เท็กอะแดปทีพเรโซแน้นเทียรีนิวรอลเน็ตเวิร์ค

A TEXT ADAPTIVE RESONANCE THEORY NEURAL NETWORK

นรเศรษฐ์ จันทสูตร

NORRASETH CHANTASUT

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต
สาขาวิชาเทคโนโลยีสารสนเทศ
บัณฑิตวิทยาลัย
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ. 2547
ISBN 974-15-1150-7

A TEXT ADAPTIVE RESONANCE THEORY NEURAL NETWORK

NORRASETH CHANTASUT

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENT FOR THE DEGREE OF
MASTER OF SCIENCE IN INFORMATION TECHNOLOGY
SCHOOL OF GRAUDATE STUDIES
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG
2004

ISBN 974-15-1150-7

หัวข้อวิทยานิพนธ์	**เท็กอะแดปทีพเรโซแน้นเทียรีนิวรอลเน็ตเวิร์ค**
นักศึกษา	**นายนรเศรษฐ์ จันทสูตร**
รหัสประจำตัว	43067138
ปริญญา	**วิทยาศาสตรมหาบัณฑิต**
สาขาวิชา	**เทคโนโลยีสารสนเทศ**
พ.ศ.	2547
อาจารย์ผู้ควบคุมวิทยานิพนธ์	**ผศ.ดร.วรพจน์ กรีสุระเดช**

บทคัดย่อ

การศึกษาส่วนใหญ่ของเดต้าไมนิ่ง (Data Mining) มุ่งเน้นเกี่ยวกับข้อมูลที่มีโครงสร้าง (Structured Data) แต่อย่างไรก็ตามข้อมูลเอกสารประเภทต่างๆ เช่น ข่าว บทความ เอกสารงานวิจัย เป็นต้น ซึ่งจะจัดเก็บ ข้อมูลที่เก็บอยู่ในฐานข้อมูลเอกสาร (Document Databases) การจัดกลุ่มเอกสารเหล่านี้จึงเป็นเรื่องที่มีความสำคัญสำหรับการวิเคราะห์การจัดแบ่งกลุ่ม วิทยานิพนธ์นี้จึงนำเสนออัลกอริทึมการจัดกลุ่มเอกสารโดยใช้ เท็กอะแดปทีพเรโซแน้นเทียรีนิวรอลเน็ตเวิร์ค (Text Adaptive Resonance Theory Neural Network) วิธีการของเท็กอะแดปทีพเรโซแน้นเทียรีนิวรอลเน็ตเวิร์ค (Text Adaptive Resonance Theory Neural Network) ถูกออกแบบเพื่อแบ่งกลุ่มข้อมูลที่เป็นข้อความโดยตรงและไม่มีการแปลงข้อความเป็นตัวเลข ในการทดลองการทำงานของโครงข่ายประสาทเทียมที่ได้รับการออกแบบใหม่นี้ ได้ใช้ชุดข้อมูล 2 ชุด ได้แก่ ข้อมูลสังเคราะห์ (Synthesized Dataset) ซึ่งสร้างขึ้นในห้องปฏิบัติการ Data Mining & Data Exploration Laboratory และ ข้อมูลข่าวรอยเตอร์ (Reuters-21578 Distribution 1.0) ซึ่งถูกรวบรวมจากข้อมูลข่าวจริงของรอยเตอร์ (Reuters) ปี ค.ศ. 1987 ค่า Entropy และ F-Measure ถูกใช้เพื่อวัดผลลัพธ์ความถูกต้องของวิธีการใหม่ จากผลการทดลองโครงข่ายประสาทเทียมที่ได้รับการปรับแต่งนี้สามารถรับข้อมูลที่เป็นข้อความได้โดยตรง และทำการจัดกลุ่มข้อมูลที่มีคุณสมบัติมีค่าเป็นข้อความได้เป็นอย่างดี

Thesis Title	A Text Adaptive Resonance Theory Neural Network
Student	Mr. Norraseth Chantasut
Student ID.	43067138
Degree	Master of Science
Programme	Information Technology
Year	2004
Thesis Advisor	Asst.Prof.Dr.Worapoj Kreesuradej

ABSTRACT

The most studies of data mining have focused on structured data such as relational, transactional, and data warehouse data. However, the most available data that consist of large amounts of text documents such as news, articles, and research papers is stored in document database. The ability to group these documents is an important requirement for clustering analysis. This research proposes A Text Adaptive Resonance Theory Neural Network for document clustering. A Text Adaptive Resonance Theory Neural Network is designed to cluster on that data set that has a non-numerical feature value. Consequently, the proposed learning algorithm works directly on textual information without text transformation into a numerical value. The experiments are conducted on 2 datasets. The first dataset is a synthesized dataset, which is generated by the Data Mining & Data Exploration Laboratory and the second dataset is a Reuter-21578 Distribution 1.0 which is collected from the documents appeared on the Reuters newswire in 1987. The Entropy and F-Measure is used to measure the effectiveness of the proposed technique. According to the experimental results, the proposed neural network has shown good performance in clustering textual data.

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงได้อย่างดี ด้วยคำแนะนำและคำปรึกษาเกี่ยวกับแนวทางในการทำวิจัยจาก ผศ.ดร.วรพจน์ กรีสุระเดช ซึ่งเป็นอาจารย์ผู้ควบคุมวิทยานิพนธ์ เพื่อที่จะหาอัลกอริทึมใหม่ที่เหมาะสมสำหรับการจัดแบ่งกลุ่มข้อมูลเอกสารข่าวโดยเฉพาะ ทั้งนี้ผู้วิจัยรู้สึกซาบซึ้งในความอนุเคราะห์จากท่านและขอกราบขอบพระคุณเป็นอย่างสูงไว้ ณ ที่นี้

นอกจากนั้นข้าพเจ้าขอขอบคุณและเอ่ยนามถึงบุคคลต่างๆ ที่ได้มีส่วนร่วมในการสร้างงานวิจัยชิ้นนี้ให้เสร็จสมบูรณ์ขึ้นมาได้

ขอกรอบขอบพระคุณบุพการี ผู้ให้สติปัญญา ความคิดอ่าน และคอยให้กำลังใจข้าพเจ้าและให้การสนับสนุนค่าใช้จ่ายส่วนที่สำคัญต่างๆ ตลอดมา

ขอขอบพระคุณ รศ.ดร.อาริต ธรรมโน และอาจารย์วารุนี เครือคล้าย รวมทั้งคณาจารย์ในคณะเทคโนโลยีสารสนเทศทุกท่านที่ได้ประสิทธิ์ประสาทความรู้ต่างๆ รวมทั้งให้คำปรึกษาเมื่อมีข้อสงสัยเพื่อใช้เป็นพื้นฐานความรู้ในการทำงานวิจัยและแก้ปัญหาที่พบในการทำงานวิจัย ตลอดจนห้องปฏิบัติการ Data Mining & Data Exploration Lab ที่ได้เอื้อเฟื้อสถานที่ตลอดการทำวิจัยนี้

ขอขอบคุณเจ้าหน้าที่คณะเทคโนโลยีสารสนเทศที่ได้ให้ความอนุเคราะห์ในความสะดวกต่างๆในการทำงานตั้งแต่ต้น

นอกจากนี้ขอขอบคุณพี่ๆ เพื่อนๆ นักศึกษาทุกคนที่คอยให้ความช่วยเหลือ ให้คำแนะนำและกำลังใจข้าพเจ้ามาตลอด รวมทั้งคุณทรงพล ชุติพงศ์พัฒนกุล และคุณสมคิด แสนเสนาะ ในการแลกเปลี่ยนความรู้และข้อมูลกลุ่มข่าวสำหรับใช้ในการทดลองการทำงานของอัลกอริทึม

สุดท้ายนี้ขอขอบพระคุณ ดร.จุฬารัตน์ ตันประเสริฐ และ ดร.นพดล ศิริเพ็ชร รวมทั้งกลุ่มงานวิจัยเทคโนโลยีคลังข้อมูล ฝ่ายวิจัยและพัฒนาสาขาเทคโนโลยีคอมพิวเตอร์เพื่อการคำนวณ ศูนย์เทคโนโลยีอิเล็กทรอนิกส์และคอมพิวเตอร์แห่งชาติ ที่ได้ให้การสนับสนุนงานวิทยานิพนธ์นี้

คุณค่าและประโยชน์อันพึงมีจากวิทยานิพนธ์ฉบับนี้ ผู้วิจัยขอมอบแด่ผู้มีพระคุณทุกท่าน

นรเศรษฐ์ จันทสูตร

สารบัญ

หน้า

บทคัดย่อภาษาไทย I

บทคัดย่อภาษาอังกฤษ II

กิตติกรรมประกาศ III

สารบัญ IV

สารบัญตาราง VI

สารบัญรูป VII

บทที่ 1 บทนำ 1

1.1 ความเป็นมา ความสำคัญของปัญหา 1

1.2 จุดประสงค์ของงานวิจัย 3

1.3 แผนการดำเนินงานวิจัย 3

1.4 เครื่องมือที่ใช้ในการวิจัย 4

1.5 ประโยชน์ที่คาดว่าจะได้รับจากงานวิจัยครั้งนี้ 4

บทที่ 2 หลักการและทฤษฎีที่เกี่ยวข้อง 5

2.1 หลักการจัดแบ่งกลุ่มเอกสาร (Document Clustering) 5

2.1.1. Document Representation 6

2.1.2. Similarity Measures 10

2.1.3. Clustering Algorithms 11

2.2 โครงข่ายประสาท (Neural Network) 15

2.3 โครงข่ายประสาทเทียม (Artificial Neural Network) 17

2.4 Adaptive Resonance Theory Neural Network 18

2.4.1 ART1 Network 19

2.4.2 ART1 Clustering Algorithms 20

2.5 การวัดคุณภาพของการจัดกลุ่ม (Cluster Evaluation Measure) 22

2.5.1 Entropy 23

2.5.2 F-Measure 24

สารบัญ (ต่อ)

หน้า

บทที่ 3 ทฤษฎีและหลักการทำงานของ Text ART Neural Network 26

3.1 Document Representation 26

3.2 Similarity Measure for Symbolic Objects 27

3.3 A Text Adaptive Resonance Theory Neural Network 29

3.4 Learning Algorithms 30

บทที่ 4 วิธีการดำเนินการวิจัย 39

4.1 ข้อมูล Synthesized Alphabet Document 39

4.2 ข้อมูล Synthesized Text Documents 40

4.3 ข้อมูลข่าว Reuters–21578 42

4.4 สรุปผลการทดลอง 47

บทที่ 5 สรุปผลการวิจัยและข้อเสนอแนะ 52

5.1 สรุปผลงานวิจัย 52

5.2 บทวิเคราะห์ 53

5.3 ข้อเสนอแนะและแนวทางในการทำวิจัยต่อ 54

เอกสารอ้างอิง 55

ภาคผนวก 58

ภาคผนวก ก แสดงตัวอย่างข่าว Reuters-21578. 59

ภาคผนวก ข แสดงตัวอย่างการทำงานของ Text ART Neural Network 68

ภาคผนวก ค ผลงานวิจัยที่ได้รับการตีพิมพ์ 72

ประวัติผู้เขียน 87

สารบัญตาราง

ตารางที่ หน้า

2.1 แสดงการเปรียบเทียบระหว่าง Biological กับ Artificial Neural Network 18

2.2 ตัวอย่างตาราง Confusion Matrix 22

2.3 ตารางแสดงค่า Entropy ของแต่ละ Cluster 23

2.4 ตารางแสดงค่า F-Measure ของแต่ละ Cluster 25

3.1 ข้อมูลเอกสาร 27

4.1 Synthesized Alphabet Document 40

4.2 แสดงผลลัพธ์ที่ได้จากการทดสอบของ Synthesized Alphabet Document 40

4.3 Synthesized Text Document 41

4.4 แสดงผลลัพธ์ที่ได้จากการทดสอบของ Synthesized Text Document 42

4.5 แสดงชุดข้อมูลสำหรับใช้เทรนนิ่งและใช้ทดสอบของข้อมูลข่าว Reuters 3 กลุ่ม 44

4.6 แสดงผลลัพธ์ที่ได้จากการทดสอบของข้อมูลข่าว Reuters 3 กลุ่ม 44

4.7 แสดงชุดข้อมูลสำหรับใช้เทรนนิ่งและใช้ทดสอบของข้อมูลข่าว Reuters 5 กลุ่ม 45

4.8 แสดงผลลัพธ์ที่ได้จากการทดสอบของข้อมูลข่าว Reuters 5 กลุ่ม 45

4.9 แสดงชุดข้อมูลสำหรับใช้เทรนนิ่งและใช้ทดสอบของข้อมูลข่าว Reuters 14 กลุ่ม 46

4.10 แสดงผลลัพธ์ที่ได้จากการทดสอบของข้อมูลข่าว Reuters 14 กลุ่ม 47

ก.1 แสดงตัวอย่างของคำที่เป็น Stop Words 61

ก.2 แสดงตัวอย่างของคำที่ได้หลังจากตัด Stemming ของคำ 63

ก.3 แสดงตัวอย่างของคำที่ตัด stemming และคำที่เป็น stop word 63

ก.4 แสดงตัวอย่างโครงสร้างของข้อมูลข่าว Reuters-21578 ที่นำเข้าโมเดล 64

ก.5 แสดงตัวอย่าง Weight ที่ได้จากการเรียนรู้ (Synthesized Text Document) 65

ก.6 แสดงตัวอย่าง Weight ที่ได้จากการเรียนรู้ (Routers-21578) 66

สารบัญรูป

รูปที่ หน้า

2.1 ขั้นตอนการแบ่งกลุ่มเอกสาร 5

2.2 แสดงแผนภาพ Dendogram ของ Hierarchical Clustering 12

2.2 แสดงแผนภาพ Cluster ของ Partitional Clustering 14

2.3 แสดงแผนภาพวาดของ โครงข่ายประสาทในสมองมนุษย์ 16

2.4 แสดงแผนภาพของโครงข่ายประสาทเทียม 18

2.5 โครงสร้างของ Binary Adaptive Resonance Theory Neural Network 20

3.1 โครงสร้างของ Text Adaptive Resonance Theory Neural Network 29

3.2 แสดง Flow Chart การทำงานของอัลกอริทึม Text ART Neural Network 33

4.1 (a) ชุดตัวอักษรที่ใช้สร้างข้อมูลใน Title 39

4.2 (b)ชุดตัวอักษรที่ใช้สร้างข้อมูลใน Keyword 39

บทที่ 1
บทนำ

1.1 ความเป็นมา ความสำคัญของปัญหา

การศึกษาส่วนใหญ่ของ Data Mining มุ่งเน้นเกี่ยวกับข้อมูลที่มีโครงสร้าง (Structured Data) แต่อย่างไรก็ตามข้อมูลสารสนเทศที่เก็บอยู่จะอยู่ในรูปของ Text Databases หรือ Document Databases ซึ่งจะจัดเก็บเอกสารประเภทต่างๆ เช่น เอกสารบทความ เอกสารงานวิจัย เอกสารข่าว เป็นต้น ข้อมูลที่เก็บอยู่ใน Document Databases ส่วนมากจะเป็นข้อมูลที่มี Attribute แบบโครงสร้างไม่ชัดเจน (Semi-Structured Data) ลักษณะของข้อมูลที่มี Measurement ในลักษณะโครงสร้างแบบ Semi-Structured คือข้อมูลที่มี Attribute ทั้งที่มีโครงสร้าง (Structured Data) และที่ไม่มีโครงสร้าง (Unstructured Data) รวมอยู่ในข้อมูล ตัวอย่างเช่น Title Author Publication_Date จัดเป็น Structured Data ส่วน Abstract และ Contents จัดเป็น Unstructured Data ในขณะที่ระบบฐานข้อมูลที่เก็บเอกสารประเภทต่างๆ ที่มีลักษณะแบบโครงสร้างที่ไม่ชัดเจน (Semi-Structured Data) ได้มีปริมาณของเอกสารมากขึ้นอย่างรวดเร็ว ส่งผลให้ปัจจุบันปริมาณข้อมูลที่เป็นข้อความหรือเอกสารมีปริมาณเพิ่มมากขึ้นอย่างรวดเร็ว วิธีการและอัลกอริทึมต่างๆของ Data Mining และ Text Processing ได้ถูกนำมาประยุกต์เพื่อใช้กับข้อมูลที่เป็นข้อความ หนึ่งในวิธีการที่น่าสนใจคือการจัดแบ่งกลุ่มเอกสาร (Document Clustering) [5]

Document Clustering นั้นคือวิธีการจัดแบ่งกลุ่มข้อความโดยอาศัยค่าความแตกต่างหรือค่าความคล้ายคลึงของวัตถุหรือข้อมูลที่นำมาเปรียบเทียบกัน ข้อมูลที่มีความคล้ายคลึงกันมากจะถูกกำหนดให้อยู่กลุ่มเดียวกันและข้อมูลที่มีความคล้ายคลึงกันน้อยจะถูกกำหนดให้อยู่คนละกลุ่มกัน ซึ่งวิธีการต่างๆนั้นจำเป็นต้องการอัลกอริทึมที่สามารถทำงานได้ดีกับข้อมูลที่สนใจหรือต้องการที่จะนำมาเป็นข้อมูลอินพุตของอัลกอริทึม เพื่อให้กระบวนการทำงานของการจัดแบ่งกลุ่มข้อความมีประสิทธิภาพดีและได้ผลลัพธ์ที่ถูกต้องในระดับที่ยอมรับได้ วิธีการและอัลกอริทึมที่นำเสนอในงานวิจัยนี้จึงมุ่งเน้นเกี่ยวกับ อัลกอริทึมสำหรับการจัดแบ่งกลุ่ม (Clustering Algorithms) เพื่อปรับปรุงและออกแบบอัลกอริทึมใหม่สำหรับการจัดแบ่งกลุ่มที่มีความเหมาะสมกับข้อมูลที่เป็นข้อความหรือเอกสารประเภทข่าวต่างๆ

การจัดกลุ่มแบบคลัสเตอริ่ง (Clustering) คือ กระบวนการการจัดกลุ่มของออบเจ็คต์เป็นกลุ่มต่างๆที่มีความคล้ายคลึงกันให้อยู่ในกลุ่มเดียวกัน และกลุ่มที่มีความแตกต่างกันให้อยู่ในกลุ่มที่แตกต่างกัน และคลัสเตอร์คือกลุ่มข้อมูลที่รวบรวมให้อยู่กลุ่มเดียวกันโดยใช้หลักการวัดความ

คล้ายคลึง ซึ่งการจัดกลุ่ม (Clustering) มีลักษณะการทำงานแบบเรียนรู้ด้วยตัวเอง (Unsupervised Learning) แตกต่างจากการจำแนกกลุ่ม (Classification) ที่มีลักษณะการทำงานแบบเรียนรู้โดยการสอน (Supervised Learning) การแบ่งกลุ่มสามารถแบ่งออกเป็น 2 วิธีการหลัก คือ Hierarchical clustering และ Partitioning clustering [1]

อัลกอริทึมที่ใช้หลักการของ Hierarchical clustering มีสองอัลกอริทึมคือ Agglomerative และ Divisive clustering อัลกอริทึมที่ใช้หลักการของ Partitioning clustering ได้แก่ K-Means, Competitive Learning Neural Network, Kohonen Neural Network และ Adaptive Resonance Theory เป็นต้น [1] ซึ่งอัลกอริทึมเหล่านี้จะรับข้อมูลเข้าเป็นตัวเลข (Numerical Feature Value) ดังนั้น การจัดกลุ่มข้อมูลที่ไม่ใช่ตัวเลข จำเป็นต้องทำการแปลงคุณสมบัติข้อมูล (Feature Type) ให้อยู่ในลักษณะของตัวเลขเพื่อให้สามารถนำข้อมูลเข้าอัลกอริทึมได้ ขั้นตอนการแปลงคุณสมบัติของออปเจ็ตที่เป็นเอกสารมาเป็นข้อมูลตัวเลข เรียกว่า การแทนเอกสาร (Document Representation) ซึ่งจะต้องแปลงข้อมูลที่เป็นเอกสารข้อความให้อยู่ในรูปแบบที่สามารถประมวลผลด้วยอัลกอริทึมทั่วไปได้ วิธีการหนึ่งที่ได้แผ่หลายให้การแทนเอกสารข้อความคือ วิธีการ Vector Space Model นำเสนอโดย G. Salton [2] มีลักษณะข้อมูลเป็น Matrix Array 2 Dimensions ประกอบด้วย TERM x Document แต่การทำ Document Representation โดยวิธีการ Vector Space Model อาจสร้างจำนวน Feature Vectors จำนวนมากทำให้เกิดปัญหา Very High Dimensional Vector Space มีผลกับการคำนวณที่ใช้เวลามากขึ้นของอัลกอริทึม และอาจทำให้ข้อมูลสูญเสียความหมายในตัวมันเองเนื่องจากการแปลงเอกสารข้อความมาเป็นอาเรย์ 2 มิติ ที่มี Feature type เป็นตัวเลข [6] ดังนั้นในปัจจุบันปัญหาการจัดกลุ่มได้ขยายปัญหาในการออกแบบอัลกอริทึมสำหรับการจัดกลุ่มข้อมูลที่เป็นตัวเลขมาเป็นปัญหาในการออกแบบอัลกอริทึมสำหรับการจัดกลุ่มที่มีลักษณะข้อมูลเป็นเอกสารข้อความหรือลักษณะข้อมูลที่ไม่ใช่ตัวเลข

ในช่วงเวลาที่ผ่านมามีโครงข่ายประสาทเทียมที่ได้รับการพัฒนาเพื่อให้สามารถรับข้อมูลที่เป็นข้อความได้ คือ Kohonen Neural Network พัฒนาเป็นโครงข่ายประสาทเทียมใหม่ชื่อ Text Processing Kohonen Neural Network และ Competitive Learning Neural Network พัฒนาเป็นโครงข่ายประสาทเทียมใหม่ชื่อ Text Processing Competitive Learning Neural Network ซึ่งโครงข่ายประสาทเทียมใหม่นี้สามารถจัดแบ่งกลุ่มข้อความได้โดยตรง โดยไม่มีแปลงรูปแบบข้อความนั้นมาเป็นตัวเลข ทำให้โครงข่ายประสาทเทียมใหม่นี้สามารถรับข้อมูลที่เป็นข้อความได้ การวัดความคล้ายคลึงสามารถทำได้สองแนวทางคือแนวทางแรกเป็นการวัดความแตกต่าง ถ้าเปรียบเทียบกับทุกๆโหนดแล้วโหนดใดมีความแตกต่างกันน้อยที่สุด โหนดนั้นคือโหนดที่ชนะ และ

แนวทางที่สองเป็นการวัดความคล้ายคลึง ถ้าเปรียบเทียบกับทุกๆโหนดแล้วโหนดใดมีความคล้ายคลึงมากที่สุด โหนดนั้นคือโหนดที่ชนะ [3]

A Text Adaptive Resonance Theory Neural Network เป็นโครงข่ายประสาทเทียมที่ได้รับการปรับแต่งการทำงานของอัลกอริทึมขึ้นเพื่อขยายความสามารถของ Adaptive Resonance Theory Neural Networks ให้สามารถรับข้อมูลเข้าเป็นข้อความ (Textual Dataset) ได้โดยตรง โดยไม่ต้องผ่านขั้นตอนการแปลงรูปแบบข้อมูลให้เป็นตัวเลขหรือข้อมูลเชิงปริมาณ (Quantitative Feature) และขั้นตอนการปรับแต่งตามความสอดคล้อง (Adaptive Resonance) ด้วยอาศัยค่าพารามิเตอร์ Vigilance ในการควบคุมระดับค่าคล้ายคลึงของ Pattern ที่ทำการ Clustering ได้ ด้วยหลักการทำงานแบบเรียนรู้ด้วยตัวเองและหลักการวัดความคล้ายคลึงของซิมโบลิคออบเจ็ต (Similarity Measure for Symbolic Object) [5] โครงข่ายประสาทเทียมที่ได้ปรับแต่งอัลกอริทึมนี้สามารถจัดกลุ่มข้อมูลที่มีคุณสมบัติเป็นข้อความได้เป็นอย่างดี

1.2 จุดประสงค์ของงานวิจัย

งานวิจัยนี้มีจุดประสงค์เพื่อพัฒนาวิธีการจัดแบ่งกลุ่มเอกสารด้วยโครงข่ายประสาทเทียมที่สามารถรับค่าข้อมูลเข้าที่เป็น Qualitative Value ได้โดยตรงโดยไม่ต้องแปลงค่าข้อมูลให้อยู่ในรูปแบบตัวเลข เพื่อใช้เป็นแนวทางในการพัฒนาและปรับปรุงอัลกอริทึม ในการจัดกลุ่มเอกสารที่เป็นข้อความ เช่น ข่าวการเมือง ข่าวเศรษฐกิจ ข่าวกีฬา เป็นต้น ให้มีประสิทธิภาพดีสามารถทำงานกับข้อมูลที่เป็นข้อความได้ดีและมีความถูกต้องในระดับที่ยอมรับได้

1.3 แผนการดำเนินงานวิจัย

1.3.1 ขั้นตอนการดำเนินงานวิจัย

1. ศึกษาผลงานวิจัยและเอกสารทางวิชาการที่เกี่ยวข้องกับการวิจัย
2. กำหนดหัวข้อ เป้าหมาย วัตถุประสงค์ และขอบเขตการทำวิทยานิพนธ์
3. ศึกษาทฤษฎีและหลักการที่เกี่ยวข้อง
4. วิเคราะห์ ออกแบบและทดลองอัลกอริทึมใหม่
5. พัฒนาโปรแกรมและทดสอบอัลกอริทึมกับข้อมูลที่กำหนด
6. วิเคราะห์ผล เปรียบเทียบ และสรุปผล
7. จัดทำเอกสารประกอบวิทยานิพนธ์

1.3.2 ระยะเวลาที่คาดว่าจะใช้ในแต่ละขั้นตอน

ขั้นตอนการทำงาน	ระยะเวลาที่ใช้ (เดือนที่)											
	1	2	3	4	5	6	7	8	9	10	11	12
ศึกษาผลงานวิจัยและเอกสารทางวิชาการ												
กำหนดหัวข้อ เป้าหมาย วัตถุประสงค์ และขอบเขตการทำวิทยานิพนธ์												
ศึกษาทฤษฎีและหลักการที่เกี่ยวข้อง												
วิเคราะห์ออกแบบ และทดลองอัลกอริทึมใหม่												
พัฒนาโปรแกรมและทดสอบอัลกอริทึมกับข้อมูลที่กำหนด												
วิเคราะห์ผล เปรียบเทียบ และสรุปผล												
จัดทำเอกสารประกอบวิทยานิพนธ์												

1.4 เครื่องมือที่ใช้ในการวิจัย

1. ฮาร์ดแวร์ประกอบด้วย PC Computer CPU Pentium 4 , Clock Rate 1.5 GHz , RAM 256 MB, HARD DISK 20 GB, MONITER 15 นิ้ว, KEYBORD, VGA CARD และ เครื่องพิมพ์อย่างละหนึ่งชุด
2. ระบบปฏิบัติการคือ MS-Windows XP Professional
3. ซอฟต์แวร์ที่ใช้คือ MATLAB 5.0

1.5 ประโยชน์ที่คาดว่าจะได้รับจากงานวิจัยครั้งนี้

1. ความรู้ความเข้าใจหลักการจัดแบ่งกลุ่มเอกสาร (Document Clustering)
2. ความรู้ความเข้าใจการทำงานของโครงข่ายประสาทเทียมแบบ Unsupervised Learning
3. แนวทางในการพัฒนาและปรับปรุงอัลกอริทึม ซึ่งคาดว่าจะทำให้ได้อัลกอริทึมใหม่

บทที่ 2
หลักการและทฤษฎีที่เกี่ยวข้อง

2.1 หลักการจัดแบ่งกลุ่มเอกสาร (Document Clustering Concepts)

การจัดแบ่งกลุ่มเอกสาร คือ กระบวนการวิเคราะห์ข้อมูลเอกสารเพื่อจัดกลุ่มของเอกสารตามคุณลักษณะความคล้ายคลึงหรือความแตกต่างของคุณลักษณะของเอกสาร ในที่นี้คุณลักษณะของเอกสารหมายถึง ชื่อเรื่องของเอกสารและคำสำคัญของเอกสาร [6] ซึ่งขั้นตอนที่สำคัญในการจัดกลุ่มเอกสารคือขั้นตอนของอัลกอริทึมที่ใช้ในการแบ่งกลุ่ม เช่น K-Mean algorithm, Agglomerative Clustering และ Neural Networks ขั้นตอนการแบ่งกลุ่มแสดงไว้ในรูปที่ 2.1

รูปที่ 2.1 ขั้นตอนการแบ่งกลุ่มเอกสาร

ข้อมูลเอกสาร (Document) จะถูกตัดคำโดยการตัดกลุ่มคำที่เรียกว่า Stop Words ออกจากข้อมูลเอกสารก่อน เมื่อตัด Stop Words เรียบร้อยแล้ว จากนั้นหารากของคำโดยใช้เทคนิคของ Streaming Words หลังจากขั้นตอนนี้ สิ่งที่ได้คือกลุ่มคำที่ใช้เป็น Features ของ Document ในที่นี้ Features ของ Document คือ Terms จากนั้นจึงสร้าง Matrix ที่ประกอบด้วย Doc x Terms ซึ่งการสร้าง Matrix นี้เป็นวิธีการทั่วไปของ Document Representation ในกระบวนการทำงานของการจัดแบ่งกลุ่มเอกสาร (Document Clustering) ส่วนที่สำคัญของการจัดแบ่งกลุ่มเอกสารสามารถแบ่งออกเป็น 3 ส่วน คือ

2.1.1 Document Representations

2.1.2 Similarity Measures

2.1.3 Clustering Algorithms

2.1.1. Document Representation

ในการคำนวณหาความคล้ายคลึงของเอกสาร จำเป็นต้องกำหนด Attribute ของเอกสารเพื่อนำมาคำนวณในสมการหาค่าความคล้ายคลึง (Similarity Measures) โดย Attribute ของเอกสารที่ถูกเลือกเพื่อใช้ในการคำนวณ เรียกว่า Measurement ซึ่งอาจเป็น Attribute ต่างๆ เช่น Title ของเอกสาร Author ของเอกสาร และ Keywords ของเอกสาร ซึ่งลักษณะข้อมูลใน Measurement ของเอกสารอยู่ในลักษณะข้อมูลเชิงคุณภาพ (Qualitative Feature)

วิธีการทั่วไปสำหรับการนำข้อมูลเชิงคุณภาพ (Qualitative Feature) จาก Measurement ของเอกสารมาแปลงเป็นข้อมูลเชิงปริมาณ (Quantitative Feature) มีหลายวิธีการ ในที่นี้จะกล่าวถึงวิธีการที่เป็นที่รู้จักและเป็นที่ใช้กันแพร่หลายทั้งหมด 3 วิธีการ ดังนี้ [6],[26]

2.1.1.1 Binary Representation

วิธีการแบบ Binary นี้จะสามารถบอกได้เพียงบอกแต่ละเอกสารมีคำที่เป็น Index Terms ปรากฏในเอกสารหรือไม่ ด้วยการแทนด้วย 0 และ 1 โดย 0 หมายถึงไม่มีคำนี้ปรากฏในเอกสาร และ 1 หมายถึงมีคำนี้ปรากฏในเอกสาร ดังเช่นในตัวอย่าง

ในตัวอย่างนี้มีเอกสาร A ถึงเอกสาร F แต่ละเอกสารมีคำที่เป็น Index Terms ดังข้างล่างนี้

Doc A	care, cat, persian
Doc B	care, care, care, cat, cat, cat, persian, persian, persian
Doc C	cat, cat, cat, cat, cat, cat, cat, cat, cat
Doc D	care, cat, dog, dog, dog, dog, dog, dog, persian
Doc E	care, cat, dog
Doc F	care

Index Terms = <care,cat,dog,persian>

ดังนั้นเราแทนเอกสารแบบไบนารี (Binary Document Representation) จะได้ผลลัพธ์ดังนี้

Doc A	= <1, 1, 0, 1>
Doc B	= <1, 1, 0, 1>
Doc C	= <0, 1, 0, 0>
Doc D	= <1, 1, 1, 1>
Doc E	= <1, 1, 1, 0>
Doc F	= <1, 0, 0, 0>

ซึ่งเราสามารถสร้าง Binary Matrix ได้ดังนี้

	care	cat	dog	persian
Doc A	1	1	0	1
Doc B	1	1	0	1
Doc C	0	1	0	0
Doc D	1	1	1	1
Doc E	1	1	1	0
Doc F	1	0	0	0

วิธีการแบบ Binary มีข้อเสียในเรื่องการไม่มีน้ำหนักของคำในเอกสารและความถี่ของคำที่ปรากฏในเอกสาร การแทนเอกสารด้วยวิธีนี้จึงขาดประสิทธิภาพในสองปัจจัยดังกล่าว

2.1.1.2 Term Frequency (TF)

วิธีการแบบ TF นี้สามารถบอกความถี่ของคำที่ปรากฏในเอกสารได้ จึงมีประสิทธิภาพดีกว่าวิธีการแบบ Binary โดยวิธีการแบบ TF เป็นการนับแต่ละคำที่ปรากฏในเอกสารว่าจะจำนวนที่ปรากฏกี่ครั้ง โดยยังคงใช้ 0 แทนความหมายว่าไม่มีคำปรากฏในเอกสาร คล้ายกับวิธีการแบบ Binary ดังตัวอย่าง

ในตัวอย่างนี้มีเอกสาร A ถึงเอกสาร F แต่ละเอกสารมีคำที่เป็น Index Terms ดังข้างล่างนี้

Doc A	care, cat, persian
Doc B	care, care, care, cat, cat, cat, persian, persian, persian
Doc C	cat, cat, cat, cat, cat, cat, cat, cat, cat
Doc D	care, cat, dog, dog, dog, dog, dog, dog, persian
Doc E	care, cat, dog
Doc F	care

Index Terms = <care,cat,dog,persian>

ดังนั้นเราแทนเอกสารแบบไบนารี (TF) จะได้ผลลัพธ์ดังนี้

TF_{docA} = <1, 1, 0, 1>

TF_{docB} = <3, 3, 0, 3>

TF_{docC} = <0, 9, 0, 0>

TF_{docD} = <1, 1, 6, 1>

TF_{docE} = <1, 1, 1, 0>

TF_{docF} = <1, 0, 0, 0>

ซึ่งเราสามารถสร้าง Term-Frequency Matrix ได้ดังนี้

	care	cat	dog	persian
Doc A	1	1	0	1
Doc B	3	3	0	3
Doc C	0	9	0	0
Doc D	1	1	6	1
Doc E	1	1	1	0
Doc F	1	0	0	0

2.1.1.3 Term Frequency * Inverted Document Frequency (TF*IDF)

วิธีการแบบ TF*IDF เป็นการหาจำนวนความถี่ของคำที่ปรากฏในเอกสารร่วมกับการหาน้ำหนักของคำที่ปรากฏในเอกสาร เนื่องจากคำที่เป็น Index Terms แต่ละคำมีความสำคัญแตกต่างกัน จึงต้องให้น้ำหนักของคำเหล่านี้ การหาน้ำหนักของคำสามารถหาได้ดังสมการนี้

$$IDF_{D,t_j} = \log_2\left(N_D / n_{D,t_j}\right) \quad (2.1)$$

โดยกำหนดให้

N_D = จำนวนทั้งหมดของเอกสาร D

n_{D,t_j} = จำนวนเอกสารที่มี index term t_j

หลังจากหาน้ำหนักของคำ (IDF) ได้เรียบร้อยแล้วให้มีคูณกับ Term-Frequency ดังสมการ

$$W_{d_i,t_j} = TF_{d_i,t_j} \times IDF_{D,t_j} \quad (2.2)$$

ในตัวอย่างนี้มีเอกสาร A ถึงเอกสาร F แต่ละเอกสารมีคำที่เป็น Index Terms ดังข้างล่างนี้

Doc A	care, cat, persian
Doc B	care, care, care, cat, cat, cat, persian, persian, persian
Doc C	cat, cat, cat, cat, cat, cat, cat, cat, cat
Doc D	care, cat, dog, dog, dog, dog, dog, dog, persian
Doc E	care, cat, dog
Doc F	care

หาจำนวนคำที่ปรากฏในเอกสารซึ่งจะได้ Term-Frequency Vector ดังนี้

TF_{docA} = <1, 1, 0, 1>

TF_{docB} = <3, 3, 0, 3>

TF_{docC} = <0, 9, 0, 0>

TF_{docD} = <1, 1, 6, 1>

TF_{docE} = <1, 1, 1, 0>

TF_{docF} = <1, 0, 0, 0>

หลังจากนั้น หาน้ำหนักของแต่ละคำ

$$IDF_{D,t_j} = \langle \log_2(6/5), \log_2(6/5), \log_2(6/2), \log_2(6/3) \rangle$$
$$= \langle 0.26, 0.26, 1.58, 1.00 \rangle$$

หลังจากหาน้ำหนักของคำ (IDF) ได้เรียบร้อยแล้วให้มีคูณกับ Term-Frequency

$$W_{docA} = \langle 1 \times 0.26, 1 \times 0.26, 0 \times 1.58, 1 \times 1.00 \rangle$$
$$= \langle 0.26, 0.26, 0.00, 1.00 \rangle$$
$$W_{docB} = \langle 3 \times 0.26, 3 \times 0.26, 0 \times 1.58, 3 \times 1.00 \rangle$$
$$= \langle 0.79, 0.79, 0.00, 3.00 \rangle$$
$$W_{docC} = \langle 0 \times 0.26, 9 \times 0.26, 0 \times 1.58, 0 \times 1.00 \rangle$$
$$= \langle 0.00, 2.37, 0.00, 0.00 \rangle$$
$$W_{docD} = \langle 1 \times 0.26, 1 \times 0.26, 6 \times 1.58, 1 \times 1.00 \rangle$$
$$= \langle 0.26, 0.26, 9.51, 1.00 \rangle$$
$$W_{docE} = \langle 1 \times 0.26, 1 \times 0.26, 1 \times 1.58, 0 \times 1.00 \rangle$$
$$= \langle 0.26, 0.26, 1.58, 0.00 \rangle$$
$$W_{docF} = \langle 1 \times 0.26, 0 \times 0.26, 0 \times 1.58, 0 \times 1.00 \rangle$$
$$= \langle 0.26, 0.00, 0.00, 0.00 \rangle$$

ซึ่งเราสามารถสร้าง Term-Frequency * Inverted-Document-Frequency Matrix ได้ดังนี้

	care	cat	dog	persian
Doc A	0.26	0.26	0.00	1.00
Doc B	0.79	0.79	0.00	3.00
Doc C	0.00	2.37	0.00	0.00
Doc D	0.26	0.26	9.51	1.00
Doc E	0.26	0.26	1.58	0.00
Doc F	0.26	0.00	0.00	0.00

อย่างไรก็ตาม วิธีการแปลงข้อมูลเชิงคุณภาพมาเป็นข้อมูลเชิงปริมาณสำหรับการจัดแบ่งเอกสาร อาจก่อให้เกิดปัญหา Very High Dimensional Vector Spaces ได้ ถ้า Index Terms มี

จำนวนมากขึ้น เช่น Collection ของเอกสารชุดหนึ่ง จำนวน 1,000 เอกสาร เมื่อผ่านขั้นตอนก่อนการประมวลผล (Preprocessing Document) อาจทำให้เกิดจำนวน Index Terms มากถึง 30-50 คำ ซึ่ง Index Terms ที่มีจำนวนมากนี้ จะมีผลกระทบกับขั้นตอนของ Document Representation ทำให้เมื่อผ่านขั้นตอนการทำ Document Representation ดังกล่าว จำนวน Feature Vector จะมีจำนวนมากขึ้นตามไปด้วย นั้นคือผลลัพธ์ส่งท้ายเมื่อผ่านขั้นตอน Document Representation จะมีขนาด Feature เป็น 30-50 Dimensions

2.1.2. Similarity Measures

Similarity Measures เป็นพื้นฐานสำหรับการกำหนดกลุ่มของข้อมูล โดยวัดค่าความคล้ายคลึงระหว่าง 2 patterns โดยต้องให้ทุก Patterns มี Feature space เหมือนกัน ในการคำนวณหาความคล้ายคลึงของ Pattern ที่มี Feature type เป็น continuous เราสามารถใช้สูตร Euclidean Distance เพื่อหาค่าความคล้ายคลึงนี้ [5],[21],[28]

$$d(i,j) = \sqrt{\left|x_{i1} - x_{j1}\right|^2 + \left|x_{i2} - x_{j2}\right|^2 + \cdots + \left|x_{ip} - x_{jp}\right|^2} \qquad (2.3)$$

กำหนดให้

$i = \langle X_{i1}, X_{i2}, \ldots X_{ip} \rangle$

$j = \langle X_{j1}, X_{j2}, \ldots, X_{jp} \rangle$

p = Feature ของ Feature Vector i และ j

อย่างไรก็ตาม สูตร Euclidean Distance สามารถใช้ได้เฉพาะกับข้อมูลประเภท Quantitative ที่อธิบายข้อมูลด้วยจำนวนตัวเลขหรือบอกถึงค่าตัวเลขปริมาณของสิ่งที่สนใจ ดังนั้นการหาความแตกต่างของข้อมูลที่ไม่ใช่ข้อมูลจำนวนตัวเลขด้วยสูตรนี้ จึงใช้ไม่ได้ ตัวอย่างข้อมูลเหล่านี้ เช่น เอกสารบทความ ชื่อหัวข้อข่าว และคำสำคัญของเอกสารต่างๆ เพราะฉะนั้นวิธีการทั่วไปสำหรับการจัดแบ่งกลุ่มข้อมูลเอกสาร จึงต้องทำการ Represent ข้อมูลให้อยู่ลักษณะข้อมูลเชิงปริมาณหรือจำนวนตัวเลข (Quantitative) ดังวิธีการในข้อ 2.1.1 ที่ได้กล่าวมาแล้ว

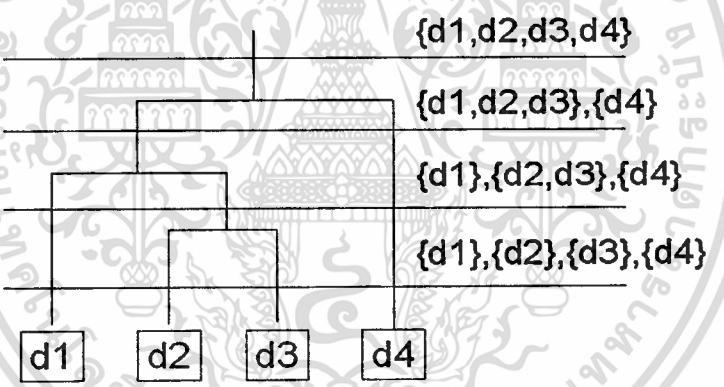
2.1.3. Clustering Algorithms

Clustering Algorithms จะทำการแบ่งกลุ่มข้อมูลออกเป็นกลุ่มๆ ที่เราเรียกว่า Cluster ข้อมูลที่คุณลักษณะคล้ายคลึงกันจะถูกจัดให้อยู่กลุ่มเดียวกัน และข้อมูลที่มีคุณลักษณะแตกต่าง

กันจะถูกจัดให้อยู่คนละกลุ่ม อัลกอริทึมที่ใช้ในการแบ่งกลุ่มจะอาศัยหลักการ Similarity Measures หรือ Dissimilarity Measures เพื่อระบุว่าข้อมูลนั้นๆมีความคล้ายคลึงหรือแตกต่างกันเท่าไร อัลกอริทึมสำหรับแบ่งกลุ่มข้อมูลที่มีอยู่สามารถแบ่งออกเป็น 2 แนวทาง คือ Hierarchical Clustering และ Partitional Clustering [21],[28]

2.1.3.1. Hierarchical Clustering

เทคนิค Hierarchical Clustering คือการจัดรวมกลุ่มข้อมูลเป็นชั้นๆตามลำดับชั้น จากข้อมูลเพียงคลัสเตอร์เดียวที่ประกอบไปด้วยชุดของข้อมูลทั้งหมดที่ระดับบนสุด เมื่อผ่านขั้น ตอนการจัดกลุ่มก็จะได้เป็นคลัสเตอร์ย่อยๆ ไปจนกระทั่งแต่ละคลัสเตอร์จะมีเฉพาะเพียงข้อมูลชุดเดียวที่ระดับล่างสุด ซึ่งผลที่ได้จากใช้งานอัลกอริทึมการจัดกลุ่มแบบ Hierarchical นี้สามารถแสดงได้เป็นโครงแบบต้นไม้ที่เรียกว่า Dendogram ดังรูปที่ 2.1 แสดงแผนภาพ Dendogram ของ Hierarchical Clustering ในการสร้างลำดับขั้นของ Hierarchical Clustering นั้นเราสามารถจำแนกได้เป็นสองวิธี คือวิธีแบบ Agglomerative และ วิธีแบบ divisive ซึ่งวิธีแบบ Agglomerative เป็นวิธีที่ใช้กันมากใน Hierarchical Clustering [2]



รูปที่ 2.2 แสดงแผนภาพ Dendogram ของ Hierarchical Clustering

Hierarchical Agglomerative Clustering

วิธีนี้จะเริ่มจากชุดของข้อมูลเดียวในแต่ละคลัสเตอร์ แล้วที่แต่ละขั้นจะทำการรวม (merge) คลัสเตอร์ที่เหมือนกันมากที่สุดสองคลัสเตอร์เข้าด้วยกัน ซึ่งจะทำซ้ำๆในขั้นตอนนี้จนกระทั่งได้จำนวนของคลัสเตอร์ที่น้อยที่สุดแล้ว หรือถ้าต้องการให้ได้ลำดับขั้นที่สมบูรณ์ก็ต้องทำขั้นตอนนี้ไปจนกระทั่งเหลือคลัสเตอร์เพียงหนึ่งคลัสเตอร์ อัลกอริทึมสำหรับหาความคล้ายคลึงระหว่างแต่ละคู่คลัสเตอร์มี 2 วิธีคือ Single-Link และ Complete-Link ในอัลกอริทึมแบบ Single-

Link ค่าความแตกต่าง (Distance) ระหว่างสองคลัสเตอร์ คือค่าความแตกต่าง (Distance) ที่น้อยที่สุด (Minimum) ระหว่าง 2 Patterns จาก 2 คลัสเตอร์ ในขณะที่อัลกอริทึมแบบ Complete-Link ค่าความแตกต่าง (Distance) ระหว่างสองคลัสเตอร์ คือค่าความแตกต่าง (Distance) ที่มากที่สุด (Maximum) ระหว่าง 2 Patterns จาก 2 คลัสเตอร์

ขั้นตอนการทำงานของ Hierarchical Agglomerative Clustering Algorithm

ขั้นตอนที่ 1 คำนวณหา Distance ระหว่างแต่ละคู่ Patterns จากคลัสเตอร์ เก็บค่า Distance ลงใน Proximity Matrix

ขั้นตอนที่ 2 หาคู่ที่มีค่าความคล้ายคลึงมากที่สุดจาก Proximity Matrix แล้วรวม 2 คลัสเตอร์เป็น 1 คลัสเตอร์ จากนั้น Update ค่า Distance ใน Proximity Matrix

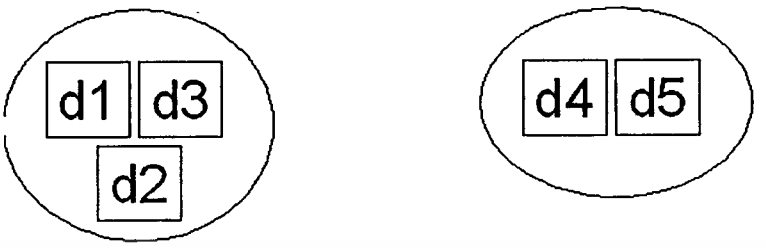
ขั้นตอนที่ 3 ถ้า Patterns ทั้งหมดจากทุกคลัสเตอร์ ถูกรวมเป็น 1 คลัสเตอร์ เรียบร้อยแล้ว ให้หยุดการทำงาน นอกจากนี้ให้กลับไปทำงานที่ขั้นตอนที่ 2

2.1.3.2 Partitional Clustering

Partitional Clustering ทำงานโดยระบุคลัสเตอร์ขึ้นมาจำนวนหนึ่งก่อนพร้อมๆกัน โดยที่ในแต่ละคลัสเตอร์จะประกอบไปด้วยข้อมูลทั้งหมดที่กระจายกันไปอยู่ในแต่ละคลัสเตอร์ แล้วทำการปรับ (update) คลัสเตอร์ซ้ำๆด้วยฟังก์ชันที่จะทำให้ได้คลัสเตอร์เหลือจำนวนน้อยที่สุด การปรับ คลัสเตอร์ไม่ใช่การนำคลัสเตอร์ที่มีอยู่มารวมกัน แต่เกิดจากปรับเปลี่ยนโยกย้ายของข้อมูลระหว่าง คลัสเตอร์เหล่านั้น ซึ่งอัลกอริทึมแบบ Partitional Clustering ไม่ได้มีการสร้าง Dendogram เหมือนวิธี Hierarchical Clustering โดยที่อัลกอริทึมแบบ Partitional Clustering ที่รู้จักกันดีเช่นอัลกอริทึมในกลุ่ม K-means เป็นต้น [21],[28]

อัลกอริทึม K-means จะมีการทำงานโดยริ่มจากการกำหนดคลัสเตอร์แบบสุ่มขึ้นมาจำนวน k คลัสเตอร์ (ซึ่งจะกำหนดจุดศูนย์กลางของแต่ละคลัสเตอร์ขึ้นมาพร้อมกันด้วย) ต่อจากนั้นให้ข้อมูลแต่ละตัวไปเป็นสมาชิกอยู่ในคลัสเตอร์ที่อยู่ใกล้กับมันมากที่สุด โดยการวัดระยะห่างระหว่างตัวมันกับจุดศูนย์กลางของคลัสเตอร์ ตัวอัลกอริทึมจะทำการคำนวณไปเรื่อยๆจนครบจำนวนของข้อมูลทุกตัวและทุกคลัสเตอร์ เพื่อให้ข้อมูลไปเป็นสมาชิกอยู่ในคลัสเตอร์ใดคลัสเตอร์หนึ่ง เมื่อเสร็จสิ้นขั้นตอนนี้แล้วอัลกอริทึมก็จะทำการหาจุดศูนย์กลาง ของทุกๆคลัสเตอร์ ที่มีข้อมูลอยู่ใหม่อีกครั้ง โดยจะหาจากค่าเฉลี่ยของระยะห่างจากจุดศูนย์กลางกับสมาชิกทุกตัวในคลัสเตอร์ ซึ่งจำนวนของคลัสเตอร์อาจจะมีการเปลี่ยนแปลงได้ในขั้นตอนนี้ โดยจำนวนจะลดลงหรือคงที่เท่านั้นและอัลกอริทึมจะวนกลับ ไปเริ่มทำการคำนวณระยะห่างระหว่างข้อมูลแต่ละตัวกับจุดศูนย์กลางของทุกๆคลัสเตอร์ที่ได้ใหม่นี้อีกครั้ง และจะทำซ้ำๆขั้นตอนเหล่านี้ไปจนกระทั่งจำนวน

ของข้อมูลในแต่ละคลัสเตอร์ไม่มีการเปลี่ยนแปลง จะเห็นได้ว่าอัลกอริทึม K-means เหมาะสมกับโครงสร้าง ของคัลสเตอร์ที่เป็นรูปทรงกลม นอกจากนี้การกำหนดคลัสเตอร์เริ่มต้นก็เป็นสิ่งสำคัญมากของอัลกอริทึม K-means ที่จะทำให้ผลของการจัดกลุ่มมีคุณภาพ



รูปที่ 2.2 แสดงแผนภาพ Cluster ของ Partitional Clustering

Criterion ที่ใช้ใน K-Mean Algorithm คือ Squared Error ดังสมการ

$$E^2(X,C) = \sum_{i=1}^{N}\sum_{j=1}^{K}\left\|x_i^j - c_j\right\|^2 \qquad (2.4)$$

กำหนดให้ x_i^j คือ ข้อมูลที่ i ใน Cluster ที่ j และ c_j คือ Centroid ของ Cluster ที่ j

ขั้นตอนการทำงานของ K-mean Algorithm

ขั้นตอนที่ 1 กำหนดค่า K เป็นค่าเริ่มต้นของจำนวนกลุ่ม Cluster

ขั้นตอนที่ 2 กำหนดให้ข้อมูลที่มีค่า Squared Error น้อยที่สุดเมื่อเปรียบเทียบกับ Centroid ของ Cluster ที่ j ให้ assign อยู่ใน Cluster นั้น

ขั้นตอนที่ 3 Update ค่า Mean (Centroid) ของ Cluster ที่ j

ขั้นตอนที่ 4 ถ้ายังไม่ถึงจุด Convergence ให้กลับที่ขั้นตอนที่ 2, Convergence สามารถเป็นได้คือ จำนวน Loop การทำงาน

อีกอัลกอริทึมหนึ่งที่น่ากล่าวถึงคือ Fuzzy Clustering มีความแตกต่างอย่างเห็นได้ชัดคือ Clustering Algorithm ทั่วไปจะระบุ Pattern ใด Pattern หนึ่งให้อยู่ใน Cluster ใด Cluster หนึ่ง (one and only one cluster) เช่น K-Mean Clustering แต่สำหรับ Fuzzy Clustering จะระบุ Pattern ใด Pattern หนึ่งให้อยู่หลาย Cluster โดยมีค่า Membership Function ที่ใช้บอกถึงระดับความสัมพันธ์ระหว่าง Pattern และ Cluster [21]

ขั้นตอนการทำงานของ Fuzzy Clustering Algorithm

ขั้นตอนที่ 1 กำหนดค่า K เป็นค่าเริ่มต้นของจำนวนกลุ่ม Cluster และสร้าง Membership Matrix (U) ที่ประกอบด้วย N x K เมื่อ N คือ จำนวน Objects หรือ Patterns และ K คือ จำนวน Cluster

สมาชิกใน Membership Matrix (U) แทนด้วย u_{ij} หมายถึงค่า Grade of Membership ของ Pattern X_i ในคลัสเตอร์ C_j ค่า u_{ij} มีค่าอยู่ระหว่าง [0,1]

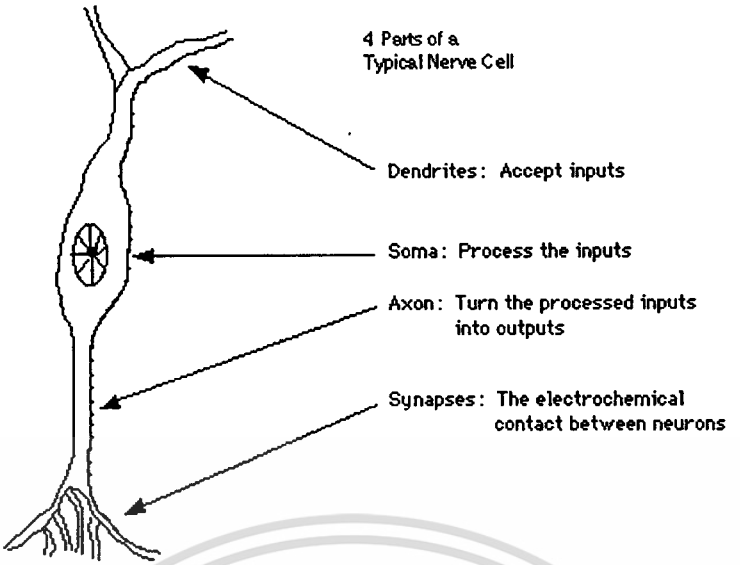
ขั้นตอนที่ 2 กำหนดให้ข้อมูลที่มีค่า Squared Error น้อยที่สุดเมื่อเปรียบเทียบกับ Centroid ของ Cluster ที่ j ให้ assign อยู่ใน Cluster นั้น โดยมีสูตรดังนี้

$$E^2(X,U) = \sum_{i=1}^{N}\sum_{k=1}^{K} u_{ij}\|x_i - c_k\|^2;\ c_k = \sum_{i=1}^{N} u_{ik}x_i \qquad (2.5)$$

ขั้นตอนที่ 3 ทำขั้นตอนที่ 2 ซ้ำ จนกระทั่ง u_{ij} ของ Membership Matrix U ไม่มีการเปลี่ยนแปลง

2.2 โครงข่ายประสาท (Neural Network)

โครงข่ายประสาทภายในสมองของมนุษย์ เป็นสิ่งสำคัญที่ธรรมชาติให้มนุษย์มาโดยที่ความพิเศษของโครงข่ายประสาทเทียมภายในสมองมนุษย์นั้นคือมันมีความสามารถ "คิดหาเหตุผล" ซึ่งทำให้มนุษย์เป็นสัตว์ที่ฝึกและเรียนรู้ได้ดีกว่าสัตว์ใดๆ ก็เพราะว่ามนุษย์มีสมองที่น่าอัศจรรย์นั้นเอง ซึ่งสมองของมนุษย์ประกอบด้วย เซลประสาทที่เรียกว่า นิวรอน (neuron) ที่เชื่อมต่อซึ่งกันและกันอย่างหนาแน่น อยู่ภายในสมองมนุษย์ราวๆ 10 พันล้านนิวรอน [7],[8],[22] ดังนั้นการใช้หลายๆนิวรอนประมวลผลพร้อมๆกัน ทำให้สมองมนุษย์สามารถปฏิบัติงานที่ซับซ้อนได้เร็วกว่าการทำงานของเครื่องคอมพิวเตอร์ในปัจจุบันมาก โดยที่แต่ละนิวรอนจะมีโครงสร้างง่ายๆ ที่ ประกอบด้วยตัวเซล หรือ soma และมีเส้นประสาทแยกออกไปสองกิ่งหลักๆ ที่เรียกว่า dendrite แบบหนึ่ง และเส้นประสาทเดี่ยวยาวที่เรียกว่า axon แบบหนึ่ง ในตัวเซลจะ มี นิวเคลียส์ (nucleus) อยู่ตรงกลาง ซึ่งบรรจุไปด้วยข้อมูล (information) เกี่ยวกับพันธุกรรม และมีของเหลวที่เรียกว่า plasma บรรจุห่อหุ้มอยู่ ตามรูป 2.3 คือแผนภาพวาดของโครงข่ายประสาทในสมองมนุษย์ [21],[22]



รูปที่ 2.3 แสดงแผนภาพวาดของ โครงข่ายประสาทในสมองมนุษย์

นิวรอนแต่ละตัวจะรับสัญญาณ (impulse) จากนิวรอนตัวอื่นๆ ผ่านทาง dendrite (receiver) ของมันและส่งผ่านสัญญาณที่กำเนิดขึ้นโดยตัวเซลของมันเองไปตามเส้นประสาทแอกซอน (Axon) ซึ่งสัญญาณจะผ่านเส้นประสาทนี้ไปยังจุดปลายทาง ที่เรียกว่าไซแนปส์ (Synapse) โดยที่ไซแนปส์ คือ จุดที่เชื่อมต่อระหว่างสองนิวรอน (axon ของนิวรอนหนึ่ง กับเดนไดรท์ (Dendrite) ของอีกนิวรอนหนึ่ง) เมื่อสัญญาณ ไปถึงจุดปลายของตัวไซแนปส์ แล้วก็จะเกิดปฏิกิริยาเคมีขึ้นมาและต่อจากนั้น ปฏิกิริยาเคมีก็จะถูกแปลงเป็นสัญญาณไฟฟ้า และถูกปล่อยออกมาโดยตัวที่เรียกว่า neurotransmitter ซึ่งสัญญาณจะ แพร่ข้ามช่องว่างไซแนปส์ (ไปยังเดนไดรท์ (Dendrite) ของนิวรอนอื่นต่อไป) และจะทำให้เกิดผลกระทบขึ้นกับไซแนปส์ โดยที่ผลกระทบที่เกิดกับไซแนปส์ สามารถจะถูกปรับได้โดยสัญญาณ (การเพิ่มขึ้นหรือลดลงของศักย์ไฟฟ้า) ที่ผ่านตัวมันดังนั้นตัวไซแนปส์ จึงสามารถเรียนรู้จากกิจกรรมที่มันเข้าไปมีส่วนร่วมได้ โดยจะขึ้นอยู่กับพฤติกรรมในอดีตที่ผ่านมาด้วย ดังนั้นการที่ความจำของมนุษย์ ซึ่งสามารถตอบสนองต่อการระลึกหรือจดจำได้นั้น เกิดจากการที่ตัวนิวรอนมีความสามารถที่จะเรียนรู้โดยผ่านประสบการณ์ ที่มันเคยมีส่วนร่วมมานั่นเอง ซึ่งโดยที่การเรียนรู้คือพื้นฐานและคุณลักษณะที่จำเป็นของโครงข่ายประสาทในสมองของมนุษย์ และด้วยความสามารถที่มันเรียนรู้ได้นี้เอง ได้เกิดความพยายาม ที่จะจำลองการทำงานของโครงข่ายประสาทนี้มาใช้ในคอมพิวเตอร์ ที่เรียกกันว่า "โครงข่ายประสาทเทียม (Artificial Neural Network)"

2.3 โครงข่ายประสาทเทียม (Artificial Neural Network)

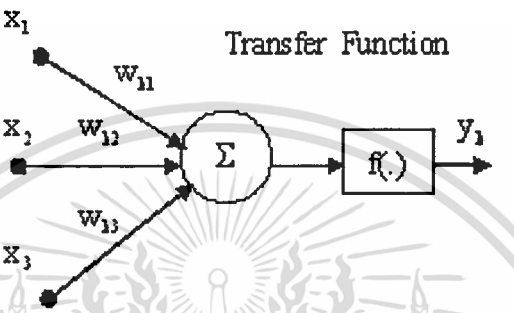
วิธีการจัดแบ่งกลุ่มโดยใช้โครงข่ายประสาทเทียม เป็นวิธีการเพื่อปรับปรุงความเหมาะหรือความสอดคล้องของกลุ่มข้อมูลที่ทำการวิเคราะห์และหลักการที่เกี่ยวกับคณิตศาสตร์ (Mathematical Model) วิธีการของโครงข่ายประสาทเทียมสำหรับการแบ่งกลุ่มที่มีอยู่ในขณะนี้ส่วนใหญ่ใช้ Winner-Take-All Learning Rule ในธรรมชาติเซลประสาทของสิ่งมีชีวิตมีหลายประเภทแล้วแต่หน้าที่ของมัน เซลประสาทในตัวของคนเราก็เช่นกัน มีอยู่หลายประเภทตามตำแหน่งและหน้าที่ของมัน เช่น เซลประสาทกล้ามเนื้อ เซลประสาทในสมอง เซลล์ประสาทที่ลิ้นและจมูก เป็นต้น [7],[8]

เซลล์ประสาทประกอบด้วยส่วนใหญ่ๆ 4 ส่วน [7] คือตัวเซลล์ประสาทหรือนิวรอน ซึ่งมีนิวเคลียสอยู่ตรงกลาง รอบๆ ตัวเซลประสาทมีสิ่งที่ยื่นออกไปเพื่อรับและส่งสัญญาณจากเซลประสาทอื่นๆ สิ่งดังกล่าวเรียกว่า แอกซอน (Axon) ที่ปลายกิ่งจะแตกออกเป็นก้านย่อยๆ เรียกว่า เดนไดรท์ (Dendrite) รอยต่อระหว่างก้านของเซลประสาทที่ต่างกันเรียกว่า ไซแนปส์ (Synapse) ซึ่งสามารถเปลี่ยนค่าความต้านทานได้ตามสัญญาณที่ส่งระหว่างกันของเซลประสาท การส่งสัญญาณระหว่างเซลประสาททำได้โดยการถ่ายเทสารประกอบโซเดียมและโพแทสเซียม [7]

ฮอดกิ้น (Hodkin) และฮักลีย์ (Huxley) [7] ซึ่งได้รับรางวัลโนเบลทางชีววิทยาได้ค้นพบว่าการไหลของสารประกอบโซเดียมและโพแทสเซียมของเซลล์ประสาทของปลาหมึกได้ทำให้เกิดความต่างศักย์ จะอยู่ระหว่าง 50 ถึง 70 มิลลิโวลต์ จากผลการศึกษาดังกล่าวทำให้เราสามารถจำลองการทำงานของเซลล์ประสาทโดยอาศัยวงจรอิเล็กทรอนิกส์ได้ โครงสร้างของโครงข่ายประสาทเทียมมีลักษณะที่คล้ายคลึงกับสมองและระบบประสาทซึ่งประกอบด้วยส่วนของการประมวลผลต่างๆ ที่เรียกว่านิวรอน (Neuron) ทุกๆ นิวรอนสามารถมีอินพุทได้หลายอินพุท แต่มีเอาต์พุทเพียงเอาต์พุทเดียว และทุกๆ เอาต์พุทจะแยกไปยังอินพุทของนิวรอนอื่นๆ ภายในโครงข่าย การติดต่อกันภายในระหว่างนิวรอนไม่ใช่ลักษณะของการต่อแบบธรรมดา ทุกๆ อินพุทจะมีค่าถ่วงน้ำหนักเป็นตัวกำหนดกำลังของการติดต่อภายในและช่วยในการตัดสินใจ การทำงานของนิวรอนในบางโครงข่ายจะถูกกำหนดไว้ตายตัว แต่บางโครงข่ายสามารถที่จะปรับแต่งได้ ซึ่งอาจจะเป็นการปรับแต่งจากภายนอกโครงข่ายหรือนิวรอนสามารถปรับตัวได้ด้วยตัวของมันเอง[4] ในจุดนี้เองแสดงถึงความสามารถในการเรียนรู้และจดจำของโครงข่ายประสาทเทียม มีโครงข่ายประสาทเทียมหลายรูปแบบ สำหรับโครงข่ายประสาทเทียมที่ใช้ในการ Clustering ส่วนใหญ่จะนำวิธีการของ Winner-Take-All มาใช้ เช่น Competitive Learning, Kohonen self-organizing maps และ Adaptive Resonance Theory

โครงข่ายประสาทเทียม ประกอบด้วยตัวประมวลผลที่เรียกว่า "นิวรอน" ที่เชื่อมต่อถึงกันและนิวรอนแต่ละตัวก็จะส่งผ่านสัญญาณจากตัวมันไปยังนิวรอนตัวอื่นๆที่อยู่ต่างเลเยอร์กัน ผ่าน

ทาง จุดเชื่อมต่อที่เรียกว่า weight โดยที่ตัวนิวรอนที่อยู่ในเลเยอร์เดียวกันจะมีหน้าที่เหมือนกัน ซึ่งแต่ละนิวรอนจะรับค่าสัญญาณอินพุทที่เชื่อมต่อกับตัวมันทั้งหมดมาประมวลผล และจะให้สัญญาณออก มาที่เอาท์พุทเพียงหนึ่งค่าเท่านั้น โดยที่สัญญาณเอาท์พุทที่ได้จากแต่ละเลเยอร์จะถูกส่งไปยังนิวรอนในเลเยอร์ปลายทางต่อไป รูปที่ 2.4 แสดงแสดงแผนภาพของ Artificial Neural Network และ ตารางที่ 2.1 แสดงเปรียบเทียบระหว่าง Biological กับ Artificial Neural Network [21],[22],[7],[8]



รูปที่ 2.4 แสดงแผนภาพของโครงข่ายประสาทเทียม

ตารางที่ 2.1 แสดงการเปรียบเทียบระหว่าง Biological กับ Artificial Neural Network

Biological Neural Network	Artificial Neural Network
Soma	Neuron
Dendrite	อินพุท
Axon	เอาท์พุท
Synapse	Weight

2.4 Adaptive Resonance Theory Neural Network

Adaptive Resonance Theory (ART) พัฒนาโดย Carpenter และ Grossberg ในปี 1987 เป้าหมายของอัลกอริทึมนี้คือ การจัดแบ่งกลุ่มข้อมูลที่เป็นตัวเลข ซึ่งมีอัลกอริทึมของ ART1 สำหรับจัดแบ่งกลุ่มข้อมูลตัวเลขที่เป็น Binary และอัลกอริทึมของ ART2 สำหรับการจัดแบ่งกลุ่มข้อมูลตัวเลขที่เป็นเลขจำนวนจริง ลักษณะการทำงานของนิวรอนนี้เป็นแบบเรียนรู้ด้วยตัวเอง (Unsupervised Learning) [4] [22],[23] Adaptive Resonance Theory (ART) ถูกออกแบบ

เพื่อให้ผู้ใช้สามารถควบคุมระดับความคล้ายคลึงของข้อมูล (Pattern) ในการจัดแบ่งกลุ่มได้ และคุณสมบัติ Stability และ Plasticity ทำให้ ART มีความสามารถจัดแบ่งกลุ่มข้อมูลที่มีลักษณะหลายหลากได้ดี [1],[2],[7],[19] คุณสมบัติ Stability หมายถึง การที่ Winning node ไม่มีการเปลี่ยนแปลงเมื่อ Input Pattern ตัวเดิมที่เคยถูกส่งเข้าสู่โมเดล ถูกส่งเข้าโมเดลอีกครั้ง สำหรับในโครงสร้างประสาทเทียมแบบ Unsupervised Learning ทั่วไป จะใช้วิธีลดอัตราการเรียนรู้ (Learning Rate) ลงให้เข้าใกล้ 0 มากที่สุด ผลที่ได้คือ Winning node เริ่มมีความ Stability มากขึ้นแต่อย่างไรก็ตามการลดอัตราการเรียนรู้ของโครงข่ายประสาทเทียมก็มีข้อเสียด้วยเช่นกัน คือ เมื่อลดอัตราการเรียนรู้ลงเป็น 0 แล้วจะไม่มีความสามารถในการจัดกลุ่ม Cluster ที่เข้าใหม่ ปัญหาในการที่โครงข่ายประสาทเทียมแบบ Unsupervised Learning ไม่สามารถเรียนรู้ข้อมูลใหม่ได้เรียกว่า ปัญหา Plasticity ดังนั้น Carpenter และ Grossberg จึงได้เสนอแนวคิด Stability และ Plasticity ขึ้น ในวิธีการที่เรียกว่าทฤษฎีการปรับตัวตามความสอดคล้อง (Adaptive Resonance Theory) โดยมีวิธีการสำหรับควบคุมระดับความคล้ายคลึงของข้อมูลตามค่า Vigilance Threshold และในปี 1987 Carpenter และ Grossberg ได้นำเสนอโมเดลที่ได้รับออกแบบที่มีคุณสมบัติ Stability และ Plasticity มี 2 โมเดล คือ ART1 (Binary ART) และ ART2 (Continuous ART)

วิธีการควบคุมระดับความคล้ายคลึงของข้อมูลทำได้โดยการรีเซตเมื่อค่าความคล้ายคลึงน้อยกว่าค่า Vigilance ที่กำหนดโดยผู้ใช้ ดังนั้นการทำงานของ ART จึงแบ่งออกได้ 2 ส่วน คือ Competition และ Resonance [4],[7],[8]

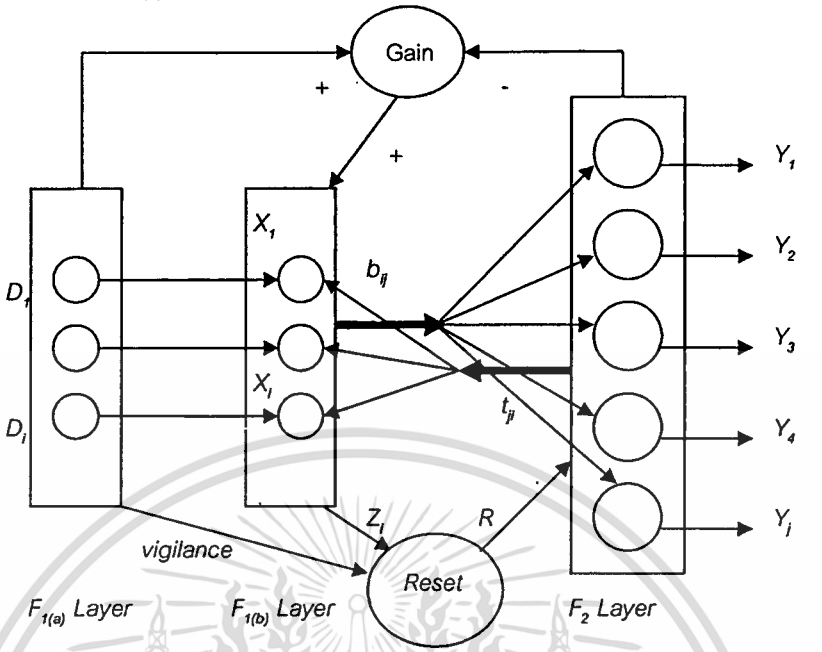
1. Competition คือส่วนที่ทำการเปรียบเทียบความเหมือนกันระหว่างข้อมูลเข้ากับโหนดต่างๆในนิวรอน

2. Resonance คือส่วนที่ตัดสินหาความสอดคล้องการจัดแบ่งกลุ่มที่เกิดจากส่วน Competition

2.4.1 ART1 Network

โครงสร้างของ ART1 ประกอบด้วย 2 เลเยอร์ คือ F_1 (Processing Layer) และF_2 (Output Layer) การทำงานของ Binary Adaptive Resonance Theory Neural Network มีการทำงานแบบ Feed Back เพื่อให้โครงข่ายประสาทเทียมนี้มีความสามารถในการควบคุมระดับความคล้ายคลึงของการจัดกลุ่มโดยใช้วิธีการรีเซต [4] ดังนั้นโครงข่ายประสาทนี้จึงมี Weights 2 ระดับ คือ Bottom-up weight ,b_{ij} ,ซึ่งจะเชื่อมต่อและส่งข้อมูลระหว่างโหนดที่ i ใน $F_{1(b)}$ layer ไปยังโหนด

ที่ j ใน F_2 layer และ Top-down weight, t_{ji} ,ซึ่งจะเชื่อมต่อและส่งข้อมูลระหว่างโหนดที่ j ใน F_2 layer และโหนดที่ i ใน $F_{1(b)}$ layer



รูปที่ 2.5 โครงสร้างของ Binary Adaptive Resonance Theory Neural

ในส่วนของการรับข้อมูลเข้า โครงข่ายประสาทเทียมนี้ถูกออกแบบมาให้รับข้อมูลที่เป็นไบนารี การเรียนรู้ข้อมูลที่เข้ามาเป็นแบบเรียนรู้ด้วยตัวเอง (Unsupervised Learning) โครงสร้างของโครงข่ายประสาทเทียมนี้ประกอบด้วย 3 ชั้น คือ

1. Input Layer ($F_{1(a)}$ layer)
2. Interface Layer ($F_{1(b)}$ layer)
3. Output Layer (F_2 layer)

Binary Adaptive Resonance Theory Neural Network มี R unit เพื่อใช้ในการรีเซตการจัดกลุ่มข้อมูล เรียกว่า วิธีการ Reset mechanism ซึ่งจะทำการรีเซตโดยเงื่อนไขที่มีค่า Vigilance Parameter โดยค่า Vigilance Parameter เป็นค่าที่ใช้ควบคุมระดับความคล้ายคลึงของข้อมูลเข้าและ Weights ในโครงข่ายประสาทเทียม

2.4.2 ART1 Clustering Algorithms

ขั้นตอนที่ 0: กำหนดค่าเริ่มต้นให้กับ bottom-up weight และ top-down weight ในแต่ละโหนดของโครงข่ายประสาทเทียม ค่าเริ่มต้นเหล่านี้อาจได้จากการสุ่มเลือกจากข้อมูลที่นำมาใช้ในกระบวนการเรียนรู้

และกำหนดค่าVigilance parameter, ρ, = (0, 1] และ L (Learning Rate) และ Top-down weight มีค่าเป็น 1

ขั้นตอนที่ 1: เมื่อยังไม่ตรงเงื่อนไขในการหยุด ให้ทำขั้นตอนที่ 2-9

ขั้นตอนที่ 2: เลือกข้อมูลเข้าทั้งหมด หาค่า Norm

$\|S\| = \sum_i S_i$, ทำขั้นตอนที่ 3-8 ต่อไป

ขั้นตอนที่ 3: กำหนดข้อมูลเข้า *D* จาก $F_{1(a)}$ ให้กับตัวแปร *X* ใน $F_{1(b)}$

$$X_i = S_i$$

ขั้นตอนที่ 4: คำนวณเปรียบเทียบข้อมูลเข้า *X* กับแต่ละโหนดของโครงข่ายประสาทเทียมจนครบทุกโหนด

$$Y_j = \sum_i b_{ij} x_i$$

ขั้นตอนที่ 5: เปรียบเทียบหาโหนด *J* ของโครงข่ายประสาทเทียมที่มีค่ามากที่สุด

ขั้นตอนที่ 6: เปรียบเทียบข้อมูลเข้า *X* กับ top-down weight ที่เชื่อมต่อกับ winning node *J*

$$x_i = s_i t_{Ji}$$

คำนวณหาค่า Norm ของ X

$$\|x\| = \sum_i x_i$$

ขั้นตอนที่ 7: ทดสอบเงื่อนไขการ reset mechanism

ถ้า $\frac{\|x\|}{\|s\|} < \rho$, ให้ Y_J = -1 (ยับยั้งโหนด *J*), ทำขั้นตอนที่ 5 อีกครั้ง

ถ้า $\frac{\|x\|}{\|s\|} \geq \rho$, ให้ทำขั้นตอนที่ 8

ขั้นตอนที่ 8: ปรับปรุงค่า bottom-up weight และ top-down weight ของโหนด *J*

$$b_{iJ}(new) = \frac{Lx_i}{L-1+\|x\|}$$

$$t_{Ji}(new) = x_i$$

ขั้นตอนที่ 9: ทดสอบเงื่อนไขการหยุด เงื่อนไขการหยุดสามารถกำหนดโดย:

9.1Weight ไม่มีการเปลี่ยนแปลง

9.2ไม่มีการรีเซต

9.3 จำนวนรอบของการเรียนรู้

โครงข่ายประสาทเทียมแบบ ART นี้มีคุณสมบัติการทำงานแบบ order-dependent ซึ่งมีความหมายคือ จำนวนของกลุ่ม (Clusters, Partitions) ขึ้นอยู่กับการกำหนดค่า Vigilance Threshold ดังนั้น ในช่วงการเรียนรู้ (Learning) ของโครงข่ายประสาทเทียมแบบ ART ค่า Vigilance จะเป็นค่า Threshold ที่ใช้ตัดสินหาความสอดคล้องของการจัดแบ่งกลุ่มข้อมูล ในกรณีที่ข้อมูลที่นำเข้าโครงข่ายประสาทเทียมไม่สามารถจัดให้อยู่ในเอาท์พุทโหนดที่มีอยู่เดิมได้ โครงข่ายประสาทเทียมจะทำการสร้างเอาท์พุทโหนดใหม่ขึ้นมาแทนและให้ข้อมูลนั้นถูกจัดให้อยู่ในเอาท์พุทโหนดใหม่ที่สร้างขึ้น ในอีกความหมายหนึ่งคือการกำหนดค่า Vigilance Threshold มีผลกับจำนวนกลุ่มที่ได้จากการทำงานของโครงข่ายประสาทเทียมแบบ ART

2.5 การวัดคุณภาพของการจัดกลุ่ม (Cluster Evaluation Measure)

การประเมินประสิทธิภาพของอัลกอริทึมที่ใช้ในการจัดกลุ่มได้จากการวัดคุณภาพของข้อมูลในแต่ละคลัสเตอร์ การวัดคุณภาพจะขึ้นอยู่กับความรู้ที่เรามี เกี่ยวกับข้อมูลที่เราใช้ในการจัดกลุ่มว่าข้อมูลนั้นๆมีการจัดเป็นหมวดหมู่อย่างไร อาทิเช่น การที่เรารู้ว่าข่าวแต่ละข่าวนั้นถูกระบุไว้ว่ามันถูกจัดให้อยู่ในหัวข้อข่าวไหน ผลลัพธ์ของอัลกอริทึมจะเก็บลงตาราง Confusion Matrix [28] (class x cluster) ในแถว (Row) จะเป็น class ของข้อมูลและ columns เป็น cluster ของอัลกอริทึม

ตารางที่ 2.2 ตัวอย่างตาราง Confusion Matrix

	Cluster1	Cluster2	Cluster3
Class1	785	25	0
Class2	5	245	7
Class3	0	3	695

ซึ่งเราสามารถนำผลของการจัดกลุ่มโดยอัลกอริทึมของเรามาเปรียบเทียบกันกับกลุ่มของข้อมูลที่ได้มีการจัดกลุ่มไว้แล้วว่าคุณภาพของการจัดกลุ่มข้อมูลของอัลกอริทึมเราเป็นอย่างไร ซึ่งข้อมูลที่ใช้ในงานวิจัยนี้เป็นข้อมูลที่มีผู้รวบรวมและได้แบ่งแยกหมวดหมู่ของข้อมูลไว้แล้วดังนั้นเราจึงเลือกใช้วิธีการวัดประสิทธิภาพ 2 ตัวชี้วัด คือ Entropy และ F-Measure [13],[23]

2.5.1 Entropy

ค่า Entropy เป็นตัวชี้วัดว่าผลลัพธ์ Entropy สามารถมีค่าได้ตั้งแต่ 0 จนถึง $\log_2(C)$ เมื่อ C คือจำนวนกลุ่มที่มีได้ของข้อมูล เช่น ถ้ามี 2 class ค่า Entropy จะมีค่าระหว่าง 0-1 หรือ ถ้ามี 3 class ค่า Entropy จะมีค่าระหว่าง 0-1.585 ค่า Entropy ที่เข้าใกล้ 0 แสดงถึงระดับประสิทธิภาพของการจัดแบ่งกลุ่ม Cluster มีความถูกต้องสูงและ Impurity ต่ำ เช่น จากตัวอย่างตารางที่ 2.2 Confusion Matrix ค่า Entropy ของทั้งเซตคือ 0.0959 และถ้าคำนวณหา Entropy ของแต่ละ Cluster จะได้ดังนี้

ตารางที่ 2.3 ตารางแสดงค่า Entropy ของแต่ละ Cluster

Cluster	Entropy
1	0.0384
2	0.3656
3	0.0559

Cluster1 มีค่า Entropy ดีที่สุด เนื่องจากมีสมาชิกของ Class อื่นเข้ามาอยู่ใน Cluster เดียวกันน้อยที่สุด คือ มีสมาชิก 5 ตัวที่เป็นของ Class2 เข้ามาปนอยู่ใน Cluster1 ซึ่งเป็นตัวแทนของ Class1 และเราเรียกปรากฏการณ์นี้ว่า ระดับ Impurity ต่ำ ขั้นตอนการคำนวณค่า Entropy เริ่มจากกำหนดให้ P คือผลที่ได้จากอัลกอริทึมจัดกลุ่ม ซึ่งมี m คลัสเตอร์ ที่ทุกคลัสเตอร์ j ใน P เราจะคำนวณหาค่า p_{ij} ซึ่งเป็นค่าความเป็นไปได้ที่จำนวนของคลัสเตอร์ j จะอยู่ใน คลาส i ค่า Entropy ของแต่ละคลัสเตอร์ j คำนวณได้จากสมการที่ 2.6 และค่าผลรวมสุทธิคำนวณจากสมการที่ 2.7 ตามลำดับ

$$E_j = -\sum_i p_{ij} \log(p_{ij}) \tag{2.6}$$

$$E_p = \sum_{j=1}^{m} \frac{N_j}{N} \times E_j \tag{2.7}$$

เมื่อ

N_j คือขนาดของคลัสเตอร์ j

N คือ จำนวนของข้อมูลทั้งหมด

m คือจำนวนคลัสเตอร์

2.5.2 F-Measure

ตัวชี้วัด ตัวที่สองที่จะกล่าวถึง ค่า F-Measure เป็นค่าที่เกิดจากการรวมกันของค่า Precision และ ค่า Recall ซึ่งสองค่านี้เป็นแนวคิดจากงานวิจัย Information Retrieval โดยที่ค่า Precision และค่า Recall ของคลัสเตอร์ j กับ คลาส i แสดงได้ดังนี้

$$P(i,j)=\frac{N_{ij}}{N_j} \tag{2.8}$$

$$R(i,j)=\frac{N_{ij}}{N_i} \tag{2.9}$$

เมื่อ

N_{ij} คือจำนวนสมาชิกของ คลาส i ในคลัสเตอร์ j

N_j คือจำนวนสมาชิกของ คลัสเตอร์ j

N_i คือจำนวนสมาชิกของ คลาส i

ค่า F-Measure ของ คลาส i หาได้จาก

$$F(i)=\frac{2PR}{P+R} \tag{2.10}$$

ที่ คลาส i เราต้องการให้คลัสเตอร์มีค่า F-Measure สูงๆที่ คลัสเตอร์ j สำหรับ คลาส i และ ค่า F-Measure จะกลายมาเป็นค่าคะแนนสำหรับ คลาส i ค่า F-Measure รวมของ P ที่ได้จากการจัดกลุ่มคือค่าเฉลี่ยของค่า F-Measure สำหรับแต่ละ คลาส i หาได้จาก

$$F_p=\frac{\sum_i\left(|i|\times F(i)\right)}{\sum_i|i|} \tag{2.11}$$

$|i|$ คือจำนวนของข้อมูลใน คลาส i

ค่า F-Measure มีค่าระหว่าง 0-1 ถ้าค่า F-Measure ที่ได้มีค่าสูงเข้าใกล้ 1 แสดงว่าข้อมูลใน Cluster ที่ได้จากการจัดกลุ่มมีความถูกต้องสูง และถ้าค่า F-Measure ที่ได้มีค่าสูงเข้าใกล้ 0 แสดงว่าข้อมูลใน Cluster ที่ได้จากการจัดกลุ่มมีความถูกต้องต่ำ เช่น จากตัวอย่างตารางที่ 2.2 Confusion Matrix ค่า F-Measure ของทั้งเซตคือ 0.9776 และถ้าคำนวณหา F-Measure ของแต่ละ Cluster จะได้ดังนี้

ตารางที่ 2.4 ตารางแสดงค่า F-Measure ของแต่ละ Cluster

Cluster	F-Measure
1	0.9906
2	0.8645
3	0.9853

บทที่ 3
ทฤษฎีและหลักการทำงานของ เท็กอะแดปทีพเรโซแน้นเทียรีนิวรอลเน็ตเวิร์ค (A Text Adaptive Resonance Theory Neural Network)

3.1 Document Representation

เพื่อแปลงข้อมูล Textual อยู่ในรูปแบบสำหรับ Learning Algorithm ข้อมูล Textual จะต้องผ่านขั้นตอน Pre-processing ดังนี้

- ตัดตัวอักษร digit และ เครื่องหมายจุดที่ระบุช่วงวรรคตอนในเนื้อหา (punctuation marks)
- แปลงคำที่เป็นตัวอักษรใหญ่ให้เป็นตัวอักษรเล็ก
- ตัดคำที่เป็น Stop Words ออก [29]
- ตัดส่วนที่เป็น Suffix ของคำออก โดยใช้อัลกอริทึม Porter เพื่อหารากของคำศัพท์ [30]

ขั้นตอนก่อนการเปรียบเทียบความคล้ายคลึงของเอกสารคือ ขั้นตอนการแทนเอกสารต่างๆ (Document Representation) ให้อยู่รูปแบบที่สามารถนำมาคำนวณได้ เอกสารต่างๆ ซึ่งประกอบด้วย ชื่อเรื่อง คำสำคัญ สามารถแทนอยู่ในรูปแบบของ Cartesian Product ได้ดังนี้ [5]

$$Doc = D_1 \times D_2 \times D_3 \times ... \times D_d \tag{3.1}$$

d คือจำนวนคุณสมบัติของเอกสาร และคุณสมบัติของเอกสารมีลักษณะข้อมูลเป็นข้อความ ตัวอย่างเช่น

$$Doc = Title \times Keyword \tag{3.2}$$

เมื่อ Title Feature คือคำที่ใช้อธิบายชื่อเรื่องของเอกสาร

Keyword Feature คือคำที่ใช้อธิบายคำที่พบบ่อยในเอกสาร หรือคำสำคัญของเอกสาร

ตารางที่ 3.1 ข้อมูลเอกสาร

	Title	Keyword
Doc A	{tax,net}	{salary,money,costs}
Doc B	{network,internet}	{tcp/ip,web,e-mail}
Doc C	{internet,google}	{search,rank,web}
Doc D	{internet,microsoft}	{desktop,software}

3.2 Similarity Measure for Symbolic Objects

การเปรียบเทียบความคล้ายคลึงของเอกสาร ระหว่างเอกสาร A และเอกสาร B ตามแนวคิดของ El-Sonbaty สามารถเขียนในสมการได้ดังนี้

$$S(A,B)=\sum_{k=1}^{d} S(A_k,B_k) \tag{3.3}$$

ในการเปรียบเทียบความคล้ายคลึงของ $S(A_k,B_k)$ มีการเปรียบเทียบ 2 ส่วนย่อยคือ ส่วนที่เป็นขนาดของ Feature เรียกว่า Span, $S_s(A,B)$ และส่วนที่เป็นเนื้อหาของ Feature เรียกว่า Content, $S_c(A,B)$ ซึ่งนิยามสมการไว้ดังนี้ [5]

ส่วนการเปรียบเทียบความคล้ายคลึงของ Span มีนิยามเป็น

$$S_s(A,B)=\frac{(l_a+l_b)}{2xl_s} \tag{3.4}$$

ส่วนการเปรียบเทียบความคล้ายคลึงของ Content มีนิยามเป็น

$$S_c(A,B)=\frac{inters}{l_s} \tag{3.5}$$

นิยามของสัญลักษณ์ในสมการมีดังนี้

l_a = จำนวนสมาชิกทั้งหมดใน Feature A

l_b = จำนวนสมาชิกทั้งหมดใน Feature B

inters = จำนวนของสมาชิกทั้งหมดที่ Intersection กันระหว่าง Feature A และ Feature B

l_s = $l_a + l_b - inters$ = จำนวนสมาชิกทั้งหมดของ Feature A และ Feature B รวมกันลบด้วยจำนวนของสมาชิกทั้งหมดที่ Intersection กันระหว่าง Feature A และ Feature B

เมื่อเปรียบเทียบความคล้ายคลึงครบทั้งสองส่วนย่อยแล้ว จึงทำการรวมค่าผลลัพธ์จากสองย่อยเข้าด้วยดังสมการ Net Similarity นี้

$$S(A_k, B_k) = S_s(A_k, B_k) + S_c(A_k, B_k) \quad (3.6)$$

ค่าความคล้ายคลึงของ Net Similarity นี้สามารถคำนวณได้ ดังตัวอย่างที่ 3.1

ตัวอย่างที่ 3.1 การใช้สูตร Similarity Measure for symbolic object

$S(DocA_{Title}, DocB_{Title})$

S({'tax', 'net'},{'network', 'internet'})

= [(2+2) / (2*(2+2-0))]+[0/(2+2-0)]

= 0.5+0.0

= 0.5

$S(DocA_{Keyword}, DocB_{Keyword})$

S({'salary', 'money', 'costs'},{'tcp/ip', 'web', 'e-mail'})

= [(3+3) / (2*(3+3-0))]+[0/(3+3-0)]

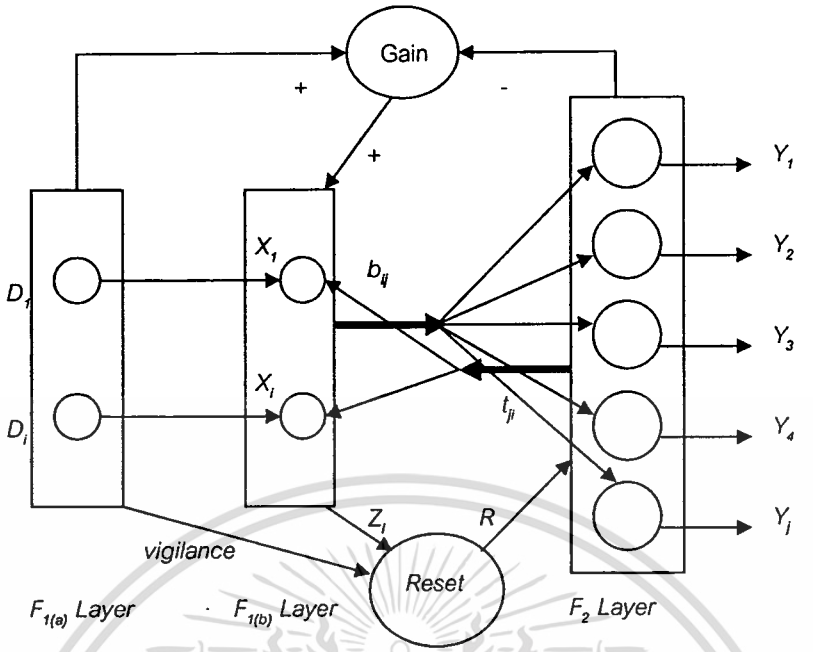
= 0.5+0.0

= 0.5

ดังนั้น Net Similarity ได้เท่ากับ 0.5+0.5 = 1.0

หมายเหตุ: ข้อมูลที่ใช้ในตัวอย่างที่ 3.1 มาจากข้อมูลในตารางที่ 3.1

3.3 A Text Adaptive Resonance Theory Neural Network



รูปที่ 3.1 โครงสร้างของ Text Adaptive Resonance Theory Neural Network

จากรูปที่ 3.1 Text ART Neural Network มีโครงสร้างไม่แตกต่างจากโครงสร้างของ Binary ART Neural Network การเรียนรู้ข้อมูลที่เข้ามาเป็นแบบเรียนรู้ด้วยตัวเอง (Unsupervised Learning) โครงสร้างของโครงข่ายนิวรอลนี้ประกอบด้วย 3 ชั้น คือ

1. Input Layer ($F_{1(a)}$ layer)
2. Interface Layer ($F_{1(b)}$ layer)
3. Output Layer (F_2 layer)

แต่มีความแตกต่างคือ ในส่วนของการรับข้อมูลเข้า โครงข่ายนิวรอลนี้ถูกออกแบบมาให้รับข้อมูลที่เป็นข้อความ (Textual data) ดังนั้น โครงข่ายนิวรอลแบบ Text ART นี้จึงมีข้อแตกต่างจาก Binary ART 3 ประการ คือ

1. พื้นที่สำหรับเก็บข้อมูลในส่วนของ bottom-up weight และ top-down weight
2. การหาโหนดชนะโดยใช้ similarity measure สำหรับ symbolic objects
3. การปรับ bottom-up weight และ top-down weight ของ winning node

Text Adaptive Resonance Theory Neural Network มี *R* unit เพื่อใช้ในการรีเซตการจัดกลุ่มข้อมูล เรียกว่า วิธีการ Reset mechanism ซึ่งจะทำการรีเซตโดยเงื่อนไขที่มีค่า Vigilance Parameter โดยค่า Vigilance Parameter เป็นค่าที่ใช้ควบคุมระดับความคล้ายคลึงของข้อมูลเข้าและ Weights ในนิวรอลเน็ตเวิร์ค

การทำงานของ Text Adaptive Resonance Theory Neural Network มีการทำงานแบบ Feed Back เพื่อให้โครงข่ายนิวรอลนี้มีความสามารถในการควบคุมระดับความคล้ายคลึงของการจัดกลุ่มโดยใช้วิธีการรีเซต [4] ดังนั้นโครงข่ายประสาทนี้จึงมี Weights 2 ระดับคือ Bottom-up weight ,b_{ij} ,ซึ่งจะเชื่อมต่อและส่งข้อมูลระหว่างโหนดที่ i ใน $F_{1(b)}$ layer ไปยังโหนดที่ j ใน F_2 layer และ Top-down weight, t_{ji} ,ซึ่งจะเชื่อมต่อและส่งข้อมูลระหว่างโหนดที่ j ใน F_2 layer และโหนดที่ i ใน $F_{1(b)}$ layer เพื่อที่จะทำให้โครงข่ายนิวรอลนี้สามารถหาความคล้ายคลึงของข้อมูลเข้าที่เป็น Qualitative value ในที่นี้คือ Text ได้ ดังนั้นทั้ง Bottom-up weight และ Top-down weight จะเก็บค่าใน Weight เป็น Qualitative value และค่าแสดงความเป็นสมาชิกของ Weight

$$b_{ij} = \{(A_{1ij}, e_{1ij}), (A_{2ij}, e_{2ij}), ..., (A_{pij}, e_{pij})\} \tag{3.7}$$

กำหนดให้ A_{pij} คือ ค่า qualitative value ของ Bottom-up Weightและ e_{pij} คือค่า Degree แสดงความเป็นสมาชิกของ A_{pij}

ค่า e_{pij} ของ A_{pij} ในโครงข่ายนิวรอลนี้มีค่าอยู่ในช่วงระหว่าง 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} เป็นสมาชิกของข้อมูลเข้า i

$$t_{ji} = \{(B_{1ji}, e_{1ji}), (B_{2ji}, e_{2ji}), ..., (B_{pji}, e_{pji})\} \tag{3.8}$$

กำหนดให้ B_{pij} คือ ค่า qualitative value ของ Top-down weight และ e_{pij} คือค่า Degree แสดงความเป็นสมาชิกของ B_{pji}

ค่า e_{pji} ของ B_{pji} ในโครงข่ายนิวรอลนี้มีค่าอยู่ในช่วงระหว่าง 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

3.4 Learning Algorithms

A Text Adaptive Resonance Theory Neural Network เป็นการขยายความสามารถของ ART1 Neural Network โดยเพิ่มเติมแนวคิดเกี่ยวกับคุณสมบัติที่มีค่าของข้อมูลเป็นแบบข้อมูลเชิงคุณภาพ Qualitative value ซึ่งทำให้อัลกอริทึมนี้สามารถจัดการกับข้อมูลแบบเชิง

คุณภาพได้โดยตรงโดยไม่ต้องผ่านกระบวนการแปลงค่า Qualitative Value เป็น Numerical Value ขั้นตอนของการเรียนรู้ของอัลกอริทึมมีดังนี้

ขั้นตอนที่ 0: กำหนดค่าเริ่มต้นให้กับ bottom-up weight และ top-down weight ในแต่ละโหนดของโครงข่ายนิวรอล ค่าเริ่มต้นเหล่านี้อาจได้จากการสุ่มเลือกจากข้อมูลที่นำมาใช้ในกระบวนการเรียนรู้

และกำหนดค่าVigilance parameter, ρ = (0,1]

ขั้นตอนที่ 1: เมื่อยังไม่ตรงเงื่อนไขในการหยุด ให้ทำขั้นตอนที่ 2-9

ขั้นตอนที่ 2: เลือกข้อมูลเข้าทั้งหมด แล้วทำการ Transpose

$$Doc = (D_1, D_2, D_3, ..., D_d)^t \text{ , ทำขั้นตอนที่ 3-8 ต่อไป} \quad (3.9)$$

ขั้นตอนที่ 3: กำหนดข้อมูลเข้า D จาก F1(a) ให้กับตัวแปร X ใน F1(b)

$$X_k = D_k \quad (3.10)$$

ขั้นตอนที่ 4: คำนวณเปรียบเทียบข้อมูลเข้า X กับแต่ละโหนดของโครงข่ายนิวรอลจนครบทุกโหนด

$$Y_j = \sum_{i=1}^{d} \sum_{n=1}^{p} S(X_i, A_{nij}).e_{nij} \quad (3.11)$$

กำหนดให้

p คือจำนวนของ bottom-up weight

d คือจำนวนของ feature values

ขั้นตอนที่ 5: เปรียบเทียบหาโหนด J ของโครงข่ายนิวรอลที่มีค่ามากที่สุด

ขั้นตอนที่ 6: เปรียบเทียบข้อมูลเข้า X กับ top-down weight ที่เชื่อมต่อกับ winning node J

$$Z = \frac{\left(\sum_{i=1}^{d} S(X_i, t_{Ji})\right) - (0.5 * d)}{(2 * d) - (0.5 * d)} \quad (3.12)$$

ขั้นตอนที่ 7: ทดสอบเงื่อนไขการ Reset Mechanism

ถ้า $Z < \rho$, ให้ $Y_J = -1$ (ยับยั้งโหนด J), ทำขั้นตอนที่ 5 อีกครั้ง

ในกรณี ทุกเอาท์พุตโหนดถูกยับยั้ง ให้สร้างเอาท์พุตโหนดใหม่

ถ้า $Z \geq \rho$, ให้ทำขั้นตอนที่ 8

ขั้นตอนที่ 8: ปรับปรุงค่า bottom-up weight และ top-down weight ของโหนด J

$$b_{ij}^{(new)} = b_{ij}^{(old)} \cup X$$

$$e_{niJ}^{(new)} = \begin{cases} f(e_{niJ}^{(old)} + \eta) \text{ if } A_{nij} \in b_{ij} \cap X, \\ f(e_{niJ}^{(old)} - \eta) \text{ if } A_{nij} \notin b_{ij} \cap X, \\ 5 * \eta; \text{Otherwise} \end{cases} \quad (3.13)$$

$$t_{Ji}^{(new)} = t_{Ji}^{(old)} \cup X$$

$$e_{nJi}^{(new)} = \begin{cases} f(e_{nJi}^{(old)} + \eta) \text{ if } B_{nji} \in t_{Ji} \cap X, \\ f(e_{nJi}^{(old)} - \eta) \text{ if } B_{nji} \notin t_{Ji} \cap X, \\ 5 * \eta; \text{Otherwise} \end{cases} \quad (3.14)$$

กำหนดให้ $f(.)$ นิยามดังนี้

$$f(x) = \begin{cases} x & \text{if } 0 \leq x \leq 1 \\ 0 & \text{if } x < 0 \\ 1 & \text{if } x > 1 \end{cases} \quad (3.15)$$

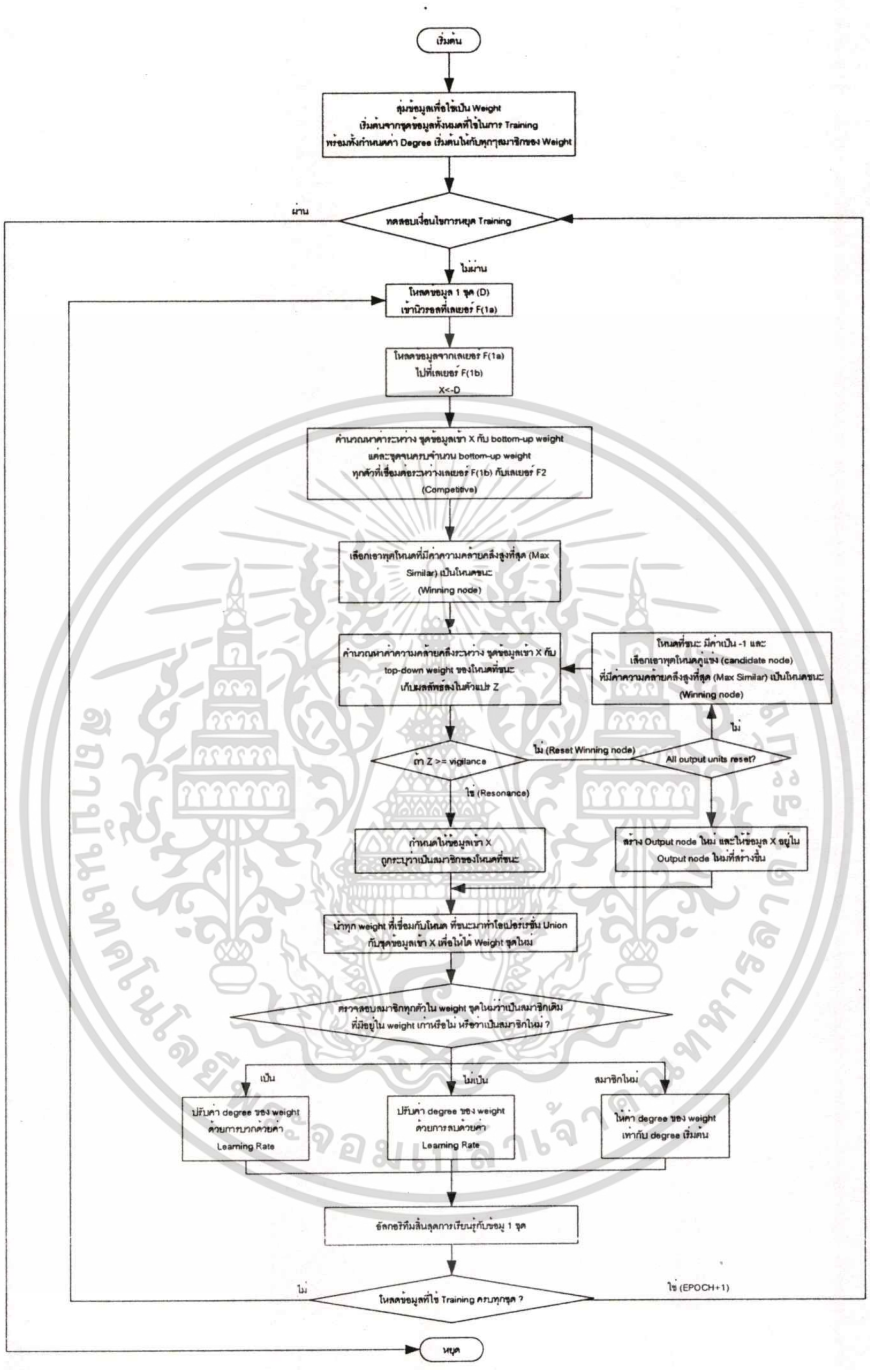
ขั้นตอนที่ 9: ทดสอบเงื่อนไขการหยุด เงื่อนไขการหยุดสามารถกำหนดโดย:

9.1Weight ไม่มีการเปลี่ยนแปลง

9.2ไม่มีการรีเซต

9.3 จำนวนรอบของการเรียนรู้

สำหรับงานวิจัยนี้ได้ทำการจำลองการทำงานของอัลกอริทึมแบบ Text ART Clustering ใน Matlab ซึ่งสามารถเขียนในรูปแบบ Flowchart ได้ดังนี้



รูปที่ 3.2 แสดง Flow Chart การทำงานของอัลกอริทึม Text ART Neural Network

3.2 ตัวอย่างการคำนวณของ Text ART Neural Network

ตัวอย่างโครงสร้างของ Text Adaptive Resonance Theory Neural Network ซึ่งมี F_1 Layer เท่ากับ 2 อินพุต และ F_2 Layer เท่ากับ 3 เอาท์พุท กำหนดค่าพารามิเตอร์ต่างๆ และตัวอย่างข้อมูลเอการอินพุตที่ผ่านการ Pre-Processing ดังนี้

Learning Rate = 0.01

Vigilance = 0.1

Learning Loop = 100

1st Training Document, D = ({'money', 'bank'},{'economic' , 'market' , 'finance' , 'interest'})

D_{Title} = {'money', 'bank'}

$D_{Keyword}$ = {'economic', 'market', 'finance', 'interest'}

Initialized bottom-up weight

b_{11} = {(interest,0.5),(bank,0.5),(credit,0.5)}

b_{21} = {(interest,0.5),(bank,0.5),(finance,0.5)}

b_{12} = {(compute,0.5),(science,0.5),(logic,0.5)}

b_{22} = {(compute,0.5),(accuracy,0.5),(math,0.5)}

b_{13} = {(logic,0.5),(bank,0.5),(biz,0.5)}

b_{23} = {(interest,0.5),(bank,0.5),(cost,0.5)}

Initialized top-down weight

t_{11} = {(market,1),(money,1),(biz,1)}

t_{12} = {(market,1),(money,1),(finance,1)}

t_{21} = {(algorithm,1),(biz,1),(crude,1)}

t_{22} = {(interest,1),(biz,1),(crude,1)}

t_{31} = {(acq,1),(economic,1),(war,1)}

t_{32} = {(money,1),(economic,1),(marvel,1)}

การคำนวณใน Competitive Layer:

$$Y = \left\lfloor S(X_{Title}, A).e + S(X_{Keyword}, A).e \right\rfloor$$

$$Y = [S_{span}(X_{Title}, A) + S_{content}(X_{Title}, A)].e + [S_{span}(X_{Keyword}, A) + S_{content}(X_{Keyword}, A)].e$$

ตัวอย่างคำนวณหาค่า Net Similarity **ของ** Output Y **ลำดับที่** 1

X_{Title} = {'money', 'bank'}

S_{span} (X_{Title}, A_{11})

รายละเอียดการคำนวณ

S_{span} ({'money', 'bank'},{'interest'}) = 0.5

S_{span} ({'money', 'bank'},{'bank'}) = 0.75

S_{span} ({'money', 'bank'},{'credit'}) = 0.5

$S_{content}$ (X_{Title},A_{11})

รายละเอียดการคำนวณ

$S_{content}$ ({'money', 'bank'},{'interest'}) = 0

$S_{content}$ ({'money', 'bank'},{'bank'}) = 0.5

$S_{content}$ ({'money', 'bank'},{'credit'}) = 0

รายละเอียดการใช้สมการ S_{span} **และสมการ** $S_{content}$ **สามารถดูได้ที่ตัวอย่างที่** 3.1

(S_{span}+ $S_{content}$) * e_{11} = [(0.5+0)*0.5, (0.75+0.5)*0.5, (0.5+0)*0.5] = [0.5, 1.25, 0.5]

Similarity **ระหว่าง** X_{Title} **และ** Bottom-up Weight **ของ** Output Y **ลำดับที่** 1

เท่ากับ (0.5+1.25+0.5) = 1.125

$X_{Keyword}$ = {'economic', 'market', 'finance', 'interest'}

S_{span} ($X_{Keyword}$, A_{21})

รายละเอียดการคำนวณ

S_{span} ({'economic', 'market', 'finance', 'interest'},{'interest'}) = 0.625

S_{span} ({'economic', 'market', 'finance', 'interest'},{'bank'}) = 0.5

S_{span} ({'economic', 'market', 'finance', 'interest'},{'finance'}) = 0.625

$S_{content}$ ($X_{Keyword}$, A_{21})

$S_{content}$ ({'economic', 'market', 'finance', 'interest'},{'interest'}) = 0.25

$S_{content}$ ({'economic', 'market', 'finance', 'interest'},{'bank'}) = 0

$S_{content}$ ({'economic', 'market', 'finance', 'interest'},{'finance'}) = 0.25

รายละเอียดการใช้สมการ S_{span} **และสมการ** $S_{content}$ **สามารถดูได้ที่ตัวอย่างที่** 3.1

(S_{span}+ $S_{content}$) * e_{21} = [(0.625+0.25)*0.5, (0.5+0)*0.5, (0.625+0.25)*0.5] = [0.875, 0.5 ,0.875]

Similarity **ระหว่าง** $X_{Keyword}$ **และ** Bottom-up Weight **ของ** Output Y **ลำดับที่** 1 **เท่ากับ** (0.875+0.5+0.875) = 1.125

Net Similarity **ของ** Output Y **ลำดับที่** 1 **เท่ากับ** 1.125+1.125 = 2.25

หลังจากนั้นทำการคำนวณหาค่า Net Similarity **ของ** Output Y **ลำดับที่** 2 **และ** 3 **จนครบทุกโหนด**

Net Similarity **ของ** Output Y **ลำดับที่** 2 **เท่ากับ** 0.75+0.75 = 1.5

Net Similarity **ของ** Output Y **ลำดับที่** 3 **เท่ากับ** 1.125+0.9375 = 2.0625

ผลลัพธ์ในตัวอย่างนี้ โหนดที่ชนะ คือ Output Y **ลำดับที่** 1 **ด้วยค่า** Net Similarity **เท่ากับ** 2.25

หลังจากนั้น เข้าสู่การคำนวณหาความสอดคล้อง (Resonance)

การคำนวณใน Resonance Layer:

X_{Title} = {'money', 'bank'}

S_{span} (X_{Title}, t_{11})

S_{span} ({'money', 'bank'}, {'market', 'money', 'biz'}) = 1.125

$S_{content}$ (X_{Title}, t_{11})

$S_{content}$ ({'money', 'bank'}, {'market', 'money', 'biz'}) = 0.25

$S(X_{Title}, t_{11}) = 1.125+0.25 = 1.375$

$X_{Keyword}$ = {'economic', 'market', 'finance', 'interest'}

$S_{span}(X_{Keyword}, t_{12})$

S_{span} ({'economic', 'market', 'finance', 'interest'}, {'market', 'money', 'finance'}) =1.2

$S_{content}(X_{Keyword}, t_{12})$

$S_{content}$ ({'economic', 'market', 'finance', 'interest'}, {'market', 'money', 'finance'}) =0.4

$S(X_{Keyword}, t_{12}) = 1.2+0.4 = 1.6$

$S(X,t_1) = 1.375+1.6 = 2.575$

$$Z = \frac{S(x, t_1) - (0.5 * d)}{(2 * d) - (0.5 * d)}$$

$$Z = \frac{(2.575) - (0.5 * 2)}{(2 * 2) - (0.5 * 2)} = 0.525$$

นำมาเปรียบเทียบกับค่าพารามิเตอร์ Vigilance Threshold ผลลัพธ์คือ Resonance

ดังนั้นให้ 1st Document เอาท์พุทที่เอาท์พุทโหนดลำดับที่ 1

หลังจากนั้นทำการปรับ Weight ของโหนดลำดับที่ 1

การปรับ Weight ของโหนดลำดับที่ 1 (โหนดที่ชนะ):

Title และ Keyword ของอินพุต Document X

X_{Title} = {'money', 'bank'}

$X_{Keyword}$ = {'economic', 'market', 'finance', 'interest'}

ค่าสมาชิกใน bottom-up weight ชุดเดิม

b_{11} = {(interest,0.5),(bank,0.5),(credit,0.5)}

b_{21} = {(interest,0.5),(bank,0.5),(finance,0.5)}

ค่าสมาชิกใน top-down weight ชุดเดิม

t_{11} = {(market,1),(money,1),(biz,1)}

t_{12} = {(market,1),(money,1),(finance,1)}

ปรับค่าสมาชิกใน bottom-up weight **ตามสมการที่** (3.13) **ได้** bottom-up weight **ชุดใหม่ ดังนี้**

b_{11} = {(interest,0.49),(bank,0.51),(credit,0.49),(money,0.05)}

b_{21} = {(interest,0.51),(bank,0.49),(finance,0.51),(economic,0.05),(market,0.05)}

ปรับค่าสมาชิกใน top-down weight **ตามสมการที่** (3.14) **ได้** top-down weight **ชุดใหม่ ดังนี้**

t_{11} = {(market,0.99),(money,1),(biz,0.99),(bank,0.05)}

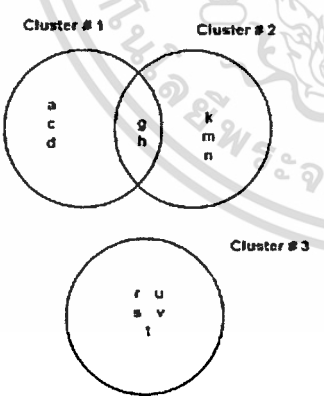
t_{12} = {(market,1),(money,0.99),(finance,1),(economic,0.05),(interest,0.05)}

บทที่ 4
วิธีการดำเนินการวิจัย

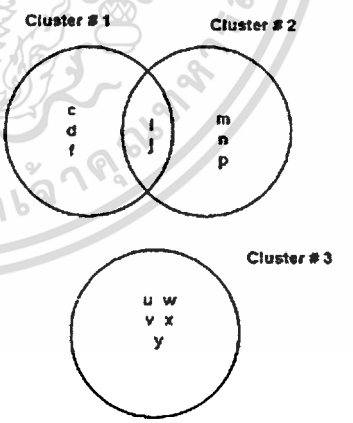
ข้อมูลที่ใช้ในการวิจัยจะมี 2 ส่วน ส่วนแรกเป็นข้อมูลตัวอักษรที่ได้จากการสุ่มจากตัวอักษรตามโครงสร้าง (Profile) ที่กำหนดขึ้นมาเอง เรียกว่า ข้อมูลสังเคราะห์ (Synthesized Dataset) การสุ่มข้อมูลขึ้นมาเองนี้ก็เพื่อใช้ในการทดสอบการประมวลผลของตัวโมเดล Text ART Neural Network ก่อนการนำไปใช้กับข้อมูลจริง ข้อมูลสังเคราะห์ (Synthesized Dataset) แบ่งออกเป็น 2 กลุ่มข้อมูล คือ Synthesized Alphabet Document และ Synthesized Text Document ส่วนที่สองจะเป็นข้อมูลข่าวรอยเตอร์ (Reuters-21578) ที่เป็นเอกสารข่าวจริง [27] การทดลองกับข้อมูลข่าว ก็เพื่อเป็นการพิจารณาถึงประสิทธิภาพของตัวโมเดลเมื่อนำไปใช้งานกับข้อมูลจริงโดยที่ข้อมูลแต่ชนิดมีรายละเอียดดังนี้

4.1 ข้อมูล Synthesized Alphabet Document

ในการทดลองเพื่อทดสอบความถูกต้องเบื้องต้นของอัลกอริทึม โดยใช้ชุดข้อมูลซึ่งสร้างขึ้นเพื่อใช้ในการฝึกฝนจำนวน 100 ชุด ซึ่งสร้างจากกลุ่มตัวอักษรจำนวน 3 กลุ่ม กลุ่มตัวอักษรนี้ใช้เป็นตัวแทนของข้อมูลที่มีค่าของคุณสมบัติเป็นข้อความ ข้อมูลแต่ละตัวประกอบด้วยคุณสมบัติ 2 คุณสมบัติ คือ Title และ Keyword



รูปที่ 4.1 a ชุดตัวอักษรที่ใช้สร้างข้อมูลใน Title



รูปที่ 4.1 b ชุดตัวอักษรที่ใช้สร้างข้อมูลใน Keyword

ตารางที่ 4.1 Synthesized Alphabet Document

กลุ่มข้อมูล	จำนวนข่าวที่ใช้เรียนรู้	จำนวนข่าวที่ใช้ทดสอบ
Class 1	34	33
Class 2	30	26
Class 3	36	41
รวม	100	100

รายละเอียดของชุดข้อมูลตัวอักษรที่ใช้ในการทดลองแสดงในตารางที่ 4.1 หลักจากเสร็จสินการเทรนนิ่งแล้วค่า Weight สุดท้ายที่ได้ก็จะเป็นตัวแทนของชุดข้อมูล ต่อจากนั้นจะทำการทดสอบความถูกต้องของการแบ่งกลุ่มของโมเดล ผลที่ได้จากการทดสอบความถูกต้องของตัวโมเดล แสดงในตารางที่ 4.2

ตารางที่ 4.2 แสดงผลลัพธ์ที่ได้จากการทดสอบของ Synthesized Alphabet Documents

Class	จำนวนสมาชิกที่ตกในแต่ละ Cluster		
	1	2	3
1	0	32	0
2	0	1	27
3	41	0	0
ค่า F measure = 0.99009			
ค่า Entropy = 0.044368			

Learning Rate (η) = 0.01

Epoch = 100 Loops

Output Nodes = 3

Vigilance = 0.1

4.2 ข้อมูล Synthesized Text Documents

ข้อมูล Synthesized Text Documents ใช้ชุดข้อมูลซึ่งสร้างขึ้นเพื่อใช้ในการฝึกฝน (Training set) จำนวน 100 รายการ และชุดข้อมูลสำหรับการทดสอบ (Testing set) จำนวน 1,500 รายการ ซึ่งสร้างจากกลุ่มตัวอักษรจำนวน 3 กลุ่ม กลุ่มตัวอักษรนี้ใช้เป็นตัวแทนของข้อมูลที่

มีค่าของคุณสมบัติเป็นข้อความ ข้อมูลแต่ละตัวประกอบด้วยคุณสมบัติ 2 คุณสมบัติ คือ Title และ Keyword

Cluster 1:

Title = {compute, algorithm, database, intel, java, network}

Keyword = {algorithm, database, predict, cluster, web, firewall}

Cluster 2:

Title = {java, network, internet, protocol cisco, 3com}

Keyword = {web, firewall, mail, smtp, http, ftp}

Cluster 3:

Title = {car, business, travel, hotel, bank, airline}

Keyword = {market, tour, benz, toyota, money, airway}

ตารางที่ 4.3 Synthesized Text Documents

กลุ่มข้อมูล	จำนวนข่าวที่ใช้เรียนรู้	จำนวนข่าวที่ใช้ทดสอบ
Class 1	36	499
Class 2	27	510
Class 3	37	491
รวม	100	1500

รายละเอียดของชุดข้อมูลตัวอักษรที่ใช้ในการทดลองแสดงในตารางที่ 4.3 หลักจากเสร็จสิ้นการเทรนนิ่งแล้วค่า Weight สุดท้ายที่ได้ก็จะเป็นตัวแทนของชุดข้อมูล ต่อจากนั้นจะทำการทดสอบความถูกต้องของการแบ่งกลุ่มของโมเดล ผลที่ได้จากการทดสอบความถูกต้องของตัวโมเดล แสดงในตารางที่ 4.4

ตารางที่ 4.4 แสดงผลลัพธ์ที่ได้จากการทดสอบของ Synthesized Text Documents

Class	จำนวนสมาชิกที่ตกในแต่ละ Cluster		
	1	2	3
1	35	484	0
2	478	32	0
3	0	0	491
ค่า F measure = 0.95561			
ค่า Entropy = 0.16297			

Learning Rate (η) = 0.01

Epoch = 100 Loops

Output Node = 3

Vigilance = 0.1

4.3 ข้อมูลข่าว Reuters – 21578

ข้อมูลข่าว Reuters–21578 เป็นข้อมูลข่าวของสำนักข่าว Reuters โดยข้อมูลชนิดนี้ได้มีผู้รวบรวมและได้จัดแยกหมวดหมู่ของข่าวไว้แล้ว โดย Reuters Ltd. (Sam Dobbins, Mike Topliss, Steve Weinstein) และ Carnegie Group, Inc. (Peggy Andersen, Monica Cellio, Phil Hayes, Laura Knecht, Irene Nirenburg) ในปี ค.ศ. 1987 ประวัติย่อยของข้อมูลชุดนี้มีดังต่อไปนี้

ในปี ค.ศ. 1990 เอกสารข่าว Reuters ได้ถูกใช้งานเพื่อจุดประสงค์งานวิจัยด้านการค้นคืนสารสนเทศ (Information Retrieval) โดยมี W. Bruce Croft แห่งคณะ Computer and Information Science ที่มหาวิทยาลัย Massachusetts at Amherst. การรวบรวมเริ่มแรก ทำโดย David D. Lewis และ Stephen Harding ในปี ค.ศ. 1990 ณ ห้องปฏิบัติการ Information Retrieval Laboratory.

จากนั้นใน ปี ค.ศ. 1991-1992 นาย David D. Lewis และนาย Peter Shoemaker จาก Center for Information and Language Studies ที่ University of Chicago. ได้จัดทำข้อมูลข่าว Reuters เพิ่มเติมอีกชิ่งมีชื่อ Dataset คือ Reuters-22173, Distribution 1.0 ซึ่งได้นำเผยแพร่ครั้งแรกในเดือนมกราคม ปี ค.ศ. 1993 ผ่าน FTP ที่ศูนย์ Intelligent Information Retrieval (W. Bruce Croft, Director) แห่งมหาวิทยาลัย Massachusetts at Amherst.

ในปี ค.ศ. 1996 ณ งานสัมมนาวิชาการ ACM SIGIR '96 conference ช่วงเดือนสิงหาคม กลุ่มนักวิจัย Text Categorization ได้ปรึกษาหารือในหัวข้อเรื่อง "how published results on

Reuters-22173 could be made more comparable across studies." ผลสรุปคือการออกเวอร์ชันของชุดข้อมูลข่าว Reuters ใหม่ที่มีรูปแบบชัดเจน ลดความคลุมเครือในรูปแบบเอกสารลง (less ambiguous formatting)

ดังนั้น Steve Finch และ David D. Lewis จึงได้ทำการปรับปรุงแก้ไข (cleanup) ข้อมูลข่าว Reuters ตั้งแต่เดือนกันยายน จนถึงเดือนพฤศจิกายน ในปี ค.ศ. 1996 โดยใช้รูปแบบการจัดการเอกสารของ Finch's SGML-tagged ผลลัพธ์คือเอกสารข่าวได้ถูกตัดทิ้งไป 595 เอกสาร เนื่องจากมีความซ้ำซ้อนอย่างมาก ดังนั้นในเวอร์ชันใหม่จึงมีเอกสารข่าวเหลืออยู่ 21,578 เอกสาร ข่าว จึงเรียกชุดข้อมูลข่าวชุดใหม่นี้ว่า Reuters-21578 Distribution 1.0. [27]

Reuters-21578 ประกอบด้วยข่าวจำนวน 21,578 เอกสารข่าว จากจำนวน 135 กลุ่มข่าว โดยข่าวแต่ละข่าวจะมีโครงสร้างข้อมูลเป็นไฟล์ SGML ตัวอย่างข่าวแสดงในภาคผนวก ข. จากข้อมูลข่าวที่มีเราจะนำข่าวทั้งหมดมาผ่านขั้นตอนการเตรียมข้อมูลก่อนนำไปใช้งาน ดังนี้

ขั้นตอนที่ 1 แยกเอาเฉพาะข้อความใน Title และใน Body ของทุกๆข่าว และทำ Index ของแต่ละข่าวว่าอยู่ใน Topic ไหน

ขั้นตอนที่ 2 นำตัวเนื้อข่าวที่ได้ (จากแท็ก Body) ทั้งหมดมาหาคำสำคัญ (keyword) ด้วยโปรแกรม copernic summarizer โดยในการหาคำสำคัญของแต่ละข่าวได้กำหนดจำนวนของคำที่ซ้ำไว้ที่ 10 คำ

ขั้นตอนที่ 3 นำ ข้อความ Title และ Keyword ที่ได้มาหา stemming ของคำรวมทั้งตัดคำที่เป็น stop word

จากข่าว Reuters- 21578 ทั้งหมดเราได้เลือกกลุ่มข่าวที่ใช้ทำการทดลองทั้งหมดจำนวน 3, 5 และ 14 กลุ่มข่าวตามลำดับโดยที่กลุ่ม 3 และ 5 กลุ่มข่าวจำนวนข้อมูลที่ใช้เทรนนิ่งและทดสอบได้กำหนดโดยสุ่มขึ้นมา ส่วนข้อมูลจำนวน 14 กลุ่มข่าวชุดข้อมูลการเทรนนิ่งและทดสอบได้ทำตามข้อกำหนดของ ModApet ที่มีรายละเอียดอยู่ในไฟล์ที่มาพร้อมกับไฟล์ข่าว Reuters-21578

ตารางที่ 4.5 แสดงชุดข้อมูลสำหรับใช้เทรนนิ่งและใช้ทดสอบของข้อมูลข่าว Reuters 3 กลุ่ม

กลุ่มข่าว	จำนวนข่าวที่ใช้เรียนรู้	จำนวนข่าวที่ใช้ทดสอบ
acq	331	1932
crude	221	496
grain	215	467
รวม	767	2895

รายละเอียดของชุดข้อมูล Reuters-21578 ที่ใช้ในการทดลองอยู่ในตารางที่ 4.5 โดย ผลที่ได้จากการทดสอบความถูกต้องของตัวโมเดล แสดงในตารางที่ 4.6

ตารางที่ 4.6 แสดงผลลัพธ์ของการทดสอบของข้อมูลข่าว Reuters-21578 จำนวน 3 กลุ่ม

โหนดเอ้าท์พุทตัวที่ (เป็นตัวแทนของกลุ่ม)	จำนวนสมาชิกที่ตกในคลัสเตอร์		
	1	2	3
1 (acq)	1762	81	89
2 (crude)	140	324	32
3 (grain)	29	35	403
ค่า F measure = 0.85313			
ค่า Entropy = 0.45886			

Learning Rate (η) = 0.001

Epoch = 81/100 Loops

Output Node = 3

Vigilance = 0.03

ตารางที่ 4.7 แสดงชุดข้อมูลสำหรับใช้เทรนนิ่งและใช้ทดสอบของข้อมูลข่าว Reuters 5 กลุ่ม

กลุ่มข่าว	จำนวนข่าวที่ใช้เรียนรู้	จำนวนข่าวที่ใช้ทดสอบ
earn	355	3181
acq	331	1932
money-fx	239	596
crude	221	496
grain	215	467
รวม	1361	6674

ตารางที่ 4.8 แสดงผลลัพธ์ของการทดสอบของข้อมูลข่าว Reuters-21578 จำนวน 5 กลุ่ม

โหนดเอ้าท์พุทตัวที่ (เป็นตัวแทนของกลุ่ม)	จำนวนสมาชิกที่ตกในคลัสเตอร์				
	1	2	3	4	5
1 (money-fx)	148	127	1486	119	52
2 (crude)	51	17	4	54	370
3 (grain)	23	422	14	7	1
4 (acq)	569	7	9	11	0
5 (earn)	102	46	64	2789	180
ค่า F measure = 0. 84963					
ค่า Entropy = 0.50952					

Learning Rate (η) = 0.001

Epoch = 136/300 Loops

Output Node = 5

Vigilance = 0.03

ตารางที่ 4.9 แสดงชุดข้อมูลสำหรับใช้เทรนนิ่งและใช้ทดสอบของข้อมูลข่าว Reuters 14 กลุ่ม

กลุ่มข่าว	จำนวนข่าวที่ใช้เรียนรู้	จำนวนข่าวที่ใช้ทดสอบ
1.earn	2433	727
2. acq	1362	492
3. money-fx	420	93
4. grain	348	73
5. crude	306	134
6. trade	313	90
7. interest	254	70
8. ship	176	64
9. wheat	169	27
10. corn	141	24
11. dlr	87	18
12. money-supply	75	13
13. oilseed	107	24
14. sugar	107	21
รวม	6298	2866

ตารางที่ 4.10 แสดงผลลัพธ์ของการทดสอบของข้อมูลข่าว Reuters-21578 จำนวน 14 กลุ่ม

โหนดเอ้าท์พุทตัวที่ (เป็นตัวแทนของกลุ่ม)	จำนวนสมาชิกที่ตกในคลัสเตอร์													
	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1(trade, interest)	1	**646**	30	13	4	12	4	0	5	0	7	5	0	0
2 (acq)	25	3	2	**231**	24	13	3	5	99	0	16	57	14	0
3 (-)	20	7	3	0	**31**	1	14	0	7	0	6	1	0	3
4 (earn)	10	1	0	0	3	5	1	20	0	**25**	1	0	0	7
5 (money-fx)	10	6	0	1	4	**89**	0	19	0	0	4	0	0	1
6 (crude)	**47**	1	0	1	3	1	1	0	1	5	1	0	0	29
7 (-)	**35**	1	1	0	15	0	10	0	0	0	4	0	0	4
8 (ship, wheat)	12	0	1	0	0	15	0	1	**32**	2	1	0	0	0
9 (-)	2	0	**15**	0	1	1	0	0	0	4	1	0	0	3
10 (grain, oilseed)	4	1	0	0	1	1	2	**8**	0	4	1	0	0	2
11 (-)	0	1	0	0	**14**	0	2	0	0	0	0	0	1	0
12 (-)	0	1	2	0	3	1	0	0	0	0	0	0	0	**6**
13 (-)	2	0	0	0	3	3	0	4	0	5	1	0	0	**6**
14 (money-supply, sugar)	3	0	0	0	0	0	1	1	0	2	0	0	**14**	0
ค่า F measure = 0.66189														
ค่า Entropy = 0.85547														

Learning Rate (η) = 0.001

Epoch = 275/300 Loops

Output Node = 14

Vigilance = 0.03

4.4 สรุปผลการทดลอง

จากผลการทดลองตารางที่ 4.2 เป็นผลการทดลองกับชุดข้อมูลตัวอักษรผล Synthesized Alphabet Documents ที่ได้คือค่า F- measure เท่ากับ 0.99009 และ ค่า Entropy เท่ากับ 0.044368 และจากผลการทดลองตารางที่ 4.4 เป็นผลการทดลองกับชุดข้อมูลตัวอักษรผล

Synthesized Text Documents ที่ได้คือค่า F- measure เท่ากับ 0.95561 และ ค่า Entropy เท่ากับ 0.16297 ซึ่งเป็นค่าที่แสดงว่าประสิทธิภาพของการตัวอัลกอริทึมนั้น สามารถจัดกลุ่มกับข้อมูลที่เรากำหนดนั้นได้ดีมากมีความถูกต้องถึงประมาณ 95-99 เปอร์เซ็นต์

ในส่วนของชุดข้อมูลข่าว Reuters-21578 จากตารางที่ 4.6 เป็นผลการทดลองของข้อมูล 3 กลุ่มซึ่งผลที่ได้ ค่า F-measure เท่ากับ 0.85313 และ ค่า Entropy เท่ากับ 0.45886 ซึ่งประสิทธิภาพของการจัดกลุ่มกับข้อมูลอยู่ในเกณฑ์ที่ดีมีความถูกต้องประมาณ 85 เปอร์เซ็นต์และสามารถระบุโหนดเอ้าท์พุทที่เป็นตัวแทนของแต่ละกลุ่มได้อย่างชัดเจน สำหรับข้อมูล 5 กลุ่มตามตารางที่ 4.8 นั้นที่ได้ ค่า F-measure เท่ากับ 0. 84963 และ ค่า Entropy เท่ากับ 0.50952 ซึ่งประสิทธิภาพของการจัดกลุ่มกับข้อมูลยังถือว่าอยู่ในเกณฑ์ที่ดีมีความถูกต้องประมาณ 85 เปอร์เซ็นต์ และสามารถระบุโหนดเอ้าท์พุทที่เป็นตัวแทนของแต่ละกลุ่มได้อย่างชัดเจนเช่นเดียวกับข้อมูล 3 กลุ่ม และ สุดท้ายสำหรับข้อมูล 14 กลุ่ม ซึ่งเป็นชุดข้อมูลขนาดใหญ่ผลที่ได้จากตารางที่ 4.10 ได้ ค่า F-measure เท่ากับ 0.66189 และ ค่า Entropy เท่ากับ 0.85547 ประสิทธิภาพของการจัดกลุ่มกับข้อมูลอยู่ในเกณฑ์ที่พอใช้ด้วยจำนวนเปอร์เซ็นต์ F-Measure 0.66 และจากผลการทดลองของข้อมูล 14 กลุ่ม นี้ยังพบอีกว่าบางโหนดเอ้าท์พุทไม่สามารถระบุได้ว่าเป็นตัวแทนของกลุ่มข้อมูลกลุ่มใด เนื่องจากข้อมูลที่ใช้ในการทดลองมีการทับซ้อนกันมาก ซึ่งจะเห็นได้ค่า Entropy ที่สูงถึง 0.85547

ดังนั้นจากผลการทดลองทั้งหมด โครงข่ายประสาทเทียมนี้มีความสามารถแบ่งกลุ่มข้อมูลได้ดีเมื่อข้อมูลนั้นเป็นข้อมูลที่มีลักษณะ Less-Overlapping Dataset และข้อดีอีกประการหนึ่งของโครงข่ายประสาทเทียมนี้คือลดขั้นตอนการเตรียมข้อมูลลงได้ เนื่องจากข้อมูลที่ผ่านการ Extraction ได้เป็นกลุ่มคำที่ใช้เป็น Identifier ของแต่ละเอกสาร ซึ่งกลุ่มคำนี้สามารถส่ง (Present) เข้าสู่โครงข่ายประสาทเทียมได้โดยตรง โดยไม่ต้องผ่านขั้นตอนการ Map หรือ Transformation ด้วยวิธีการทั่วไปของ Document Representation ทั้งแบบ Binary, Term-Frequency และ TFxIDF เพื่อให้ได้ Feature Space ของเอกสารที่เก็บข้อมูลเชิงปริมาณ (Quantitative Feature Space) นอกจากนี้ยังได้นำผลการทดลองที่ได้เปรียบเทียบกับผลการทดลองของ TPCLNN [18] ซึ่งได้ทำการจัดแบ่งกลุ่มเอกสารข่าว Reuter-21578 และใช้หลักการทำงานแบบ Unsupervised Learning Neural Network ใน 4 ส่วนดังนี้

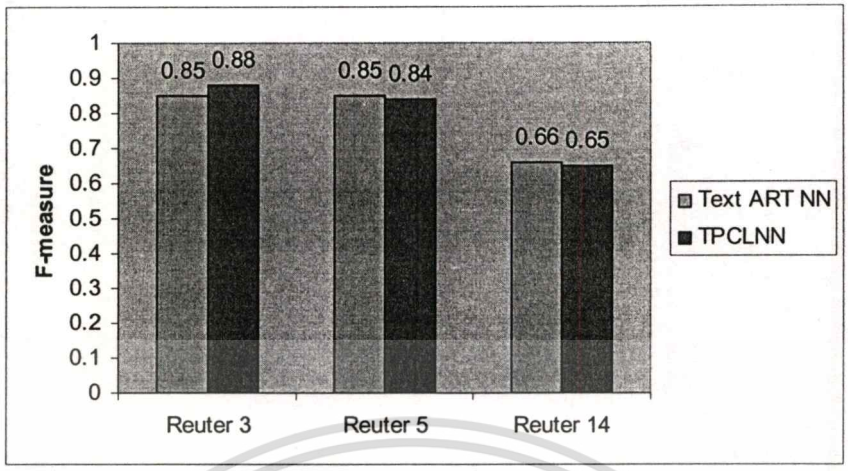
ส่วนที่ 1 เปรียบเทียบในส่วนผลลัพธ์ที่ได้จากการวัดประสิทธิภาพด้วยค่า F-Measure

ส่วนที่ 2 เปรียบเทียบในส่วนผลลัพธ์ที่ได้จากการวัดประสิทธิภาพด้วยค่า Entropy

ส่วนที่ 3 เปรียบเทียบในส่วนผลลัพธ์ที่ได้จากการวัดจำนวนรอบการฝึก Epoch

ส่วนที่ 4 เปรียบเทียบในส่วนการทำงานของอัลกอริทึม

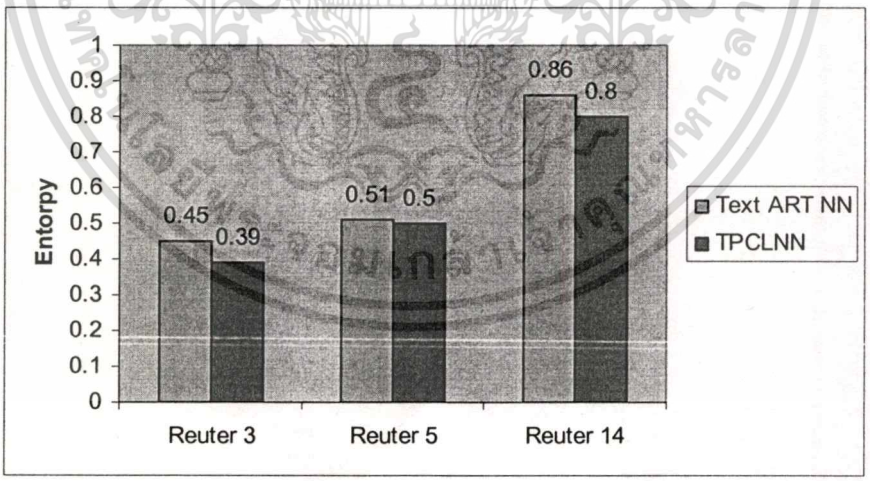
ส่วนที่ 1 เปรียบเทียบในส่วนผลลัพธ์ที่ได้จากการวัดประสิทธิภาพด้วยค่า F-Measure



กราฟที่ 4.1 แสดงการเปรียบเทียบค่า F-Measure ระหว่าง Text ART Neural Network และ Text Processing Competitive Learning Neural Network

ในส่วนที่ 1 นี้จากกราฟที่ 4.1 ประสิทธิภาพของการจัดกลุ่มเอกสารด้วย F-Measure ของ Text ART Neural Network มีประสิทธิภาพดีกว่า แต่ยังคงใกล้เคียงกัน

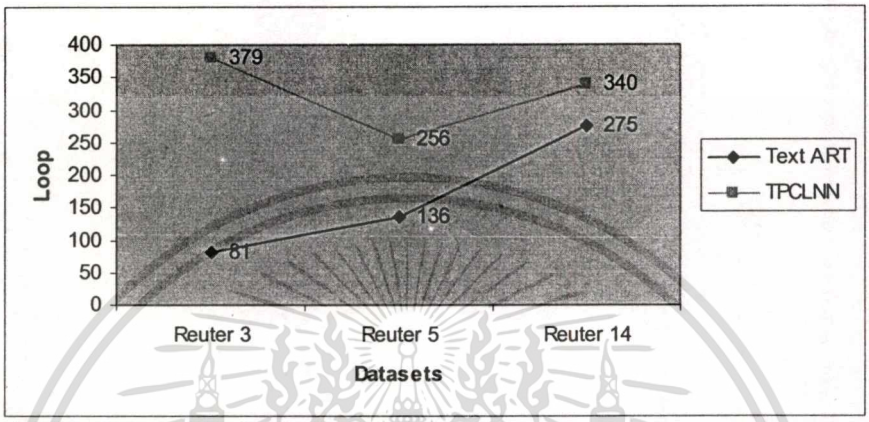
ส่วนที่ 2 เปรียบเทียบในส่วนผลลัพธ์ที่ได้จากการวัดประสิทธิภาพด้วยค่า Entropy



กราฟที่ 4.2 แสดงการเปรียบเทียบค่า Entropy ระหว่าง Text ART Neural Network และ Text Processing Competitive Learning Neural Network (TPCLNN)

ในส่วนที่ 2 นี้จากกราฟที่ 4.2 ประสิทธิภาพของการจัดกลุ่มเอกสารด้วย Entropy ของ TPCLNN ทีประสิทธิภาพดีกว่า แต่ยังคงใกล้เคียงกัน ดังนั้นจากส่วนที่ 1 และส่วนที่ 2 สรุปได้ว่า ประสิทธิภาพการจัดกลุ่มเอกสารของ Text ART Neural Network และ TPCLNN มีประสิทธิภาพดีใกล้เคียงกัน

ส่วนที่ 3 เปรียบเทียบในส่วนผลลัพธ์ที่ได้จากการวัดจำนวนรอบการฝึก Epoch



กราฟที่ 4.3 แสดงการเปรียบเทียบจำนวน Epoch ระหว่าง Text ART Neural Network และ Text Processing Competitive Learning Neural Network (TPCLNN)

ในส่วนที่ 3 สรุปได้จากกราฟที่ 4.3 ดังนี้ รอบการฝึก (Epoch) ของ Text ART Neural Network ใช้จำนวนรอบน้อยกว่า โดยขั้นตอนการทำงานของ Text ART Neural Network เปรียบเทียบขั้นตอนการทำงานของ TPCLNN จะอธิบายไว้ในส่วนที่ 4 ต่อไป

ส่วนที่ 4 เปรียบเทียบในส่วนการทำงานของอัลกอริทึม

ในส่วนที่ 4 ในการคำนวณการทำงานอัลกอริทึมของ Text ART Neural Network และ TPCLNN ใช้หลักการทำงานบนพื้นฐานของการเรียนรู้แบบไม่มีผู้สอน (Unsupervised Learning) ร่วมกับหลักการวัดความแตกต่างของเอกสาร ซึ่งแต่ละครั้งในการทำงานจะมีเพียงเอาท์พุตโหนดเดียวเท่านั้นที่จะเป็นผู้ชนะและได้รับการปรับ Weights แต่มีความแตกต่างกัน ดังแสดงในตารางที่ 4.11

ตารางที่ 4.11 เปรียบเทียบการทำงานอัลกอริทึมของ Text ART Neural Network และ TPCLNN

	Text ART Neural Network	TPCLNN
Similarity/Dissimilarity Measure	Similarity	Dissimilarity
Winning nodes	Max{Y_j}	Min{Y_j}
Update weights	Weight ของโหนดชนะ	Weight ของโหนดชนะ
Control degree of similarity	มี	ไม่มี

การเลือกโหนดชนะของ Text ART Neural Network จะเลือกโหนดที่มีค่าสูงที่สุด เนื่องจากใช้หลักการจัดแบ่งเอกสารใช้หลักการ Similarity Measure for Symbolic Objects ซึ่งหลักการนี้ Objects ที่มีความคล้ายกันมาก จะมีค่า Similarity สูง ในขณะที่การเลือกโหนดชนะของ TPCLNN จะเลือกโหนดที่มีค่าน้อยที่สุด เนื่องจากใช้หลักการจัดแบ่งเอกสารใช้หลักการ Dissimilarity Measure for Symbolic Objects ซึ่งหลักการนี้ Objects ที่มีความคล้ายกันมาก จะมีค่า Dissimilarity ต่ำ

แต่มีความแตกต่างกันคือ Text ART Neural Network มีคุณสมบัติการควบคุมระดับความคล้ายคลึงของการจัดกลุ่ม (Control degree of similarity) ซึ่งกำหนดโดยค่า Vigilance Threshold และทำให้ Text ART Neural Network มี Weights 2 ชุด คือ Bottom-up Weights และ Top-down Weights โดยที่เอกสารอินพุตจะต้องถูกเปรียบเทียบความคล้ายคลึงด้วย Similarity Measure for Symbolic Objects เป็นจำนวน 2 ครั้ง ต่อ 1 รอบการฝึก โดยรอบที่หนึ่งหาโหนดผู้ชนะ คำนวณด้วยสูตร Similarity Measure for Symbolic Objects ระหว่างเอกสารอินพุตและ Bottom-up Weight และรอบที่สอง หาความตรงกันในเลือกโหนดผู้ชนะด้วยสูตร Similarity Measure for Symbolic Objects ระหว่างเอกสารอินพุตและ Top-down Weight โดยมีค่า Vigilance Threshold ทำหน้าที่ควบคุมระดับความคล้ายกันของเอกสารอินพุตกับ Top-down Weight ของนิวรอลเน็ตเวิร์ค ในขณะที่ TPCLNN มี Weights 1 ชุด เอกสารอินพุตจึงถูกเปรียบเทียบความคล้ายคลึงด้วย Dissimilarity Measure for symbolic objects เป็นจำนวน 1 ครั้ง ต่อ 1 รอบการฝึก โดย หาโหนดผู้ชนะ คำนวณด้วยสูตร Dissimilarity Measure for Symbolic Objects ระหว่างเอกสารอินพุตและ Weight ของนิวรอลเน็ตเวิร์ค ดังนั้นจำนวนขั้นตอนในการคำนวณของ Text ART Neural Network จึงมีจำนวนขั้นตอนมากกว่าเมื่อเปรียบเทียบกับจำนวนขั้นตอนในการคำนวณของ TPCLNN แต่เมื่อคิดจำนวนรอบการฝึก (Training Epoch) จากผลการทดลองText ART Neural Network ใช้จำนวนรอบการฝึกน้อยกว่า

บทที่ 5
สรุปผลการวิจัยและข้อเสนอแนะ

5.1 สรุปผลงานวิจัย

งานวิจัยนี้ ได้นำเสนอวิธีการจัดกลุ่มเอกสารโดยใช้หลักการของโครงข่ายประสาทเทียมที่เรียนรู้แบบหาผู้ชนะแบบ Adaptive Resonance Theory ร่วมกับการหาความคล้ายคลึงกันของเอกสารแบบ Similarity Measure for Symbolic Objects จากหลักการทั้งหมดที่กล่าวมาทำให้โครงข่ายประสาทเทียมที่พัฒนาขึ้นมานั้นสามารถรับข้อมูลอินพุทที่เป็นข้อความ (text) เข้าไปประมวลผลได้โดยตรง โดยที่ไม่จำเป็นต้องแปลงอินพุท ให้เป็นข้อมูลในเชิงตัวเลขก่อนแต่อย่างใด ซึ่งวิธีการนี้ช่วยลดปัญหาของการเกิด High Dimension ของคำเมื่อใช้วิธีการของ Vector Space Model ในการแบ่งกลุ่มที่มีข้อมูลขนาดใหญ่ ซึ่ง Feature ที่ถูกเลือกสำหรับการนำเข้าในโครงข่ายประสาทเทียมประกอบด้วย กลุ่มคำของชื่อเรื่อง (Set of Title)และกลุ่มคำของคำสำคัญ (Set of Keyword) ของเอกสาร ดังนั้นจำนวน Input Nodes ของโครงข่ายประสาทเทียมจึงเท่ากับ 2 Input Nodes

เพื่อกำหนดระดับความคล้ายคลึงของเอกสารในการจัดแบ่งกลุ่ม โครงข่ายประสาทเทียมนี้จึงใช้ค่า Vigilance เป็นตัวแปรที่กำหนดค่า Threshold ดังนั้นโครงข่ายประสาทเทียมนี้จึงมีการทำงานหลัก 2 ส่วนคือ

- ส่วนแรกคือ Competitive ส่วนนี้ทำหน้าที่หาโหนดผู้ชนะตามกฎของ Winning-Take-All โดยโหนดผู้ชนะ (Winning-Node) คือโหนดที่มีค่าเอาท์พุตสูงที่สุด และให้โหนดที่ค่าเอาท์พุตรองลงมาเป็นโหนดคู่แข่ง (Candidate Node) โดยการคำนวณค่าความคล้ายคลึงระหว่างเอกสารอินพุตกับ Bottom-Up Weight
- อีกส่วนคือ Resonance ส่วนนี้จะทำหน้าที่ทดสอบเงื่อนไขที่สำคัญ โดยการคำนวณค่าความคล้ายคลึงระหว่างเอกสารอินพุตกับ Top-Down Weight ซึ่งจะเก็บค่าคำนวณที่ได้ลงในตัวแปร Z ถ้าค่า Z มีค่าสูงกว่าค่า Vigilance แสดงว่ามีความสอดคล้องกับเงื่อนไข (Resonance) จึงยอมรับให้เอาท์พุตโหนดนั้นเป็นโหนดผู้ชนะ และทำการปรับ Weight ของเอาท์พุตโหนดที่ชนะนั้นตามเงื่อนไขที่กล่าวไว้ในบทที่ 3

ในส่วนของการวัดประสิทธิภาพของผลการทดลองในงานวิจัยนี้ได้ใช้ค่า Entropy และค่า F- Measure เป็นตัวชี้วัดประสิทธิภาพของการจัดกลุ่ม ซึ่งจากผลการทดลองในบทที่ 4 เมื่อนำผลที่

ได้มาเปรียบเทียบกับผลจากงานวิจัยโครงข่ายประสาทเทียมที่เรียนรู้แบบหาผู้ชนะ (A Text Processing Competitive Learning Neural Network (TPCLNN)) [18] ซึ่งสามารถให้เปอร์เซ็นต์ของ F-Meausre เท่ากับ 65% พบว่า Text ART Clustering มีประสิทธิภาพดีกว่า TPCLNN

5.2 บทวิเคราะห์

จากการศึกษาการจัดแบ่งกลุ่มเอกสารโดยใช้ Text ART Neural Network พบว่าวิธีการนี้มีข้อดีเหนือวิธีการอื่นๆ ในหลายประเด็น

5.2.1 ลดความซับซ้อนของขั้นตอน Pre-processing

เนื่องจากขั้นตอนของการทำ Document Representation ของงานวิจัยนี้ แตกต่างจากขั้นตอนการทำ Document Representation ทั่วไป ซึ่งในงานวิจัยนี้ ลักษณะข้อมูลได้จาก Document Representation แบบ Title x Keyword ประกอบด้วยกลุ่มคำของชื่อเรื่องและกลุ่มคำของคำสำคัญของเอกสาร จึงสามารถส่งเข้าโครงข่ายประสาทเทียมแบบ Text Adaptive Resonance Theory ได้โดยตรงซึ่งลดขั้นตอนการแปลง Textual Values เป็น Numerical Values และยังช่วยลดปัญหา Input Dimensional Feature Space ลงได้ด้วยเช่นกัน

5.2.2 ค่า Vigilance

เนื่องจากค่าเอาท์พุตโหนดของโครงข่ายประสาทเทียมอาจมีค่าที่สูงใกล้เคียงกัน เพื่อเลือกเอาท์พุตโหนดที่มีความคล้ายคลึงและสอดคล้องกับเอกสารมากที่สุด ค่า Vigilance จึงเป็นตัวแปรที่สำคัญที่ใช้สำหรับกำหนด Threshold ของการแบ่งกลุ่มเอกสารได้เป็นอย่างดี โดยค่า Vigilance มีค่าระหว่าง 0 จนถึง 1 ค่า Vigilance ที่อยู่ใกล้ 1 หมายถึงความคล้ายคลึงและสอดคล้องของการแบ่งกลุ่มระดับสูง นอกจากนี้ค่า Vigilance มีผลกับจำนวนกลุ่มของ Cluster

5.2.3 ตัวแทนกลุ่มเอกสาร (Centroid)

เนื่องจาก Bottom-up weight และ Top-down weight ของโครงข่ายประสาทเทียมนี้สามารถเก็บคำที่ได้จากขั้นตอนปรับ Weight ดังนั้นสมาชิกของ Weight ของแต่ละเอาท์พุตโหนดสามารถเป็นคำสำคัญของแต่ละกลุ่มเอกสารและใช้เป็นตัวแทนกลุ่ม (Centroid) ได้ พร้อมทั้งระบุระดับความสัมพันธ์ของคำใน Weight แต่ละชุดได้

5.2.4 การใช้งานได้อย่างกว้างขวาง

วิธีการดังกล่าวสามารถใช้ประยุกต์ได้กับระบบจัดการข้อมูลเอกสารอื่นๆ ได้อย่างกว้างขวางเนื่องจากข้อมูลเอกสารในปัจจุบันมีปริมาณมากมหาศาล และระบบจัดการข้อมูลเอกสารมีหลากหลายระบบ เช่น ระบบค้นคืนสารสนเทศ (Information Retrieval Systems) ระบบ

คลังข้อมูลเอกสาร (Document Warehouse Systems) และระบบการจัดหมวดหมู่เอกสาร (Text Categorization Systems) เป็นต้น เพียงเพิ่มเติมวิธีการนี้ลงในระบบจัดการข้อมูลเอกสารดังกล่าวที่ต้องการ

5.3 ข้อเสนอแนะและแนวทางในการทำวิจัยต่อ

เนื่องจากในงานวิจัยนี้ได้ทดลองกับเฉพาะข้อมูลที่เป็นข้อมูลข่าว (Text Document) ด้วยวิธีการ Text ART Neural Network จึงน่าจะมีการนำวิธีการดังกล่าวมาใช้กับ Web Document เช่น เอกสารที่อยู่ในรูปแบบ HTML หรือ Ascii Text และยังมีความเป็นไปได้ในการนำอัลกอริทึมนี้ไปประยุกต์เพื่อพัฒนาเป็นเครื่องมือช่วยในระบบค้นคืนสารสนเทศ (Information Retrieval) แทนที่ผู้ใช้หรือระบบค้นคืนสารสนเทศจะต้องค้นหาเอกสารทั้งหมดที่มีอยู่ในระบบและมีความสัมพันธ์กับคำสืบค้น ผู้ใช้สามารถดูผลจาก Document Cluster เพื่อหาเอกสารที่เกี่ยวข้องได้ Systems) วิธีการนี้จึงน่าจะช่วยในส่วนการลดจำนวนปริมาณค้นหา (Scanning) ในระบบดังกล่าวลงได้เป็นอย่างดี

นอกจากนี้ยังมีความเป็นไปได้ในการนำอัลกอริทึมนี้ไปประยุกต์เพื่อพัฒนาเป็นเครื่องมือของระบบคลังข้อมูลเอกสาร (Document Warehouse) ในส่วนของ Document Loading สำหรับเอกสารจากหลายๆ แหล่งที่มา ซึ่งในระบบคลังข้อมูลเอกสารอาจมีจำนวนเอกสารในระบบอยู่จำนวนมากมหาศาล ดังนั้นการทำ Document Clustering เพื่อระบุกลุ่มเอกสารที่คล้ายกันในระบบคลังข้อมูลเอกสาร จึงน่าจะช่วยในส่วนการลดจำนวนปริมาณค้นหาเอกสารในระบบดังกล่าว และช่วยในส่วน Pre-Processing ของการจัดหมวดหมู่เอกสาร (Text Categorization) สุดท้ายนี้แนวทางในการพัฒนางานวิจัยต่อคือวิธีการแนวใหม่ที่สามารถจัดแบ่งกลุ่มข้อมูลเอกสารที่มีลักษณะ Overlap สูงได้

เอกสารอ้างอิง

[1] Carpenter, G. A., and Grosssberg, S. "Associative learning, adaptive pattern recognition and cooperative decision making by neural networks," Hybrid and Optical Computing, SPIE, 1986.

[2] Carpenter, G. A., and Grosssberg, S. (1987a). "A massively parallel architecture for a self-organizing neural pattern recognition machine," Computer Vision, Graphics, and Image Processing, 1987.

[3] Carpenter, G. A., and Grosssberg, S.(1987b). "ART 2: Stable self-organization of pattern recognition codes for analog input patterns," (with G.A. Carpenter). Applied Optics, 1987.

[4] Carpenter, G. A., and Grosssberg, S. "The ART of adaptive pattern recognition by a self-organizing network," Computer, 1988.

[5] Han Jiawei and Kamber Micheline. Data Mining: Concepts and Technique. Margan Kaufmann Publishers, New York, 2001.

[6] Salton Gerard. Automatic Text Processing: the transformation, analysis, and retrieval of information by computer. Addison Wesley Publishing Company, New York, 1989.

[7] Jacek M. Zurada. Introduction to Artificial Neural Systems. West Publishing Company, New York, 1992.

[8] Fausett Lanrene. Fundamentals of Neural Networks Architecture, Algorithms and Application. Prentice Hall International, New Jersey, 1994.

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

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

[11] T.V. Ravi and K.C. Gowda. "Clustering of Symbolic Objects Using Gravitational Approach." IEEE Trans. On Syst. , Man, Cybern., vol. 29, 1999. no. 6, pp. 888-894.

[12] Hsin-Chang Yang and Chung-Hong Lee. "Automatic Category Generation for Text Documents by Self-Organizing Maps." IEEE Trans, 2000. pp. 581-586

[13] Michael Steinbach , George Karypis and Vipin Kumar, "A Comparison of Document Clustering Techniques," Proceedings of Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Boston. MA. USA, 2000.

[14] M. Benkhalifa and A. Bensaid, "Text Categorization using the Semi-Supervised Fuzzy c-Mean Algorithm," IEEE, pp. 561-565, 1999

[15] King-Ip Lin and Ravikumar Kondadadi, "A Similarity-Based Soft Clustering Algorithm For Documents," IEEE, 2001

[16] Florian Beil, Martin Ester and Xiaowei Xu, "Frequent Term-Based Text Clustering," ACM SIGKDD 02, 2002

[17] ทรงพล ชุติพงศ์พัฒนกุล, "เท็กโปรเซสซิ่งโคโฮเนนนิวรอลเน็ตเวร์คโดยใช้กระบวนการเรียนรู้แนวใหม่", วิทยานิพนธ์วิทยาศาสตรมหาบัณฑิต สาขาเทคโนโลยีสารสนเทศ บัณฑิตวิทยาลัย, สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง, 2546.

[18] สมคิด แสนเสนาะ, "การแบ่งกลุ่มเอกสารโดยใช้เทคนิคการประมวลผลข้อความด้วยโครงข่ายประสาทเทียมที่เรียนรู้แบบหาผู้ชนะ", วิทยานิพนธ์วิทยาศาสตรมหาบัณฑิต สาขาเทคโนโลยีสารสนเทศ บัณฑิตวิทยาลัย, สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง, 2547.

[19] Louis Massey , "On the quality of ART1 text clustering," ACM Neural Networks, Vol. 16, Special issue: Advances in neural networks research -- IJCNN'03 , pp. 771 – 778, 2003.

[20] G.A. Carpenter and S. Grossberg. "Adaptive Resonance Theory (ART)," In: Handbook of Brain Theory and Neural Networks, Ed: Arbib M.A. , MIT Press, 1995.

[21] A.K. Jain, M.N. Murty and P. J Flyynn. "Data Clustering: A review", ACM Computing Surveys, Vol. 31, No. 3, Sept 1999.

[22] N.K. Bose and P. Liang. Neural Network Fundamentals with Graphs, Algorithms, and Applications, McGRAW-HILL International Editions, Electrical Engineering Series, 1996.

[23] Tom M. Mitchell. Machine Learning. McGRAW-HILL International Editions, Computer Sciences Series, 1996.

[24] James A. Freeman and David M. Skapura. Neural Networks Algorithms, Applications, and Programming Techniques. Addison-Wesley, 1992.

[25] D.Sullivan., Document Warehousing and Text Mining, John Wiley & Sons, Inc. ISBN: 0- 471-39959-0.

[26] G.Salton, A.Wong,C.S. Yang, "A Vector Space Model for Automatic Indexing", Communications of the ACM, 18(11):613--620, 1971

[27] Lewis D.D., "Reuters-21578 text categorization test collection distribution 1.0.", http://www.daviddlewis.com/.

[28] A. K. Jain and R. C. Dubes, Algorithms for Clustering Data, Prentice Hall, 1988.

[29] "List of Stop Words", http://netra.wustl.edu/~adpol/courses/cs504/lab6/stop-word.lst

[30] Porter, M.F., "An Algorithm For Suffix Stripping," Program 14 (3), no. 3, pp 130-137, July,1980.

ภาคผนวก

ภาคผนวก ก.

แสดงตัวอย่างข่าว Reuters-21578

ก.1 ไฟล์ข่าวก่อนการตัดแท็ก

<REUTERS TOPICS="YES" LEWISSPLIT="TRAIN" CGISPLIT="TRAINING-SET"
OLDID="5679" NEWID="136">
<DATE>26-FEB-1987 17:11:01.51</DATE>
<TOPICS><D>grain</D></TOPICS>
<PLACES><D>usa</D></PLACES>
<PEOPLE></PEOPLE>
<ORGS></ORGS>
<EXCHANGES></EXCHANGES>
<COMPANIES></COMPANIES>
<UNKNOWN>BC-GAO-LIKELY-TO-SHOW-CE 02-26 0126</UNKNOWN>
<TEXT>
<TITLE>GAO LIKELY TO SHOW CERTS MORE COSTLY THAN CASH</TITLE>
<DATELINE>WASHINGTON, Feb 26</DATELINE>
<BODY>A study on grain certificates due out
shortly from the Government Accounting Office (GAO) could show
that certificates cost the government 10 to 15 pct more than
cash outlays, administration and industry sources said.
Analysis that the GAO has obtained from the Agriculture
Department and the Office of Management and Budget suggests
that certificates cost more than cash payments, a GAO official
told Reuters.
GAO is preparing the certificate study at the specific
request of Sen. Jesse Helms (R-N.C.), former chairman of the
senate agriculture committee.

The report, which will focus on the cost of certificates compared to cash, is scheduled to be released in mid March.

The cost of certificates, said the GAO source, depends on the program's impact on the USDA loan program.

If GAO determines that certificates encourage more loan entries or cause more loan forfeitures, then the net cost of the program would go up. However, if it is determined that certificates have caused the government grain stockpile to decrease, the cost effect of certificates would be less.

GAO will not likely suggest whether the certificates program should be slowed or expanded, the GAO official said.

But a negative report on certificates "will fuel the fire against certificates and weigh heavily on at least an increase in the certificate program," an agricultural consultant said.

The OMB is said to be against any expansion of the program, while USDA remains firmly committed to it.

Reuter

</BODY></TEXT>

</REUTERS>

ก.2 ไฟล์ข่าวหลังจากไดตัดแท็กเอาเฉพาะข้อมูลในแท็ก TITLE และ แท็ก KEYWORD

ไฟล์ข่าวหลังจากไดตัดแท็กเอาเฉพาะข้อมูลในแท็ก TITLE และ แท็ก KEYWORD แล้วได้ข้อมูลดังนี้

ส่วนของ TITLE มีดังนี้

GAO LIKELY TO SHOW CERTS MORE COSTLY THAN CASH

ส่วนของ BODY

A study on grain certificates due out

shortly from the Government Accounting Office (GAO) could show that certificates cost the government 10 to 15 pct more than Cash outlays, administration and industry sources said.

Analysis that the GAO has obtained from the Agriculture Department and the Office of Management and Budget suggests that certificates cost more than cash payments, a GAO official told Reuters.

GAO is preparing the certificate study at the specific request of Sen. Jesse Helms (R-N.C.), former chairman of the senate agriculture committee.

The report, which will focus on the cost of certificates compared to cash, is scheduled to be released in mid March.

The cost of certificates, said the GAO source, depends on the program's impact on the USDA loan program.

If GAO determines that certificates encourage more loan entries or cause more loan forfeitures, then the net cost of the program would go up. However, if it is determined that certificates have caused the government grain stockpile to decrease, the cost effect of certificates would be less.

GAO will not likely suggest whether the certificates program should be slowed or expanded, the GAO official said.

But a negative report on certificates "will fuel the fire against certificates and weigh heavily on at least an increase in the certificate program," an agricultural consultant said.

The OMB is said to be against any expansion of the program, while USDA remains firmly committed to it.

ตารางที่ ก.1 แสดงตัวอย่างของคำที่เป็น Stop Words

a about above according across actually adj after afterwards again against all almost alone along
already also although always among amongst an and another any anyhow anyone

anything anywhere are aren't around as at
b be became because become becomes becoming been before beforehand begin beginning behind being below beside besides between beyond billion both but by
c can can't cannot caption co co. could couldn't
d did didn't do does doesn't don't down during
e each eg eight eighty either else elsewhere end ending enough etc even ever every everyone everything everywhere except
f few fifty first five for former formerly forty found four from further
h had has hasn't have haven't he he'd he'll he's hence her here here's hereafter hereby herein hereupon hers herself him himself his how however hundred
i i'd i'll i'm i've ie if in inc. indeed instead into is isn't it it's its itself
l last later latter latterly least less let let's like likely ltd
m made make makes many maybe me meantime meanwhile might million miss more moreover most mostly mr mrs much must my myself
n namely neither never nevertheless next nine ninety no nobody none nonetheless noone nor not nothing now nowhere
o of off often on once one one's only onto or other others otherwise our ours ourselves out over overall own
p per perhaps
r rather recent recently
s same seem seemed seeming seems seven seventy several she she'd she'll she's should shouldn't since six sixty so some somehow someone something sometime sometimes somewhere still stop such

t taking ten than that that'll that's that've the their them themselves then thence there there'd there'll there're there's there've thereafter thereby therefore therein thereupon these they they'd they'll they're they've thirty this those though thousand three through throughout thru thus to together too toward towards trillion twenty two
u under unless unlike unlikely until up upon us used using
v very via
w was wasn't we we'd we'll we're we've well were weren't what what'll what's what've whatever when whence whenever where where's whereafter whereas whereby wherein whereupon wherever whether which while whither who who'd who'll who's whoever whole whom whomever whose why will with within without won't would wouldn't
y yes yet you you'd you'll you're you've your yours yourself yourselves

ตารางที่ ก.2 แสดงตัวอย่างของคำที่ได้หลังจากตัด Stemming ของคำ

Title	keyword
gao, like, show, cert,cost, cash	certificate, gao, cost, loan, agriculture ,government, usda, report, committee

ตารางที่ ก.3 แสดงตัวอย่างของคำที่ตัด stemming และคำที่เป็น stop word

ก่อนตัด stemming	หลังตัด stemming
tons	ton
omits	omit
accepts	accept
minerals	mineral
groups	group
called	call
acquires	acquire

acquires	acquire
drafts	draft
certs	cert
earning	earn
driving	drive
meeting	meet
planning	plan
according	accord
a	-
is	-
am	-
are	-
on	-
in	-
the	-
an	-
it	-
to	-
then	-
that	-
of	-
while	-

ตารางที่ ก.4 แสดงตัวอย่างโครงสร้างของข้อมูลข่าว Reuters-21578 ที่นำเข้าโมเดล

Title	Keyword
data(1,1).text(1,:) = {'bahia'}; data(1,1).text(2,:) = {'cocoa'}; data(1,1).text(3,:) = {'revew'};	data(1,2).text(1,:) = {'dlr'}; data(1,2).text(2,:) = {'york'}; data(1,2).text(3,:) = {'sale'}; data(1,2).text(4,:) = {'cocoa'}; data(1,2).text(5,:) = {'crop'}; data(1,2).text(6,:) = {'mln'}; data(1,2).text(7,:) = {'bag'}; data(1,2).text(8,:) = {'comissaria'}; data(1,2).text(9,:) = {'smith'}; data(1,2).text(10,:) = {'bahia'};

ตารางที่ ก.5 แสดงตัวอย่าง Weight ที่ได้จากการเรียนรู้ (Synthesized Text Document)

Bottom-up weight

Neuron	Weights
Output Node 1	{(3com,1.0),(cisco,0.99), (internet,0.98),(java,1.0), (network,0.99),(protocol,0.98)}
Output Node 2	{(airline,1.0),(bank,1.0), (business,0.98),(car,0.99), (hotel,0.98),(intel,0.0), (network,0.0),(travel,1.0)}
Output Node 3	{(algorithm,1.0),(compute,0.99); (database,0.97),(intel,0.98), (java,1.0),(network,1.0)}

Top-down weight

Neuron	Weights
Output Node1	{(3com,1.0),(cisco,0.99), (internet,0.98),(java,1.0), (network,0.99),(protocol,0.98)}
Output Node 2	{(airline,1.0),(algorithm,0.0), (bank,0.97),(business,0.98), (car,0.99),(database,0.0), (hotel,0.98),(intel,0.0), (network,0.0),(travel,0.99)}
Output Node 3	{(algorithm,1.0),(compute,0.99), (database,0.96),(intel,0.98), (java,1.0),(network,1.0)}

ตารางที่ ก.6 แสดงตัวอย่าง Weight ที่ได้จากการเรียนรู้ (Routers-21578)

Bottom-up weight

Neuron	Weights
Neuron 1	{(complete,0.001),(kraft,0.001),(quaker,0.001),(carling,0.02), (elders,0.002) (buyout,0.003),(disclosure,0.003),(favors,0.003),(period,0.003),(shad,0.003 (shortening,0.003),(agree,0.004),(stars,0.004),(store,0.004), (firm,0.005),(ups,0.005),(group,0.006),(bank,0.008),(bid,0.013), (acquire,0.025),(pct,0.029), (stake,0.116),(unit,0.171),(merge,0.221),(buy,0
Neuron 2	{(expansion,0.001),(analyst,0.002),(good,0.002),(guinea,0.002), (papua,0.002),(prospect,0.002),(arab,0.003),(heavy,0.003),(supplies,0.003 (term,0.004), (bombings,0.004),(colombian,0.004),(group,0.004), (pipelines,0.004), (suspend,0.004),(boosts,0.005), (output,0.005),(post,0.0 (acquisition,0.01),(opec,0.011), (american,0.013),(crude,0.018), (price,0.02 (pct,0.157),(oil,1.0)}
Neuron 3	{(midwest,0.001),(movement,0.001), (slow,0.001),(carloadings,0.002), (fall,0.002),(ahead,0.003),(cbt,0.003),(look,0.003), (spring,0.003), (traders,0.003),(lanka,0.04),(overnight,0.04),(sri,0.04),(tendering,0.04), (crop,0.05),(main,0.05),(projection,0.05),(reduce,0.05),(rice,0.05), (plantings,0.09),(maize,0.016),(set,0.026),(export,0.229),(usda,0.398), (corn,0.477),(grain,0.671),(wheat,0.964)}

Top-down weight

Neuron	Weights
Neuron1	{(group,0.001),(kraft,0.001), (quaker,0.001),(unit,0.001), (merge,0.002), (carling,0.002),(elders,0.002),(disclosure,0.003),(favors,0.003),(period,0 (shad,0.003),(shortening,0.003),(acquire,0.004),(agree,0.004), (stars,0.0 (store,0.004), (firm,0.004),(stake,0.005),(ups,0.005),(pct,0.008), (buy,0.0
Neuron 2	{ (expansion,0.001),(opec,0.001), (analyst,0.002),(good, 0.002), (g 0.002), (papua, 0.002),(prospect, 0.002),(american,0.003), (arab, 0.003), 0.003), (heavy, 0.003),(supplies, 0.003),(term, 0.003),(bombings, 0.003), (colombian,0.004),(group, 0.004), (pipelines, 0.004),(suspend, 0.004), (acquisition,0.005),(boosts,0.005), (output,0.005),(pct,0.005), (oil,0.916)
Neuron 3	{ (country,0.001),(midwest,0.001), (movement,0.001),(slow,0.001), (carloadings,0.002),(fall,0.002),(ahead,0.003),(cbt,0.003), (look,0.003), (spring,0.003), (traders,0.003),(lanka,0.004),(overnight,0.004), (plantings,0.004), (sri,0.004),(tendering,0.005), (crop,0.005),(main,0.005 (projection,0.005),(reduce,0.005),(rice,0.005),(maize,0.005), (export,0.0 (usda,0.006),(oil,0.009),(grain,0.116),(wheat,0.238)}

ภาคผนวก ข.

แสดงตัวอย่างการทำงานของ Text ART Neural Network

Training

Set Initialze Degree of botton-up weights = 0.500

Set Initialze Degree of top-down weights = 1.000

Set Learning Rate parameter = 0.010

Set Learning Loop parameter = 100

Set Vigilance parameter = 0.100

Number of input units = 2

Number of interface units = 2

Number of output units = 3

X(35) Y(1) NET = 2.35000 , Z = 0.15476 Loop(1) <<< NOT RESET || COMPETITIVE COMPLETE >>>

X(57) Y(1) NET = 1.87500 , Z = 0.34286 Loop(1) <<< NOT RESET || COMPETITIVE COMPLETE >>>

X(83) Y(1) NET = 1.99107 , Z = 0.50000 Loop(1) <<< NOT RESET || COMPETITIVE COMPLETE >>>

X(64) Y(1) NET = 2.17857 , Z = 0.70833 Loop(1) <<< NOT RESET || COMPETITIVE COMPLETE >>>

X(16) Y(1) NET = 1.00000 ,Z = 0.00000 Loop(1) ---> RESET :: FIND CANDIDATE NEURONS

>>>>> X(16) Y(2) Loop(1) !!! ALL UNITS RESET !!! ADD NEW OUTPUT NEURON , Z = 0.00000

<<<<<

X(98) Y(1) NET = 1.82500 ,Z = 0.00000 Loop(1) ---> RESET :: FIND CANDIDATE NEURONS

X(98) Y(2) NET = 1.30000 , Z = 0.17143 Loop(1) <<< NOT RESET || COMPETITIVE COMPLETE >>>

X(97) Y(2) NET = 1.92857 , Z = 0.30000 Loop(1) <<< NOT RESET || COMPETITIVE COMPLETE >>>

X(32) Y(1) NET = 1.59615 , Z = 0.50000 Loop(1) <<< NOT RESET || COMPETITIVE COMPLETE >>>

X(51) Y(1) NET = 1.77622 , Z = 0.70000 Loop(1) <<< NOT RESET || COMPETITIVE COMPLETE >>>

X(73) Y(1) NET = 1.73427 , Z = 0.50000 Loop(1) <<< NOT RESET || COMPETITIVE COMPLETE >>>

X(17) Y(1) NET = 1.72115 , Z = 0.45000 Loop(1) <<< NOT RESET || COMPETITIVE COMPLETE >>>

X(65) Y(2) NET = 2.17857 , Z = 0.37500 Loop(1) <<< NOT RESET || COMPETITIVE COMPLETE >>>

X(23) Y(2) NET = 2.39286 , Z = 0.62500 Loop(1) <<< NOT RESET || COMPETITIVE COMPLETE >>>

X(89) Y(1) NET = 1.47115 , Z = 0.62500 Loop(1) <<< NOT RESET || COMPETITIVE COMPLETE >>>

X(45) Y(1) NET = 1.69231 , Z = 0.22500 Loop(1) <<< NOT RESET || COMPETITIVE COMPLETE >>>

X(13) Y(2) NET = 2.64286 , Z = 0.62500 Loop(1) <<< NOT RESET || COMPETITIVE COMPLETE >>>

X(8) Y(1) NET = 1.57692 ,Z = 0.00000 Loop(1) ---> RESET :: FIND CANDIDATE NEURONS

X(8) Y(2) NET = 1.18750 ,Z = 0.00000 Loop(1) ---> RESET :: FIND CANDIDATE NEURONS

>>>>> X(8) Y(3) Loop(1) !!! ALL UNITS RESET !!! ADD NEW OUTPUT NEURON , Z = 0.00000

<<<<<

X(42) Y(1) NET = 1.69231 ,Z = 0.00000 Loop(1) ---> RESET :: FIND CANDIDATE NEURONS

X(42) Y(2) NET = 1.16667 ,Z = 0.00000 Loop(1) ---> RESET :: FIND CANDIDATE NEURONS

X(42) Y(3) NET = 1.00000 , Z = 0.33333 Loop(1) <<< NOT RESET || COMPETITIVE COMPLETE >>>

X(99) Y(2) NET = 2.64286 , Z = 0.75000 Loop(1) <<< NOT RESET || COMPETITIVE COMPLETE >>>

X(49) Y(3) NET = 2.10000 , Z = 0.58333 Loop(1) <<< NOT RESET || COMPETITIVE COMPLETE >>>

X(26) Y(2) NET = 2.14286 , Z = 0.45000 Loop(1) <<< NOT RESET || COMPETITIVE COMPLETE >>>

X(9) Y(2) NET = 2.14286 , Z = 0.37500 Loop(1) <<< NOT RESET || COMPETITIVE COMPLETE >>>

X(30) Y(3) NET = 1.80357 ,Z = 0.00000 Loop(1) ---> RESET :: FIND CANDIDATE NEURONS

X(30) Y(1) NET = 1.78297 , Z = 0.45000 Loop(1) <<< NOT RESET || COMPETITIVE COMPLETE >>>

X(22) Y(3) NET = 1.67857 , Z = 0.26667 Loop(1) <<< NOT RESET || COMPETITIVE COMPLETE >>>

X(21) Y(2) NET = 2.50000 , Z = 0.29167 Loop(1) <<< NOT RESET || COMPETITIVE COMPLETE >>>

X(37) Y(2) NET = 2.31250 , Z = 0.40000 Loop(1) <<< NOT RESET || COMPETITIVE COMPLETE >>>

X(71) Y(3) NET = 2.07143 , Z = 0.37500 Loop(1) <<< NOT RESET || COMPETITIVE COMPLETE >>>

X(100) Y(3) NET = 1.83036 , Z = 0.37500 Loop(1) <<< NOT RESET || COMPETITIVE COMPLETE

>>>

X(84) Y(3) NET = 2.01786 , Z = 0.26667 Loop(1) <<< NOT RESET || COMPETITIVE COMPLETE >>>

X(47) Y(3) NET = 2.20536 , Z = 0.62500 Loop(1) <<< NOT RESET || COMPETITIVE COMPLETE >>>

.....

.....

.....

.....

X(93) Y(2) NET = 2.35000 , Z = 0.70833 Loop(7) <<< NOT RESET || COMPETITIVE COMPLETE >>>

X(40) Y(2) NET = 3.45000 , Z = 0.55000 Loop(7) <<< NOT RESET || COMPETITIVE COMPLETE >>>

X(24) Y(1) NET = 2.50000 , Z = 0.54167 Loop(7) <<< NOT RESET || COMPETITIVE COMPLETE >>>

X(7) Y(3) NET = 3.00000 , Z = 0.47500 Loop(7) <<< NOT RESET || COMPETITIVE COMPLETE >>>

X(27) Y(3) NET = 2.75000 , Z = 0.67500 Loop(7) <<< NOT RESET || COMPETITIVE COMPLETE >>>

X(69) Y(1) NET = 2.50000 , Z = 0.35000 Loop(7) <<< NOT RESET || COMPETITIVE COMPLETE >>>

X(15) Y(1) NET = 2.25000 , Z = 0.22500 Loop(7) <<< NOT RESET || COMPETITIVE COMPLETE >>>

X(12) Y(3) NET = 3.00000 , Z = 0.54167 Loop(7) <<< NOT RESET || COMPETITIVE COMPLETE >>>

X(14) Y(3) NET = 2.50000 , Z = 0.57500 Loop(7) <<< NOT RESET || COMPETITIVE COMPLETE >>>

X(99) Y(2) NET = 2.95000 , Z = 0.60000 Loop(7) <<< NOT RESET || COMPETITIVE COMPLETE >>>

X(26) Y(2) NET = 2.65000 , Z = 0.45000 Loop(7) <<< NOT RESET || COMPETITIVE COMPLETE >>>

X(75) Y(3) NET = 2.75000 , Z = 0.87500 Loop(7) <<< NOT RESET || COMPETITIVE COMPLETE >>>

X(72) Y(3) NET = 2.25000 , Z = 0.53333 Loop(7) <<< NOT RESET || COMPETITIVE COMPLETE >>>

X(50) Y(3) NET = 2.50000 , Z = 0.37500 Loop(7) <<< NOT RESET || COMPETITIVE COMPLETE >>>

X(48) Y(1) NET = 3.25000 , Z = 0.40000 Loop(7) <<< NOT RESET || COMPETITIVE COMPLETE >>>

X(91) Y(3) NET = 2.50000 , Z = 0.32500 Loop(7) <<< NOT RESET || COMPETITIVE COMPLETE >>>

X(41) Y(1) NET = 2.25000 , Z = 0.26667 Loop(7) <<< NOT RESET || COMPETITIVE COMPLETE >>>

X(39) Y(3) NET = 2.50000 , Z = 0.35000 Loop(7) <<< NOT RESET || COMPETITIVE COMPLETE >>>

X(77) Y(2) NET = 2.35000 , Z = 0.22500 Loop(7) <<< NOT RESET || COMPETITIVE COMPLETE >>>

X(58) Y(3) NET = 2.50000 , Z = 0.53333 Loop(7) <<< NOT RESET || COMPETITIVE COMPLETE >>>

X(71) Y(1) NET = 2.50000 , Z = 0.36667 Loop(7) <<< NOT RESET || COMPETITIVE COMPLETE >>>

X(64) Y(1) NET = 2.50000 , Z = 0.70833 Loop(8) <<< NOT RESET || COMPETITIVE COMPLETE >>>

X(16) Y(2) NET = 2.50000 , Z = 0.35000 Loop(8) <<< NOT RESET || COMPETITIVE COMPLETE >>>

X(98) Y(3) NET = 2.75000 , Z = 0.40000 Loop(8) <<< NOT RESET || COMPETITIVE COMPLETE >>>

X(97) Y(3) NET = 2.50000 , Z = 0.30000 Loop(8) <<< NOT RESET || COMPETITIVE COMPLETE >>>

(4) Y(1) NET = 3.00000 , Z = 0.36667 Loop(8) <<< NOT RESET || COMPETITIVE COMPLETE >>>

X(38) Y(1) NET = 1.75000 , Z = 0.37500 Loop(8) <<< NOT RESET || COMPETITIVE COMPLETE >>>

X(34) Y(2) NET = 2.50000 , Z = 0.41667 Loop(8) <<< NOT RESET || COMPETITIVE COMPLETE >>>

X(67) Y(2) NET = 2.25000 , Z = 0.22500 Loop(8) <<< NOT RESET || COMPETITIVE COMPLETE >>>

X(14) Y(1) NET = 2.50000 , Z = 0.29167 Loop(8) <<< NOT RESET || COMPETITIVE COMPLETE >>>

X(96) Y(1) NET = 2.50000 , Z = 0.50000 Loop(8) <<< NOT RESET || COMPETITIVE COMPLETE >>>

X(78) Y(3) NET = 3.00000 , Z = 0.30000 Loop(8) <<< NOT RESET || COMPETITIVE COMPLETE >>>

X(28) Y(2) NET = 2.50000 , Z = 0.54167 Loop(8) <<< NOT RESET || COMPETITIVE COMPLETE >>>

X(93) Y(3) NET = 2.25000 , Z = 0.62500 Loop(8) <<< NOT RESET || COMPETITIVE COMPLETE >>>

X(74) Y(1) NET = 2.75000 , Z = 0.33333 Loop(8) <<< NOT RESET || COMPETITIVE COMPLETE >>>

X(29) Y(1) NET = 2.25000 , Z = 0.35000 Loop(8) <<< NOT RESET || COMPETITIVE COMPLETE >>>

X(43) Y(1) NET = 2.50000 , Z = 0.70000 Loop(8) <<< NOT RESET || COMPETITIVE COMPLETE >>>

X(53) Y(1) NET = 2.50000 , Z = 0.35000 Loop(8) <<< NOT RESET || COMPETITIVE COMPLETE >>>

X(86) Y(2) NET = 2.25000 , Z = 0.43333 Loop(8) <<< NOT RESET || COMPETITIVE COMPLETE >>>

X(15) Y(2) NET = 2.25000 , Z = 0.41667 Loop(8) <<< NOT RESET || COMPETITIVE COMPLETE >>>

X(54) Y(3) NET = 3.00000 , Z = 0.62500 Loop(8) <<< NOT RESET || COMPETITIVE COMPLETE >>>

X(24) Y(2) NET = 2.50000 , Z = 0.54167 Loop(8) <<< NOT RESET || COMPETITIVE COMPLETE >>>

X(85) Y(1) NET = 2.75000 , Z = 0.62500 Loop(8) <<< NOT RESET || COMPETITIVE COMPLETE >>>

(Learning Loop = 100) Epoch = 8

The vigilance value = 0.1000

The number of class = 3

Entropy : 0.00000

F-Measure : 1.00000

Number of Reset : 0

Data Swaping : 0

confusion =

0	0	30
38	0	0
0	32	0

Clock = [year:2004] [month:4] [day:19] [hour:14] [minute:0] [seconds:51]

Epoch 8 ==> ***** No Data Swap Move Across Clusters ****

The learning loop (8) is finish

The process is completion.

Running time is 8.05 minutes.

Experimental Result of clustering.

The total elements` number of cluster number[1] = 38

The total elements` number of cluster number[2] = 32

The total elements` number of cluster number[3] = 30

The 100 rows of data set can be computed as 3 clusters.

The total of new neurons is 2

The vigilance value is 0.100

The training set is data2

Entropy of clustering data is 0.00000

F-Measure of clustering data is 1.00000

Confusion Matrix is in train_model.mat ,variable namely, confusion

ภาคผนวก ค.

ผลงานวิจัยที่ได้รับการตีพิมพ์

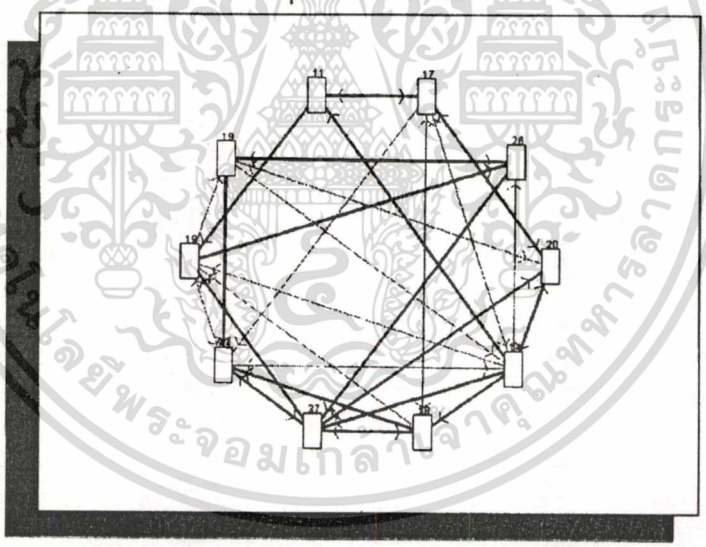
1. Worapoj Kreesuradej and Norraseth Chantasut, "A Text Processing Adaptive Resonance Theory Neural Network," Intelligent Engineering Systems Through Artificial Neural Networks: Proceedings of the 2002 Artificial Neural Network In Engineering Conference (ANNIE'02), ASME Press, MO, USA, Vol.12, 2002. pp.625-630,
2. Worapoj Kreesuradej, Warune Kruaklai and Norraseth Chantasut, "Clustering Text Data Using Text ART Neural Network," WSEAS Transactions on Systems, Issue 1, Vol. 3, January, 2004. pp. 200-205,

INTELLIGENT ENGINEERING SYSTEMS THROUGH ARTIFICIAL NEURAL NETWORKS

VOLUME 12

SMART ENGINEERING SYSTEM DESIGN:
NEURAL NETWORKS, FUZZY LOGIC, EVOLUTIONARY PROGRAMMING, DATA MINING, AND COMPLEX SYSTEMS

Editors:
Cihan H. Dagli
Anna L. Buczak
Joydeep Ghosh
Mark J. Embrechts
Okan Ersoy
Stephen W. Kercel



TEXT PROCESSING ADAPTIVE RESONANCE THEORY NEURAL NETWORK

WORAPOJ KREESURADEJ
Faculty of Information Technology,
King Mongkut's Institute of
Technology Ladkrabang,
Bangkok, 10520, Thailand

NORRASETH CHANTASUT
Faculty of Information Technology,
King Mongkut's Institute of
Technology Ladkrabang,
Bangkok, 10520, Thailand

ABSTRACT

This paper proposes a Text Processing Adaptive Resonance Theory Neural Network for document clustering. Unlike the conventional clustering algorithms, a Text Processing Adaptive Resonance Theory Neural Network works directly on textual information without transforming text data into a numerical value. The main contribution of this paper is to show how to adapt the concepts of ART1 clustering on a data set, which has a qualitative feature values. The Text Processing Adaptive Resonance Theory Neural Network utilizes of the concept of similarity measure for symbolic objects, which is different from the conventional similarity measure for objects whose feature values are numerical values. The proposed neural network assigns cluster labels to the objects.

INTRODUCTION

Continuous numeric data are well known as a classical data type for clustering and the conventional clustering algorithm works on the objects which are numerical values. Symbolic objects are extensions of classical data type. The features of symbolic objects may include one or more elementary objects and data set may have a variable number of features (Gowda and Diday,1992). There is some research dealing with symbolic objects. Gowda and Diday proposed dissimilarity and similarity measure based on position, span, and content of symbolic objects. The core algorithm is agglomerative clustering. Gowda and Ravi proposed divisive clustering for symbolic objects. El-SonBaty Y.A. and Ismail M.A. proposed fuzzy clustering for symbolic data. T.V. Ravi and K.C. Gowda proposed clustering of symbolic objects using gravitational approach. The text data is considered as a symbolic objects.

Several clustering algorithms for objects whose feature values are numerical values are the conventional clustering algorithms. Recently, clustering problems are extended for text document clustering and non-numerical values. To cluster text documents by using the conventional clustering algorithms, each document or object has to be mapped onto some representation that it has quantitative features. One of most widely used representation is the vector space model (Gerard Salton,1989). The utilization of the vector space model may lead to a very high dimensional feature space. In addition, this feature space is generally not free from correlation.

The proposed clustering algorithm in this paper is different from the conventional clustering algorithm which works numerical data type. The proposed clustering algorithm works directly on textual data without transforming text data into numerical values or quantitative features. The inputs of the proposed neural network directly receive a qualitative feature. Then, based on a new unsupervised learning and the concepts of similarity measure for symbolic objects, the proposed neural network assigns cluster labels to the objects

DOCUMENT REPRESENTATION

A document , *Doc* , for clustering task can be written as the Cartesian product of specific values of its features Doc_k's as (Gowda and Diday,1992).

$$Doc = D_1 \times D_2 \times D_3 \times ... \times D_d \quad (1)$$

Different from a vector space model, the feature values are words that describe the features. As an example, a document can be written as Cartesian product of *Title* feature and *Keyword* feature as

$$Doc = Title \times Keyword \quad (2)$$

where the values of the *Title* features are words that describe the title of the document and the values of the *Keyword* features are a set of keywords of the document.

SIMILARITY MEASURE

Here, according to Gowda and Diday, similarity between two documents A and B is defined as

$$S(A,B) = \sum_{k=1}^{d} S(A_k, B_k) \quad (3)$$

For the kth feature, $S(A_k, B_k)$ is defined using the following three components.

1) $S_p(A_k, B_k)$ due to position.

2) $S_s(A_k, B_k)$ due to span.

3) $S_c(A_k, B_k)$ due to content.

The similarity component due to "position" arises only when the feature type is quantitative. The position indicates the relative positions of two feature values on real line. The similarity component due to "span" indicates the relative sizes of the feature values without referring to common part between them. The similarity component due to "content" is a measure of common parts between two feature values (Gowda and Diday, 1992).

For qualitative type of features, the similarity component due to "position" absent. The two components that contribute to similarity are "span" and "content".

Let

l_a = length of A_k or number of elements in A_k
l_b = length of B_k or number of elements in B_k
inters = length of intersection of A_k and B_k
l_s = span length of A_k and B_k combined $= l_a + l_b - inters$

The similarity component due to "span" is defined as

$$S_s(A_k, B_k) = \frac{(l_a + l_b)}{2 \times l_s} \tag{4}$$

The similarity component due to "content" is defined as

$$S_c(A_k, B_k) = \frac{\text{inters}}{l_s} \tag{5}$$

Net similarity between A_k and B_k is defined as

$$S(A_k, B_k) = S_s(A_k, B_k) + S_c(A_k, B_k) \tag{6}$$

The net similarity has the degree of similar between [0.5, 2]. As an example, the similar between the first object , Doc1 , and the second object , Doc2 , shown in Table 1 , can be computed as following.

Table 1: Document Data

DOCUMENT	*TITLE*	*KEYWORD*
Doc1	JAVA Handbook	JAVA Object Oriented Approach
Doc2	C++ Handbook	C++ Object Oriented Approach
Doc3	MatLab Handbook	MatLab matrix laboratory

S(Doc1,Doc2) = S(JAVA Handbook, C++ Handbook) +
S(JAVA Object Oriented Approach, C++ Object Oriented Approach)
S(JAVA HandBook, C++ Handbook) = [(13+12) / 2 .(17)] + (8/17)
= 0.7353+0.4706 = 1.2059
The net similarity of *Title* feature is 1.2059.

S(JAVA Object Oriented Approach, C++ Object Oriented Approach)
= [(29+28)/2.(41)]+ (16/41) = 0.6951+0.3902 =1.0854
The net similarity of *Keyword* feature is 1.0854.

Then the net similarity between Doc1 and Doc2 for two features of values is 1.2059 + 1.0854 = 2.2913. The concepts of similarity measure will be used to measure the similarity between documents in the next section.

THE TEXT PROCESSING ADAPTIVE RESONANCE THEORY NEURAL NETWORK

In this section, a new learning algorithm for document clustering is presented. The architecture of the proposed neural networks is shown in Fig 1.

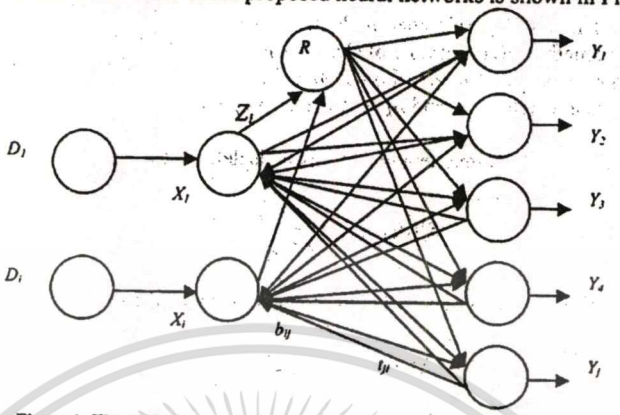


Figure 1: The architecture of Text Processing Adaptive Resonance Theory Neural Network

Unlike conventional neural networks, input layer of proposed neural network receive qualitative values by using unsupervised learning. To compute qualitative values, the bottom-up weight, b_{ij}, and the top-down weights, t_{ji}, have to contain qualitative values and degree of association of the qualitative values.

$$b_{ij} = \{(A_{1ij}, e_{1ij}),(A_{2ij}, e_{2ij}),...,(A_{pij}, e_{pij})\} \quad (7)$$

where A_{pij} is the pth qualitative value of the weight and e_{pij} is the degree of association of this qualitative value to the input ith. e_{pij}'s have value between interval 0 to 1. $e_{pij} = 0$ if the qualitative value, A_{pij}, is not a part of the input ith, While $e_{pij} = 1$ if the qualitative value has strong association with the input ith.

$$t_{ji} = \{(B_{1ji}, e_{1ji}),(B_{2ji}, e_{2ji}),...,(B_{pji}, e_{pji})\} \quad (8)$$

where B_{pji} is the pth qualitative value of the weight and e_{pji} is the degree of association of this qualitative value to the input jth. e_{pji}'s have value between interval 0 to 1. $e_{pji} = 0$ if the qualitative value, B_{pji}, is not a part of the input jth, While $e_{pji} = 1$ if the qualitative value has strong association with the input jth.

LEARNING ALGORITHM

The details of the proposed learning algorithm can be presented as below:

Step 0: Initialize bottom-up weight and top-down weight can be initialized from the training data set arbitrarily. Vigilance parameter, ρ, = [0, 4]

Step 1: While stopping condition is false, do step 2-9

Step 2: For each input qualitative values, transpose

$Doc = (D_1, D_2, D_3, ..., D_d)^t$, do step 3-8 (9)

Step 3: Set input D from the $F_{1(a)}$ to X variable on the $F_{1(b)}$

$X_i = D_i$ (10)

Step 4: For each the jth output unit, compute

$$Y_j = \sum_{k=1}^{d} \sum_{n=1}^{p} S(X_k, A_{nkj}) e_{nkj} \quad (11)$$

where

p is number of values of bottom-up weight

d is number of feature values

Step 5: Find index J such that is a maximum

Step 6: For all top-down weight that connect to the winning node J

$$Z = S(X_i, t_{Ji}) e_{Ji} \quad (12)$$

Step 7: Test for reset mechanism

If $Z < \rho$, then $Y_J = -1$ (inhibit node J), do step 5 again

If $Z \geq \rho$, then do step 8

Step8: Update bottom-up weight and top-down weight for node J

$b_{ij}^{(new)} = b_{ij}^{(old)} \cap X$

$$e_{niJ}^{(new)} = \begin{cases} f(e_{niJ}^{(old)} + \eta) \text{ if } A_{nij} \in b_{ij} \cap X, \\ f(e_{niJ}^{(old)} - \eta) \text{ if } A_{nij} \notin b_{ij} \cap X \end{cases}$$

$t_{Ji}^{(new)} = X$

$$e_{nJi}^{(new)} = \begin{cases} f(e_{nJi}^{(old)} + \eta) \text{ if } B_{nji} \in t_{ji} \cap X, \\ f(e_{nJi}^{(old)} - \eta) \text{ if } B_{nji} \notin t_{ji} \cap X \end{cases} \quad (13)$$

where $f(.)$ is defined as below:

$$f(x) = \begin{cases} x & \text{if } 0 \leq x \leq 1 \\ 0 & \text{if } x < 0 \\ 1 & \text{if } x > 1 \end{cases} \quad (14)$$

Step 9: Test for stopping condition. Might consist of any o the following below:

No weight changes,

No unit reset,

Maximum number of epochs reached

EXPERIMENTAL RESULTS

In this section, the experimental results of the proposed neural network are represented. A training data set 100 data consist of 3 clusters and a testing data set 100 data consist of 3 clusters are synthesized. Each document in the training data set can be represented by *Title* feature and *Keyword* feature.

Each feature of a text data is qualitative value. For this experiment, some English alphabets are used to represent values of each feature. A set of title

feature for a member of cluster number 1 is a subset of {a,c,d,g,h} and a set of keyword feature for a member of cluster number 1 is a subset of {c,d,f,i,j}. A set of title feature for a member of cluster number 2 is a subset of {g,h,k,m,n} and a set of keyword feature for a member of cluster number 2 is a subset of {i,j,m,n,p}. A set of title feature for a member of cluster number 3 is a subset of {r,s,t,u,v} and a set of keyword feature for a member of cluster number 3 is a subset of {u,v,w,x,y}.

To measure the accuracy of the proposed neural network, the clustering accuracy, r, is defined as

$$r = [1 - (\sum_{i=1}^{c} doc)/n] * 100 \qquad (15)$$

Where c is a number of data that are incorrectly assigned to an incorrectly cluster. n is a number of all data.

The experimental results of the testing data set consist of 32 data from cluster number 1 , 27 data from cluster number 2 and 41 data from cluster number 3 . According to the testing data set, the proposed neural network can gives the clustering accuracy as 99.00 %. This shows that the proposed network has well performance in clustering text data.

CONCLUSIONS

In this paper, a text processing adaptive resonance theory neural network is proposed. From the experimental results, the proposed neural network can work directly textual data without mapping the qualitative value into numerical value as well performance. In the future, some experiment results that are conducted on the Reuter-21578 news articles will be reported.

NOMENCLATURE

$S(A,B)$: similarity measure between A and B objects
S_s : similarity component "span"
S_c : similarity component "content"
r : clustering accuracy

REFERENCES

Gerard Salton, "Automatic Text Processing: the transformation, analysis, and retrieval of information by computer," Addison Wesley Publishing Company, New York, 1989.

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

Lanrene Fausett, "Fundamentals of Neural Networks Architecture, Algorithms and Application," Prentice Hall International, New Jersey, 1994.

El-SonBaty Y.A. and Ismail M.A., "Fuzzy Clustering for Symbolic Data," *IEEE Trans. On Fuzzy Systems*, vol. 6, no. 2, pp. 195-204, May. 1998.

T.V. Ravi and K.C. Gowda, "Clustering of Symbolic Objects Using Gravitational Approach," *IEEE Trans. On Syst. , Man, Cybern.*, vol. 29, no. 6, pp. 888-894, 1999.

Jiawei Han and Micheline Kamber, "Data Mining: Concepts and Technique," Margan Kaufmann Publishers, New York, 2001.

WSEAS TRANSACTIONS on SYSTEMS

Issue 1, Volume 3, January 2004
ISSN 1109-2777 http://www.wseas.org

Speed Control of a Switched Reluctance Motor Using Beibie 1
G.Zandesh, J.Moghani, C.Lucas, D.Shahmirzadi, B.N.Araabi O.Namaki, H.kord

A parallel environment using taboo search and genetic algorithms for solving partitioning problems in codesign 8
Abderrazak Henni, Mouloud Koudil, Karima Benatchba, Hassane Oumsalem, Kamel Chaouche

Spatial Electric Load Distribution Forecasting using Simulated Annealing 14
Hector Gustavo Arango and Germano Lambert Torres

A solving prototype dedicated to a transport problem case study 20
Meriam Kefi,Khaled Ghedira

A Fuzzy Approach to Uncertainty Principle in Quantum Theory 26
Shahriyar Kaboli, Nasser Sadati

The Optimal Configuration of Cogeneration Systems Based on Natural Gas: a Parallel Evolutionary Approach 31
Marco Goldbarg, Elizabeth Goldbarg, Francisco Neto

Generating Periodic Functions 37
Matilde Legua, Jose A. Mora±o, Luis M. Sánchez Ruiz

A Transgenetic Algorithm for the Permutation Flow-shop Sequencing Problem 40
Elizabeth Goldbarg,Marco Goldbarg,Wagner Costa

Active noise control for a one-dimensional acoustic duct using feedback control techniques: modelling and simulation 46
Zhenyu Yang

Contribuition to Model Eucalyptus's Growth Curves in Portugal 55
Anabela Gouveia da Silva

Application of Satisfiability algorithms to time-table problems 61
Fahima Nader, Mouloud Koudil, Karima Benatchba, Lotfi Admane, Said Gharout, Nacer Hamani

Cam Optimal Design with Evolutionary Algorithms 66
Alberto Borboni

knowledge-based automatic fault detection for dynamic physical systems 72
C.H. Lo, Y.K. Wong and A.B. Rad

Hybrid simulation of qualitative bond graph model 78
C.H. Lo, Y.K. Wong and A.B. Rad

Parametric time domain identification of a flexible robotic arm using evolutionary algorithms 84
Dimitris Koulocheris, Vasilis Dertimanis, Harry Vrazopoulos

A Hybrid Evolution Strategy for vehicle suspension optimization 90

Clustering Text Data Using Text ART Neural Network

WORAPOJ KREESURADEJ,
WARUNE KRUAKLAI
Data Mining and Data Exploration Laboratory,
Faculty of Information Technology,
King Mongkut's Institute of Technology Ladkrabang,
Ladkrabang, Bangkok 10520
THAILAND
worapoj@it.kmitl.ac.th,
warune@it.kmitl.ac.th
http://www.it.kmitl.ac.th/

NORRASETH CHANTASUT
High Performance Computing Research and Development Division,
National Electronics and Computer Technology Center,
National Science and Technology Development Agent,
112 Thailand Science Park
Klong Luang, Pathumthani 12120
THAILAND
norraseth.chantasut@nectec.or.th
http://www.hpcc.nectec.or.th/

Abstract: Most studies of data mining have focus on structured data such as relational, transactional, and data warehouse data. However, the most available information is stored in text database, which consist of large amounts of text documents such as news articles, research papers, and e-mail messages. Data stored in most text databases are unstructured data, such as abstract and contents. The ability to deal with different types of attributes is a typical requirement of clustering in data mining. Thus, mining unstructured data has become an increasingly important task in text mining. The main contribution of this paper is to cluster on a data set, which has a non-numerical feature value. Unlike the conventional clustering algorithms such as the K-Means algorithm, which forms clusters in numerical values domains, a Text ART Neural Network works directly on textual information without text transformation into a numerical value. The experimental results are represented that conducted on 2 datasets. The first dataset is a Synthesized Text Document and the second dataset is a Reuter-21578 Distribution 1.0. The F-Measure equation use to measure the effectiveness of the proposed technique. According to the experimental results, the proposed neural network has well performance in clustering text data that the F-Measure of the experiment is 95.56% and 83.31% respectively.

Key-Words: - Text Mining, Document Clustering, Unsupervised Learning, Artificial Neural Networks

1 Introduction

The text mining research areas focus about patterns in natural language text and use particular techniques to extract useful knowledge from unstructured data. The several techniques of text mining are proposed. These techniques can be divided into document clustering techniques, document classification techniques, and text summarization techniques. Due to the huge amounts of textual data collection in text database, document clustering has become a highly topic in text mining research. In this paper we focus on document clustering problems.

The conventional clustering of data can be divided into hierarchical clustering and partitioning clustering. The hierarchical clustering creates a hierarchical decomposition of the data such as agglomerative algorithm and divisive algorithms. In section of the partitioning clustering generates a partition of the data to recover natural groups present in the data. Which of the data is in the form of a pattern matrix and the feature space of pattern matrix contain a numerical value.

Numeric data are well known as a classical data type for clustering and the classical clustering algorithm such as K-Means algorithm, works on the objects which are numerical values. Textual information is extensions of classical data type. Recently, clustering problems are extended for document clustering and non-numerical values. To cluster documents by using the conventional clustering algorithms, each document or object has to be mapped onto some representation that it has quantitative features. One of most widely used representation is the vector space model [2]. The utilization of the vector space model may lead to a very high dimensional feature space. In addition, this feature space is generally not free from correlation. There is some research dealing with non-numerical data [7][8][9].

The proposed clustering algorithm in this paper is different from the conventional clustering

algorithm which works numerical data type. The proposed clustering algorithm works directly on textual data without text transformation into numerical values or quantitative features. The inputs of the proposed neural network directly receive a textual feature value. Then, based on a new unsupervised learning and the concepts of similarity measure for symbolic objects [5], the proposed neural network assigns cluster labels to the objects. The experimental results are represented that conducted on 2 datasets. The first dataset is a Synthesized Text Document and the second dataset is a Reuter-21578 Distribution 1.0. The Reuters-21578, Distribution 1.0 test collection is available from David D. Lewis' professional home page, currently: http://www.research.att.com/~lewis/ or http://www.daviddlewis.com/

2 Text Document Representation

To transform text dataset into an appropriate representation for the learning algorithm, we have to prepare text data with data preprocessing steps following below:

1) the first step we remove digits and punctuation marks.

2) the second step is conversion words onto lower case.

3) the next step is removal stop words

4) use Porter's stemming algorithm to find out root of words. [7]

And represent a text data, *Doc*, for clustering task as the Cartesian Product of specific values of its features Doc_k's as [5][6].

$$Doc = D_1 \times D_2 \times D_3 \times ... \times D_d \qquad (1)$$

The feature values have qualitative features [5], Different from a vector space model [2]. The feature values are words that describe the features. As an example, a text document can be written as Cartesian product of *Title* feature and *Keyword* feature as

$$Doc = Title \times Keyword \qquad (2)$$

where the values of the *Title* features are words that describe the title of the document and the values of the *Keyword* features are a set of keywords of the document

3 The Text ART Neural Network

In this section, a new learning algorithm for text document clustering is presented. The architecture of the proposed neural networks is similar with ART1 architecture. But it is different on feature type of weights, similarity measure, and weights adjustment mechanism.

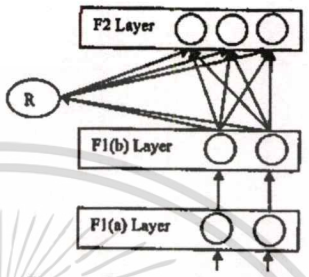


Figure 1. The architecture of Text ART Neural Network

3.1 Feature Type of Weights

To recognize qualitative feature values, the bottom-up weights and the top-down weights have to contain qualitative feature values and degree of association of the qualitative feature values.

$$b_{ij} = \{(A_{1j}, e_{1j}), (A_{2j}, e_{2j}), ..., (A_{pj}, e_{pj})\} \qquad (3)$$

where A_{pj} is the pth qualitative value of the weight and e_{pj} is the degree of association of this qualitative value to the input ith. e_{pj}'s have value between interval 0 to 1. $e_{pj} = 0$ if the qualitative value, A_{pj}, is not a part of the input ith, While $e_{pj} = 1$ if the qualitative value has strong association with the input ith.

$$t_{ji} = \{(B_{1j}, e_{1j}), (B_{2j}, e_{2j}), ..., (B_{pj}, e_{pj})\} \qquad (4)$$

where B_{pj} is the pth qualitative value of the weight and e_{pj} is the degree of association of this qualitative value to the input jth. e_{pj}'s have value between interval 0 to 1. $e_{pj} = 0$ if the qualitative value, B_{pj}, is not a part of the input jth, While $e_{pj} = 1$ if the qualitative value has strong association with the input jth.

3.2 Similarity Measure

Similarity measure is introduced in the literature for symbolic objects [5]. We follow the similarity measure introduced by Gowda and Diday [5] to formalize the problem of text document clustering. Here, according to Gowda and Diday, similarity between two documents A and B is defined as

$$S(A,B) = \sum_{k=1}^{d} S(A_k, B_k) \qquad (5)$$

For the kth feature, $S(A_k, B_k)$ is defined using the following three components.

1) $S_s(A_k, B_k)$ due to span

2) $S_c(A_k, B_k)$ due to content

3) $S_p(A_k, B_k)$ due to position

The similarity component due to "span" indicates the relative sizes of the feature values without referring to common part between them. The similarity component due to "content" is a measure of common parts between two feature values. [5]

For qualitative type of features, the similarity component due to "position" is absent. The two components that contribute to similarity are "span" and "content"

Let

l_a = length of A_k or number of elements in A_k

l_b = length of B_k or number of elements in B_k

$inters$ = length of intersection of A_k and B_k

l_s = span length of A_k and B_k combined

$= l_a + l_b - inters$

The similarity component due to "span" is defined as

$$S_s(A_k, B_k) = \frac{(l_a + l_b)}{2 \times l_s} \qquad (6)$$

The similarity component due to "content" is defined as

$$S_c(A_k, B_k) = \frac{inters}{l_s} \qquad (7)$$

Net similarity between A_k and B_k is defined as

$$S(A_k, B_k) = S_s(A_k, B_k) + S_c(A_k, B_k) \qquad (8)$$

3.3 Learning Algorithm

The F_2 layer is a competitive layer. The output unit with the largest net input becomes the candidate to learn the input patterns, which are qualitative values. Whether or not this output unit is allowed to learn the input pattern depends on how similar its top-down weight is to the input pattern. This decision is made by the reset unit, based on qualitative values it receives from the interface the $F_{1(a)}$ layer. If the output unit is not allowed to learn, it is inhibited and a new unit is selected as the candidate [3]. The concepts of similarity measure introduced by Gowda and Diday will be applied to the ART1 learning algorithm. The details of the proposed learning algorithm can be presented as below:

Step 0: Initialize bottom-up weight and top-down weight can be initialized from the training data set arbitrarily.

Step 1: While stopping condition is false, do step 2-9

Step 2: For each input qualitative values, transpose

$Dec = (D_1, D_2, D_3, ..., D_d)^t$, do step 3-8

Step 3: Set input D from the $F_{1(a)}$ to X variable on the $F_{1(b)}$

$$X_i = D_i$$

Step 4: For each the jth output unit, compute

$$Y_j = \sum_{k=1}^{d} \sum_{n=1}^{p} S(X_k, A_{nkj}) \qquad (9)$$

where

p is number of values of bottom-up weight

d is number of feature values

Step 5: Find index J such that is a maximum

Step 6: For all top-down weight that connect to the winning node J :

$$Z = S(X_k, B_{nkJ}) \qquad (10)$$

Step 7: Test for reset mechanism

If $Z < \rho$, then $Y_J = -1$ (inhibit node J), do step 5 again

If $Z \geq \rho$, then do step 8

Step8: Update bottom-up weight and top-down weight for node J

$b_{ij}^{(new)} = b_{ij}^{(old)} \cap X$

$$e_{nij}^{(new)} = \begin{cases} f(e_{ij}^{(old)} + \eta) \text{ if } A_{nij} \in b_{ij} \cap X, \\ f(e_{ij}^{(old)} - \eta) \text{ if } A_{nij} \notin b_{ij} \cap X, \\ 5 * \eta; \text{Otherwise} \end{cases} \qquad (11)$$

$$t_B^{(new)} = t_B^{(old)} \cup X$$

$$e_{xB}^{(new)} = \begin{cases} f(e_{xB}^{(old)} + \eta) \text{ if } B_{xB} \in t_B \cap X, \\ f(e_{xB}^{(old)} - \eta) \text{ if } B_{xB} \notin t_B \cap X, \\ 5 * \eta; \text{Otherwise} \end{cases} \quad (12)$$

where $f(.)$ is defined as below:

$$f(x) = \begin{cases} x & \text{if } 0 \le x \le 1 \\ 0 & \text{if } x < 0 \\ 1 & \text{if } x > 1 \end{cases} \quad (13)$$

Step 9: Test for stopping condition. Might consist of any of the following below:

No weight changes,
No unit reset,
Maximum number of epochs reached

4 Evaluation Measure

To measure the effectiveness of the proposed neural network, we use the F-Measure to estimate the clustering accuracy, is defined below [4]. The F-measure of cluster j and class i is define as

$$F(i,j) = \frac{2 * recall(i,j) * precision(i,j)}{recall(i,j) + precision(i,j)} \quad (14)$$

$$recall(i,j) = \left[\frac{n_{ij}}{n_i}\right] * 100 \quad (15)$$

$$precision(i,j) = \left[\frac{n_{ij}}{n_j}\right] * 100 \quad (16)$$

where n_{ij} is the number of members of class i in cluster j

n_j is the number of members of cluster j

n_i is the number of members of class i

For an entire clustering the F-measure of overall can be computed with equation below

$$F = \sum_i \frac{n_i}{n} \max\{F(i,j)\} \quad (17)$$

The F-Measure values are in the interval [0-100] percent and larger F-Measure values indicate higher clustering quality

5 Experimental Results

In this section, the experimental results of the proposed neural network are represented that consist of 2 datasets. The first dataset is a Synthesized English Text Document and the second dataset is a Reuter-21578. In the experiment phase, we reduce time-consuming in training phase by let the number of training data is less than the number of testing data. Each document in the training data set can be represented by *Title* feature and *Keyword* feature.

5.1 Synthesized Text Document

A set of title feature for a member of cluster number 1 is a subset of {*compute, algorithm, database, intel, java, network*} and a set of keyword feature for a member of cluster number 1 is a subset of {*algorithm, database, predict, cluster, web, firewall*}.

A set of title feature for a member of cluster number 2 is a subset of {*java, network, internet, protocol cisco, 3com*} and a set of keyword feature for a member of cluster number 2 is a subset of {*web, firewall, mail, smtp, http, ftp*}.

A set of title feature for a member of cluster number 3 is a subset of {*car, business, travel, hotel, bank, airline*} and a set of keyword feature for a member of cluster number 3 is a subset of {*market, tour, benz, toyota, money, airway*}.

In the first dataset, a training data set 100 data consist of 3 clusters and a testing data set 1,500 data consist of 3 clusters are synthesized.

Table 1. Synthesized Text Document

Class	No. of training	No. of testing
Class 1	36	499
Class 2	27	510
Class 3	37	491

According to the testing dataset, the proposed neural network can gives the F-measure values by 95.561%. The example of the proposed model after learning phase, is presented in table 2 and 3 as below:

Table 2. Bottom-up weight

Neuron	Weights
Neuron 1	{(3com,1.0),(cisco,0.99), (internet,0.98),(java,1.0), (network,0.99),(protocol,0.98)}
Neuron 2	{(airline,1.0),(bank,1.0), (business,0.98),(car,0.99), (hotel,0.98),(intel,0.0),

	(network,0.0),(travel,1.0)}
Neuron 3	{(algorithm,1.0),(compute,0.99), (database,0.97),(intel,0.98), (java,1.0),(network,1.0)}

Table 3. Top-down weight

Neuron	Weights
Neuron1	{(3com,1.0),(cisco,0.99), (internet,0.98),(java,1.0), (network,0.99),(protocol,0.98)}
Neuron 2	{(airline,1.0),(algorithm,0.0), (bank,0.97),(business,0.98), (car,0.99),(database,0.0), (hotel,0.98),(intel,0.0), (network,0.0),(travel,0.99)}
Neuron 3	{(algorithm,1.0),(compute,0.99), (database,0.96),(intel,0.98), (java,1.0),(network,1.0)}

5.2 Reuter-21578 Distribution 1.0 dataset

The Reuter-21578 text categorization test collection Distribution 1.0 dataset complied by David D. Lewis. The collection consists of 22 data files, an SGML DTD file describing the data file format, and six files describing the categories used to index the data. It is distributed in 22 files. Each of the first 21 files (reut2-000.sgm through reut2-020.sgm) contains 1000 documents, while the last (reut2-021.sgm) contains 578 documents.

In the second dataset, we select 767 news articles from 3 categories for use in training neural network and select 2,895 news articles from same categories for use in testing neural network. As shown in table 2 below.

Table 2. Reuter-21578 Distribution 1.0 dataset

Categories	No. of training	No. of testing
Acq	331	1932
Crude	221	496
Grain	215	467

According to the testing dataset, the proposed neural network can gives the F-measure values by 85.313%.The example of the proposed model after learning phase, is presented in table 4 and 5 as below:

Table 4. Bottom-up weight

Neuron	Weights
Neuron 1	{(complete,0.001),(kraft,0.001), (quaker,0.001),(carling,0.02), (elders,0.002),(buyout,0.003), (disclosure,0.003),(favors,0.003),
Neuron 2	{(expansion,0.001),(analyst,0.002), (good,0.002),(guines,0.002), (papua,0.002),(prospect,0.002), (arab,0.003),(heavy,0.003), (supplies,0.003),(term,0.004), (bombings,0.004),(colombian,0.004), (group,0.004),(pipelines,0.004), (suspend,0.004),(boosts,0.005), (output,0.005),(post,0.08), (acquisition,0.01),(opec,0.011), (american,0.013),(crude,0.018), (price,0.027),(pct,0.157), (oil,1.0)}
Neuron 3	{(midwest,0.001),(movement,0.001), (slow,0.001),(carloadings,0.002), (fall,0.002),(ahead,0.003), (cbt,0.003),(look,0.003), (spring,0.003),(traders,0.003), (lanka,0.04),(overnight,0.04), (sri,0.04),(tendering,0.04), (crop,0.05),(main,0.05), (projection,0.05),(reduce,0.05), (rice,0.05),(plantings,0.09), (maize,0.016),(set,0.026), (export,0.229),(usda,0.398), (corn,0.477),(grain,0.671), (wheat,0.964)}

Table 5. Top-down weight

Neuron	Weights
Neuron1	{(group,0.001),(kraft,0.001), (quaker,0.001),(unit,0.001), (merge,0.002),(carling,0.002), (elders,0.002),(disclosure,0.003), (favors,0.003),(period,0.003), (ahead,0.003),(shortening,0.003), (acquire,0.004),(agree,0.004), (stars,0.004),(store,0.004), (firm,0.004),(stake,0.005), (ope,0.005),(pct,0.008), (buy,0.017)}
Neuron 2	{ (expansion,0.001),(opec,0.001), (analyst,0.002),(good, 0.002), (guines, 0.002),(papua, 0.002), (prospect, 0.002),(american,0.003),

	(arab, 0.003),(crude, 0.003), (heavy, 0.003),(supplies, 0.003), (term, 0.003),(bombings, 0.003), (colombian,0.004),(group, 0.004), (pipelines, 0.004),(suspend, 0.004), (acquisition,0.005),(boosts,0.005), (output,0.005),(pct,0.005), (oil,0.916)}
Neuron 3	{ (country,0.001),(midwest,0.001), (movement,0.001),(slow,0.001), (carloadings,0.002),(fall,0.002), (ahead,0.003),(cbi,0.003), (look,0.003),(spring,0.003), (traders,0.003),(lanka,0.004), (overnight,0.004),(plantings,0.004), (sri,0.004),(tendering,0.005), (crop,0.005),(main,0.005), (projection,0.005),(reduce,0.005) (rice,0.005),(maize,0.005), (export,0.005),(usda,0.006), (oil,0.009),(grain,0.116) (wheat,0.238)}

The feature types of weight contain qualitative feature values and degree of association of the qualitative feature values. After training phase success, the value of weights can recognize term-frequency of which appear in the text document. The degree of association of feature values can be indicator that the word in feature value is a highly term-frequency.

Table 6. Evaluation of clustering data

Dataset	F-Measure
Synthesized Text Document	95.561%
Reuter-21578 Distribution 1.0	85.313%

The clustering accuracy of each dataset is evaluated by F-measure. The result of first dataset has well performance more than the second dataset because of the number of member in Reuter-21578 dataset overlapping has more than the number of member overlapping in Synthesized Text Document. Thus the clustering accuracy of the Synthesized Text Document has better.

6 Conclusion

In this paper, we introduced a Text ART Neural Network based on similarity measure for symbolic objects. We applied the proposed neural networks for text clustering. We use the number of training data less than the number of testing data to reduce time-consuming in training phase. According to the experimental results, the proposed neural network has well performance in clustering text data. In the future work, some experiment results that are conducted on the Reuter-21578 news articles remain categories will be reported.

References:

[1] Jiawei Han and Micheline Kamber, *Data Mining: Concepts and Technique*, Morgan Kaufmann Publishers, New York, 2001

[2] Gerard Salton, *Automatic Text Processing: the transformation, analysis, and retrieval of information by computer*, Addison Wesley Publishing Company, New York, 1989

[3] Laurene Fausett, *Fundamentals of Neural Networks Architecture, Algorithms and Application*, Prentice Hall International, New Jersey, 1994

[4] Michael Steinbach , George Karypis and Vipin Kumar, *A Comparison of Document Clustering Techniques*, Technical Report #00-034, Department of Computer Science and Engineering, University of Minnesota.

[5] El-SonBaty Y.A. and Ismail M.A., Fuzzy Clustering for Symbolic Data, *IEEE Trans. On Fuzzy Systems*, Vol. 6, No. 2, pp. 195-204, May. 1998.

[6] T.V. Ravi and K.C. Gowda, Clustering of Symbolic Objects Using Gravitational Approach, *IEEE Trans. On Syst. , Man, Cybern.*, Vol. 29, No. 6, pp. 888-894, 1999.

[7] M. Benkhalifa and A. Bensaid, Text Categorization using the Semi-Supervised Fuzzy c-Mean Algorithm, *IEEE*, pp. 561-565, 1999

[8] King-Ip Lin and Ravikumar Kondadadi, A Similarity-Based Soft Clustering Algorithm For Documents, *IEEE*, 2001

[9] Florian Beil, Martin Ester and Xiaowei Xu, Frequent Term-Based Text Clustering, *ACM SIGKDD 02*, 2002

[10] http://www.research.att.com/~lewis/

ประวัติผู้เขียน

นายนรเศรษฐ์ จันทสูตร เกิดวันที่ 5 เมษายน 2519 ที่ อำเภอเมือง จังหวัดขอนแก่น

สำเร็จการศึกษา มัธยมศึกษาตอนต้นและตอนปลาย (สายวิทย์ฯ-คณิตฯ) จากโรงเรียนบดินทรเดชา (สิงห์ สิงหเสนี) ปีการศึกษา 2536

สำเร็จการศึกษา วิทยาศาสตรบัณฑิต (วิทยาการคอมพิวเตอร์) จากสถาบันราชภัฏมหาสารคาม ปีการศึกษา 2542

ปีพุทธศักราช 2542-2545 เข้าทำงานตำแหน่งเจ้าหน้าที่คอมพิวเตอร์ ศูนย์สารสนเทศกรมประมง กระทรวงเกษตรและสหกรณ์

ปีพุทธศักราช 2546 เข้าทำงานตำแหน่งผู้ช่วยนักวิจัย งานวิจัยเทคโนโลยีคลังข้อมูล ฝ่ายวิจัยและพัฒนาสาขาเทคโนโลยีคอมพิวเตอร์เพื่อการคำนวณ (RDC) ศูนย์เทคโนโลยีอิเล็กทรอนิกส์และคอมพิวเตอร์แห่งชาติ (NECTEC) สำนักงานพัฒนาวิทยาศาสตร์และเทคโนโลยีแห่งชาติ (NSTDA) กระทรวงวิทยาศาสตร์และเทคโนโลยี