = \mathbf{0.9164}$$

$$\eta_{52} = (0.0287 + 0.0074) / 2 = 0.018$$

$$\eta_{53} = (0.1746 + 0.0151) / 2 = 0.0948$$

คลัสเตอร์ที่ชนะคือ คลัสเตอร์ที่ 1 ซึ่งอยู่ในคลาส 1 ดังนั้น ทำนายถูก จึงทำการปรับค่าน้ำหนักโดยใช้กรณีที่ 1 ทำการปรับค่าน้ำหนักของโหนดที่ชนะและโหนดที่อยู่คลาสเดียวกับโหนดที่ชนะ ณ เวลา  $t$  ใดๆ โดยใช้สมการที่ 3.4 ถึง 3.6 จะได้

1. 0.5644 0.5603 คลาส 1
2. 0.4236 0.7407 คลาส 2
3. 0.3981 0.3750 คลาส 1

รูปแบบอินพุตที่ 6                      0.1412    0.5278                      คลาส 2

$$\eta_{61} = (0 + 0.8591) / 2 = \mathbf{0.4296}$$

$$\eta_{62} = (0 + 0.0015) / 2 = 0.0008$$

$$\eta_{63} = (0 + 0.0527) / 2 = 0.0264$$

คลัสเตอร์ที่ชนะคือ คลัสเตอร์ที่ 1 ซึ่งอยู่ในคลาส 1 ดังนั้น ทำนายผิด จึงทำการปรับค่าน้ำหนักโดยใช้กรณีที่ 2 ตามขั้นตอนดังนี้

- 1) หาโหนดใกล้เคียง (nearby cluster) ได้เป็นโหนดที่ 2
- 2) หาค่า *memwin* และ *memnear*

$$memwin = 0.4296$$

$$memnear = 0.0008$$

3) ตรวจสอบการเพิ่มโหนดในชั้นเอาต์พุตโดยพิจารณาจาก

$(memwin - memnear) = 0.4288$  ซึ่งมากกว่า 0.25 ดังนั้นทำการสร้างโหนดในชั้นเอาต์พุตเพิ่มจะได้

- 1) 0.5644 0.5603 คลาส 1
- 2) 0.4236 0.7407 คลาส 2
- 3) 0.3981 0.3750 คลาส 1
- 4) 0.1412 0.5278 คลาส 2

รูปแบบอินพุตที่ 7                      0.3588 0.8356                      คลาส 2

$$\begin{aligned} \eta_{71} &= (0.0023 + 0) / 2 &= 0.0012 \\ \eta_{72} &= (0.5478 + 0.2751) / 2 &= \mathbf{0.4115} \\ \eta_{73} &= (0.1665 + 0) / 2 &= 0.0833 \\ \eta_{74} &= (0.0011 + 0) / 2 &= 0.0006 \end{aligned}$$

คลัสเตอร์ที่ชนะคือ คลัสเตอร์ที่ 2 ซึ่งอยู่ในคลาส 2 ดังนั้น ทำนายถูก จึงทำการปรับค่าน้ำหนักโดยใช้กรณีที่ 1 ทำการปรับค่าน้ำหนักของโหนดที่ชนะและโหนดที่อยู่คลาสเดียวกับโหนดที่ชนะ ณ เวลา  $t$  ใดๆ โดยใช้สมการที่ 3.4 ถึง 3.6 จะได้

- 1) 0.5644 0.5603 คลาส 1
- 2) 0.3903 0.7894 คลาส 2
- 3) 0.3981 0.3750 คลาส 1
- 4) 0.1412 0.5278 คลาส 2

รูปแบบอินพุตที่ 8                      0.6991 0.8843                      คลาส 2

$$\begin{aligned} \eta_{81} &= (0.0742 + 0) / 2 &= 0.0012 \\ \eta_{82} &= (0 + 0.2753) / 2 &= \mathbf{0.4115} \\ \eta_{83} &= (0 + 0) / 2 &= 0 \\ \eta_{84} &= (0 + 0) / 2 &= 0 \end{aligned}$$

คลัสเตอร์ที่ชนะคือ คลัสเตอร์ที่ 2 ซึ่งอยู่ในคลาส 2 ดังนั้น ทำนายถูก จึงทำการปรับค่าน้ำหนักโดยใช้กรณีที่ 1 ทำการปรับค่าน้ำหนักของโหนดที่ชนะและโหนดที่อยู่คลาสเดียวกับโหนดที่ชนะ ณ เวลา  $t$  ใดๆ โดยใช้สมการที่ 3.4 ถึง 3.6 จะได้

- 1) 0.5644 0.5603 คลาส 1
- 2) 0.5488 0.8381 คลาส 2

- 3) 0.3981 0.3750 คลาส 1  
 4) 0.1412 0.5278 คลาส 2

รูปแบบอินพุตที่ 9	0.4236 0.7407	คลาส 2
$\eta_{91} =$	$(0.0584 + 0.0095) / 2$	$= 0.0339$
$\eta_{92} =$	$(0.1056 + 0.2565) / 2$	$= 0.181$
$\eta_{93} =$	$(0.7282 + 0) / 2$	$= 0.3641$
$\eta_{94} =$	$(0 + 0.0015) / 2$	$= 0.0008$

คลัสเตอร์ที่ชนะคือ คลัสเตอร์ที่ 3 ซึ่งอยู่ในคลาส 1 ดังนั้น ทำนายผิด จึงทำการปรับค่าน้ำหนักโดยใช้กรณีที่ 2 ตามขั้นตอนดังนี้

- 1) หาโหนดใกล้เคียง (nearby cluster) ได้เป็นโหนดที่ 2
- 2) หาค่า  $memwin$  และ  $memnear$

$$memwin = 0.3641$$

$$memnear = 0.181$$

- 3) ตรวจสอบการเพิ่มโหนดในชั้นเอาท์พุทโดยพิจารณาจาก

$(memwin - memnear) = 0.1831$  ซึ่งน้อยกว่า 0.25 ดังนั้นทำการตรวจสอบโหนดที่ชนะและโหนดใกล้เคียงทีละคุณลักษณะ โดยแต่ละคุณลักษณะจะถูกตรวจสอบค่าความเป็นสมาชิกไปที่ละคู่เพื่อนำไปใช้ในการปรับค่าน้ำหนักต่อไป

คุณลักษณะที่ 1

$$memwin_{93}^1 = 0.72819$$

$$memnear_{93}^1 = 0.10556$$

$(memwin$  และ  $memnear > 0.01)$  และ  $(memwin > memnear)$  ดังนั้นเข้าสู่กรณีที่ 2.2 ทำการปรับค่า  $\sigma$  ของโหนดทั้ง 2 ดังนี้

- ปรับค่า  $\sigma$  ของโหนดที่ชนะโดยใช้สมการที่ 3.11 ได้ค่า  $\sigma_{w\_new} = 0.056114$
- ปรับค่า  $\sigma$  ของโหนดใกล้เคียงโดยใช้สมการที่ 3.12 ได้ค่า  $\sigma_{n\_new} = 0.062021$

คุณลักษณะที่ 2

$$memwin_{93}^2 = 0$$

$$memnear_{93}^2 = 0.25653$$

$(memwin < memnear)$  ดังนั้นเข้าสู่กรณีที่ 2.4 ไม่มีการปรับค่าใดๆ

เมื่อตรวจสอบครบทุกคุณลักษณะจะได้

- 1) 0.5644 0.5603 คลาส 1
- 2) 0.5488 0.8381 คลาส 2
- 3) 0.4706 0.3845 คลาส 1
- 4) 0.1412 0.5278 คลาส 2

<u>รูปแบบอินพุตที่ 10</u>	0.6551	0.2384	คลาส2
$\eta_{10,1} =$	$(0.0376 + 0) / 2$		$= 0.1538$
$\eta_{10,2} =$	$(0.1984 + 0) / 2$		$= 0.0992$
$\eta_{10,3} =$	$(0.0076 + 0.0469) / 2$		$= 0.0273$
$\eta_{10,4} =$	$(0 + 0) / 2$		$= 0$

คลัสเตอร์ที่ชนะคือ คลัสเตอร์ที่ 1 ซึ่งอยู่ในคลาส 1 ดังนั้น ทำนายผิด จึงทำการปรับค่าน้ำหนักโดยใช้กรณีที่ 2 ตามขั้นตอนดังนี้

- 1) หาโหนดใกล้เคียง (nearby cluster) ได้เป็นโหนดที่ 2
- 2) หาค่า *memwin* และ *memnear*

$$memwin = 0.1538$$

$$memnear = 0.0992$$

- 3) ตรวจสอบการเพิ่มโหนดในชั้นเอาท์พุตโดยพิจารณาจาก

$(memwin - memnear) = 0.054$  ซึ่งน้อยกว่า 0.25 ดังนั้นทำการตรวจสอบโหนดที่ชนะและโหนดใกล้เคียงทีละคุณลักษณะ โดยแต่ละคุณลักษณะจะถูกตรวจสอบค่าความเป็นสมาชิกไปที่ละคู่เพื่อนำไปใช้ในการปรับค่าน้ำหนักต่อไป

คุณลักษณะที่ 1

$$memwin_{10,1}^1 = 0.30756$$

$$memnear_{10,2}^1 = 0.19838$$

$(memwin$  และ  $memnear > 0.01)$  และ  $(memwin > memnear)$  ดังนั้นเข้าสู่กรณีที่ 2.2 ทำการปรับค่า  $\sigma$  ของโหนดทั้ง 2 ดังนี้

-ปรับค่า  $\sigma$  ของโหนดที่ชนะโดยใช้สมการที่ 3.11 ได้ค่า  $\sigma_{w\_new} = 0.056114$

-ปรับค่า  $\sigma$  ของโหนดใกล้เคียงโดยใช้สมการที่ 3.12 ได้ค่า  $\sigma_{n\_new} = 0.062021$

คุณลักษณะที่ 2

$$memwin_{10,2}^2 = 0$$

$$memnear_{10,2}^2 = 0$$

$memwin$  และ  $memnear \leq 0.01$  (เกิน  $3\sigma$ ) ดังนั้นเข้าสู่กรณีที่ 2.1 ทำการปรับค่าน้ำหนักของ โหนดทั้ง 2 ดังนี้

-ปรับค่าน้ำหนักของโหนดที่ชนะโดยใช้สมการที่ 3.9

$$w_{jk}^{w-new} = 0.59255$$

-ปรับค่าน้ำหนักของโหนดใกล้เคียงโดยใช้สมการที่ 3.10

$$w_{jk}^{n-new} = 0.77816$$

เมื่อทำงานครบ 10 ชุดข้อมูลจะได้

- 1) 0.5644 0.5925 คลาส 1
- 2) 0.5488 0.7781 คลาส 2
- 3) 0.4706 0.3845 คลาส 1
- 4) 0.1412 0.5278 คลาส 2

ตารางที่ 3.1 ค่าความกว้างของ  $\sigma$  ทางซ้ายของแต่ละคลัสเตอร์ในแต่ละคุณลักษณะ

คลัสเตอร์	คุณลักษณะที่1	คุณลักษณะที่2
1	0.0591	0.0591
2	0.0620	0.0591
3	0.0561	0.0591
4	0.0591	0.0591

ตารางที่ 3.2 ค่าความกว้างของ  $\sigma$  ทางขวาของแต่ละคลัสเตอร์ในแต่ละคุณลักษณะ

คลัสเตอร์	คุณลักษณะที่1	คุณลักษณะที่2
1	0.0561	0.0591
2	0.0620	0.0591
3	0.0591	0.0591
4	0.0591	0.0591

ตารางที่ 3.3 ชุดของอินพุตกับคลัสเตอร์ที่ชนะ

อินพุต	คลัสเตอร์	ค่าความเป็นสมาชิก
1	Cluster 3	0.7106
2	Cluster 4	0.4629
3	Cluster 1	0.8202
4	Cluster 3	0.748
5	Cluster 1	0.89
6	Cluster 4	1
7	Cluster 2	0.3159
8	Cluster 2	0.1258
9	Cluster 2	0.4744
10	Cluster 1	0.1354

ตารางที่ 3.4 ค่าความเป็นสมาชิกของแต่ละคลัสเตอร์

คลัสเตอร์	คลาส 1	คลาส 2	ค่าความเป็นสมาชิกรวม
1	0.8202+0.89	0.1354	1.8456
2	-	0.3159+0.1258+0.4744	0.9161
3	0.7106+0.748	-	1.4586
4	0.4629	1	1.4629

จากสมการที่ 3.16 สามารถหาค่า Entropy ของแต่ละคลัสเตอร์ได้ดังนี้

$$E_1 = (-1) \left[ \frac{1.7102}{1.8456} \log \left( \frac{1.7102}{1.8456} \right) + \frac{0.1354}{1.8456} \log \left( \frac{0.1354}{1.8456} \right) \right]$$

$$E_1 = (-1) [(0.9266) \log(0.9266) + (0.0734) \log(0.0734)] = 0.3784$$

$$E_2 = (-1) \left[ 0 + \frac{0.9161}{0.9161} \log \left( \frac{0.9161}{0.9161} \right) \right] = 0$$

$$E_3 = (-1) \left[ \frac{1.4586}{1.4586} \log \left( \frac{1.4586}{1.4586} \right) + 0 \right] = 0$$

$$E_4 = (-1) \left[ \frac{0.4629}{1.4629} \log \left( \frac{0.4629}{1.4629} \right) + \frac{1}{1.4629} \log \left( \frac{1}{1.4629} \right) \right]$$

$$E_4 = (-1) [(0.3164) \log(0.3164) + (0.6836) \log(0.6836)] = 0.9004$$

จากสมการที่ 3.15 ได้ค่า Entropy ของแต่ละคลัสเตอร์โดยภาพรวม  $H_j$  ได้ดังนี้

$$H_1 = \left( \left( \frac{3}{10} \right) \times 0.3784 \right) = 0.1135$$

$$H_2 = \left( \left( \frac{3}{10} \right) \times 0 \right) = 0$$

$$H_3 = \left( \left( \frac{2}{10} \right) \times 0 \right) = 0$$

$$H_4 = \left( \left( \frac{2}{10} \right) \times 0.9004 \right) = 0.1801$$

เมื่อทำครบ 1 รอบการทำงานจะได้ค่าน้ำหนักดังนี้

- 1) 0.5488      0.7781    คลาส 2
- 2) 0.4706      0.3845    คลาส 1

ทำซ้ำ จนครบตามจำนวน 900 รอบที่กำหนดไว้จะได้ชุดของค่าน้ำหนักดังนี้

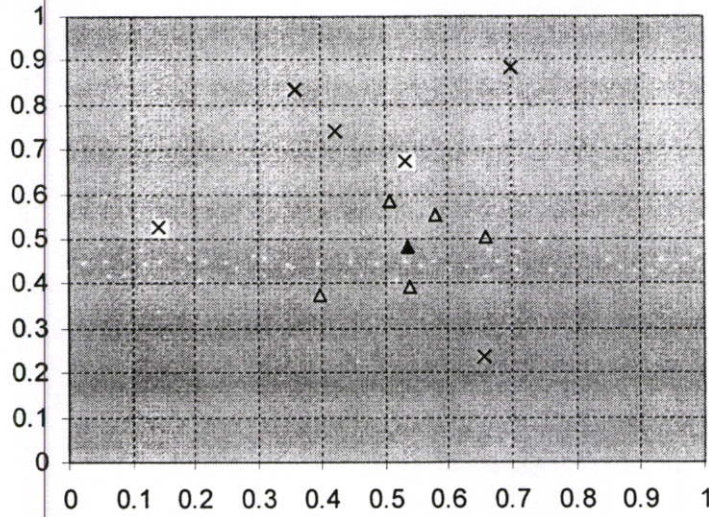
- 1) 0.5344      0.6741    คลาส 2
- 2) 0.1412      0.5278    คลาส 2
- 3) 0.5376      0.4830    คลาส 1

ตารางที่ 3.5 ค่าความกว้างของ  $\sigma$  ทางซ้ายของแต่ละคลัสเตอร์ในแต่ละคุณลักษณะ

คลัสเตอร์	คุณลักษณะที่ 1	คุณลักษณะที่ 2
1	0.0541	0.0291
2	0.0591	0.0291
3	0.0591	0.0891

ตารางที่ 3.6 แสดงค่าความกว้างของ  $\sigma$  ทางขวาของแต่ละคลัสเตอร์ในแต่ละคุณลักษณะ

คลัสเตอร์	คุณลักษณะที่ 1	คุณลักษณะที่ 2
1	0.0641	0.0591
2	0.0591	0.0591
3	0.0491	0.0591



รูปที่ 3.6 แสดงค่าน้ำหนักที่ได้หลังจากการเรียนรู้โดยเครื่องหมาย  $\Delta$  แทนคลาส 1 และเครื่องหมาย  $\times$  แทนคลาส 2

### 3.4 การเปรียบเทียบหลักการทำงานของแบบจำลอง

จากการศึกษาการพัฒนาโครงข่ายประสาทเทียมแบบการปรับตัวเองภายในเพื่อที่จะนำมาใช้จัดหมวดหมู่ข้อมูลพบว่าในแต่ละงานวิจัยมีแนวทางในการพัฒนาที่มีแนวคิดและการใช้เทคนิคที่หลากหลาย ซึ่งก่อนทำการเปรียบเทียบหลักการทำงานของแบบจำลองที่นำเสนอกับแบบจำลองต่างๆ ที่ได้ศึกษามาในบทที่ 2 จะทำการสรุปหลักการทำงานของแต่ละแบบจำลองก่อนคร่าวๆ ได้ดังนี้

1. งานวิจัยของ Markus P.N. Suganthan เรื่อง Hierarchical Overlapped SOM's for Pattern Classification [9] ใน 1 รอบการเรียนรู้ของ Hierarchical Overlapped SOM ประกอบด้วย การเรียนรู้ 3 ช่วง ในแต่ละช่วงมีการปรับค่าน้ำหนักตามรูปแบบการเรียนรู้ที่เลือกใช้ โดยช่วงแรกใช้ การเรียนรู้แบบไม่มีผู้สอนของ SOM เพื่อทำการจัดกลุ่มข้อมูลอย่างคร่าวๆ ในช่วงที่สองใช้ การเรียนรู้แบบไม่มีผู้สอนของ LVQ2 เพื่อทำการเรียนรู้ที่ละกลุ่มย่อยๆ จนครบ แล้วทำการ prune กลุ่ม ที่ได้จากการเรียนรู้ที่ผ่านมา 2 ช่วง จากนั้นช่วงที่ 3 ใช้การเรียนรู้แบบมีผู้สอนของ LVQ2 เพื่อทำการ ปรับค่าน้ำหนักหลังจากการ prune ให้เหมาะสมอีกครั้ง
2. งานวิจัยของ Markus Victor-Emil NEAGOE และ Armand-Dragos ROPOT เรื่อง Concurrent Self-Organizing Maps for Pattern Classification [10] ใน 1 รอบการเรียนรู้ของ CSOM ประกอบด้วยการเรียนรู้แบบไม่มีผู้สอนของ SOM ที่มีจำนวนเท่ากับจำนวนคลาสของข้อมูล โดยแต่ละโครงข่ายทำการเรียนรู้ปรับค่าน้ำหนักของโครงข่ายให้เหมาะสมกับข้อมูลที่ถูกแบ่งกลุ่มไว้ก่อนหน้าแล้ว เมื่อผ่านการเรียนรู้จะได้โครงข่ายที่ดีที่สุดสำหรับข้อมูลแต่ละกลุ่ม
3. งานวิจัยของ Markus Hagenbuchner และ Ah Chung Tsoi เรื่อง A supervised training algorithm for self-organizing maps for structure [11] จะใช้การเรียนรู้แบบไม่มีผู้สอนและมีผู้สอน ร่วมกัน โดยกรณีที่ข้อมูลเข้ามาเรียนรู้ไม่สมบูรณ์หรือไม่มีกลุ่มจะใช้การเรียนรู้แบบไม่มีผู้สอน ใน กระบวนการเรียนรู้แบบมีผู้สอนส่วนของการปรับค่าน้ำหนักมีการใช้ rejection term เพื่อปฏิเสธ กรณีโหนดที่ถูกเลือก (โหนดชนะ) มีกลุ่มต่างจากข้อมูลที่เข้ามาเรียนรู้ ลักษณะของข้อมูลที่นำมาใช้ ในการเรียนรู้จะมีการใช้ข้อมูลอื่นๆ เพิ่มขึ้นตามที่ผู้ใช้ต้องการได้ เช่น ความสัมพันธ์ระหว่างข้อมูล ซึ่งข้อมูลจะถูกนำมาเชื่อมต่อกับข้อมูลเข้าที่มีอยู่ก่อนแล้วจึงนำไปใช้เรียนรู้
4. งานวิจัยของ Arthur Hsu และ Saman K.Halgamuge เรื่อง Semi-supervised Learning of Dynamic Self-Organising [12] ขั้นตอนการเรียนรู้ประกอบด้วยขั้นตอนการแข่งขันซึ่งใช้ยูคลิดีเนียนดิสแทนซ์ (Euclidean distance) เป็นเกณฑ์การตัดสินใจโหนดชนะแล้วจึงทำนายกลุ่มของ ข้อมูล โดยการพิจารณาจากคะแนนของแต่ละกลุ่มในชั้นเข้าที่พุด จากนั้นจะพิจารณาการสร้าง โหนดใหม่จากค่าความผิดพลาดสะสมของโหนดที่ชนะว่ามีค่าค่าความผิดพลาดสะสมเกินกว่าค่า Growth Threshold ที่กำหนดไว้หรือไม่ถ้าเกินจะทำการสร้างโหนดใหม่ในชั้น GSOM ในขั้นตอน

การปรับค่าน้ำหนัก โหนดเพื่อนบ้านจะปรับตามหลักการเรียนรู้แบบไม่มีผู้สอนของ SOM ส่วน โหนดที่ชนะปรับโดยพิจารณาจากระดับการกระตุ้นและกลุ่มที่เลือก

5. งานวิจัยของ Markus Hagenbuchner และ Ah Chung Tsoi เรื่อง A Comparison between a Modified Counter Propagation and an Extended Self-Organizing Map in Remotely Sensed Data Classification [13] การเรียนรู้ของ Extended SOM ในขั้นตอนการแข่งขันใช้ยูคลิดีเนียนคิสแทนซ์ (Euclidean distance) เป็นเกณฑ์การตัดสินใจโหนดชนะส่วนขั้นตอนการปรับค่าน้ำหนักพิจารณาเป็น 2 กรณี คือการปรับค่าน้ำหนักจากชั้นอินพุตไปชั้น Kohonen จะปรับให้สอดคล้องกับอินพุต ส่วนการปรับค่าน้ำหนักจากชั้น Kohonen ไปชั้นเอาต์พุตจะปรับให้สอดคล้องกับอินพุตและค่าเป้าหมายที่คาดหวังของมัน

ในส่วนของงานวิจัยที่นำเสนอใน 1 รอบการเรียนรู้ประกอบด้วยขั้นตอนการแข่งขันโดยใช้ค่าความเป็นสมาชิก (membership) ของอินพุตกับโหนดเป็นเกณฑ์การตัดสินใจโหนดชนะ ส่วนขั้นตอนการปรับค่าน้ำหนักจะปรับตามหลักการเรียนรู้แบบมีผู้สอน โดยทำการพิจารณาจากกลุ่มของข้อมูล ในกรณีที่คลาสของโหนดที่ชนะไม่เหมือนกับคลาสของอินพุตก่อนการปรับค่าน้ำหนักจะมีการพิจารณาการเพิ่มโหนดในชั้นคลัสเตอร์โดยพิจารณาจากค่าความแตกต่างของค่าความเป็นสมาชิกของอินพุตกับโหนดที่ชนะกับค่าความเป็นสมาชิกของอินพุตกับโหนดใกล้เคียงเปรียบเทียบกับค่า threshold การเพิ่มคลัสเตอร์ที่กำหนดไว้ แล้วจึงทำการปรับค่าน้ำหนักตามแต่ละกรณี เมื่อทำการเรียนรู้ครบทุกอินพุตแล้วจึงตรวจสอบการลบคลัสเตอร์โดยเปรียบเทียบกับค่า Entropy ของแต่ละคลัสเตอร์กับค่า threshold การลบคลัสเตอร์ ถ้าค่า Entropy ของคลัสเตอร์สูงกว่าค่า threshold การลบคลัสเตอร์ก็ทำการลบคลัสเตอร์นั้นทิ้งไป

#### การเปรียบเทียบหลักการทำงานของแบบจำลอง

เมื่อพิจารณาจากหลักการทำงานของแต่ละแบบจำลองแล้วพบว่า

- ใน 1 รอบการทำงานของแบบจำลองที่นำเสนอใช้การเรียนรู้เพียง 1 ขั้นตอนใน 1 รอบการทำงาน ในขณะที่แบบจำลอง Hierarchical Overlapped SOM ใช้การเรียนรู้ถึง 3 ขั้นตอนใน 1 รอบการทำงานและแบบจำลอง Concurrent Self-Organizing Maps ใช้การเรียนรู้เท่ากับจำนวนกลุ่มของข้อมูล จะเห็นได้ว่าเป็นกระบวนการการเรียนรู้ค่อนข้างช้าซ้อนและใช้เวลาค่อนข้างสูงเมื่อเทียบกับแบบจำลองที่นำเสนอ

- ในส่วนของเกณฑ์ที่ใช้ในการพิจารณาการลบคลัสเตอร์ แบบจำลองที่นำเสนอพิจารณาการลบจากผลคูณของค่า Entropy กับขนาดของคลัสเตอร์นั้นเทียบกับข้อมูลที่ใช้เรียนรู้โดยค่า Entropy เป็นค่าที่วัดคุณภาพของการจัดกลุ่มโดยพิจารณาจากข้อมูลที่ประกอบกันอยู่ในคลัสเตอร์นั้น ถ้าค่า Entropy ที่เข้าใกล้ 0 แสดงว่าประสิทธิภาพของการจัดแบ่งกลุ่มมีความถูกต้องสูงเนื่องจากมีสมาชิกของกลุ่มอื่นเข้ามาอยู่ในคลัสเตอร์เดียวกันน้อย กรณีที่ค่า Entropy เท่ากับ 0 แสดง

ว่าระดับประสิทธิภาพของการจัดแบ่งกลุ่มมีความถูกต้องสูงสุดสมาชิกทุกตัวภายในคลัสเตอร์นั้นอยู่กลุ่มเดียวกัน ในขณะที่ Hierarchical Overlapped SOM พิจารณาการลบคลัสเตอร์จากการเป็นตัวแทนของข้อมูล โดยแบบจำลองเสนอว่า คลัสเตอร์ที่มีข้อมูลเป็นสมาชิกน้อยกว่า 3 ตัวจะถูกลบ ซึ่งเป็นเกณฑ์ที่ให้ผลดีระดับหนึ่งแต่ไม่สามารถบอกได้ว่าแต่ละคลัสเตอร์มีประสิทธิภาพมากน้อยแค่ไหน สมาชิกภายในคลัสเตอร์อยู่ในกลุ่มเดียวกันหรือคนละกลุ่มและการที่จะหาจำนวนสมาชิกที่ใช้เป็นตัวกำหนดการลบที่เหมาะสมสำหรับทุกข้อมูลที่หลากหลายเป็นเรื่องยากไม่สามารถสรุปได้ว่าจำนวนสมาชิก 3 ตัวเหมาะสมกับทุกข้อมูล

- ในเรื่องของข้อมูลที่นำมาใช้ในการเรียนรู้ เมื่อมีข้อมูลเข้ามาเรียนรู้แบบจำลองที่นำเสนอสามารถทำการเรียนรู้ได้ทันที ในขณะที่ Concurrent Self-Organizing Maps ก่อนทำการเรียนรู้ต้องทำการแบ่งข้อมูลที่ใช้เรียนรู้ออกเป็นกลุ่มๆ ตามกลุ่มที่มีอยู่ก่อนที่จะส่งให้แต่ละโครงข่ายที่เตรียมไว้สำหรับแต่ละกลุ่มทำการเรียนรู้ ส่วนในกรณีของงานวิจัยเรื่อง Semi-supervised Learning of Dynamic Self-Organising Maps ก่อนการเรียนรู้แบบจำลองต้องใช้เวลาในการคำนวณหาค่า Growth Threshold (GT) คือค่าความผิดพลาดสะสมสูงสุดที่โหนดสามารถมีได้ เพื่อที่จะนำไปใช้พิจารณาการสร้างโหนดใหม่ในชั้น GSOM ต่อไป

- ในส่วนของขั้นตอนการแข่งขัน แบบจำลองที่นำเสนอจะใช้ค่าความเป็นสมาชิก (membership) ของอินพุตกับโหนดเป็นเกณฑ์การตัดสินใจโหนดขณะแทนการใช้ยูคลิเดียนดิสแทนซ์ (Euclidean distance) เหมือนในงานวิจัยเรื่อง Semi-supervised Learning of Dynamic Self-Organising จากแนวคิดเรื่องการใช้เซตของความคลุมเครือ (Fuzzy Set) อธิบายความน่าจะเป็นของเหตุการณ์ที่มีลักษณะคลุมเครือไม่ชัดเจนหรือมีความไม่แน่นอน ซึ่งน่าจะสามารถนำมาประยุกต์ใช้ข้อมูลลักษณะที่มีความคลุมเครือได้ดี โดยผู้วิจัยได้ทำการทดลองการเปรียบเทียบการจัดหมวดหมู่ของข้อมูลโดยใช้กระบวนการในการเรียนรู้ที่เหมือนกันแต่ใช้เกณฑ์การตัดสินใจโหนดขณะที่แตกต่างกัน จากผลการทดลองที่ได้ดังแสดงในตารางที่ 3.7 ซึ่งการสนับสนุนแนวคิดนี้

ตารางที่ 3.7 ผลการทดลองเปรียบเทียบการใช้เกณฑ์การตัดสินใจโหนดขณะที่แตกต่างกัน

ข้อมูล	Flower1		Flower2		Sawtooth		Fan	
	ถูกต้อง	คลัสเตอร์	ถูกต้อง	คลัสเตอร์	ถูกต้อง	คลัสเตอร์	ถูกต้อง	คลัสเตอร์
Membership	98.44	239.8	98.51	273	99.85	80.4	99.93	83.8
Euclidean Distance	87.2	307.2	88.7	319.8	99.85	95	100	12

## บทที่ 4

# ผลการทดสอบการจัดหมวดหมู่ของข้อมูลโดยใช้

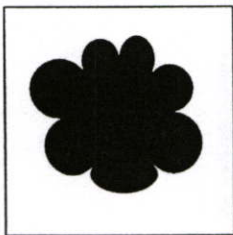
## Supervised SOM Using Fuzzy Membership Function

ในบทนี้จะกล่าวถึงข้อมูลและค่าพารามิเตอร์ต่างๆ ที่ใช้ในการทดลองและผลที่ได้จากการทดลอง โดยข้อมูลที่นำมาใช้ในการทดลองมี 2 ชุด คือ 1) ชุดข้อมูลทดสอบ [14] ซึ่งประกอบด้วย 4 ชุดข้อมูล คือ Flower1, Flower2, Sawtooth และ Fan 2) ข้อมูลมาตรฐานซึ่งประกอบด้วย 5 ชุดข้อมูล คือ Iris, Wine, Ionosphere, Haberman และ Heart-Statlog ในการทดลองนี้จะทำการเปรียบเทียบค่าเปอร์เซ็นต์ความถูกต้อง โดยทำการเปรียบเทียบกับ Fuzzy ARTMAP, Learning Vector Quantization (LVQ), Backpropagation Neural Network (BPN) และ Radial Basis Function (RBF) ซึ่งเป็น โมเดล ทางด้านการจัดหมวดหมู่ข้อมูลที่ได้รับการยอมรับและถูกนำไปประยุกต์ใช้อย่างแพร่หลาย

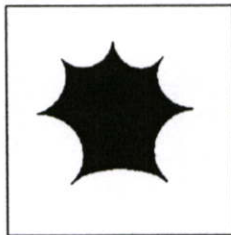
### 4.1 ข้อมูลที่ใช้ในการทดลอง

#### 4.1.1 ข้อมูลทดสอบ [14]

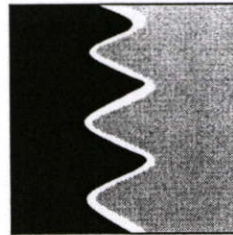
ข้อมูลมีทั้งหมด 4 ชุดข้อมูลดังรูปที่ 4.1 ข้อมูลแต่ละรูปมีขนาด 500 x 500 pixel โดยแต่ละรูปแบบอินพุตได้จากการสุ่มคู้ันดับ (x,y) จากรูปที่สร้างขึ้น โดยมีการสุ่มรูปแบบอินพุตขึ้นมาทั้งหมด 4,000 รูปแบบอินพุต และนำชุดข้อมูลที่ได้มาผ่านขั้นตอนการ normalize ให้เป็นเลขจำนวนจริงที่อยู่ในช่วง 0 ถึง 1



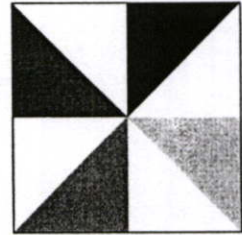
ก) ชุดข้อมูล Flower1



ข) ชุดข้อมูล Flower2



ค) ชุดข้อมูล Sawtooth



ง) ชุดข้อมูล Fan

รูปที่ 4.1 ลักษณะของชุดข้อมูลทดสอบ [14]

ชุดข้อมูลทดสอบแต่ละชุดจะประกอบด้วยข้อมูลที่ใช้ในการเรียนรู้ 2,000 รูปแบบอินพุต และข้อมูลที่ใช้ทดสอบ 2,000 รูปแบบอินพุต โดยแต่ละคลาสมีรายละเอียดดังตารางที่ 4.1 และ 4.2

ตารางที่ 4.1 การแบ่งข้อมูลของชุดข้อมูล Flower1, Flower2 และ Sawtooth

คลาส	ข้อมูลที่ใช้ในการเรียนรู้โมเดล	ข้อมูลที่ใช้ในการทดสอบโมเดล
1	1,000	1,000
2	1,000	1,000

ตารางที่ 4.2 การแบ่งข้อมูลของชุดข้อมูล Fan

คลาส	ข้อมูลที่ใช้ในการเรียนรู้โมเดล	ข้อมูลที่ใช้ในการทดสอบโมเดล
1	500	500
2	500	500
3	500	500
4	500	500

#### 4.1.2 ข้อมูลมาตรฐาน

##### 1) Iris

ข้อมูล Iris ถูกเก็บโดย Anderson ในปี 1935 เป็นการเก็บข้อมูลขนาดของกลีบเลี้ยงและกลีบดอกของดอก Iris 3 ประเภท คือ Iris Setosa, Iris Virginica และ Iris Versicolour โดยแต่ละประเภทประกอบด้วยข้อมูลตัวอย่างจำนวน 50 ตัวอย่างเพื่อนำมาศึกษาจำแนกว่าจากข้อมูลที่กำหนดให้เป็นดอก Iris ชนิดใด

ในชุดข้อมูลดอก Iris แต่ละข้อมูลประกอบด้วย 4 คุณลักษณะดังนี้

1. Sepal length คือ ความยาวของกลีบเลี้ยง
2. Sepal width คือ ความกว้างของกลีบเลี้ยง
3. Petal length คือ ความยาวของกลีบดอก
4. Petal width คือ ความกว้างของกลีบดอก

ตัวอย่างชุดข้อมูลเช่น 7.2, 3.2, 6.0, 1.8, Iris-Virginica

4.9, 3.0, 1.4, 0.2, Iris-Setosa

7.0, 3.2, 4.7, 1.4, Iris-Versicolor

ในการทดลองจะทำการแบ่งชุดข้อมูลตัวอย่าง 150 ตัวอย่างออกเป็น 2 ส่วนคือข้อมูลที่ใช้ในการเรียนรู้แบบจำลอง 120 ตัวอย่าง และข้อมูลที่ใช้ในการทดสอบแบบจำลอง 30 ตัวอย่างรายละเอียดดังแสดงที่ตาราง 4.3

### ตารางที่ 4.3 การแบ่งข้อมูลของชุดข้อมูล Iris

คลาส	ข้อมูลที่ใช้ในการเรียนรู้โมเดล	ข้อมูลที่ใช้ในการทดสอบโมเดล
1 (Iris Setosa)	40	10
2 (Iris Virginica)	40	10
3 (Iris Versicolour)	40	10

### 2) Wine

ข้อมูล Wine ถูกเก็บโดย Stefan Aeberhard สถาบัน Institute of Pharmaceutical and Food Analysis and Technologies ในเดือนกรกฎาคม ปี 1991 ข้อมูล Wine เป็นชุดข้อมูลที่ได้จากการวิเคราะห์ทางเคมีของเหล้าไวน์ในประเทศอิตาลี 3 ประเภทคือประเภทที่ 1, 2 และ 3 ซึ่งแบ่งตามการเพาะปลูกที่แตกต่างกัน ชุดของข้อมูลประกอบด้วยข้อมูลตัวอย่างจำนวน 178 ตัวอย่างแต่ละประเภทประกอบด้วยข้อมูลตัวอย่างจำนวน 59, 71 และ 48 ตัวอย่างตามลำดับ ในแต่ละข้อมูลตัวอย่างประกอบด้วย 13 คุณลักษณะซึ่งเป็นรายละเอียดของส่วนผสมของไวน์

ตัวอย่างชุดข้อมูลเช่น

1,13.72,1.43,2.5,16.7,108,3.4,3.67,.19,2.04,6.8,.89,2.87,1285

2,12.37,.94,1.36,10.6,88,1.98,.57,.28,.42,1.95,1.05,1.82,520

3,12.86,1.35,2.32,18,122,1.51,1.25,.21,.94,4.1,.76,1.29,630

ในการทดลองจะทำการแบ่งชุดข้อมูลตัวอย่าง 178 ตัวอย่างออกเป็น 2 ส่วนคือข้อมูลที่ใช้ในการเรียนรู้แบบจำลอง 90 ตัวอย่างและข้อมูลที่ใช้ในการทดสอบแบบจำลอง 88 ตัวอย่างรายละเอียดดังแสดงที่ตาราง 4.4

### ตารางที่ 4.4 การแบ่งข้อมูลของชุดข้อมูล Wine

คลาส	ข้อมูลที่ใช้ในการเรียนรู้โมเดล	ข้อมูลที่ใช้ในการทดสอบโมเดล
1	30	29
2	36	35
3	24	24

### 3) Ionosphere

ข้อมูล Ionosphere ถูกเก็บโดย Vince Sigillito จาก Applied Physics Laboratory, Johns Hopkins University ในปี 1989 โดยทำการสะสมข้อมูลเรดาร์เพื่อดูการปลดปล่อยอิเล็กตรอนของชั้นบรรยากาศโดยแบ่งข้อมูลออกเป็น 2 ประเภท คือ Good (เรดาร์จะตอบกลับมาโดยแสดงให้เห็นโครงสร้างในชั้นบรรยากาศ) และ Bad (เรดาร์ไม่ตอบอะไรกลับมา) ชุดของข้อมูลประกอบด้วยข้อมูลตัวอย่างจำนวน 351 ตัวอย่างแต่ละประเภทประกอบด้วยข้อมูลตัวอย่างจำนวน

225 และ 126 ตัวอย่างตามลำดับ ในแต่ละข้อมูลตัวอย่างประกอบด้วย 34 คุณลักษณะที่มีค่าเป็นข้อมูลแบบต่อเนื่อง

ตัวอย่างชุดข้อมูลเช่น 0,0,0,0,0,1,-1,0,0,-1,-1,0,0,0,1,1,-1,-1,0,0,0,1,1,1,1,0,0,1,1,0,0,b. 1,0,0.96355,-0.07198,1,-0.14333,1,-0.21313,1,-0.36174,0.92570,-0.43569,0.94510,-0.40668,0.90392,-0.46381,0.98305,-0.35257,0.84537,-0.66020,0.75346,-0.60589,0.69637,-0.64225,0.85106,-0.65440,0.57577,-0.69712,0.25435,-0.63919,0.45114,-0.72779,0.38895,-0.73420,g. 1,0,-0.01864,-0.08459,0,0,0,0,0.11470,-0.26810,-0.45663,-0.38172,0,0,-0.33656,0.38602,-0.37133,0.15018,0.63728,0.22115,0,0,0,0,-0.14803,-0.01326,0.20645,-0.02294,0,0,0.16595,0.24086,-0.08208,0.38065,b.

ในการทดลองจะทำการแบ่งชุดข้อมูลตัวอย่าง 351 ตัวอย่างออกเป็น 2 ส่วนคือข้อมูลที่ใช้ในการเรียนรู้แบบจำลอง 200 ตัวอย่างและข้อมูลที่ใช้ในการทดสอบแบบจำลอง 151 ตัวอย่างรายละเอียดดังแสดงที่ตาราง 4.5

ตารางที่ 4.5 การแบ่งข้อมูลของชุดข้อมูล Ionosphere

คลาส	ข้อมูลที่ใช้ในการเรียนรู้โมเดล	ข้อมูลที่ใช้ในการทดสอบโมเดล
1 (good)	101	124
2 (bad)	99	27

#### 4) Haberman

ข้อมูล Haberman ถูกเก็บโดย Tjen-Sien Lim ในปี 1999 เป็นข้อมูลศึกษาของการอยู่รอดของคนไข้ที่เข้ารับการรักษาด้วยโรคมะเร็งทรวงอกระหว่างปี 1958 – 1970 ของ University of Chicago's Billings Hospital โดยจำแนกข้อมูลออกเป็น 2 ประเภทตามสถานะคนไข้ คือ 1) คนไข้มีชีวิตอยู่เกิน 5 ปี และ 2) คนไข้ตายภายใน 5 ปี ชุดข้อมูลประกอบด้วย 306 รูปแบบอินพุทในแต่ละข้อมูลประกอบด้วย 3 คุณลักษณะ ดังนี้

1. Age of patient at time of operation
2. Patient's year of operation (year - 1900)
3. Number of positive auxiliary nodes detected

ตัวอย่างชุดข้อมูลเช่น 30, 64, 1, 1  
37, 63, 0, 1  
38, 69, 21, 2

ในการทดลองจะทำการแบ่งชุดข้อมูลตัวอย่าง 306 ตัวอย่างออกเป็น 2 ส่วนคือข้อมูลที่ใช้ในการเรียนรู้แบบจำลอง 214 ตัวอย่างและข้อมูลที่ใช้ในการทดสอบแบบจำลอง 92 ตัวอย่างรายละเอียดดังแสดงที่ตาราง 4.6

ตารางที่ 4.6 การแบ่งข้อมูลของชุดข้อมูล Haberman

คลาส	ข้อมูลที่ใช้ในการเรียนรู้โมเดล	ข้อมูลที่ใช้ในการทดสอบโมเดล
1 (คนไข้มีชีวิตอยู่เกิน 5 ปี)	157	68
2 (คนไข้ตายภายใน 5 ปี)	57	24

### 5) Heart Statlog

ข้อมูล Heart-statlog ถูกเก็บโดย University of California, San Diego medical Center ข้อมูล Heart-statlog มีจำนวนทั้งสิ้น 270 รูปแบบอินพุต แบ่งข้อมูลออกเป็น 2 ประเภทคือ 1) Absent (ไม่เป็นมะเร็ง) จำนวน 150 ข้อมูล และ 2) Present (เป็นมะเร็ง) จำนวน 120 ข้อมูลในแต่ละข้อมูลประกอบด้วย 13 คุณลักษณะที่มีค่าเป็นข้อมูลแบบต่อเนื่อง

ตัวอย่างชุดข้อมูลเช่น 70, 1, 4, 130, 322, 0, 2, 109, 0, 2.4, 2, 3, 3, present  
67, 0, 3, 115, 564, 0, 2, 160, 0, 1.6, 2, 0, 7, absent  
57, 1, 2, 124, 261, 0, 0, 141, 0, 0.3, 1, 0, 7, present

ในการทดลองจะทำการแบ่งชุดข้อมูลตัวอย่าง 270 ตัวอย่างออกเป็น 2 ส่วนคือข้อมูลที่ใช้ในการเรียนรู้แบบจำลอง 135 ตัวอย่างและข้อมูลที่ใช้ในการทดสอบแบบจำลอง 135 ตัวอย่างรายละเอียดดังแสดงที่ตาราง 4.7

ตารางที่ 4.7 การแบ่งข้อมูลของชุดข้อมูล Heart-Statlog

คลาส	ข้อมูลที่ใช้ในการเรียนรู้โมเดล	ข้อมูลที่ใช้ในการทดสอบโมเดล
1 (Absent)	75	75
2 (Present)	60	60

## 4.2 การกำหนดพารามิเตอร์ของระบบ

1. จำนวนรอบที่ใช้ในการเรียนรู้แบบจำลอง 1,000 รอบ
2. จำนวนคลัสเตอร์เริ่มต้นของแต่ละคลาส = 1
3. ค่า threshold การเพิ่มคลัสเตอร์ ( $th$ )
4. ค่า threshold การลบคลัสเตอร์ ( $thde$ )
5. ค่าเริ่มต้นของการหาค่าอัตราการเรียนรู้ ( $u_0$ ) = 2

6. ค่าเริ่มต้นของการหาค่าขนาดความกว้างของ Gaussian function ( $v_0$ ) = 2.3

ในการทดลองกับข้อมูลแต่ละชุดข้างต้นเรากำหนดให้ค่า  $th$  และ  $thde$  เปลี่ยนแปลงอยู่ในช่วง 0.001 ถึง 0.5 แล้วจึงเลือกผลการจัดหมวดหมู่ที่ดีที่สุด

### 4.3 ผลการทดลอง

ในงานวิจัยนี้ได้ทำการทดลองโดยเปรียบเทียบกับโมเดล Fuzzy ARTMAP, LVQ, BPN และ RBF ผลที่ได้จากการทดลองจะถูกแสดงใน 3 ลักษณะ ดังนี้

1) ค่าเปอร์เซ็นต์ความถูกต้องเฉลี่ยของชุดข้อมูลทดสอบ ซึ่งเป็นค่าเฉลี่ยที่ได้จากการทดลองที่มีการกำหนดค่าพารามิเตอร์ที่เหมือนกันทั้งหมด 5 ครั้ง โดยในแต่ละครั้งจะมีการสุ่มเลือกค่าน้ำหนักเริ่มต้นที่แตกต่างกัน โดยใช้เกณฑ์การหยุดเรียนรู้เมื่อค่าเปอร์เซ็นต์ความถูกต้องของการเรียนรู้เท่ากับ 100 เปอร์เซ็นต์หรือเมื่อเรียนรู้ครบ 1,000 รอบ เฉพาะกรณีของ BPN ใช้เกณฑ์การหยุดเรียนรู้เมื่อครบ 10,000 รอบ

2) ค่าเปอร์เซ็นต์ความถูกต้องต่ำสุดของชุดการทดลองในข้อ 1

3) ค่าเปอร์เซ็นต์ความถูกต้องสูงสุดของชุดการทดลองในข้อ 1

#### 4.3.1 ชุดข้อมูลทดสอบ

##### 1) ชุดข้อมูล Flower1

จากการทดลองกับชุดข้อมูล Flower1 โดยทำการเปลี่ยนแปลงค่าของตัวแปรต่างๆ ที่ใช้ในแบบจำลองเพื่อให้ได้ผลการจัดหมวดหมู่ที่ดีที่สุดของแต่ละวิธี ผลการทดลองแสดงดังตารางที่ 4.8

ตารางที่ 4.8 ผลการทดลองของข้อมูล Flower1

โมเดล	ความถูกต้องเฉลี่ย (%)		ความถูกต้องต่ำสุด (%)		ความถูกต้องสูงสุด (%)	
	ถูกต้อง	คลัสเตอร์	ถูกต้อง	คลัสเตอร์	ถูกต้อง	คลัสเตอร์
Propose Model	98.56	276.4	98.35	315	98.65	233
Fuzzy ARTMAP	97.46	147.8	96.75	148	97.85	143
LVQ	96.94	110	96.15	110	97.5	110
BPN	91.96	8	89.55	8	95	8
RBF	97.08	93	96.8	93	97.4	93

ค่าของตัวแปรที่ใช้ในแบบจำลองที่นำเสนอ คือ  $th = 0.02$ ,  $thde = 0.04$ , ค่าของตัวแปรที่ใช้ใน Fuzzy ARTMAP คือ Vigilance = 0.9, อัตราการเรียนรู้ = 0.2, ค่าของตัวแปรที่ใช้ใน

LVQ คือจำนวนคลัสเตอร์ = 110, อัตราการเรียนรู้ = 0.001, ค่าของตัวแปรที่ใช้ใน BPN คือจำนวน โหนดชั้นซ่อน = 8 โหนดและค่าของตัวแปรที่ใช้ใน RBF คือ จำนวนคลัสเตอร์ = 93, อัตราการเรียนรู้ = 0.01

## 2) ชุดข้อมูล Flower2

จากการทดลองกับชุดข้อมูล Flower2 โดยทำการเปลี่ยนแปลงค่าของตัวแปรต่างๆ ที่ใช้ในแบบจำลองเพื่อให้ได้ผลการจัดหมวดหมู่ที่ดีที่สุดของแต่ละวิธี ผลการทดลองแสดงดังตารางที่ 4.9

ตารางที่ 4.9 ผลการทดลองของข้อมูล Flower2

โมเดล	ความถูกต้องเฉลี่ย (%)		ความถูกต้องต่ำสุด (%)		ความถูกต้องสูงสุด (%)	
	ถูกต้อง	คลัสเตอร์	ถูกต้อง	คลัสเตอร์	ถูกต้อง	คลัสเตอร์
Propose Model	98.55	232	98.4	171	98.7	259
Fuzzy ARTMAP	97.99	141.6	97.7	139	98.3	144
LVQ	96.4	112	96.2	112	96.65	112
BPN	92.35	12	89.5	12	94.5	12
RBF	97.99	89	97.55	89	98.25	89

ค่าของตัวแปรที่ใช้ในแบบจำลองที่นำเสนอ คือ  $th = 0.05$ ,  $thde = 0.07$ , ค่าของตัวแปรที่ใช้ใน Fuzzy ARTMAP คือ Vigilance = 0.9, อัตราการเรียนรู้ = 0.4, ค่าของตัวแปรที่ใช้ใน LVQ คือจำนวนคลัสเตอร์ = 112, อัตราการเรียนรู้ = 0.001, ค่าของตัวแปรที่ใช้ใน BPN คือจำนวน โหนดชั้นซ่อน = 12 โหนดและค่าของตัวแปรที่ใช้ใน RBF คือ จำนวนคลัสเตอร์ = 89, อัตราการเรียนรู้ = 0.01

## 3) ชุดข้อมูล Sawtooth

จากการทดลองกับชุดข้อมูล Sawtooth โดยทำการเปลี่ยนแปลงค่าของตัวแปรต่างๆ ที่ใช้ในแบบจำลองเพื่อให้ได้ผลการจัดหมวดหมู่ที่ดีที่สุดของแต่ละวิธี ผลการทดลองแสดงดังตารางที่ 4.10

ตารางที่ 4.10 ผลการทดลองของข้อมูล Sawtooth

โมเดล	ความถูกต้องเฉลี่ย (%)		ความถูกต้องต่ำสุด (%)		ความถูกต้องสูงสุด (%)	
	ถูกต้อง	คลัสเตอร์	ถูกต้อง	คลัสเตอร์	ถูกต้อง	คลัสเตอร์
Propose Model	99.85	72.8	99.7	59	100	67
Fuzzy ARTMAP	99.82	135.8	99.7	142	100	115
LVQ	99.85	110	99.7	110	99.9	110
BPN	97.35	12	96.1	12	99.55	12
RBF	99.66	91	99.55	91	99.8	91

ค่าของตัวแปรที่ใช้ในแบบจำลองที่นำเสนอ คือ  $th = 0.15$ ,  $thde = 0.1$ , ค่าของตัวแปรที่ใช้ใน Fuzzy ARTMAP คือ Vigilance = 0.9, อัตราการเรียนรู้ = 0.1, ค่าของตัวแปรที่ใช้ใน LVQ คือจำนวนคลัสเตอร์ = 110, อัตราการเรียนรู้ = 0.001, ค่าของตัวแปรที่ใช้ใน BPN คือจำนวนโหนดชั้นซ่อน = 12 โหนด และค่าของตัวแปรที่ใช้ใน RBF คือ จำนวนคลัสเตอร์ = 91, อัตราการเรียนรู้ = 0.01

#### 4) ชุดข้อมูล Fan

จากการทดลองกับชุดข้อมูล Fan โดยทำการเปลี่ยนแปลงค่าของตัวแปรต่างๆ ที่ใช้ในแบบจำลองเพื่อให้ได้ผลการจัดหมวดหมู่ที่ดีที่สุดของแต่ละวิธี ผลการทดลองแสดงดังตารางที่ 4.11

ตารางที่ 4.11 ผลการทดลองของข้อมูล Fan

โมเดล	ความถูกต้องเฉลี่ย (%)		ความถูกต้องต่ำสุด (%)		ความถูกต้องสูงสุด (%)	
	ถูกต้อง	คลัสเตอร์	ถูกต้อง	คลัสเตอร์	ถูกต้อง	คลัสเตอร์
Propose Model	99.93	83.8	99.85	47	99.95	86
Fuzzy ARTMAP	100	4	100	4	100	4
LVQ	99.95	4	99.85	4	100	4
BPN	100	8	100	8	100	8
RBF	99.98	112	99.95	112	100	112

ค่าของตัวแปรที่ใช้ในแบบจำลองที่นำเสนอ คือ  $th = 0.03$ ,  $thde = 0.009$ , ค่าของตัวแปรที่ใช้ใน Fuzzy ARTMAP คือ Vigilance = 0.3, อัตราการเรียนรู้ = 0.1, ค่าของตัวแปรที่ใช้ใน

LVQ คือจำนวนคลัสเตอร์ = 4, อัตราการเรียนรู้ = 0.001, ค่าของตัวแปรที่ใช้ใน BPN คือจำนวน โหนดชั้นซ่อน = 8 โหนดและค่าของตัวแปรที่ใช้ใน RBF คือ จำนวนคลัสเตอร์ = 112, อัตราการเรียนรู้ = 0.01

#### 4.3.2 ชุดข้อมูลมาตรฐาน

##### 1) ชุดข้อมูล Iris

จากการทดลองกับชุดข้อมูล Iris โดยทำการเปลี่ยนแปลงค่าของตัวแปรต่างๆ ที่ใช้ในแบบจำลองเพื่อให้ได้ผลการจัดหมวดหมู่ที่ดีที่สุดของแต่ละวิธี ผลการทดลองแสดงดังตารางที่

4.12

ตารางที่ 4.12 ผลการทดลองของข้อมูล Iris

โมเดล	ความถูกต้องเฉลี่ย (%)		ความถูกต้องต่ำสุด (%)		ความถูกต้องสูงสุด (%)	
	ถูกต้อง	คลัสเตอร์	ถูกต้อง	คลัสเตอร์	ถูกต้อง	คลัสเตอร์
Propose Model	100	8.8	100	19	100	4
Fuzzy ARTMAP	98	15.6	96.67	16	98.33	14
LVQ	100	15	100	15	100	15
BPN	100	3	100	3	100	3
RBF	100	32	100	32	100	32

ค่าของตัวแปรที่ใช้ในแบบจำลองที่นำเสนอ คือ  $th = 0.09$ ,  $thde = 0.02$ , ค่าของตัวแปรที่ใช้ใน Fuzzy ARTMAP คือ Vigilance = 0.8, อัตราการเรียนรู้ = 0.2, ค่าของตัวแปรที่ใช้ใน LVQ คือจำนวนคลัสเตอร์ = 15, อัตราการเรียนรู้ = 0.001, ค่าของตัวแปรที่ใช้ใน BPN คือจำนวน โหนดชั้นซ่อน = 3 โหนดและค่าของตัวแปรที่ใช้ใน RBF คือ จำนวนคลัสเตอร์ = 32, อัตราการเรียนรู้ = 0.01

##### 2) ชุดข้อมูล Wine

จากการทดลองกับชุดข้อมูล Wine โดยทำการเปลี่ยนแปลงค่าของตัวแปรต่างๆ ที่ใช้ในแบบจำลองเพื่อให้ได้ผลการจัดหมวดหมู่ที่ดีที่สุดของแต่ละวิธี ผลการทดลองแสดงดังตารางที่

4.13

ตารางที่ 4.13 ผลการทดลองของข้อมูล Wine

โมเดล	ความถูกต้องเฉลี่ย (%)		ความถูกต้องต่ำสุด (%)		ความถูกต้องสูงสุด (%)	
	ถูกต้อง	คลัสเตอร์	ถูกต้อง	คลัสเตอร์	ถูกต้อง	คลัสเตอร์
Propose Model	97.96	5.4	97.73	5	98.86	6
Fuzzy ARTMAP	96.59	26.4	95.45	27	97.73	25
LVQ	95	48	94.32	48	95.45	32
BPN	97.95	16	97.73	16	98.86	16
RBF	96.82	35	96.59	35	97.73	35

ค่าของตัวแปรที่ใช้ในแบบจำลองที่นำเสนอ คือ  $th = 0.5$ ,  $thde = 0.4$ , ค่าของตัวแปรที่ใช้ใน Fuzzy ARTMAP คือ  $Vigilance = 0.8$ , อัตราการเรียนรู้ = 0.7, ค่าของตัวแปรที่ใช้ใน LVQ คือจำนวนคลัสเตอร์ = 48, อัตราการเรียนรู้ = 0.001 และค่าของตัวแปรที่ใช้ใน BPN คือจำนวนโหนดชั้นซ่อน = 16 โหนดและค่าของตัวแปรที่ใช้ใน RBF คือ จำนวนคลัสเตอร์ = 35, อัตราการเรียนรู้ = 0.01

### 3) ชุดข้อมูล Ionosphere

จากการทดลองกับชุดข้อมูล Ionosphere โดยทำการเปลี่ยนแปลงค่าของตัวแปรต่างๆ ที่ใช้ในแบบจำลองเพื่อให้ได้ผลการจัดหมวดหมู่ที่ดีที่สุดของแต่ละวิธี ผลการทดลองแสดงดังตารางที่ 4.14

ตารางที่ 4.14 ผลการทดลองของข้อมูล Ionosphere

โมเดล	ความถูกต้องเฉลี่ย (%)		ความถูกต้องต่ำสุด (%)		ความถูกต้องสูงสุด (%)	
	ถูกต้อง	คลัสเตอร์	ถูกต้อง	คลัสเตอร์	ถูกต้อง	คลัสเตอร์
Propose Model	96.16	72.8	94.7	76	97.35	66
Fuzzy ARTMAP	97.75	129.2	96.69	133	98.01	127
LVQ	94.83	34	94.04	34	95.36	34
BPN	93.51	38	90.07	38	95.36	38
RBF	98.81	50	98.01	50	99.34	50

ค่าของตัวแปรที่ใช้ในแบบจำลองที่นำเสนอ คือ  $th = 0.01$ ,  $thde = 0.13$ , ค่าของตัวแปรที่ใช้ใน Fuzzy ARTMAP คือ  $Vigilance = 0.9$ , อัตราการเรียนรู้ = 0.5, ค่าของตัวแปรที่ใช้ใน

LVQ คือจำนวนคลัสเตอร์ = 34, อัตราการเรียนรู้ = 0.001, ค่าของตัวแปรที่ใช้ใน BPN คือจำนวน โหนดชั้นซ่อน = 38 โหนดและค่าของตัวแปรที่ใช้ใน RBF คือ จำนวนคลัสเตอร์ = 50, อัตราการเรียนรู้ = 0.01

#### 4) ชุดข้อมูล Haberman

จากการทดลองกับชุดข้อมูล Haberman โดยทำการเปลี่ยนแปลงค่าของตัวแปรต่างๆ ที่ใช้ในแบบจำลองเพื่อให้ได้ผลการจัดหมวดหมู่ที่ดีที่สุดของแต่ละวิธี ผลการทดลองแสดงดังตารางที่ 4.15

ตารางที่ 4.15 ผลการทดลองของข้อมูล Haberman

โมเดล	ความถูกต้องเฉลี่ย (%)		ความถูกต้องต่ำสุด (%)		ความถูกต้องสูงสุด (%)	
	ถูกต้อง	คลัสเตอร์	ถูกต้อง	คลัสเตอร์	ถูกต้อง	คลัสเตอร์
Propose Model	76.09	20.2	72.83	17	78.26	20
Fuzzy ARTMAP	73.91	4.2	73.91	5	73.91	4
LVQ	74.35	2	69.57	2	76.09	2
BPN	75.22	9	76.09	9	73.91	9
RBF	69.56	2	52.17	2	73.91	2

ค่าของตัวแปรที่ใช้ในแบบจำลองที่นำเสนอ คือ  $th = 0.07$ ,  $thde = 0.01$ , ค่าของตัวแปรที่ใช้ใน Fuzzy ARTMAP คือ Vigilance = 0.3, อัตราการเรียนรู้ = 0.7, ค่าของตัวแปรที่ใช้ใน LVQ คือจำนวนคลัสเตอร์ = 2, อัตราการเรียนรู้ = 0.001, ค่าของตัวแปรที่ใช้ใน BPN คือจำนวน โหนดชั้นซ่อน = 9 โหนด และค่าของตัวแปรที่ใช้ใน RBF คือ จำนวนคลัสเตอร์ = 2, อัตราการเรียนรู้ = 0.01

#### 5) ชุดข้อมูล Heart-Statlog

จากการทดลองกับชุดข้อมูล Heart-Statlog โดยทำการเปลี่ยนแปลงค่าของตัวแปรต่างๆ ที่ใช้ในแบบจำลองเพื่อให้ได้ผลการจัดหมวดหมู่ที่ดีที่สุดของแต่ละวิธี ผลการทดลองแสดงดังตารางที่ 4.16

ตารางที่ 4.16 ผลการทดลองของข้อมูล Heart-Statlog

โมเดล	ความถูกต้องเฉลี่ย (%)		ความถูกต้องต่ำสุด (%)		ความถูกต้องสูงสุด (%)	
	ถูกต้อง	คลัสเตอร์	ถูกต้อง	คลัสเตอร์	ถูกต้อง	คลัสเตอร์
Propose Model	81.04	43.4	78.52	56	83.7	46
Fuzzy ARTMAP	80.3	63	77.78	62	81.48	60
LVQ	84.44	2	84.44	2	84.44	2
BPN	77.93	18	72.59	18	80.74	18
RBF	85.48	35	83.7	35	87.41	35

ค่าของตัวแปรที่ใช้ในแบบจำลองที่นำเสนอคือ  $th = 0.03$ ,  $thde = 0.02$ , ค่าของตัวแปรที่ใช้ใน Fuzzy ARTMAP คือ  $Vigilance = 0.8$ , อัตราการเรียนรู้ = 0.3, ค่าของตัวแปรที่ใช้ใน LVQ คือจำนวนคลัสเตอร์ = 2, อัตราการเรียนรู้ = 0.001, ค่าของตัวแปรที่ใช้ใน BPN คือจำนวนโหนดชั้นซ่อน = 18 โหนดและค่าของตัวแปรที่ใช้ใน RBF คือ จำนวนคลัสเตอร์ = 35, อัตราการเรียนรู้ = 0.01

#### 4.4 สรุปผลการทดลอง

##### 4.4.1 สรุปผลการทดลองของชุดข้อมูลทดสอบ

1) ข้อมูล Flower1 ค่าความถูกต้องเฉลี่ยของแบบจำลองที่นำเสนอมีค่าสูงกว่ากับค่าความถูกต้องเฉลี่ยของ Fuzzy ARTMAP 1.1% สูงกว่าค่าความถูกต้องเฉลี่ยของ LVQ 1.62% สูงกว่าค่าความถูกต้องเฉลี่ยของ BPN 6.6% และสูงกว่าค่าความถูกต้อง RBF 1.48%

2) ข้อมูล Flower2 ค่าความถูกต้องเฉลี่ยของแบบจำลองที่นำเสนอมีค่าสูงกว่าค่าความถูกต้องเฉลี่ยของ Fuzzy ARTMAP 0.56% สูงกว่าค่าความถูกต้องเฉลี่ยของ LVQ 2.15% สูงกว่าค่าความถูกต้องของ BPN 6.2% และสูงกว่าค่าความถูกต้อง RBF 0.56%

3) ข้อมูล Sawtooth ค่าความถูกต้องเฉลี่ยของแบบจำลองที่นำเสนอมีค่าสูงกว่าค่าความถูกต้องเฉลี่ยของ Fuzzy ARTMAP 0.03% เทียบเท่ากับค่าความถูกต้องเฉลี่ยของ LVQ สูงกว่าค่าความถูกต้องของ BPN 2.5 % และสูงกว่าค่าความถูกต้อง RBF 0.19%

4) ข้อมูล Fan ค่าความถูกต้องเฉลี่ยของแบบจำลองที่นำเสนอมีค่าต่ำกว่าค่าความถูกต้องเฉลี่ยของ Fuzzy ARTMAP 0.07% ต่ำกว่าค่าความถูกต้องเฉลี่ยของ LVQ 0.02% ต่ำกว่าค่าความถูกต้องของ BPN 0.07% และต่ำกว่าค่าความถูกต้องของ RBF 0.05 %

#### 4.4.2 สรุปผลการทดลองของชุดข้อมูลมาตรฐาน

1) ข้อมูล Iris ค่าความถูกต้องเฉลี่ยของแบบจำลองที่นำเสนอมีค่าเทียบเท่ากับค่าความถูกต้องเฉลี่ยของ LVQ, BPN, RBF และสูงกว่าค่าความถูกต้องเฉลี่ยของ Fuzzy ARTMAP 2%

2) ข้อมูล Wine ค่าความถูกต้องเฉลี่ยของแบบจำลองที่นำเสนอมีค่าสูงกว่าค่าความถูกต้องเฉลี่ยของ Fuzzy ARTMAP 1.37% สูงกว่าค่าความถูกต้องเฉลี่ยของ LVQ 2.96% สูงกว่าค่าความถูกต้องของ BPN 0.01% และสูงกว่าค่าความถูกต้องเฉลี่ยของ RBF 1.14%

3) ข้อมูล Ionosphere ค่าความถูกต้องเฉลี่ยของแบบจำลองที่นำเสนอมีค่าต่ำกว่าค่าความถูกต้องเฉลี่ยของ Fuzzy ARTMAP 1.59% สูงกว่าค่าความถูกต้องเฉลี่ยของ LVQ 1.33% สูงกว่าค่าความถูกต้องของ BPN 2.65% และต่ำกว่าค่าความถูกต้องเฉลี่ยของ RBF 2.65%

4) ข้อมูล Haberman ค่าความถูกต้องเฉลี่ยของแบบจำลองที่นำเสนอมีค่าสูงกว่าค่าความถูกต้องเฉลี่ยของ Fuzzy ARTMAP 2.18% สูงกว่าค่าความถูกต้องเฉลี่ยของ LVQ 1.71% สูงกว่าค่าความถูกต้องของ BPN 0.87% และสูงกว่าค่าความถูกต้องเฉลี่ยของ RBF 6.53%

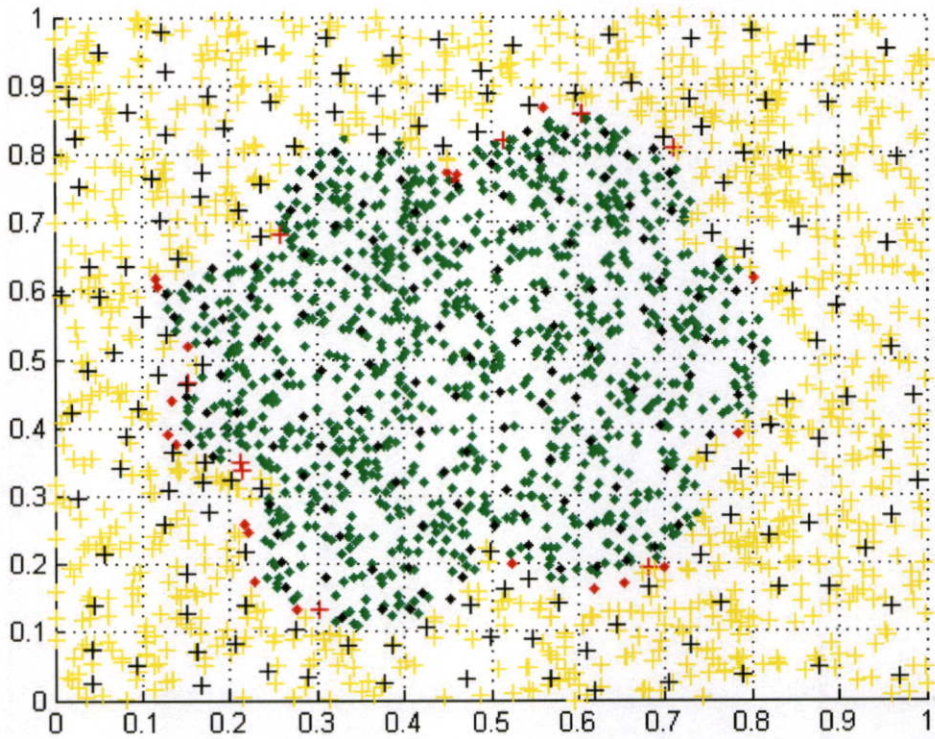
5) ข้อมูล Heart-Statlog ค่าความถูกต้องเฉลี่ยของแบบจำลองที่นำเสนอมีค่าสูงกว่าค่าความถูกต้องเฉลี่ยของ Fuzzy ARTMAP 0.74% ต่ำกว่าค่าความถูกต้องเฉลี่ยของ LVQ 3.4% สูงกว่าค่าความถูกต้องของ BPN 3.11% และต่ำกว่าค่าความถูกต้องเฉลี่ยของ RBF 4.44%

ดังนั้นเมื่อทำการเปรียบเทียบค่าเปอร์เซ็นต์ความถูกต้องพบว่าแบบจำลองที่นำเสนอมีค่าเปอร์เซ็นต์ความถูกต้องเมื่อเปรียบเทียบกับ Fuzzy ARTMAP, LVQ, BPN และ RBF ดังนี้

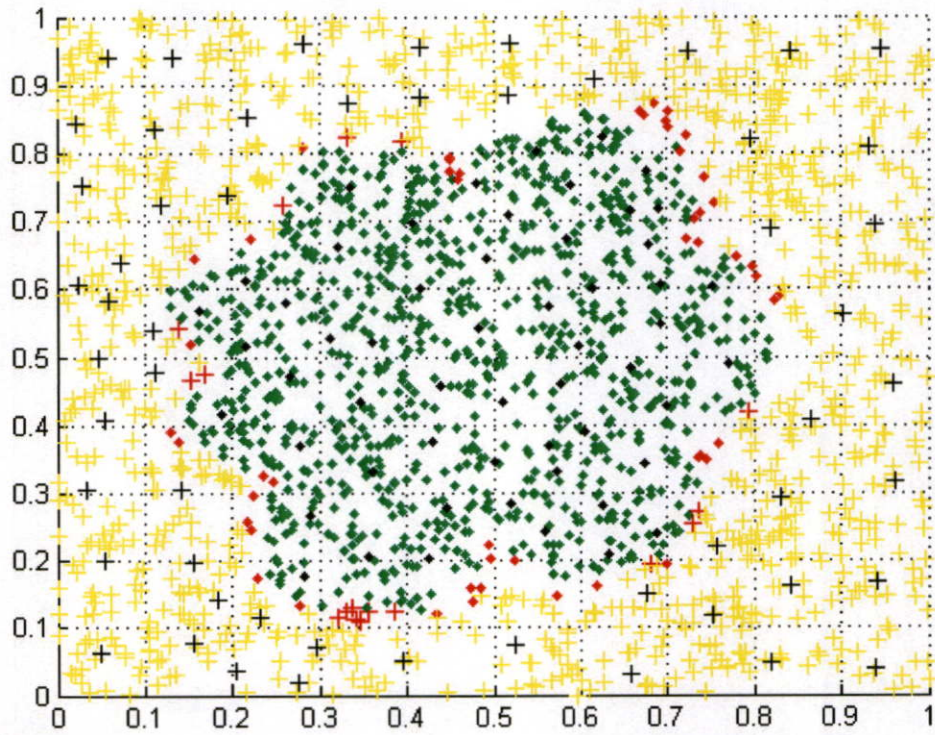
- สูงกว่า BPN จำนวน 7 ชุดข้อมูล คือ Flower1, Flower2, Sawtooth, Wine, Haberman, Heart-Statlog และ Ionosphere
- สูงกว่า RBF จำนวน 5 ชุดข้อมูล คือ Flower1, Flower2, Sawtooth, Wine, และ Haberman,
- สูงกว่า LVQ จำนวน 5 ชุดข้อมูล คือ Flower1, Flower2, Wine, Haberman, และ Ionosphere
- สูงกว่า Fuzzy ARTMAP จำนวน 7 ชุดข้อมูล คือ Flower1, Flower2, Sawtooth, Iris, Wine, Haberman และ Heart-Statlog
- เทียบเท่า LVQ จำนวน 2 ชุดข้อมูล คือ Sawtooth, Iris
- เทียบเท่า BPN และ RBF จำนวน 1 ชุดข้อมูล คือ Iris
- ต่ำกว่า Fuzzy ARTMAP จำนวน 2 ชุดข้อมูล คือ Fan, Ionosphere
- ต่ำกว่า LVQ จำนวน 2 ชุดข้อมูล คือ Fan และ Heart-Statlog
- ต่ำกว่า RBF จำนวน 3 ชุดข้อมูล คือ Fan, Ionosphere และ Heart-Statlog
- ต่ำกว่า BPN จำนวน 1 ชุดข้อมูล คือ Fan

เมื่อทำการเปรียบเทียบผลการทดลองของแบบจำลองที่นำเสนอกับ LVQ ซึ่งเป็นแบบจำลองที่มีลักษณะการทำงานใกล้เคียงกัน พบว่าแบบจำลองที่นำเสนอให้ผลการทดลองในชุดข้อมูลทดสอบ [14] ที่ดีกว่า LVQ จำนวน 2 ชุดข้อมูล คือ Flower 1, Flower 2 เทียบเท่ากับจำนวน 1 ชุดข้อมูล คือ Sawtooth, และต่ำกว่าจำนวน 1 ชุดข้อมูล คือ Fan ในกรณีที่ข้อมูลมีการกระจายของกลุ่มที่คลุมเครือแบบจำลองที่นำเสนออาจให้ผลการทำนายที่ผิดพลาดตรงช่วงของรอยต่อของแต่ละกลุ่มข้อมูล จากผลการทดลองที่ได้แสดงให้เห็นว่าแบบจำลองที่นำเสนอให้ผลการจัดหมวดหมู่ข้อมูลที่ดีกว่า LVQ ดังรูปแสดง Scatter plot ของชุดข้อมูลทดสอบของแต่ละวิธี รูปที่ 4.2 ถึง 4.9 โดยกำหนดให้

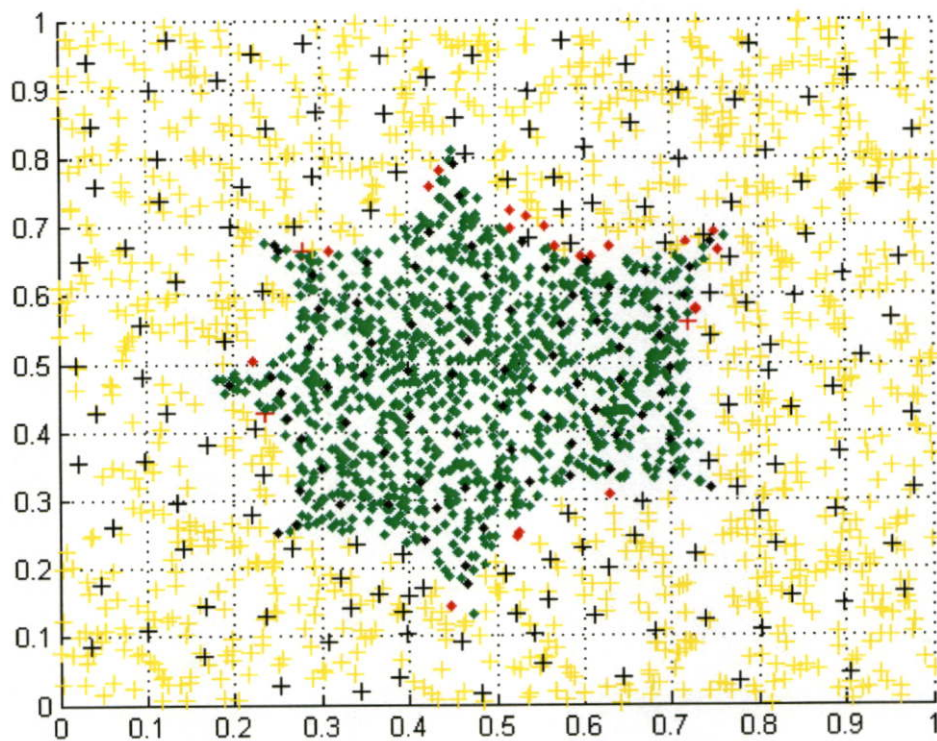
- ◆ แสดงถึงรูปแบบอินพุทในคลาสที่ 1
- ◆ แสดงถึงจุดศูนย์กลางของคลัสเตอร์ในคลาสที่ 1
- + แสดงถึงรูปแบบอินพุทในคลาสที่ 2
- + แสดงถึงจุดศูนย์กลางของคลัสเตอร์ในคลาสที่ 2
- \* แสดงถึงรูปแบบอินพุทในคลาสที่ 3
- \* แสดงถึงจุดศูนย์กลางของคลัสเตอร์ในคลาสที่ 3
- แสดงถึงรูปแบบอินพุทในคลาสที่ 4
- แสดงถึงจุดศูนย์กลางของคลัสเตอร์ในคลาสที่ 4
- ◆ แสดงถึงรูปแบบอินพุทในคลาสที่ 1 ที่ทำนายผิด
- + แสดงถึงรูปแบบอินพุทในคลาสที่ 2 ที่ทำนายผิด
- \* แสดงถึงรูปแบบอินพุทในคลาสที่ 3 ที่ทำนายผิด
- แสดงถึงรูปแบบอินพุทในคลาสที่ 4 ที่ทำนายผิด



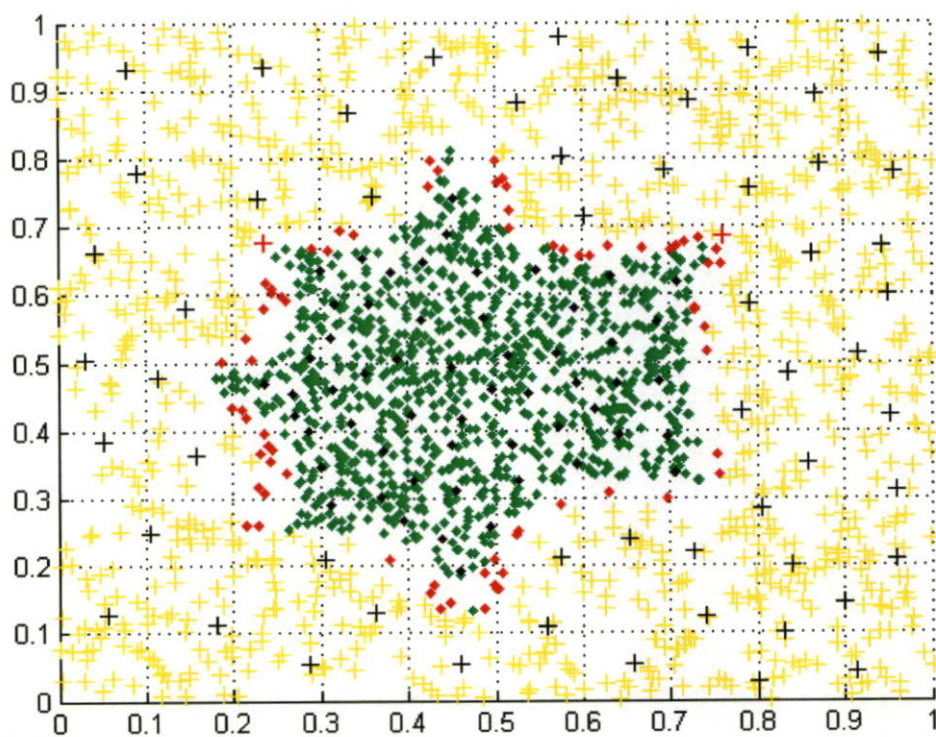
รูปที่ 4.2 Scatter Plot ของข้อมูลที่ใช้ในการทดสอบแบบจำลองที่นำเสนอของชุดข้อมูล Flower 1



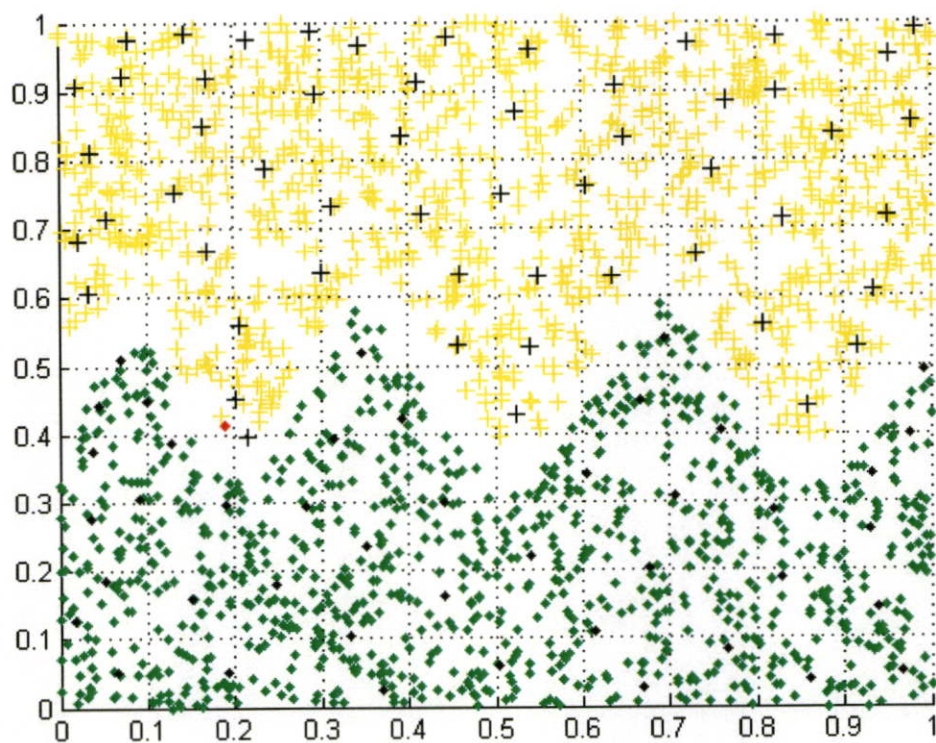
รูปที่ 4.3 Scatter Plot ของข้อมูลที่ใช้ในการทดสอบ LVQ ของชุดข้อมูล Flower 1



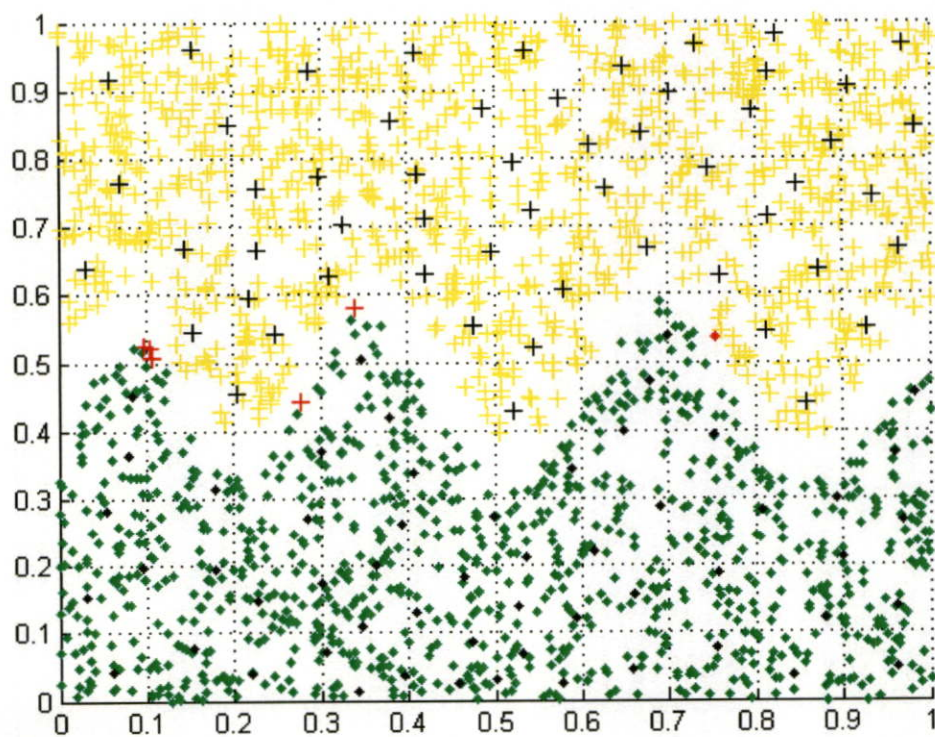
รูปที่ 4.4 Scatter Plot ของข้อมูลที่ใช้ในการทดสอบแบบจำลองที่นำเสนอของชุดข้อมูล Flower 2



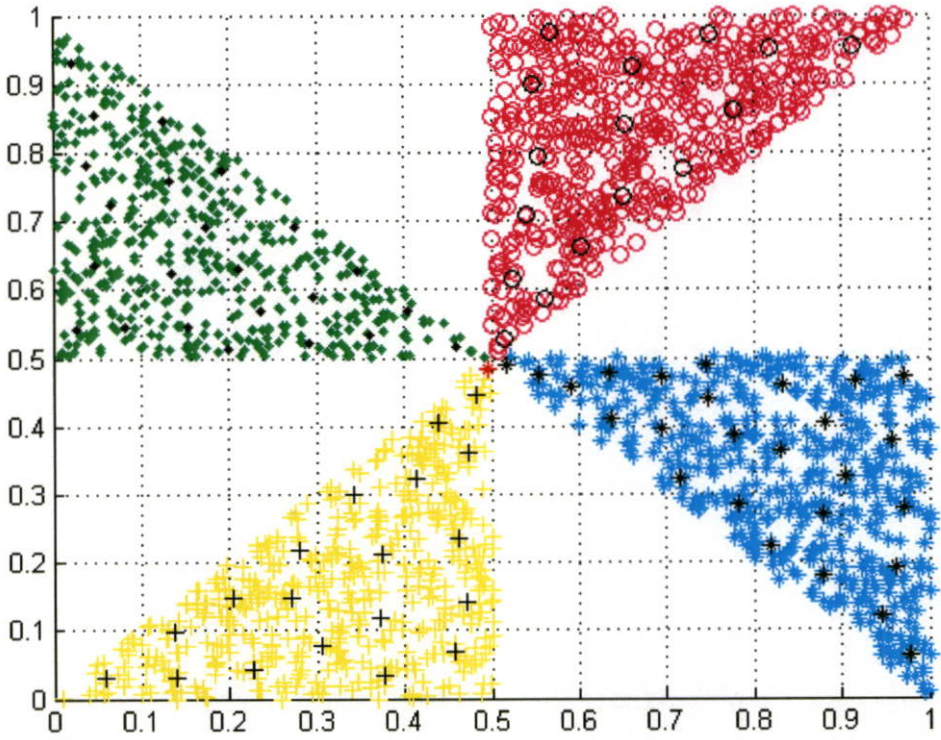
รูปที่ 4.5 Scatter Plot ของข้อมูลที่ใช้ในการทดสอบ LVQ ของชุดข้อมูล Flower 2



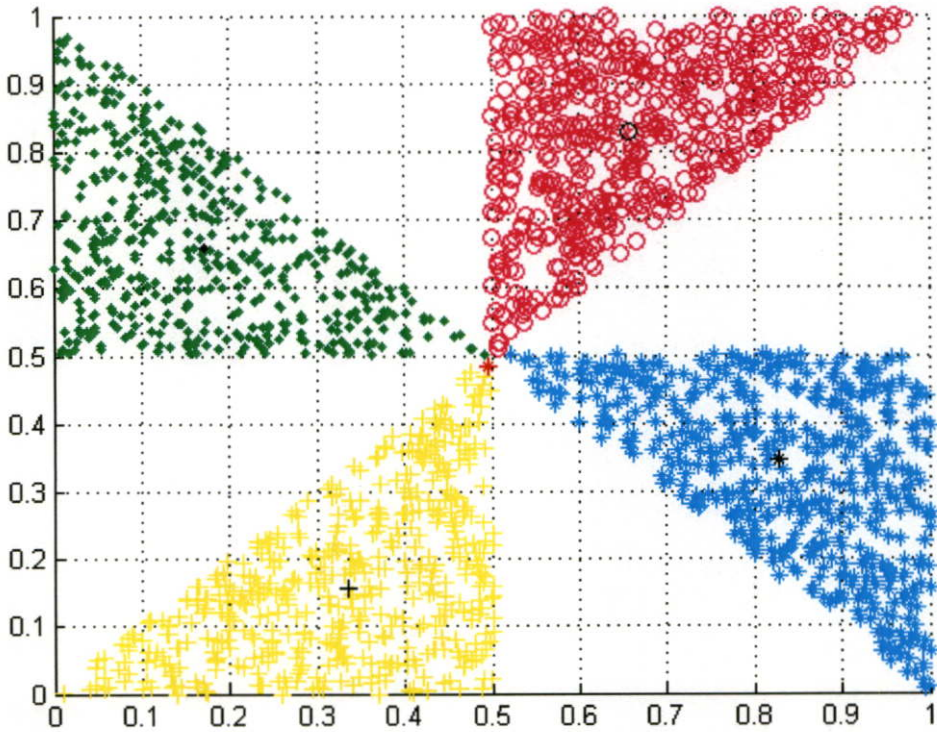
รูปที่ 4.6 Scatter Plot ของข้อมูลที่ใช้ในการทดสอบแบบจำลองที่นำเสนอของชุดข้อมูล Sawtooth



รูปที่ 4.7 Scatter Plot ของข้อมูลที่ใช้ในการทดสอบ LVQ ของชุดข้อมูล Sawtooth



รูปที่ 4.8 Scatter Plot ของข้อมูลที่ใช้ในการทดสอบแบบจำลองที่นำเสนอของชุดข้อมูล Fan



รูปที่ 4.9 Scatter Plot ของข้อมูลที่ใช้ในการทดสอบ LVQ ของชุดข้อมูล Fan

## บทที่ 5

# สรุปผลการวิจัย และข้อเสนอแนะ

### 5.1 สรุปผลงานวิจัย

งานวิจัยนี้ได้นำเสนอการพัฒนาแบบจำลองการจัดหมวดหมู่ข้อมูลแบบใหม่ โดยมีการนำโครงข่ายประสาทเทียมแบบการปรับตัวเองภายใน (Self Organizing Map) มาปรับให้มีความสามารถในการจัดหมวดหมู่ข้อมูลโดยนำหลักการค่าความเป็นสมาชิกของเซตของความคลุมเครือเข้ามาใช้ในการหาโหนดที่ชนะในขั้นตอนการแข่งขัน และในขั้นตอนการเรียนรู้ได้นำกลไกการเรียนรู้แบบมีผู้สอน (Supervised Learning) เข้ามาประยุกต์ใช้พร้อมทั้งมีกระบวนการที่สามารถเพิ่มลดจำนวนโหนดในชั้นเอาต์พุตได้ ซึ่งจากแนวคิดข้างต้นทำให้ได้มาซึ่งแบบจำลองการจัดหมวดหมู่ข้อมูลแบบใหม่ที่มีประสิทธิภาพในการทำนายกลุ่มของข้อมูลทำให้มีความถูกต้องในการทำนายที่ดีมากยิ่งขึ้น

หลักการทำงานของงานวิจัยที่นำเสนอเริ่มจากการสุ่มรูปแบบอินพุตมาจากแต่ละคลาส พร้อมทั้งกำหนดค่าพารามิเตอร์ต่างๆ ในขั้นตอนการหาโหนดที่ชนะในงานวิจัยใช้การพิจารณาจากค่าความเป็นสมาชิก (membership) ของอินพุตกับโหนดที่มีค่าสูงที่สุด ต่อไปจึงพิจารณาปรับค่าน้ำหนักโดยพิจารณาจากคลาสของโหนดที่ชนะกับคลาสของอินพุตแล้วจึงทำการปรับค่าน้ำหนักตามแต่ละกรณีที่กำหนดไว้ เมื่อทำการเรียนรู้จนครบชุดข้อมูลทดลอง (training data) จะทำการตรวจสอบการลบบคัสเตอร์โดยทำการคำนวณสัดส่วนค่า Entropy ของแต่ละคัสเตอร์แล้วนำค่าที่ได้ไปเปรียบเทียบกับค่า threshold การลบบคัสเตอร์ ( $thde$ ) ถ้าค่า Entropy ของคัสเตอร์ไหนมากกว่าค่า threshold การลบบคัสเตอร์ คัสเตอร์นั้นจะถูกลบไป ขั้นตอนทั้งหมดจะถูกทำซ้ำจนครบตามจำนวนรอบที่กำหนดไว้

ข้อมูลที่นำมาใช้ในขั้นตอนการเรียนรู้และทดสอบมี 2 ชุดคือข้อมูลทดสอบมีทั้งหมด 4 ชุดข้อมูลและข้อมูลที่นำมาจากฐานข้อมูล UCI จำนวน 5 ชุดข้อมูล โดยข้อมูลที่นำมาใช้เป็นข้อมูลที่เป็นตัวเลขที่ผ่านการ Normalize แล้ว

### 5.2 ปัญหาที่พบในงานวิจัย

- 1) พารามิเตอร์ที่ใช้ในการประมวลผลมีจำนวนมากดังนั้นจึงยากในการหาค่าพารามิเตอร์ที่เหมาะสมสำหรับและชุดข้อมูล
- 2) ในขั้นตอนการปรับลบบคัสเตอร์โดยการคำนวณสัดส่วนค่า Entropy ของแต่ละคัสเตอร์แบบจำลองต้องใช้เวลาในการประมวลผลค่อนข้างมาก

### 5.3 แนวทางการพัฒนาในอนาคต

ทำการทดลองเพิ่มเติมโดยเปลี่ยนใช้ฟังก์ชันความเป็นสมาชิกแบบอื่นๆ เช่น ฟังก์ชันสามเหลี่ยม, ฟังก์ชันสี่เหลี่ยมคางหมู, ฟังก์ชันรูประฆังคว่ำ, ฟังก์ชันรูปตัวเอส เป็นต้น

## บรรณานุกรม

- [1] Jiawei Han and Micheline Kamber. 2001, **Data Mining: Concept and Techniques**. San Francisco : Morgan Kaufmann (an imprint of Academic Press).
- [2] J-S.R. Jang, C-T. Sun, E.Mizutani. 1997 **Neuro- Fuzzy and Soft Computing A Computational Approach to Learning and Machine Intelligence**. Upper Saddle River, N.J. : Prentice Hall International
- [3] L.A. Zadeh “Fuzzy Sets.” Information and Control, vol. 8, 1965. pp. 338-353
- [4] Michael Negnevitsky. 2001 **Artificial Intelligence : A Guide to Intelligent Systems**. Addison-Wesley
- [5] T. Kasuba. “Simplified Fuzzy ARTMAP.” vol. 8, Nov, 1993. pp. 18-25. **AI Expert**
- [6] T. Kohonen. “The self-organizing map.” vol. 78, Issue. 9, Sept. 1990. pp. 1464-1480. **Proceedings of the IEEE**
- [7] Yen John. **Fuzzy logic : intelligence, control, and information**. Upper Saddle River, N.J. : Prentice Hall, c1999
- [8] Z. Kurd and T.P. Kelly. “Using Fuzzy Self-Organising Maps for Safety Critical Systems.” 2004. pp 17-30.
- [9] P.N. Suganthan. “Hierarchical Overlapped SOM’s for Pattern Classification.” in **IEEE Transactions on Neural Networks.**, vol. 10, no 1, January 1999. pp. 193-196.
- [10] Victor-Emil Neagoe, Armand-Dragos Ropot. “Concurrent Self-Organizing Maps for Pattern Classification.” in **Proceedings of the First IEEE International Conference on Cognitive Informatics (ICCI’02)**, pp. 304
- [11] M. Hagenbuchner and A.C. Tsoi. “A supervised training algorithm for self-organizing map for structures” in **Pattern Recognition Letters.**, vol. 26, Sep 2005. pp. 1874-1884 .
- [12] Hsu, A. and Halgamuge, S.K. 2006. “Semi-supervised Learning of Dynamic Self-Organising Maps.” pp. 915-924. in **ICONIP’06**. Springer Berlin.
- [13] Y Chen, MI Pazner and W Wu. 2007. “A Comparison between a Modified Counter Propagation Network and an Extended Self-Organizing Map in Remotely Sensed Data Classification.” **Mathematical Geology**. vol. 39, no 6, August 2007 pp. 559–574

- [14] จิตรภรณ์ มุลวงศ์. 2550. “การจัดกลุ่มข้อมูลโดยใช้วิวัฒนาการทางด้านพฤติกรรมทางสังคมของมนุษย์.” วิทยานิพนธ์วิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ บัณฑิตวิทยาลัย, สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง.

## ภาคผนวก

## ภาคผนวก ก

### ผลงานวิจัยที่ได้รับการตีพิมพ์เผยแพร่

A.Thammano and J.kiatwuthiamorn. "Solving Classification Problems Using Supervised Self-Organizing Map." **Proceedings of the 7<sup>th</sup> IEEE International Symposium on Signal Processing and Information Technology(ISSPIT2007).** Cairo, Egypt, 2007.

## Solving Classification Problems Using Supervised Self-Organizing Map

Arit Thammano\* and Jiraporn Kiatwuthiamorn\*\*

Computational Intelligence Laboratory  
Faculty of Information Technology

King Mongkut's Institute of Technology Ladkrabang,  
Bangkok, 10520 Thailand

E-mail: arit@it.kmitl.ac.th\* and j.kiatwuthiamorn@hotmail.com\*\*

**Abstract** - This paper proposes the new approach to deal with the classification problems by modifying the well-known Kohonen self-organizing map in order to make it able to solve classification problems. During training, the fuzzy membership function is used in place of the Euclidean distance to find the best matching cluster for the input pattern. In order to improve the efficiency of proposed model, the fuzzy entropy concept is employed to reduce the number of nodes in the cluster layer. The performance of the proposed model was compared with the fuzzy ARTMAP neural network. The results on five benchmark problems are very encouraging.

**Keywords** - Classification, Neural Network, Self-Organizing Map, Data Mining

### I. INTRODUCTION

Classification is the process of finding a set of models that describe and distinguish data classes or concepts, for the purpose of being able to use the model to predict the class of objects whose class label is unknown [1]. A variety of techniques have been applied to deal with the classification problems, such as artificial neural networks, a decision tree, and statistical methods. Among a number of classification methods in use, artificial neural networks are the most widely known. Many previous research works [2, 3, 4, 5, 6] show that neural network classifiers have better performance, lower classification error rate, and more robust to noise than other classification methods.

In this paper, the supervised self-organizing map is proposed. This proposed supervised SOM is the modification of the well-known Kohonen self-organizing map [7] in order to make it able to solve classification problems. The performance of the proposed method is evaluated against the fuzzy ARTMAP neural network, which is one of the best neural network classifiers available.

This paper is divided into 4 sections. Following this introduction, section 2 presents the architecture of the proposed supervised self-organizing map and its learning algorithm. A brief description of the experimental data and the experimental results are given in section 3. Finally, section 4 is the conclusions.

### II. SUPERVISED SELF-ORGANIZING MAP

The architecture of the proposed model is a three-layer feedforward neural network as shown in Figure 1. The first

layer is the input layer, which consists of  $N$  nodes. Each node represents a feature component of the input data. The second layer is the cluster layer. The nodes in this second layer are constructed during the training phase and each node represents a reference pattern. The third layer is the output layer. Each node in the output layer represents a class.

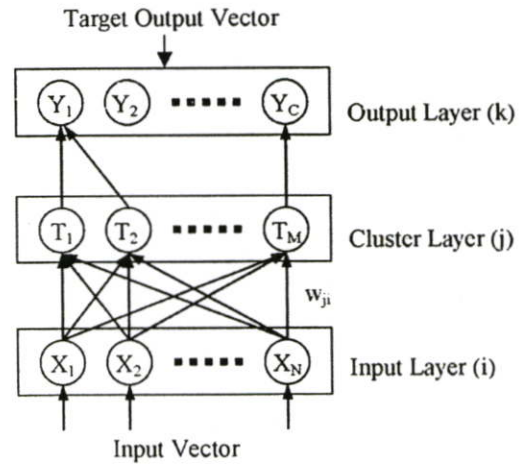


Fig. 1. Architecture of the Proposed Supervised SOM

At the beginning of the learning algorithm,  $q$  input patterns from the training data set of each class are arbitrarily selected, and assigned to be the initial cluster centers. Next, the input vector is presented to the model, together with its respective target output vector. The input vector is denoted by  $X = (x_1, x_2, \dots, x_N)$ , where  $N$  is the number of features in  $X$ . Each node in the second layer is fully connected to the nodes in the input layer via the connections  $w_{ji}$ . The weight vector  $w_j$  of dimension  $N$  represents the reference pattern of the  $j^{\text{th}}$  node in the cluster layer. Once the model receives the input and its associated target output  $(X, Y)$ , the input vector  $X$  is directly transmitted to the cluster layer. Each node in the cluster layer then calculates the membership degree to which the input vector  $X$  belongs to its cluster  $j$ .

$$T_j(X) = \frac{\sum_{i=1}^N \mu_j(x_i)}{N} \quad (1)$$

$$\mu_j(x_i) = \text{Gaussian}(x_i; w_{ji}, \sigma_{jl}, \sigma_{jr})$$

$$= \begin{cases} \exp\left[-\frac{(x_i - w_{ji})^2}{2\sigma_{jl}^2}\right], & x_i < w_{ji} \\ 1, & x_i = w_{ji} \\ \exp\left[-\frac{(x_i - w_{ji})^2}{2\sigma_{jr}^2}\right], & x_i > w_{ji} \end{cases} \quad (2)$$

where  $i = 1, 2, \dots, N$

$j = 1, 2, \dots, M$

$N$  is the number of features in input vector  $X$ .

$M$  is the number of nodes in cluster layer.

$\mu_j(x_i)$  is the membership degree to which the input data  $x_i$  belongs to each fuzzy membership function.

$\sigma_{jl}$  and  $\sigma_{jr}$  denote the width on the left side and on the right side of the  $j^{\text{th}}$  membership function, respectively.

Next, the system makes a cluster choice by selecting the winning node  $J$  with maximum choice function value from all the nodes  $j$  in the cluster layer.

$$J = \arg \max_j (\Gamma_j(X)) \quad (3)$$

If the winning node  $J$  belongs to the correct class defined by the target output vector, the weight vector of the winning node ( $w_j$ ) and those of its neighboring nodes whose classes are the same as the winning node  $J$  will be updated according to the following equation:

$$w_m(t+1) = w_m(t) + \Delta w_m(t) \quad (4)$$

$$\Delta w_m(t) = \alpha(N_m, t)[x_m(t) - w_m(t)] \quad (5)$$

$$\alpha(N_m, t) = \alpha(t) \exp\left(\frac{-\|r_m - r_j\|^2}{\sigma^2(t)}\right) \quad (6)$$

$$\alpha(t) = \exp[-(u_0 + 0.007(t-1))] \quad (7)$$

$$\sigma(t) = \exp[-(v_0 + 0.004(t-1))] \quad (8)$$

where  $\alpha(N_m, t)$  is the neighborhood function, which controls the amount by which the weights of the neighboring nodes are updated.

$\alpha(t)$  is the learning rate, which decreases exponentially to zero in time.

$\sigma(t)$  controls the spread of the neighborhood function. It is a decreasing function of time.

However, if the winning node does not represent the class to which  $X$  belongs, the system will search for the next best cluster node  $j^*$  whose class is the same as the target output. Next, the vigilance criterion in (9) is evaluated.

$$\mu_j(X) - \mu_{j^*}(X) < \rho \quad (9)$$

If the criterion in (9) is not satisfied, a new cluster node  $J$  is recruited and its weight is initialized using equation (10). Then the connection between a new cluster node and the target output is created.

$$w_j = X \quad (10)$$

On the contrary, if the criterion in (9) is met, the parameters which control each connection weight of the  $J$  and  $j^*$  nodes ( $w_{ji}$ ,  $\sigma_{ji}$ ,  $w_{j^*i}$ , and  $\sigma_{j^*i}$ ) are then updated according to Case 1: If both  $\mu_{ji}(X)$  and  $\mu_{j^*i}(X)$  are less than or equal to 0.01,

$$w_{j^*i}(t+1) = w_{j^*i}(t) + \eta[x_i(t) - w_{j^*i}(t)] \quad (11)$$

$$w_{ji}(t+1) = w_{ji}(t) - \eta[x_i(t) - w_{ji}(t)] \quad (12)$$

where  $\eta$  is the learning rate.

Case 2: If both  $\mu_{ji}(X)$  and  $\mu_{j^*i}(X)$  are greater than 0.01,

$$\sigma_{j^*i}(t+1) = \sigma_{j^*i}(t) + 0.05\sigma(0) \quad (13)$$

$$\sigma_{ji}(t+1) = \sigma_{ji}(t) - 0.05\sigma(0) \quad (14)$$

Case 3: If  $\mu_{ji}(X)$  is greater than 0.01 and  $\mu_{j^*i}(X)$  is less than 0.01,

$$w_{j^*i}(t+1) = w_{j^*i}(t) + \eta[x_i(t) - w_{j^*i}(t)] \quad (15)$$

$$\sigma_{ji}(t+1) = \sigma_{ji}(t) - 0.05\sigma(0) \quad (16)$$

Case 4: If  $\mu_{ji}(X)$  is less than  $\mu_{j^*i}(X)$ ,

$$w_{j^*i}(t+1) = w_{j^*i}(t) \quad (17)$$

$$\sigma_{j^*i}(t+1) = \sigma_{j^*i}(t) \quad (18)$$

$$w_{ji}(t+1) = w_{ji}(t) \quad (19)$$

$$\sigma_{ji}(t+1) = \sigma_{ji}(t) \quad (20)$$

Following the weight updating step, the training continues with the next input pattern. After all input patterns have been processed, the system proceeds with the cluster reduction step. The cluster reduction step attempts to reduce the number of nodes in the cluster layer by deleting the cluster nodes which satisfy the following equation:

$$\left(\frac{n_j}{P}\right)E_j > \beta \quad (21)$$

$$E_j = -\sum_{k=1}^C p_{jk} \log(p_{jk}) \quad (22)$$

$$P_{jk} = \frac{\sum_{X \in c_j, X \in c_k} T_j(X)}{\sum_{X \in c_j} T_j(X)} \quad (23)$$

where  $n_j$  is the number of input patterns in the  $j^{\text{th}}$  cluster.

$P$  is the total number of patterns in the training data set.

$E_j$  is the fuzzy entropy of the  $j^{\text{th}}$  cluster.

$C$  is the number of classes into which the  $P$  input patterns are divided.

During testing, each test vector is applied in turn and its class is predicted. The class whose cluster node returns the maximum output value is the result of the prediction.

### III. EXPERIMENTAL RESULTS

To test the performance, the proposed algorithm was benchmarked against the fuzzy ARTMAP neural network. The experiments were conducted on 5 benchmark data sets:

1. The first data set is the well-known iris data (Figure 2). The iris data [8] has been widely used in the classification problem. The sepal length, sepal width, petal length, and petal width of 150 iris flowers from 3 species (Iris-setosa, Iris-versicolor, and Iris-virginica) are measured in centimeters, and are used as the input of the problem. The training set contains 120 records, while the testing set contains 30 records.
2. The second data set is the Haberman's survival data (Figure 3). It is publicly available from the UCI machine learning database repository [9]. This dataset contains 306 cases from a study that was conducted between 1958 and 1970 at the University of Chicago's Billings Hospital on the survival of patients who had undergone surgery for breast cancer. Three numerical attributes are used to predict the output class (1 or 2). The class value of 1 is corresponded to "the patient survived 5 years or longer" and the value of 2 is corresponded to "the patient died within 5 years." There is no standard split between the training and testing sets. In this paper, the data was divided into a training set of 214 examples and a testing set of 92 examples.
3. The third data set (Figure 4) is the ionosphere data created by the Space Physics Group at Johns Hopkins University. The data is taken from the UCI machine learning database repository [9]. It contains the radar data collected by a system in Goose Bay, Labrador. This data set has 200 instances in the training set and 151 instances in the testing set. Each instance is described by 34 continuous attributes and belongs to one of two classes ("good" or "bad"). "Good" radar returns are those showing evidence of some type of structure in the ionosphere. "Bad" radar returns are those that do not.
4. The fourth data set (Figure 5) is the heart disease problem. It is retrieved from Statlog Project Datasets [10]. The

problem concerns the prediction of the absence (1) or presence (2) of heart disease given the results of various medical tests carried out on a patient. This data set contains 13 attributes and 270 records. There are 150 records of class 1 and 120 records of class 2. In this paper, the 270 records in the database were randomly divided into a training set of 135 records and a testing set of 135 records.

5. The fifth data set is the wine recognition data (Figure 6). It is retrieved from the UCI machine learning database repository [9]. This data is the result of a chemical analysis of wines grown in the same region but from three different cultivars. Thirteen continuous attributes are used to determine the type of wine (class 1, 2, or 3). There are 59 instances of class 1, 71 instances of class 2, and 48 instances of class 3. In this paper, the 178 instances in the database were randomly divided into a training set of 90 instances and a testing set of 88 instances.

5.1	3.5	1.4	0.2	Iris-setosa
4.9	3.0	1.4	0.2	Iris-setosa
7.0	3.2	4.7	1.4	Iris-versicolor
6.4	3.2	4.5	1.5	Iris-versicolor
6.3	3.3	6.0	2.5	Iris-virginica
5.8	2.7	5.1	1.9	Iris-virginica

Fig. 2. Samples of the Iris Data

33,58,10,1  
33,60,0,1  
34,58,30,1  
34,59,0,2  
34,66,9,2  
41,60,23,2

Fig. 3. Samples of the Haberman's Survival Data

1,0,0.99539,-0.05889,0.85243,0.02306,0.83398,-0.37708,  
1,0.03760,0.85243,-0.17755,0.59755,-0.44945,0.60536,  
-0.38223,0.84356,-0.38542,0.58212,-0.32192,0.56971,  
-0.29674,0.36946,-0.47357,0.56811,-0.51171,0.41078,  
-0.46168,0.21266,-0.34090,0.42267,-0.54487,0.18641,  
-0.45300,g  
1,0,1,-0.18829,0.93035,-0.36156,-0.10868,-0.93597,1,  
-0.04549,0.50874,-0.67743,0.34432,-0.69707,-0.51685,  
-0.97515,0.05499,-0.62237,0.33109,-1,-0.13151,  
-0.45300,-0.18056,-0.35734,-0.20332,-0.26569,  
-0.20468,-0.18401,-0.19040,-0.11593,-0.16626,  
-0.06288,-0.13738,-0.02447,b  
1,0,1,-0.45161,1,1,0.71216,-1,0,0,0,0,0,-1,0.14516,  
0.54094,-0.39330,-1,-0.54467,-0.69975,1,0,0,1,0.90695,  
0.51613,1,1,-0.20099,0.25682,1,-0.32382,1,b

Fig. 4. Samples of the Johns Hopkins University Ionosphere Database

70.0,1.0,4.0,130.0,322.0,0.0,2.0,109.0,0.0,2.4,2.0,3.0,  
3.0,2  
67.0,0.0,3.0,115.0,564.0,0.0,2.0,160.0,0.0,1.6,2.0,0.0,  
7.0,1  
57.0,1.0,2.0,124.0,261.0,0.0,0.0,141.0,0.0,0.3,1.0,0.0,  
7.0,2  
64.0,1.0,4.0,128.0,263.0,0.0,0.0,105.0,1.0,0.2,2.0,1.0,  
7.0,1  
74.0,0.0,2.0,120.0,269.0,0.0,2.0,121.0,1.0,0.2,1.0,1.0,  
3.0,1

Fig. 5. Samples of the Heart Disease Problem

14.23,1.71,2.43,15.6,127.2,8.3,0.6,0.28,2.29,5.64,1.04,  
3.92,1065,1  
12.37,0.94,1.36,10.6,88,1.98,0.57,0.28,0.42,1.95,1.05,  
1.82,520,2  
12.86,1.35,2.32,18,122,1.51,1.25,0.21,0.94,4.1,0.76,1.29,  
630,3

Fig. 6. Samples of the Wine Recognition Data

The performance is measured in terms of the classification accuracy. In each experiment, the parameters of both models –  $\rho_a$  and  $\beta$  for the fuzzy ARTMAP, and  $\rho$  and  $\beta$  for the proposed supervised SOM – are varied throughout their ranges in order to get the best out of the two models. Results of the experiments are shown in Table 1.

For the iris data (data set 1), both models – the fuzzy ARTMAP neural network and the proposed model – give a perfect accuracy (100%). These best prediction performance are obtained from the fuzzy ARTMAP neural network using the vigilance parameter ( $\rho_a$ ) = 0.5 and the learning rate ( $\beta$ ) = 0.1, and from the proposed model using  $\rho = 0.1$  and  $\beta = 0.01$ .

For the data set 2, the best performance of the fuzzy ARTMAP (75%) is obtained when the vigilance parameter is equal to 0.65 and  $\beta = 0.4$ , while the best performance of the supervised SOM (81.52%) is obtained when  $\rho = 0.07$  and  $\beta = 0.01$ .

For the data set 3, the fuzzy ARTMAP produces 2 misclassifications out of a total of 151 examples: this corresponds to an accuracy of 98.68%. The vigilance parameter and the learning rate are set at 0.8 and 0.6 respectively. On the other hand, the supervised SOM with  $\rho = 0.01$  and  $\beta = 0.09$  produces 4 misclassification (97.35%).

For the data set 4, the proposed model with  $\rho = 0.07$  and  $\beta = 0.06$  marginally outperforms the fuzzy ARTMAP with  $\rho_a = 0.5$  and  $\beta = 0.5$  (85.19% versus 84.44%).

For the data set 5, the supervised SOM obtains the accuracy of 98.86%, while the fuzzy ARTMAP obtains the accuracy of 97.73%. The vigilance parameter and the learning rate of the fuzzy ARTMAP are set at 0.7 and 1 respectively, while  $\rho$  and  $\beta$  of the supervised SOM are set at 0.15 and 0.1 respectively.

Table 1. Experimental Results

Data set	Fuzzy ARTMAP		Supervised Self-Organizing Map	
	Correct (%)	Cluster	Correct (%)	Cluster
1. Iris	100	4	100	4
2. Haberman	75	15	81.52	37
3. Ionosphere	98.68	94	97.35	64
4. Heart	84.44	35	85.19	31
5. Wine	97.73	18	98.86	6

#### IV. CONCLUSIONS

In this paper, a supervised version of the Kohonen self-organizing map is proposed and its performance is compared with the Fuzzy ARTMAP neural network. In comparison to the fuzzy ARTMAP, the time used for training the proposed algorithm is larger. However, its classification accuracy compared with fuzzy ARTMAP is found to be superior. In addition, the number of reference patterns of the proposed model is also less.

#### REFERENCES

- [1] J. Han and M. Kamber, *Data Mining: Concepts and Techniques*, Academic Press, 2001.
- [2] J. R. Quinlan, "Comparing Connectionist and Symbolic Learning Methods," *Proceedings of a Workshop on Computational Learning Theory and Natural Learning Systems*, Vol. 1, The MIT Press, 1994, pp. 445-456.
- [3] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, Prentice Hall, 1995.
- [4] J. W. Shavlik, R. J. Mooney, and G. G. Towell, "Symbolic and Neural Learning Algorithms: An Experimental Comparison," *Machine Learning*, Vol. 6, No. 2, 1991, pp. 111-143.
- [5] S. Danaher, G. Herries, T. Selige, and M. Mac Surtan, "A Comparison of the Characterisation of Agricultural Land Using Singular Value Decomposition and Neural Networks," *Neurocomputation in Remote Sensing Data Analysis*, Springer, 1997.
- [6] J. Lee, R. C. Weger, S. K. Sengupta, and R. M. Welch, "A Neural Network Approach to Cloud Classification," *IEEE Transactions on Geoscience and Remote Sensing*, Vol. 28, No. 5, 1990, pp. 846-855.
- [7] T. Kohonen, "The Self-Organizing Map," *Proceedings of the IEEE*, Vol. 78, No. 9, 1990, pp. 1464-1480.
- [8] R. A. Fisher, "The Use of Multiple Measurements in Taxonomic Problems," *Annual Eugenics*, Vol. 7, Part II, 1936, pp. 179-188.
- [9] C. L. Blake and C. J. Merz, *UCI Repository of machine learning databases* [<http://www.ics.uci.edu/~mllearn/MLRepository.html>], University of California, Department of Information and Computer Science, 1998.
- [10] Statlog Project Datasets. Retrieved March 28, 2003. From <http://www.luicc.up.pt/ML/statlog/datasets/heart/>.

## ประวัติผู้เขียน

ชื่อ	นางสาวจิราพร เกียรติวุฒิมร
วัน เดือน ปีเกิด	30 ตุลาคม 2521
ที่อยู่	เลขที่ 59 ซอยรามคำแหง 53/1 ถนนรามคำแหง แขวงหัวหมาก เขตบางกะปิ กทม. 10240
ประวัติการศึกษา	2544 วิทยาศาสตรบัณฑิต สาขาวิทยาการคอมพิวเตอร์ มหาวิทยาลัยรามคำแหง