

ยูนิฟายอะแดปทีฟเรโซแนนซ์ทฤษฎีสำหรับข้อมูลเชิงตัวเลขและข้อมูลข้อความ

UNIFIED ADAPTIVE RESONANCE THEORY NEURAL NETWORKS
FOR NUMERIC AND TEXT DATA

จุฑามาส ชอเรืองวิวัฒน์
JUTHAMAS CHORUENGWIWAT

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต

สาขาวิชาเทคโนโลยีสารสนเทศ

บัณฑิตวิทยาลัย

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ. 2549

ISBN 974-8308-19-7

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

ยูนิฟายอะแดปทีฟเรโซแนนซ์ทฤษฎีสำหรับข้อมูลเชิงตัวเลขและข้อมูลข้อความ

UNIFIED ADAPTIVE RESONANCE THEORY NEURAL NETWORKS
FOR NUMERIC AND TEXT DATA



จุฑามาส ฉอเรืองวิวัฒน์

JUTHAMAS CHORUENGWIWAT

ฉพ.

๗๖๒๘๕

๒๕๔๙

เลขหมู่.....
เลขทะเบียน..... 69057
วัน,เดือน,ปี..... 7 ก.พ. 2550

1158656A
.b.....
.i.....

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต

สาขาวิชาเทคโนโลยีสารสนเทศ

บัณฑิตวิทยาลัย

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ. 2549

ISBN 974-8308-19-7

**UNIFIED ADAPTIVE RESONANCE THEORY NEURAL NETWORKS
FOR NUMERIC AND TEXT DATA**

JUTHAMAS CHORUENGWIWAT

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

2006

ISBN 974-8308-19-7

COPYRIGHT 2006

SCHOOL OF GRADUATE STUDIES

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

หัวข้อวิทยานิพนธ์	ยูนิฟายอะแดปทีฟเรโซแนนซ์ทฤษฎีสำหรับข้อมูลเชิงตัวเลขและข้อมูลข้อความ
นักศึกษา	นางสาวจุฑามาส หอเรื่องวิวัฒน์
รหัสประจำตัว	44067416
ปริญญา	วิทยาศาสตรมหาบัณฑิต
สาขาวิชา	เทคโนโลยีสารสนเทศ
พ.ศ.	2549
อาจารย์ผู้ควบคุมวิทยานิพนธ์	รศ. ดร. วรพจน์ กรีสระเดช

บทคัดย่อ

งานวิจัยนี้นำเสนออัลกอริทึมใหม่สำหรับจัดกลุ่มข้อมูล โดยนำฟัซซี่อะแดปทีฟเรโซแนนซ์ทฤษฎี (Fuzzy Adaptive Resonance Theory - Fuzzy ART) ซึ่งเป็นโครงข่ายประสาทเทียมที่มีความสามารถในการจัดกลุ่มข้อมูลเชิงตัวเลข มาประยุกต์ร่วมกับการเปรียบเทียบความคล้ายคลึงของเอกสาร ทั้งนี้เพื่อให้อัลกอริทึมมีคุณสมบัติเรียนรู้ด้วยตนเองและใช้จัดกลุ่มข้อมูลได้ทั้งข้อมูลเชิงตัวเลขและข้อมูลข้อความ ซึ่งสามารถใช้กับข้อมูลข้อความได้โดยตรงและไม่มีการแปลงข้อความเป็นตัวเลข

ในงานวิจัยได้นำเสนอการทดสอบอัลกอริทึมโดยทำการจัดกลุ่มข้อมูลที่สร้างขึ้นเองและข้อมูลที่นิยมใช้ในการทดสอบประสิทธิภาพการจัดกลุ่ม ทั้งนี้ข้อมูลทุกชุดจะเป็นข้อมูลที่ทราบข้อมูลกลุ่ม (Class) เพื่อให้สามารถคำนวณค่า F-measure, Purity และ Entropy เพื่อวัดประสิทธิภาพของอัลกอริทึมที่นำเสนอ กับ K-means และ Hierarchical Clustering ซึ่งผลการประเมินประสิทธิภาพของอัลกอริทึมที่นำเสนอสามารถจัดกลุ่มชุดข้อมูลที่นำมาทดสอบได้และให้ผลใกล้เคียงกับการจัดกลุ่มด้วย K-means

Thesis Title	Unified Adaptive Resonance Theory Neural Networks for Numeric and Text Data
Student	Ms. Juthamas Choruengwiwat
Student ID.	44067416
Degree	Master of Science
Program	Information Technology
Year	2006
Thesis Advisor	Assoc. Prof. Dr. Worapoj Kreesuradej

ABSTRACT

This research proposes a new neural network, Unified Adaptive Resonance Theory Neural Networks (Unified ART), for clustering text and numeric data. The neural network is based on Fuzzy Adaptive Resonance Theory and document similarity concepts. Hence, Unified ART is an unsupervised neural network which has ability to clustering both text and numeric data. Unified ART can work directly on textual information without text transformation into a numeric value.

The research also conducts an experimental with generated data, zoo data and flags data. Each data-set is clustered with three algorithms: Unified ART, K-means and Hierarchical Clustering. The experimental results show that the proposed neural network has ability in clustering. An evaluation of the proposed algorithm is almost similar to K-means algorithm.

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงได้อย่างดีด้วยคำแนะนำและคำปรึกษาเกี่ยวกับแนวทางในการทำวิจัยจาก รศ. ดร. วรพจน์ กรีสระเดช ซึ่งเป็นอาจารย์ผู้ควบคุมวิทยานิพนธ์ เพื่อที่จะหาอัลกอริทึมใหม่ที่น่าสนใจในงานวิจัยนี้ ซึ่งได้ใช้ระยะเวลาในการศึกษาวิจัยปรับปรุงแก้ไขเป็นเวลาหลายปี ทั้งนี้ผู้วิจัยรู้สึกซาบซึ้งในความอนุเคราะห์จากท่านและขอกราบขอบพระคุณเป็นอย่างสูง ที่ให้โอกาสและชี้แนะเสมอมา ไว้ ณ ที่นี้

นอกจากนั้นข้าพเจ้าขอขอบคุณและเอย่ยนามถึงบุคคลต่างๆ ที่ได้มีส่วนร่วมในการสร้างงานวิจัยชิ้นนี้ให้เสร็จสมบูรณ์ขึ้นมาได้

ขอกราบขอบพระคุณบุพการี ผู้คอยสนับสนุนและให้กำลังใจข้าพเจ้าในการศึกษา แต่ครั้งยังเยาว์จวบจนปัจจุบัน

ขอขอบคุณพี่ๆ เพื่อนๆ นักศึกษาทุกคนที่ได้คอยให้ความช่วยเหลือและแนะนำในการทำทดลองต่างๆ

ขอขอบคุณพี่ๆ ทุกท่านที่ใส่ใจถามไถ่ในเรื่องการศึกษาและการทำงาน อันเป็นกำลังใจสำคัญ ขอขอบคุณพี่ๆ เพื่อนๆ บริษัทอินเทอร์เน็ตประเทศไทยจำกัด(มหาชน) ทุกคนที่ให้กำลังใจและช่วยผลักดันให้ข้าพเจ้าได้ทำให้งานวิจัยชิ้นนี้เสร็จสมบูรณ์

คุณค่าและประโยชน์อันพึงมีจากวิทยานิพนธ์ฉบับนี้ ผู้วิจัยขอมอบแด่ผู้มีพระคุณทุกท่าน

จุฑามาส ฉอเรืองวิวัฒน์

สารบัญ

	หน้า
บทคัดย่อภาษาไทย.....	I
บทคัดย่อภาษาอังกฤษ.....	II
กิตติกรรมประกาศ.....	III
สารบัญ.....	IV
สารบัญตาราง.....	VII
สารบัญรูป.....	VIII
บทที่ 1 บทนำ.....	1
1.1 จุดประสงค์ของการศึกษา.....	2
1.2 สมมุติฐานของการศึกษา.....	2
1.3 ขอบเขตของการศึกษา.....	3
1.4 ขั้นตอนของการศึกษา.....	3
บทที่ 2 ทฤษฎีพื้นฐานและงานวิจัยที่เกี่ยวข้อง	4
2.1 การจัดกลุ่ม	4
2.1.1 Hierarchical Clustering Algorithms	5
2.1.2 Partitional Clustering Algorithms	5
2.1.3 Artificial Neural Network for Clustering	6
2.2 โครงข่ายประสาทเทียม (Neural Networks)	7
2.3 โครงข่ายประสาทเทียมแบบอะแดปทีฟเรโซแนนทีเยรี	9
2.3.1 โครงข่ายประสาทเทียมแบบอะแดปทีฟเรโซแนนทีเยรี 1	10
2.3.2 โครงข่ายประสาทเทียมแบบพีชชีอะแดปทีฟเรโซแนนทีเยรี	12
2.4 การเปรียบเทียบความคล้ายคลึงของเอกสาร.....	14
2.4.1 การคำนวณสำหรับข้อมูลเชิงปริมาณที่เป็นช่วง	15
2.4.2 การคำนวณสำหรับข้อมูลเชิงปริมาณที่เป็นข้อมูลตัวเลข (ทศนิยมและจำนวนเต็ม).....	15
2.4.3 การคำนวณสำหรับข้อมูลเชิงคุณภาพ (ข้อความ)	16
2.5 เท็กอะแดปทีฟเรโซแนนทีเยรีนิวโรลเน็ตเวิร์ค.....	17

สารบัญ(ต่อ)

	หน้า
บทที่ 3 ยูนิฟายอะแคปทีเพเรโซแนนทีวี่.....	20
3.1 สถาปัตยกรรมของโครงข่ายยูนิฟายอะแคปทีเพเรโซแนนทีวี่	20
3.1.1 โครงสร้างของโครงข่าย	20
3.1.2 ค่าถ่วงน้ำหนักของโครงข่าย	23
3.1.3 การเรียนรู้ของโครงข่าย	23
3.1.4 ฟังก์ชันการเปรียบเทียบความคล้ายคลึง	25
3.1.4.1 การเปรียบเทียบความคล้ายคลึงสำหรับข้อมูลที่เป็นตัวเลข	25
3.1.4.2 การเปรียบเทียบความคล้ายคลึงสำหรับข้อมูลที่เป็นข้อความ	25
3.1.5 ฟังก์ชันการปรับค่าถ่วงน้ำหนัก	27
3.2 อัลกอริทึมที่ใช้ในงานวิจัย	29
3.3 ตัวอย่างการเรียนรู้ของโครงข่าย	33
3.3.1 การเรียนรู้ของโครงข่าย	34
บทที่ 4 การทดลองและผลการทดลอง	41
4.1 เกณฑ์การวัดประสิทธิภาพ	42
4.1.1 Entropy	42
4.1.2 Purity	42
4.1.3 F-Measure	42
4.1.4 Euclidean Distance	43
4.2 การทดลองและผลการทดลอง	43
4.2.1 ข้อมูล Generate 1500 ชุด	43
4.2.2 ข้อมูล Zoo	47
4.2.3 ข้อมูล Flags	52
4.3 สรุปผลการทดลอง	57
บทที่ 5 สรุปผลการวิจัยและข้อเสนอแนะ	61
5.1 สรุปผลงานวิจัย	61
5.2 ข้อเสนอแนะและแนวทางในการทำวิจัยต่อ.....	62

สารบัญ (ต่อ)

	หน้า
เอกสารอ้างอิง	63
ภาคผนวก	65
ประวัติผู้เขียน.....	75

สารบัญตาราง

ตารางที่	หน้า
2.1 สัญลักษณ์ที่ใช้ใน ART	14
2.2 สัญลักษณ์ที่ใช้ในการเปรียบเทียบความคล้ายคลึงสำหรับข้อมูลตัวเลข	16
2.3 สัญลักษณ์ที่ใช้ในการเปรียบเทียบความคล้ายคลึงของข้อความ	17
3.1 สัญลักษณ์ที่ใช้ในยูนิฟายอะแคปทีพีเร โซแน้นเทียร์	22
3.2 ตัวอย่างค่าถ่วงน้ำหนักของโครงข่ายที่รับข้อมูลข้อความ	27
3.3 ตัวอย่างค่าถ่วงน้ำหนักของโครงข่ายที่รับข้อมูลข้อความและเชิงตัวเลข	28
3.4 ข้อมูล Micro-computer	33
3.5 คลัสเตอร์จากการเรียนรู้ของโครงข่าย	40
4.1 แสดงตัวอย่างข้อมูล 10 ตัวอย่างแรกของข้อมูลชุด Generate 1500 ชุด	45
4.2 แสดงผลการจัดกลุ่มข้อมูล Generate 1500 ชุดด้วย.....	46
4.3 แสดงผลการจัดกลุ่มข้อมูล Generate 1500 ชุดด้วยK-means	46
4.4 แสดงผลการจัดกลุ่มข้อมูล Generate 1500 ชุดด้วย Hierarchical	46
4.5 แสดงตัวอย่างข้อมูล 10 ตัวอย่างแรกของข้อมูลชุด Zoo พีเจอร์ที่ 1-8	48
4.6 แสดงตัวอย่างข้อมูล 10 ตัวอย่างแรกของข้อมูลชุด Zoo พีเจอร์ที่ 9-16	48
4.7 แสดงผลการจัดกลุ่มข้อมูล Zoo ด้วยยูนิฟายอะแคปทีพีเร โซแน้นเทียร์	50
4.8 แสดงผลการจัดกลุ่มข้อมูล Zoo ด้วย K-means	50
4.9 แสดงผลการจัดกลุ่มข้อมูล Zoo ด้วย Hierarchical	51
4.10แสดงตัวอย่างข้อมูล 10 ตัวอย่างแรกของข้อมูลชุด Flags พีเจอร์ที่ 1-6	52
4.11แสดงตัวอย่างข้อมูล 10 ตัวอย่างแรกของข้อมูลชุด Flags พีเจอร์ที่ 7-14	53
4.12แสดงตัวอย่างข้อมูล 10 ตัวอย่างแรกของข้อมูลชุด Flags พีเจอร์ที่ 15-21	53
4.13แสดงตัวอย่างข้อมูล 10 ตัวอย่างแรกของข้อมูลชุด Flags พีเจอร์ที่ 22-29	54
4.14แสดงผลการจัดกลุ่มข้อมูล Flags ด้วยยูนิฟายอะแคปทีพีเร โซแน้นเทียร์	55
4.15แสดงผลการจัดกลุ่มข้อมูล Flags ด้วย K-means	56
4.16แสดงผลการจัดกลุ่มข้อมูล Flags ด้วย Hierarchical	56
4.17แสดงผลการประเมินประสิทธิภาพของอัลกอริทึมต่างๆ	57
4.18แสดงจำนวนพีเจอร์หลังการแปลงข้อมูลสำหรับการทดลองแต่ละอัลกอริทึม	58
4.19 แสดงค่า Gain Ratio ของพีเจอร์ต่างๆในชุดข้อมูล Zoo	58

สารบัญรูป

รูปที่	หน้า
2.1 ตัวอย่างการจัดกลุ่มข้อมูล	4
2.2 Agglomerative และ Divisive hierarchical clustering	5
2.3 K-means Algorithm	6
2.4 ตัวอย่างการจัดกลุ่มข้อมูลด้วยวิธี K-means	6
2.5 เซลล์ประสาท.....	7
2.6 โครงข่ายประสาทเทียม	8
2.7 โครงสร้างของ ART1	10
2.8 อัลกอริทึมของ ART1	11
2.9 โครงสร้างของ Fuzzy ART	12
2.10 อัลกอริทึมของ Fuzzy ART	13
2.11 โครงสร้างของเท็กอะแคปทีพเร โชนั้นเท็กรีนิวรอลเน็ตเวิร์ค.....	18
3.1 โครงข่ายที่ใช้ในงานวิจัย	21
3.2 อัลกอริทึมที่ใช้ในงานวิจัย	30
4.1 การกระจายข้อมูลของพีเจอร์ที่เป็นข้อมูลข้อความ	44
4.2 การกระจายข้อมูลของพีเจอร์ที่เป็นข้อมูลเชิงตัวเลข	44
4.3 แสดงผลการประเมินประสิทธิภาพของการจัดกลุ่มข้อมูล Generate 1500 ชุด	47
4.4 แสดงการยุบรวมพีเจอร์	49
4.5 แสดงผลการประเมินประสิทธิภาพของการจัดกลุ่มข้อมูล Zoo	51
4.6 แสดงผลการประเมินประสิทธิภาพของการจัดกลุ่มข้อมูล Flags	57

บทที่ 1

บทนำ

โลกธุรกิจปัจจุบันเป็นระบบการตลาดที่มีความซับซ้อนและการแข่งขันที่สูงขึ้นกว่าแต่ก่อน การวางแผนการจัดการต่างๆจึงมีความจำเป็นต้องอาศัยข้อมูลที่รวบรวมไว้ ทั้งนี้การจัดการกับข้อมูลที่มีอยู่เพื่อให้ได้ประโยชน์สูงสุดจากข้อมูลนั้นถือว่ามีความสำคัญกว่าการเก็บข้อมูลปริมาณมากแต่ไม่สามารถนำมาใช้ประโยชน์ได้ ตัวอย่างเช่น การบริหารลูกค้าสัมพันธ์ (Customer Relationship Management-CRM) เป็นแนวทางการนำเอาข้อมูลมาใช้ประโยชน์ในการนำไปวางแผนการบริหารการตลาดและการบริหารความเสี่ยง โดยใช้วิธีการเก็บข้อมูลส่วนตัวลูกค้าและข้อมูลการใช้บริการของลูกค้ามาผ่านกระบวนการวิเคราะห์ลักษณะการใช้บริการและความสัมพันธ์ของข้อมูล เพื่อการวางแผนส่งเสริมการขายต่อไป ซึ่งวิธีการวิเคราะห์ที่นิยมใช้คือการทำ Data Mining เพื่อหารูปแบบและแนวโน้มของข้อมูล โดยเทคนิคที่นำมาใช้ในการทำ Data Mining ที่นิยมก็คือการจัดกลุ่มข้อมูลที่เรียกว่า การทำ Database Segmentation หรือ Clustering ซึ่งมีวัตถุประสงค์เพื่อการจัดกลุ่มข้อมูลที่ใกล้เคียงกันออกเป็น ส่วน (Segment/Cluster) โดยข้อมูลที่อยู่ในส่วนนั้นจะต้องมีความสัมพันธ์ภายในส่วนนั้นมาก (Homogeneity) และข้อมูลที่อยู่คนละส่วนกันจะต้องมีความสัมพันธ์กันน้อย (Heterogeneity)

การทำ Clustering เป็นการพยายามจัดกลุ่มข้อมูลที่มีอยู่ตามลักษณะเด่นของข้อมูลชุดนั้น เนื่องจากการทำ Mining จะทำให้เราสามารถเห็นข้อเท็จจริงได้เอง โดยอัลกอริทึมที่มีความสามารถในการจัดกลุ่มข้อมูลที่เป็นเวกเตอร์ เช่น K-means ซึ่งใช้ค่าเฉลี่ยเป็นตัวแทนกลุ่ม ดังนั้นการวัดความแตกต่างจะเป็นความแตกต่างจากตัวแทนกลุ่ม ส่วนวิธี Hierarchical Clustering เป็นการจัดข้อมูลแบบลำดับชั้น ซึ่งมี 2 แบบคือเป็นแบบ bottom-up และ top-down ทั้งนี้การประมวลผลค่อนข้างใช้เวลามากและไม่เหมาะกับข้อมูลปริมาณมาก วิธีที่นิยมและมีการคิดค้นพัฒนาต่อจำนวนมากคือการนำโครงข่ายประสาทเทียม (Neural Networks) มาใช้ ซึ่งโครงข่ายประสาทเทียมที่มีความสามารถในการจัดกลุ่มที่นิยมนำมาพัฒนาได้แก่ Kohonen Neural Networks, Adaptive Resonance Theory เป็นต้น

อัลกอริทึมพื้นฐานส่วนใหญ่จะมีความสามารถในการจัดกลุ่มข้อมูลที่เป็นไบนารีหรือข้อมูลที่เป็นเชิงตัวเลข (Number/Analog) เช่น งานวิจัยของ Carpenter G. A., Grossberg S. and Rosen D. A. [5] ที่นำเสนอ Fuzzy ART ที่สามารถใช้กับข้อมูลที่เป็นตัวเลขได้ ส่วนอัลกอริทึมสำหรับข้อมูลข้อความ (Text) ก็มีการพัฒนาความสามารถอัลกอริทึมในเรื่องการจัดกลุ่ม โดยยุคแรกๆจะเป็นการแปลงรูปข้อมูลข้อความนั้นให้เป็นข้อมูลเชิงตัวเลข ซึ่งอาจทำให้เสียความหมายบางอย่างของข้อมูลและใช้เวลามากในการแปลงรูปข้อมูล งานวิจัยต่อมาจึงพัฒนาให้สามารถนำเข้า

ข้อมูลข้อความได้โดยตรง ตัวอย่างเช่นงานวิจัยของ ทรงพล ชูติพงศ์พัฒนกุล[13] ได้นำเสนอวิธีการใช้ Kohonen Neural Networks ประยุกต์ร่วมกับการหาความต่างของเอกสาร[4] งานวิจัยของ สมคิด แสนเสนาะ[14] ได้นำเสนอวิธีการใช้โครงสร้างของ Competitive Learning Neural Networks algorithm มาใช้ร่วมกับการหาความต่างของเอกสาร[4] และงานวิจัยของ นรเศรษฐ์ จันทสูตร ที่นำเสนอการใช้ Adaptive Resonance Theory Neural Networks ร่วมกับการหาความเหมือนของเอกสาร

ทั้งนี้รูปแบบปกติของข้อมูลนั้นมักจะมีหลากหลาย คือ 1 ชุดของข้อมูลนั้นมักจะประกอบไปด้วยข้อมูลที่เป็นข้อความและตัวเลข ซึ่งในส่วนที่เป็นข้อมูลข้อความเช่น ส่วนประกอบของสินค้า ที่อยู่ของลูกค้า และข้อมูลเชิงตัวเลข เช่น ปริมาณของสารประกอบในสินค้า จำนวนบุตรของลูกค้า เป็นต้น งานวิจัยนี้จึงประยุกต์ความสามารถของ Adaptive Resonance Theory ที่มีความสามารถในการจัดกลุ่มข้อมูลมาใช้งานร่วมกับ Similarity Measure เพื่อใช้ในการจัดกลุ่มข้อมูลข้อความ และรวมเอาความสามารถของ Fuzzy ART ในการจัดกลุ่มข้อมูลเชิงตัวเลข มาเป็นอัลกอริทึมที่มีความสามารถในการจัดกลุ่มข้อมูลทั้ง 2 รูปแบบได้

1.1 จุดประสงค์ของการศึกษา

ข้อมูลมักจะประกอบไปด้วยข้อมูลข้อความและเชิงตัวเลข ในขณะที่อัลกอริทึมในการจัดกลุ่มโดยทั่วไปมักจะใช้วิธีการตัดทอนข้อมูลบางส่วนหรือการแทนข้อมูลด้วยตัวแทนอาจทำให้สูญเสียความหมายของข้อมูล ดังนั้นเพื่อให้กระบวนการทำงานของการจัดแบ่งกลุ่มข้อความมีประสิทธิภาพดีและได้ผลลัพธ์ที่ถูกต้อง วิธีการและอัลกอริทึมที่นำเสนอในงานวิจัยนี้จึงมุ่งเน้นเกี่ยวกับ อัลกอริทึมสำหรับการจัดแบ่งกลุ่ม (Clustering Algorithms) เพื่อปรับปรุงและออกแบบอัลกอริทึมใหม่สำหรับการจัดแบ่งกลุ่มที่ใช้ได้กับข้อมูลข้อความและเชิงตัวเลขโดยไม่จำเป็นต้องแปลงข้อมูล เพื่อนำเสนออัลกอริทึมใหม่ที่มีความสามารถในการจัดกลุ่มข้อมูลได้โดยตรงเพื่อลดการสูญเสียของข้อมูลจากการแทนค่า

1.2 สมมุติฐานของการศึกษา

การจัดกลุ่มข้อมูลที่มีความนิยมมากวิธีการหนึ่งคือ โครงข่ายประสาทเทียมซึ่งมีการเรียนรู้จดจำ เช่นเดียวกับมนุษย์ ซึ่งเป็นเทคโนโลยีที่เป็นที่นิยมในการนำมาประยุกต์ใช้งาน โดยโครงข่ายประสาทเทียมอะแดปทีฟเรโซแนนซ์ทฤษฎี (Adaptive Resonance Theory) เป็นโครงข่ายที่พัฒนาขึ้นมาใช้กับการจัดกลุ่มอย่างต่อเนื่องตั้งแต่ปี ค.ศ. 1987 โดยมีการพัฒนาให้ประยุกต์ร่วมกับวิธีการต่างๆ ให้สามารถรับข้อมูลอินพุตได้หลากหลายมากขึ้น จากข้อมูลไบนารีเป็นข้อมูลเชิงตัวเลข จนปัจจุบันมีงานวิจัยที่สามารถรับข้อมูลข้อความได้โดยตรง

ทั้งนี้งานวิจัยที่ผ่านมายังคงเป็นการรับข้อมูลอินพุตที่มีประเภทเดียว นั่นคือข้อมูลอินพุตจะมีลักษณะเป็นข้อมูลเชิงตัวเลขทั้งหมดหรือข้อความทั้งหมด

งานวิจัยนี้จึงนำโครงข่ายประสาทเทียมอะแดปทีฟเรโซแนนซ์ทฤษฎี (Adaptive Resonance Theory) มาประยุกต์ร่วมกับตัววัดความคล้าย (Similarity Measure) เพื่อพัฒนาอัลกอริทึมใหม่ที่มีความสามารถในการจัดกลุ่มสำหรับข้อมูลข้อความและเชิงตัวเลข

1.3 ขอบเขตของการศึกษา

งานวิจัยนี้ศึกษาการแบ่งกลุ่มข้อมูลที่เป็นข้อความและเชิงตัวเลข โดยไม่รวมข้อมูลที่เป็นช่วงเช่น 12-32 เช่นข้อมูลอายุในชุดข้อมูลพื้นฐานบุคคลในการตอบแบบสอบถาม ทั้งนี้หากต้องการใช้กับอัลกอริทึมที่นำเสนอจะต้องแปลงค่าข้อมูลโดยอาจจะใช้ค่าเฉลี่ยแทน

1.4 ขั้นตอนของการศึกษา

วิทยานิพนธ์ฉบับนี้แบ่งเนื้อหาออกได้เป็น 5 บทด้วยกันคือ

บทที่ 1 กล่าวถึงความเป็นมาของงานวิจัย ความมุ่งหมายและวัตถุประสงค์ สมมติฐาน ทฤษฎีที่ใช้ ขอบเขตของการวิจัย และขั้นตอนการศึกษา

บทที่ 2 กล่าวถึงทฤษฎีและงานวิจัยที่เกี่ยวข้องในงานวิจัย โดยนำเสนอวิธีการจัดกลุ่มวิธีการต่างๆ รวมทั้งโครงข่ายประสาทเทียมและพื้นฐานของอะแดปทีฟเรโซแนนซ์ทฤษฎี (Adaptive Resonance Theory) การเปรียบเทียบความคล้ายคลึงของเอกสารด้วยตัววัดความคล้าย (Similarity Measure) ซึ่งเป็นทฤษฎีหลักที่ใช้นำมาประยุกต์ใช้กับงานวิจัย

บทที่ 3 กล่าวถึงยูนิฟายอะแดปทีฟเรโซแนนซ์ทฤษฎี (Unified Adaptive Resonance Theory) ในส่วนของหลักการ โครงสร้าง รวมถึงวิธีการเรียนรู้ของโครงข่าย

บทที่ 4 กล่าวถึงการทดสอบโครงข่ายกับชุดข้อมูลต่างๆ รวมทั้งการประเมินประสิทธิภาพของผลการจัดกลุ่ม โดยได้นำเสนอการทดสอบกับอัลกอริทึมที่นำเสนอในงานวิจัย และการทดสอบการจัดกลุ่มด้วยอัลกอริทึมอื่น

บทที่ 5 เป็นบทสรุปผลการวิจัยและข้อเสนอแนะ

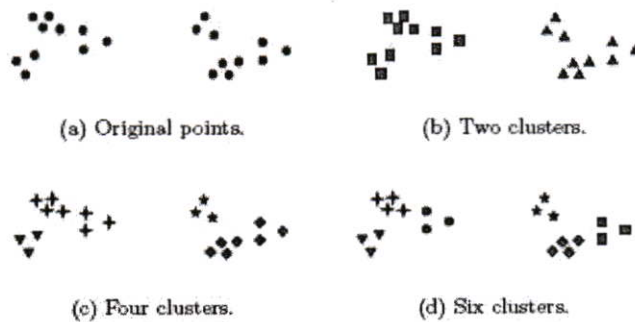
บทที่ 2

ทฤษฎีพื้นฐานและงานวิจัยที่เกี่ยวข้อง

ในหัวข้อนี้จะกล่าวถึงทฤษฎีและงานวิจัยที่เกี่ยวข้องที่เป็นพื้นฐานในการพัฒนางานวิจัย ซึ่งจะกล่าวถึง การจัดกลุ่ม โครงข่ายประสาทเทียม โครงสร้างของอะแดปทีฟเรโซแนนซ์ทฤษฎีเครือข่าย (Adaptive Resonance Theory Networks) และการเปรียบเทียบความเหมือนด้วย Similarity Measure

2.1 การจัดกลุ่ม

การจัดกลุ่มด้วยวิธีการคลัสเตอร์ริง (Clustering) มีจุดมุ่งหมายหลักคือการจัดกลุ่มของข้อมูลทั้งหมด โดยใช้คุณสมบัติของข้อมูลเป็นตัวกำหนดว่าข้อมูลใดควรจัดให้อยู่กลุ่มเดียวกันหรือต่างกลุ่มกัน ดังนั้นการจัดกลุ่มด้วยการคลัสเตอร์ริงจึงถือว่าเป็นการทำงานแบบไม่มีผู้สอน (Unsupervised learning) โดยที่การกำหนดพารามิเตอร์ที่แตกต่างกันก็ส่งผลให้ได้กลุ่มของข้อมูลที่แตกต่างกันด้วย



รูปที่ 2.1 ตัวอย่างการจัดกลุ่มข้อมูล[17]

จากรูปที่ 2.1 เป็นตัวอย่างของการจัดกลุ่มซึ่งจะพบว่าจากกลุ่มข้อมูลที่แสดงใน (a) นั้นสามารถจัดกลุ่มเป็นตามรูป (b), (c) และ (d) โดยในการแสดงผลการจัดกลุ่มได้ใช้รูปทรงเพื่อแสดงว่าข้อมูลใดเป็นกลุ่มเดียวกัน เช่นในการจัดกลุ่มด้วย K-means ในกรณีนี้ก็จะเห็นว่าได้ผลลัพธ์ของการจัดกลุ่มที่แตกต่างกัน เพราะตัวแปรจำนวนกลุ่มเป็นตัวแปรที่สำคัญในการจัดกลุ่มด้วย K-means ดังนั้นความถูกต้องของการจัดกลุ่มจึงขึ้นอยู่กับตัวแปรต่างๆ ที่จะต้องพิจารณาถึงสมมติฐานของข้อมูลและความเหมาะสมกับแอปพลิเคชันที่นำมาประยุกต์ใช้งานด้วย

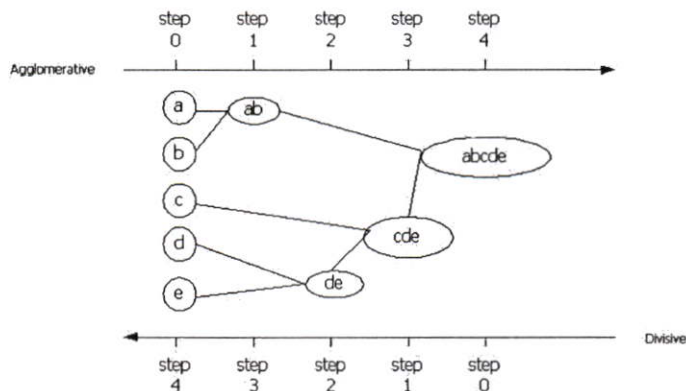
ในการจัดกลุ่ม ข้อมูลจะถูกแสดงด้วยเวกเตอร์ เรียกว่า feature vectors ซึ่งค่าของ feature vector แสดงได้ด้วย $x = (x_1, \dots, x_n)$ โดยวิธีการที่เป็นที่นิยมได้แก่ Hierarchical Clustering

Algorithms, Partitional Clustering Algorithms และ Artificial Neural Network for Clustering [6], [17], [18]

2.1.1 Hierarchical Clustering Algorithms

Hierarchical Clustering ทำงานโดยการจัดกลุ่ม object ของข้อมูล (data objects) มาอยู่ในรูปต้นไม้ของกลุ่มข้อมูล (a tree of clusters) โดยทั่วไปจะสามารถแบ่ง Hierarchical Clustering ออกได้เป็น 2 แบบคือ Agglomerative และ Divisive hierarchical clustering ขึ้นกับว่าต้องการทำในลักษณะ bottom-up หรือ top-down ซึ่งในรูปที่ 2.2 ได้แสดงขั้นตอนการทำ Hierarchical ทั้งในแบบ Agglomerative และ Divisive

Agglomerative hierarchical clustering เป็นวิธีการแบบ bottom-up ซึ่งเริ่มจากการเอาแต่ละ object ไปไว้ในกลุ่มของมันจากนั้นก็ทำการรวมกลุ่มเข้ามาเรื่อยๆ จนกระทั่งทุก object อยู่ในกลุ่มเดียวกันหมด หรือ เมื่อมีเงื่อนไขขอบเขตที่แน่นอน Divisive hierarchical clustering เป็นวิธีการแบบ top-down ซึ่งตรงกันข้ามกับแบบ Agglomerative hierarchical clustering ซึ่งเริ่มจากเอาทุกๆ object ไว้ในกลุ่มเดียวกัน จากนั้นก็ค่อยๆ แบ่งให้กลุ่มเล็กลงเรื่อยๆ จนกระทั่งแต่ละ object อยู่ในกลุ่มของมันเอง หรือ เมื่อมีเงื่อนไขขอบเขตที่แน่นอน เช่น ได้จำนวนกลุ่มตามที่ต้องการแล้วหรือกลุ่มที่ใกล้กันมีระยะห่างตามที่ต้องการแล้ว



รูปที่ 2.2 Agglomerative และ Divisive hierarchical clustering [24]

นอกจากนี้ยังมีการพัฒนาคุณภาพของการทำ Hierarchical Clustering โดยนำไปรวมกับเทคนิคอื่นๆ เพื่อได้เทคนิคใหม่ที่มีประสิทธิภาพมากขึ้นเช่น BIRCH, CURE, ROCK, Chameleon

2.1.2 Partitional Clustering Algorithms

Partitional clustering เป็นวิธีการที่พยายามจัดแบ่งข้อมูลออกเป็น ส่วน ซึ่งอาจจะเริ่มต้นโดยการกำหนดจำนวนกลุ่มที่จะได้จากการจัดกลุ่มและใช้ฟังก์ชันคำนวณเพื่อหา กลุ่มที่เหมาะสมให้กับข้อมูล วิธีการที่นิยมในการจัดกลุ่มประเภทนี้คือ K-means Clustering ซึ่งเป็นอัลกอริทึมที่ใช้ในการ

แบ่งกลุ่มข้อมูล n objects ออกเป็น k partition หรือ k กลุ่ม โดยจะมีการกำหนดจุดศูนย์กลางของกลุ่มข้อมูลซึ่งถือเป็นตัวแทนของกลุ่มนั้น และข้อมูลแต่ละตัวจะถูกจัดให้อยู่ในกลุ่มที่ข้อมูลนั้นใกล้ศูนย์กลางของกลุ่มมากที่สุด ซึ่งสามารถสรุปขั้นตอนการทำงานได้รูปที่ 2.3

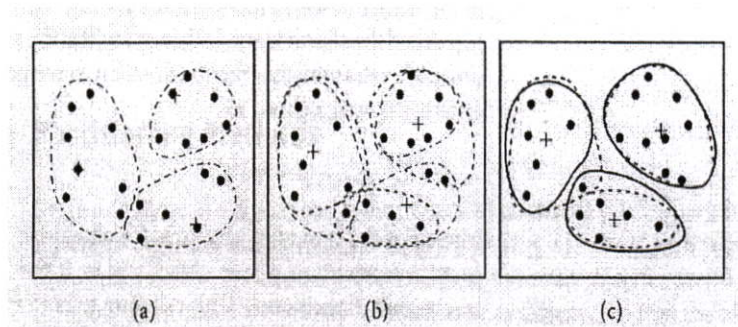
Algorithm Basic K-means algorithm.

- 1: Select K points as initial centroids.
 - 2: **repeat**
 - 3: Form K clusters by assigning each point to its closest centroid.
 - 4: Recompute the centroid of each cluster.
 - 5: **until** Centroids do not change.
-

รูปที่ 2.3 K-means Algorithm [17]

วิธี K-means Clustering นี้จะเหมาะสำหรับข้อมูลที่เมื่อจัดกลุ่มแล้ว ข้อมูลภายในกลุ่มมีความสัมพันธ์กันมาก และวิธีนี้สามารถใช้กับข้อมูลปริมาณมากได้อย่างมีประสิทธิภาพเนื่องจากมีค่า Complexity เป็น $O(nkt)$ โดย n คือจำนวนข้อมูล k คือจำนวนกลุ่ม และ t คือจำนวน iteration

ตัวอย่างการทำ K-means Clustering แสดงในรูปที่ 2.4 ซึ่งเป็นการจัดกลุ่มโดยให้ $k = 3$ ดังรูปที่ 2.4(a) จากนั้นกำหนดค่ากลางของกลุ่มโดยครั้งแรกจะใช้ค่าของข้อมูลที่คิดว่าน่าจะเป็นศูนย์กลางได้ ซึ่งแทนด้วยสัญลักษณ์ + ในภาพ จากนั้นก็เริ่มการคำนวณซึ่งทำให้ได้ขอบเขตของกลุ่มเปลี่ยนแปลงไปตามรูป 2.4(b) ซึ่งการทำซ้ำจะสิ้นสุดเมื่อไม่มีการเปลี่ยนแปลงขอบเขตของกลุ่ม



รูปที่ 2.4 ตัวอย่างการจัดกลุ่มข้อมูลด้วยวิธี K-means [24]

2.1.3 Artificial Neural Network for Clustering

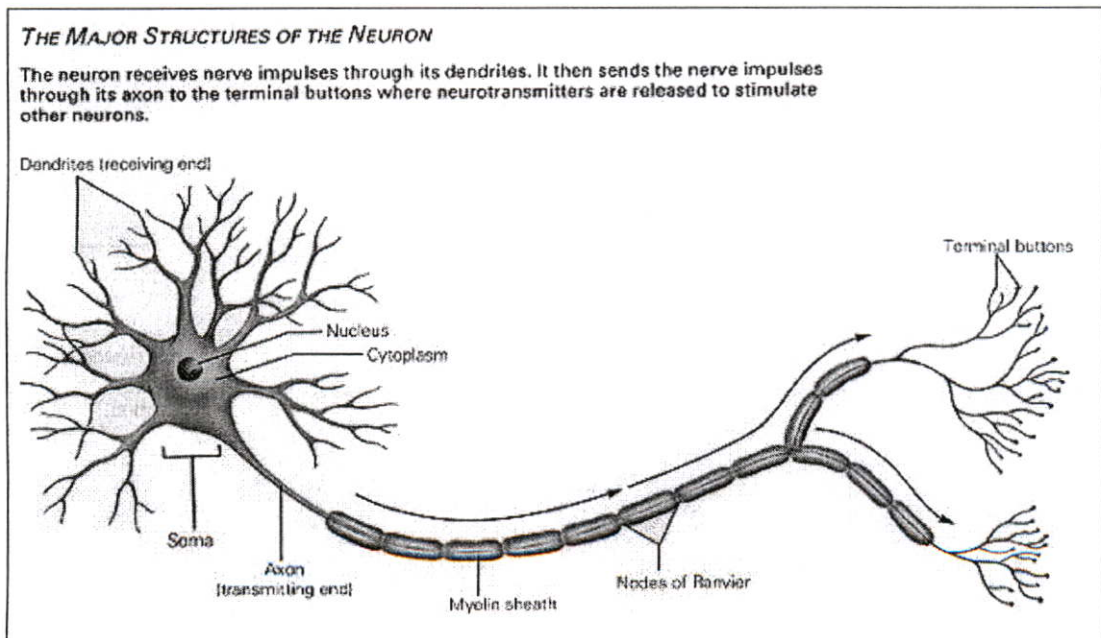
วิธีการ Artificial Neural Network หรือโครงข่ายประสาทเทียมเป็นที่นิยมทั้งในการจัดกลุ่มด้วย Clustering และ Classification ซึ่งวิธีการที่นิยมได้แก่ Kohonen's Learning vector quantization (LVQ), Self Organizing map (SOM) และ Adaptive Resonance Theory (ART) โดยที่ลักษณะของโครงข่ายจะมีชั้นและค่าถ่วงน้ำหนัก โดยจะในขั้นตอนของการเรียนรู้จะมีการปรับเปลี่ยนค่าถ่วงน้ำหนัก ทั้งนี้การเรียนรู้ของโครงข่ายจะคำนึงถึง

- Stability หมายถึงการที่โครงข่ายไม่มีการเปลี่ยนแปลงใดๆเลยในช่วงของการเรียนรู้
- Plasticity หมายถึงการที่โครงข่ายเปลี่ยนแปลงเมื่อมีข้อมูลใหม่

โดยที่เมื่อเริ่มต้นการเรียนรู้การเกิด Plasticity ควรจะมีค่าสูง และเมื่อเรียนรู้ไปนานมากขึ้น ควรจะมีค่า Stability มากขึ้น

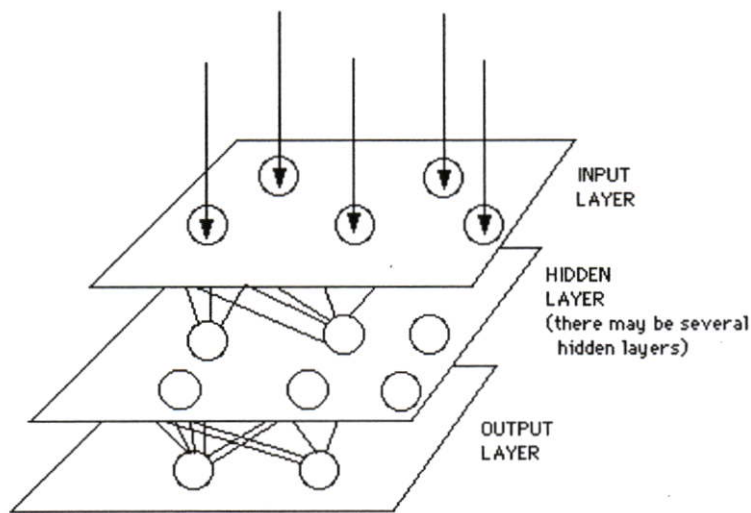
2.2 โครงข่ายประสาทเทียม (Neural Networks)

โครงข่ายประสาทเทียมเกิดขึ้นจากความพยายามให้คอมพิวเตอร์มีความสามารถใกล้เคียงกับมนุษย์ คือ คิดเป็นและเรียนรู้ได้ [6] ดังนั้นจึงพยายามเลียนแบบโครงสร้างสมอง ซึ่งโครงสร้างของสมองนั้นประกอบด้วยนิวรอลหรือเซลล์ประสาทจำนวนหลายพันล้านเซลล์ และแต่ละเซลล์จะเชื่อมต่อกับเซลล์อื่นๆอีกพันหรือหมื่นเซลล์ ซึ่งเซลล์ชนิดนี้จะพิเศษกว่าเซลล์อื่นๆในร่างกายเนื่องจากมี Dendrite และ Axon ในการติดต่อสื่อสารกันระหว่างเซลล์ด้วยกระบวนการทางเคมี [9] ดังแสดงส่วนประกอบของเซลล์ในรูปที่ 1 โดย Dendrite จะทำหน้าที่รับข้อมูลมาสู่เซลล์ ในขณะที่ Axon จะทำหน้าที่ส่งข้อมูลไปยังเซลล์อื่น ทั้งนี้เซลล์ประสาทในแต่ละส่วนของร่างกายก็อาจจะมีรูปร่างที่แตกต่างกันไปตามความเหมาะสม จากการศึกษาทางชีววิทยาพบว่าสมองจะมีการเก็บข้อมูลที่ได้รับเข้ามาเป็นแพทเทิร์นหรือรูปแบบที่เกิดจากการเรียนรู้อยู่ [10] ซึ่งแพทเทิร์นที่ซับซ้อนนี้ทำให้มนุษย์มีความสามารถในการจดจำและแบ่งแยกสิ่งต่างๆได้



รูปที่ 2.5 เซลล์ประสาท [8]

โครงสร้างของโครงข่ายประสาทเทียมจะแบ่งเป็นชั้นๆ(Layer) โดยแต่ละชั้นจะประกอบด้วยโหนดหรือหน่วยในการประมวลผล (Processing Element) จำนวนชั้นของโครงข่ายจะขึ้นกับรูปแบบโครงข่ายซึ่งโครงข่ายโดยทั่วไปจะมีชั้นอินพุต ชั้นเอาต์พุต และชั้นภายใน โดยชั้นอินพุตหรือ Input Layer จะรับข้อมูลเข้ามาจากไฟล์หรือจากแอปพลิเคชันอื่น ชั้นเอาต์พุตหรือ Output Layer จะส่งข้อมูลจากการประมวลผลออกจากโครงข่าย ซึ่งระหว่างชั้นอินพุตและเอาต์พุต อาจมีชั้นภายในหรือ Hidden Layer เชื่อมต่ออยู่ โดยชั้นภายในนี้อาจจะมีมากกว่าหนึ่งชั้นก็ได้ขึ้นกับรูปแบบและความเหมาะสม ซึ่งแต่ละโหนดของชั้นภายในนี้จะเชื่อมต่อกับโหนดของชั้นอื่นที่อยู่ติดกัน



รูปที่ 2.6 โครงข่ายประสาทเทียม[10]

การพัฒนาโครงข่ายประสาทเทียมเริ่มจากปี ค.ศ. 1943 ที่ Warren McCulloch และ Walter Pitts ได้ออกแบบนิรอรอลที่เป็นต้นกำเนิดของโครงข่ายประสาทเทียม ซึ่งต่อมา Donald Hebb ได้ออกแบบกฎการเรียนรู้โครงข่ายประสาทเทียมเป็นคนแรกในปี ค.ศ.1949 และจากนั้นก็มีการพัฒนารูปแบบอื่นๆตามการใช้งานที่แตกต่างกันออกไป เช่น Stephen Grossberg ได้พัฒนาแบบจำลอง Avalanche ขึ้นในปี ค.ศ.1967 ซึ่งมีความสามารถในการเข้าใจคำพูด P. Werbos, D. Parker และ D. Rumelhart ได้พัฒนาแบบจำลอง Back Propagation ในปี ค.ศ.1974-1985 เพื่อใช้งานด้านการอ่านเขียน ทำนายหุ้น [7]

โดยโครงข่ายประสาทเทียมที่ออกแบบมาเพื่อการใช้งานด้านจดจำรูปแบบเพื่อการจัดกลุ่มข้อมูล (pattern recognition) นั่นคือแบบจำลอง Adaptive Resonance Theory (พัฒนา ค.ศ.1978-1986) , Boltzmann และ Cauchy machine (พัฒนา ค.ศ.1985-1986) ซึ่งแบบจำลอง Adaptive Resonance Theory มีความนิยมในการนำไปใช้งานและพัฒนารูปแบบแตกย่อยออกไป

2.3 โครงข่ายประสาทเทียมแบบอะแดปทีฟเรโซแนนซ์ทฤษฎี (Adaptive Resonance Theory)

อะแดปทีฟเรโซแนนซ์ทฤษฎี (Adaptive Resonance Theory) หรือ ART พัฒนาโดย Stephen Grossberg และ Gail Carpenter ซึ่งนำเสนอเมื่อปีค.ศ. 1987 เพื่อใช้งานงานด้านจดจำรูปแบบ การจัดกลุ่มข้อมูล (pattern recognition) โดยเริ่มจากการออกแบบอัลกอริทึมเพื่อใช้กับข้อมูลไบนารีและใช้ชื่อว่า ART1 และพัฒนา ART2 ในเวลาต่อมาให้สามารถใช้กับข้อมูลอนาล็อกได้ และเมื่อปีค.ศ. 1991 Carpenter Grossberg และ Rosen ได้เสนอ FUZZY ART ที่ใช้พื้นฐานจาก ART1 มาพัฒนาให้มีความสามารถใช้กับข้อมูลอนาล็อก นอกจากนี้ยังมีโครงข่ายที่เกิดขึ้นจากการพัฒนาเพิ่มเติมอีกมากมาย เช่น ARTMAP, Fuzzy ARTMAP, Adaptive Hamming Net (AHN), Gaussian ART (GA), Simplified Fuzzy ARTMAP (SFAM), Simplified ART (SART) [11]

ART เป็นโครงข่ายประสาทเทียมแบบเรียนรู้ด้วยตนเอง (Unsupervised Learning) ที่ใช้ Competitive learning และกฎ Winner-take-all ซึ่งโครงสร้างของ ART เป็นโครงข่ายประสาทเทียมที่มี 2 ชั้น (Layer) คือชั้นที่เป็นอินพุตและชั้นที่เป็นเอาต์พุต ซึ่งในงานวิจัยบางแห่งอาจจะแบ่งชั้นย่อยในชั้นอินพุตเป็นชั้นอินพุตและชั้นอินเตอร์เฟส โดยชั้นอินพุตจะรับข้อมูลจากภายนอกโครงข่ายแล้วส่งต่อให้ชั้นอินเตอร์เฟส จากนั้นชั้นอินเตอร์เฟสซึ่งเชื่อมต่อกับชั้นเอาต์พุตด้วยค่าถ่วงน้ำหนัก 2 ชุด คือ Bottom-up Weight (Forward Connection) และ Top-down Weight (Backward Connection)

ในชั้นเอาต์พุตของ ART (F_2 Layer) ถือว่าเป็น Competitive Layer คือในรอบของการเรียนรู้จะมีการคำนวณเพื่อกำหนดค่าให้แต่ละโหนดในชั้นนี้ ซึ่งโหนดที่มีค่ามากที่สุดจะถือว่าเป็นผู้ชนะในรอบของการเรียนรู้นั้นๆ นั่นคือโหนดนั้นจะได้สิทธิในการเรียนรู้ข้อมูลที่เข้ามาตามกฎ Winner-take-all โดยการเรียนรู้นั้นจะเป็นการเปรียบเทียบข้อมูลที่เข้ามากับค่าถ่วงน้ำหนัก Top-down Weight (Backward Connection) ซึ่งค่าถ่วงน้ำหนักนี้เป็น Long-term memory (LTM) ที่ทำให้ ART สามารถแก้ปัญหาของโครงข่ายประสาทเทียมในเรื่อง Plasticity/Stability Dilemma ที่เป็นปัญหาในการเรียนรู้ข้อมูลใหม่และการจดจำข้อมูลหรือความเดิมเอาไว้

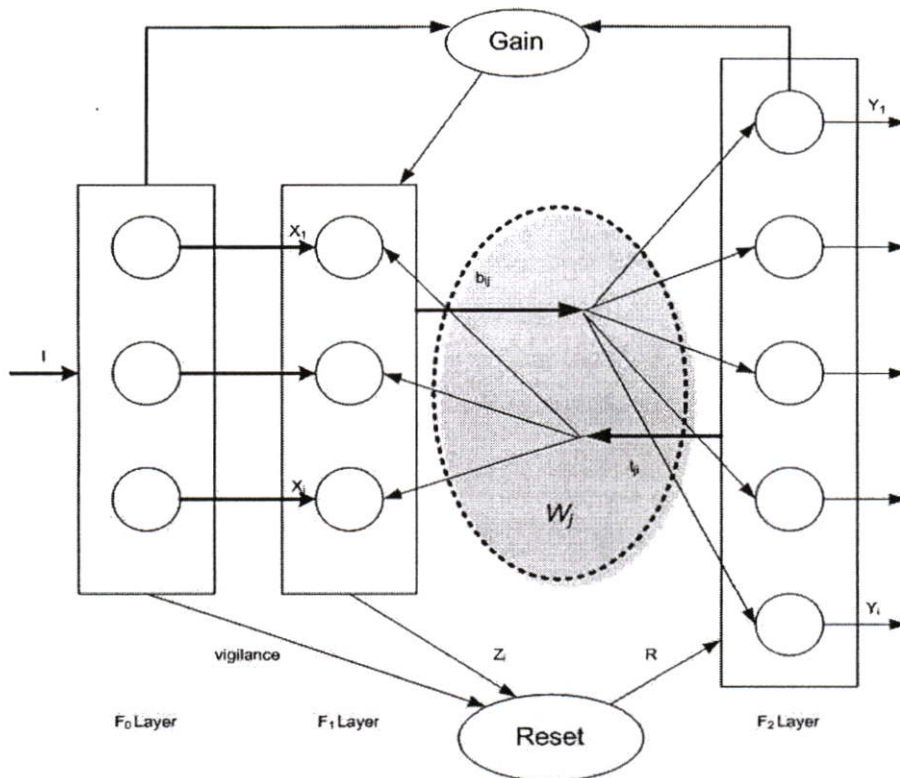
โครงสร้าง ART จะประกอบด้วย 2 subsystem คือ attentional subsystem และ orienting subsystem ซึ่งใน attentional subsystem จะเป็นส่วนที่มีค่าถ่วงน้ำหนักซึ่งจะมีค่าเปลี่ยนแปลงตามรอบของการเรียนรู้หรือเป็นส่วนของการเรียนรู้โดยอาศัยความรู้เดิมที่มีอยู่ ส่วนใน orienting subsystem จะมี reset unit ที่ทำหน้าที่ในการตรวจสอบผลจากการเรียนรู้ว่าเพียงพอต่อการยอมรับเป็นความรู้ใหม่หรือไม่ ในการทำงานของโครงข่าย ART นั้น ค่าที่เกิดขึ้นในแต่ละโหนดของโครงข่ายจากการเรียนรู้เรียกว่า short-term memory (STM) เนื่องจากค่าดังกล่าวจะเปลี่ยนแปลงไปเรื่อยๆตามรูปแบบของข้อมูลที่เข้ามา ส่วนการถ่วงน้ำหนักให้แก่การเชื่อมต่อของชั้นอินพุตและชั้น

เอาท์พุท เรียกว่า long-term memory (LTM) เนื่องจากค่าถ่วงน้ำหนักที่เกิดขึ้นจะถือว่าเป็นเพิ่มเพลทของแต่ละโหนดเอาท์พุทที่เชื่อมต่ออยู่

ในงานวิจัยนี้จะเป็นการศึกษา ART1 และ Fuzzy ART เพื่อนำมาพัฒนาเป็น Unified ART ให้มีความสามารถในการจัดกลุ่มข้อมูล โดยการใช้ตัวแปรต่างๆ ในการนำเสนออัลกอริทึมมีดังนี้

2.3.1 โครงข่ายประสาทเทียมแบบอะแดปทีฟที่พรซิแน้นเทียร์ 1 (ART1)

ART1 เป็นโครงข่ายประสาทเทียมที่พัฒนาจากโครงสร้าง ART โดย ART1 จะเป็นโครงข่ายที่ใช้กับข้อมูล (\mathbf{x}) ที่เป็นไบนารี นั่นคือ $x_i \in \{0,1\}$ ซึ่งขนาดของเวกเตอร์จะคำนวณได้จากผลรวมของส่วนประกอบทั้งหมด เช่น $\|\mathbf{x}\| = \sum_{i=1}^m x_i$ ทั้งนี้ผลรวมที่เกิดขึ้นก็จะเท่ากับจำนวนของส่วนประกอบที่ไม่เป็นศูนย์ของเวกเตอร์นั่นเอง



รูปที่ 2.7 โครงสร้างของ ART1 [25]

โครงสร้างของ ART1 มีการใช้ค่าถ่วงน้ำหนัก 2 ชุดดังแสดงในรูปที่ 2.7 คือ bottom-up weight ซึ่งเชื่อมต่อและส่งข้อมูลระหว่างโหนดที่ i ใน F_1 Layer ไปยังโหนดที่ j ใน F_2 Layer และ top-down weight เชื่อมต่อและส่งข้อมูลระหว่างโหนดที่ j ใน F_2 Layer ไปยังโหนดที่ i ใน F_1 Layer

ทั้งนี้จากการศึกษาผลงานวิจัยของ Baraldi และ Parmiggiani ได้ทำการพิสูจน์ว่าการมีค่าถ่วงน้ำหนัก 2 ชุดของ ART นั้นมีค่าเท่ากับการมีค่าถ่วงน้ำหนักเพียงชุดเดียวดังแสดงด้วย W_j แทน

ค่าถ่วงน้ำหนักของโครงข่ายแบบ Feed-forward [11] โครงข่าย ฟังก์ชันที่ใช้ในการคำนวณของโครงข่าย ART1 นั้นจะมีหลักๆอยู่ 3 ฟังก์ชัน ได้แก่

1. Choice function หรือ Activation function จะเป็นฟังก์ชันที่ให้ผลลัพธ์เป็นค่าความเหมือนสำหรับโหนดเอาต์พุตแต่ละโหนด ซึ่งเกิดจากการเปรียบเทียบค่าของอินพุตที่เข้ามากับค่าถ่วงน้ำหนักที่ถือเป็นแพทเทิร์นของโหนดเอาต์พุตที่เชื่อมอยู่
2. Match function จะเป็นฟังก์ชันที่คำนวณว่าความเหมือนนั้นเพียงพอสำหรับการเรียนรู้หรือไม่ ซึ่งจะถือว่าคุณค่าดังกล่าวมีความเหมือนกันมากเมื่อค่านี้มีค่ามากขึ้น และจะถือว่าการ resonance เมื่อนำไปเปรียบเทียบกับค่า vigilance แล้วมีค่ามากกว่า
3. Update function จะเป็นฟังก์ชันสำหรับการปรับค่าถ่วงน้ำหนักสำหรับโหนดเอาต์พุตหลังจากเกิดการ resonance นั่นคือฟังก์ชันจะทำการปรับปรุงข้อมูลรูปแบบหรือแพทเทิร์นเดิมด้วยการนำข้อมูลหรือความรู้ใหม่มาเพิ่มเข้าไป

```

Initialize: Network Parameters
c = 0
while (X not empty) {
  Learning Loop:

  get x;
  new = true;
  Loop j = 1 : c {
    
$$I(x, w_j) = \frac{\|x \cap w_j\|}{\alpha + \|w_j\|}$$

    
$$Y_j = I(x, w_j)$$

  }
  Loop i = 1 : c {
    J = index of max I(x, w_j)
    
$$M(x, w_j) = \frac{\|x \cap w_j\|}{\|x\|}$$

    if M(x, w_j) > ρ {
      
$$U(x, w_j) = (1 - \beta)w_j + \beta(x \cap w_j)$$

      
$$w_j = U(x, w_j)$$

      new = false;
      break;
    }
    else
      
$$Y_j = -1$$

  }
  newNode(new);
}

```

รูปที่ 2.8 อัลกอริทึมของ ART1

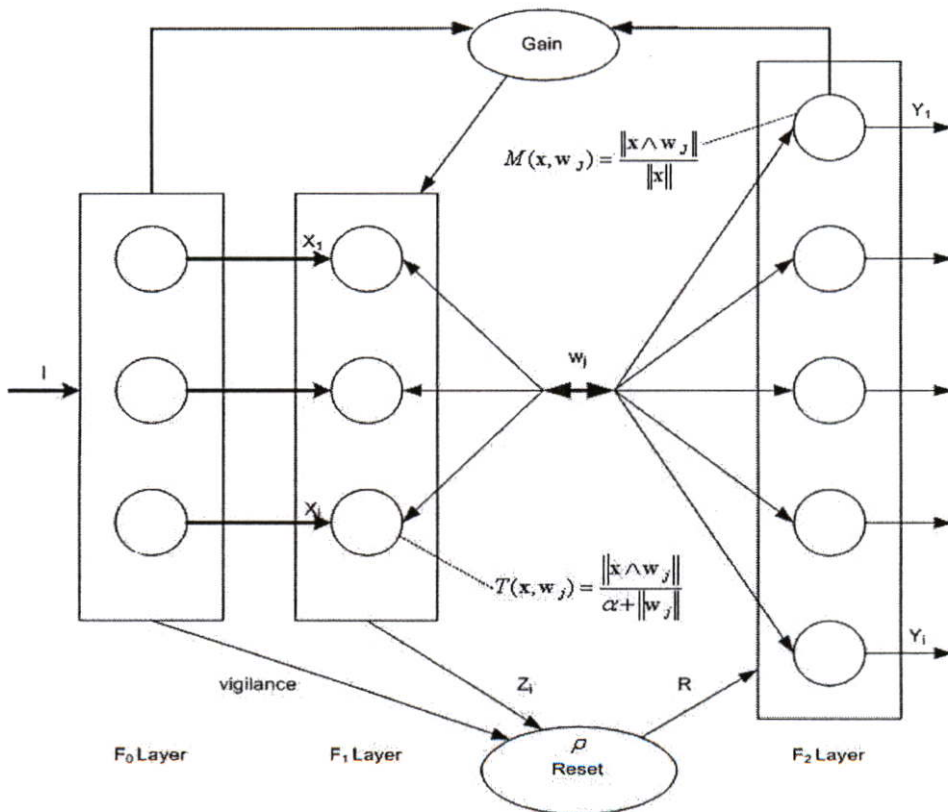
ซึ่งรายละเอียดการเรียนรู้ของโครงข่าย ART1 จะเป็นไปตามอัลกอริทึมดังแสดงในรูปที่ 2.8 โดย โอเปอเรเตอร์ที่ใช้ในฟังก์ชันหลักของโครงข่ายจะเป็น \cap ซึ่งมีความหมายคล้ายการ อินเตอร์

เซตชั้น โดย $\mathbf{a} \cap \mathbf{b} = (a_1 \text{ AND } b_1, a_2 \text{ AND } b_2, \dots, a_m \text{ AND } b_m)$ และ $\|\mathbf{a}\|$ หมายถึงขนาดของ \mathbf{a} โดย $\|\mathbf{a}\| = \sum_{i=1}^m a_i$

ในการหาโหนดที่มีค่าสูงสุดหรือ Winning Node นั้น ถ้าไม่สามารถหาโหนดที่เหมาะสมได้ คือทุกโหนดโดนปรับค่าเป็น -1 ทั้งหมดหรือทุกโหนดถูก Reset ค่าใหม่หมด (การ Reset ในงานวิจัยนิยมให้เป็น -1 หรือ 0) จะมีการเพิ่มโหนดบนชั้นเอาท์พุทโดยให้ค่าถ่วงน้ำหนักของโหนด หรือเทมเพลตใหม่นั้นเท่ากับค่าของข้อมูลที่เข้ามา $\mathbf{w}_c = \mathbf{x}$ และเพิ่มข้อมูลเข้าไปในชุดของข้อมูลถ่วงน้ำหนักของโครงข่าย $\mathbf{W} = \mathbf{W} \oplus \mathbf{w}_c$ เมื่อโหนดที่เพิ่มใหม่คือโหนดที่ c

2.3.2 โครงข่ายประสาทเทียมแบบฟัซซี่อะแดปทีฟเรโซแนนซ์ (FUZZY ART)

Fuzzy ART พัฒนามาจาก ART1 ที่รับข้อมูลไบนารี ให้สามารถรองรับข้อมูลอนาล็อกได้ ทำให้โอเปอร์เรเตอร์ที่ใช้ใน ART และ Fuzzy ART แตกต่างกัน โดยฟังก์ชันที่ใช้ใน Fuzzy ART จะใช้ \wedge แทน \cap ใน ART1 ซึ่ง $a \wedge b = (\min(a_1, b_1), \min(a_2, b_2), \dots, \min(a_m, b_m))$ ทั้งนี้ข้อมูลที่ให้นำมาเข้าโครงข่ายจะต้องมีการปรับค่า (Normalize) เพื่อป้องกันการเพิ่มจำนวนโหนดเอาท์พุท โดยค่าของข้อมูลหลังจากปรับจะอยู่ในช่วง 0 ถึง 1



รูปที่ 2.9 โครงสร้างของ Fuzzy ART

การปรับค่าสามารถทำได้หลายวิธี โดยวิธีที่ S. Grossberg และ G. Carpenter นำเสนอ [5] คือการใช้ค่าคอมพลิเมนต์ $\mathbf{x} = (\mathbf{x}, \mathbf{x}^c) = (x_1, x_2, \dots, x_m, 1-x_1, 1-x_2, \dots, 1-x_m)$ หรือการเปรียบเทียบค่าจากโดเมนของข้อมูล โดยการคำนวณปรับค่าจะเกิดบนชั้น F_0 ที่เป็น Interface Layer ทั้งนี้สำหรับ Fuzzy ART ขนาดของ \mathbf{x} สามารถหาได้จาก $\|\mathbf{x}\| = m$ โดยขั้นตอนการเรียนรู้ของโครงข่าย Fuzzy ART จะเหมือนกับโครงข่าย ART1 ซึ่งโครงสร้างแสดงดังรูปที่ 2.9 และอัลกอริทึมแสดงในรูปที่ 2.10

```

Initialize: Network Parameters
C = 0
while (X not empty) {
  Learning Loop:
  get  $\mathbf{x}$ ;
  new = true;
  Loop j = 1 : C {
     $T(\mathbf{x}, \mathbf{w}_j) = \frac{\|\mathbf{x} \wedge \mathbf{w}_j\|}{\alpha + \|\mathbf{w}_j\|}$ 
     $\bar{Y}_j = T(\mathbf{x}, \mathbf{w}_j)$ 
  }
  Loop i = 1 : C {
     $J = \text{index of max } T(\mathbf{x}, \mathbf{w}_j)$ 
     $M(\mathbf{x}, \mathbf{w}_j) = \frac{\|\mathbf{x} \wedge \mathbf{w}_j\|}{\|\mathbf{x}\|}$ 
    if  $M(\mathbf{x}, \mathbf{w}_j) > \rho$  {
       $U(\mathbf{x}, \mathbf{w}_j) = (1-\beta)\mathbf{w}_j + \beta(\mathbf{x} \wedge \mathbf{w}_j)$ 
       $\mathbf{w}_j = U(\mathbf{x}, \mathbf{w}_j)$ 
      new = false;
      break;
    }
    else
       $\bar{Y}_j = -1$ 
  }
  newNode(new);
}

```

รูปที่ 2.10 อัลกอริทึมของ Fuzzy ART

จากอัลกอริทึมรูปที่ 2.8 และ 2.10 พบว่าโดยอัลกอริทึมของ ART1 และ Fuzzy ART มีความใกล้เคียงกันมากโดยจะเห็นว่ามีการปรับในส่วนของฟังก์ชันการเปรียบเทียบความคล้ายคลึงและฟังก์ชันการปรับค่าล่วงหน้าหน้าที่มีการคำนวณที่แตกต่างกันทั้งนี้สัญลักษณ์ที่ใช้ในโครงสร้างและอัลกอริทึมของ ART1 และ Fuzzy ART สามารถสรุปได้ดังตารางที่ 2.1

ตารางที่ 2.1 สัญลักษณ์ที่ใช้ใน ART

Notation	Description
I	แพทเทิร์นอินพุท
\mathbf{x}	ข้อมูลหรือแพทเทิร์นที่ F_1 Layer โดยที่ $\mathbf{x} = (x_1, x_2, \dots, x_m)$
b_{ij} / w_{ij}	Bottom-up weight
t_{ji} / w_{ji}	Top-down weight
ρ	ค่า Vigilance Parameter ซึ่งเป็นค่าที่กำหนดระดับความคล้ายคลึงของโครงข่าย
c	จำนวน โหนดบน F_2 Layer หรือจำนวนคลัสเตอร์
α	ค่า Choice Parameter โดยปกติจะมีค่าเป็น 1 สำหรับ Fast Learning
T_j	ค่าการคำนวณ Choice Function บน F_2 Layer
Y	เวกเตอร์ของค่า Choice Function บน F_2 Layer
$ \cdot $	ขนาดของตัวแปร
Min(.)	ค่าน้อยที่สุด

2.4 การเปรียบเทียบความคล้ายคลึงของเอกสาร (Similarity Measure)

จากการที่การจัดกลุ่มเป็นการรวมข้อมูลที่มีลักษณะกันไว้ด้วยกัน K.C. Gowda และ E. Diday [3] ได้นำเสนอวิธีการจัดกลุ่มโดยใช้การเปรียบเทียบความคล้ายคลึงของเอกสารซึ่งจะแทนเอกสารต่างๆ (Document Representation) ให้อยู่ในรูปแบบที่สามารถนำมาคำนวณได้ โดยแทนเอกสารต่างๆด้วยคุณสมบัติ (Feature) ที่สามารถแทนเอกสารนั้นได้ในรูป Cartesian Product ของคุณสมบัติเอกสารได้เป็น [5]

นั่นคือแต่ละเอกสารจะสามารถแทนด้วย $Doc = D_1 \times D_2 \times D_3 \times \dots \times D_k$ เมื่อเอกสารหรือ Doc ประกอบด้วยคุณสมบัติ k คุณสมบัติ และ D_k คือคุณสมบัติของเอกสารที่ k ทั้งนี้สามารถแบ่งลักษณะของคุณสมบัติออกได้เป็น

1. คุณสมบัติเชิงปริมาณ (Quantitative Features) คือ
 - a. ข้อมูลตัวเลขที่เป็นจุดทศนิยม เช่น ความสูง 4.25 เมตร
 - b. ข้อมูลตัวเลขที่เป็นจำนวนเต็ม เช่น จำนวนบุตร 6 คน
 - c. ข้อมูลตัวเลขที่เป็นช่วง เช่น ระยะทาง 10-15 กิโลเมตร
2. คุณสมบัติเชิงคุณภาพ (Qualitative Features) คือ
 - a. ข้อมูลที่เป็นข้อความ เช่น สี “แดง”
 - b. ข้อมูลที่เป็นลำดับ เช่น อันดับที่ “21”

c. ข้อมูลที่เป็นชุดของข้อมูล เช่น พาหนะที่โดยสารมาลาดกระบัง “รถ, รถโดยสาร, รถตู้, รถไฟ”

3. โครงสร้างของตัวแปร (Structured Variables) เป็นลักษณะของการถ่ายทอด เช่น รถอาจแตกออกได้เป็น Ford , Fiat, Renault และ Benz

การเปรียบเทียบความคล้ายคลึงของเอกสาร ระหว่างเอกสาร A และเอกสาร B สามารถเขียนในสมการได้ดังนี้

$$S(A, B) = \sum_{k=1}^d S(A_k, B_k) \quad (2.1)$$

ในการเปรียบเทียบความคล้ายคลึงของ $S(A_k, B_k)$ จะเป็นการรวมความคล้ายคลึงในส่วนย่อย คือ $S(A_k, B_k) = S_s(A_k, B_k) + S_c(A_k, B_k) + S_p(A_k, B_k)$ โดย

1. $S_s(A_k, B_k)$ คือส่วนที่เป็นขนาดหรือจำนวนของคุณลักษณะหรือ Span
2. $S_c(A_k, B_k)$ คือส่วนที่เป็นเนื้อหาของคุณลักษณะหรือ Content
3. $S_p(A_k, B_k)$ คือส่วนที่เป็นตำแหน่งของคุณลักษณะหรือ Position ซึ่งจะมีค่าเฉพาะคุณสมบัติเชิงปริมาณ (Quantitative Features)

2.4.1 การคำนวณสำหรับข้อมูลเชิงปริมาณที่เป็นช่วง

การเปรียบเทียบความคล้ายคลึงของข้อมูลเอกสารเอกสาร A และ B จะแทนด้วย $S(A, B)$ ซึ่งการเปรียบเทียบจะเกิดระหว่างคุณสมบัติของเอกสาร คือระหว่าง A_k และ B_k หรือ $S(A_k, B_k)$ โดยที่นิยามของสัญลักษณ์ในสมการแสดงดังตารางที่ 2.2

ค่าความคล้ายคลึงของเอกสารจะประกอบด้วยค่าความคล้ายคลึงของแต่ละคุณลักษณะ ซึ่งแต่ละคุณลักษณะจะสามารถคำนวณได้ดังนี้

$$S_s(A, B) = \frac{(l_a + l_b)}{2 \times l_s} \quad (2.2)$$

$$S_c(A, B) = \frac{l_i}{l_s} \quad (2.3)$$

$$S_p(A, B) = 1 - \left| \frac{a_i - b_i}{|U_k|} \right| \quad (2.4)$$

2.4.2 การคำนวณสำหรับข้อมูลเชิงปริมาณที่เป็นข้อมูลตัวเลข (ทศนิยมและจำนวนเต็ม)

สมมติข้อมูลเอกสารที่ต้องการเปรียบเทียบคือ A และ B โดยที่นิยามของสัญลักษณ์ในสมการเหมือนกับข้อมูลเชิงปริมาณที่เป็นช่วงและมีคุณสมบัติเพิ่มเติมดังนี้

$$1) a_i = a_u$$

$$2) b_l = b_u$$

$$3) l_a = l_b = l_i = 0$$

ค่าความคล้ายคลึงของเอกสารจะประกอบด้วยค่าความคล้ายคลึงของแต่ละคุณลักษณะ ซึ่งแต่ละคุณลักษณะจะสามารถคำนวณได้ดังนี้

$$S_s(A, B) = \frac{(l_a + l_b)}{2 \times l_s} \quad (2.5)$$

$$S_c(A, B) = \frac{l_i}{l_s} = 0 \because l_i = 0 \quad (2.6)$$

$$S_p(A, B) = 1 - \frac{|a_l - b_l|}{|U_k|} \quad (2.7)$$

ตารางที่ 2.2 สัญลักษณ์ที่ใช้ในการเปรียบเทียบความคล้ายคลึงสำหรับข้อมูลตัวเลข

Notation	Description
a_l	ค่าต่ำสุดใน A_k
a_u	ค่ามากสุดใน A_k
b_l	ค่าต่ำสุดใน B_k
b_u	ค่ามากสุดใน B_k
U_k	จำนวนข้อมูลของช่วงที่มีข้อมูลมากที่สุด
l_i	จำนวนของสมาชิกทั้งหมดที่อินเตอร์เซกชันกันระหว่าง A_k และ B_k
l_s	$ \max(a_u, b_u) - \min(a_l, b_l) $
l_a	$ a_u - a_l $
l_b	$ b_u - b_l $

2.4.3 การคำนวณสำหรับข้อมูลเชิงคุณภาพ (ข้อความ)

สมมติข้อมูลเอกสารที่ต้องการเปรียบเทียบคือ A และ B ซึ่งข้อมูลเชิงคุณภาพจะไม่มี ความคล้ายคลึงของตำแหน่ง นั่นคือมีความคล้ายคลึงของตำแหน่ง $S_p(A, B) = 0$ โดยที่นิยามของ สัญลักษณ์ในสมการแสดงในตาราง 2.3 ซึ่งเป็นการปรับนิยามค่าต่างๆ ให้สอดคล้องกับประเภทของ ข้อมูล ตัวอย่างเช่นสูตรการคำนวณค่า l_s ในตาราง คือจำนวนสมาชิกทั้งหมดของ A_k และ B_k รวมกันลบด้วยจำนวนของสมาชิกทั้งหมดที่ อินเตอร์เซกชัน กันระหว่าง A_k และ B_k แทนการหา ค่า \max, \min ซึ่งไม่เหมาะสมกับข้อมูลเชิงคุณภาพ

ค่าความคล้ายคลึงของเอกสารจะประกอบด้วยค่าความคล้ายคลึงของแต่ละคุณลักษณะ ซึ่งแต่ละคุณลักษณะจะสามารถคำนวณได้ดังนี้

$$S_s(A, B) = \frac{(l_a + l_b)}{2 \times l_s} \tag{2.8}$$

$$S_c(A, B) = \frac{l_i}{l_s} \tag{2.9}$$

ตารางที่ 2.3 สัญลักษณ์ที่ใช้ในการเปรียบเทียบความคล้ายคลึงของข้อความ

Notation	Description
l_a	จำนวนสมาชิกทั้งหมดใน A_k
l_b	จำนวนสมาชิกทั้งหมดใน B_k
l_i	จำนวนของสมาชิกทั้งหมดที่อินเตอร์เซกชันกันระหว่าง A_k และ B_k
l_s	$l_a + l_b - l_i$

ทั้งนี้ จะเห็นว่า Similarity Measure นั้นมีการนำเสนอสำหรับข้อมูลเชิงข้อความและข้อมูลตัวเลข แต่เนื่องจากความเหมาะสมในการประยุกต์ใช้งานร่วมกับ Fuzzy ART ที่มีความสามารถในการจัดกลุ่มแบบไม่มีผู้ชี้แนะ ทำให้นักวิจัยนี้ใช้โครงข่าย Fuzzy ART และการเปรียบเทียบความคล้ายคลึงแบบ Fuzzy ART กับข้อมูลตัวเลข และปรับโครงข่ายเพิ่มเติมในการรับข้อมูลข้อความ และใช้การเปรียบเทียบความคล้ายคลึงด้วย Similarity Measure ดังจะกล่าวรายละเอียดในบทต่อไป

2.5 เท็ทอะแดปทีฟเรโซแนนซ์เทียร์นีวอรอลเน็ตเวิร์ค

เท็ทอะแดปทีฟเรโซแนนซ์เทียร์นีวอรอลเน็ตเวิร์ค (Text Adaptive Resonance Theory Neural Network) นำเสนอโดย นรเศรษฐ์ จันทสูตร[15] เป็นโครงข่ายประสาทเทียมแบบอะแดปทีฟเรโซแนนซ์เทียร์มาประยุกต์ร่วมกับ Similarity Measure เพื่อให้ได้โครงข่ายที่มีความสามารถในการจัดกลุ่มข้อมูลข้อความ โครงสร้างของโครงข่ายแสดงดังรูปที่ 2.11 แสดงโครงสร้างของโครงข่ายซึ่งประกอบด้วย 3 ชั้นคือ Input Layer (F1(a) layer) , Interface Layer (F1(b) layer) และ Output Layer (F2 layer)

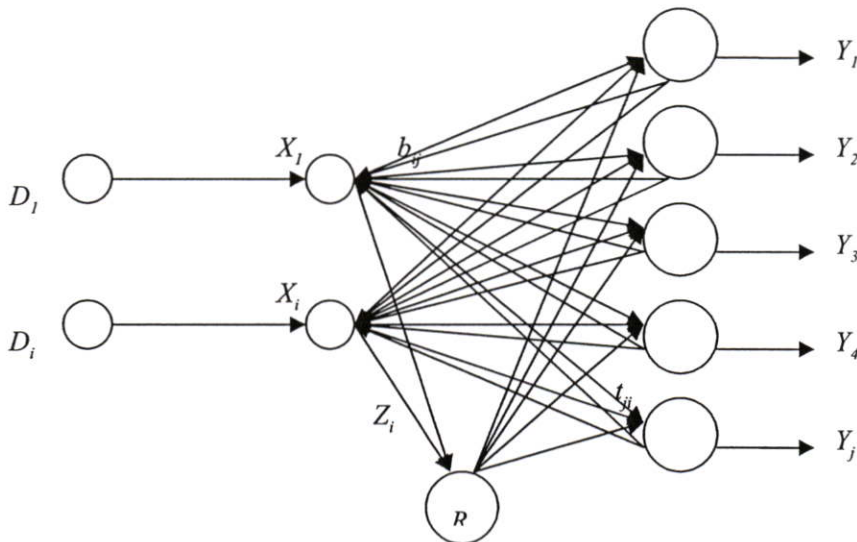
การทำงานของโครงข่ายเป็นแบบ Feed Back ทั้งนี้งานวิจัยนำเสนอโครงสร้างที่มีค่าถ่วงน้ำหนัก 2 ชุด คือ Bottom-up weight (b_{ij}) ซึ่งจะเชื่อมต่อและส่งข้อมูลระหว่างโหนดที่ i ใน $F_{1(b)}$ layer ไปยังโหนดที่ j ใน F_2 layer และ Top-down weight (t_{jp}) ซึ่งจะเชื่อมต่อและส่งข้อมูลระหว่างโหนดที่ j ใน F_2 layer และโหนดที่ i ใน $F_{1(b)}$ layer ทั้งนี้ค่าในค่าถ่วงน้ำหนักสำหรับข้อมูลข้อความก็จะเป็นข้อความ(qualitative value) และจะมีการเก็บค่าแสดงความเป็นสมาชิกหรือค่าดีกรีสำหรับข้อความแต่ละตัวในค่าถ่วงน้ำหนัก โดยโครงสร้างของค่าถ่วงน้ำหนักทั้ง 2 ชุด สามารถเขียนได้เป็น

$$b_{ij} = \{(A_{1ij}, e_{1ij}), (A_{2ij}, e_{2ij}), \dots, (A_{pij}, e_{pij})\} \quad (2.10)$$

เมื่อกำหนดให้ A_{pij} คือ ค่า qualitative value ของ Bottom-up Weight และ e_{pij} คือค่า Degree แสดงความเป็นสมาชิกของ A_{pij} โดยค่า e_{pij} ของ A_{pij} ในโครงข่ายนิวรอนนี้มีค่าอยู่ในช่วงระหว่าง 0 ถึง 1 ซึ่งค่า Degree นี้เป็นค่าที่ให้ระดับความเป็นสมาชิกของ A_{pij} กับ ข้อมูลเข้า ถ้า Degree ของ A_{pij} มีค่าเท่ากับ 0 ให้ความหมายว่า Qualitative value ของ A_{pij} ไม่ได้เป็นส่วนหนึ่งของข้อมูลเข้า i ถ้า Degree ของ A_{pij} มีค่าเท่ากับ 1 ให้ความหมายว่า Qualitative value ของ A_{pij} เป็นสมาชิกของข้อมูล i

$$t_{ji} = \{(B_{1ji}, e_{1ji}), (B_{2ji}, e_{2ji}), \dots, (B_{pji}, e_{pji})\} \quad (2.11)$$

กำหนดให้ B_{pji} คือ ค่า qualitative value ของ Top-down weight และ e_{pji} คือค่า Degree แสดงความเป็นสมาชิกของ B_{pji} โดยค่า e_{pji} ของ B_{pji} ในโครงข่ายนิวรอนนี้มีค่าอยู่ในช่วงระหว่าง 0 ถึง 1 ซึ่งค่า Degree นี้เป็นค่าที่ให้ระดับความเป็นสมาชิกของ B_{pji} กับ ข้อมูลเข้า ถ้า Degree ของ B_{pji} มีค่าเท่ากับ 0 ให้ความหมายว่า Qualitative value ของ B_{pji} ไม่ได้เป็นส่วนหนึ่งของข้อมูลเข้า j ถ้า Degree ของ B_{pji} มีค่าเท่ากับ 1 ให้ความหมายว่า Qualitative value ของ B_{pji} เป็นสมาชิกของข้อมูลเข้า j



รูปที่ 2.11 โครงสร้างของเทกอะแคปทีเพเรโซแนนทียรีนิวรอนเน็ตเวิร์ค

โครงข่ายมี R unit เพื่อใช้ในการรีเซตค่า vigilance ซึ่งเป็นค่าที่ใช้ควบคุมระดับความคล้ายคลึงของข้อมูลอินพุตกับค่าถ่วงน้ำหนักของโครงข่าย ทั้งนี้ค่า vigilance ที่ใช้จะมีค่าอยู่ในช่วง

ระหว่าง 0 ถึง 1 โดย vigilance ที่ใกล้ 0 หมายถึงข้อมูลอินพุตกับค่าถ่วงน้ำหนักของโครงข่ายมีความใกล้เคียงกันน้อย

จะเห็นได้ว่าโครงข่ายที่อะแดปทีฟเรโซแนนซ์เทียรีนิวโรลเน็ตเวิร์คมีความสามารถในการจัดกลุ่มข้อมูลเชิงข้อความได้ดี ดังผลการจัดกลุ่มข้อมูลข่าว Reuter-21578 โดยให้ค่าความถูกต้องถึง 85 เปอร์เซ็นต์ ทั้งนี้แสดงให้เห็นทั้งความสามารถในการเรียนรู้ของโครงข่ายอะแดปทีฟเรโซแนนซ์เทียรีและวิธีการหาความคล้ายคลึงของข้อมูลข้อความด้วย Similarity Measure ดังนั้นในงานวิจัยที่จะนำเสนอนี้จึงพัฒนาเพื่อเพิ่มขีดความสามารถของโครงข่ายเดิมให้สามารถใช้งานกับข้อมูลเชิงตัวเลขและข้อมูลข้อความในโครงข่ายเดียวกัน เพื่อลดขั้นตอนในการทำ preprocessing ดังจะได้นำเสนอในบทต่อไป

บทที่ 3

ยูนิฟายอะแดปทีฟเรโซแนนทีเยรี

ข้อมูลในความเป็นจริงที่ใช้ปัจจุบัน มักจะประกอบด้วยส่วนประกอบหรือฟิลด์ต่างๆของข้อมูล โดยส่วนประกอบนั้นมักจะมีรูปแบบที่หลากหลาย ทั้งนี้ขึ้นกับประเภทและลักษณะของข้อมูล เช่น ข้อมูลการพยากรณ์อากาศมักจะมีส่วนประกอบของข้อมูลจะเป็นตัวเลขทั้งหมด ตัวอย่างเช่น อุณหภูมิ ความเร็วลม ปริมาณน้ำฝน ในขณะที่ส่วนประกอบของข้อมูลประวัติลูกค้ามักจะมีทั้งข้อมูลที่เป็นตัวเลข ข้อความ ตัวอย่างเช่น อายุ อาชีพ

จากการศึกษางานวิจัยที่ผ่านมาพบว่าสามารถนำมาประยุกต์เพื่อใช้กับการจัดข้อมูลข้อความ โครงสร้างของอัลกอริทึมที่นำเสนอจึงพัฒนาให้รองรับประเภทของข้อมูลให้มากขึ้น โดยงานวิจัยได้เสนอการพัฒนาโครงข่ายพีชชีอะแดปทีฟเรโซแนนทีเยรีที่มีความสามารถในการจัดกลุ่มข้อมูลเชิงตัวเลขให้เป็นโครงข่ายที่สามารถจัดกลุ่มข้อมูลข้อความและข้อมูลเชิงตัวเลขได้

3.1 สถาปัตยกรรมของโครงข่ายยูนิฟายอะแดปทีฟเรโซแนนทีเยรี

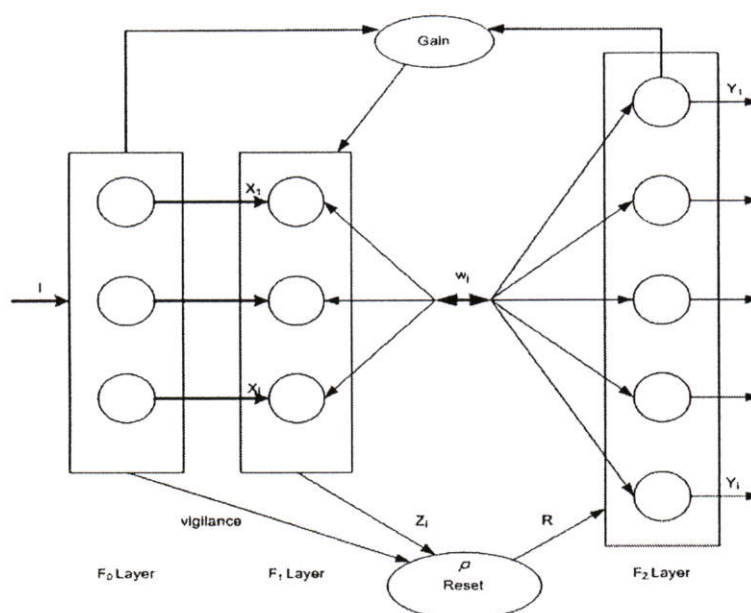
โครงข่ายพีชชีอะแดปทีฟเรโซแนนทีเยรีจากการนำเสนอของ S. Grossberg และ G. Carpenter [5] เป็นโครงข่ายประสาทเทียมที่พัฒนามาจากอะแดปทีฟเรโซแนนทีเยรี 1 โดยโครงสร้างโครงข่ายจะมี 3 ชั้นคืออินพุทเลเยอร์ (Input Layer) อินเตอร์เฟซเลเยอร์ (Interface Layer) และ เอาท์พุทเลเยอร์ (Output Layer) และมีค่าถ่วงน้ำหนักเพียงชุดเดียว

งานวิจัยของพีชชีอะแดปทีฟเรโซแนนทีเยรีจะเห็นว่าชั้นอินพุทเลเยอร์จะเป็นชั้นที่รับแพทเทิร์นของข้อมูลและส่งผ่านมายังชั้นอินเตอร์เฟซเลเยอร์ในลักษณะของเวกเตอร์ ส่วนในชั้นของอินเตอร์เฟซเลเยอร์ และ เอาท์พุทเลเยอร์ จะใช้ในการคำนวณเปรียบเทียบระหว่างค่าอินพุทแพทเทิร์นและค่าถ่วงน้ำหนักด้วย Choice Function และ Match Function โดยค่าจากการคำนวณ Choice Function จะเก็บในชั้นเอาท์พุทเลเยอร์ ในขณะที่ค่าจากการคำนวณด้วย Match Function จะเก็บในชั้นอินเตอร์เฟซเลเยอร์

3.1.1 โครงสร้างของโครงข่าย

ในงานวิจัยพัฒนาโครงข่ายยูนิฟายอะแดปทีฟเรโซแนนทีเยรีจากโครงข่ายพีชชีอะแดปทีฟเรโซแนนทีเยรี โดยโครงสร้างของโครงข่ายมี 3 ชั้นคืออินพุทเลเยอร์ (Input Layer) อินเตอร์เฟซเลเยอร์ (Interface Layer) และ เอาท์พุทเลเยอร์ (Output Layer) ดังรูปที่ 3.1 โดยในชั้นอินพุทจะประกอบไปด้วยโหนดต่างๆ ซึ่งแต่ละโหนดหมายถึงแต่ละคุณลักษณะหรือฟีเจอร์ (feature) ซึ่งสามารถรับได้ทั้งข้อมูลข้อความและข้อมูลตัวเลข โดยที่ต้องกำหนดโหนดว่าเป็นโหนดที่รับข้อมูลประเภทใด เพื่อ

รองรับข้อมูลที่มีค่าเป็นตัวเลขแต่ไม่ได้มีความหมายในเชิงตัวเลขเช่นในกรณีข้อมูลลำดับที่หรือข้อมูลที่มีการแปลงข้อความให้เป็นตัวเลขแล้ว



รูปที่ 3.1 โครงข่ายที่ใช้ในงานวิจัย

ชุดข้อมูลจากอินพุตเลเยอร์จะส่งต่อไปยังอินเตอร์เฟซเลเยอร์ของโครงข่ายซึ่งจะเป็นชั้นที่ปรับข้อมูลจากชั้นอินพุตให้เป็นเวกเตอร์ที่ใช้ในการเรียนรู้โครงข่ายต่อไป โดยสำหรับข้อมูลเชิงตัวเลขซึ่งต้องมีการ Normalize และ หาค่า Complement code ทั้งนี้จำนวน โหนดบนชั้นอินเตอร์เฟซเลเยอร์จะเท่ากับชั้นอินพุตเลเยอร์

เอาท์พุตเลเยอร์จะประกอบด้วยโหนดต่างๆ ซึ่งแต่ละโหนดหมายถึงกลุ่มของข้อมูลหรือคลัสเตอร์ที่โครงข่ายสร้างขึ้นหลังจากเรียนรู้ โดยการเรียนรู้ของโครงข่ายจะเริ่มต้นจากจำนวนกลุ่มข้อมูลเป็นศูนย์หรือก็คือไม่มีโหนดบนชั้นเอาท์พุตเลเยอร์ การสร้างโหนดบนเอาท์พุตเลเยอร์จะเป็นกระบวนการหนึ่งในการเรียนรู้ โดยงานวิจัยเลือกวิธีการจำกัดจำนวนคลัสเตอร์ให้โครงข่ายเพื่อแก้ปัญหาการขาด stability เนื่องจากพบปัญหาการแตกจำนวนกลุ่มออกเป็นจำนวนมาก ทำให้การเรียนรู้รอบใหม่มักเกิดการย้ายกลุ่มของข้อมูลเป็นจำนวนมาก นอกจากนี้ยังแก้ปัญหาเรื่องประสิทธิภาพของการจัดกลุ่ม เนื่องจากการแตกจำนวนกลุ่มไปเรื่อยๆจะได้คลัสเตอร์ที่มีข้อมูลจากคลาสเดียวซึ่งจะให้ค่า Purity สูง ในขณะที่แต่ละคลัสเตอร์อาจจะมีสมาชิกเพียงไม่กี่ตัวเมื่อเทียบกับจำนวนข้อมูลทั้งหมด ทำให้ค่า F-measures ต่ำ

นอกจากชั้นของโครงข่ายยังมีส่วนที่สำคัญของโครงข่ายคือ Weight Connection ดังรูปที่ 3.1 ซึ่งค่าถ่วงน้ำหนักถือเป็นความจำของโครงข่ายที่ใช้การเรียนรู้และแบ่งแยกกลุ่มข้อมูลออกเป็นคลัสเตอร์ต่างๆ

ในการพัฒนายูนิฟายอะแดปทีฟเรโซแนนทียรีได้มีการกำหนดการใช้ตัวแปรต่างๆในการนำเสนออัลกอริทึม ดังแสดงในตารางที่ 3.1

ตารางที่ 3.1 สัญลักษณ์ที่ใช้ในยูนิฟายอะแดปทีฟเรโซแนนทียรี

Notation	Description
p	จำนวนชุดข้อมูลทั้งหมด
C	Cluster ที่เกิดจากการเรียนรู้
m	จำนวนโหนดบน F_1 Layer ซึ่งมี i เป็นอินเด็กซ์
n	จำนวนโหนดบน F_2 Layer ซึ่งมี j เป็นอินเด็กซ์
J	อินเด็กซ์ของ Y ที่มีค่าสูงสุด
Y	ค่า Choice Function ของอินพุตกับแต่ละโหนดเอาต์พุต $Y = \{Y_1, Y_2, \dots, Y_n\}$
Z	ค่า Match Function ของโหนดเอาต์พุตกับค่าอินพุต
X	เซตของข้อมูล x ทั้งหมดที่ใช้ในการเรียนรู้ โดย $X = \{x_1, x_2, \dots, x_p\}$
x	ข้อมูลหรือแพทเทิร์นที่ F_1 Layer โดยที่ $x = (x_1, x_2, \dots, x_m)$
β	ค่า Learning Rate Parameter โดยมีค่าอยู่ในช่วง 0 ถึง 1
α	ค่า Choice Parameter
λ	ค่า Vigilance Decrease Rate
θ	เปอร์เซ็นต์ของจำนวนข้อมูลที่ยอมรับได้ สำหรับทดสอบการปรับค่า vigilance
ρ	ค่า Vigilance Parameter ซึ่งเป็นค่าที่กำหนดระดับความคล้ายคลึงของโครงข่าย
E	ค่าคิกริตั้งต้นของโครงข่ายสำหรับข้อมูลข้อความ
W	เซตค่าถ่วงน้ำหนักหรือแพทเทิร์น โดยที่ $W = \{w_1, w_2, \dots, w_n\}$
w_j	$w_j = (w_{j1}, w_{j2}, \dots, w_{jm})$
$w_{jm}(\text{text})$	ค่าถ่วงน้ำหนักของข้อมูลข้อความ $w_{jm} = (w_{jm1}(e_{jm1}), w_{jm2}(e_{jm2}), \dots)$
$w_{jm}(\text{num})$	ค่าถ่วงน้ำหนักของข้อมูลตัวเลข $w_{jm} = (w_{jm1}, w_{jm2})$
$S(A, B)$	ค่าความคล้ายคลึงของ A และ B ด้วยการคำนวณของ Similarity Measure
$T(x, w_j)$	Choice Function
$M(x, w_j)$	Match Function
$U(x, w_j)$	Update Function สำหรับค่าถ่วงน้ำหนักที่ j
$U(x, e_j)$	Update Function สำหรับคิกริตั้งต้นของค่าถ่วงน้ำหนักที่ j
$A \wedge B$	$A \wedge B = \min(A, B)$
$\ A\ $	$\ A\ = \sum_{k=1}^M A_k $ เมื่อ $A = (A_1, \dots, A_M)$

3.1.2 ค่าถ่วงน้ำหนักของโครงข่าย

จากงานวิจัยงานวิจัยของ Baraldi และ Parmiggiani [11] ที่ได้ทำการพิสูจน์ว่าการมีค่าถ่วงน้ำหนัก 2 ชุดของ ART นั้นมีค่าเท่ากับการมีค่าถ่วงน้ำหนักเพียงชุดเดียวของโครงข่ายแบบ Feed-forward และจากโครงสร้างของพีชชีอะแคปทีฟเรโซแนนทีเยร์ในงานวิจัยจึงใช้ค่าถ่วงน้ำหนักเพียงชุดเดียวโดยไม่มีการแยกเป็น bottom-up และ top-down

ค่าถ่วงน้ำหนักเป็นเวกเตอร์ที่เชื่อมระหว่างอินเตอร์เฟซเลเยอร์ กับ เอาท์พุทเลเยอร์โดยในงานวิจัยจะแทนค่าถ่วงน้ำหนักด้วย W ซึ่งแสดงในรูปเซตของข้อมูลได้เป็น $W = \{w_1, w_2, \dots, w_n\}$ โดย n หมายถึงจำนวนโหนดบน F_2 Layer หรือจำนวนคลัสเตอร์ของโครงข่าย และ w เป็นเวกเตอร์ของข้อมูลโดย $w_j = (w_{j1}, w_{j2}, \dots, w_{jm})$ โดย m หมายถึงจำนวนคุณลักษณะหรือพีเจอร์ของเวกเตอร์ ซึ่งโครงข่ายของยูนิฟายอะแคปทีฟเรโซแนนทีเยร์จะเริ่มต้นจากไม่มีโหนดบนเอาท์พุทเลเยอร์ ทำให้โครงข่ายไม่ต้องกำหนดค่าถ่วงน้ำหนักเริ่มต้นนั่นคือเริ่มต้นจากไม่มีค่าถ่วงน้ำหนักของโครงข่าย

ทั้งนี้ค่าถ่วงน้ำหนักสำหรับข้อมูลข้อความและข้อมูลเชิงตัวเลขจะมีความแตกต่างกัน โดยค่าถ่วงน้ำหนักสำหรับข้อมูลข้อความนั้นจะเป็นเซตของพีเจอร์และจะอ้างอิงร่วมกับค่าดิกรี e เพื่อบอกระดับของความน่าจะเป็นที่ข้อมูลจะเป็นตัวแทนของกลุ่มนั่นคือ ค่าถ่วงน้ำหนักของโหนด ที่ j พีเจอร์ที่ m สามารถเขียนได้เป็น $w_{jm} = (w_{jm1}(e_{jm1}), w_{jm2}(e_{jm2}), \dots)$ โดยระหว่างการเรียนรู้ถ้าดิกรีมีค่าเป็นศูนย์จะนำข้อมูลตัวนั้นออกจากการเป็นตัวแทนกลุ่ม ส่วนพีเจอร์สำหรับข้อมูลเชิงตัวเลขจะมีสมาชิกเพียง 2 ค่าโดยค่าแรกจะรองรับข้อมูลอินพุทและอีกค่าจะรองรับค่า Complement code คือ $w_{jm} = (w_{jm1}, w_{jm2})$

3.1.3 การเรียนรู้ของโครงข่าย

ชุดข้อมูลทั้งหมดหรืออบเจกต์ทั้งหมดที่ใช้ในงานวิจัยจะแทนด้วย X ซึ่งแสดงในรูปเซตของข้อมูลได้เป็น $X = \{x_1, x_2, \dots, x_p\}$ โดย p หมายถึงจำนวนข้อมูลทั้งหมด การเรียนรู้จะเริ่มจากชั้นอินพุทเลเยอร์รับ x_1 และส่งผ่านมายังอินเตอร์เฟซเลเยอร์โดยให้เวกเตอร์ x เป็นตัวแทนของข้อมูลโดย $x = (x_1, x_2, \dots, x_m)$ ซึ่ง m หมายถึงจำนวนคุณลักษณะหรือพีเจอร์ของเวกเตอร์ โดยที่ในงานวิจัยนี้ให้พีเจอร์สำหรับข้อมูลข้อความสามารถเป็นเซตของพีเจอร์ นั่นคือ $x_1 = (x_{11}, x_{12}, x_{13}, \dots)$ ส่วนพีเจอร์สำหรับข้อมูลเชิงตัวเลขจะต้องทำการปรับค่าหรือ Normalize ข้อมูลเชิงตัวเลขให้อยู่ระหว่าง 0 ถึง 1 เพื่อพร้อมสำหรับการคำนวณ Choice Function และหาค่า Complement code ทำให้ข้อมูลเชิงตัวเลขจะมีสมาชิก 2 ตัว $x_1 = (x_{11}, x_{12})$

การเรียนรู้ของโครงข่ายอะแคปทีฟเรโซแนนทีเยร์มีค่าตัวแปรหรือ Parameter ที่ต้องกำหนดในการเริ่มต้นของอัลกอริทึมได้แก่

- จำนวนโหนดบนชั้นอินพุทเลเยอร์: เป็นข้อมูลเริ่มต้นในการสร้างโครงข่าย

- ชนิดของโหนดว่าเป็นข้อความหรือเชิงตัวเลข: เพื่อป้องกันการความผิดพลาดในการตีความหมายของข้อมูลคงที่ได้กล่าวมาแล้ว
- จำนวนคลัสเตอร์สูงสุด: การเรียนรู้จะสร้างคลัสเตอร์ได้ไม่เกินจำนวนคลัสเตอร์สูงสุด
- เปอร์เซ็นต์ต่ำสุดของข้อมูลที่ไม่ผ่าน vigilance: เมื่อมีข้อมูลที่ไม่ผ่าน vigilance มากกว่าเปอร์เซ็นต์ของจำนวนข้อมูลที่กำหนดไว้ โครงข่ายจะทำการปรับค่า vigilance ลงด้วยค่า Vigilance decrease rate
- ค่า vigilance: ใช้ในการเปรียบเทียบกับค่าบนอินเตอร์เฟซเลเยอร์และเอาท์พุทเลเยอร์ ซึ่งค่า vigilance จะอยู่ในช่วง 0-1 โดยเมื่อเข้าใกล้ 1 จะหมายถึงความเหมือนที่มากขึ้น
- ค่าคิรีตั้งต้นสำหรับข้อมูลข้อความ: เป็นค่าคิรีว่าข้อมูลควรจะเป็นตัวแทนมากเท่าใด ซึ่งค่าคิรีจะอยู่ในช่วง 0-1 โดยเมื่อเข้าใกล้ 1 จะหมายถึงคิรีที่มากขึ้น
- ค่า Learning Rate: เป็นอัตราในการเรียนรู้โดยจะใช้ในการ เพิ่ม/ลด ค่าคิรีของข้อมูลข้อความ
- ค่า Vigilance decrease rate: เป็นอัตราการลดค่า vigilance
- จำนวนรอบของการเรียนรู้: สำหรับกำหนดรอบเพื่อหยุดการ เรียนรู้ เนื่องจากงานวิจัย ใช้เงื่อนไขจำนวนรอบเป็นเงื่อนไขการหยุดการเรียนรู้

การเรียนรู้ของโครงข่ายเกิดจากการเปรียบเทียบค่าอินพุทกับคลัสเตอร์ของโครงข่าย โดยฟังก์ชันที่ใช้ในการเปรียบเทียบได้แก่ Choice Function ที่ใช้ในการเปรียบเทียบความคล้ายคลึงระหว่างอินพุทกับคลัสเตอร์แต่ละคลัสเตอร์ เพื่อหาคลัสเตอร์ที่ให้ค่าการเปรียบเทียบสูงสุดเป็นคลัสเตอร์ที่ชนะ ส่วน Match Function จะใช้ในการเปรียบเทียบความคล้ายคลึงของคลัสเตอร์ที่ชนะกับอินพุท และนำมาทดสอบกับค่า vigilance

งานวิจัยของ M.Blume และ S.C.Esener [23] ที่เสนอการเปรียบเทียบในส่วนของ Match Function ให้เป็นการคำนวณหาค่า Match Function ของทุกคลัสเตอร์ เพื่อลดความซับซ้อนของโครงข่าย ในงานวิจัยนี้ได้ใช้การคำนวณหาค่า Match Function ของทุกคลัสเตอร์เพื่อใช้ในการจัดกลุ่มให้กับข้อมูลที่ทดสอบกับค่า vigilance ไม่ผ่าน

และด้วยวิธีการเปรียบเทียบความคล้ายคลึงที่จะเสนอในหัวข้อถัดไป จะให้ผลการคำนวณฟังก์ชันทั้ง 2 ฟังก์ชันในช่วงที่มากกว่า 0-1 ขึ้นกับจำนวนพีเจอร์ของข้อมูล ดังนั้นการกำหนด vigilance ในการเปรียบเทียบกับ Match Function ในงานวิจัยจะทำโดยการปรับค่า vigilance ตามแต่ละชุดข้อมูล

อีกฟังก์ชันหนึ่งที่มีความสำคัญในอัลกอริทึมคือ Update Function ซึ่งใช้ในการปรับค่าถ่วงน้ำหนักซึ่งเป็นตัวแทนของคลัสเตอร์ โดยสำหรับพีเจอร์ที่รองรับข้อมูลเชิงตัวเลขจะยังคงใช้

หลักการเดียวกันกับพีชชีอะแคปทีเพเรโซแน้นเทียร์ี่ ส่วนพีเจอร์ี่ที่รองรับข้อมูลจะใช้หลักการของอะแคปทีเพเรโซแน้นเทียร์ี่ 1

3.1.4 ฟังก์ชันการเปรียบเทียบความคล้ายคลึง

ในงานวิจัยใช้การเปรียบเทียบความคล้ายคลึงอยู่ 2 ส่วนคือ Choice Function และ Match Function ดังที่ได้กล่าวมาแล้ว ซึ่งจากโครงสร้างโครงข่ายที่รองรับข้อมูล 2 ประเภท ทำให้ฟังก์ชันที่ใช้ในการคำนวณแตกต่างกันในการเปรียบเทียบข้อมูลแต่ละประเภท

3.1.4.1 การเปรียบเทียบความคล้ายคลึงสำหรับข้อมูลที่เป็นตัวเลข

จากวิธีการเปรียบเทียบความคล้ายคลึง ของ K.C. Gowda และ E. Diday ซึ่งเสนอการเปรียบเทียบโดยใช้ Similarity Measure ตามหัวข้อที่ 2.5.2 พบว่าค่า Similarity Measure จะเท่ากับค่าของการเปรียบเทียบตำแหน่ง (position) นั่นคือ $S(A,B) = S_p(A,B)$ โดยที่

$$S_p(A,B) = 1 - \frac{|a_l - b_l|}{|U_k|}$$

ซึ่งจากสมการจะเห็นว่า $S_p(A,B) = S_p(B,A)$ ดังนั้นงานวิจัยจึงใช้

การเปรียบเทียบความคล้ายคลึงในส่วน Choice Function และ Match Function ตามอัลกอริทึมของ Fuzzy ART ตามหัวข้อที่ 2.3.2 คือ

– Choice Function

$$T(\mathbf{x}, \mathbf{w}_j) = \frac{\|\mathbf{x} \wedge \mathbf{w}_j\|}{\alpha + \|\mathbf{w}_j\|} \quad (3.1)$$

เป็นการเปรียบเทียบค่าอินพุตกับค่าถ่วงน้ำหนักแต่ละคลัสเตอร์ เพื่อหาคลัสเตอร์โหนดที่อินพุตมีความคล้ายคลึงมากที่สุด โดยที่ $\|\mathbf{x} \wedge \mathbf{w}_j\|$ คือ ค่าที่น้อยที่สุดระหว่าง \mathbf{x} ซึ่งเป็นค่าอินพุตและ \mathbf{w}_j ซึ่งเป็นค่าถ่วงน้ำหนักของคลัสเตอร์ที่ j ซึ่งในงานวิจัยที่เป็น Fast-Learning จะกำหนดให้ค่า $\alpha = 0$

– Match Function

$$M(\mathbf{x}, \mathbf{w}_j) = \frac{\|\mathbf{x} \wedge \mathbf{w}_j\|}{\|\mathbf{x}\|} \quad (3.2)$$

เป็นการเปรียบเทียบว่าโหนดที่ขณะนั้นมีความคล้ายคลึงกับอินพุตหรือไม่โดยที่ $\|\mathbf{x} \wedge \mathbf{w}_j\|$ คือ ค่าที่น้อยที่สุดระหว่าง \mathbf{x} ซึ่งเป็นค่าอินพุตและ \mathbf{w}_j ซึ่งเป็นค่าถ่วงน้ำหนักของคลัสเตอร์ที่ j โดยที่ j เป็นโหนดที่ขณะ

3.1.4.2 การเปรียบเทียบความคล้ายคลึงสำหรับข้อมูลที่เป็นข้อความ

การเปรียบเทียบข้อมูลข้อความได้ประยุกต์งานวิจัยของ K.C. Gowda และ E. Diday [3] ที่นำเสนอวิธีการจัดกลุ่มข้อมูลโดยการใช้ Similarity Measure คำนวณหาความ

คล้ายคลึงระหว่างเอกสารทีละ 2 ชุด เพื่อสร้างตารางเปรียบเทียบค่าความคล้ายคลึง แล้วนำค่าที่คำนวณได้มาเป็นตัวชี้วัดในการจัดกลุ่ม โดยที่การเปรียบเทียบความคล้ายคลึงของ **A** และ **B** นั้นจะมีค่าเท่ากับความคล้ายคลึงของ **B** และ **A** นั่นคือ การเปรียบเทียบความคล้ายคลึงของ K.C. Gowda และ E. Diday นั้น $S(\mathbf{A}, \mathbf{B}) = S(\mathbf{B}, \mathbf{A})$

ค่าความคล้ายคลึงของ **A** ที่มีต่อ **B** โดยที่ $\mathbf{A} = (a_1, a_2, \dots, a_k)$ เมื่อ **A** มีสมาชิกจำนวน k ตัวหรือมีคุณลักษณะของข้อมูล k ค่า โดยใช้ Similarity Measure ในงานวิจัยแทนด้วย $S(\mathbf{A}, \mathbf{B})$ ซึ่ง $S(\mathbf{A}, \mathbf{B}) = \sum_{k=1}^d S(a_k, b_k) = S_s(\mathbf{A}, \mathbf{B}) + S_c(\mathbf{A}, \mathbf{B}) + S_p(\mathbf{A}, \mathbf{B})$ โดยที่นิยามของสัญลักษณ์ในสมการเป็นตามตารางที่ 3.1

เมื่อนำมาประยุกต์ใช้กับข้อมูลประเภทข้อความ พบว่าความคล้ายคลึงของข้อมูลประเภทข้อความจะมีความคล้ายคลึงของตำแหน่งข้อมูล (position) เป็นศูนย์ และเมื่อนำมาใช้กับการเปรียบเทียบว่า **A** ควรจะอยู่กลุ่ม **B** หรือไม่ ดังนั้นจึงไม่พิจารณาความคล้ายคลึงของจำนวนข้อมูล (span) เนื่องจากจำนวนข้อมูลใน **B** จะเป็นเซตของข้อมูลที่บ่งบอกลักษณะของกลุ่มซึ่งจำนวนข้อมูลของ **A** มักจะมีค่าน้อยกว่า **B** และสามารถมีสมาชิกเพิ่มขึ้นเรื่อยๆ ดังนั้นการคำนวณหาความคล้ายคลึงของชุดข้อมูลจะเป็นความคล้ายคลึงของเนื้อหา นั่นคือ $S(\mathbf{A}, \mathbf{B}) = S_c(\mathbf{A}, \mathbf{B})$ ซึ่ง $S_c(\mathbf{A}, \mathbf{B}) = \frac{l_i}{l_s}$ ดังนั้น $S_c(\mathbf{A}, \mathbf{B})$ จะมีค่ามากที่สุดเมื่อ $l_s = l_i$ โดยที่ $l_s = l_a + l_b - l_i$ แต่เนื่องจาก แนวโน้ม l_b จะมีค่ามากขึ้นเพราะจำนวนสมาชิกของ **B** ที่เพิ่มมากขึ้นดังที่กล่าวมาแล้ว ดังนั้นในการคำนวณค่าความเหมือนของเนื้อหาสำหรับแต่ละฟังก์ชันจะเป็น

– Choice Function

$$T(\mathbf{x}, \mathbf{w}_j) = S(\mathbf{x}, \mathbf{w}_j) \quad (3.3)$$

จาก $S(\mathbf{A}, \mathbf{B}) = \frac{l_i}{l_s}$ เมื่อปรับมาใช้กับ $S(\mathbf{x}, \mathbf{w}_j)$ จะได้ว่า l_i หมายถึงจำนวนข้อมูลที่เหมือนกันของอินพุต \mathbf{x} กับค่าถ่วงน้ำหนักของโหนดที่ j และ l_s จะเป็นค่ามากที่สุดระหว่างจำนวนข้อมูลอินพุตกับ l_i นั่นคือ $l_s = \max(l_a, l_i)$ เมื่อ l_a แทนจำนวนข้อมูลอินพุต

– Match Function

$$M(\mathbf{x}, \mathbf{w}_j) = S(\mathbf{w}_j, \mathbf{x}) \quad (3.4)$$

จาก $S(\mathbf{B}, \mathbf{A}) = \frac{l_i}{l_s}$ เมื่อปรับมาใช้กับ $S(\mathbf{w}_j, \mathbf{x})$ จะได้ว่า l_i หมายถึงจำนวนข้อมูลที่เหมือนกันของอินพุต \mathbf{x} กับค่าถ่วงน้ำหนักของโหนดที่ j และ l_s จะเป็นค่ามากที่สุดระหว่างจำนวนข้อมูลของค่าถ่วงน้ำหนักโหนดที่ j กับ l_i นั่นคือ $l_s = \max(l_b, l_i)$ เมื่อ l_b แทนจำนวนข้อมูลของค่าถ่วงน้ำหนักโหนดที่ j

ตัวอย่างการคำนวณค่า Choice Function และ Match Function ด้วย Similarity Measure ที่ทำการปรับเพื่อให้เข้ากับงานวิจัย โดยสมมติโครงข่ายที่มีค่าถ่วงน้ำหนักของเอาท์พุท โหนดมีจำนวน 3 โหนด และจำนวนอินพุทโหนดมี 3 โหนด ให้ค่าถ่วงน้ำหนักเป็นดังตารางที่ 3.2 ซึ่งแต่ละแถวแสดงค่าถ่วงน้ำหนักของแต่ละคลัสเตอร์และคอลัมภ์แสดงข้อมูลของแต่ละฟีเจอร์ โดยข้อมูลในแต่ละฟีเจอร์เป็นข้อมูลข้อความและมีตัวเลขในวงเล็บแสดงคิกริชของข้อมูล

ตารางที่ 3.2 ตัวอย่างค่าถ่วงน้ำหนักของโครงข่ายที่รับข้อมูลข้อความ

	ฟีเจอร์ที่ 1	ฟีเจอร์ที่ 2	ฟีเจอร์ที่ 3
คลัสเตอร์ที่ 1	i(0.5); n(0.5)	nn (0.5)	kkk (0.5); nnn (0.5)
คลัสเตอร์ที่ 2	r (0.7); v (0.7); s (0.3); t (0.3)	uu (0.7); vv (0.7); rr (0.3); tt (0.3)	vvv (0.7); ttt (0.5); uuu (0.5)
คลัสเตอร์ที่ 3	h (0.5)	gg (0.5); hh (0.5)	ggg (0.5); hhh (0.5)

ตัวอย่างการคำนวณ สมมติข้อมูลอินพุทถัดมาเป็นเวกเตอร์ของข้อมูล 3 ฟีเจอร์ ($\{v\}$, $\{tt; vv\}$, $\{uuu; vvv\}$) การคำนวณค่า Choice Function สำหรับคลัสเตอร์ที่ 2 ได้จากผลรวมของแต่ละฟีเจอร์เท่ากับ $(1+1+1)=3$ โดยค่าการคำนวณแต่ละฟีเจอร์เป็น

- ฟีเจอร์ที่ 1 : $S(\mathbf{x}, \mathbf{w}_j) = l_i/l_s = 1$ โดยที่ $l_a = 1, l_i = 1 \therefore l_s = 1$
- ฟีเจอร์ที่ 2 : $S(\mathbf{x}, \mathbf{w}_j) = l_i/l_s = 1$ โดยที่ $l_a = 2, l_i = 2 \therefore l_s = 2$
- ฟีเจอร์ที่ 3 : $S(\mathbf{x}, \mathbf{w}_j) = l_i/l_s = 1$ โดยที่ $l_a = 2, l_i = 2 \therefore l_s = 2$

การคำนวณค่า Match Function สำหรับโหนดบนเอาท์พุทเลเยอร์เป็นการหาผลรวมค่าความคล้ายคลึงของแต่ละฟีเจอร์ ซึ่งการคำนวณหาค่า Match Function ของคลัสเตอร์ที่ 2 ได้เท่ากับ $(0.25+0.50+0.66)=1.41$ โดยค่าการคำนวณแต่ละฟีเจอร์เป็น

- ฟีเจอร์ที่ 1 : $S(\mathbf{w}, \mathbf{x}_j) = l_i/l_s = 0.25$ โดยที่ $l_a = 4, l_i = 1 \therefore l_s = 4$
- ฟีเจอร์ที่ 2 : $S(\mathbf{w}, \mathbf{x}_j) = l_i/l_s = 0.50$ โดยที่ $l_a = 4, l_i = 2 \therefore l_s = 4$
- ฟีเจอร์ที่ 3 : $S(\mathbf{w}, \mathbf{x}_j) = l_i/l_s = 0.66$ โดยที่ $l_a = 3, l_i = 2 \therefore l_s = 3$

ซึ่งค่าที่ได้จาก Match Function จะถูกนำไปเปรียบเทียบกับค่า Vigilance ต่อไป โดยจะเห็นได้ว่าค่า Match Function สูงสุดของแต่ละคลัสเตอร์ที่เป็นไปได้คือ 3 ดังนั้นถ้าในกรณีนี้ต้องการเปรียบเทียบกับ Vigilance เป็น 0.5 อาจทำได้โดยการปรับ vigilance เป็น $(0.5*3) = 1.5$

3.1.5 ฟังก์ชันการปรับค่าถ่วงน้ำหนัก

การปรับค่าถ่วงน้ำหนักอาจจะแบ่งออกได้เป็น 2 กรณีคือเป็นโหนดใหม่หรือโหนดเก่า ซึ่งในกรณีที่โหนดใหม่อาจถือว่าการตั้งค่าอินพุทให้เป็นค่าถ่วงน้ำหนักของโหนดใหม่นั้นและ

ให้ค่าดีกรีของข้อมูลข้อความเป็นค่าดีกรีตั้งต้น ส่วนในกรณีไหนคใหม่ จากที่โครงข่ายรองรับข้อมูลข้อความและข้อมูลเชิงตัวเลข วิธีการในการปรับค่าถ่วงน้ำหนักสำหรับพีเจอร์ที่รับข้อมูลต่างกันจะมีข้อแตกต่างกัน ซึ่งในกรณีของข้อมูลข้อความนั้นนอกจากการปรับค่าถ่วงน้ำหนักแล้วจะต้องทำการปรับค่าดีกรีของข้อมูลข้อความด้วย

- การปรับค่าถ่วงน้ำหนักสำหรับข้อมูลข้อความ

Update Function:

$$U(\mathbf{x}, \mathbf{w}_j) = (\mathbf{x} \cup \mathbf{w}_j) \quad (3.5)$$

Update Function สำหรับ Degree:

$$U(\mathbf{x}, \mathbf{e}_j) = \begin{cases} \mathbf{e}_j + \beta; & \text{if } \mathbf{w}_j \in \mathbf{x} \\ \mathbf{e}_j - \beta; & \text{if } \mathbf{w}_j \notin \mathbf{x} \\ E; & \text{if } \mathbf{x} \notin \mathbf{w}_j \end{cases} \quad (3.6)$$

- การปรับค่าถ่วงน้ำหนักสำหรับข้อมูลเชิงตัวเลข

Update Function:

$$U(\mathbf{x}, \mathbf{w}_j) = (1 - \beta)\mathbf{w}_j + \beta(\mathbf{x} \wedge \mathbf{w}_j) \quad (3.7)$$

ตารางที่ 3.3 ตัวอย่างค่าถ่วงน้ำหนักของโครงข่ายที่รับข้อมูลข้อความและเชิงตัวเลข

	พีเจอร์ที่ 1	พีเจอร์ที่ 2	พีเจอร์ที่ 3
คลัสเตอร์ที่ 1	i(0.5); n(0.5)	nn (0.5)	0.32, 0.65
คลัสเตอร์ที่ 2	r (0.7); v (0.7); s (0.3); t (0.3)	uu (0.7); vv (0.7); rr (0.3); tt (0.3)	0.54, 0.48
คลัสเตอร์ที่ 3	h (0.5)	gg (0.5); hh (0.5)	0.92, 0.06

ตัวอย่างการปรับค่าถ่วงน้ำหนักให้กับโหนดเก่า โดยสมมติให้ค่าถ่วงน้ำหนักเป็นดังตารางที่ 3.3 และข้อมูลอินพุตถัดมาเป็นเวกเตอร์ของข้อมูล 3 พีเจอร์ ($\{v\}$, $\{tt; vv\}$, $\{0.49, 0.51\}$) โดยคลัสเตอร์ที่ชนะคือคลัสเตอร์ที่ 2 หากกำหนดให้ค่าดีกรีตั้งต้นของข้อมูลข้อความเป็น 0.5 และ Learning Rate เป็น 0.2 การคำนวณค่าถ่วงน้ำหนักใหม่จะคำนวณให้แก่แต่ละพีเจอร์ได้เป็น

- พีเจอร์ที่ 1: $U(\mathbf{x}, \mathbf{w}_j) = \{v, r, s, t\}$ $U(\mathbf{x}, \mathbf{e}_j) = \{0.9, 0.5, 0.1, 0.1\}$
- พีเจอร์ที่ 2: $U(\mathbf{x}, \mathbf{w}_j) = \{tt, vv, uu, rr\}$ $U(\mathbf{x}, \mathbf{e}_j) = \{0.5, 0.9, 0.5, 0.1\}$
- พีเจอร์ที่ 3: $U(\mathbf{x}, \mathbf{w}_j) = \{0.49, 0.48\}$

ดังนั้นค่าถ่วงน้ำหนักของโครงข่ายภายหลังการปรับคลัสเตอร์ที่ 2 ตาม Update Function ได้เป็น $W_2 = (\{v, r, s, t\}, \{tt, vv, uu, rr\}, \{0.49, 0.48\})$

3.2 อัลกอริทึมที่ใช้ในงานวิจัย

จากการพัฒนาโครงข่ายและปรับการคำนวณของฟังก์ชันเปรียบเทียบความคล้ายคลึงเพื่อใช้ในงานวิจัย สามารถแสดงอัลกอริทึมที่ใช้ในงานวิจัยดังรูปที่ 3.2 ซึ่งจะเห็นว่าแบ่งออกได้เป็น 5 ส่วนหลักๆคือ การคำนวณ Choice Function การคำนวณ Match Function การทดสอบ Vigilance การปรับค่าถ่วงน้ำหนัก และการปรับค่า vigilance

การเปรียบเทียบความคล้ายคลึงด้วย Choice Function เป็นกระบวนการเปรียบเทียบความคล้ายคลึงว่าอินพุตมีความเหมือนกับแต่ละคลัสเตอร์เป็นเท่าใด ในขณะที่การเปรียบเทียบความคล้ายคลึงด้วย Match Function เป็นกระบวนการเปรียบเทียบความคล้ายคลึงว่าคลัสเตอร์มีความเหมือนกับอินพุตเป็นเท่าใด ซึ่งมีโอกาสที่โหนดที่ให้ค่าผลรวม Choice Function สูงสุด จะไม่ใช่โหนดที่ให้ค่าผลรวม Choice Function สูงสุด

โครงข่ายจะให้โหนดที่มีค่าผลรวมของ Choice Function สูงที่สุดเป็นโหนดที่ชนะ และจะทำการทดสอบ vigilance ของโหนดที่ชนะ โดยเป็นการเปรียบเทียบค่า vigilance กับผลรวมค่า Match Function ของโหนดนั้น ซึ่งหากค่าผลรวม Match Function มีค่าต่ำกว่า vigilance ก็จะไม่ผ่านการเปรียบเทียบ โครงข่ายก็จะ reset ค่า Choice Function ของโหนดนั้นให้เป็น -1 และหาโหนดที่ชนะต่อไป

การทดสอบ vigilance จะสิ้นสุดด้วย 2 กรณีคือสามารถหาโหนดที่ผ่านการทดสอบหรือไม่ มีโหนดใดผ่านการทดสอบ ซึ่งในกรณีหลังนั้น โครงข่ายจะต้องตัดสินใจว่าจะเพิ่ม โหนดหรือให้โหนดที่มีค่าผลรวม Match Function สูงสุดเป็นโหนดที่ผ่านการทดสอบ ทั้งนี้พิจารณาโดยประเมินจากจำนวนคลัสเตอร์สูงสุดที่อนุญาตให้โครงข่ายสร้างขึ้นใหม่ นั่นคือจะให้โหนดที่ให้ค่าผลรวม Match Function สูงสุดผ่านการทดสอบเมื่อไม่มีโหนดอื่นใดที่เหมาะสมเมื่อไม่สามารถสร้างโหนดใหม่

เมื่อได้โหนดที่ผ่านการทดสอบจากขั้นตอนที่ผ่านมา โครงข่ายจะทำการปรับค่าถ่วงน้ำหนัก ซึ่งจะมีโหนดที่ผ่านการทดสอบ 2 กรณีคือเป็นโหนดใหม่หรือโหนดเก่า ขั้นตอนการปรับค่าถ่วงน้ำหนักของโหนดใหม่คือการให้ค่าอินพุตเป็นค่าถ่วงน้ำหนักของโหนดใหม่นั้น ส่วนการปรับค่าถ่วงน้ำหนักของโหนดใหม่จะปรับตามการคำนวณดังอัลกอริทึม

จากขั้นตอนการทดสอบ vigilance หากมีข้อมูลจำนวนมากที่ไม่สามารถผ่านการทดสอบ โดยให้ค่าผลรวมของ Match Function สูงกว่า vigilance แสดงว่าค่า vigilance ที่โครงข่ายใช้ในการทดสอบมีค่าสูงเกินไป โครงข่ายก็จะทำการลดค่า vigilance ทั้งนี้ตัวแปรในงานวิจัยที่มีผลต่อการปรับค่า vigilance คือ เปอร์เซนต์ของจำนวนข้อมูลและอัตราการลดค่า vigilance

```

Initialize: Network Parameters
C = 0
maxC = max cluster
while (X not empty) {
Learning Loop:
  get x;
  new = true;

  Do: Choicing comparison
  foreach outputnode j: comparing x to each output node {
    foreach feature of x: comparing feature to feature {
      if (Text feature)  $T(x,w_j) = S(x,w_j)$ 
      if (Num feature)  $T(x,w_j) = \frac{\|x \wedge w_j\|}{\alpha + \|w_j\|}$ 
    }
     $Y_j = \text{sum } T(x,w_j) \text{ of input } x$ 
  }

  Do: Matching comparison
  reset = true;
  while (reset) { : find a winning node
    foreach feature of X: comparing feature to feature {
      if (Text feature)  $M(x,w_j) = S(w,x_j)$ 
      if (Num feature)  $M(x,w_j) = \frac{\|x \wedge w_j\|}{\|x\|}$ 
    }
     $Z_j = \text{sum } M(x,w_j) \text{ of input } x$ 
    J = index of max  $Y_j$ 
    K = index of max  $Z_j$ 

    Do: Vigilance testing
    if  $Z_j < \rho$  {
      if  $Y_j >= 0$  { : find new candidate
         $Y_j = -1$ 
        new = false;
        reset = true;
      } else { : all candidate reset
        if (C <= maxC) {
           $Y_j = -1$ 
          new = true;
          reset = false;
        } else {
          J = K
          new = false;
          reset = false;
        }
      }
    } else {
      new = false;
      reset = false;
    }
  }

  Do: Update Weight
  if (new) {
    newNode(new);
    foreach feature of X: {
       $w_j = x$ 
       $e_j = E$ 
    }
  } else {
    foreach feature of X: update feature memory {
      if (Text feature)
         $U(x,w_j) = (x \cup w_j)$ 
         $U(x,e_j) = \begin{cases} e_j + \beta; & \text{if } w_j \in x \\ e_j - \beta; & \text{if } w_j \notin x \\ E; & \text{if } x \notin w_j \end{cases}$ 
      if (Num feature)  $U(x,w_j) = (1-\beta)w_j + \beta(x \wedge w_j)$ 
    }
     $w_j = U(x,w_j)$ 
     $e_j = U(x,e_j)$ 
  }
}

Do: Update vigilance
if no. of data not pass vigilance >= ( $\theta$  percent of data) {
  decrease vigilance with  $\lambda$ 
}
}

```

รูปที่ 3.2 อัลกอริทึมที่ใช้ในงานวิจัย

อัลกอริทึมข้างต้นสามารถอธิบายเป็นขั้นตอนได้ดังต่อไปนี้

- 1) กำหนดค่าตัวแปรที่เป็นโครงสร้างของโครงข่าย
 - จำนวนโหนดบนชั้นอินพุทเลเยอร์
 - ชนิดของโหนดว่าเป็นข้อความหรือเชิงตัวเลข
 - จำนวนคลัสเตอร์สูงสุด
 - เปอร์เซนต์ต่ำสุดของข้อมูลที่ไม่ผ่าน vigilance
 - ค่า Vigilance
 - ค่าคิรีตั้งต้นสำหรับข้อมูลเชิงข้อความ (Default Degree)
 - ค่า Learning Rate
 - ค่า Vigilance decrease rate
 - จำนวนรอบของการเรียนรู้
- 2) ตรวจสอบเงื่อนไขการหยุด ถ้าเป็นจริงให้จบการเรียนรู้ของโครงข่าย ถ้าเป็นเท็จทำขั้นตอนที่ 3-8 ต่อไป
- 3) สำหรับแต่ละชุดข้อมูล ส่งเข้ามาจะทำการประมวลผลชั้นอินเตอร์เฟซเลเยอร์ โดยถ้าเป็นข้อมูลตัวเลขจะทำการ Normalize และหาค่า complement code ก่อนทำขั้นตอนต่อไปและตั้งค่าตัวแปรสำหรับเรียนรู้ข้อมูลเป็นดังนี้
 - การสร้างโหนดใหม่เป็น จริง
- 4) เปรียบเทียบอินพุทกับทุกคลัสเตอร์บนชั้นเอาต์พุท โดยคำนวณออกมาเป็นค่า Y ของแต่ละคลัสเตอร์ ซึ่งได้จากผลรวมจากการคำนวณของทุกพีเจอร์ของคลัสเตอร์ด้วย Choice function
 - สำหรับข้อมูลเชิงข้อความ

$$T(\mathbf{x}, \mathbf{w}_j) = S(\mathbf{x}, \mathbf{w}_j) \quad (3.8)$$
 - สำหรับข้อมูลตัวเลข

$$T(\mathbf{x}, \mathbf{w}_j) = \frac{\|\mathbf{x} \wedge \mathbf{w}_j\|}{\alpha + \|\mathbf{w}_j\|} \quad (3.9)$$
- 5) เปรียบเทียบอินพุทกับทุกคลัสเตอร์บนชั้นเอาต์พุท โดยคำนวณออกมาเป็นค่า Z ของแต่ละคลัสเตอร์ ซึ่งได้จากผลรวมจากการคำนวณของทุกพีเจอร์ของคลัสเตอร์ด้วย Match function
 - สำหรับข้อมูลเชิงข้อความ

$$M(\mathbf{x}, \mathbf{w}_j) = S(\mathbf{w}_j, \mathbf{x}) \quad (3.10)$$

– สำหรับข้อมูลตัวเลข

$$M(\mathbf{x}, \mathbf{w}_j) = \frac{\|\mathbf{x} \wedge \mathbf{w}_j\|}{\|\mathbf{x}\|} \quad (3.11)$$

6) หาโหนด J ที่ให้ค่า Choice Function มากที่สุด $J = \max(Y_i)$ และ โหนด K ที่ให้ค่า Match Function มากที่สุด

7) ทดสอบค่า Z กับ vigilance โดยแบ่งเป็น 2 กรณีคือ

a. Z น้อยกว่า จะ reset ค่า Y ของโหนดเป็น -1 และหาโหนดตัวแทนต่อไป จนกว่าทุกโหนด reset หมด ซึ่งถ้า reset หมดจะตรวจสอบจำนวน cluster โดยจะสร้างโหนดใหม่หากจำนวน cluster ยังไม่เท่ากับจำนวน cluster สูงสุด ซึ่งกรณีที่ไม่สามารถสร้างโหนดใหม่ จะให้โหนดที่ขณะเป็นโหนดที่ให้ค่า Match Function สูงสุด

b. Z มากกว่าหรือเท่ากับจะเกิดการ resonance โดยเข้าสู่การปรับค่าถ่วงน้ำหนักต่อไป

8) คำนวณค่าถ่วงน้ำหนักโดยมี 2 กรณี

a. กรณีสร้างโหนดใหม่ จะใช้ค่าอินพุตเป็นค่าถ่วงน้ำหนักของโหนดใหม่

b. กรณีปรับค่าถ่วงน้ำหนักโหนดเดิม

– สำหรับข้อมูลเชิงข้อความ

$$U(\mathbf{x}, \mathbf{w}_j) = (\mathbf{x} \cup \mathbf{w}_j) \quad (3.12)$$

และ Update Function สำหรับ Degree ของข้อมูลดังนี้

$$U(\mathbf{x}, \mathbf{e}_j) = \begin{cases} \mathbf{e}_j + \beta; & \text{if } \mathbf{w}_j \in \mathbf{x} \\ \mathbf{e}_j - \beta; & \text{if } \mathbf{w}_j \notin \mathbf{x} \\ E; & \text{if } \mathbf{x} \notin \mathbf{w}_j \end{cases} \quad (3.13)$$

– สำหรับข้อมูลตัวเลข

$$U(\mathbf{x}, \mathbf{w}_j) = (1 - \beta)\mathbf{w}_j + \beta(\mathbf{x} \wedge \mathbf{w}_j) \quad (3.14)$$

9) ทดสอบเงื่อนไขการปรับ vigilance โดยถ้าไม่ผ่านจะทำการปรับลดค่า vigilance

10) ทดสอบเงื่อนไขการหยุด เช่น นับจำนวนรอบของการเรียนรู้ หรือเปรียบเทียบผลการเรียนรู้กับครั้งก่อน แล้วส่งผลการทดสอบเงื่อนไขการหยุดไปขั้นตอนที่ 2 เพื่อพิจารณาการหยุด

3.3 ตัวอย่างการเรียนรู้ของโครงข่าย

จากโครงสร้างของโครงข่ายที่สามารถใช้งานกับข้อมูลเชิงตัวเลขและข้อมูลข้อความได้ ดังนั้นการประยุกต์ใช้งานกับข้อมูลดังตารางที่ 3.4 ซึ่งเป็นข้อมูล Micro-computer ที่มีพีเจอร 5 พีเจอรคือ Display, MicroProcessor, RAM, ROM, Keys ซึ่งมีข้อมูลทั้ง 2 ประเภทโดยที่ 2 พีเจอรแรกเป็นข้อมูลข้อความและข้อมูลเชิงตัวเลข 4 พีเจอร

ชุดข้อมูลจากตารางที่ 3.4 สามารถแสดงได้เป็น {'Color TV','6502',48,10,52} โดยเวกเตอร์นี้สามารถเข้าเรียนรู้ในโครงข่ายที่นำเสนอได้โดยตรง ในขณะที่ถ้าใช้การจัดกลุ่มด้วย Hierarchical หรือ K-means จะต้องทำการเปลี่ยนข้อมูลให้อยู่ในรูปเชิงตัวเลขทั้งหมด {1,1,48,10,52} โดย 1 แทน Color TV ในพีเจอรแรก และแทน 6502 ในพีเจอรถัดมา

สำหรับข้อมูลตัวเลขที่มีลักษณะเป็นช่วงของข้อมูล เพื่อให้สามารถใช้งานกับโครงข่ายได้ ในงานวิจัยนี้จะทำการแปลงรูปข้อมูลดังกล่าวให้เป็นข้อมูลเชิงตัวเลขโดยใช้ค่าเฉลี่ยเป็นตัวแทนของข้อมูลนั้น เช่นข้อมูล Micro-computer Atari800 มีข้อมูล Keys เป็นช่วงคือ 57-63 ในงานวิจัยจึงทำการแปลงรูปโดยใช้ค่าเฉลี่ยคือ 60 มาใช้เป็นตัวแทนของข้อมูล Keys เพื่อให้โครงข่ายเรียนรู้ต่อไป

ตารางที่ 3.4 ข้อมูล Micro-computer

Microcomputer	Original					Transform	
	Display	MP	RAM	ROM	Keys	ROM	Keys
0 Apple II	Color TV	6502	48	10	52	10	52
1 Atari 800	Color TV	6502	48	10	57-63	10	60
2 Commodore VIC 20	Color TV	6502A	32	11-16	64-73	13.5	68.5
3 Exidi Sorcerer	B&W TV	Z80	48	4	57-63	4	60
4 Zenith H8	Built-in	8080A	64	1	64-73	1	68.5
5 Zenith H89	Built-in	Z80	64	8	64-73	8	68.5
6 HP-85	Built-in	HP	32	80	92	80	92
7 Horizon	Terminal	Z80	64	8	57-63	8	60
8 Ohio Sc. Challenger	B&W TV	6502	32	10	53-56	10	54.5
9 Ohio Sc. II Series	B&W TV	6502C	48	10	53-56	10	54.5
10 TRS-80 I	B&W TV	Z80	48	12	53-56	12	54.5
11 TRS-80 III	Built-in	Z80	48	14	64-73	14	68.5

ผลลัพธ์ที่ได้จากโครงข่ายที่นำเสนอจะมี 2 ส่วนคือ คลัสเตอร์ที่เกิดขึ้น และค่าถ่วงน้ำหนัก หลังจากการเรียนรู้ โดยค่าถ่วงน้ำหนักของแต่ละคลัสเตอร์เสมือนเป็นเทมเพลตหรือรูปแบบของคลัสเตอร์ เนื่องจากค่าถ่วงน้ำหนักเป็น long-term memory ของโครงข่าย ที่ใช้ในการเปรียบเทียบความคล้ายคลึงกับข้อมูลอินพุตเพื่อตัดสินใจว่าคลัสเตอร์ใดเหมาะสมที่สุดสำหรับชุดข้อมูลอินพุต

3.3.1 การเรียนรู้ของโครงข่าย

ในหัวข้อนี้จะแสดงตัวอย่างการคำนวณ เพื่อแสดงถึงขั้นตอนการเรียนรู้ของโครงข่ายตามอัลกอริทึมที่แสดงข้างต้น โดยใช้ข้อมูล

ข้อมูลที่ 1 ({'j', 'n'}, {'nn'}, 60, 107)

ข้อมูลที่ 2 ({'t', 'u', 'v'}, {'tt', 'vv'}, 24, 79)

ข้อมูลที่ 3 ({'m', 'n'}, {'jj', 'mm', 'nn'}, 60, 106)

ข้อมูลที่ 4 ({'g', 'h'}, {'dd', 'hh'}, 2, 95)

เริ่มต้นการคำนวณตามแต่ละขั้นตอนของอัลกอริทึมได้ดังนี้

กำหนดค่าตัวแปรที่เป็น โครงสร้างของโครงข่าย

- จำนวนโหนดบนชั้นอินพุตเลเยอร์เป็น 4 โหนด
 - โหนด 2 โหนดแรกเป็นข้อมูลข้อความและที่เหลือเป็นข้อมูลเชิงตัวเลข
 - จำนวนคลัสเตอร์สูงสุด เป็น 3
 - เปอร์เซ็นต์ต่ำสุดของข้อมูลที่ไม่ว่าน vigilance เป็น 30
 - ค่า Vigilance เป็น 0.7 แต่ค่าสูงสุดของความคล้ายคลึงที่เป็นไปได้คือ 3 ดังนั้นปรับค่า vigilance เป็น 2.1
 - ค่าคิรีตั้งต้นสำหรับข้อมูลเชิงข้อความเป็น 0.5
 - ค่า Learning Rate เป็น 0.2
 - ค่า Vigilance decrease rate เป็น 0.02
 - จำนวนรอบของการเรียนรู้เป็น 1 รอบ
- 1) เงื่อนไขการหยุดคือเรียนรู้ 1 รอบ ดังนั้นทำขั้นตอนที่ 3-8 ต่อไป โดยในขั้นตอนที่ 3-8 นั้นจะมีจำนวนรอบเท่ากับจำนวนข้อมูล

รอบที่ 1

- 2) สำหรับแต่ละชุดข้อมูลที่เข้ามา ทำการประมวลผลชั้นอินเตอร์เฟซเลเยอร์ โดยสำหรับ ข้อมูลตัวเลขจะทำการ Normalize และหาค่า complement code ก่อน

ข้อมูลที่ 1: ($\{j, 'n'\}$, $\{nn'\}$, 60, 107)

$X = (\{j, 'n'\}$, $\{nn'\}$, 0.24, 0.12753, 0.76, 0.87247)

newNode = True

$Y = \{-1\}$

$Z = \{0\}$

- 3) คำนวณหาค่า Y ซึ่งในที่นี้ยังไม่มีโหนดบนเอาต์พุต ทำให้ไม่มีโหนดให้เปรียบเทียบกับอินพุต ค่า Y จึงไม่ได้คำนวณเพิ่มเติมแต่เป็นค่าที่ตั้งไว้

$Y = \{-1\}$

- 4) คำนวณหาค่า Z ซึ่งในที่นี้ยังไม่มีโหนดบนเอาต์พุต ทำให้ไม่มีโหนดให้เปรียบเทียบกับอินพุต ค่า Z จึงไม่ได้คำนวณเพิ่มเติมแต่เป็นค่าที่ตั้งไว้

$Z = \{0\}$

- 5) โหนด J ที่มีค่ามากที่สุด ในที่นี้คือโหนด 1 โดยค่า $\max(Y_i) = -1$ และโหนด K ที่ให้ค่าสูงสุดคือโหนดที่ 1 โดยค่า $\max(Z_i) = -1$

- 6) ทดสอบค่า vigilance ค่า Z น้อยกว่า vigilance โดยค่า Y ของโหนดเป็น -1 และทุกโหนด และจำนวนคลัสเตอร์ยังไม่เกินจำนวนคลัสเตอร์สูงสุด ให้ค่าการสร้างโหนดใหม่เป็นจริง

- 7) การสร้างโหนดใหม่เป็นจริง ดังนั้นสร้างโหนดใหม่และให้ค่าถ่วงน้ำหนักของโหนดเป็นค่าอินพุต

$W = [W1]$

$W1 = [\{j (0.5), n (0.5)\}, \{nn (0.5)\}, 0.24, 0.12753, 0.76, 0.87247]$

- 8) ผ่านการทดสอบการปรับ vigilance เนื่องจากเป็นการสร้างโหนดใหม่

รอบที่ 2

- 3) สำหรับแต่ละชุดข้อมูลที่เข้ามา ทำการประมวลผลชั้นอินเตอร์เฟซเลเยอร์ โดยสำหรับ ข้อมูลตัวเลขจะทำการ Normalize และหาค่า complement code ก่อน

ข้อมูลที่ 2: ({'t', 'u', 'v'}, {'rr', 'vv'}, 24, 79)

$X = \{ \{ 't', 'u', 'v' \}, \{ 'rr', 'vv' \}, 0.096, 0.09416, 0.904, 0.90584 \}$

newNode = True

$Y = \{-1\}$

$Z = \{0\}$

- 4) คำนวณหาค่า Y

$$Y = \left\{ 0/3 + 0/2 + \frac{|0.096 + 0.09416 + 0.76 + 0.87247|}{|0.096 + 0.09416 + 0.904 + 0.90584|} \right\}$$

$Y = \{0+0+(1.82263/2)\}$

$Y = \{0.91131\}$

- 5) คำนวณค่า Z

$$Z = \left\{ 0 + 0 + \frac{|0.096 + 0.09416 + 0.76 + 0.87247|}{|0.24 + 0.76 + 0.12753 + 0.87247|} \right\}$$

$Z = \{0+0+(1.82263/2)\}$

$Z = \{0.91131\}$

- 6) โหนด J ที่มีค่าสูงสุดคือ โหนดที่ 1 และ โหนด K ที่มีค่าสูงสุดคือ โหนดที่ 1
- 7) ค่า $Z < \text{vigilance}$ เป็นจริง ดังนั้นให้ $Y = -1$ ทั้งนี้ทุกโหนดเป็น -1 และจำนวนคลัสเตอร์ ยังไม่เกินคลัสเตอร์สูงสุด ให้การสร้างโหนดใหม่เป็นจริง
- 8) จากที่ newNode = true ดังนั้นทำการสร้างโหนดบนเอาท์พุทเลเยอร์ใหม่โดยมีค่าถ่วงน้ำหนักคือค่าอินพุท

$W = [W1, W2]$

$W1 = [\{ j(0.5), n(0.5) \}, \{ nn(0.5) \}, 0.24, 0.12753, 0.76, 0.87247]$

$W2 = [\{ t(0.5), u(0.5), v(0.5) \}, \{ rr(0.5), vv(0.5) \}, 0.096, 0.09416, 0.904, 0.90584]$

- 9) ผ่านการทดสอบการปรับ vigilance เนื่องจากเป็นการสร้างโหนดใหม่

รอบที่ 3

- 3) สำหรับแต่ละชุดข้อมูลที่เข้ามา ทำการประมวลผลชั้นอินเตอร์เฟซเลเยอร์ โดยสำหรับข้อมูลตัวเลขจะทำการ Normalize และหาค่า complement code ก่อน

ข้อมูลที่ 3: ({'m', 'n'}, {'jj', 'mm', 'nn'}, 60, 106)

$X = \{ \{ 'm', 'n' \}, \{ 'jj', 'mm', 'nn' \}, 0.24, 0.12634, 0.76, 0.87366 \}$

newNode = True

$Y = \{-1\}$

$Z = \{0\}$

- 4) คำนวณหาค่า Y

$Y = \{1.8327, 0.9119\}$ โดยที่

$$Y1 = \left\{ 1/2 + 1/3 + \frac{|0.24 + 0.12634 + 0.76 + 0.87247|}{|0.24 + 0.76 + 0.12753 + 0.87247|} \right\}$$

$$Y1 = \{0.5 + 0.3333 + (1.9988/2)\} = 1.8327$$

$$Y2 = \left\{ 0/2 + 0/3 + \frac{|0.096 + 0.09416 + 0.76 + 0.87366|}{|0.096 + 0.09416 + 0.904 + 0.90584|} \right\}$$

$$Y2 = \{0 + 0 + (1.8238/2)\} = 0.9119$$

- 5) คำนวณค่า Z

$Z = \{2.4994, 0.9119\}$ โดยที่

$$Z = \left\{ 1/2 + 1/1 + \frac{|0.24 + 0.12634 + 0.76 + 0.87247|}{|0.24 + 0.12634 + 0.76 + 0.87366|} \right\}$$

$$Z1 = \{0.5 + 1 + (1.9988/2)\} = 2.4994$$

$$Z_2 = \left\{ 0/3 + 0/2 + \frac{|0.096 + 0.09416 + 0.76 + 0.87366|}{|0.096 + 0.09416 + 0.904 + 0.90584|} \right\}$$

$$Z2 = 0 + 0 + 0.9119 = 0.9119$$

- 6) โหนด J ที่มีค่าสูงสุดคือ โหนดที่ 1 และ โหนด K ที่มีค่าสูงสุดคือ โหนดที่ 1
 7) ค่า $Z < \text{vigilance}$ เป็นเท็จ ดังนั้นให้การสร้างโหนดใหม่เป็นเท็จ
 8) newNode = false ดังนั้นทำการ Update ค่าถ่วงน้ำหนักของโหนด 1 ได้เป็น

$$W = [W1, W2]$$

$$W1 = [\{j, n, m\}, \{nn, jj, mm\}, 0.24, 0.12729, 0.76, 0.87247]$$

$$W2 = [\{t, u, v\}, \{rr, vv\}, 0.096, 0.09416, 0.904, 0.90584]$$

โดยที่ ค่า degree สำหรับข้อมูลข้อความ W11, W12, W21 และ W22 เป็น

$$- W11 = \{j (0.3), n (0.7), m(0.5)\} \text{ และ } W12 = \{nn (0.7), jj(0.5), mm(0.5)\}$$

$$- W21 = \{t(0.5), u(0.5), v(0.5)\} \text{ และ } W22 = \{rr(0.5), vv(0.5)\}$$

9) ผ่านการทดสอบการปรับ vigilance

รอบที่ 4

3) สำหรับแต่ละชุดข้อมูลที่เข้ามา ทำการประมวลผลชั้นอินเตอร์เฟซเลเยอร์ โดยสำหรับ ข้อมูลตัวเลขจะทำการ Normalize และหาค่า complement code ก่อน

ข้อมูลที่ 4: ({'g', 'h'}, {'dd', 'hh'}, 2, 95)

$$X = (\{'g', 'h'\}, \{'dd', 'hh'\}, 0.008, 0.11323, 0.992, 0.88677)$$

$$\text{newNode} = \text{True}$$

$$Y = \{-1\}$$

$$Z = \{0\}$$

4) คำนวณหาค่า Y

$$Y = \{0.87695, 0.94646\} \text{ โดยที่}$$

$$Y1 = \left\{ 0/2 + 0/2 + \frac{|0.008 + 0.11323 + 0.76 + 0.87247|}{|0.24 + 0.12729 + 0.76 + 0.87247|} \right\}$$

$$Y1 = \{0+0+0.87695\} = 0.87695$$

$$Y2 = \left\{ 0/2 + 0/2 + \frac{|0.008 + 0.09416 + 0.904 + 0.88677|}{|0.096 + 0.09416 + 0.904 + 0.90584|} \right\}$$

$$Y1 = \{0+0+0.94646\} = 0.94646$$

5) คำนวณค่า Z = {0.87695, 0.94646, }

$$Z = \left\{ 0/3 + 0/3 + \frac{|0.008 + 0.09416 + 0.904 + 0.88677|}{|0.008 + 0.11323 + 0.992 + 0.88677|} \right\}$$

$$Z1 = 0.87695$$

$$Z = \left\{ 0/3 + 0/3 + \frac{|0.008 + 0.09416 + 0.904 + 0.88677|}{|0.008 + 0.11323 + 0.992 + 0.88677|} \right\}$$

$$Z2 = 0.94646$$

- 6) โหนด J ที่มีค่าสูงสุดคือ โหนดที่ 2 และ โหนด K ที่มีค่าสูงสุดคือ โหนดที่ 2
- 7) ค่า Z ของโหนดที่ 2 น้อยกว่า vigilance ดังนั้นให้ Y ของโหนดที่ 2 เป็น -1 และทดสอบค่า Z ของโหนดที่ 1 ซึ่งก็น้อยกว่า vigilance ดังนั้นจะพิจารณาจำนวนคลัสเตอร์ซึ่งยังไม่เกินจำนวนคลัสเตอร์สูงสุด ให้การสร้างโหนดใหม่เป็นจริง
- 8) จากที่การสร้างโหนดใหม่เป็นจริง ทำการสร้างโหนดบนเอาต์พุตเลเยอร์ใหม่โดยมีค่าถ่วงน้ำหนักคือค่าอินพุต

$$W = [W1, W2, W3]$$

$$W1 = [\{j, n, m\}, \{nn, jj, mm\}, 0.24, 0.12729, 0.76, 0.87247]$$

$$W2 = [\{t, u, v\}, \{rr, vv\}, 0.096, 0.09416, 0.904, 0.90584]$$

$$W3 = [\{g, h\}, \{dd, hh\}, 0.008, 0.11323, 0.992, 0.88677]$$

โดยที่ ค่า degree สำหรับข้อมูลข้อความ W11, W12, W21 และ W22 เป็น

$$- W11 = \{j(0.3), n(0.7), m(0.5)\} \text{ และ } W12 = \{nn(0.7), jj(0.5), mm(0.5)\}$$

$$- W21 = \{t(0.5), u(0.5), v(0.5)\} \text{ และ } W22 = \{rr(0.5), vv(0.5)\}$$

$$- W31 = \{g(0.5), h(0.5)\} \text{ และ } W32 = \{dd(0.5), hh(0.5)\}$$

- 9) ผ่านการทดสอบการปรับ vigilance เนื่องจากเป็นการสร้างโหนดใหม่
- 10) เรียนรู้ครบ 1 รอบ ให้จำนวนรอบการเรียนรู้เป็น 1

จากการเรียนรู้ข้อมูลข้างต้นโครงข่ายสามารถจัดกลุ่มข้อมูลออกได้เป็น 3 คลัสเตอร์ โดยสามารถสรุปผลการจัดกลุ่มได้ดังตารางที่ 3.5 ซึ่งผลการจัดกลุ่มดังตารางในส่วนของสมาชิกคลัสเตอร์เกิดจากการเก็บข้อมูลในระหว่างการเรียนรู้ ส่วนของตัวแทนคลัสเตอร์นั้นได้จากค่าถ่วงน้ำหนักภายหลังการเรียนรู้ของโครงข่าย

จากที่โครงข่ายที่ใช้ในงานวิจัยสามารถใช้งานกับข้อมูลข้อความได้โดยตรงจะเห็นว่าผลการจัดกลุ่มข้อมูลในรูปแบบตัวแทนคลัสเตอร์นั้น โครงข่ายสามารถระบุสมาชิกที่เป็นตัวแทนข้อมูลข้อความของกลุ่มได้โดยตรง

ตารางที่ 3.5 คลัสเตอร์จากการเรียนรู้ของโครงข่าย

คลัสเตอร์ที่	สมาชิกในคลัสเตอร์	ตัวแทนของคลัสเตอร์
1	ข้อมูลที่ 1	{j, n, m}, {nn, jj, mm}, 0.24, 0.12729, 0.76, 0.87247
2	ข้อมูลที่ 2,3	{t, u, v}, {rr, vv}, 0.096, 0.09416, 0.904, 0.90584
3	ข้อมูลที่ 4	{g, h}, {dd, hh}, 0.008, 0.11323, 0.992, 0.88677

ทั้งนี้ผลการจัดกลุ่มในส่วนของสมาชิกในคลัสเตอร์จะสามารถนำมาใช้ในการประเมินประสิทธิภาพต่อไป ส่วนของตัวแทนของคลัสเตอร์ในกรณีที่ใช้งานกับข้อมูลที่มีจำนวนมากนั้นอาจมีจำนวนตัวแทนของคลัสเตอร์มาก ดังนั้นการทดลองในบทที่ 4 จะไม่นำเสนอในส่วนของตัวแทนของคลัสเตอร์ที่ได้ภายหลังการจัดกลุ่ม

บทที่ 4

การทดลองและผลการทดลอง

งานวิจัยได้ทดลองจัดกลุ่มกับข้อมูลที่มีข้อมูลคลาส ซึ่งเป็นข้อมูลที่บอกกลุ่มของชุดข้อมูล โดยใช้ชุดข้อมูลจาก <http://www.ics.uci.edu/~mllearn/MLRepository.html> ของภาควิชา Information and Computer Sciences ของ University of California, Irvine ซึ่งได้รวบรวมชุดข้อมูลสำหรับการทดสอบอัลกอริทึมการจัดกลุ่มไว้ และได้สร้างชุดข้อมูลทดลองขึ้นมาภายใต้ข้อกำหนดเพื่อทดสอบผลการจัดกลุ่มอีกด้วย

การประเมินผลการจัดกลุ่มหรือ Cluster evaluation เป็นกระบวนการในการทำ Cluster Analysis ซึ่งการประเมินผลอาจจะดูได้จาก ความถูกต้องของจำนวนกลุ่ม การเปรียบเทียบกับข้อเท็จจริง การเปรียบเทียบความสัมพันธ์กลุ่มที่เกิดขึ้น ทั้งนี้ขึ้นกับวิธีการจัดกลุ่มว่าเป็นในรูปแบบใด เช่น K-means มักจะประเมินผลด้วยค่า SSE (Sum square error) ดังนั้นจะสามารถแบ่งการประเมินผลออกได้เป็น Unsupervised, Supervised และ Relative [17]

Unsupervised เปรียบเทียบโดยไม่มีข้อมูลภายนอก เช่นการคำนวณหา SSE ซึ่งโดยทั่วไปจะมีการวัด 2 อย่างคือ cluster cohesion เป็นการวัดความสัมพันธ์ภายในกลุ่มที่เกิดขึ้นและ cluster separation จะเป็นการวัดความสัมพันธ์ระหว่างกลุ่ม ในขณะที่ Supervised จะเป็นการพิจารณาผลลัพธ์ของกลุ่มที่กำหนดได้ (cluster label) กับกลุ่มที่แท้จริง (class label) ส่วน Relative จะเป็นการเปรียบเทียบกลุ่มที่เกิดขึ้น โดยจะเลือกวิธีการ Unsupervised หรือ Supervised ก็ได้

ในการนำเสนอการทดลองจัดกลุ่มด้วยโครงข่ายยูนิฟายอะแคปทีเพร โชแน้นเทียร์แล้ว งานวิจัยยังทำการทดลองกับชุดข้อมูลเดียวกันนี้ด้วย K-means และ Hierarchical clustering ซึ่งจะนำผลการทดลองของทั้ง 3 อัลกอริทึมมาประเมินประสิทธิภาพต่อไป ทั้งนี้การจัดกลุ่มด้วยอัลกอริทึมอื่นในงานวิจัยนี้จะใช้โปรแกรม Cluster 3.0 ซึ่งพัฒนาโดย Michael Eisen และใช้ JavaTreeView ของ Alok Saldanha ในการแสดงผลของการจัดกลุ่มด้วยโปรแกรม Cluster 3.0 ทั้งนี้สามารถดาวน์โหลดและหาข้อมูลเพิ่มเติมได้ที่ <http://bonsai.ims.u-tokyo.ac.jp/~mdehoon/software/cluster/>

ซึ่งในการทดลองจัดกลุ่มข้อมูลทั้งหมดนั้น ถ้าจัดกลุ่มด้วย K-means จะใช้การคำนวณหาค่าความคล้ายคลึงด้วย Euclidian Distance และกำหนดจำนวนกลุ่มเป็นจำนวนคลาส ส่วน Hierarchical Clustering จะให้โปรแกรมสร้าง Complete linkage และจะถือว่าการจัดกลุ่มที่ได้จำนวนกลุ่มเท่ากับจำนวนคลาสเป็นผลลัพธ์ของการจัดกลุ่ม

4.1 เกณฑ์การวัดประสิทธิภาพ

ในงานวิจัยนี้เป็นอัลกอริทึมบนโครงข่ายประสาทเทียมที่เป็นการเรียนรู้แบบไม่มีการชี้แนะ (Unsupervised Learning) แต่ได้ใช้ชุดข้อมูลที่ทราบข้อเท็จจริงของกลุ่มอยู่แล้ว ดังนั้นจึงสามารถประเมินผลด้วยวิธีการแบบมีการชี้แนะ (Supervised) ที่เปรียบเทียบกลุ่มที่กำหนดได้ (cluster label) กับกลุ่มที่แท้จริง (class label) ซึ่งเทคนิคที่นิยมใช้ได้แก่ Entropy, purity, F-measure

ผลของการจัดกลุ่มนั้นเราคาดหวังว่ากลุ่มหรือคลัสเตอร์ที่เกิดขึ้นจะมีมาจากคลาสใดคลาสหนึ่ง โดยทั่วไปมักจะพบว่าคลัสเตอร์ที่ได้มักมีสมาชิกที่มาจากหลายคลาสแต่จะมีคลาสใดคลาสหนึ่งเป็นหลัก ซึ่งค่าประเมินผลของกลุ่มอาจจะดีขึ้นได้เมื่อแตกกลุ่มออกมากขึ้นคืออาจจะให้ค่าของ purity และ entropy ที่ดีขึ้น แต่ในกรณีที่แตกออกมากเกินไปก็ให้ค่าการประเมินประสิทธิภาพที่ต่ำได้เช่นกัน เนื่องจาก F-measure จะประเมินประสิทธิภาพโดยคำนึงถึงข้อมูลคลัสต์ด้วย เมื่อข้อมูลคลัสต์เดียวกันกระจายออกเป็นกลุ่มจำนวนมากก็ทำให้ผลการประเมินประสิทธิภาพต่ำลงได้

4.1.1 Entropy

ค่า Entropy ของแต่ละคลัสเตอร์ i คำนวณได้จาก $e_i = -\sum_{j=1}^L p_{ij} \log_2 p_{ij}$ โดยที่ L คือจำนวนของคลาส และ p_{ij} เป็นการกระจายของข้อมูลซึ่งก็คือค่าความน่าจะเป็นที่ข้อมูลของคลัสเตอร์ i จะเป็นสมาชิกของคลาส j ซึ่งคำนวณได้จาก $p_{ij} = m_{ij} / m_i$ โดยที่ m_i คือจำนวนข้อมูลในคลัสเตอร์ i และ m_{ij} คือจำนวนข้อมูลของคลาส j ในคลัสเตอร์ i และค่ารวม entropy ของการจัดกลุ่มคำนวณได้จาก $e = \sum_{i=1}^K \frac{m_i}{m} e_i$ โดยที่ K คือจำนวนคลัสเตอร์และ m คือจำนวนข้อมูลทั้งหมด

ค่า Entropy จะมีค่าได้ตั้งแต่ 0 จนถึง $\log_2(m)$ โดยค่า Entropy ที่เข้าใกล้ 0 แสดงถึงประสิทธิภาพของการจัดแบ่งกลุ่มที่มีความถูกต้องสูง

4.1.2 Purity

คำนวณ Purity ของคลัสเตอร์ i ได้จาก $p_i = \max_j p_{ij}$ โดย p_{ij} คือการกระจายของข้อมูล และค่ารวม purity ของการจัดกลุ่มคือ $purity = \sum_{i=1}^K \frac{m_i}{m} p_i$

ค่า Purity จะมีค่าได้ตั้งแต่ 0 จนถึง 1 โดยค่า purity ที่เข้าใกล้ 1 จะแสดงถึงประสิทธิภาพของการจัดแบ่งกลุ่มที่มีความถูกต้องสูง

4.1.3 F-measure

คำนวณค่า F-measure ของคลัสเตอร์ i ที่เกี่ยวข้องกับคลาส j ในรูปเมตริกซ์ได้จาก $F(i, j) = (2 \times precision(i, j) \times recall(i, j)) / (precision(i, j) + recall(i, j))$ โดยที่ precision และ recall ของคลัสเตอร์ i ต่อคลาส j คำนวณได้จาก $precision(i, j) = p_{ij}$ และ $recall(i, j) = m_{ij} / m_j$ โดย m_j คือจำนวนข้อมูลในคลาส j

ค่า F-measure จะมีค่าได้ตั้งแต่ 0 จนถึง 1 โดยค่า F-Measure ที่เข้าใกล้ 1 จะแสดงถึงประสิทธิภาพของการจัดแบ่งกลุ่มที่มีความถูกต้องสูง

4.1.4 Euclidean Distance

ในแต่ละคลัสเตอร์ i ซึ่งเมื่อให้ผลการจัดกลุ่มมีจำนวน K คลัสเตอร์ และ x_i, m แทนข้อมูลและค่ากลางของคลัสเตอร์ตามลำดับ Distance ของแต่ละคลัสเตอร์จะคำนวณได้จาก

$$e_K^2 = \sum_{i=1}^{n_K} (x_i^{(K)} - m^{(K)})^2$$

และค่า Euclidean Distance ของการจัดกลุ่มคือผลรวมของ Distance ซึ่ง

$$E_K^2 = \sum_{i=1}^K e_i^2 \text{ ซึ่งในการหาค่ากลางของคลัสเตอร์จะคำนวณจาก } m^{(K)} = \frac{1}{n_K} \sum_{i=1}^{n_K} x_i^{(K)}$$

ค่ากลางของแต่ละกลุ่มจะคำนวณได้จากการหาค่าเฉลี่ยจากเวกเตอร์ของข้อมูลภายในกลุ่ม เช่นเวกเตอร์ในคลัสเตอร์มี 2 เวกเตอร์คือ $[1, 1, 2]$ และ $[0, 1, 4]$ จะคำนวณค่ากลางออกมาได้เป็น $[0.5, 1, 3]$ และหา Distance ของคลัสเตอร์ที่ 1 ได้เป็น $((1-0.5)^2 + (1-1)^2 + (2-3)^2)$ และคลัสเตอร์ที่ 2 เป็น $((0-0.5)^2 + (1-1)^2 + (4-3)^2)$ ซึ่งทำให้ Distance รวมเป็น 2.5

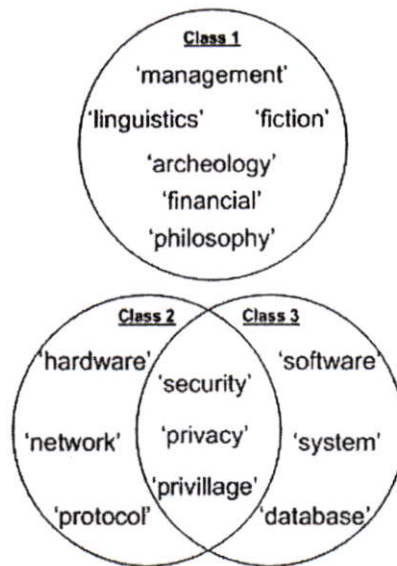
4.2 การทดลองและผลการทดลอง

การทดลองในงานวิจัยใช้ข้อมูลจำนวน 3 ชุด โดยข้อมูลชุดแรกเป็นข้อมูลที่สร้างขึ้นอย่างสุ่มเพื่อใช้ในการทดสอบอัลกอริทึม ส่วนข้อมูลสองชุดสุดท้ายคือข้อมูล Zoo และ Flags ซึ่งเป็นข้อมูลที่นิยมใช้ในการทดสอบอัลกอริทึมในการจัดกลุ่มทั้งการจัดกลุ่มแบบที่ผู้ชี้แนะ และ ไม่มีผู้ชี้แนะ

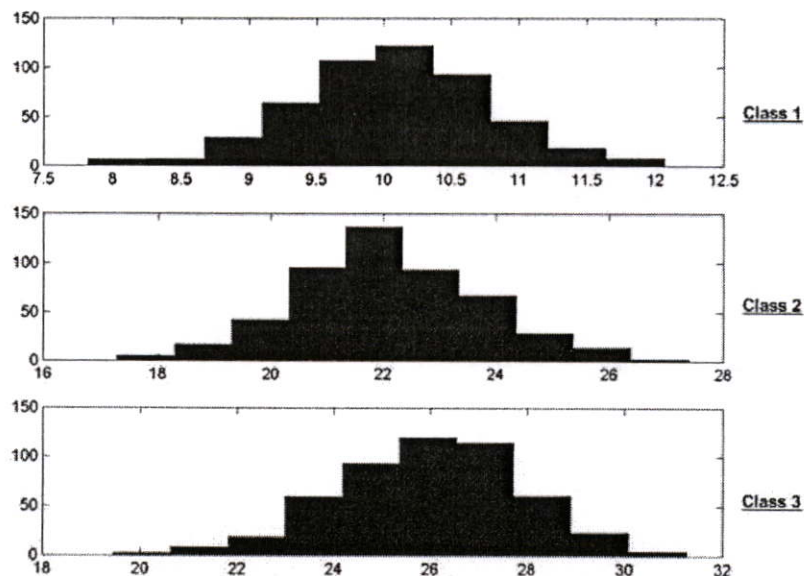
4.2.1 ข้อมูล Generate 1500 ชุด

ข้อมูล Generate เป็นการสร้างข้อมูลขึ้นมาเพื่อทำการทดสอบอัลกอริทึม โดยได้สร้างชุดข้อมูลขึ้นมาจำนวน 1500 ชุด ซึ่งมีฟีเจอร์ทั้งหมด 6 ฟีเจอร์ โดยมีฟีเจอร์ที่เป็นข้อมูลข้อความ 3 ฟีเจอร์ และเป็นข้อมูลเชิงตัวเลขจำนวน 3 ฟีเจอร์ โดยข้อมูลชุดนี้สามารถแบ่งออกได้เป็น 3 คลาส ซึ่งลักษณะการกระจายของข้อมูลข้อมูลแสดงดังรูปที่ 4.1 โดยการสร้างข้อมูลแต่ละฟีเจอร์จะสุ่มข้อมูลจากแต่ละ คลาสขึ้นมา ส่วนข้อมูลเชิงตัวเลขจะมีการกระจายของข้อมูลตามรูปที่ 4.2

การสร้างชุดข้อมูลจะใช้ข้อมูลจากรูปที่ 4.1 และ 4.2 ในการสร้างชุดข้อมูล โดยสำหรับข้อมูลข้อความจะใช้การสุ่มข้อมูลภายในคลาสนั้นมา ซึ่งจากรูปจะเห็นว่าในส่วนของข้อมูลข้อความจะมีการทับซ้อนของข้อมูลคลาสนั้น 2 และ 3 ส่วนข้อมูลเชิงตัวเลขนั้นได้สร้างข้อมูลโดยมีการกระจายตามรูปที่ 4.2 ซึ่งจะเห็นได้ว่าการซ้อนทับของข้อมูลคลาสนั้น 2 และ 3



รูปที่ 4.1 การกระจายข้อมูลของพีเจอร์ที่เป็นข้อมูลข้อความ



รูปที่ 4.2 การกระจายข้อมูลของพีเจอร์ที่เป็นข้อมูลเชิงตัวเลข

ในกรณีที่จัดกลุ่มด้วยยูนิฟายอะแคปทีเพเร โชนั้นเทียบรีจะสามารถนำชุดข้อมูลดังตารางที่ 4.1 เข้าสู่โครงข่ายเพื่อทำการจัดกลุ่มได้เลยโดยไม่ต้องทำการแปลงข้อมูล ส่วนการทดลองด้วย K-means และ Hierarchical จะต้องทำการแปลงข้อมูลข้อความให้เป็นเวกเตอร์จากโดเมนของข้อมูลทั้งหมด เช่นพีเจอร์ที่ 1 แปลงเป็นเวกเตอร์ได้จากโดเมนที่มีค่าเป็นไปได้จำนวน 15 ค่าคือ [management, linguistics, fiction, archeology, financial, philosophy, hardware, network, protocol, security, privacy, privilege, software, system, database] ดังนั้นจากที่ข้อมูลที่ 1 พีเจอร์ที่ 1 มีค่าเป็น

'software' ซึ่งเป็นค่าลำดับที่ 13 ของข้อมูลจึงสามารถแปลงได้เป็นเวกเตอร์ [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0]

จะเห็นได้ว่าเวกเตอร์ที่แทนข้อมูลแต่ละฟีเจอร์จะมีขนาดเท่ากับค่าที่เป็นไปได้ทั้งหมดหรือโดเมนของฟีเจอร์นั้น ซึ่งสำหรับข้อมูล Generate ที่สร้างขึ้นนี้ให้แต่ละฟีเจอร์ข้อความสุ่มจากข้อมูลชุดเดียวกันคือทุกฟีเจอร์ข้อความมีค่าที่เป็นไปได้ 15 ค่า ดังนั้นหลังจากการแปลงข้อมูลจะทำให้ชุดข้อมูลที่เดิมมี 6 ฟีเจอร์ จะกลายเป็นข้อมูลที่มี 48 ฟีเจอร์เพื่อใช้กับการจัดกลุ่มด้วย K-means หรือ Hierarchical

ตารางที่ 4.1 แสดงตัวอย่างข้อมูล 10 ตัวอย่างแรกของข้อมูลชุด Generate 1500 ชุด

Data	feature1	feature2	feature3	feature4	feature5	feature6	group
1	software	security	security	26.348	25.4828	27.6274	3
2	privacy	privacy	system	24.7794	27.9922	24.1643	3
3	network	network	privilege	22.3732	20.7994	19.2551	2
4	privilege	privilege	network	24.8204	22.9701	22.3933	2
5	fiction	financial	management	9.2568	9.2474	11.3702	1
6	network	privilege	network	21.3897	23.042	21.234	2
7	privacy	hardware	security	18.968	21.1032	22.9716	2
8	network	network	network	19.6159	21.0884	24.0617	2
9	security	security	software	27.2077	27.7869	24.8719	3
10	privacy	network	network	20.7813	20.1832	23.3461	2

ในการทดลองจัดกลุ่มด้วยยูนิฟายอะแดปทีฟเรโซแนนทีเรีย กำหนดค่าเริ่มต้นสำหรับการเรียนรู้ของโครงข่ายเป็น

- จำนวนโหนดบนชั้นอินพุทเลเยอร์ = 6
- จำนวนคลัสเตอร์สูงสุด = 3
- เปอร์เซนต์ต่ำสุดของข้อมูลที่ไม่ผ่าน vigilance = 30
- ค่า Vigilance = 0.7
- ค่าคิกรีดั้งเดิมสำหรับข้อมูลเชิงข้อความ = 0.5
- ค่า Learning Rate = 0.2
- ค่า Vigilance decrease rate = 0.02
- จำนวนรอบการเรียนรู้ = 30

ผลการจัดกลุ่มด้วยยูนิฟายอะแดปทีฟเรโซแนนทีเยร์ K-means และ Hierarchical แสดงดังตารางที่ 4.2, 4.3 และ 4.4 ตามลำดับ ซึ่งจะเห็นว่าผลการจัดกลุ่มด้วย K-means ให้ค่าความถูกต้อง ดีที่สุด ดังแสดงในรูปที่ 4.3

ตารางที่ 4.2 แสดงผลการจัดกลุ่มข้อมูล Generate 1500 ชุดด้วยยูนิฟายอะแดปทีฟเรโซแนนทีเยร์

Class	จำนวนสมาชิกที่ตกในแต่ละ Cluster		
	1	2	3
1	0	0	500
2	60	436	4
3	457	43	0
Entropy = 0.34008		F-measure = 0.92885	
Purity = 0.92867		Distance = $1.8259 * 10^4$	

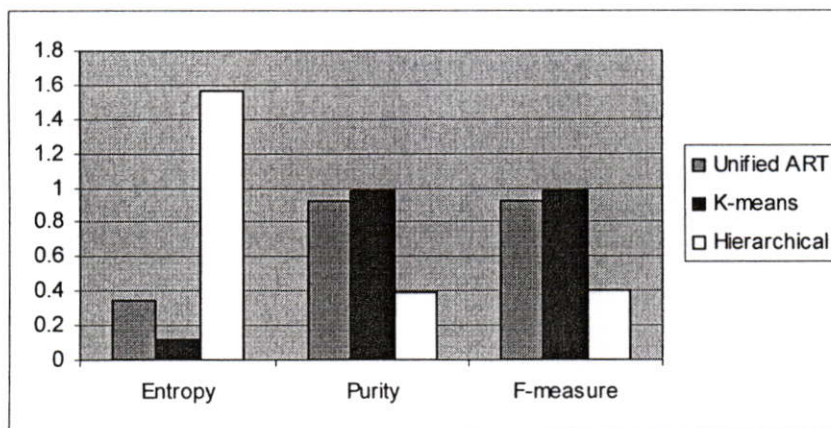
ตารางที่ 4.3 แสดงผลการจัดกลุ่มข้อมูล Generate 1500 ชุดด้วย K-means

Class	จำนวนสมาชิกที่ตกในแต่ละ Cluster		
	1	2	3
1	0	500	0
2	9	0	491
3	481	0	19
Entropy = 0.12123		F-measure = 0.98134	
Purity = 0.98133		Distance = $1.2451 * 10^4$	

ตารางที่ 4.4 แสดงผลการจัดกลุ่มข้อมูล Generate 1500 ชุดด้วย Hierarchical

Class	จำนวนสมาชิกที่ตกในแต่ละ Cluster		
	1	2	3
1	38	318	144
2	84	279	137
3	117	237	146
Entropy = 1.55860		F-measure = 0.40520	
Purity = 0.38733		Distance = $2.1141 * 10^5$	

จากตารางที่ 4.2 แสดงผลการจัดกลุ่มด้วยยูนิฟายอะแคปทีฟเรโซแนนทีเยรี โดยคลัสเตอร์เป็นตัวแทนของข้อมูลคลาสที่ 3, 2 และ 1 ตามลำดับ ตารางที่ 4.3 แสดงผลการจัดกลุ่มด้วย K-means ซึ่งคลัสเตอร์เป็นตัวแทนของคลาสที่ 3, 1 และ 2 ตามลำดับ ตารางที่ 4.4 แสดงผลการจัดกลุ่มด้วย Hierarchical โดยคลัสเตอร์เป็นตัวแทนของข้อมูลคลาสที่ 3, 1 และ 3 ตามลำดับ



รูปที่ 4.3 แสดงผลการประเมินประสิทธิภาพของการจัดกลุ่มข้อมูล Generate 1500 ชุด

4.2.2 ข้อมูล Zoo

ข้อมูลชุด Zoo มีข้อมูลมีจำนวน 101 ชุด ซึ่งมีฟีเจอร์ที่บอกคุณลักษณะของข้อมูลทั้งหมด 16 ฟีเจอร์ โดยมี 15 ฟีเจอร์เป็นข้อมูลบูลีนที่มีค่า 0 หรือ 1 เพื่อบอกว่าสัตว์ชนิดนั้นมีฟีเจอร์นั้นหรือไม่ ส่วนข้อมูลเชิงตัวเลขมี 1 ฟีเจอร์ โดยข้อมูลชุดนี้สามารถแบ่งออกได้เป็น 7 คลาส ซึ่งสามารถดาวน์โหลดข้อมูลได้ที่ <http://www.ailab.si/orange/doc/datasets/zoo.tab> หรือ <ftp://ftp.ics.uci.edu/pub/machine-learning-databases/zoo/> ทั้งนี้สามารถจำแนกประเภทของฟีเจอร์ได้ดังนี้

- ข้อมูลบูลีน ได้แก่ Hair, Feathers, Eggs, Milk, Airbone, Aquatic, Predator, Toothed, Backbone, Breathes, Venomous, Fins, Tail, Domestic, Catsize
- ข้อมูลเชิงตัวเลข ได้แก่ Leg
- ข้อมูลคลาส ได้แก่ Type

จากการพิจารณาข้อมูล Zoo ดังตารางที่ 4.5 และ 4.6 ที่แสดงส่วนหนึ่งของข้อมูล พบว่า ฟีเจอร์ของข้อมูลส่วนหนึ่งเป็นการเก็บเพื่อบอกว่าข้อมูลสัตว์ชนิดนั้นมีฟีเจอร์นั้นหรือไม่ ทำให้ข้อมูลสามารถใช้ทดลองจัดกลุ่มด้วย K-means และ Hierarchical Clustering ได้

ตารางที่ 4.5 แสดงตัวอย่างข้อมูล 10 ตัวอย่างแรกของข้อมูลชุด Zoo ฟีเจอร์ที่1-8

	Name	hair	feathers	Eggs	milk	airbone	aquatic	predator	toothed
1	aardvark	1	0	0	1	0	0	1	1
2	antelope	1	0	0	1	0	0	0	1
3	bass	0	0	1	0	0	1	1	1
4	bear	1	0	0	1	0	0	1	1
5	boar	1	0	0	1	0	0	1	1
6	buffalo	1	0	0	1	0	0	0	1
7	calf	1	0	0	1	0	0	0	1
8	carp	0	0	1	0	0	1	0	1
9	catfish	0	0	1	0	0	1	1	1
10	cavy	1	0	0	1	0	0	0	1

ตารางที่ 4.6 แสดงตัวอย่างข้อมูล 10 ตัวอย่างแรกของข้อมูลชุด Zoo ฟีเจอร์ที่9-16

	Name	backbone	breathes	venomous	fins	tail	domestic	catsize	legs
1	aardvark	1	1	0	0	0	0	1	4
2	antelope	1	1	0	0	1	0	1	4
3	bass	1	0	0	1	1	0	0	0
4	bear	1	1	0	0	0	0	1	4
5	boar	1	1	0	0	1	0	1	4
6	buffalo	1	1	0	0	1	0	1	4
7	calf	1	1	0	0	1	1	1	4
8	carp	1	0	0	1	1	1	0	0
9	catfish	1	0	0	1	1	0	0	0
10	cavy	1	1	0	0	0	1	0	4

ส่วนการทดลองกับงานวิจัยที่นำเสนอ เนื่องจากโครงข่ายสามารถรับข้อมูลข้อความได้ โดยตรงจึงไม่มีความจำเป็นต้องแปลงข้อมูลเป็นตัวเลข โดยการใช้ตัวแปรบูลีน ดังนั้นงานวิจัยได้ทำการปรับข้อมูลบูลีนทั้ง 15 ฟีเจอร์ ให้เป็นข้อมูลข้อความฟีเจอร์เดียว เนื่องจากข้อมูลทั้ง 15 ฟีเจอร์นั้นเป็นการตรวจสอบลักษณะของสัตว์ โดยข้อความของฟีเจอร์ตานั้น เมื่อรวมกันแล้วไม่สามารถเกิด

การเข้าใจผิดอันเนื่องมาจากคำพ้องรูป ตัวอย่างเช่นถ้าข้อมูลเป็น {'hair', 'toothed'} จะหมายถึงมีขน และมีฟัน

1	A	B	C	D	E	F	G	H	I	J
2	Data	Name	hair	feathers	Eggs	milk	airbone	aquatic	predator	toothed
3	1	aardvark	1	0	0	1	0	0	1	1
4	2	antelope	1	0	0	1	0	0	0	1
5	3	bass	0	0	1	0	0	1	1	1
6	4	bear	1	0	0	1	0	0	1	1
7	5	boar	1	0	0	1	0	0	1	1
8	6	buffalo	1	0	0	1	0	0	0	1
9	7	calf	1	0	0	1	0	0	0	1
10	8	carp	0	0	1	0	0	1	0	1
11	9	catfish	0	0	1	0	0	1	1	1
12	10	cavy	1	0	0	1	0	0	0	1

TEXT: 15 features

NUMERIC: 1 feature

1	A	B	C	D
2	Data	Name	feature1	feature2
3	1	aardvark	hair,milk,predator,toothed,backbone,breathes,catsize	4
4	2	antelope	hair,milk,toothed,backbone,breathes,tail,catsize	4
5	3	bass	eggs,aquatic,predator,toothed,backbone,fins,tail	0
6	4	bear	hair,milk,predator,toothed,backbone,breathes,catsize	4
7	5	boar	hair,milk,predator,toothed,backbone,breathes,tail,catsize	4
8	6	buffalo	hair,milk,toothed,backbone,breathes,tail,catsize	4
9	7	calf	hair,milk,toothed,backbone,breathes,tail,domestic,catsize	4
10	8	carp	eggs,aquatic,toothed,backbone,fins,tail,domestic	0

TEXT: 1 feature

NUMERIC: 1 feature

รูปที่ 4.4 แสดงการยุบรวมพีเจอร์

ดังนั้นในการทดลองจึงยุบรวมพีเจอร์ดังแสดงในรูปที่ 4.4 นั่นคือข้อมูลบูลีนทั้ง 15 พีเจอร์ เดิมที่มีค่า 0 หรือ 1 โดยจากข้อมูลบูลีนนั้นให้ 1 หมายถึงมีพีเจอร์นั้น และ 0 จะหมายถึงข้อมูลสัตว์ ตัวนั้นไม่มีพีเจอร์นั้นอยู่ ตัวอย่างเช่นข้อมูลของ aardvark ที่เห็นว่ามีบูลีนเป็น 1 ที่พีเจอร์ hair, mailk, predator, backbone, breathes และ catsize ดังนั้นข้อมูลของ aardvark เมื่อยุบรวมพีเจอร์แล้ว ก็จะได้ พีเจอร์ที่มีเซตของข้อมูลเป็น { hair, mailk, predator, backbone, breathes, catsize }

ในการทดลองจัดกลุ่มด้วยยูนิฟายอะแคปทีฟเรโซแนนทีเยรี กำหนดค่าเริ่มต้นสำหรับการ เรียนรู้ของโครงข่ายเป็น

- จำนวนโหนดบนชั้นอินพุทเลเยอร์ = 2
- จำนวนคลัสเตอร์สูงสุด = 7
- เปอร์เซนต์ต่ำสุดของข้อมูลที่ไม่ว่าน vigilance = 30
- ค่า Vigilance = 0.7
- ค่าคิกรีดั้งเดิมสำหรับข้อมูลเชิงข้อความ = 0.5
- ค่า Learning Rate = 0.2
- ค่า Vigilance decrease rate = 0.02
- จำนวนรอบการเรียนรู้ = 30

ผลการจัดกลุ่มด้วยยูนิฟายอะแคปทีฟเรโซแนนทีเยรี K-means และ Hierarchical แสดงดัง ตารางที่ 4.7, 4.8 และ 4.9 ตามลำดับ ซึ่งจะเห็นว่า การจัดกลุ่มด้วยยูนิฟายอะแคปทีฟเรโซแนนทีเยรี

ให้ค่า F-measure และ Purity ดีที่สุด ในขณะที่การจัดกลุ่มด้วย K-means ให้ค่า Entropy ดีที่สุด ดังแสดงในรูปที่ 4.5

ตารางที่ 4.7 แสดงผลการจัดกลุ่มข้อมูล Zoo ด้วยยูนิฟายอะแคปทีฟเรโซแนนซ์

Class	จำนวนสมาชิกที่ตกในแต่ละ Cluster						
	1	2	3	4	5	6	7
1	37	4	0	0	0	0	0
2	0	0	20	0	0	0	0
3	0	2	1	1	1	0	0
4	0	13	0	0	0	0	0
5	1	0	0	0	3	0	0
6	0	0	0	0	0	8	0
7	0	0	0	4	0	1	5
Entropy = 0.46001				F-measure = 0.8811			
Purity = 0.89109				Distance = 148.9901			

ตารางที่ 4.8 แสดงผลการจัดกลุ่มข้อมูล Zoo ด้วย K-means

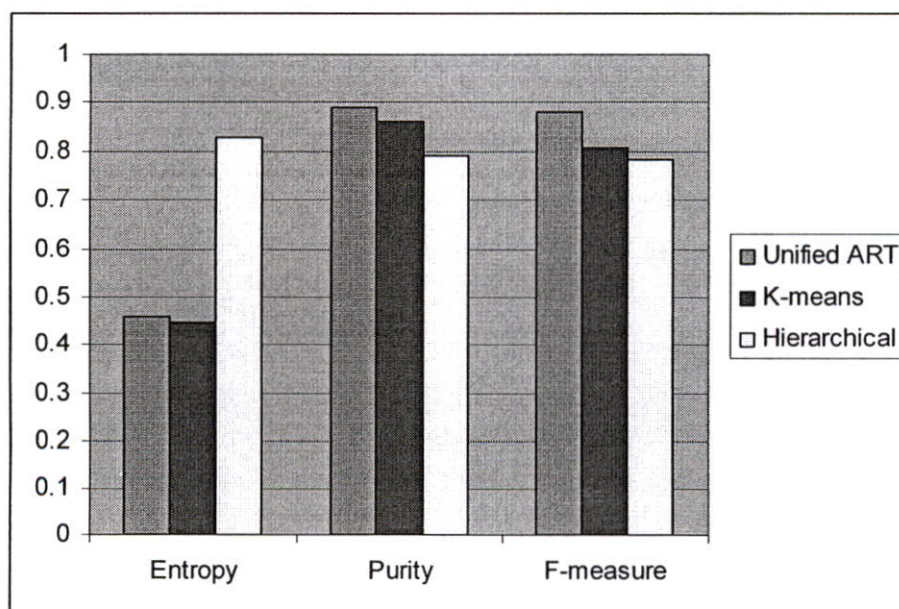
Class	จำนวนสมาชิกที่ตกในแต่ละ Cluster						
	1	2	3	4	5	6	7
1	0	7	0	0	31	0	3
2	0	0	0	20	0	0	0
3	2	0	0	0	0	2	1
4	0	0	0	0	0	0	13
5	0	0	0	0	0	4	0
6	0	0	8	0	0	0	0
7	4	0	4	0	0	2	0
Entropy = 0.44709				F-measure = 0.80597			
Purity = 0.86139				Distance = 181.8216			

ตารางที่ 4.9 แสดงผลการจัดกลุ่มข้อมูล Zoo ด้วย Hierarchical

Class	จำนวนสมาชิกที่ตกในแต่ละ Cluster						
	1	2	3	4	5	6	7
1	0	0	3	37	1	0	0
2	0	0	0	0	20	0	0
3	0	3	0	0	2	0	0
4	13	0	0	0	0	0	0
5	0	0	0	0	4	0	0
6	0	0	0	0	8	0	0
7	0	0	0	0	6	2	2

Entropy = 0.82882	F-measure = 0.78198
Purity = 0.79208	Distance = 485.1269

จากตารางที่ 4.7 แสดงผลการจัดกลุ่มด้วยยูนิฟายอะแคปทีฟเรโซแนนทีเยรี โดยคลัสเตอร์เป็นตัวแทนของข้อมูลคลาสที่ 1, 4, 2, 7, 5, 6 และ 7 ตามลำดับ ตารางที่ 4.8 แสดงผลการจัดกลุ่มด้วย K-means โดยคลัสเตอร์เป็นตัวแทนของข้อมูลคลาสที่ 7, 1, 6, 2, 1, 5 และ 4 ตามลำดับ ตารางที่ 4.9 แสดงผลการจัดกลุ่มด้วย Hierarchical โดยคลัสเตอร์เป็นตัวแทนของข้อมูลคลาสที่ 4, 3, 1, 1, 2, 7 และ 7 ตามลำดับ



รูปที่ 4.5 แสดงผลการประเมินประสิทธิภาพของการจัดกลุ่มข้อมูล Zoo

4.2.3 ข้อมูล Flags

ข้อมูลชุด Flags มีข้อมูลจำนวน 194 ชุด ซึ่งมี ฟีเจอร์ทั้งหมด 28 ฟีเจอร์ โดยมี 18 ฟีเจอร์ที่เป็นข้อมูลข้อความและข้อมูลบูลีน ส่วนข้อมูลเชิงตัวเลขมีจำนวน 10 ฟีเจอร์ โดยข้อมูลชุดนี้สามารถแบ่งออกได้เป็น 8 คลาส ซึ่งสามารถดาวน์โหลดข้อมูลได้ที่ <http://www.ailab.si/orange/doc/datasets/flag.tab> หรือ <ftp://ftp.ics.uci.edu/pub/machine-learning-databases/flags/> ทั้งนี้สามารถจำแนกประเภทของฟีเจอร์ได้ดังนี้

- ข้อมูลข้อความ ได้แก่ landmass, zone, language, mainhue, topleft, botright
- ข้อมูลบูลีน ได้แก่ red, green, blue, gold, white, black, orange, crescent, triangle, icon, animate, text
- ข้อมูลเชิงตัวเลข ได้แก่ area, population, bars, stripes, colours, circles, crosses, saltires, quarters, sunstars
- ข้อมูลคลาส ได้แก่ Region

จากการพิจารณาข้อมูล Flags ดังตารางที่ 4.10-4.13 พบว่าข้อมูลส่วนหนึ่งเป็นข้อมูลบูลีนเพื่อบอกว่าข้อมูลมีฟีเจอร์นั้นหรือไม่เช่นเดียวกับข้อมูล Zoo ทำให้สามารถใช้ทดลองจัดกลุ่มด้วย K-means และ Hierarchical Clustering ได้

ตารางที่ 4.10 แสดงตัวอย่างข้อมูล 10 ตัวอย่างแรกของข้อมูลชุด Flags ฟีเจอร์ที่ 1-6

	Name	landmass	zone	area	population	language	bars
1	Afghanistan	5	1	648	16	10	0
2	Albania	3	1	29	3	6	0
3	Algeria	4	1	2388	20	8	2
4	American-Samoa	6	3	0	0	1	0
5	Andorra	3	1	0	0	6	3
6	Angola	4	2	1247	7	10	0
7	Anguilla	1	4	0	0	1	0
8	Antigua-Barbuda	1	4	0	0	1	0
9	Argentina	2	3	2777	28	2	0
10	Argentine	2	3	2777	28	2	0

ตารางที่ 4.11 แสดงตัวอย่างข้อมูล 10 ตัวอย่างแรกของข้อมูลชุด Flags พีเจอร์รี่ที่ 7-14

	Name	stripes	colours	red	green	blue	gold	white	black
1	Afghanistan	3	5	1	1	0	1	1	1
2	Albania	0	3	1	0	0	1	0	1
3	Algeria	0	3	1	1	0	0	1	0
4	American-Samoa	0	5	1	0	1	1	1	0
5	Andorra	0	3	1	0	1	1	0	0
6	Angola	2	3	1	0	0	1	0	1
7	Anguilla	1	3	0	0	1	0	1	0
8	Antigua-Barbuda	1	5	1	0	1	1	1	1
9	Argentina	3	2	0	0	1	0	1	0
10	Argentine	3	3	0	0	1	1	1	0

ตารางที่ 4.12 แสดงตัวอย่างข้อมูล 10 ตัวอย่างแรกของข้อมูลชุด Flags พีเจอร์รี่ที่ 15-21

	Name	orange	mainhue	circles	crosses	saltires	quarters	sunstars
1	Afghanistan	0	green	0	0	0	0	1
2	Albania	0	red	0	0	0	0	1
3	Algeria	0	green	0	0	0	0	1
4	American-Samoa	1	blue	0	0	0	0	0
5	Andorra	0	gold	0	0	0	0	0
6	Angola	0	red	0	0	0	0	1
7	Anguilla	1	white	0	0	0	0	0
8	Antigua-Barbuda	0	red	0	0	0	0	1
9	Argentina	0	blue	0	0	0	0	0
10	Argentine	0	blue	0	0	0	0	1

ตารางที่ 4.13 แสดงตัวอย่างข้อมูล 10 ตัวอย่างแรกของข้อมูลชุด Flags พีเจอร์ที่ 22-29

	Name	crescent	triangle	icon	animate	text	toleft	botright	religion
1	Afghanistan	0	0	1	0	0	black	green	2
2	Albania	0	0	0	1	0	red	red	6
3	Algeria	1	0	0	0	0	green	white	2
4	American-Samoa	0	1	1	1	0	blue	red	1
5	Andorra	0	0	0	0	0	blue	red	0
6	Angola	0	0	1	0	0	red	black	5
7	Anguilla	0	0	0	1	0	white	blue	1
8	Antigua-Barbuda	0	1	0	0	0	black	red	1
9	Argentina	0	0	0	0	0	blue	blue	0
10	Argentine	0	0	0	0	0	blue	blue	0

ส่วนการทดลองกับงานวิจัยที่นำเสนอ เนื่องจากโครงข่ายสามารถรับข้อมูลข้อความได้ โดยตรงจึงไม่มีความจำเป็นต้องแปลงข้อมูลเป็นตัวเลข โดยการใช้ตัวแปรบูลีน ดังนั้นงานวิจัยได้ทำการปรับข้อมูลบูลีนออกเป็นข้อมูลข้อความ 2 พีเจอร์ที่ โดยพีเจอร์ที่แรกคือกลุ่มของข้อมูลสี ซึ่งถ้าข้อมูลเป็น { red, green, blue, gold, white, black, orange } จะหมายถึงมีทุกสี และข้อมูลอีกกลุ่มคือข้อมูลสัญลักษณ์ ซึ่งถ้าข้อมูลเป็น { crescent, triangle, icon, animate, text } จะหมายถึงมีสัญลักษณ์เหล่านี้ทั้งหมด

ในการทดลองจัดกลุ่มด้วยยูนิฟายอะแคปทีฟเรโซแนนทีเยรี กำหนดค่าเริ่มต้นสำหรับการเรียนรู้ของโครงข่ายเป็น

- จำนวนโหนดบนชั้นอินพุทเลเยอร์ = 18
- จำนวนคลัสเตอร์สูงสุด = 8
- เปรอเซ็นต์ต่ำสุดของข้อมูลที่ไม่ว่าน vigilance = 30
- ค่า Vigilance = 0.7
- ค่าคิรีตั้งต้นสำหรับข้อมูลเชิงข้อความ = 0.5
- ค่า Learning Rate = 0.2
- ค่า Vigilance decrease rate = 0.02
- จำนวนรอบการเรียนรู้ = 30

ผลการจัดกลุ่มด้วยยูนิฟายอะแคปทีฟเรโซแนนซ์ K-means และ Hierarchical แสดงดังตารางที่ 4.14, 4.15 และ 4.16 ตามลำดับ ซึ่งจะเห็นว่าการจัดกลุ่มด้วยยูนิฟายอะแคปทีฟเรโซแนนซ์ให้ค่า Purity และ Entropy ดีที่สุด ในขณะที่การจัดกลุ่มด้วย K-means ให้ค่า F-measure ดีที่สุด ดังแสดงในรูปที่ 4.6

จากตารางที่ 4.14 แสดงผลการจัดกลุ่มด้วยยูนิฟายอะแคปทีฟเรโซแนนซ์ โดยคลัสเตอร์เป็นตัวแทนของข้อมูลคลาสที่ 8, 2, 8, 8, 1, 2, 1 และ 5 ตามลำดับ ตารางที่ 4.15 แสดงผลการจัดกลุ่มด้วย K-means ซึ่งมีคลัสเตอร์ที่ไม่สามารถบอกได้ว่าเป็นตัวแทนของคลาสใดคือคลัสเตอร์ที่ 2, 7 และ 8 โดยคลัสเตอร์ที่ 1, 3, 4, 5 และ 6 เป็นตัวแทนของข้อมูลคลาสที่ 6, 2, 2, 2 และ 1 ตามลำดับ ตารางที่ 4.16 แสดงผลการจัดกลุ่มด้วย Hierarchical โดยคลัสเตอร์เป็นตัวแทนของข้อมูลคลาสที่ 1, 8, 1, 1, 1, 8, 1 และ 1 ตามลำดับ

ตารางที่ 4.14 แสดงผลการจัดกลุ่มข้อมูล Flags ด้วยยูนิฟายอะแคปทีฟเรโซแนนซ์

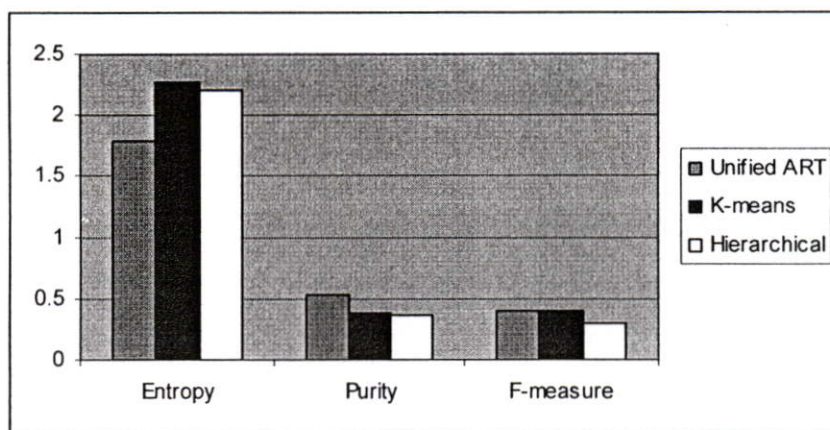
Class	จำนวนสมาชิกที่ตกในแต่ละ Cluster							
	1	2	3	4	5	6	7	8
1	11	1	8	6	16	1	11	6
2	0	12	5	2	2	14	0	1
3	0	0	1	1	3	3	0	0
4	1	0	0	0	0	1	1	1
5	0	6	5	0	0	1	0	15
6	1	1	4	4	4	1	0	0
7	1	1	0	0	0	2	0	0
8	12	2	11	11	1	0	2	1
Entropy = 1.7909		F-measure = 0.39195						
Purity = 0.52577		Distance = $8.8967 * 10^8$						

ตารางที่ 4.15 แสดงผลการจัดกลุ่มข้อมูล Flags ด้วย K-means

Class	จำนวนสมาชิกที่ตกในแต่ละ Cluster							
	1	2	3	4	5	6	7	8
1	3	0	3	5	1	46	2	0
2	0	4	4	8	4	11	3	2
3	0	0	2	1	0	5	0	0
4	0	0	0	1	0	2	0	1
5	0	4	4	7	0	10	2	0
6	2	0	0	5	1	7	0	0
7	0	0	0	1	0	3	0	0
8	1	2	3	4	1	24	3	2
Entropy = 2.2715		F-measure = 0.39554						
Purity = 0.38144		Distance = $1.5505 * 10^8$						

ตารางที่ 4.16 แสดงผลการจัดกลุ่มข้อมูล Flags ด้วย Hierarchical

Class	จำนวนสมาชิกที่ตกในแต่ละ Cluster							
	1	2	3	4	5	6	7	8
1	7	22	4	3	6	2	9	7
2	5	28	0	0	0	3	0	0
3	0	6	0	0	1	1	0	0
4	0	3	0	0	0	0	1	0
5	4	23	0	0	0	0	0	0
6	1	14	0	0	0	0	0	0
7	1	3	0	0	0	0	0	0
8	1	29	0	0	2	6	2	0
Entropy = 2.1946		F-measure = 0.29499						
Purity = 0.36598		Distance = $8.6290 * 10^8$						



รูปที่ 4.6 แสดงผลการประเมินประสิทธิภาพของการจัดกลุ่มข้อมูล Flags

4.3 สรุปผลการทดลอง

จากการทดลองจัดกลุ่มข้อมูลในหัวข้อที่ 4.2 การจัดกลุ่มข้อมูล Generate 1500 ชุดด้วยยูนิฟายอะแคปทีเพเรโซแนนทีเยรี ให้ผลการประเมินประสิทธิภาพด้วยค่า Entropy เท่ากับ 0.34008 ค่า Purity เท่ากับ 0.92867 และค่า F-measure เท่ากับ 0.92885 ซึ่งแสดงถึงมีความถูกต้อง 92 เปอร์เซ็นต์ ผลการจัดกลุ่มยังแสดงให้เห็นว่าแต่ละคลัสเตอร์จะเป็นตัวแทนของคลาสใดคลาสหนึ่ง เมื่อเปรียบเทียบค่าความถูกต้องของการจัดกลุ่มด้วย K-means และ Hierarchical พบว่า ให้ความถูกต้องเป็น 98 เปอร์เซ็นต์ และ 40 เปอร์เซ็นต์ตามลำดับ ซึ่ง K-means ให้ผลการจัดกลุ่มที่เป็นตัวแทนของคลาสใดคลาสหนึ่ง ในขณะที่ Hierarchical มีบางคลัสเตอร์ที่ไม่สามารถระบุคลาสได้

เนื่องจากข้อมูลชุด Generate 1500 ชุดเป็นชุดที่สร้างขึ้นโดยทราบลักษณะการซ้อนทับของข้อมูล คือมีการซ้อนทับของข้อมูลข้อความในคลาสที่ 2 และ 3 และซ้อนทับของข้อมูลเชิงตัวเลขคลาสที่ 2 และ 3 เมื่อพิจารณาผลการจัดกลุ่มของยูนิฟายอะแคปทีเพเรโซแนนทีเยรีที่ 4.2 พบว่าข้อมูลคลาสที่ 2 และ 3 ถูกกระจายไปคลัสเตอร์ต่างๆ แต่ก็ยังมีคลัสเตอร์หลักเพียงคลัสเตอร์เดียว เมื่อพิจารณาดารางที่ 4.3 ซึ่งเป็นผลการจัดข้อมูลด้วย K-means พบว่าสามารถแยกข้อมูลคลาสที่ 2 และ 3 ซึ่งมีการซ้อนทับข้อมูลได้ดี และเมื่อพิจารณาผลการจัดข้อมูลด้วย Hierarchical ดังดารางที่ 4.4 พบว่าไม่สามารถแยกคลาสที่ 2 และ 3 ออกจากกันได้ดังจะเห็นได้จากข้อมูลคลาสที่ 2 ส่วนใหญ่จะถูกจัดให้อยู่ในคลัสเตอร์ที่ 2 เช่นเดียวกับกับข้อมูลส่วนใหญ่ของคลาสที่ 3 ก็ถูกจัดให้อยู่ในคลัสเตอร์ที่ 2

การจัดกลุ่มกับข้อมูล Zoo ด้วยยูนิฟายอะแคปทีเพเรโซแนนทีเยรี ให้ผลการประเมินประสิทธิภาพด้วยค่า Entropy เท่ากับ 0.46001 ค่า Purity เท่ากับ 0.89109 และค่า F-measure เท่ากับ 0.88110 ซึ่งแสดงถึงความถูกต้องในการจัดกลุ่มถึง 88 เปอร์เซ็นต์ ผลการจัดกลุ่มยังแสดงให้เห็นว่าแต่ละคลัสเตอร์จะเป็นตัวแทนของคลาสใดคลาสหนึ่ง เมื่อเปรียบเทียบค่าความถูกต้องของการจัด

กลุ่มด้วย K-means และ Hierarchical พบว่า ให้ความถูกต้องเป็น 80 เปอร์เซ็นต์ และ 78 เปอร์เซ็นต์ ตามลำดับ และผลการจัดกลุ่มจะได้คลัสเตอร์ที่เป็นตัวแทนของคลาสใดคลาสหนึ่ง

การจัดกลุ่มข้อมูล Flags ด้วยยูนิฟายอะแคปทีเพรโซแน้นเทียรี ให้ผลการประเมินประสิทธิภาพด้วยค่า Entropy เท่ากับ 1.79090 ค่า Purity เท่ากับ 0.52577 และค่า F-measure เท่ากับ 0.39195 ซึ่งแสดงถึงความถูกต้อง 39 เปอร์เซ็นต์ โดยคลัสเตอร์ที่ 1 ให้ผลการจัดกลุ่มที่ไม่ดีนัก เนื่องจากถึงแม้คลัสเตอร์จะเป็นตัวแทนคลาสที่ 8 แต่ก็มีสมาชิกของคลาสที่ 8 ไม่ถึงครึ่งหนึ่งของสมาชิกทั้งหมด แต่ถึงอย่างไรผลการจัดกลุ่มก็ยังคงแสดงให้เห็นว่าแต่ละคลัสเตอร์จะเป็นตัวแทนของคลาสใดคลาสหนึ่ง เมื่อเปรียบเทียบค่าความถูกต้องของการจัดกลุ่มด้วย K-means และ Hierarchical พบว่า ให้ความถูกต้องเป็น 39 เปอร์เซ็นต์ และ 29 เปอร์เซ็นต์ตามลำดับ ซึ่ง Hierarchical ให้ผลการจัดกลุ่มที่เป็นตัวแทนของคลาสใดคลาสหนึ่ง ในขณะที่ K-means มีบางคลัสเตอร์ที่ไม่สามารถระบุคลาสได้

ตารางที่ 4.17 แสดงผลการประเมินประสิทธิภาพของอัลกอริทึมต่างๆ

Data	Algorithm	Entropy	Purity	F-measure	Distance
1.Generate 1500	Unified ART	0.34008	0.92867	0.92885	$1.8259 * 10^4$
	K-means	0.12123	0.98133	0.98134	$1.2451 * 10^4$
	Hierarchical	1.55860	0.38733	0.40520	$2.1141 * 10^5$
2.Zoo	Unified ART	0.46001	0.89109	0.88110	148.9901
	K-means	0.44709	0.86139	0.80597	181.8216
	Hierarchical	0.82882	0.79208	0.78198	485.1269
3.Flags	Unified ART	1.79090	0.52577	0.39195	$8.8967 * 10^8$
	K-means	2.27150	0.38144	0.39554	$1.5505 * 10^8$
	Hierarchical	2.19460	0.36598	0.29499	$8.6290 * 10^8$

การประเมินการจัดกลุ่มของ Generate 1500 กับ Zoo ด้วย Distance ให้ผลที่สอดคล้องกับ F-measure คืออัลกอริทึมที่ให้ผลการประเมินด้วย F-measure ดีก็จะมี Distance น้อย ส่วนผลการจัดกลุ่มของ Flags การจัดกลุ่มด้วย Hierarchical ให้ค่าการประเมินด้วย Distance ดีกว่า Unified ART

ทั้งนี้จากการทดลองกับข้อมูลทั้ง 3 ชุดคงแสดงผลการประเมินประสิทธิภาพด้วยวิธีต่างๆ ดังรูปที่ 4.3, 4.4 และ 4.5 นั้นสามารถสรุปผลการจัดกลุ่มได้ด้วยวิธีการต่างๆ ได้ดังตารางที่ 4.10 ซึ่งแสดงให้เห็นว่าผลการจัดกลุ่มด้วยยูนิฟายอะแคปทีเพรโซแน้นเทียรีมีความถูกต้องใกล้เคียงกับการจัดกลุ่มด้วย K-means ในข้อมูลทั้ง 3 ชุดที่นำมาทดลองในงานวิจัยนี้

ตารางที่ 4.18 แสดงจำนวนฟีเจอร์หลังการแปลงข้อมูลสำหรับการทดลองแต่ละอัลกอริทึม

	จำนวนฟีเจอร์หลังการแปลงข้อมูล		
	Unified ART	K-means	Hierarchical
1.Generate 1500	6	48	48
2.Zoo	2	16	16
3.Flags	18	28	28

ตารางที่ 4.19 แสดงค่า Gain Ratio ของฟีเจอร์ต่างๆในชุดข้อมูล Zoo

ฟีเจอร์	Gain Ratio	Gini
hair	0.30	0.08
feathers	0.46	0.14
eggs	0.37	0.11
milk	0.39	0.12
airbone	0.29	0.08
aquatic	0.18	0.03
predator	0.09	0.01
toothed	0.49	0.11
backbone	0.39	0.08
breathes	0.33	0.07
venomous	0.16	0.02
fins	0.28	0.07
tail	0.30	0.06
domestic	0.02	0.00
catsize	0.17	0.04
legs	0.25	0.08

โดยในการทดลองด้วยอัลกอริทึมต่างๆจะมีกระบวนการแปลงข้อมูลเพื่อให้สามารถใช้งานได้ด้วยอัลกอริทึมอื่นๆ ซึ่งทำให้ชุดข้อมูลสำหรับการทดลองในแต่ละอัลกอริทึมจะมีจำนวนฟีเจอร์ที่ไม่เท่ากัน ทั้งนี้การจัดกลุ่มข้อมูลข้อความที่ K-means และ Hierarchical จะต้องแปลงเป็นเวกเตอร์ก่อนนั้น มักทำให้แต่ละฟีเจอร์ต้องสร้างเวกเตอร์ขนาดใหญ่เพื่อรองรับข้อมูล ในขณะที่ยูนิฟายอะแคปทีฟเรโซแนนซ์ทฤษฎีจะสามารถใช้งานได้ทันทีโดยไม่ต้องแปลงเป็นเวกเตอร์ ซึ่งจากตารางที่ 4.18

แสดงให้เห็นว่ายูนิฟายอะแดปทีฟเรโซแนนทีวี่สามารถนำมาประยุกต์ใช้ในการจัดกลุ่มข้อมูลได้ โดยใช้จำนวนพีเจอร์น้อยกว่าวิธีการอื่น นั่นคือสามารถลดความซับซ้อนของ K-means และ Hierarchical ที่มักจะต้องแปลงข้อมูลจนทำให้ข้อมูลมีความซับซ้อนมากขึ้น

ทั้งนี้จากผลการทดลองกับข้อมูล Zoo ที่ในงานวิจัยให้ผลการจัดกลุ่มที่ดีกว่า K-means นั้น ในการทดลองมีการยุบรวมพีเจอร์จากบวกลบ 15 พีเจอร์เหลือเพียงข้อมูลข้อความพีเจอร์เดียว ซึ่งเหมือนกับเป็นการลดความสำคัญของพีเจอร์นั้นๆลง แต่ยังคงพีเจอร์ของ legs ซึ่งเป็นตัวเลขแสดงจำนวนขาไว้ ดังนั้นเพื่อวัดความสำคัญของพีเจอร์ว่าข้อมูล legs มีความสำคัญต่อการจัดกลุ่มมากจนทำให้ผลการจัดกลุ่มออกมาถูกต้องหรือไม่ จึงได้ทำการวัดระดับความสำคัญของพีเจอร์ด้วยค่า Gain Ratio ให้ผลดังตารางที่ 4.19 ซึ่งแสดงให้เห็นว่าพีเจอร์ที่มีค่า Gain Ratio สูงสุดคือ toothed นั่นคือพีเจอร์ legs ไม่ได้เป็นพีเจอร์ที่สำคัญสูงสุดในการจัดกลุ่ม

บทที่ 5

สรุปผลการวิจัยและข้อเสนอแนะ

5.1 สรุปผลงานวิจัย

งานวิจัยนี้ได้นำเสนอวิธีการจัดกลุ่มโดยใช้หลักการของโครงข่ายประสาทเทียมแบบพีชชีอะแคปทีเพรโซแน้นเทียร์ มาประยุกต์กับการหาความคล้ายคลึงของข้อความด้วย Similarity Measure ดังที่ได้นำเสนอมาแล้ว ซึ่งทำให้โครงข่ายมีความสามารถในการจัดกลุ่มข้อมูลที่มีพีเจอร์เป็นข้อมูลข้อความและข้อมูลเชิงตัวเลขได้โดยสามารถรับข้อมูลอินพุตที่เป็นข้อความได้โดยตรง โดยที่ไม่จำเป็นต้องแปลงอินพุตให้เป็นข้อมูลเชิงตัวเลข ทั้งนี้พีเจอร์ที่รับข้อมูลข้อความนั้นมีลักษณะเป็นเซตนั้นคือสามารถรับกลุ่มของคำ งานวิจัยมีอัลกอริทึมหลักๆที่ปรับปรุงจากงานวิจัยอื่นๆดังนี้

— ค่าถ่วงน้ำหนัก

ค่าถ่วงน้ำหนักของโครงข่ายจะมีเพียงชุดเดียว ซึ่งแตกต่างจากโครงข่ายอะแคปทีเพรโซแน้นเทียร์เดิมและงานวิจัยที่มีการประยุกต์ใช้โครงข่ายดังกล่าวในการจัดกลุ่มข้อมูลซึ่งจะมีค่าถ่วงน้ำหนัก 2 ชุด ทั้งนี้พบว่าการใช้ค่าถ่วงน้ำหนักเพียงชุดเดียวนั้น โครงข่ายยังคงมีความสามารถในการจัดกลุ่มอยู่เช่นเดิม โดยสมาชิกของค่าถ่วงน้ำหนักของแต่ละเอาท์พุท โหนดสามารถใช้เป็นตัวแทนหรือเป็นเทมเพลตของแต่ละคลัสเตอร์ ซึ่งค่าถ่วงน้ำหนักของโครงข่ายในส่วนของข้อมูลข้อความจะมีการเก็บค่าระดับความสัมพันธ์ทำให้สามารถบอกระดับความสัมพันธ์ของแต่ละคำของข้อมูลข้อความได้

— ค่า vigilance

ระดับความคล้ายคลึงในการจัดกลุ่มของโครงข่ายประสาทเทียมจะใช้ค่า vigilance เป็นตัวแปรที่กำหนดค่า Threshold ซึ่งงานวิจัยนี้ได้ปรับอัลกอริทึมของโครงข่ายให้สามารถปรับลดค่า vigilance ตามความเหมาะสมจากจำนวนข้อมูลที่ไม่ผ่านการทดสอบ vigilance และเนื่องจากผลรวมค่าความคล้ายคลึงของการเปรียบเทียบข้อมูลอินพุตกับค่าถ่วงน้ำหนักในแต่ละครั้งจะมีค่าไม่คงที่ และมีโอกาสมากกว่า 1 ดังนั้นในการกำหนดค่า vigilance จึงใช้การเสกค่าจากค่าสูงสุดที่เป็นไปได้ของผลรวมค่าความคล้ายคลึง

— จำนวนโหนดของโครงข่าย

จำนวนโหนดของโครงข่ายในชั้นอินพุตจะขึ้นอยู่กับการพิจารณาลักษณะของพีเจอร์ โดยที่การรวมพีเจอร์ประเภทข้อมูลข้อความอาจจะเหมาะสมในบางกรณี เช่นการเก็บข้อมูลในลักษณะว่ามีหรือไม่มีพีเจอร์นั้นๆ ซึ่งจำนวนโหนดที่น้อยลงมีผลต่อการเรียนรู้ของโครงข่ายที่เร็วขึ้น จำนวน

โหนดบนชั้นอินเทอร์เฟซจะเท่ากับจำนวนโหนดบนชั้นอินพุต ส่วนจำนวนโหนดบนชั้นเอาต์พุตจะขึ้นอยู่กับความรู้ของโครงข่าย โดยเริ่มต้นจะไม่มีโหนดบนชั้นเอาต์พุตและจำนวนเอาต์พุตจะเพิ่มขึ้นเรื่อยๆเมื่อข้อมูลอินพุตไม่ผ่านการทดสอบ vigilance แต่จำนวนเอาต์พุตจะเพิ่มไปได้จนถึงจำนวนโหนดสูงสุดเท่านั้นเพื่อไม่ให้เกิดปัญหาจำนวนกลุ่มที่มากเกินไปและปัญหาการย้ายกลุ่มไปเรื่อยๆของข้อมูลอินพุต

งานวิจัยได้นำเสนอการทดลองเพื่อประเมินประสิทธิภาพของอัลกอริทึมที่นำเสนอและได้ทำการเปรียบเทียบกับประสิทธิภาพของอัลกอริทึมอื่น โดยผลการจัดกลุ่มที่ได้นั้นถือว่าให้ความถูกต้องใกล้เคียงกับ K-means ซึ่งเป็นอัลกอริทึมแบบ Partitional Clustering ที่ได้รับความนิยม และผลการจัดกลุ่มก็ยังคงแสดงให้เห็นว่าแต่ละคลัสเตอร์จะเป็นตัวแทนของคลาสใดคลาสหนึ่ง

5.2 ข้อเสนอแนะและแนวทางในการทำวิจัยต่อ

อัลกอริทึมที่นำเสนอสามารถนำไปประยุกต์เพื่อพัฒนาเป็นเครื่องมือในการจัดกลุ่มข้อมูลที่มีอยู่จริงซึ่งมักจะมีลักษณะข้อมูลที่หลากหลาย โดยงานวิจัยนี้เริ่มต้นจากการศึกษา Symbolic Object ซึ่งจะประกอบไปด้วยข้อมูลข้อความ ข้อมูลเชิงตัวเลข และข้อมูลแบบเป็นช่วง งานวิจัยนี้สามารถรับข้อมูลข้อความและข้อมูลเชิงตัวเลขเท่านั้น ซึ่งเป็นเพียงการเริ่มพัฒนาไปสู่การจัดกลุ่มข้อมูลที่เป็น Symbolic Object

ทั้งนี้อัลกอริทึมที่นำเสนอมีความสามารถในการจัดกลุ่มข้อมูลที่มีการซ้อนทับ (Overlap) ซึ่งให้ผลการจัดกลุ่มที่มีความถูกต้องค่อนข้างสูงจากการทดลองกับข้อมูลตัวอย่างโดยในขั้นตอนการเรียนรู้ช่วงแรกโครงข่ายจะมีการเพิ่มโหนดบนชั้นเอาต์พุตและปรับค่าถ่วงน้ำหนักอยู่ตลอดเวลา นั่นคือโครงข่ายมีความเป็น Plasticity และในการเรียนรู้แต่ละรอบนั้นค่าถ่วงน้ำหนักของโครงข่ายมีแนวโน้มคงที่ ซึ่งก็มีผลให้สมาชิกของคลัสเตอร์มีแนวโน้มคงที่ด้วย นั่นคือโครงข่ายมีความ Stability

สุดท้ายนี้แนวทางในการพัฒนางานวิจัยต่อการปรับโครงข่ายให้รับข้อมูลที่เป็น Symbolic Object เนื่องจากเป็นลักษณะข้อมูลที่มีอยู่จริงที่พบได้ทั่วไป

เอกสารอ้างอิง

- [1] Cabena P. et al. **Discovering data mining from concept to implementation.** Prentice Hall PTR. 1997.
- [2] Lanrene F. **Fundamentals of Neural Networks Architecture, Algorithms and Application.** New Jersey : Prentice Hall International. 1994.
- [3] Gowda K.C. and Diday E. "Symbolic Clustering Using a New Similarity Measure." IEEE Trans. On Syst., Man, Cybern., vol. 22, no. 2, March/April 1992. pp. 368-378.
- [4] El-SonBaty YA and Ismail M. A. "Fuzzy Clustering for Symbolic Data." IEEE Trans. On Fuzzy Syst., vol. 6, no. 2, May 1998. pp. 195-204.
- [5] Carpenter G. A., Grossberg S. and Rosen D. A. "Fuzzy ART: Fast Stable Learning and Categorization of Analog Patterns by an Adaptive Resonance System." Pergamon Press plc., Neural Network, Vol. 4, 1991, pp. 759-771.
- [6] Freeman A.J. and Skapura M.D. **Neural Networks: algorithms, applications, and - programming techniques.** Addison-Wesley. 1992.
- [7] นายสุทธิชัย มณีรัตน์รุ่งโรจน์ 2539 "การจำลองการทำงานโครงข่ายประสาทเทียม-Artificial neural networks simulation" วิทยานิพนธ์วิทยาศาสตรมหาบัณฑิต สาขาวิชาวิทยาการคอมพิวเตอร์และเทคโนโลยีสารสนเทศ บัณฑิตวิทยาลัย, สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง.
- [8] [Online]. Available: <http://en.wikipedia.org/wiki/Neuron>
- [9] [Online]. Available: <http://faculty.washington.edu/chudler/cells.html>
- [10] Dave A. and George M. 1992. **Artificial Neural Networks Technology.** [Online]. Available: <http://www.dacs.dtic.mil/techs/neural/neural.title.html>
- [11] [Online]. Available: <http://medusa.sdsu.edu/Robotics/Neuromuscular/Theses/Hongyu/-chapter3.pdf>
- [12] M. Narasimha Murty et al. Jan 2004. **Clustering Large Symbolic Datasets.** [Online]. Available: http://www.ceremade.dauphine.fr/~touati/workshop/PDF/-18_Clustering_Large_Symb_Datasets_Narashima.pdf
- [13] นายทรงพล ชุตินวงศ์พัฒนกุล 2546 "เท็กโปรเซสซิงโคโฮเนนนิวรอลเน็ตเวิร์คโดยใช้กระบวนการเรียนรู้แนวใหม่" วิทยานิพนธ์วิทยาศาสตรมหาบัณฑิต สาขาวิชาวิทยาการคอมพิวเตอร์

และเทคโนโลยีสารสนเทศ บัณฑิตวิทยาลัย, สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง.

- [14] นายสมคิด แสนเสนาะ 2547 “การแบ่งกลุ่มเอกสาร โดยใช้เทคนิคการประมวลผลข้อความด้วยโครงข่ายประสาทเทียมที่เรียนรู้แบบหาผู้ชนะ” วิทยานิพนธ์วิทยาศาสตรมหาบัณฑิต สาขาวิชาวิทยาการคอมพิวเตอร์และเทคโนโลยีสารสนเทศ บัณฑิตวิทยาลัย, สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง.
- [15] นายนรเศรษฐ์ จันทสูตร 2547 “เท็กอะแดปทีฟเรโซแนนซ์เรียร์นิวรอลเน็ตเวิร์ค” วิทยานิพนธ์วิทยาศาสตรมหาบัณฑิต สาขาวิชาวิทยาการคอมพิวเตอร์และเทคโนโลยีสารสนเทศ บัณฑิตวิทยาลัย, สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง.
- [16] Murphy, P.M. & Aha, D.W. (1991). UCI Repository of Machine Learning Databases. Dept. Machine-readable depository, Information & Computer Science, University of California at Irvine, CA. [Online]. Available: <http://www.ics.uci.edu/~mlearn/MLRepository.html>.
- [17] Pang-Ning T., Michael S. and Vipin K. **Introduction to Data mining**. Addison-Wesley. 2005. [Online]. Available: <http://www-users.cs.umn.edu/~kumar/dmbook/index.php>
- [18] A.K. Jain, M.N. Murty and P.J. Flynn “Data Clustering: A Review.” ACM Computing Surveys, vol. 31, no. 3, September 1999. pp. 264-323.
- [19] A.M. Bagirov et al. “Unsupervised and supervised data classification via nonsmooth and global optimization” School of Information Technology and Mathematical Sciences, The University of Ballarat.
- [20] T.V. Ravi and K.C. Gowda. “Clustering of Symbolic Objects Using Gravitational Approach” IEEE Trans. On Syst., Man, Cybern., vol. 29, no. 6, December 1999. pp. 888-894.
- [21] Dinesh. M.S. et al. “Classification of Symbolic data using Fuzzy set theory.” IEEE Trans. On Knowledge-Based Intelligent Electronic Systems, May 1997. pp. 383-386.
- [22] R.S. Michalski and R.E. Stepp. “Automated Construction of Classifications: Conceptual Clustering Versus Numerical Taxonomy.” IEEE Trans. On Pattern Analysis and Machine Intelligence, vol. 5, no. 4, July 1983. pp. 395-410.
- [23] M. Blume and S.C. Esener. “An efficient mapping of Fuzzy ART onto a neural architecture” University of California at San Diego. Neural Networks, 10(3), 1997. pp. 409-411.
- [24] B.V. Robert Pieter Schotema, **Customer relationship management the ultimate guide to the efficient use of CRM**. SCN Education, Vieweg , 2001
- [25] G.A. Carpenter, S. Grossberg. “Adaptive Resonance Theory” Cambridge, Massachusetts: MIT Press, September 1998.

ภาคผนวก

ผลงานวิจัยที่ได้รับการตีพิมพ์เผยแพร่

จุฑามาส ทอเรื่องวิวัฒน์ และ วรพจน์ กรีสระเดช, “ยูนิฟายอะแคปทีพเรโซแนนทียรีสำหรับข้อมูลเชิงตัวเลขและข้อมูลข้อความ,” The 9th National Computer Science and Engineering Conference, UTCC, Thailand, October, 2005. pp. 763-769



The 9th National Computer Science and Engineering Conference

October 27-28, 2005

University of Thai Chamber of Commerce, Bangkok Thailand

Organized by:

Department of Computer Engineering, School of Engineering,
University of Thai Chamber of Commerce

In Cooperation with:

Electrical Engineering, Electronics, Computer, Telecommunications
and Information Technology Association of Thailand (ECTI)



IEEE Communications Society, Thailand Chapter

Sponsored by:

University of Thai Chamber of Commerce



National Electronics and Computer Technology Center (NECTEC)



Sun Microsystems (Thailand)



CS LoosInfo Public Company Limited



Pearson Education Indochina Limited



The OGA Group

ยูนิฟายอะแดปทีฟเรโซแนนซ์ทฤษฎีสำหรับข้อมูลเชิงตัวเลขและข้อมูลข้อความ Unified Adaptive Resonance Theory Neural Networks for Numeric and Text Data

จุฑามาศ ลอเรืองวิวัฒน์¹ และ วรพจน์ กวีสุรเศรษฐ์

คณะเทคโนโลยีสารสนเทศ, สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหาร ลาดกระบัง

กรุงเทพมหานคร 10250

juthamas@inet.co.th¹ worapoj@it.kmitl.ac.th²

บทคัดย่อ

งานวิจัยนี้นำเสนอโครงข่ายประสาทเทียมยูนิฟายอะแดปทีฟเรโซแนนซ์ทฤษฎีซึ่งพัฒนาขึ้นเพื่อใช้ในการกลุ่มข้อมูล โดยสามารถรองรับข้อมูลเชิงตัวเลขและข้อมูลข้อความ โครงข่ายเกิดจากการปรับปรุงพีชคณิตแบบฟัซซีเรโซแนนซ์ทฤษฎี (Fuzzy ART) มาประยุกต์ใช้ร่วมกับการเปรียบเทียบความคล้ายคลึง (Similarity Measure) ซึ่งทำให้โครงข่ายสามารถจัดกลุ่มข้อมูลข้อความได้โดยตรงโดยไม่ต้องทำการปรับปรุงข้อมูล ทั้งนี้การทดสอบความสามารถของโครงข่ายโดยทำการทดสอบการเรียนรู้กับกลุ่มข้อมูลที่สร้างขึ้นด้วยวิธีการสุ่มพบว่าโครงข่ายสามารถแยกความแตกต่างของข้อมูลโดยจัดกลุ่มได้อย่างถูกต้อง

Abstract

This paper proposes a new neural network, called a unified adaptive resonance theory neural network, for clustering objects whose feature values may combine numeric and text data. The proposed model is based on the concepts of fuzzy ART neural networks and similarity measure of symbolic objects. The proposed model works directly on textual and numerical information without mapping objects onto some representations that have quantitative features.

The inputs of the proposed neural network can directly receive both a textual value and a numerical value without mapping the text value into a numerical value. The preliminary experimental results are shown the unified adaptive resonance theory neural network can correctly clustering data.

Key Words: Neural networks, adaptive resonance theory, data clustering, data mining.

1. บทนำ

การจัดกลุ่มมีจุดประสงค์หลักในการจัดข้อมูลที่ไม่ทราบความสัมพันธ์ออกเป็นกลุ่ม โดยใช้ค่าความคล้ายคลึงเป็นเกณฑ์ในการแตกกลุ่ม ทั้งนี้ภายหลังการจัดกลุ่มจะได้ข้อมูลที่มีความเหมือนกันถูกจัดให้อยู่กลุ่มเดียวกัน [7] ซึ่งข้อมูลในปัจจุบันในธุรกิจต่างๆมักเป็นข้อมูลที่มีความซับซ้อนคือชุดข้อมูลจะประกอบด้วยเซตของคุณลักษณะตัวอย่างเช่น ในการจัดกลุ่มข้อมูลรถยนต์ ชุดข้อมูลจะประกอบด้วย รุ่น น้ำหนัก รูปทรง เครื่องยนต์ ซึ่งจะเห็นว่าข้อมูลจะประกอบไปด้วยข้อมูลเชิงตัวเลขและข้อมูลข้อความ ดังนั้นการจัดกลุ่มข้อมูลที่จะครอบคลุมการใช้งานจริงจึงควรจะสามารถรองรับข้อมูลทั้ง 2 ประเภทได้

โครงข่ายประสาทเทียมก็เป็นวิธีที่ได้รับความนิยม โดยรูปแบบที่พบบ่อยได้แก่ Kohonen, SOM, Competitive Learning และ ART อัลกอริทึมพื้นฐานส่วนใหญ่จะมีความสามารถในการจัดกลุ่มข้อมูลที่เป็นไบนารีหรือข้อมูล

ที่เป็นเชิงตัวเลข (Number/Analog) เช่น งานวิจัยของ Carpenter G. A., Grossberg S. and Rosen D. A. [5] ที่นำเสนอ Fuzzy ART ที่สามารถใช้กับข้อมูลที่เชิงตัวเลขได้ ส่วนอัลกอริทึมสำหรับข้อมูลข้อความ (Text) ก็มีการพัฒนาความสามารถอัลกอริทึมในเรื่องการจัดกลุ่ม โดยยุคแรกๆจะเป็นการแปลงรูปข้อมูลข้อความนั้นให้เป็นข้อมูลเชิงตัวเลข ซึ่งอาจทำให้เสียความหมายบางอย่างของข้อมูล และใช้เวลานานในการแปลงรูปข้อมูล และต่อมาได้มีการพัฒนาให้สามารถรับข้อมูลข้อความได้โดยตรง ได้แก่ การประยุกต์ใช้ Kohonen[11] Competitive Learning[12] และ Text Processing [13]

ในงานวิจัยนี้จะนำเสนอโครงข่ายประสาทเทียมที่พัฒนามาจากอะแคปทีฟเวโรเซียนเน็ตเวิร์ก และประยุกต์การวัดคล้ายคลึงของข้อมูลมาใช้แทนฟังก์ชันการเปรียบเทียบแบบเดิมของโครงข่าย โดยหัวข้อต่างๆแบ่งเป็น หัวข้อที่ 2 เสนอการเปรียบเทียบข้อมูลด้วยวิธีการหาความคล้ายคลึง หัวข้อที่ 3 นำเสนอพื้นฐาน ART เมื่อค้นและโครงข่ายที่พัฒนาขึ้นมาใหม่ และหัวข้อที่ 4 จะเป็นผลของการทดลองในการให้โครงข่ายเรียนรู้ชุดข้อมูลที่สร้างขึ้นมาอย่างสุ่มเพื่อการทดสอบ

2. การเปรียบเทียบหาค่าความคล้ายคลึงด้วย

Similarity Measure

ชุดข้อมูลมักจะถูกประกอบด้วยส่วนประกอบที่เป็นคุณลักษณะของชุดข้อมูลนั้น ซึ่งสามารถเขียนในรูป Cartesian product เช่น ชุดข้อมูล A มีสมาชิกหรือมีคุณลักษณะของข้อมูล k ค่า จะสามารถเขียนเป็น $A = A_1 \times A_2 \times \dots \times A_k$ โดยที่ A_k เป็นเซตของข้อมูลที่มีสมาชิก d ค่า $A_k = (a_1, a_2, \dots, a_d)$ งานวิจัยของ K.C. Gowda และ E. Diday [3] ได้เสนอการหาค่าความคล้ายคลึงของข้อมูล 2 ชุด โดยแทนค่าความคล้ายคลึงของชุดเอกสารด้วย $S(A, B)$ ซึ่งคำนวณได้จากผลรวมค่าความคล้ายคลึงของแต่ละคุณสมบัตินี้

$S(A, B) = S(A_1, B_1) + \dots + S(A_k, B_k)$ ทั้งนี้ ค่าความคล้ายคลึงคำนวณได้จาก 3 ส่วน

- $S_p(A_k, B_k)$ เปรียบเทียบในส่วนของตำแหน่ง โดย

$$S_p(A_k, B_k) = 1 - \frac{|a_i - b_j|}{|U_k|} \quad (1)$$

- $S_s(A_k, B_k)$ ที่เปรียบเทียบในส่วนของกระจายหรือจำนวนของข้อมูล โดย

$$S_s(A_k, B_k) = \frac{(I_a + I_b)}{2 \times I_s} \quad (2)$$

- $S_c(A_k, B_k)$ ที่เปรียบเทียบในส่วนของเนื้อหา โดย

$$S_c(A_k, B_k) = \frac{I_i}{I_s} \quad (3)$$

เมื่อ

a_i = ค่าค่าสุดใน A_k

a_u = ค่ามากที่สุดใน A_k

b_i = ค่าค่าสุดใน B_k

b_u = ค่ามากที่สุดใน B_k

$I_s = |\max(a_u, b_u) - \min(a_i, b_i)|$

$I_a = |a_u - a_i|$ หรือจำนวนสมาชิกของ A_k

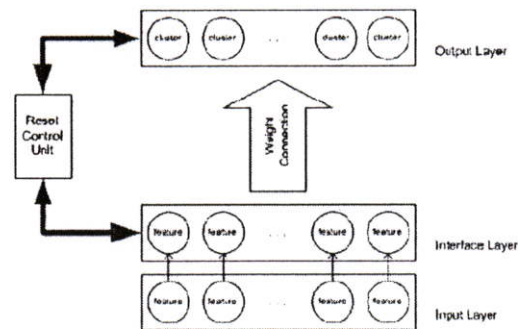
$I_b = |b_u - b_i|$ หรือจำนวนสมาชิกของ B_k

I_i = จำนวนสมาชิกที่เหมือนกันของ A_k และ B_k

U_k = ช่วงของข้อมูลหรือโดเมนของคุณลักษณะที่ k

ทั้งนี้ค่าการคำนวณข้างต้นเป็นการเปรียบเทียบข้อมูลทั้งหมด คือเป็นการเปรียบเทียบความคล้ายคลึง A_k และ B_k โดยผลของการเปรียบเทียบจะได้ค่ามากที่สุด เมื่อ $A_k = B_k$ ซึ่งในงานวิจัยนี้ การเปรียบเทียบจะเป็นการวัดค่าความใกล้เคียงของข้อมูลกับกลุ่มต่างๆของโครงข่าย ดังนั้นจึงปรับการเปรียบเทียบในส่วนของเนื้อหาเป็นดังสมการที่ 4

$$S_c(A_k, B_k) = \frac{I_i}{I_s} \begin{cases} I_s - I_a; I_a > I_i \\ I_s - I_b; I_b \leq I_i \end{cases} \quad (4)$$



รูปที่ 1 แสดงโครงสร้าง Unified ART

3. ยูนิฟายอะแดพทีฟเรโซแนนซ์เทียรี

อะแดพทีฟเรโซแนนซ์เทียรี (Adaptive Resonance Theory) หรือ ART พัฒนาโดย Stephen Grossberg และ Gail Carpenter ซึ่งนำเสนอเมื่อปีค.ศ. 1987 เพื่อใช้งานทางด้านจดจำรูปแบบ การจับกลุ่มข้อมูล (pattern recognition) โดยเริ่มจากการออกแบบอัลกอริทึมเพื่อใช้กับข้อมูลไบนารี และใช้ชื่อว่า ART1 และพัฒนา ART2 ในเวลาต่อมาให้สามารถใช้กับข้อมูลอนาล็อกได้ และเมื่อปีค.ศ. 1991 Carpenter Grossberg และ Rosen ได้เสนอ FUZZY ART ที่ใช้พื้นฐานจาก ART1 มาพัฒนาให้มีความสามารถใช้กับข้อมูลอนาล็อก นอกจากนี้ยังมีโครงข่ายที่เกิดขึ้นจากการพัฒนาเพิ่มเติมอีกมากมาย เช่น ARTMAP, Fuzzy

ARTMAP, Adaptive Hamming Net (AHN), Gaussian ART (GA), Simplified Fuzzy ARTMAP (SFAM), Simplified ART (SART) [9]

ART เป็นโครงข่ายประสาทเทียมแบบเรียนรู้ด้วยตนเอง (Unsupervised Learning) ที่ใช้ Competitive learning และกฎ Winner-take-all ซึ่งโครงข่ายของ ART เป็นโครงข่ายประสาทเทียมที่มี 2 ชั้น (Layer) คือชั้นที่เป็นอินพุตและชั้นที่เป็นเอาต์พุต ซึ่งในงานวิจัยบางแห่งอาจจะแบ่งชั้นย่อยในชั้นอินพุตเป็นชั้นอินพุตและชั้นอินเทอร์เฟส โดยชั้นอินพุตจะรับข้อมูลจากภายนอกโครงข่ายแล้วส่งต่อไปให้ชั้นอินเทอร์เฟส จากนั้นชั้นอินเทอร์เฟสซึ่งเชื่อมต่อกับ

Notation	Description
p	number of sampling data
m	number of input nodes with i as index
n	number of output nodes with j as index
X	set of patterns of sample, $X = \{x_1, x_2, \dots, x_p\}$
\mathbf{x}	an input pattern on input layer, $\mathbf{x} = (x_1, x_2, \dots, x_m)$
β	learning rate
ρ	vigilance
W	weight connection
w_j	weight connection from all input node to output node j
e_j	degree of weight connection from all input node to output node j
$S(\mathbf{x}, \mathbf{w}_j)$	similarity between \mathbf{x} and \mathbf{w}_j

ตารางที่ 1 แสดงพารามิเตอร์ที่ใช้ในงานวิจัย

```

Initialize:
  L, ρ, Default Degree
C = 0
while (X not empty)
  Learning Loop
  {
    get x;
    new = true;
    loop j = 1:C
       $T_j = T(x, w_j)$ ;
    loop i = 1:C
      {
         $J = \text{index of max } T_j$ 
        if  $M(x, w_j) > \rho$ 
          {
             $w_j = U(x, w_j)$ ;
            new = false;
            break;
          }
        else
           $T_j = -1$ ;
      }
    newNode(new);
  }

```

รูปที่ 2 แสดงอัลกอริทึมของโครงข่าย Unified ART

ชั้นเอาร์ทุกด้วยค่าถ่วงน้ำหนัก 2 จุด คือ Bottom-up Weight (Forward Connection) และ Top-down Weight (Backward Connection) ทั้งนี้ได้มีงานวิจัยที่นำเสนอการเชื่อมต่อระหว่างชั้นว่า การใช้ค่าถ่วงน้ำหนัก 2 จุด สามารถแทนที่ด้วยการใช้การใช้ค่าถ่วงน้ำหนักเพียงจุดเดียว ซึ่งให้ผลของการเรียนรู้เท่ากัน[9]

Unified ART พัฒนาคือจาก ART ซึ่งโครงข่าย Unified ART แสดงดังรูปที่ 1 โดยในชั้นอินพุทจะประกอบไปด้วย โหนดต่างๆ ซึ่งแต่ละ โหนดหมายถึงแต่ละคุณลักษณะหรือ feature ซึ่งสามารถรับได้ทั้งข้อมูลข้อความและข้อมูลตัวเลข ในชั้นเอาร์พุทจะประกอบด้วย โหนดต่างๆ ซึ่งแต่ละ โหนดหมายถึงกลุ่มของข้อมูลหรือ cluster ที่โครงข่ายสร้างขึ้นหลังจากการเรียนรู้ ส่วน Weight Connection หมายถึงค่าถ่วงน้ำหนักที่เป็นความจำของโครงข่ายที่ใช้การเรียนรู้และแบ่งแยกกลุ่มข้อมูล โดยพารามิเตอร์ต่างๆที่ใช้ในงานวิจัย ตามที่แสดงในตารางที่ 1

ในการเรียนรู้จะดำเนินการตามอัลกอริทึมดังรูปที่ 2 ซึ่งจะมีฟังก์ชัน NEWNODE(new) ซึ่งหมายถึงการเพิ่มโหนดเอาร์พุทหรือเพิ่มจำนวนกลุ่ม โดยโหนดที่เพิ่มจะให้ค่าเริ่มต้นเป็นค่าอินพุทนั่นเอง ทั้งนี้ฟังก์ชันที่สำคัญในการคำนวณจะแบ่งตามประเภทของข้อมูลดังนี้

- ข้อมูลที่เป็นตัวเลข
 - Choice Function สำหรับข้อมูลที่เป็นตัวเลข

$$T(x, w_j) = S(x, w_j)$$
 - Match Function

$$M(x, w_j) = S(x, w_j)$$
 - Update Function

$$U(x, w_j) = (1 - \beta)w_j + \beta(x \wedge w_j)$$
- ข้อมูลที่เป็นข้อความ
 - Choice Function สำหรับข้อมูลที่เป็นข้อความ

$$T(x, w_j) = S(x, w_j)$$

- Match Function

$$M(x, w_j) = S(x, w_j)$$

- Update Function

$$U(x, w_j) = (x \cup w_j)$$

และ Update Function สำหรับ Degree ของข้อมูลดังนี้

$$U(x, e_j) = \begin{cases} e_j + \alpha; & \text{if } w_j \in x \\ e_j - \alpha; & \text{if } w_j \notin x \\ \text{default degree; if } x \notin w_j \end{cases}$$

ซึ่งจะปรับค่า degree ให้อยู่ในช่วง 0-1 โดยถ้าผลการคำนวณมากกว่า 1 จะปรับให้เท่ากับ 1 และถ้ามีค่าน้อยกว่า 0 จะปรับค่าให้เท่ากับ 0 โดยในกรณีที่ค่า degree ของข้อมูลเป็น 0 จะเอาข้อมูลนั้นออกจากชุดของค่าตัวงน้ำหนัก

4. ผลการทดลอง

ในการทดลองการเรียนรู้เพื่อให้สามารถวัดผลของการจัดกลุ่มได้ งานวิจัยนี้จำได้สร้างจากกลุ่มข้อมูลที่กำหนดอย่างสุ่มขึ้นมา 100 ชุด ซึ่งประกอบด้วยคุณลักษณะที่เป็นข้อความและตัวเลขอย่างละ 2 ค่า โดยข้อมูลแบบข้อ

Cluster	Feature1	Feature2	Feature3	Feature4	No. of data from		
					Class1	Class2	Class3
1	{j,k,m,n}	{i,j,mm,nn}	0.8028	0.9813	0	36	0
2	{a,c,g,h}	{aa,cc,dd,gg,hh}	0.0000	0.8785	33	0	0
3	{u,v,t,r}	{uu,vv,rr,tt}	0.3827	0.6916	0	0	31

ตารางที่ 2 แสดงผลการเรียนรู้ของโครงข่าย โดยกำหนดค่า vigilance = 0.6

Cluster	Feature1	Feature2	Feature3	Feature4	No. of data from		
					Class1	Class2	Class3
1	{j,k,m,n}	{i,j,mm,nn}	0.8033	0.9813	0	36	0
2	{h}	{hh}	0.0169	0.8785	3	0	0
3	{a,g,h}	{aa,cc,dd,gg,hh}	0.0000	0.8785	30	0	0
4	{u,v,t,r}	{uu,vv,rr,tt}	0.3827	0.6916	0	0	31

ตารางที่ 3 แสดงผลการเรียนรู้ของโครงข่าย โดยกำหนดค่า vigilance = 0.7

5. สรุป

งานวิจัยนี้ได้นำเสนอวิธีการจัดกลุ่มข้อมูลโดยประยุกต์โครงข่ายประสาทเทียมและการเปรียบเทียบความคล้ายคลึงมาใช้ ซึ่งทำให้โครงข่ายที่นำเสนอมีความสามารถในการใช้งานทั้งข้อมูลเชิงข้อความและ

ความสามารถเป็นเซตของข้อมูล ดังแสดงในรูปที่ 3 โดยในคอลัมน์ class คือหมายเลขของกลุ่ม

การเรียนรู้ของโครงข่ายโดยกำหนดค่า vigilance เป็น 0.6 และค่าคิรีเริ่มต้นเป็น 0.7 ให้ผลลัพธ์ดังแสดงในตารางที่ 2 ซึ่งจะเห็นว่าโครงข่ายสามารถจัดกลุ่มได้ถูกต้องคือกลุ่มของข้อมูลหรือ Cluster ที่โครงข่ายจัดได้นั้น จะเป็นข้อมูลจาก class 2, 1 และ 3 ตามลำดับ

ในงานวิจัยได้ทำการทดสอบอีกครั้งโดยการกำหนดค่า vigilance เป็น 0.7 และค่าคิรีเริ่มต้นเป็น 0.7 ผลการทดลองแสดงให้เห็นว่าค่าคิรีความเหมือนกันของข้อมูลในการยอมรับให้อยู่กลุ่มเดียวกันที่มากขึ้นมีผลต่อการจัดกลุ่มของโครงข่าย โดยทำให้จำนวนกลุ่มของโครงข่ายเพิ่มมากขึ้น ดังแสดงผลลัพธ์ในตารางที่ 3 ซึ่งจะเห็นว่าโครงข่ายสามารถจัดกลุ่มได้เป็น 4 กลุ่ม โดยเป็นข้อมูลจาก class 2, 1, 1 และ 3 ตามลำดับ

ทั้งนี้ในส่วนของคอลัมน์ Feature นั้นคือค่าตัวงน้ำหนักที่เกิดขึ้นหลังจากการเรียนรู้ ซึ่งก็คือ template ของกลุ่มต่างๆที่เกิดจากการเรียนรู้ของโครงข่าย

ข้อมูลตัวเลข โดยสำหรับข้อมูลเชิงข้อความนั้นสามารถใช้งานได้โดยไม่ต้องมีการแปลงข้อมูลก่อน

ทั้งนี้ผลของการทดลองแสดงให้เห็นว่าโครงข่ายสามารถเรียนรู้ข้อมูลที่ใช้ในการทดสอบได้ดี โดยสามารถจัดกลุ่มข้อมูลได้อย่างถูกต้อง

data	class	feature1	feature2	feature3	feature4	data	class	feature1	feature2	feature3	feature4
1	2	jn	mmnm	0.80328	0.98131	61	2	r	mmnm	1	0.98131
2	1	hk	mmnm	0.032787	0.8785	62	2	kmn	hk	0.96721	0.99065
3	2	ljn	mmnm	0.80328	0.98131	63	1	stv	mmnm	0.37705	0.70053
4	1	cdjh	ggjh	0.016393	0.8972	64	1	lv	mmnm	0.32787	0.89169
5	3	v	sslluuw	0.32787	0.69159	65	3	stv	mmnm	0.37705	0.70053
6	3	rstv	ssw	0.37705	0.69159	66	2	r	jjk	1	0.98131
7	2	jn	mmnm	0.96721	0.98131	67	1	ack	aa ggjh	0	0.89785
8	2	m	lmmnm	0.96721	0.98131	68	1	djh	aa ggjh	0.032787	0.89785
9	2	n	lmmnm	0.80328	0.98131	69	1	ac ghj	ddjh	0	0.89785
10	3	v	rllw	0.32787	0.72897	70	3	rslu	mmnm	0.32787	0.70053
11	2	jn	mmnm	0.80328	0.99065	71	2	km	mmnm	0.26328	0.99065
12	1	cdjh	ddjh	0.032787	0.8785	72	1	djh	cc dd ggjh	0.816393	0.8785
13	2	kk	jjk	0.80328	0.98131	73	1	r	hh	0.016393	0.8785
14	2	kmn	mmnm	0.96721	0.98131	74	1	cgh	ddjh	0	0.8972
15	3	stvw	ssw	0.42623	0.69159	75	3	uv	mmnm	0.37705	0.89169
16	3	stvw	ssw	0.42623	0.69159	76	3	stvw	mmnm	0.32787	0.72897
17	3	stvw	ssw	0.37705	0.69159	77	1	cdh	aa dd ggjh	0.416393	0.8972
18	2	jn	mmnm	0.80328	0.99065	78	2	mm	kkmmnm	0.80328	0.98131
19	3	stlax	st	0.37705	0.70053	79	2	km	kk	0.80328	0.99065
20	3	stvw	ssw	0.32787	0.72897	80	3	uv	ssw	0.37705	0.89169
21	1	cdjh	dd ggjh	0	0.89785	81	1	ad ghj	aa dd hh	0	0.89785
22	2	kk	mm	0.96721	0.98131	82	2	km	mm	0.96721	1
23	3	stvw	ssw	0.32787	0.70053	83	3	rsluv	ssw	0.42623	0.72897
24	1	cdjh	aa dd ggjh	0.032787	0.89785	84	2	stvw	mmnm	0.32787	0.72897
25	3	stvw	stvw	0.32787	0.70053	85	2	jn	jjj	0.80328	0.99065
26	3	stvw	ssw	0.32787	0.70053	86	2	jn	jjj	1	1
27	1	ghj	dd ggjh	0	0.89785	87	2	r	mm	0.80328	0.98131
28	2	kmn	kkmmnm	0.96721	0.98131	88	2	jn	mm	0.56721	0.99065
29	2	m	jjk	1	0.98131	89	2	mm	jjj	0.80328	0.99065
30	2	kmn	mm	1	1	90	1	ad ghj	aa dd ggjh	0.016393	0.8785
31	2	kmn	mmnm	0.80328	0.98131	91	1	gh	ggjh	0	0.8972
32	1	cdjh	cc dd hh	0.032787	0.8785	92	3	uv	mm	0.37705	0.89169
33	2	jn	lmm	0.96721	1	93	2	kmn	jjj	0.56721	0.99065
34	1	cd ghj	cc dd hh	0	0.89785	94	2	kmn	jjj	0.96721	0.99065
35	1	ghj	hh	0.016393	0.89785	95	1	ad h	dd ggjh	0.032787	0.8972
36	1	ad ghj	cc dd hh	0.032787	0.8785	96	1	ah	cc dd hh	0.016393	0.8972
37	3	stvw	stvw	0.42623	0.69159	97	2	ah	ggjh	0.016393	0.8972
38	3	stvw	stvw	0.32787	0.69159	98	2	kmn	mmnm	0.96721	1
39	3	stvw	stvw	0.37705	0.69159	99	1	ack	dd ggjh	0.016393	0.89785
40	3	stvw	stvw	0.42623	0.72897	100	2	kmn	jjj	0.56721	0.98131
41	2	kmn	kkmmnm	0.96721	0.98131	99	1	cgh	ddjh	0.032787	0.89785
42	1	cd ghj	hh	0.032787	0.89785	100	3	uv	mmnm	0.37705	0.70053
43	2	jjk	mmnm	1	0.99065	99	3	uv	ss ll uu w	0.42623	0.69159
44	2	kmn	jjj	0.80328	1	94	2	jn	jjj	0.80328	1
45	2	kmn	mmnm	1	1	95	1	djh	cc dd hh	0	0.89785
46	1	ghj	cc dh	0	0.8785	96	3	stvw	ss ll w	0.37705	0.72897
47	3	stvw	stvw	0.37705	0.72897	97	3	uv	mmnm	0.37705	0.72897
48	3	stvw	stvw	0.37705	0.69159	98	1	agh	aa cc hh	0.016393	0.8785
49	1	h	hh	0.016393	0.8785	99	1	agh	ggjh	0	0.8972
50	1	cdjh	hh	0.016393	0.8972	100	1	gh	ddjh	0.016393	0.89785

รูปที่ 3 แสดงข้อมูลที่ใช้ในการเรียนรู้ของโครงข่าย

6. เอกสารอ้างอิง

[1] Cabena P. et al, Discovering data mining from concept to implementation., Prentice Hall PTR. 1997.

[2] Lanrene F., Fundamentals of Neural Networks Architecture, Algorithms and Application., New Jersey : Prentice Hall International. 1994.

[3] Gowda K.C. and Diday E., "Symbolic Clustering Using a New Similarity Measure.", IEEE Trans. On Syst., Man, Cybern., vol. 22, no. 2, March/April 1992. pp. 368-378.

[4] El-SonBaty YA and Ismail M. A., "Fuzzy Clustering for Symbolic Data.", IEEE Trans. On Fuzzy Syst., vol. 6, no. 2, May 1998. pp. 195-204.

- [5] Carpenter G. A., Grossberg S. and Rozen D. A., "Fuzzy ART: Fast Stable Learning and Categorization of Analog Patterns by an Adaptive Resonance System.", Pergamon Press plc., Neural Network, Vol. 4, 1991, pp. 759-771.
- [6] Freeman A.J. and Skapura M.D., Neural Networks: algorithms, applications, and programming techniques., Addison-Wesley. 1992.
- [7] A.K.Jain, M.N. Murty and P.J. Flynn, "Data Clustering: A Review", ACM Computing Surveys, Vol. 31, No. 3, September 1999, pp. 264-323
- [8] Gautam B., Jerry B.W. and Douglas H.F. , "ITERATE: A Conceptual Clustering Algorithm for Data Mining" IEEE, Applications and Reviews, Vol. 28, No. 2, May 1998. pp. 219-241
- [9] ____, [Online].Available: <http://medusa.sdsu.edu:Robotics/Neuromuscular/Theses/Hongyu/chapter3.pdf>
- [10] M. Narasimha Murty et al. Jan 2004. Clustering Large Symbolic Datasets. [Online]. Available: http://www.ceremade.dauphine.fr/~touati/workshop/PDF/18_Clustering_Large_Symb_Datasets_Narashima.pdf
- [11] นายทรงพล ชุติพงศ์พัฒนาภักดิ์, "ที่กโปเรชตังโกล โธเนนนิวอลเน็คเวิร์คโดยใช้กระบวนการเรียนรู้แบบใหม่", วิทยานิพนธ์วิทยาศาสตร์มหาบัณฑิต สาขาวิชาวิทยาการคอมพิวเตอร์และเทคโนโลยีสารสนเทศ บัณฑิตวิทยาลัย, สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง, 2546
- [12] นายสมคิด แสนเสนาะ, "การแบ่งกลุ่มเอกสารโดยใช้เทคนิคการประมวลผลข้อความด้วยโครงข่ายประสาทเทียมที่เรียนรู้แบบหาผู้ชนะ", วิทยานิพนธ์วิทยาศาสตร์มหาบัณฑิต สาขาวิชาวิทยาการคอมพิวเตอร์และเทคโนโลยีสารสนเทศ บัณฑิตวิทยาลัย, สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง, 2547
- [13] นายนครินทร์ จันทสุต, "ที่กอะแคปทิฟเรโซแนนซ์ที่นิวอลเน็คเวิร์ค" วิทยานิพนธ์วิทยาศาสตร์มหาบัณฑิต สาขาวิชาวิทยาการคอมพิวเตอร์และเทคโนโลยีสารสนเทศ บัณฑิตวิทยาลัย, สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง, 2547

ประวัติผู้เขียน

ชื่อ-สกุล	นางสาวจุฑามาส นอเรืองวิวัฒน์
วัน-เดือน-ปีเกิด	วันที่ 9 พฤศจิกายน 2521 ที่อำเภอเมือง จังหวัดสมุทรปราการ
ประวัติการศึกษา	พ.ศ. 2542 ปริญญาตรีวิทยาศาสตร์บัณฑิต สาขาสถิติประยุกต์ จาก ภาควิชาสถิติประยุกต์ คณะวิทยาศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้า เจ้าคุณทหารลาดกระบัง
ประสบการณ์การทำงาน	
พ.ศ. 2542-2543	บริษัท ไอซีซีอินเตอร์เนชันแนล จำกัด (มหาชน) ในตำแหน่ง โปรแกรมเมอร์
พ.ศ. 2543-2544	บริษัท เอเชียเบลล์ จำกัด ในตำแหน่ง โปรแกรมเมอร์
พ.ศ. 2546-ปัจจุบัน	บริษัท อินเทอร์เน็ตประเทศไทย จำกัด (มหาชน) ในตำแหน่ง โปรแกรม- โคออดิเนเตอร์