การแบ่งกลุ่มเอกสารโดยใช้เทคนิคการประมวลผลข้อความด้วยโครงข่ายประสาทเทียมที่เรียนรู้แบบหาผู้ชนะ

DOCUMENT CLUSTERING USING A TEXT PROCESSING COMPETITIVE LEARNING NEURAL NETWORK

สมคิด แสนเสนาะ

SOMKID SAENSANOR

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต

สาขาวิชาเทคโนโลยีสารสนเทศ

บัณฑิตวิทยาลัย

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ. 2547

ISBN 974-324-970-2

DOCUMENT CLUSTERING USING A TEXT PROCESSING COMPETITIVE LEARNING NEURAL NETWORK

SOMKID SAENSANOR

A THESIS SUBMITTED IN PARTIAL FULFILLMENT

OF THE REQUIREMENT FOR THE DEGREE OF

MASTER OF SCIENCE PROGRAM IN INFORMATION TECHNOLOGY

SCHOOL OF GRADUATE STUDIES

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

2004

ISBN 974-324-970-2

หัวข้อวิทยานิพนธ์	การแบ่งกลุ่มเอกสารโดยใช้เทคนิคการประมวลผลข้อความด้วยโครงข่ายประสาทเทียมที่เรียนรู้แบบหาผู้ชนะ
นักศึกษา	นายสมคิด แสนเสนาะ
รหัสประจำตัว	42067118
ปริญญา	วิทยาศาสตรมหาบัณฑิต
สาขาวิชา	เทคโนโลยีสารสนเทศ
พ.ศ.	2547
อาจารย์ผู้ควบคุมวิทยานิพนธ์	ผศ.ดร.วรพจน์ กรีสุระเดช

บทคัดย่อ

งานวิจัยนี้มีวัตถุประสงค์เพื่อจัดแบ่งกลุ่มเอกสาร (Document Clustering) โดยใช้เทคนิคการประมวลผลข้อความด้วยโครงข่ายประสาทเทียมที่เรียนรู้แบบหาผู้ชนะ (A Text Processing Competitive Learning Neural Network) โดยที่ข้อมูลเข้า (input) ของโครงข่ายนี้จะรับค่าที่เป็นข้อความเข้าไปโดยตรง ซึ่งแตกต่างจากโครงข่ายประสาทเทียมทั่วไปที่จะแปลงค่าข้อความเหล่านี้ไปเป็นตัวเลขก่อนที่จะส่งเข้าไปประมวลผลในโครงข่าย โดยอัลกอริทึมที่ใช้กับโครงข่ายของเราจะเรียนรู้แบบไม่มีผู้สอน (Unsupervised Learning) ร่วมกับการใช้หลักการของการวัด ความแตกต่าง (Dissimilarity Measure) ระหว่างวัตถุสัญลักษณ์ (Symbolic Object) ช่วยในการคำนวณ เพื่อหาค่าความแตกต่างของแต่ละเอกสาร โดยงานวิจัยนี้มุ่งที่จะแบ่งกลุ่มเอกสารที่มีเนื้อหาประเภทเดียวกันก็จัดให้อยู่ในกลุ่มเดียวกัน

Thesis Title	Document Clustering Using a Text Processing Competitive Learning Neural Network
Student	Mr.Somkid Saensanor
Student ID.	42067118
Degree	Master of Science
Programme	Information Technology
Year	2004
Thesis Advisor	Asst. Prof. Dr Worapoj Kreesuradej

ABSTRACT

This paper proposes document clustering using a text processing competitive learning neural network. The text processing competitive learning neural network works directly on textual information without mapping documents onto some representations those have qualitative features. The inputs of the proposed neural network directly receive a qualitative value without mapping the qualitative value into a numerical value. Then, base on a new unsupervised learning algorithm and the concepts of dissimilarity measure for symbolic objects. The proposed neural network assigns cluster labels to the object.

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงได้อย่างดี ด้วยจากความกรุณาให้คำแนะนำ และคำปรึกษา พร้อมทั้งชี้แนะแนวทางการแก้ไขปัญหาของงานวิจัยอย่างต่อเนื่องของ ผศ.ดร.วรพจน์ กรีสุรเดช ซึ่งเป็นอาจารย์ผู้ควบคุมวิทยานิพจน์ฉบับนี้ข้าพเจ้าซาบซึ้งในความอนุเคราะห์ของท่าน และขอขอบพระคุณเป็นอย่างสูง

กราบขอบพระคุณบิดา-มารดาและญาติพี่น้องของข้าพเจ้าที่ได้ให้กำลังใจและช่วยเหลือโดยตลอด

ขอขอบคุณ คุณพิบูล เทพบุตร ที่ให้คำแนะนำและช่วยเหลือการโปรแกรมด้วยภาษา Perl

ขอขอบคุณ เพื่อนๆนักศึกษาทุกคนรวมทั้งรุ่นพี่รุ่นน้อง ในคณะเทคโนโลยีสารสนเทศ ที่ให้กำลังใจตลอดมา ขอขอบคุณอาจารย์และเจ้าหน้าที่ทุกท่านของคณะเทคโนโลยีสารสนเทศ ที่ให้ความรู้และช่วยเหลืออย่างดียิ่ง

สุดท้ายขอขอบคุณบัณฑิตวิทยาลัย ที่ได้ให้ทุนสนันสนุนการทำวิทยานิพนธ์ครั้งนี้

คุณค่าและประโยชน์จากวิทยานิพจน์นี้ข้าพเจ้าขอมอบแด่ผู้มีพระคุณทุกท่าน

สมคิด แสนเสนาะ

สารบัญ

หน้า

สารบัญ (ต่อ)

หน้า

สารบัญ (ต่อ)

หน้า

สารบัญตาราง

สารบัญรูป

รูปที่ หน้า

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ความก้าวหน้าอย่างรวดเร็วของเทคโนโลยีคอมพิวเตอร์ในปัจจุบัน ได้ทำให้ข้อมูลไม่ว่าจะเป็นข้อมูลที่อยู่บนเครือข่ายอินเทอร์เน็ต หรือ ข้อมูลที่หน่วยงานและองค์กรต่างๆได้เก็บไว้ในรูปแบบที่สามารถใช้งานได้กับคอมพิวเตอร์ ได้มีจำนวนเพิ่มขึ้นอย่างรวดเร็ว ซึ่งทั้งความหลากหลายและปริมาณที่มากของข้อมูล จึงเป็นการยากและต้องใช้เวลามากในการที่จะจัดการหรือ ค้นหาข้อมูลที่ต้องการได้โดยไม่ใช้เครื่องมือใดๆช่วยในการค้นหา ดังเช่น ในปัจจุบันได้มีโปรแกรมประยุกต์ประเภทเสิร์จเอนจิ้นช่วยให้ผู้ใช้งานอินเทอร์เน็ตสามารถค้นหาข้อมูลที่ต้องการได้อย่างรวดเร็ว ซึ่งหลักการทำงานที่เป็นส่วนสำคัญของโปรแกรมประเภทนี้คือ จะทำการรวบรวมข้อมูลหน้าเวปเพจที่อยู่บนอินเทอร์เน็ตมาเก็บไว้ให้มากที่สุด แล้วนำข้อมูลเหล่านั้นมาผ่านกระบวนการจัดแบ่งกลุ่มข้อมูล เพื่อที่จะจัดแบ่งข้อมูลทั้ง หมดนั้นออกเป็นกลุ่มๆไปตามหมวดหมู่ตามที่ได้มีการกำหนดไว้ หรือ อาจจะเป็นการจัดแบ่งกลุ่มตามความเหมือนกันของเอกสารก็ได้ ซึ่งวัตถุประสงค์ของการจัดเก็บข้อมูลเป็นหมวดหมู่ตามกลุ่มของข้อมูลที่ได้มานั้น ก็เพื่อที่จะทำให้การค้นหาข้อมูลตามที่ผู้ใช้ร้องขอมานั้นเป็นไปอย่างรวดเร็ว และมีความถูกต้องตรงตามความต้องการของผู้ใช้งานมากที่สุด

การจัดกลุ่มเอกสาร (Document Clustering) คือกระบวนการในการแบ่งกลุ่มเอกสารออกเป็นกลุ่มๆโดยอัตโนมัติด้วยคอมพิวเตอร์ กล่าวคือ เป็นการทำงานบนสมมุติฐานที่ว่าเราไม่รู้เลยว่าเอกสารทั้งหมดที่เราจะจัดแบ่งกลุ่มนั้นจะแบ่งได้เป็นจำนวนกี่กลุ่ม และแต่ละกลุ่มประกอบด้วยเอกสารใดบ้าง ตัวอย่างเช่น สมมุติว่าเรามีเอกสารข่าวอยู่จำนวนหนึ่ง เราก็จะนำเอาคุณลักษณะ (feature) บางคุณลักษณะที่เด่นๆของแต่ละเอกสารที่มีอำนาจจำแนกแยกแยะเอกสารออกจากกันได้ เช่น หัวข้อข่าว (title) และ คำสำคัญ (keyword) ของข่าว เป็นต้น นำมาเป็นข้อมูลอินพุท ให้กับอัลกอริทึมสำหรับจัดแบ่งกลุ่มเอกสาร ซึ่งผลลัพธ์ที่ได้หลังจากเสร็จสิ้นกระบวนการทำงาน คือ เอกสารจะถูกแบ่งออกเป็นกลุ่มๆ โดยที่เอกสารที่อยู่กลุ่มเดียวกัน จะมีค่าความเหมือนกัน (similarity) กับเอกสารที่อยู่ในกลุ่ม (cluster) เดียวกันสูงกว่าเอกสารที่อยู่ในกลุ่มอื่นๆ

หลักของการจัดกลุ่มข้อมูลทั่วๆไป คือการที่เราจะต้องเลือกเอาคุณลักษณะเด่นบางอย่างของข้อมูลมาผ่านอัลกอริทึม เพื่อที่ช่วยจำแนกแยกแยะว่าข้อมูลแต่ละตัวหรือแต่กลุ่มมันมีความเหมือนกันหรือต่างกันมากน้อยแค่ไหน โดยที่ได้มีหลายๆอัลกอริทึมที่ใช้ในการจัดกลุ่มข้อมูลที่มีอินพุทเป็นชุดข้อมูลแบบเวกเตอร์ เช่น อัลกอริทึม K-means, Hierarchical Clustering, Fuzzy C-means และ Adaptive Resonance Theory (ART) เป็นต้น

โดยที่อัลกอริทึม K-means เป็นเทคนิคที่สามารถนำมาใช้กับการจัดกลุ่มข้อมูลเอกสารได้ แต่ไม่ค่อยจะได้ผลที่ดีมากนัก เนื่องจากผลรวมที่เป็นค่าทางสถิติที่ได้เป็นค่าเฉลี่ยของแต่ละกลุ่มทำให้สมาชิกแต่ละตัวในแต่ละกลุ่มอาจมีค่าไม่เหมือนกัน (Dissimilarity) ได้สูง และค่าเฉลี่ยอาจจะไม่ใช่ค่าผลรวมที่ดีของสมาชิกทั้งหมดในกลุ่ม และเมื่อจำนวนของกลุ่มมีมากขึ้นอัลกอริทึมก็จะมีการทำงานนานขึ้นแต่อย่างไรก็ตามสำหรับการแบ่งเอ้าท์พุทเพียงสองสามกลุ่มและ การนำไปใช้ในการเตรียมข้อมูลก่อนนำไปประมวลผล (pre-selected word) ตัวอัลกอริทึม K-means สามารถใช้ได้ดี ส่วนเทคนิคแบบ Hierarchical Agglomerative Clustering จะทำงานโดยการรวมเอกสารเข้าด้วยกัน (merge)โดยการทำซ้ำๆ ไปเรื่อยๆ ทำให้ได้โครงสร้างของข้อมูลเป็นชั้นๆ สำหรับในแต่ละครั้งของการทำซ้ำ การวัดความแตกต่างระหว่างกลุ่มทำได้โดยการหาความเหมือนกันของแต่ละกลุ่มจาก ข้อมูลแต่ละตัว ซึ่งจะต้องใช้การคำนวณอย่างมาก แต่มันก็ให้ค่าความแตกต่างของข้อมูลที่อยู่ใกล้เคียงกันได้อย่างดี โดยที่เทคนิคนี้เหมาะสมกับข้อมูลแบบ nearest-neighbor เนื่องจากการทำงานแบบวนกลับไปกลับมา (recursive) เพื่อรวมคู่ของข้อมูลที่เหมือนกันเข้าเป็นกลุ่มเดียวกัน ดังนั้น เทคนิคนี้จึงไม่เหมาะกับการทำงานที่มีข้อมูลมากเนื่องจากจะต้องใช้เวลาในการคำนวณนานนั้นเอง เทคนิคต่อมา อัลกอริทึม Fuzzy C-means เป็นอัลกอริทึมที่มีหลักการทำงานบนพื้นฐานของฟัซซีเซ็ต จากคุณสมบัติของ ฟัซซี ทำให้วิธีการจัดกลุ่มแบบ ฟัซซี สามารถที่จะแบ่งกลุ่มข้อมูลที่ทำให้สมาชิกตัวหนึ่งๆ สามารถอยู่ได้มากว่าหนึ่งกลุ่มในเวลาเดียวกัน ซึ่งนิยมใช้กับการแบ่งกลุ่มข้อมูลที่ไม่สามารถแบ่งออกได้อย่างชัดเจน เช่น การจัดแบ่งกลุ่มรูปภาพ (Image Clustering) เป็นต้น ส่วนอัลกอริทึม Adaptive Resonance Theory (ART) เป็นอัลกอริทึมที่เป็นแบบโครงข่ายประสาทเทียม (Neural Network) โดยที่อัลกอริทึมแบบโครงข่ายประสาทเทียมนี้จะมีความง่ายในการนำไปใช้งานหลังจากที่เราได้เทรนนิ่งตัวโครงข่าย (model) จนถึงจุดที่เราพอใจแล้ว แต่ข้อเสียของเทคนิคที่ใช้โครงข่ายประสาทเทียม คือเป็นการยากที่เราจะสามารถเข้าใจความหมายที่แฝงอยู่ที่มันได้เก็บไว้ในweight ของตัวมันหลังจากสิ้นสุดการเทรนนิ่ง

จากที่ได้กล่าวมาแล้วว่า เทคนิคการจัดกลุ่มข้างต้นจะใช้กับข้อมูลที่มีอินพุทเป็นค่าเชิงตัวเลขแต่เมื่อข้อมูลที่ต้องการจัดแบ่งกลุ่มเป็นข้อความ (text) เทคนิคเดิมที่มีและเป็นที่นิยมใช้กันอย่างแพร่หลายคือการแปลง (Transform) ข้อมูลที่เป็นข้อความเหล่านั้นให้อยู่ในรูปแบบของเวกเตอร์ที่เรียกว่าเวกเตอร์สเพซโมเดล ซึ่งแนวความคิดหลักของ เวกเตอร์สเพซโมเดล คือ จะแทนแต่ละเอกสารด้วยเวกเตอร์ weight ของคำที่เกิดบ่อยในแต่ละเอกสารซึ่งแต่ละสมาชิกของเวกเตอร์คือแทนการมีอยู่ (presence) หรือไม่มีอยู่ (absence) ของคำในเอกสารนั้นๆ ซึ่งวิธีการนี้ถึงแม้สามารถใช้งานได้ดีกับการแบ่งกลุ่มเอกสาร แต่เมื่อใช้กับเอกสารจำนวนมากๆ จะทำให้เกิด High – Dimensional Vector ของคำ ซึ่งทำให้คำแต่ละคำในเอกสารไม่เป็นอิสระจากคำอื่นๆ [1]

โดยประเด็นสำคัญของวิทยานิพนธ์ฉบับนี้ได้นำเสนอเทคนิคการจัดแบ่งกลุ่มเอกสาร โดยใช้อัลกอริทึมแบบโครงข่ายประสาทเทียมเรียนรู้แบบหาผู้ชนะ (Competitive Learning Neural

Network) จุดเด่นของอัลกอริทึมนี้คือง่ายในการพัฒนา โดยที่ข้อมูลอินพุทสำหรับโครงข่ายประสาทเทียมของเราจะแตกต่างจากวิธีการนำอินพุทเข้าโครงข่ายประสาทเทียมทั่วๆไป ที่รับอินพุทเป็นค่าตัวเลข แต่อัลกอริทึมของเราจะรับอินพุทที่เป็นข้อความเข้าไปโดยตรง ซึ่งตัวอัลกอริทึมในส่วนของการเรียนรู้เพื่อหาผู้ชนะจะทำงานร่วมกับหลักการวัดความต่าง (Dissimilarity Measure) ที่ใช้กับ Symbolic Object ซึ่งข้อดีของวิธีที่ใช้ในงานวิจัยนี้คือจะแก้ปัญหาของการเกิด High – Dimensional Vector ของคำ และความยุ่งยากในการแปลงข้อมูลลง

1.2 ความมุ่งหมายและวัตถุประสงค์ของงานวิจัย

1.2.1 เพื่อศึกษาพัฒนาโครงข่ายประสาทเทียมที่เรียนรู้แบบหาผู้ชนะมาใช้ในการจัดกลุ่มเอกสารที่สามารถ รับอินพุทเป็นข้อความได้โดยตรง

1.2.2 เพื่อศึกษาวิธีการวัดความแตกต่างของเอกสาร

1.2.3 เพื่อเป็นแนวทางในการพัฒนาโครงข่ายประสาทเทียมในการจัดกลุ่มเอกสารต่อไปในอนาคต

1.3 สมมุติฐานของการวิจัย

1.3.1 เอกสารที่ใช้ในการศึกษาคือข่าว Reuters-21578 ซึ่งเป็นข่าวที่มีเนื้อข้อความเป็นภาษาอังกฤษ

1.3.2 เอกสารข่าวต้องมีครบสมบูรณ์ทั้ง หัวเรื่องข่าว (TOPICS) หัวข้อข่าว (TITLE) และเนื้อข่าว (BODY)

1.4 ขอบเขตการดำเนินงานวิจัย

1.4.1 งานวิจัยนี้ครอบคลุมเฉพาะการจัดกลุ่มเอกสารข่าว Reuters-21578

1.5 ขั้นตอนการศึกษา

1.5.1 นำข่าวทั้งหมดมาตัดเอาเฉพาะเนื้อหาที่ต้องการ

1.5.2 นำข่าวทั้งหมดจากข้อ 1.5.1 มาหาคำสำคัญด้วยโปรแกรม Copernic Summarizer

1.5.3 นำข่าวจากข้อ 1.5.2 เข้าโครงข่ายประสาทเทียมที่พัฒนาขึ้น

1.5.4 สรุปผลและ วิเคราะห์ปัญหาที่เกิดขึ้นพร้อมชี้แนะแนวทางการแก้ไขปัญหา

1.6 ประโยชน์ที่คาดว่าจะได้รับ

1.6.1 เพื่อพัฒนาโครงข่ายประสาทเทียมที่เรียนรู้แบบหาผู้ชนะกับงานจัดกลุ่มเอกสาร

1.6.2 เป็นพื้นฐานในการพัฒนาโปรแกรมประยุกต์กับงานจัดกลุ่มเอกสารแบบอัตโนมัติ

1.7 โครงสร้างวิทยานิพนธ์

วิทยานิพนธ์ฉบับนี้ ประกอบด้วยรายละเอียดของเนื้อหาเป็นบทต่างๆที่ผู้เขียนได้เรียบเรียงขึ้นดังนี้

บทที่ 1 บทนำ ได้กล่าวถึงความเป็นมา และชี้ประเด็นสำคัญของปัญหา โดยได้สรุปงานวิจัยบางชิ้นที่เกี่ยวข้องเพื่อให้ผู้ศึกษาวิทยานิพนธ์ฉบับนี้ได้เห็นภาพกว้างของงานวิจัยฉบับนี้ และได้กล่าวถึงความมุ่งหมาย วัตถุประสงค์ของการศึกษา สมมุติฐานของการวิจัย ขอบเขตของการศึกษา ขั้นตอนการศึกษา และประโยชน์ที่คาดว่าจะได้รับจากงานวิจัยนี้

บทที่ 2 ทฤษฎีหลักและงานวิจัยที่เกี่ยวข้อง ในบทนี้จะกล่าวถึงตัวทฤษฎีโครงข่ายประสาทเทียมที่เรียนรู้แบบหาผู้ชนะ (Competitive Learning Neural Network) รวมทั้งทฤษฎีการวัดความต่างของเอกสาร (Dissimilarity Measure) ที่นำมาประยุกต์ใช้ร่วมกับตัวโครงข่ายประสาทเทียม และได้แสดงตัวอย่างการคำนวณของสูตรต่างๆเพื่อให้ผู้ศึกษาวิทยานิพนธ์มีความเข้าใจมากยิ่งขึ้น

บทที่ 3 การแบ่งกลุ่มเอกสารโดยใช้เทคนิคการประมวลผลข้อความด้วยโครงข่ายประสาทเทียมที่เรียนรู้แบบหาผู้ชนะ (Document Clustering Using a Text Processing Competitive Learning Neural Network) ในบทนี้จะกล่าวถึงอัลกอริทึมที่พัฒนาขึ้นพร้อมทั้ง แสดงขั้นตอนการประมวลผลของอัลกอริทึมอย่างละเอียด และอธิบายถึงขบวนการเรียนรู้ (Learning Rule) ของตัวอัลกอริทึม

บทที่ 4 วิธีการดำเนินการวิจัย บทนี้จะกล่าวถึงขั้นตอนการเตรียมข้อมูลซึ่งข้อมูลที่ใช้จะแบ่งเป็นสองส่วนคือ ส่วนที่หนึ่งข้อมูลที่ใช้สำหรับทดสอบตัวอัลกอริทึมเพื่อประเมินประสิทธิภาพของตัวอัลกอริทึมจะเป็นข้อมูลที่สุ่มขึ้นมาจากต้นแบบ (Profile) ที่กำหนดขึ้นเอง และส่วนที่สองเป็นข้อมูลข่าวที่จะใช้ในการทดลองจริง และ แสดงผลการทดลองของข้อมูลทั้งสองแบบ

บทที่ 5 สรุปผลงานวิจัยและข้อเสนอ บทนี้จะสรุปภาพรวมทั้งหมดที่ได้จากงานวิจัย และกล่าวถึงปัญหาบางประการที่พบจากงานวิจัย พร้อมทั้งได้เสนอแนะแนวทาง ในการวิจัยต่อไป

บทที่ 2
ทฤษฎีหลักและงานวิจัยที่เกี่ยวข้อง

2.1 เทคนิคการจัดกลุ่ม (Clustering Algorithm Techniques)

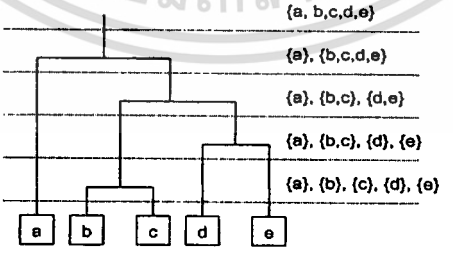
การจัดกลุ่ม (Clustering) คือการแบ่งชุดของข้อมูลทั้งหมดที่เราสนใจออกเป็นกลุ่มๆ โดยใช้คุณลักษณะ ของข้อมูลเป็นตัวแบ่งแยกข้อมูลโดยที่แต่ละกลุ่มที่ได้นั้นจะเรียกว่า คลัสเตอร์ ซึ่งในแต่ละคลัสเตอร์จะประกอบด้วยข้อมูลที่มีค่าความเหมือนกันภายในคลัสเตอร์เดียวกันสูงและในทำนองกลับกันค่าความเหมือนกันของข้อมูลระหว่างแต่ละคลัสเตอร์ ก็จะมีค่าต่ำกว่าข้อมูลที่อยู่ภายในคลัสเตอร์เดียวกัน

2.1.1 การจัดกลุ่มเอกสาร (Document Clustering)

เทคนิคหลักๆ ของการจัดกลุ่มเอกสารแบ่งได้เป็นสองแบบ คือ Hierarchical Clustering และ Partitional Clustering

2.1.1.1. Hierarchical Clustering

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



รูปที่ 2.1 แสดงแผนภาพ Dendogram ของ Hierarchical Clustering

2.1.1.1.1 Agglomerative Hierarchical Clustering (AHC)

วิธีนี้จะเริ่มจากชุดของข้อมูลเดียวในแต่ละคลัสเตอร์ แล้วที่แต่ละขั้นจะทำการรวม (merge) คลัสเตอร์ที่เหมือนกันมากที่สุดสองคลัสเตอร์เข้าด้วยกัน ซึ่งจะทำซ้ำๆในขั้นตอนนี้จนกระทั่งได้จำนวนของคลัสเตอร์ที่น้อยที่สุดแล้ว หรือถ้าต้องการให้ได้ลำดับชั้นที่สมบูรณ์ก็ต้องทำขั้นตอนนี้ไปจนกระทั่งเหลือคลัสเตอร์เพียงหนึ่งคลัสเตอร์ ซึ่งคู่ของกลุ่มเอกสารที่ถูกเลือกสำหรับการรวมนั้นได้จากการพิจารณาว่ามีความเหมือนกันมากที่สุดภายใต้เงื่อนไขที่กำหนด ซึ่งมันเป็นวิธีที่ไม่ยุ่งยาก แต่จำเป็นต้องใช้การคำนวณหาค่าระยะห่าง (Distance) ระหว่างสองคลัสเตอร์ก็จะได้เป็นค่าความเหมือนกันของแต่ละคลัสเตอร์ มีอยู่สามวิธีที่นิยมใช้กันสำหรับการคำนวณเพื่อค่าระยะห่างระหว่างคลัสเตอร์ ดังนี้

- Single Linkage Method ความเหมือนกันระหว่างคลัสเตอร์ S และ T คำนวณได้จากค่าความแตกต่างที่ต่ำที่สุดระหว่างแต่ละสมาชิกที่อยู่ในคลัสเตอร์ทั้งสอง วิธีนี้เรียกว่าการจัดกลุ่มแบบ " nearest neighbor "

$$\|T-S\| = \min_{\substack{x\in T \\ y\in S}} \|x-y\| \tag{2.1}$$

- Complete Linkage Method ความเหมือนกันระหว่างคลัสเตอร์ S และ T คือคำนวณบนพื้นฐานของ ความแตกต่างที่สูงที่สุดระหว่างสมาชิก ที่อยู่ในคลัสเตอร์ทั้งสอง วิธีนี้เรียกว่าการจัดกลุ่มแบบ " furthest neighbor "

$$\|T-S\| = \max_{\substack{x\in T \\ y\in S}} \|x-y\| \tag{2.2}$$

- Average Linkage Method ความเหมือนกันระหว่างคลัสเตอร์ S และ T คำนวณได้จากค่าเฉลี่ยของความแตกต่างระหว่างแต่ละสมาชิกที่อยู่ในคลัสเตอร์ทั้งสอง ซึ่งวิธีนี้จะทำโดยการนำสมาชิกทั้งหมดมาหาความแตกต่างระหว่างกันเป็นคู่ๆ จนครบจำนวนสมาชิกทุกตัว วิธีนี้รู้จักกันในชื่อ UPGMA (Unweighted Pair - Group Method using Arithmetic Average)

$$\|T-S\| = \frac{\sum_{y\in S}^{x\in T}\|x-y\|}{|S|\bullet|T|} \tag{2.3}$$

2.1.1.1.2 Divisive Hierarchical Clustering (DHC)

วิธีนี้จะเริ่มทำงานจากบนลงล่างโดยเริ่มต้นที่ชุดข้อมูลทั้งหมดคือหนึ่งคลัสเตอร์ และ แต่ละขั้นตอนจะแยก (split) ออกเป็นคลัสเตอร์ๆ จนกระทั่งเหลือเพียงคลัสเตอร์เดียวเดี่ยวๆ โดยมีสองสิ่งที่เราพิจารณาคือ (1) คลัสเตอร์ไหนจะถูกแยกออกไป และ (2) จะแยกอย่างไร โดยปกติแล้วจะใช้วิธีการค้นหาจนทั่วทั้งหมดในการหาคลัสเตอร์ที่จะแยก การใช้วิธี ค้นหาแบบนี้ ทำให้การทำงานจะช้ามาก เพื่อให้การทำงานเร็วขึ้นได้มีการใช้วิธีง่ายๆคือการเลือกคลัสเตอร์ที่มีขนาดใหญ่สุดในการแยก และกับวิธีพิจารณาว่าคลัสเตอร์ไหนเหมือนกันน้อยที่สุด หรือจะใช้ทั้งเงื่อนไข ขนาด และความเหมือนกันก็ได้ Steinbach และคณะ [1] ได้ศึกษาเปรียบเทียบวิธีทั้งสองแล้วพบว่าผลที่ได้มีความแตกต่างกันน้อยมาก ดังนั้นการคลัสเตอร์ที่ใหญ่ที่สุดมาแบ่งเป็นคลัสเตอร์ย่อยต่อไปจึงเป็นวิธีที่ง่ายและรวดเร็ว

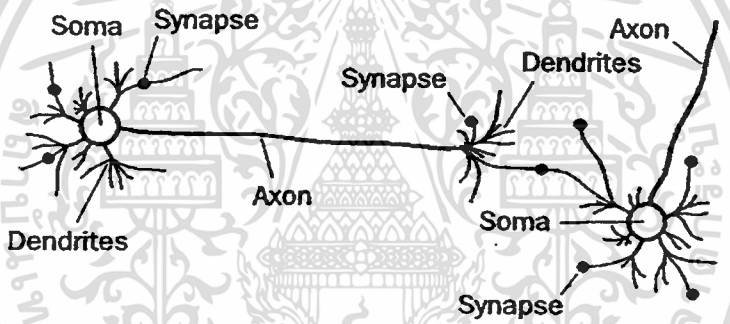
2.1.1.2 Partitional Clustering

Partitional Clustering ทำงานโดยระบุคลัสเตอร์ขึ้นมาจำนวนหนึ่งก่อนพร้อมๆกัน โดยที่ในแต่ละคลัสเตอร์จะประกอบไปด้วยข้อมูลทั้งหมดที่กระจายกันไปอยู่ในแต่ละคลัสเตอร์ แล้วทำการปรับ (update) คลัสเตอร์ซ้ำๆด้วยฟังค์ชั่นที่จะทำให้ได้คลัสเตอร์เหลือจำนวนน้อยที่สุด การปรับคลัสเตอร์ไม่ใช่การนำคลัสเตอร์ที่มีอยู่มารวมกัน แต่เกิดจากปรับเปลี่ยนโยกย้ายของข้อมูลระหว่างคลัสเตอร์เหล่านั้น ซึ่งอัลกอริทึมแบบ Partitional Clustering ไม่ได้มีการสร้าง Dendogram เหมือนวิธี Hierarchical Clustering โดยที่อัลกอริทึมแบบ Partitional Clustering ที่รู้จักกันดีเช่นอัลกอริทึมในกลุ่ม K-means เป็นต้น

อัลกอริทึม K-means จะมีการทำงานโดยริ่มจากการกำหนดคลัสเตอร์แบบสุ่มขึ้นมาจำนวน k คลัสเตอร์ (ซึ่งจะกำหนดจุดศูนย์กลางของแต่ละคลัสเตอร์ขึ้นมาพร้อมกันด้วย) ต่อจากนั้นให้ข้อมูลแต่ละตัวไปเป็นสมาชิกอยู่ในคลัสเตอร์ที่อยู่ใกล้กับมันมากที่สุด โดยการวัดระยะห่างระหว่างตัวมันกับจุดศูนย์กลางของคลัสเตอร์ ตัวอัลกอริทึมจะทำการคำนวณไปเรื่อยๆจนครบจำนวนของข้อมูลทุกตัวและทุกคลัสเตอร์ เพื่อให้ข้อมูลไปเป็นสมาชิกอยู่ในคลัสเตอร์ใดคลัสเตอร์หนึ่ง เมื่อเสร็จสิ้นขั้นตอนนี้แล้วอัลกอริทึมก็จะทำการหาจุดศูนย์กลาง ของทุกๆคลัสเตอร์ ที่มีข้อมูลอยู่ใหม่อีกครั้ง โดยจะหาจากค่าเฉลี่ยของระยะห่างจากจุดศูนย์กลางกับสมาชิกทุกตัวในคลัสเตอร์ ซึ่งจำนวนของคลัสเตอร์อาจจะมีการเปลี่ยนแปลงได้ในขั้นตอนนี้ โดยจำนวนจะลดลงหรือคงที่เท่านั้น และอัลกอริทึมจะวนกลับ ไปเริ่มทำการคำนวณระยะห่างระหว่างข้อมูลแต่ละตัวกับจุดศูนย์กลางของทุกๆคลัสเตอร์ที่ได้ใหม่นี้อีกครั้ง และจะทำซ้ำๆขั้นตอนเหล่านี้ไปจนกระทั่งจำนวนของข้อมูลในแต่ละคลัสเตอร์ไม่มีการเปลี่ยนแปลง จะเห็นได้ว่าอัลกอริทึม K-means เหมาะสมกับโครงสร้างของคัลสเตอร์ที่เป็นรูปทรงกลม นอกจากนี้การกำหนดคลัสเตอร์เริ่มต้นก็เป็นสิ่งสำคัญมากของอัลกอริทึม K-means ที่จะทำให้ผลของการจัดกลุ่มมีคุณภาพ

2.2 โครงข่ายประสาท (Neural Network)

โครงข่ายประสาทภายในสมองของมนุษย์ เป็นสิ่งสำคัญที่ธรรมชาติให้มนุษย์มาโดยที่ความพิเศษของโครงข่ายประสาทเทียมภายในสมองมนุษย์นั้นคือมันมีความสามารถ "คิดหาเหตุผล" ซึ่งทำให้มนุษย์เป็นสัตว์ที่ฝึกและเรียนรู้ได้ดีกว่าสัตว์ใดๆ ก็เพราะว่ามนุษย์มีสมองที่น่าอัศจรรย์นั้นเอง ซึ่งสมองของมนุษย์ประกอบด้วย เซลประสาทที่เรียกว่า นิวรอล (neuron) ที่เชื่อมต่อซึ่งกันและกันอย่างหนาแน่น อยู่ภายในสมองมนุษย์ราวๆ 10 พันล้านนิวรอล และ การเชื่อมต่อระหว่างนิวรอลนั้นเราเรียกว่า synapse ซึ่งจะมีอยู่ราวๆ 60×10^3 การเชื่อมต่อ [3] ดังนั้นการใช้หลายๆนิวรอลประมวลผลพร้อมๆกัน ทำให้สมองมนุษย์สามารถปฏิบัติงานที่ซับซ้อนได้เร็วกว่าการทำงานของเครื่องคอมพิวเตอร์ในปัจจุบันมาก โดยที่แต่ละนิวรอลจะมีโครงสร้างง่ายๆที่ ประกอบด้วยตัวเซล หรือ soma และมีเส้นประสาทแยกออกไปสองกิ่งหลักๆ ที่เรียกว่า dendrite แบบหนึ่ง และเส้นประสาทเดี่ยวยาวที่เรียกว่า axon แบบหนึ่ง ในตัวเซลจะ มี นิวเคลียส์ (nucleus) อยู่ตรงกลาง ซึ่งบรรจุไปด้วยข้อมูล (information) เกี่ยวกับพันธุ์กรรม และมีของเหลวที่เรียกว่า plasma บรรจุห่อหุ้มอยู่ ตามรูป 2.2 คือแผนภาพวาดของโครงข่ายประสาทในสมองมนุษย์



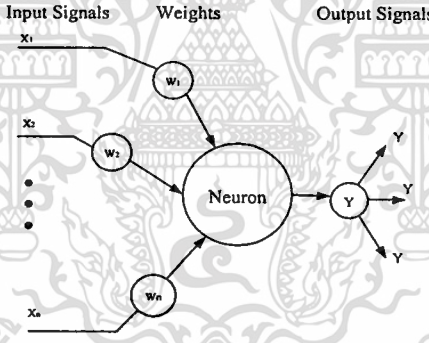
รูปที่ 2.2 แสดงแผนภาพวาดของ โครงข่ายประสาทในสมองมนุษย์

นิวรอลแต่ละตัวจะรับสัญญาณ (impulse) จากนิวรอลตัวอื่นๆ ผ่านทาง dendrite (receiver) ของมันและส่งผ่านสัญญาณที่กำเนิดขึ้นโดยตัวเซลของมันเองไปตามเส้นประสาทaxon (transmitter) ซึ่งสัญญาณจะผ่านเส้นประสาทนี้ไปยังจุดปลายทาง ที่เรียกว่า synapse โดยที่ synapse คือ จุดที่เชื่อมต่อระหว่างสองนิวรอล (axon ของนิวรอลหนึ่ง กับ dendrite ของอีกนิวรอลหนึ่ง) เมื่อสัญญาณไปถึงจุดปลายของตัว synapse แล้วก็จะเกิดปฏิกิริยาเคมีขึ้นมาและต่อจากนั้น ปฏิกิริยาเคมีก็จะถูกแปลงเป็นสัญญาณไฟฟ้า และถูกปล่อยออกมาโดยตัวที่เรียกว่า neurotransmitter ซึ่งสัญญาณจะแพร่ข้ามช่องว่าง synapse (ไปยัง dendrite ของนิวรอลอื่นต่อไป) และจะทำให้เกิดผลกระทบขึ้นกับ synapse โดยที่ผลกระทบที่เกิดกับ synapse สามารถจะถูกปรับได้โดยสัญญาณ (การเพิ่มขึ้นหรือลดลงของศักย์ไฟฟ้า) ที่ผ่านตัวมันดังนั้นตัว synapse จึงสามารถเรียนรู้จากกิจกรรมที่มันเข้าไปมี

ส่วนร่วมได้ โดยจะขึ้นอยู่กับพฤติกรรมในอดีตที่ผ่านมาด้วย ดังนั้นการที่ความจำของมนุษย์ ซึ่งสามารถตอบสนองต่อการระลึกหรือจดจำได้นั้น เกิดจากการที่ตัวนิวรอลมีความสามารถที่จะเรียนรู้โดยผ่านประสบการณ์ ที่มันเคยมีส่วนร่วมมานั่นเอง ซึ่งโดยที่การเรียนรู้คือพื้นฐานและคุณลักษณะที่จำเป็นของโครงข่ายประสาทในสมองของมนุษย์ และด้วยความสามารถที่มันเรียนรู้เรียนรู้ ได้นี้เอง ได้เกิดความพยายาม ที่จะจำลองการทำงานของโครงข่ายประสาทนี้มาใช้ในคอมพิวเตอร์ ที่เรียกกันว่า "โครงข่ายประสาทเทียม (Artificial Neural Network)"

2.3 โครงข่ายประสาทเทียม (Artificial Neural Network)

โครงข่ายประสาทเทียม ประกอบด้วยตัวประมวลผลที่เรียกว่า "นิวรอล" ที่เชื่อมต่อถึงกันและนิวรอลแต่ละตัวก็จะส่งผ่านสัญญาณจากตัวมันไปยังนิวรอลตัวอื่นๆที่อยู่ต่างเลเยร์กัน ผ่านทางจุดเชื่อมต่อที่เรียกว่า weight โดยที่ตัวนิวรอลที่อยู่ในเลเยร์เดียวกันจะมีหน้าที่เหมือนกัน ซึ่งแต่ละนิวรอลจะรับค่าสัญญาณอินพุทที่เชื่อมต่อกับตัวมันทั้งหมดมาประมวลผล และจะให้สัญญาณออกมาที่เอ้าท์พุทเพียงหนึ่งค่าเท่านั้น โดยที่สัญญาณเอ้าท์พุทที่ได้จากแต่ละเลเยร์จะถูกส่งไปยังนิวรอลในเลเยร์ปลายทางต่อไป รูปที่ 2.3 แสดงแสดงแผนภาพของ Artificial Neural Network และ ตารางที่ 2.1 แสดงเปรียบเทียบระหว่าง Biological กับ Artificial Neural Network



รูปที่ 2.3 แสดงแผนภาพของโครงข่ายประสาทเทียม

ตารางที่ 2.1 แสดงการเปรียบเทียบระหว่าง Biological กับ Artificial Neural Network [3]

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

2.3.1 การเรียนรู้ของโครงข่ายประสาทเทียม

ในแต่ละนิวรอลจะเชื่อมต่อกันด้วย weight ซึ่งจะมีค่าเลขน้ำหนัก ที่สัมพันธ์กับตัวมัน โดยที่ค่า weight จะเป็นส่วนที่เก็บข้อมูลของการเรียนรู้ของโครงข่ายประสาทเทียม ตัวโครงข่ายประสาทเทียมจะ 'เรียน' ผ่านการปรับเปลี่ยนไปมาซ้ำๆ ของ weight ตามสภาวะแวดล้อมภายนอกจากผ่านทางโหนดอินพุท การสร้างโครงข่ายประสาทเทียมในขั้นต้นเราจะต้องกำหนดก่อนว่าโครงข่ายของเราจะให้มีจำนวนนิวรอลเท่าไร แล้วจึงตัดสินใจเลือก learning algorithm ที่จะใช้ และสุดท้ายเราจะเทรนนิ่งตัวนิวรอลเน็ตเวิร์คด้วยค่า weight เริ่มต้นของโครงข่ายและปรับ weight ของโครงข่าย จากชุดข้อมูลที่ใช้ในการเทรนนิ่ง ในปี 1943 Warren McCulloch และ Walter Pilts [4] ได้นำเสนอแนวคิดง่ายๆที่ยังคงเป็นพื้นฐานของโครงข่ายประสาทเทียมในปัจจุบัน โครงข่ายประสาทเทียมที่ว่าคือตัวนิวรอลจะคำนวณผลรวมของ weight กับข้อมูลอินพุทและเปรียบเทียบผลลัพธ์กับค่า threshold (θ) ถ้าค่าผลรวมอินพุทน้อยกว่าค่า threshold ตัวนิวรอลเอ้าท์พุทจะได้ค่า -1 แต่ถ้าค่าผลรวมอินพุทมากกว่าหรือเท่ากับ threshold ตัวนิวรอลเอ้าท์พุทจะได้ค่า +1 ซึ่งนิวรอลใช้ฟังค์ชั่นกระตุ้นดังสมการที่ 2.4

$$X = \sum_{i=1}^{n} x_i w_i \tag{2.4}$$

$$Y = \begin{cases} +1 & \text{if } X \geq \theta \\ -1 & \text{if } X < \theta' \end{cases}$$

เมื่อ X คือ ผลรวมของ weight กับข้อมูล อินพุท ของนิวรอลแต่ละตัว

x_i คือ ค่าของ อินพุท i

w_i คือ weight ของ อินพุท i

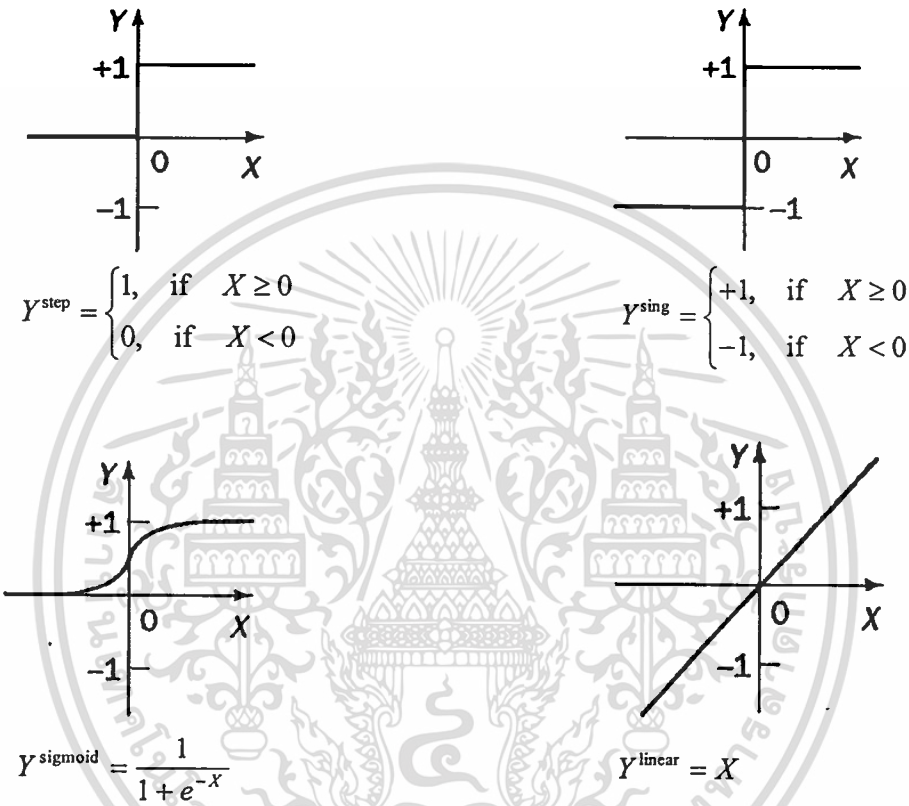
n คือ จำนวนของนิวรอลอินพุท

Y คือ เอ้าท์พุทนิวรอล

ซึ่งชนิดของฟังค์ชั่นกระตุ้นแบบนี้เรียกว่าฟังค์ชั่น sign โดยที่ผลที่เอ้าท์พุทของนิวรอลที่ทำงานด้วยฟังค์ชั่นกระตุ้น sing นี้สามารถเขียนได้ดังสมการที่ 2.5

$$Y = \text{sign}\left[\sum_{i=1}^{n} x_i w_i - \theta\right] \tag{2.5}$$

ฟังค์ชั่นกระตุ้นที่นิยมนำมาใช้มี 4 ชนิดคือ step ,sign ,line และ sigmoid ฟังค์ชั่นที่แสดงในรูปที่ 2.4 โดยที่ step และ sing ฟังค์ชั่นบางครั้งเรียกว่าเป็น hard limit ฟังค์ชั่น ซึ่งปกติมักจะนำไปใช้ในงานลักษณะช่วยการตัดสินใจ (decision making) เช่นสำหรับงาน classification และ pattern recognition เป็นต้นส่วนฟังค์ชั่น sigmoid ใช้สำหรับแปลงรูปอินพุท ให้มีค่าใดๆระหว่างบวก และ ลบ ไม่สิ้นสุดไปยังค่าที่ยอมรับได้ในช่วงระหว่าง 0 และ 1 โดยที่ฟังค์ชั่นแบบนี้จะนิยมใช้ในโครงข่ายแบบ back-propagation ส่วน ฟังค์ชั่น linear ปกติใช้สำหรับ linear approximation



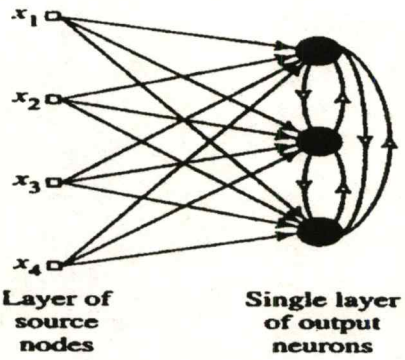
รูปที่ 2.4 แสดงฟังค์ชั่นกระตุ้น4 แบบที่นิยมใช้กัน

2.4 Competitive Learning and Winner – take – all Network

สถาปัตยกรรมของโครงข่ายประสาทเทียมแบบ Competitive Learning ประกอบชั้น (layer) ข้อมูลเพียงชั้นเดียวโดยที่แต่ละโหนดของนิวรอลเอ้าท์พุท จะเชื่อมต่อกับทุกๆโหนดของนิวรอลอินพุท ดังแสดงในรูปที่ 2.5

ซึ่งโหนดอินพุททั้งหมดจะรับค่าอินพุทชุดเดียวกันเข้ามาทำการประมวลผล โดยหลักการทำงานพื้นฐานของโครงข่ายประสาทเทียมแบบเรียนรู้เพื่อหาผู้ชนะนั้น ใช้หลักการของการเลือกผู้ชนะ ของนิวรอลเอ้าท์พุทที่แข่งขันกันในหมู่พวกมัน (winner – take – all network) ซึ่งเอ้าท์พุทที่ดี

ที่สุด (ค่าต่ำสุด หรือ สูงสุด ขึ้นอยู่กับเรากำหนด) ก็จะได้รับการประกาศให้เป็นผู้ชนะโดย โครงข่าย



รูปที่ 2.5 แสดงโครงสร้างของโครงข่ายประสาทเทียมที่เรียนรู้แบบหาผู้ชนะ

ประสาทเทียมแบบเรียนรู้เพื่อหาผู้ชนะนั้น จะมีเพียงโหนดเอ้าท์พุทเพียงโหนดเดียวเท่านั้นที่จะถูกประกาศให้เป็นผู้ชนะในแต่ละชุดของข้อมูลที่เข้ามา [5] เช่นสมมุติว่าเรามี N อินพุท $x_1, x_2 ... x_N$ และเราต้องการที่จะสร้าง เอ้าท์พุท N ตัว โดยที่เฉพาะนิวรอล เอ้าท์พุท ที่ตรงกันกับ อินพุท มากที่สุดตัวเดียวจะมีค่าเป็น 1 ส่วนนิวรอล เอ้าท์พุท อื่นทั้งหมดจะมีค่าเป็นศูนย์ เขียนเป็นสมการได้ดังนี้

$$y_k = \begin{cases} 1 & \text{if } x_k \text{ largest} \\ 0 & \text{if otherwise} \end{cases} \tag{2.6}$$

ซึ่งนิวรอลจะ เรียนรู้โดยการปรับ weight โดยถ้า นิวรอลไหนถูกประกาศว่าเป็นผู้ชนะ การปรับ weight ที่เชื่อมต่ออยู่กับโหนดชนะเขียนได้ตามสมการที่ 2.7

$$\Delta w_{kj} = \begin{cases} \eta\left(x_j - w_{kj}\right) & \text{if neuron } k \text{ wins the competition} \\ 0 & \text{if neuron } k \text{ loses the competition} \end{cases} \tag{2.7}$$

เมื่อ

$\eta =$ learning rate

$x =$ input pattern

$j =$ input node

จากรูปที่ 2.6 แสดงภาพเรขาคณิตทรงกลม ที่อธิบายการเรียนรู้ของโครงข่ายประสาทเทียมที่เรียนรู้แบบหาผู้ชนะ ซึ่งจุดแทนอินพุทเวกเตอร์ กากบาทแทน weight เวกเตอร์โดยขนาดของอินพุทเวกเตอร์ และ weight เป็นค่าที่ normalize แล้วซึ่งจะเห็นว่าทั้ง 12 อินพุทเวกเตอร์ จะสามารถแบ่งออกได้เป็น 3 เอ้าท์พุทโหนด [4]

การทำงานเริ่มต้นจาก weight เวกเตอร์จะถูกสุ่มขึ้นมาก่อนดังรูปที่ 2.6 (ก) เมื่อได้ weight เวกเตอร์แล้วจะเริ่มการเทรนนิ่ง ในขณะเทรนนิ่งนั้น weight จะคำนวณระยะห่างระหว่างตัวมันกับอินพุทเวกเตอร์ในแต่ละเอ้าท์พุทโหนด แล้วก็จะปรับค่าตัวเองไปเรื่อยๆตามสมการที่ 2.7 จนถึงจุดๆหนึ่งที่ weightไม่มีการเปลี่ยนแปลงแล้วจึงสิ้นสุดการเทรนนิ่ง ดังรูปที่ 2.6 (ข) โดยที่ weight เวกเตอร์ที่ได้แต่ละตัวจะอยู่ตรงจุดศูนย์กลางของแต่ละกลุ่มของอินพุทเวกเตอร์ ซึ่ง weight เวกเตอร์ก็คือค่าเฉลี่ยที่เป็นตัวแทนของ อินพุท เวกเตอร์ในแต่ละกลุ่มนั่นเอง

(ก) สภาวะเริ่มต้นการทำงาน (ข) สถาวะสิ้นสุดการทำงาน

รูปที่ 2.6 แสดงภาพเรขาคณิตทรงกลมที่อธิบายการเรียนรู้ของโครงข่ายประสาทเทียมที่เรียนรู้แบบหาผู้ชนะ

ซึ่งวิธีการของอัลกอริทึมที่เรียนรู้แบบหาผู้ชนะนี้ เปรียบเสมือนกับการใช้หลักสถิติ “ความน่าจะเป็น” ของข้อมูลอินพุท [6] ด้วยคุณสมบัตินี้ทำให้อัลกอริทึมที่เรียนรู้แบบหาผู้ชนะ เหมาะสมที่จะใช้ในการจัดแบ่งกลุ่มชุดข้อมูล ซึ่งในงานวิจัยนี้ได้นำจุดเด่นของอัลกอริทึมที่เรียนรู้แบบหาผู้ชนะ มาปรับใช้ร่วมกับการหาความต่างของ เอกสาร ซึ่งรายละเอียดจะกล่าวถึงในบทที่ 3

2.5 การแทนข้อความเอกสาร (Document Representation)

ความหมายของ เอกสาร สำหรับในงานวิจัยนี้ก็คือชุดของคำ (set of word) ที่ประกอบกันขึ้นในหลาย ๆ รูปแบบ โดยทั่วไปแล้วรูปแบบของคำที่เกิดขึ้นนี้จะมีความซับซ้อนมาก ดังนั้นเพื่อ

ลดความซับซ้อนของเอกสารลงและเพื่อให้ง่ายในการจัดการกับเอกสารจำนวนมาก เราจะแทนเอกสารฉบับเต็มแต่ละฉบับด้วยคุณลักษณะ (feature) ของเอกสาร ดังนั้นเอกสารหนึ่งเอกสารซึ่งในที่นี้จะเขียนแทนด้วยคำว่า Doc สามารถแทนได้ด้วยผลคูณ Cartesian ของคุณลักษณะของมัน ดังสมการที่ 2.8 แสดงการแทนข้อความเอกสารด้วยคุณลักษณะของมัน

$$\text{Doc} = D_1 * D_2 * ... * D_k \tag{2.8}$$

เมื่อ

D_k คือ คุณลักษณะ ตัวที่ k ของเอกสาร

ตัวอย่างเช่นในต้นฉบับเอกสารข่าว Reuters-21578 ที่ใช้ทดลองหนึ่งข่าวจะประกอบไปด้วยตัวอย่างคุณลักษณะที่สำคัญดังต่อไปนี้

ตารางที่ 2.2 แสดงตัวอย่างคุณลักษณะของเอกสารข่าว Reuters-21578

คุณลักษณะ	ความหมาย
DATE	วันที่เสนอข่าว
TOPICS	หัวเรื่องข่าว
PLACES	สถานที่เกิดข่าว
TITLE	หัวข้อข่าว
BODY	รายละเอียดเนื้อข่าว

ซึ่งงานวิจัยนี้ได้เลือกเอาคุณลักษณะของเอกสารข่าวมาเป็นข้อมูลอินพุท 2 คุณลักษณะ คือ คุณลักษณะ Title และ คุณลักษณะ Body โดยคุณลักษณะ Body ของเอกสารคือส่วนที่เราจะนำมาผ่านขั้นตอนการเลือกเฉพาะคำสำคัญจำนวนหนึ่งที่ใช้แทนเนื้อข่าวทั้งหมดของแต่ละข่าว ชุดของคำที่ได้ใหม่นี้เราจะเรียกเป็นคุณลักษณะ Keyword ดังนั้นเราจึงเขียนแทนเอกสารข่าวหนึ่งข่าวได้ด้วยผลคูณ Cartesian ของ คุณลักษณะ Title และ คุณลักษณะ Keyword ได้ดังนี้

$$\text{Doc} = \text{Title} * \text{Keyword} \tag{2.9}$$

โดยที่ คุณลักษณะ Title ของเอกสารที่ใช้ในการทดลองยังคงเป็นชุดของคำที่ใช้เป็นหัวข้อข่าวตามต้นฉบับเดิมและคุณลักษณะ Keyword ของเอกสารได้จากการหาคำที่เกิดขึ้นซ้ำๆจากเนื้อข่าว

เดียวกัน โดยจะกล่าวถึงรายละเอียดและเครื่องมือที่ใช้ในการหา Title และ Keyword ของเอกสารในส่วนของการเตรียมข้อมูล บทที่ 4 และในภาคผนวก ก

2.6 การวัดความแตกต่างกันของเอกสาร (Dissimilarity Measure of Document)

ปัญหาที่สำคัญอย่างหนึ่งของการแบ่งกลุ่มเอกสารโดยอัตโนมัติด้วยคอมพิวเตอร์ก็คือการที่จะทำอย่างไรเพื่อให้คอมพิวเตอร์สามารถจะแยกแยะได้ว่าเอกสารแต่ละเอกสารมีความต่างกัน หรือเหมือนกันอย่างไร โดยที่ได้มีงานวิจัยจำนวนมากที่ได้นำเสนอวิธีการที่จะเป็นการวัด (Measure) ว่าเอกสารนั้นมีค่าความเหมือนกันกับเอกสารอื่นๆอย่างไร ซึ่งจะช่วยให้คอมพิวเตอร์สามารถแยกแยะความเหมือนหรือความต่างกันระหว่างแต่ละเอกสารได้ ในงานวิจัยนี้ได้เลือกเอาวิธีการวัดความแตกต่างกันของเอกสารที่มีอยู่ในงานวิจัยของ El-Sonbaty [7] ซึ่งได้แสดงวิธีการเปรียบเทียบความแตกต่างกันระหว่าง Symbolic Object ไว้ 2 ชนิดคือ หนึ่ง ชนิดข้อมูลที่มีคุณลักษณะเป็นแบบเชิงปริมาณ (Quantitative) และ สอง ชนิดข้อมูลที่มีคุณลักษณะเป็นแบบเชิงคุณภาพ (Qualitative) เนื่องจากในส่วนของงานวิจัยนี้ได้เลือกข้อมูลที่นำมาใช้ในการทดลองเป็นเอกสารข่าว ตามที่ได้กล่าวมาแล้วนั้นถือเป็นข้อมูลชนิดเชิงคุณภาพ ดังนั้นในงานวิจัยนี้จึงเลือกเฉพาะหลักการหาความแตกต่างกันของเอกสารซึ่งมีคุณลักษณะชนิดข้อมูลเป็นแบบเชิงคุณภาพ นำมาประยุกต์ใช้กับงานวิจัยนี้ดังที่จะกล่าวถึงต่อไป

2.6.1 หลักการหาความแตกต่างกันของเอกสารซึ่งมีคุณลักษณะชนิดเป็นแบบเชิงคุณภาพ

โดยที่ความแตกต่างระหว่างเอกสาร A และ B สามารถเขียนแทนด้วยสมการได้ดังนี้

$$D(A,B)=\sum_{k=1}^{d} D(A_k,B_k) \qquad (2.10)$$

เมื่อ

D = ความแตกต่างกันของเอกสาร

k = คุณลักษณะของเอกสาร

d = จำนวนทั้งหมดของคุณลักษณะที่ต้องการเปรียบเทียบ

โดยที่ค่า คุณลักษณะ ของแต่ละเอกสารจะมีองค์ประกอบสองส่วนคือ ความแตกต่างกันในเชิง Span ซึ่งเขียนแทนด้วย D_s และ ความแตกต่างกันในเชิง Content ซึ่งเขียนแทนด้วย D_c โดยที่แต่ละองค์ประกอบสามารถหาได้จากสมการที่ 2.11 และ 2.12 ตามลำดับ

$$D_s(A_k, B_k) = \frac{|\text{length of } A_k - \text{ length of } B_k|}{\text{span length of } A_k \text{ and } B_k} \quad (2.11)$$

$$D_c(A_k, B_k) = \frac{|\text{length of } A_k + \text{ length of } B_k - 2 \times \text{length of intersection of } A_k \text{and } B_k|}{\text{span length of } A_k \text{ and } B_k} \quad (2.12)$$

โดยที่ length ของ คุณลักษณะใดๆของเอกสาร คือเท่ากับจำนวนของสมาชิกทั้งหมดใน คุณลักษณะ นั้น และ ค่า pan length ของคุณลักษณะใดๆ ของสองเอกสารคือผลลัพธ์ที่ได้จากการนำจำนวนทั้งหมดของสมาชิกทั้งสองเอกสารมา Union กัน และผลลัพธ์สุดท้ายของค่าความแตกต่างสุทธิระหว่างเอกสาร A_k และ B_k คือผลรวมของทั้งความแตกต่างกันในเชิง Span และ ความแตกต่างกันในเชิง Content ตามสมการ 2.13

$$D(A_k, B_k) = D_s(A_k, B_k) + D_c(A_k, B_k) \quad (2.13)$$

2.6.2 ตัวอย่างการคำนวณหาค่าความแตกต่างระหว่างเอกสาร

สมมุติข้อมูลตัวอย่างของเอกสารข่าวประกอบด้วยรายละเอียดตามตารางที่ 2.3 จากตัวอย่างข้อมูลเอกสารตามตารางที่ 2.3 ค่าความแตกต่างระหว่างเอกสาร Doc1 และ Doc2 แสดงการคำนวณได้ตามสมการ 2.10 ดังนี้

$$D(Doc1, Doc2) = D(Doc1\_Title, Doc2\_Title) + D(Doc1\_keyword, Doc2\_keyword)$$

ตารางที่ 2.3 แสดงข้อมูลตัวอย่างของเอกสารข่าว

เอกสาร	Title	Keyword
Doc1	oil, stock	opec, barrel, dlr
Doc2	wheat, rice, stock	price, ton, dlr
Doc3	gas, stock	price, stock

พิจารณาความแตกต่างของเอกสารที่ คุณลักษณะ Title ในเชิง span ตามสมการที่ 2.11 คือ

$$D_s(Doc1\_Title, Doc2\_Title) = \frac{|\text{length of } Doc1\_Title - \text{length of } Doc2\_Title|}{\text{span length of } Doc1\_Title \text{ and } Doc2\_Title}$$

$$= \left|\frac{2-3}{4}\right| = 0.25$$

พิจารณาความแตกต่างของเอกสารที่ คุณลักษณะ Title ในเชิง content ตามสมการที่ 2.12 คือ

$$D_c(Doc1\_Title, Doc2\_Title) = \frac{|\text{length of } Doc1\_Title + \text{length of } Doc2\_Title - Z|}{\text{span length of } Doc1\_Title \text{ and } Doc2\_Title}$$

$Z = 2 * \text{length of intersection of } Doc1\_Title \text{ and } Doc2\_Title$

$$= \left|\frac{2+3-(2\times1)}{4}\right| = 0.75$$

ความแตกต่างของเอกสารที่ คุณลักษณะ Keyword ในเชิง span คือ

$$D_s(Doc1\_Keyword, Doc2\_Keyword) = \frac{|\text{length of } Doc1\_Keyword - \text{length of } Doc2\_Keyword|}{\text{span length of } Doc1\_Keyword \text{ and } Doc2\_Keyword}$$

$$= \left|\frac{3-3}{5}\right| = 0$$

ความแตกต่างของเอกสารที่ คุณลักษณะ Keyword ในเชิง content คือ

$$D_c(Doc1\_Keyword, Doc2\_Keyword) = \frac{|\text{length of } Doc1\_Keyword + \text{length of } Doc2\_Keyword - Y|}{\text{span length of } Doc1\_Keyword \text{ and } Doc2\_Keyword}$$

$Y = 2 * \text{length of intersection of } Doc1\_Keyword \text{ and } Doc2\_Keyword$

$$= \left|\frac{3+3-(2\times1)}{5}\right| = 0.8$$

จากสมการที่ 2.13 ความแตกต่างสุทธิระหว่าง Doc1 และ Doc2

$$= 0.25 + 0.75 + 0 + 0.8$$

$$=1.8$$

ต่อมาหาค่าความแตกต่างระหว่างเอกสาร Doc1 และ Doc3 แสดงการคำนวณได้ดังนี้

$$D(Doc1, Doc3) = D(Doc1\_Title, Doc3\_Title) + D(Doc1\_keyword, Doc3\_keyword)$$

พิจารณาความแตกต่างของเอกสารที่ คุณลักษณะ Title ในเชิง span คือ

$$D_s(Doc1\_Title, Doc3\_Title) = \frac{|\text{length of } Doc1\_Title - \text{length of } Doc3\_Title|}{\text{span length of } Doc1\_Title \text{ and } Doc3\_Title}$$

$$=\left|\frac{2-2}{3}\right| = 0$$

พิจารณาความแตกต่างของเอกสารที่ คุณลักษณะ Title ในเชิง content คือ

$$D_c(Doc1\_Title, Doc3\_Title) = \frac{|\text{length of } Doc1\_Title + \text{length of } Doc3\_Title - V|}{\text{span length of } Doc1\_Title \text{ and } Doc3\_Title}$$

$$V = 2 * \text{length of intersection of } Doc1\_Title \text{ and } Doc3\_Title$$

$$=\left|\frac{2+2-(2\times1)}{3}\right| = 0.67$$

ความแตกต่างของเอกสารที่ คุณลักษณะ Keyword ในเชิง span คือ

$$D_s(Doc1\_Keyword, Doc3\_Keyword) = \frac{|\text{length of } Doc1\_Keyword - \text{length of } Doc3\_Keyword|}{\text{span length of } Doc1\_Keyword \text{ and } Doc3\_Keyword}$$

$$=\left|\frac{3-2}{5}\right| = 0.2$$

ความแตกต่างของเอกสารที่ คุณลักษณะ Keyword ในเชิง content คือ

$$D_c(Doc1\_Keyword, Doc3\_Keyword) = \frac{|\text{length of } Doc1\_Keyword + \text{length of } Doc3\_Keyword - S|}{\text{span length of } Doc1\_Keyword \text{ and } Doc3\_Keyword}$$

$S = 2*\text{length of intersection of } Doc1\_Keyword \text{ and } Doc3\_Keyword$

$$= \left| \frac{3+2-(2\times 0)}{5} \right| = 1$$

จากสมการที่ 2.13 ความแตกต่างสุทธิระหว่าง Doc1 และ Doc3

$$= 0 + 0.67 + 0.2 + 1$$
$$= 1.87$$

จากตัวอย่างจะเห็นว่าค่าความต่างระหว่าง Doc1 และ Doc2 จะมีค่าน้อยกว่าค่าความต่างระหว่าง Doc1 และ Doc3 โดยที่หลักการวัดความแตกต่างระหว่างเอกสารที่กล่าวมานี้จะนำไปใช้ร่วมกับการทำงานของโครงข่ายประสาทเทียมแบบเรียนรู้เพื่อหาผู้ชนะ ซึ่งจะแสดงรายละเอียดในบทที่ 3

2.7 การวัดคุณภาพของการจัดกลุ่ม (Cluster Evaluation Criteria)

เพื่อที่จะสะท้อนถึงผลของการจัดกลุ่มว่ามีคุณภาพเพียงใด เราสามารถประเมินคุณภาพของอัลกอริทึมที่ใช้ในการจัดกลุ่มได้จากการวัดคุณภาพของข้อมูลในแต่ละคลัสเตอร์ การวัดคุณภาพจะขึ้นอยู่กับความรู้ที่เรามี เกี่ยวกับข้อมูลที่เราใช้ในการจัดกลุ่มว่าข้อมูลนั้นๆมีการจัดเป็นหมวดหมู่อย่างไร อาทิเช่นการที่เรารู้ว่าข่าวแต่ละข่าวนั้นถูกระบุไว้ว่ามันถูกจัดให้อยู่ในหัวข้อข่าวไหน ซึ่งเราสามารถนำผลของการจัดกลุ่มโดยอัลกอริทึมของเรามาเปรียบเทียบกันกับกลุ่มของข้อมูลที่ได้มีการจัดกลุ่มไว้แล้วว่าคุณภาพของการจัดกลุ่มข้อมูลของอัลกอริทึมเราเป็นอย่างไร ซึ่งเรียกการวัดแบบนี้ว่า การวัดคุณภาพแบบภายนอก (external quality measure) และ ถ้าข้อมูลที่เราใช้วัดไม่ได้ถูกจัดกลุ่มไว้ก่อนแล้วเราก็จะวัดโดยการเปรียบเทียบแต่ละคลัสเตอร์ที่แตกต่างกันนั้นโดยไม่มีการอ้างถึงความรู้จากภายนอกเราเรียกการวัดแบบนี้ว่า การวัดคุณภาพแบบภายใน (internal quality measure) ซึ่งข้อมูลที่ใช้ในงานวิจัยนี้เป็นข้อมูลที่มีผู้รวบรวมและได้แบ่งแยกหมวดหมู่ของข้อมูลไว้แล้วดังนั้นเราจึงเลือกใช้วิธีการวัดคุณภาพของการจัดกลุ่มแบบการวัดคุณภาพแบบภายนอก (external quality measure) ซึ่งเลือกเอาวิธีการวัดแบบนี้มาใช้ด้วยกัน สองชนิดคือ Entropy และ F-Measure [2]

2.7.1 Entropy

ค่า Entropy เป็นตัวชี้วัดการวัดคุณภาพแบบภายนอก ที่ใช้บอกว่าแต่ละคลัสเตอร์มีสมาชิกที่

ประกอบไปด้วยข้อมูลจากคลาสเดียวกันมากน้อยแค่ไหน เช่นใช้ในการชี้วัดคัลสเตอร์ที่แต่ละระดับของ hierarchical clustering เป็นต้น ซึ่งค่า Entropy จะบอกเราว่าสมาชิกในคลัสเตอร์มีลักษณะเหมือนกัน (homogeneity) มากน้อยเพียงใดโดยที่ถ้าค่า Entropy ที่คำนวณได้มีค่าใกล้ศูนย์แสดงว่าสมาชิกในคลัสเตอร์นั้นมีความเหมือนกันสูง โดยที่ถ้าค่า Entropy มีค่าเท่ากับศูนย์แสดงว่าคลัสเตอร์นั้นประกอบไปด้วยข้อมูลที่ไม่มีลักษณะต่างกันเลยโดยขั้นตอนการคำนวณค่า Entropy เริ่มจากกำหนดให้ P คือผลที่ได้จากอัลกอริทึมจัดกลุ่ม ซึ่งมี m คลัสเตอร์ ที่ทุกคลัสเตอร์ j ใน P เราจะคำนวณหาค่า p_{ij} ซึ่งคือค่าความเป็นที่สมาชิกคลัสเตอร์ j จะอยู่ใน คลาส i โดยที่ค่า Entropy ของแต่ละคลัสเตอร์ j คำนวณได้จาก

$$E_j = -\sum_i p_{ij} \log(p_{ij}) \tag{2.14}$$

ค่าผลรวม Entropy สุทธิสำหรับสำหรับชุดของคลัสเตอร์คือ

$$E_{cs} = \sum_{j=1}^{m} \frac{N_j}{N} \times E_j \tag{2.15}$$

เมื่อ

N_j คือขนาดของคลัสเตอร์ j

N คือ จำนวนของข้อมูลทั้งหมด

m คือจำนวนคลัสเตอร์

2.7.2 F-Measure

ค่า F-Measure เป็นตัวชี้วัด การวัดคุณภาพแบบภายนอก ตัวที่สองที่จะกล่าวถึง ค่า F-Measure เป็นค่าที่เกิดจากการรวมกันของค่า precision และ ค่า recall ซึ่งสองค่านี้เป็นแนวคิดจากงานวิจัย information retrieval โดยที่ค่า precision และค่า recall ของคลัสเตอร์ j กับ คลาส i แสดงได้ดังนี้

$$P = \text{Precision}(i,j) = \frac{N_{ij}}{N_j} \tag{2.16}$$

$$R = \text{Recall}(i,j) = \frac{N_{ij}}{N_i} \tag{2.17}$$

เมื่อ

N_{ij} คือจำนวนสมาชิกใน คลาส i ใน คลัสเตอร์ j

N_j คือจำนวนสมาชิกของ คลัสเตอร์ j

N_i คือจำนวนสมาชิกของ คลาส i

ค่า F-Measure ของ คลัสเตอร์ j และ คลาส i หาได้จาก

$$F(i,j) = \frac{2 * P(i,j) * R(i,j)}{P(i,j) + R(i,j)} \tag{2.18}$$

ค่า F-Measure สำหรับแต่ละ คลาส i คำนวณได้จาก

$$F = \sum_{i} \frac{N_i}{N} \text{Max}\{F(i,j)\} \tag{2.19}$$

เมื่อ

N คือ จำนวนของเอกสารทั้งหมด

Max คือ ค่าสูงสุด (Maximum)

ถ้าค่า F-Measure ที่ได้มีค่าสูง (ค่าอยู่ระหว่าง 0-1) แสดงว่าข้อมูลในคัสเตอร์ที่ได้มีคุณภาพดี

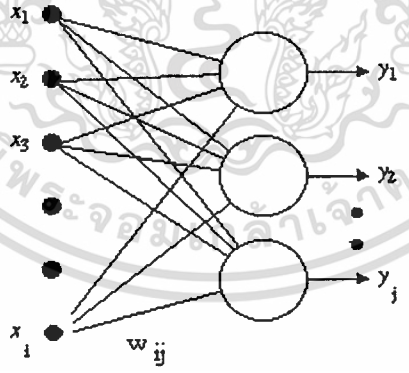
บทที่ 3

การประมวลผลข้อความด้วยโครงข่ายประสาทเทียม ที่เรียนรู้แบบหาผู้ชนะ

(A Text Processing Competitive Learning Neural Network (TPCLNN))

3.1 สถาปัตยกรรมของ TPCLNN

TPCLNN เป็นสถาปัตยกรรมโครงข่ายประสาทเทียมแบบ Competitive Learning neural network Algorithm ที่ทำงานร่วมกับการหาความต่างของเอกสาร สถาปัตยกรรมของ TPCLNN ประกอบด้วยส่วนเลเยร์ของข้อมูล 1 เลเยร์เหมือนกับโครงสร้างของ Competitive Learning neural network ทั่วไป ที่โหนดอินพุทแต่ละโหนดจะมี weight ที่เชื่อมต่อไปยังโหนดเอ้าท์พุททุกๆตัว (fully connection) โดยที่โหนดของอินพุทจะถูกกำหนดตามจำนวน "คุณลักษณะ" ของเอกสาร (ข้อมูลเข้า) ส่วนโหนดเอ้าท์พุท จำนวนของโหนดเอ้าท์พุทจะไม่ได้กำหนดจำนวนคงที่แน่นอนไว้ ซึ่งจำนวนโหนดเอ้าท์พุทจะมีจำนวนเท่าไรขึ้นอยู่กับการออกแบบและปรับเปลี่ยนได้ในขั้นตอนการทดลอง สถาปัตยกรรมของ TPCLNN แสดงได้ตามรูปที่ 3.1 โดย x_i คือข้อมูลอินพุท ส่วน y_j คือ เอ้าท์พุท และ w_{ij} คือ weight ที่เชื่อมต่อระหว่างเอ้าท์พุท j กับ อินพุท i



รูปที่ 3.1 แสดงสถาปัตยกรรมของ TPCLNN

ซึ่งความแตกต่างระหว่าง TPCLNN กับโครงข่ายประสาทเทียมทั่วๆไป คือที่โหนดอินพุท TPCLNN จะรับค่าเป็นข้อมูลเชิงคุณภาพ (qualitative value) โดยค่า weigh ที่ต่อจากโหนดอินพุท

'i' กับโหนดเอ้าท์พุท 'j' คือ w_{ij} ซึ่งนิยามได้ดังนี้

$$w_{ij} = \left\{ \left(A_{1ij}, e_{1ij} \right), \left(A_{2ij}, e_{2ij} \right), \left(A_{3ij}, e_{3ij} \right), \ldots, \left(A_{pij}, e_{pij} \right) \right\} \quad (3.1)$$

เมื่อ A_{pij} คือข้อมูลเชิงคุณภาพ (ในที่นี่คือ text) p^{th} ของค่า weight และ 'e_{pij}' คือค่า degree ของความสัมพันธ์ของค่าเชิงคุณภาพกับค่าข้อมูลอินพุท 'i' ซึ่งค่า e_{pij} มีค่าระหว่าง 0 ถึง 1 ถ้าค่า e_{pij} = 0 แสดงว่าข้อมูล A_{pij} ไม่เป็นส่วนหนึ่งของอินพุท 'i' แต่ถ้าค่า e_{pij} = 1 แสดงว่าข้อมูล A_{pij} มีความสัมพันธ์สูงที่สุดกับข้อมูลที่เข้ามาที่โหนดอินพุท 'i'

3.2 Learning algorithm

อัลกอริทึมของเราทำงานบนพื้นฐานของการเรียนรู้แบบไม่มีผู้สอน(unsupervised learning) โดยใช้การเรียนรู้แบบที่เรียกว่า winner-take-all ร่วมกับหลักการวัดความต่างของเอกสารดังที่ได้กล่าวมาแล้วในบทที่2 ตามวิธีการของการเรียนรู้แบบ winner-take-all ตัวนิวรอล ที่จะได้รับการปรับ weight จะต้องเป็นนิวรอลที่ได้รับการประกาศว่าเป็นผู้ชนะ ซึ่งในการแข่งขันแต่ละครั้งจะมีเพียงนิวรอลเดียวเท่านั้นที่จะเป็นผู้ชนะและได้ปรับ weight ในการหานิวรอลที่ชนะ อัลกอริทึมของเราใช้หลักการหาความต่างของเอกสารโดยที่ การคำนวณค่าความต่างระหว่างอินพุทแต่ละตัวเทียบกับค่า weight ทุกตัวที่เชื่อมต่ออยู่ระหว่างโหนดเอ้าท์พุทและโหนดอินพุทตัวนั้นๆ ซึ่งค่าที่โหนดเอ้าท์พุทตัวใดมีค่าต่ำที่สุดก็จะเป็นผู้ชนะ โดยรายละเอียดการทำงานของอัลกอริทึมมีขั้นตอนดังนี้

Step 0: initialize weights w_{ij}. Each weight can be initialized from the training data set arbitrarily.

Step 1: While stopping condition is false, do step 2-6

Step 2: For each input vector

$X = \left(x_1, x_2, \ldots, x_d \right)^T$, do step 3-6

Step 3: For each output unit 'j', compute

$$\left\| X - W_j \right\| = \sum_{k=1}^{d} \sum_{n=1}^{p} D\left(x_k, A_{pkj} \right)$$

p = Number of values of w_{ki}

d = Number of inputs

Step 4: Find index J such that $\|x-w_j\|$ is a minimum

Step 5: For all weights that connect to the wining node J ,i.e.,

$$w_{iJ} = \left\{ \begin{matrix} (A_{1iJ}, e_{1iJ}), \\ (A_{2iJ}, e_{2iJ}), \\ (A_{3iJ}, e_{3iJ}), \ldots, \\ (A_{piJ}, e_{piJ}) \end{matrix} \right\} \text{ for i = 1, 2, 3, ..., k}$$

$$w_{iJ}^{(new)} = w_{iJ}^{(old)} \cup x_i$$

Step 5.1: Update degree values of weights

If $w_{iJ}^{(new)} \in w_{iJ}^{(old)}$ *Then*

$$e_{piJ}^{(new)} = \begin{cases} f\left(e_{piJ}^{(old)} + \eta\left(1-D\right)\right) \text{ if } A_{piJ} \in w_{iJ} \cap x_i \\ f\left(e_{piJ}^{(old)} - \eta\left(1-D\right)\right) \text{ if } A_{piJ} \notin w_{iJ} \cap x_i \end{cases}$$

Step 5.2: Update degree values of weights

If $w_{iJ}^{(new)} \notin w_{iJ}^{(old)}$ *Then* $e_{piJ}^{(new)} = I$

Where f(.) is defined as below:

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

η = learning rate constant

D = the net dissimilarity value of index J

I = Initial value has value 0 to 1

Step 6: Test stopping condition. Like classical competitive neural networks, there are several strategies for stopping condition such as train data set until data in each one node not change.

3.2.1 ตัวอย่างการปรับ weight ของ TCPLNN

เพื่อให้เข้าใจขั้นตอนการเรียนรู้ของ TCPLNN จะขอยกตัวอย่างมาอธิบายดังนี้ กำหนดให้ learning rate (η) = 0.1 กำหนดค่า degree เริ่มต้นของแต่ละสมาชิกใน weight = 0.3 โดยมีโหนดเอ้าท์พุทสองโหนด และมีค่า weight ของแต่ละโหนดดังนี้

weight1 = [(A, 0.3), (B, 0.3), (C, 0.3)]

weight2 = [(D, 0.3), (E, 0.3), (F, 0.3)]

เมื่ออินพุทชุดแรกที่เข้ามา = [(A), (B)]

ดังนั้น weight1 คือ weight ของโหนดที่ถูกประกาศให้ชนะ (มีค่าความต่างระหว่างตัวมันกับชุดอินพุทน้อยที่สุดวิธีการคำนวณอยู่ในบทที่ 2) ซึ่ง Weight1 จะได้รับการปรับกลายเป็น weight ใหม่ดังนี้

weight1.1 = [(A, 0.4), (B, 0.4), (C, 0.2)]

โดยที่ weight2 ยังคงมีค่าเหมือนเดิม

เมื่ออินพุทชุดที่เข้าชุดต่อมา = [(A)]

weight ของโหนดที่ชนะ คือ weight1.1 ดังนั้น weight1.1 จะได้รับการปรับกลายเป็น weight ใหม่อีกครั้งดังนี้

weight1.2 = [(A, 0.5), (B, 0.3), (C, 0.1)]

โดยที่ weight2 ก็ยังคงมีค่าเหมือนเดิม

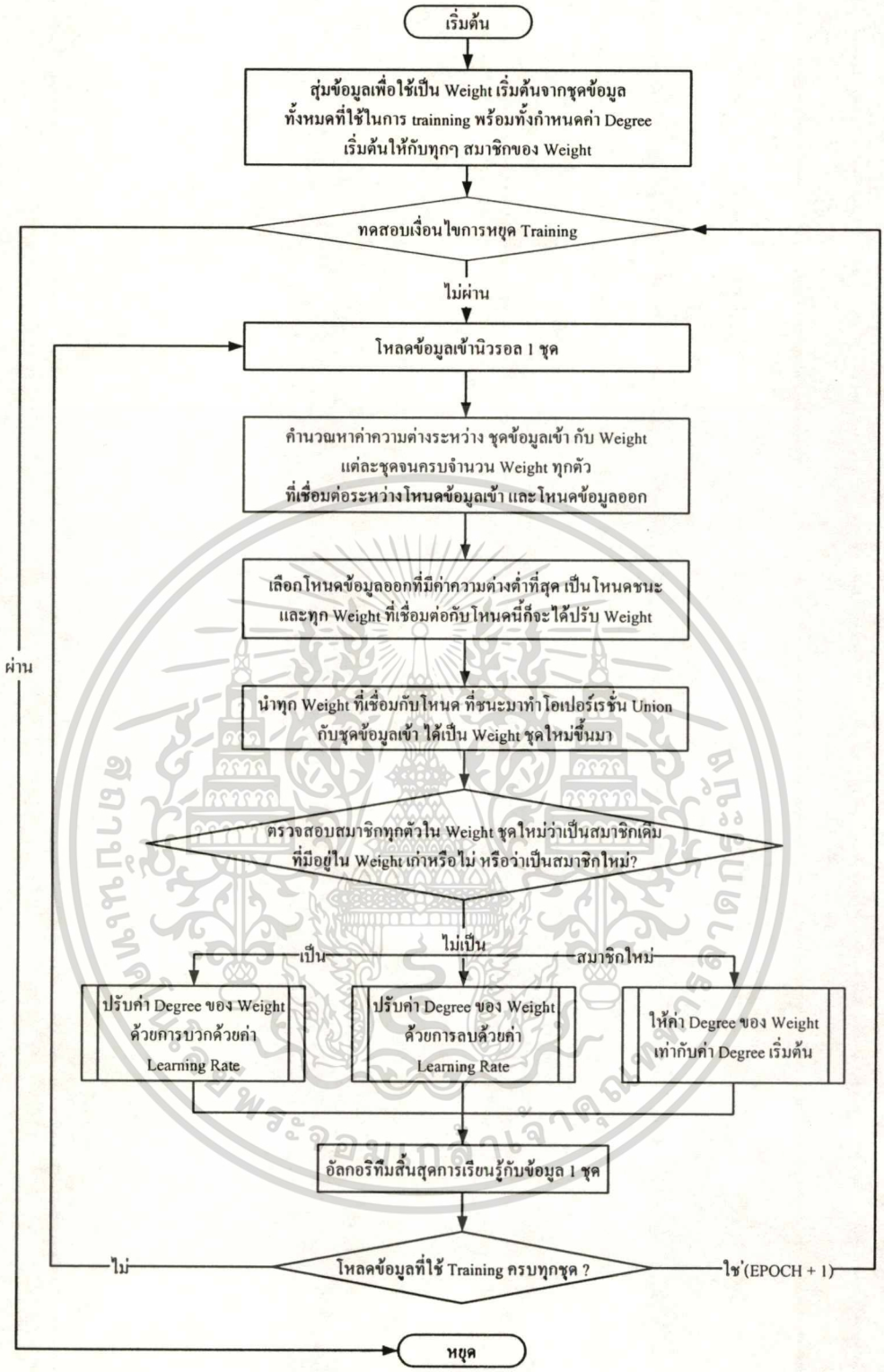
ต่อมาอินพุทเข้ามา = [(D), (F)]

weight ของโหนดที่ชนะ คือ weight2 ดังนั้น weight2 จะได้รับการปรับกลายเป็น weight ใหม่ดังนี้

weight2.1 = [(D, 0.4), (E, 0.2), (F, 0.4)]

เมื่อเราหยุดเทรนนิ่งที่ตรงนี้ ค่า weight สุดท้ายที่ได้และจะถูกนำไปใช้คือ weight1.2 และ weight 2.1

จากตัวอย่างข้างต้นจะเห็นว่า ถ้าโหนดที่ชนะนั้นมีอินพุท ที่ตรงกับค่า weight บ่อยๆก็จะทำให้ค่า degree ของค่า weight ตัวนั้นมีค่าสูงขึ้นไปเรื่อยๆ ในทำนองกลับกันถ้าค่าใน weight ตัวใดไม่ตรงกับชุดอินพุท ตัวมันก็จะถูกลดค่า degree ลงเรื่อยๆจนเมื่อมีค่าเท่ากับ 0 แล้วจะก็ถูกตัดออกจาก weight ในที่สุด ซึ่งผลที่ได้จะทำให้ weight แต่ละตัวชุดก็คือตัวแทนของข้อมูลแต่ละกลุ่มนั่นเอง



รูปที่ 3.2 แสดง Flow Chart การทำงานของอัลกอริทึม TPCLNN

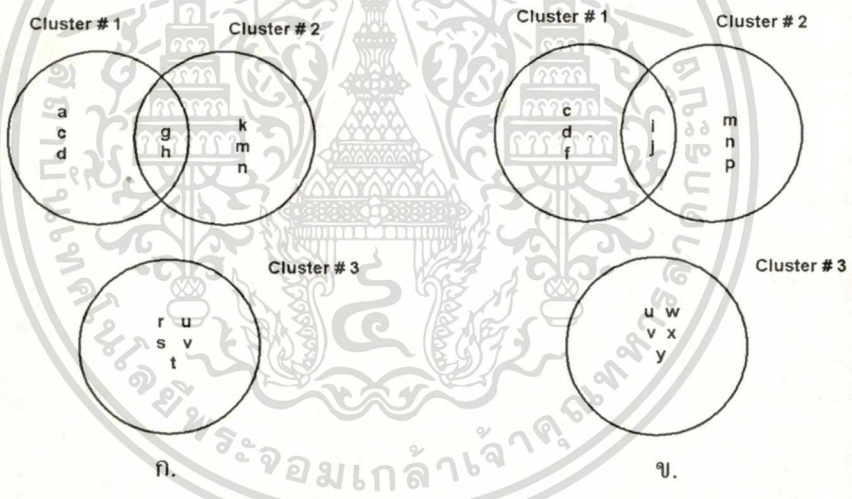
บทที่4
วิธีการดำเนินการวิจัย

4.1 การเตรียมชุดข้อมูล

ข้อมูลที่ใช้ในการวิจัยจะมี 2 ส่วน ส่วนแรกเป็นข้อมูลตัวอักษรที่ได้จากการสุ่มจากตัวอักษรตามโครงสร้าง (profile) ที่กำหนดขึ้นมาเอง การสุ่มข้อมูลขึ้นมาเองนี้ก็เพื่อใช้ในการทดสอบการประมวลผลของตัวโมเดล TPCLNN ก่อนการนำไปใช้กับข้อมูลจริง และส่วนที่สองจะเป็นข้อมูลข่าว Reuters – 21578 ที่เป็นเอกสารข่าวจริงๆที่ได้จากแหล่งข้อมูลบนเวบไซต์ [8] การทดลองกับข้อมูลข่าว ก็เพื่อเป็นการพิจารณาถึงประสิทธิภาพของตัวโมเดลเมื่อนำไปใช้งานกับข้อมูลจริงโดยที่ข้อมูลแต่ชนิดมีรายละเอียดดังนี้

4.1.1 ข้อมูลตัวอักษร

เป็นข้อมูลที่ได้จากการสุ่มของตัวอักษรจำนวน 3 กลุ่มที่มีโครงสร้างดังรูปที่ 4.1



รูปที่ 4.1 แสดงโครงสร้างของข้อมูลตัวอักษรที่ใช้ในการทดลอง ก.ชุดข้อมูล Title ข. ชุดข้อมูล Keyword

ซึ่งจำนวนของข้อมูลที่สุ่มขึ้นมาสำหรับการทดลองแสดงรายละเอียดในตารางที่ 4.1 และในตารางที่ 4.2 จะเป็นรายละเอียดของข้อมูลที่จะใช้ในการทดสอบโมเดล TPCLNN หลังจากผ่านการเทรนนิ่งมาแล้วส่วนตารางที่ 4.3 เป็นตัวอย่างบางส่วนของข้อมูลตัวอักษรที่สุ่มขึ้นมา

ตารางที่ 4.1 แสดงชุดข้อมูลตัวอักษรที่ใช้เทรนนิ่ง

ข้อมูล	จำนวน (ชุด)	คลัสเตอร์ 1	คลัสเตอร์ 2	คลัสเตอร์ 3
Data Train	300	100	100	100

ตารางที่ 4.2 แสดงชุดข้อมูลตัวอักษรที่ใช้ในการทดสอบ

ข้อมูล	จำนวน (ชุด)	คลัสเตอร์ 1	คลัสเตอร์ 2	คลัสเตอร์ 3
Data Test	1000	333	320	347

ตารางที่ 4.3 แสดงตัวอย่างของข้อมูลตัวอักษรที่ใช้ในการทดลอง

ลำดับที่	TITLE	KEYWORD
1	a,g,d	c,d,i
2	h,n	j,m
3	r,s,t	u,v,y

4.1.2 ข้อมูลข่าว Reuters – 21578

ข้อมูลข่าว Reuters – 21578 เป็นข้อมูลข่าวของสำนักข่าว Reuters โดยข้อมูลชนิดนี้ได้มีผู้รวบรวมและได้จัดแยกหมวดหมู่ของข่าวไว้แล้ว ซึ่งประกอบด้วยข่าวจำนวน 21578 ข่าว จากจำนวน 135 กลุ่มข่าวโดยข่าวแต่ละข่าวจะมีโครงสร้างข้อมูลเป็นไฟล์ sgml ตัวอย่างข่าวแสดงในภาคผนวก ข.

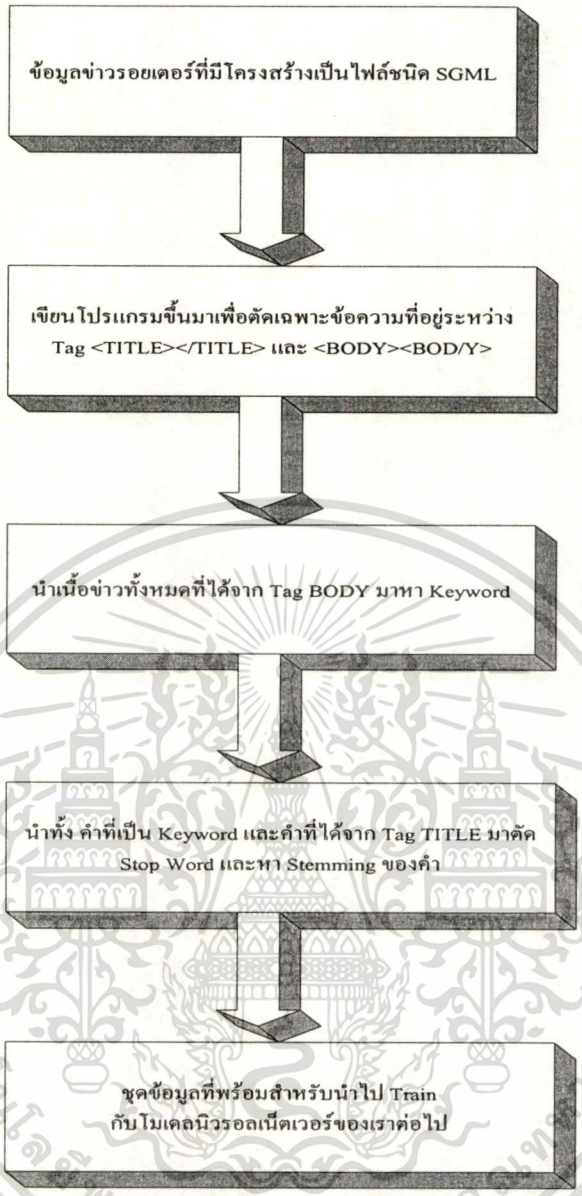
จากข้อมูลข่าวที่มีเราจะนำข่าวทั้งหมดมาผ่านขั้นตอนการเตรียมข้อมูลก่อนนำไปใช้งาน ดังนี้

ขั้นที่ 1 แยกเอาเฉพาะข้อความใน Topic และใน Body ของทุกๆข่าว และทำ Index ของแต่ละข่าวว่าอยู่ใน Topic ไหน

ขั้นที่ 2 นำตัวเนื้อข่าวที่ได้จากแท็ก Bodyทั้งหมดมาหาคำสำคัญ (keyword) ด้วยโปรแกรม copernic summarizer โดยในการหาคำสำคัญของแต่ละข่าวได้กำหนดจำนวนของคำที่ซ้ำไว้ที่ 10 คำ

ขั้นที่ 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 Apet [8] ที่มีรายละเอียดอยู่ในไฟล์ที่มาพร้อมกับไฟล์ข่าว Reuters-21578 ซึ่งแสดงรายละเอียดข้อมูลที่จะใช้ เทรนนิ่ง และ ทดสอบ ในตารางที่ 4.4-4.6



รูปที่ 4.2 แสดงแผนผังขั้นตอนการเตรียมชุดข้อมูล

ตารางที่ 4.4 แสดงชุดข้อมูลสำหรับใช้เทรนนิ่งและใช้ทดสอบของข้อมูลข่าว Reuters 3 กลุ่ม

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

ตารางที่ 4.5 แสดงชุดข้อมูลสำหรับใช้เทรนนิ่งและใช้ทดสอบของข้อมูลข่าว Reuters 5 กลุ่ม

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

ตารางที่ 4.6 แสดงชุดข้อมูลสำหรับใช้เทรนนิ่งและใช้ทดสอบของข้อมูลข่าว Reuters 14 กลุ่ม

กลุ่มข่าว	จำนวนข่าวที่ใช้เทรน	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

4.2 สภาวะแวดล้อมของการวิจัย

เครื่องคอมพิวเตอร์ที่ใช้ในการทดลองประกอบด้วย CPU Pentium 4 1.6 GHz หน่วยความจำขนาด 256 MB ฮาร์ดดิสก์ 30 GB รันบนระบบปฏิบัติการ Microsoft Windows XP

ใช้ปรแกรม Matlab เวอร์ชั่น 6.1 เป็นโปรแกรมที่ใช้พัฒนาอัลกอริทึม ใช้โปรแกรม Copernic Summarize 2.0 สำหรับหาคำสำคัญของเนื้อข่าว และ ใช้โปรแกรม UltraEdit-32 เวอร์ชั่น 9.20a สำหรับเขียนสคริปภาษา Perl สำหรับการเตรียมข้อมูล

4.3 ผลการทดลอง

4.3.1 ชุดข้อมูลตัวอักษร

รายละเอียดของชุดข้อมูลตัวอักษรที่ใช้ในการทดลองแสดงในตารางที่ 4.1 โดยในการเทรนนิ่งตัวโครงข่ายผลลัพธ์ที่ได้จากการเทรนนิ่งแสดงในตารางที่ 4.7

ตารางที่ 4.7 แสดงผลลัพธ์ที่ได้จากการเทรนนิ่งของข้อมูลตัวอักษร 3 กลุ่ม

โหนดเอ้าท์พุท	ค่า weight ที่ได้		แสดงความเป็นตัวแทนของคลัสเตอร์
	Title	Keyword	
1	s,v,r	v,y,w,x	3
2	d,g,h	d,j	1
3	h,m,n	j,m,p	2

Learning Rate (η) = 0.001 Initial Learning Rate = 0.001 Epoch = 100 โหนดเอ้าท์พุท = 3

หลักจากเสร็จสินการเทรนนิ่งแล้วค่า weight สุดท้ายที่ได้ก็จะเป็นตัวแทนของชุดข้อมูล ต่อจากนั้นจะทำการทดสอบความถูกต้องของการแบ่งกลุ่มของโมเดล โดยรายละเอียดของชุดข้อมูลตัวอักษรที่ใช้ทดสอบตามตารางที่ 4.2 ผลที่ได้จากการทดสอบความถูกต้องของตัวโมเดล แสดงในตารางที่ 4.8

ตารางที่ 4.8 แสดงผลลัพธ์ที่ได้จากการทดสอบของข้อมูลอักษร 3 กลุ่ม

กลุ่มข้อมูล	จำนวนสมาชิกที่ตกในแต่ละคลัสเตอร์		
	1	2	3
1	0	**324**	23
2	0	17	**316**
3	**319**	1	0
ค่า F measure = 0.96			
ค่า Entropy = 0.16			

Learning Rate (η) = 0.01, Initial Learning Rate = 0.3 Epoch = 100 โหนดเอ้าท์พุท = 3

4.3.2 ชุดข้อมูลข่าว Reuters-21578

รายละเอียดของชุดข้อมูล Reuters-21578 ที่ใช้ในการทดลองอยู่ในตารางที่ 4.4 – 4.6 โดยผลที่ได้จากการทดสอบความถูกต้องของตัวโมเดล แสดงในตารางที่ 4.9 - 4.11

ตารางที่ 4.9 แสดงผลลัพธ์ของการทดสอบของข้อมูลข่าว Reuters จำนวน 3 กลุ่ม

โหนดเอ้าท์พุทตัวที่ (เป็นตัวแทนของกลุ่ม)	จำนวนสมาชิกที่ตกในคลัสเตอร์		
	1	2	3
1 (acq)	**1822**	81	29
2 (crude)	140	**344**	12
3 (grain)	29	35	**403**
ค่า F measure = 0.88			
ค่า Entropy = 0.39			

Learning Rate (η) = 0.01 Initial Learning Rate = 0.3 Epoch = 379 โหนดเอ้าท์พุท = 3

ตารางที่ 4.10 แสดงผลลัพธ์ของการทดสอบของข้อมูลข่าว Reuters จำนวน 5 กลุ่ม

โหนดเอ้าท์พุทตัวที่ (เป็นตัวแทนของกลุ่ม)	จำนวนสมาชิกที่ตกในคลัสเตอร์				
	1	2	3	4	5
1 (money-fx)	158	137	119	**1466**	52
2 (crude)	49	16	**378**	52	1
3 (grain)	30	**418**	12	6	1
4 (acq)	**569**	8	8	10	1
5 (earn)	116	40	55	185	**2785**
ค่า F measure = 0.84					
ค่า Entropy = 0.5					

Learning Rate (η) = 0.01 Initial Learning Rate = 0.3 Epoch = 256 โหนดเอ้าท์พุท = 5

ตารางที่ 4.11 แสดงผลลัพธ์ของการทดสอบของข้อมูลข่าว Reuters จำนวน 14 กลุ่ม

โหนดเอ้าท์พุทตัวที่ (เป็นตัวแทนของกลุ่ม)	จำนวนสมาชิกที่ตกในคลัสเตอร์													
	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1(trade, interest)	0	16	29	**646**	2	12	3	1	5	0	6	0	6	1
2 (acq)	26	**225**	1	4	23	11	4	4	117	0	15	58	4	0
3 (-)	24	6	0	0	**34**	1	17	0	4	0	3	1	0	3
4 (earn)	10	1	0	0	3	6	3	20	0	**22**	1	0	0	7
5 (money-fx)	11	7	1	1	4	**87**	0	19	0	0	3	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	16	0	**30**	1	2	1	0	0	1
9 (-)	2	0	0	0	1	2	0	**13**	0	5	1	0	0	3
10 (grain, oilseed)	5	0	0	0	1	1	3	7	0	5	1	0	0	1
11 (-)	0	2	0	0	**13**	0	3	0	0	0	0	0	0	0
12 (-)	1	0	1	0	4	1	0	0	0	0	0	0	0	**6**
13 (-)	2	0	0	0	2	3	1	4	0	**6**	1	0	0	5
14 (money-supply, sugar)	4	0	0	0	0	1	1	1	0	2	0	0	0	**12**
ค่า F measure = 0.65														
ค่า Entropy = 0.8														

Learning Rate (η) = 0.01, Initial Learning Rate = 0.3, Epoch = 340 โหนดเอ้าท์พุท = 14

4.4 สรุปผลการทดลอง

จากผลการทดลองตารางที่ 4.8 เป็นผลการทดลองกับชุดข้อมูลตัวอักษรผลที่ได้คือค่า F-measure เท่ากับ 0.96 และ ค่า Entropy เท่ากับ 0.16 ซึ่งเป็นค่าที่แสดงว่าประสิทธิภาพของการตัวอัลกอริทึมนั้น สามารถจัดกลุ่มกับข้อมูลที่เรากำหนดนั้นได้ดีมากมีความถูกต้องถึงประมาณ 96 เปอร์เซ็นต์ สำหรับในส่วนของชุดข้อมูลข่าว Reuters-21578 จากตารางที่ 4.9 เป็นผลการทดลองของข้อมูล 3 กลุ่มซึ่งผลที่ได้ ค่า F-measure เท่ากับ 0.88 และ ค่า Entropy เท่ากับ 0.39 ซึ่งประสิทธิภาพของการจัดกลุ่มกับข้อมูลอยู่ในเกณฑ์ที่ดีมีความถูกต้องประมาณ 88 เปอร์เซ็นต์และสามารถระบุโหนดเอ้าท์พุทที่เป็นตัวแทนของแต่ละกลุ่มได้อย่างชัดเจน สำหรับข้อมูล 5 กลุ่มตาม

ตารางที่ 4.10 นั้นที่ได้ ค่า F-measure เท่ากับ 0.84 และ ค่า Entropy เท่ากับ 0.5 ซึ่งประสิทธิภาพของการจัดกลุ่มกับข้อมูลยังถือว่าอยู่ในเกณฑ์ที่ดีมีความถูกต้องประมาณ 84 เปอร์เซ็นต์ และสามารถระบุโหนดเอ้าท์พุทที่เป็นตัวแทนของแต่ละกลุ่มได้อย่างชัดเจนเช่นเดียวกับข้อมูล 3 กลุ่ม และสุดท้ายสำหรับข้อมูล 12 กลุ่ม ซึ่งเป็นชุดข้อมูลขนาดใหญ่ผลที่ได้จากตารางที่ 4.11 ได้ ค่า F-measure เท่ากับ 0.65 และ ค่า Entropy เท่ากับ 0.8 ประสิทธิภาพของการจัดกลุ่มกับข้อมูลอยู่ในเกณฑ์ที่พอใช้ และจากผลการทดลองของข้อมูล 14 กลุ่ม นี้ยังพบอีกว่าบางโหนดเอ้าท์พุทไม่สามารถระบุได้ว่าเป็นตัวแทนของกลุ่มข้อมูลกลุ่มใด เนื่องจากข้อมูลที่ใช้ในการทดลองมีการทับซ้อนกันมาก ซึ่งจะเห็นได้ค่า Entropy ที่สูงถึง 0.8

ข้อแนะนำเพิ่มเติมเพื่อเป็นแนวทางสำหรับสำหรับการกำหนดค่า Learning Rate ให้กับตัวโครงข่าย TPCLNN สำหรับการเทรนกับชุดข้อมูลอื่นๆ โดยเราจะพิจารณาดังนี้

ในแต่ละรอบ (epoch) ของการเทรนเราต้องการให้ชุดของข้อมูลที่มีความถี่ของการเกิดขึ้นบ่อยๆ ยังคงอยู่เป็นตัวแทนของ weight ตลอดจนจบในแต่ละรอบของการเทรน

ถ้าต้องการให้ข้อมูลแต่ละตัวยังคงเป็นตัวแทนของ weight เป็นจำนวน B เปอร์เซ็นต์ของรอบหลังจากที่ถูกเจอครั้งแรก จะได้จำนวนครั้งสูงสุดที่ข้อมูลตัวนั้นๆ จะไม่ถูกเจอติดๆกัน (α) โดยที่ α จะคำนวณได้จาก

$$\alpha = \frac{A \times B}{100}$$

A = จำนวนของข้อมูลทั้งหมดที่ใช้เทรน

B = จำนวนเปอร์เซ็นต์ของรอบที่ข้อมูลจะอยู่ใน weight หลังจากที่ถูกเจอครั้งแรก

ซึ่งหมายความว่า ถ้าข้อมูลตัวไหนที่ไม่ถูกเจอในระหว่างการเทรนในแต่ละรอบเป็นจำนวน α ครั้งติดต่อกัน (หลังจากที่ถูกเจอในครั้งแรกแล้ว) ข้อมูลตัวนั้นก็จะหายไปจาก weight (เนื่องจากถูกลดค่า degree จนเท่ากับ 0) ดังนั้นค่า Learning Rate ที่ใช้ในการเทรนจะคำนวณได้จาก

$$\eta = \frac{\text{Initial Learning Rate}}{\alpha}$$

บทที่ 5
สรุปผลงานวิจัยและข้อเสนอแนะ

5.1 สรุปผลงานวิจัย

งานวิจัยนี้ ได้นำเสนอวิธีการจัดกลุ่มเอกสาร โดยใช้หลักการของโครงข่ายประสาทเทียมที่เรียนรู้แบบหาผู้ชนะ ร่วมกับการหาความแตกต่างกันของเอกสาร จากหลักการทั้งหมดที่กล่าวมาทำให้โครงข่ายประสาทเทียมที่พัฒนาขึ้นมานั้นสามารถรับข้อมูลอินพุทที่เป็นข้อความ (text) เข้าไปประมวลผลได้โดยตรง โดยที่ไม่จำเป็นต้องแปลงอินพุท ให้เป็นข้อมูลในเชิงตัวเลขก่อนแต่อย่างใด ซึ่งวิธีการนี้ช่วยลดปัญหาของการเกิด High Dimension ของคำเมื่อใช้วิธีการของ Vector Space Model ในการแบ่งกลุ่มที่มีข้อมูลขนาดใหญ่

โดยสามารถสรุปหลักการทำงานและการเรียนรู้ (Learning Rule) ของอัลกอริทึมได้ดังนี้

5.1.1 การหาโหนดชนะ (Wining Node)

ในงานวิจัยนี้ได้ใช้หลักการหาความแตกต่างกันของข้อมูลแบบ Symbolic object ในส่วนของ object ที่เป็นชนิดเชิงคุณภาพ (qualitative) มาประยุกต์ใช้ในการหาโหนดชนะซึ่งการเปรียบเทียบความแตกต่างกันของเอกสารที่ใช้ในงานวิจัยนี้นั้นเกิดจากการนำอินพุทหนึ่งชุดมาเปรียบเทียบกับค่าของ weight ทุกๆตัวทีละตัวจนครบกับจำนวน weight ที่เชื่อมต่ออยู่ทั้งหมด โดยที่ค่าความแตกต่างสุทธิระหว่างอินพุทกับค่า weight ที่เชื่อมต่ออยู่กับโหนดเอ้าท์พุทตัวใดมีค่าต่ำที่สุด โหนดเอ้าท์พุทที่เชื่อมต่ออยู่กับค่า weight นั้นก็จะได้รับการประกาศให้เป็นผู้ชนะ (wining-take-all) และค่า weight ทั้งหมดที่เชื่อมต่ออยู่กับตัวโหนดนั้นก็จะได้รับการปรับ weight ต่อไป

5.1.2 การปรับ Weight

การปรับ Weight ในโครงข่ายประสาทเทียมก็คือการที่ตัวโมเดลพยายามที่จะเรียนรู้รูปแบบของข้อมูลที่เข้ามานั้นเอง ซึ่ง weight ในโมเดลของงานวิจัยนี้ประกอบด้วยสองส่วนคือ ส่วนที่เป็นข้อมูลเชิงคุณภาพ (text) และส่วนที่เป็น Degree ของ text ซึ่งการปรับ weight เราจะนำ text แต่ละคำของอินพุทขณะนั้นมาทำการเปรียบเทียบกับสมาชิกทุกตัวของ weight ขณะนั้นซึ่งเราจะขอเรียก weight นี้ว่าเป็น old-weight โดยที่เงื่อนไขการปรับ weight จะเกิดขึ้นได้ 3 กรณีดังนี้

กรณีที่ 1 ถ้า text ตัวนั้นของอินพุทไม่ตรงกับสมาชิกตัวใดใน old-weight เลย

กรณีที่ 2 ถ้า text ตัวนั้นซ้ำกับข้อมูลตัวใดตัวหนึ่งใน old-weight

กรณีที่ 3 ถ้าสมาชิกใน old-weight ไม่ตรงกับ text ของอินพุทเลย

จากทั้ง 3 กรณีที่กล่าวมาจะเป็นเงื่อนไขต่อเนื่องสำหรับการปรับ weight ในส่วนที่เป็น degree โดยค่า degree จะมีค่าอยู่ระหว่าง 0-1 โดยในการปรับค่า degree แต่ละครั้งจะใช้ค่า learning rate มามีส่วนร่วมโดยจะพิจารณาจากทั้ง 3 กรณีดังนี้

- เมื่อเกิดกรณีที่ 1 นำ text ตัวนั้นของอินพุทเพิ่มเข้าไปเป็นสมาชิกใหม่ให้กับ old-weight แล้วกำหนดค่า degree ของสมาชิกใหม่ให้เท่ากับค่าเริ่มต้น (initial)
- เมื่อเกิดกรณีที่ 2 นำ ค่า Learning Rate บวกเพิ่มเข้าไปกับค่า degree เดิมของold-weight ตัวที่ซ้ำนั้น
- เมื่อเกิดกรณีที่ 3 นำ ค่า Learning Rate ไปลบค่า degree เดิม old-weight ทั้งหมด

ในส่วนของการวัดประสิทธิภาพของอัลกอริทึมในงานวิจัยนี้ได้ใช้ค่า Entropy และค่า F-Measure เป็นตัวชี้วัดประสิทธิภาพของการจัดกลุ่ม ซึ่งจากผลการทดลองในบทที่ 5 เมื่อนำผลที่ได้มาเปรียบเทียบกับผลจากงานวิจัยของ M. Steinbach และคณะ [1] พบว่าตัวอัลกอริทึมของเรามีประสิทธิภาพที่ดีกว่าผลจากงานวิจัยนี้ [1] และเมื่อเปรียบเทียบกับ TPKNN พบว่างานวิจัยชิ้นนี้มีประสิทธิภาพดีกว่าเล็กน้อย [16]

5.2 ปัญหาที่พบในงานวิจัยนี้

ในงานวิจัยนี้ได้ทดลองกับข้อมูลที่เป็นข่าว Reuters-21578 ซึ่งสรุปปัญหาที่พบได้ดังนี้

5.2.1 ความทับซ้อนกัน (overlap) ของข่าว

เนื่องจากข่าวบางข่าวมีความหมายของข่าว ที่สามารถจัดให้อยู่ในหัวข้อข่าวได้มากกว่า 1 หัวข้อข่าว เมื่อเป็นเช่นนี้จึงทำให้ผลของการจัดกลุ่มด้วยตัวโมเดลของเราจึงให้ผลที่ผิดพลาดกับข่าวประเภทนี้

5.2.2 การหาคำสำคัญที่ไม่ตรงกับความหมายของเนื้อข่าว และการตัด stemming ของคำที่ยังไม่ถูกต้องทั้งหมด

ปัญหาในจุดนี้เกิดจากวิธีการหา Keyword ของเราใช้วิธีของการนับจำนวนที่เกิดขึ้นบ่อยในแต่ละข่าว ซึ่งการใช้วิธีนี้กับข่าวบางข่าวก็ไม่สามารถทำให้ได้ keyword ที่เป็นตัวแทนของเนื้อข่าวตามความหมายนั้นจริงๆ หรืออย่างใน Title ของข่าวบางข่าวก็พบว่าคำที่ปรากฏใน Title แต่กลับไม่ปรากฏอยู่ในเนื้อข่าวเลย และปัญหาที่เกิดจากการตัด stemming ของคำในงานวิจัยนี้ที่ยังไม่ถูกต้องทั้งหมด

5.3 แนวทางการพัฒนาในอนาคต

5.3.1 ปรับปรุงในส่วนของการหาคุณลักษณะและ stemming ของคำ

จากปัญหานี้ แนวทางในการพัฒนาในอนาคตจำเป็นต้องหาวิธีการกำหนดคุณลักษณะของเอกสารให้ได้ใกล้เคียงกับความหมายที่แท้จริงของเนื้อข่าวมากที่สุด ซึ่งอาจจะเพิ่มคุณลักษณะอื่นๆ เข้าไปเช่นรูปแบบโครงสร้างของประโยคเป็นต้น ซึ่งถ้าสามารถกำหนดคุณลักษณะที่ได้ความหมายใกล้เคียงกับเนื้อข่าว และตัด stemming ของคำได้ดีขึ้น ก็จะทำให้อัลกอริทึมสามารถจัดกลุ่มได้ถูกต้องมากขึ้น

5.3.2 ทดลองเพิ่มในส่วนข้อมูลลักษณะอื่นๆ

ในงานวิจัยนี้ได้ทดลองกับเฉพาะข้อมูลที่เป็นข้อมูลข่าวเท่านั้น ซึ่งต่อไปควรจะได้นำไปทดลองกับข้อมูลที่มีลักษณะของเนื้อหาแบบอื่นบางเช่นเนื้อหาเวปบนอินเตอร์เน็ตเป็นต้น นอกจากนี้ยังสามารถนำอัลกอริทึมนี้ไปประยุกต์เพื่อพัฒนาเป็นแอพพริเคชั่น สำหรับเป็นเครื่องมือช่วยในการค้นหาข้อมูล อย่างเช่น เสิร์จเอนจิ้น เป็นต้น

เอกสารอ้างอิง

[1] M. Steinbach, G. Karypis, and V. Kumar., "A comparison of document clustering techniques TextMining", Workshop, KDD, 2000.

[2] Khaled M. Hammouda. "Web Mining: Clustering Web Documents A Preliminary Review", Department of Systems Design Engineering, University of Waterloo, Waterloo, Ontario, Canada, 2000.

[3] Michael Negnevitsky, "Artificial Intelligence: A Guide to Intelligent Systems",Addision-Wesley,2001.

[4] S.Haykin. "Neural Networks: A Comprehensive Foundation", Prentice Hall International, Inc., ISBN: 0-13-908385-5.

[5] Martin T.Hagan,Howard B.Demuth,Mark Beale , "Neural Network Design", PWS Publishing Company,1995.

[6] Jacek M. Zurada, "Introduction to Artificial Neural Systems" Info access Distribution Pte Ltd, 1992.

[7] El-Sonbaty Y.A. and Ismail M.A., "Fuzzy Clustering for symbolic Data", IEEE Trans. On Fuzzy Systems, vol.6, no.2, pp. 1 95-204, 1998.

[8] Lewis D.D., "Reuters-21578 text categorization test collection distribution 1.0.", http:// www.reseach.att.com/lewis,1999.

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

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

[11] A. K. Jain , M. N. Murty , P. J. Flynn, "Data clustering: a review", ACM Computing Surveys (CSUR), v.31 n.3, p.264-323, Sept. 1999.

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

[13] Merkl D. "Text Data Mining" A Handbook of Natural Language Processing:Techniques and Applications for The Processing of Language as Text, Edited by Dale R., Moisl H., and H., Mercel Dekker, New York, 1998.

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

[15] K.C.Gowda and T.V. Ravi , "Divisive Clustering of symbolic clustering using the concept of both similarity and dissimilarity", Pattern recognition, Vol.28,No. 8, PP.1277-1282,1995.

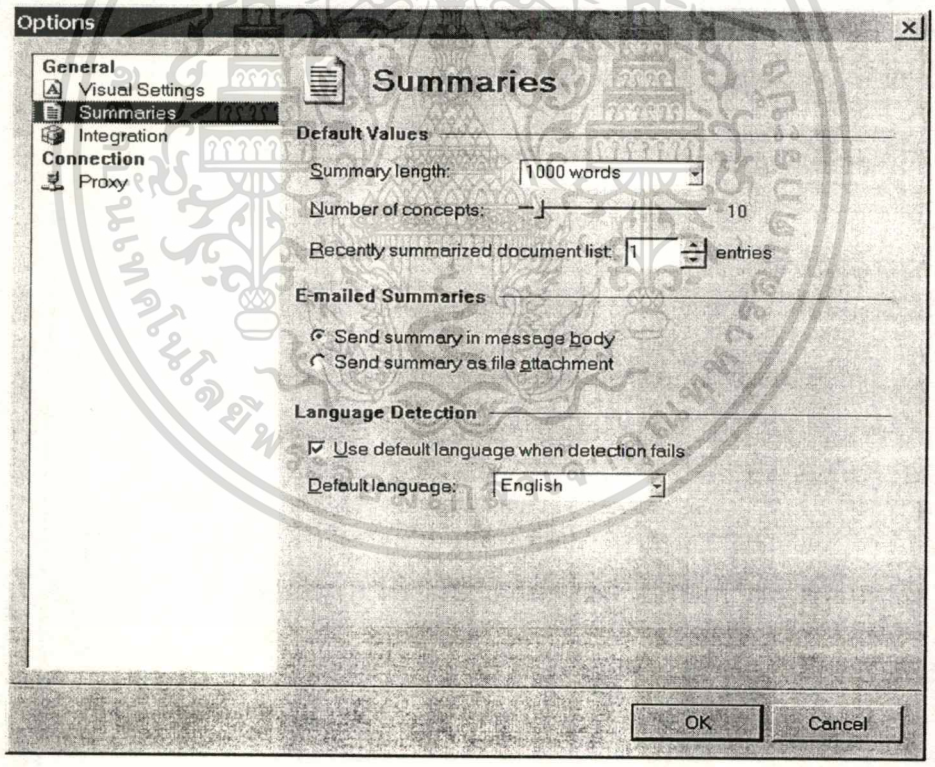
[16] ทรงพล ชุติพงศ์พัฒนกุล, "เท็กโปรเซสซิ่งโคโฮเนนนิวรอลเน็ตเวร์ค โดยใช้กระบวนการเรียนรู้แนวใหม่", วิทยานิพนธ์วิทยาศาสตรมหาบัณฑิต สาขาเทคโนโลยีสารสนเทศ บัณฑิตวิทยาลัย, สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง, 2545.

ภาคผนวก ก.

การใช้โปรแกรม Copernic Summarizer หาคำสำคัญของข่าว

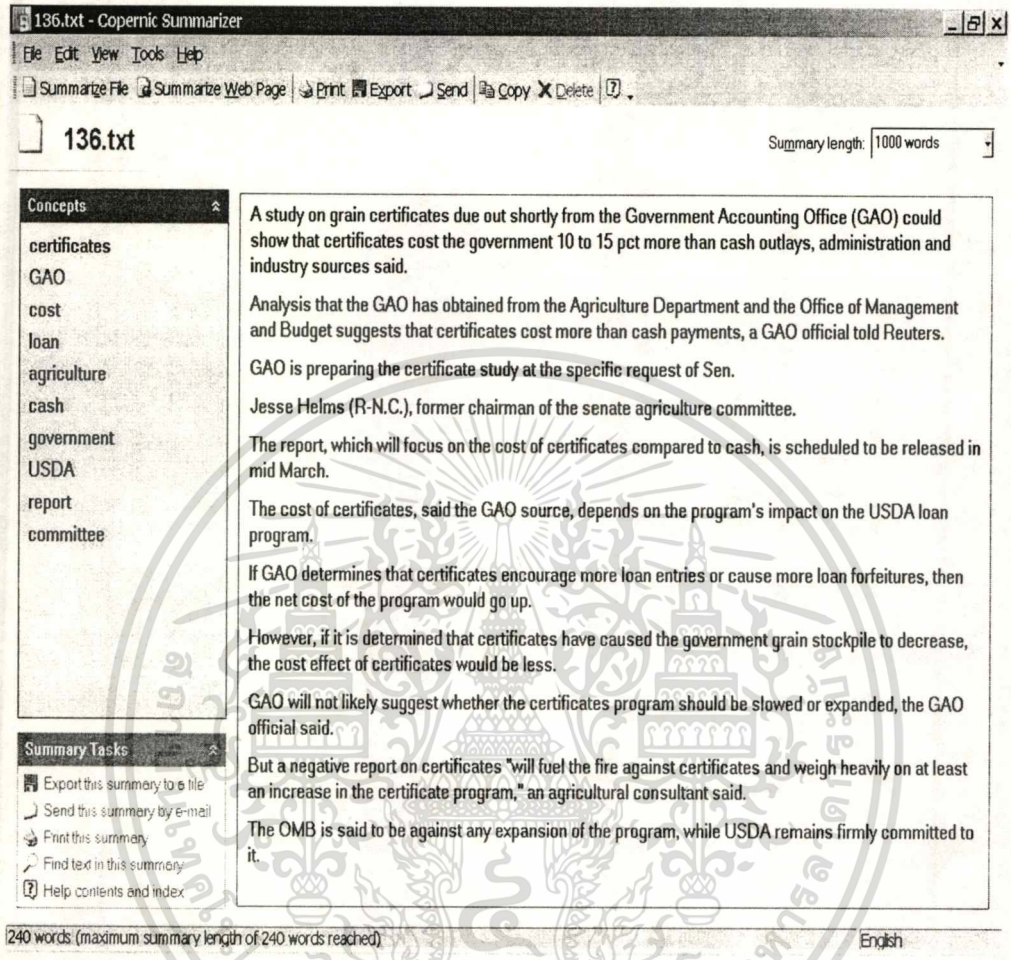
โปรแกรม Copernic Summarizer เป็นโปรแกรมที่มีความสามารถหลัก คือ ความสามารถในการหาคำสำคัญของเอกสารที่เป็นไฟล์ได้หลายรูปแบบ เช่น หาคำสำคัญจากหน้าเวปเพจ จากเท็กซ์ไฟล์ เป็นต้น ซึ่งผู้พัฒนาโปรแกรมนี้คือบริษัท Copernic Technologies จำกัด โดยสามารถดาวน์โหลดโปรแกรมมาทดลองใช้งานได้ที่ www.copernic.com โดยที่เวชั่นที่ใช้ในงานวิจัยนี้คือ เวอร์ชั่น 2.0 สำหรับการติดตั้งตัวโปรแกรมก็เหมือนกับการติดตั้งโปรแกรมบน Windows โดยทั่วไปจึงไม่ขอกล่าวถึงส่วนของการติดตั้งโปรแกรมในที่นี้

หลังจากได้ติดตั้งโปรแกรมเรียบร้อยแล้ว ให้ปรับค่าพารามิเตอร์ชื่อ "Number of concepts" ซึ่งเป็นพารามิเตอร์สำคัญ ที่ให้เรากำหนดค่าเพื่อให้โปรแกรมประมวลผลโดยหาคำสำคัญที่พบมากที่สุดในไฟล์เรียงตามลำดับตามจำนวนที่เรากำหนด ซึ่งในงานวิจัยนี้ได้กำหนดให้หาคำสำคัญ ไว้ที่ 10 ลำดับแรก ดังแสดงรูปที่ 1.ก



รูปที่ ก.1 แสดงการตั้งค่าคำสำคัญเท่ากับ 10 คำ

ต่อจากนั้นจะสั่งรันโปรแกรมโดยตัวอย่างผลลัพธ์ที่ได้หลังจากการรันไฟล์ 136.txt ซึ่งเป็นข่าวในหัวข้อ earn โดย 10 คำสำคัญแสดงในกรอบที่อยู่ซ้ายมือ ซึ่งคำสำคัญที่ได้นี้คือส่วนที่เป็น Keyword ที่ใช้เป็นอินพุทสำหรับการประมวลผลด้วยโครงข่ายในงานวิจัยนี้นั้นเอง



รูปที่ ก.2 แสดงผลลัพธ์ที่ได้หลังจากการรันโปรแกรม Copernic Summarizer

ภาคผนวก ข.

แสดงตัวอย่างข่าว Reuters-21578

ข.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>

ข.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 ข.1 แสดงตัวอย่างของคำที่ได้หลังจากตัด stemming ของคำแล้ว

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

ตารางที่ ข.2 แสดงตัวอย่างของคำที่ตัด stemming และคำที่เป็น stop word

ก่อนตัด stemming	หลังตัด stemming
tons	ton
omits	omit
accepts	accept
minerals	mineral
groups	group
called	call
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 ข.3 แสดงตัวอย่างโครงสร้างของข้อมูลข่าว Reuters-21578 ที่นำเข้าโมเดล

Title	Keyword
data(1,1).text(1,:) = {'bahia'}; data(1,1).text(2,:) = {'cocoa'}; data(1,1).text(3,:) = {'review'};	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'};
data(2,1).text(1,:) = {'standard'}; data(2,1).text(2,:) = {'oil'}; data(2,1).text(3,:) = {'form'}; data(2,1).text(4,:) = {'finance'}; data(2,1).text(5,:) = {'unit'};	data(2,2).text(1,:) = {'standard'}; data(2,2).text(2,:) = {'oil'}; data(2,2).text(3,:) = {'venture'}; data(2,2).text(4,:) = {'north'}; data(2,2).text(5,:) = {'america'}; data(2,2).text(6,:) = {'dlr'}; data(2,2).text(7,:) = {'joint'}; data(2,2).text(8,:) = {'manage'}; data(2,2).text(9,:) = {'commit'}; data(2,2).text(10,:) = {'oversight'};

ภาคผนวก ค.
ผลงานวิจัยที่ได้รับการตีพิมพ์

1. Worapoj Kreesuradej and Somkid Saensanor "Document Clustering Using A Text Processing Competitive Learning Neural Network", The 2001 International Technical Conference on Circuits and Communications, Tokushima, July 2001, Japan, pp.1125-1127.

2. Worapoj Kreesuradej and Somkid Saensanor "The Text Processing Competitive Learning Neural Network", The 3rd International Symposium on Communications and Information Technologies, October 2003, Songkhal, Thailand, pp.617-620.

Proceedings of ITC-CSCC '01, Tokushima, Japan

Document Clustering Using A Text Processing Competitive Learning Neural Network

Worapoj Kreesuradej and Somkid Sansanor
Advanced Computer Application and Design Research Group,
Faculty of Information Technology,
King Mongkut's Institute of Technology Ladkrabang,
Ladkrabang, Bangkok 10520 Thailand
Tel: (662) 737-2551-4 ext. 522
Fax: (662) 326-9074
Email: worapoj@it.kmitl.ac.th, s2067118@kmitl.ac.th

Abstract: This paper proposes document clustering using a text processing competitive learning neural network The text processing competitive learning neural networks works directly on textual information without mapping documents onto some representations that have quantitative features. The inputs of the proposed neural network directly receive a qualitative value without mapping the qualitative value into a numerical value. Then, based on a new unsupervised learning algorithm and the concepts of dissimilarity measure for symbolic objects, the proposed neural network assigns cluster labels to the objects.

1. Introduction

Several clustering techniques for objects whose feature values are numerical values are well known. Several neural networks such as ART1, ART2, and competitive learning neural networks are proposed for clustering such objects. Recently, clustering problems are extended for document clustering. To cluster documents by using typical neural networks, each document has to be mapped onto some representations that have quantitative features. One of most widely used representation is the vector-space model [1]. Then, the typical neural networks can be applied for clustering documents. However, the utilization of the vector-space model may led to a very-high dimensional feature space. In addition, this feature space is generally not free from correlation [1].

Unlike the previous works, this paper proposes a text processing competitive learning neural network to solve that problem. The text processing competitive learning neural networks works directly on textual information without mapping documents onto some representations that have quantitative features. The inputs of the proposed neural network directly receive a qualitative value without mapping the qualitative value into a numerical value. Then, based on a new unsupervised learning algorithm and the concepts of dissimilarity measure for symbolic objects, the proposed neural network assigns cluster labels to the objects.

2. Document Representation

Here, a document, Doc, for clustering task can be written as the Cartesian product of specific values of its features D_k's as [2]

$$Doc = D_1 * D_2 * \ldots * D_k$$

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

$$Doc = Title * Keyword$$

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

3. Dissimilarity Measure

To formalize the problem of document clustering, a definition of dissimilarity between documents must be defined. Here, according to El-Sonbaty [2], dissimilarity between two document A and B is defined as

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

For the k^{th} feature, $D(A_k,B_k)$ can be decomposed into two components: the dissimilarity component due to span, D_s (A_k,B_k) and the dissimilarity component due to content, D_c (A_k,B_k). Each Component can be defined as below:

$$D_s(A_k,B_k) = \frac{\left|\text{Length of } A_k - \text{Length of } B_k\right|}{\text{span lenght of } A_k \text{ and } B_k}$$

$$D_c(A_k,B_k) = \frac{\left|\text{Length of } A_k + \text{Length of } B_k - \text{Length of intersection of } A_k \text{ and } B_k\right|}{\text{span lenght of } A_k \text{ and } B_k}$$

where the length of feature is number of its elements and the span length of two feature value is defined as number of element in their union. The net dissimilarity between A_k and B_k is

$$D(A_k,B_k) = D_s(A_k,B_k) + D_c(A_k,B_k)$$

the concepts of dissimilarity measure will be used to measure the dissimilarity between documents in the next section.

4. The Text Processing Competitive Learning Neural Network

The architecture of the proposed neural networks is shown in Figure 1.

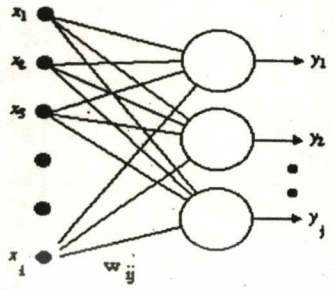


Figure 1 the architecture of the text processing competitive learning neural network

Basically, the proposed network consists of two layers. A layer of input units each of which is fully connected to a set of output units. These output units are arranged in some topology where the most common choice is represented by a two-dimensional grid. Unlike conventional neural networks, input units of the proposed neural network receive qualitative values. The weigh from the input unit 'i' to the output unit 'j', w_{ij}, is defined as

$$w_{ij} = \{(A_{1ij}, e_{1ij}), (A_{2ij}, e_{2ij}), (A_{3ij}, e_{3ij}), \ldots (A_{pij}, e_{pij})\},$$

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

5. Learning Algorithm

Basically, the proposed learning algorithm, which is an unsupervised learning, is based on the winner-take-all learning and the concepts of dissimilarity measure in section 3. According to the winner-take-all learning, the neurons compete among each other to be the one that fires. As only one neuron will fire, it can be declared the winner. To find a wining neural, the proposed algorithm, based on the concepts of the dissimilarity measure, computes dissimilarity between input patterns, which are qualitative values, and that unit's weight vector. Then, the output node that has the smallest values is declared the wining node. The details of the proposed algorithm can be presented as below:

Step0: initialize weights w_{ij}. Each weight can be initialized from the training data set arbitrarily.

Step1: While stopping condition is false, do step 2-6

Step2: For each input vector

$$X = (x_1, x_2, \ldots, x_d)^T \text{ , do step 3-6}$$

Step3: For each output unit 'j', compute

$$\|x - w_j\| = \sum_{k=1}^{d} \sum_{n=1}^{p} D(x_k, A_{pkj}) \bullet e_{pkj} .$$

p = no. of values of w_{kj}
d = no. of inputs

Step4: Find index J such that $\|x-w_j\|$ is a minimum

Step5: For all weights that connect to the wining node J,

$$i.e., w_{iJ} = \left\{ \begin{matrix} (A_{1iJ}, e_{1iJ}), \\ (A_{2iJ}, e_{2iJ}), \\ (A_{3iJ}, e_{3iJ}), \ldots, \\ (A_{piJ}, e_{piJ}) \end{matrix} \right\} \text{ for } i = 1,2,3,\ldots,k$$

$$w_{iJ}^{(new)} = w_{iJ}^{(old)} \cup x_i$$

and

$$e_{piJ}^{(new)} = \begin{cases} f\left(e_{piJ}^{(old)} + \eta\right) & \text{if } A_{piJ} \in w_{iJ} \cap x_i \\ f\left(e_{piJ}^{(old)} - \eta\right) & \text{if } A_{piJ} \notin w_{iJ} \cap x_i \end{cases},$$

where f(.) is defined as below:

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

Step6: Test stopping condition. Like classical competitive neural networks, there are several strategies for stopping condition such as reducing learning rate, η, until η=0.

6. Future Works

In the future, some experiment results that are conducted on the Reuter-21578 new articles [4] will be reported. The result of the selecting corpus will be that the labeling is reflect some semantic coherence that can be trusted.

References

[1] Merkl D., "Text Data Mining," *A Handbook of Natural Language Processing: Techniques and Applications for The Processing of Language as Text*, Edited by Dale R., Moisl H., and Somers H., Mercel Dekker, New York, 1998.

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

[3] 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, 1998.

[4] Lewis D.D. Reuter-21578 text categorization test collection distribution 1.0. http://www.research.att.com/lewis , 1999

A Text Processing Competitive Learning Neural Network

Worapoj Kreesuradej, Ph.D., Somkid Saensanor,
Songpol Chutipongpattanakul and Warune Kruaklai

Faculty of Information Technology,
King Mongkut's Institute of Technology Ladkrabang,
Ladkrabang, Bangkok 10520 Thailand
Tel: (662) 737-2551-4 ext. 522
Fax: (662) 326-9074
Email: worapoj@it.kmitl.ac.th, s2067118@kmitl.ac.th,
s1067011@kmitl.ac.th, warune@it.kmitl.ac.th.

Abstract

This paper proposes document clustering using a text processing competitive learning neural network. The text processing competitive learning neural network works directly on textual information without mapping taxts onto some representations that have quatitative features. The inputs of the proposed neural network directly receive a qualitative value without mapping the qualitative value into a numerical value.Then, based on a new unsupervised learning algorithm and the concepts of dissimilarity measure for symbolic objects.The proposed neural network assigns cluster labels to the Reuter-21578 text categorization test collection distribution 1.0.The results experimental of this paper we persents with 5 topics of Reuter news groups.

1. Introduction

Several clustering techniques for objects whose feature values are numerical values are well known. Several neural networks such as ART1, ART2, and competitive learning neural networks are proposed for clustering such objects. Recently, clustering problems are extended for document clustering. To cluster documents by using typical neural networks, each document has to be mapped onto some representations that have quantitative features. One of most widely used representation is the vector-space model. Then, the typical neural networks can be applied for clustering documents. However, the utilization of the vector-space model may led to a very-high dimensional feature space. In addition, this feature space is generally not free from correlation.

Unlike the previous works, this paper proposes a text processing competitive learning neural network to solve that problem. The text processing competitive learning neural networks works directly on textual information without mapping documents onto some representations that have quantitative features. The inputs of the proposed neural network directly receive a qualitative value without mapping the qualitative value into a numerical value. Then, based on a new unsupervised learning algorithm and the concepts of dissimilarity measure for symbolic objects, the proposed neural network assigns cluster labels to the objects.

2. Dissimilarity measure

According to El-Sonbaty [2], dissimilarity between two document A and B is defined as

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

For the k^{th} feature, $D(A_k,B_k)$ can be decomposed into two components: the dissimilarity component due to span, D_s (A_k,B_k) and the dissimilarity component due to content, D_c. This paper represented only D_c (A_k,B_k). Component due to content can be defined as below:

$$D_c(A_k, B_k) = \frac{|Y - 2 * Z|}{\text{span lenght of } A_k \text{ and } B_k} \qquad (2)$$

Where :

Y = Length of A_k + Length of B_k

Z = Length of intersection of A_k and B_k

where the length of feature is number of its elements and the span length of two feature value is defined as number of element in their union. The net dissimilarity between A_k and B_k is

$$D(A_k, B_k) = D_c(A_k, B_k) \qquad (3)$$

the concepts of dissimilarity measure will be used to measure the dissimilarity between documents in the next section.

3. A text processing competitive learning neural network

The architecture of the proposed neural networks is shown in Figure 1.

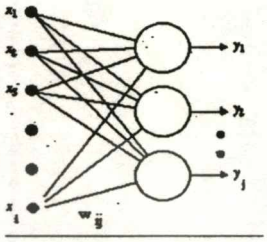


Fig. 1 the architecture of a text processing competitive learning neural network.

Basically, the proposed network consists of two layers. A layer of input units each of which is fully connected to a set of output units. These output units are arranged in some topology where the most common choice is represented by a two-dimensional grid. Unlike conventional neural networks, input units of the proposed neural network receive qualitative values. The weigh from the input unit 'i' to the output unit 'j', w_{ij}, is defined as

$$w_{ij} = \{(A_{1ij}, e_{1ij}), (A_{2ij}, e_{2ij}), (A_{3ij}, e_{3ij}), \ldots, (A_{pij}, e_{pij})\}$$

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

4. Learning algorithm

Basically, the proposed learning algorithm, which is an unsupervised learning, is based on the winner-take-all learning and the concepts of dissimilarity measure in section 2. According to the winner-take-all learning, the neurons compete among each other to be the one that fires. As only one neuron will fire, it can be declared the winner. To find a wining neural, the proposed algorithm, based on the concepts of the dissimilarity measure, computes dissimilarity between input patterns, which are qualitative values, and that unit's weight vector. Then, the output node that has the smallest values is declared the wining node. The details of the proposed algorithm can be presented as below:

Step 0: initialize weights w_{ij}. Each weight can be initialized from the training data set arbitrarily.

Step 1: While stopping condition is false, do step 2-6

Step 2: For each input vector

$X = (x_1, x_2, \ldots, x_d)^T$, do step 3-6

Step 3: For each output unit 'j', compute

$$\|X - W_j\| = \sum_{k=1}^{d} \sum_{n=1}^{p} D(x_k, A_{pkj})$$

p = Number of values of w_{kj}
d = Number of inputs

Step 4: Find index J such that $\|X - W_J\|$ is a minimum

Step 5: For all weights that connect to the wining node J,i.e.,

$$w_{iJ} = \left\{ \begin{matrix} (A_{1iJ}, e_{1iJ}), \\ (A_{2iJ}, e_{2iJ}), \\ (A_{3iJ}, e_{3iJ}), \ldots, \\ (A_{piJ}, e_{piJ}) \end{matrix} \right\} \text{ for } i = 1,2,3,\ldots,k$$

$$w_{iJ}^{(new)} = w_{iJ}^{(old)} \cup x_i$$

step 5.1: Update degree values of weights

If $w_{iJ}^{(new)} \in w_{iJ}^{(old)}$ *Then*

$$e_{piJ}^{(new)} = \begin{cases} f\left(e_{piJ}^{(old)} + \eta(1 - D)\right) & \text{if } A_{piJ} \in w_{iJ} \cap x_i \\ f\left(e_{piJ}^{(old)} - \eta(1 - D)\right) & \text{if } A_{piJ} \notin w_{iJ} \cap x_i \end{cases}$$

step 5.2: Update degree values of weights

If $w_{iJ}^{(new)} \notin w_{iJ}^{(old)}$ *Then*

$$e_{ij}^{(new)} = I$$

where f(.) is defined as below:

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

and:
= Learning rate constant
D = The net dissimilarity value of index J
I = Initial value have value between 0 to 1

Step 6: Test stopping condition. Like classical competitive neural networks, there are several strategies for stopping condition such as train data set until data in each one node not change.

5. Data set

5.1 Document representation

Here, a document, Doc, for clustering task can be written as the Cartesian product of specific values of its features D_k's as [2]

$$Doc = D_1 * D_2 * \ldots * D_k$$

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

$$Doc = Title * Keyword$$

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

5.2 Reuter-21578 data set

In all of the data sets, we have removed stop words,i.e., common words sush as "a", "are" and "the". After removed words next step we have perfoemed finded keywords of all body news using word frequency teachique. The summary of documents used in this paper is shown in table 1. Data sets are form Reuter-21578 text categorization test collection distribution 1.0.

Table 1. Summary of Reuter-21578 data set.

Topics	No. of Data Testing	No. of Data training
Earn	3183	355
Acq	1932	331
Money-fx	596	239
Crude	496	221
Grain	467	215

6. Experimental result

6.1 Synthesized english alphabets data

For this experiment, some english alphabets are used to represent value of each features.The values of title feature and keyword feature of each cluster are shown in figure 2 and 3 respectively.

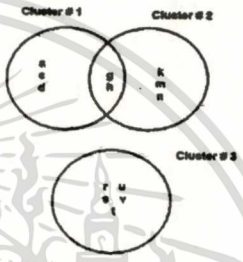


Fig. 2 Title feature profile

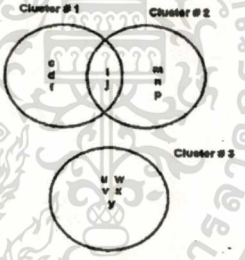


Fig. 3 Keyword feature profile

As an example, some documents that are generated according to data profile in figure 2 and figure 3 are shown in table 2.

Table 2. Some documents of the synthesized english alphabets training data.

No.	Data Value	
	Title	Keyword
1	a,d,g,h	d,j
2	n,k	i,m,n,p
3	r,s,t,v	u,w,y
4	h,k,n	j,p

Here, the training data set consist of 34 documents from cluster number 1, 31 documents from cluster number 2 and 35 documents from cluster number 3. Then, a testing data set of 1000 documents is also generated based on the data profile in figure 2 and figure 3. The testing data set consists of 298 documentsfrom cluster number 1, 277 documents from cluster number 2 and 339 documents from cluster number 3.

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

$$r = \left[1 - \left(\frac{\sum_{i=1}^{c} doc_i}{n} \right) \right] * 100\% \qquad (4)$$

Where doc_i is a number of documents that are incorrectly assigned to a wrong cluster.

According to the testing set, the clustering accuracy, i.e. r, of the proposed network gives the accuracy 94 percent.

6.2 Reuter-21578 data

For this experiment, we uesd data in table 1 to tained and tested our algoritham. The best results of some learning rate are shown in table 3 – large is better for F-masure and small is better for Entropy. Table 4 shown numbers of data in each node output, that is topic earn has node 5 is agent, topic acq has node 1 is agent, topic money-fx has node 2 is agent, topic grain has node 4 is agent, topic earn has node 3 is agent.

Table 3. An Entropy and F-measuer at learning rate = 0.00001.

Epoch	Entropy	F-measure
145	0.5355	0.8457
146	0.5331	0.8458
147	0.5070	0.8565
148	0.5017	0.8593
149	0.5210	0.8528
150	0.5131	0.8534

Table 4. An output of data in each output node at learning rate = 0.00001 and epoch = 148 .

Topics	Output Node					
	1	2	3	4	5	Acc (%)
Earn	186	117	17	35	2826	88.8
Acq	1675	85	33	49	90	86.7
Money-fx	29	470	95	0	2	78.9
Crude	48	99	4	344	1	69.4
Grain	30	26	391	17	3	83.7
Average Accuracy Total (%) = 81.5						

Acc. = Accuracy

The result of clustering with data Reuter-21578. An algorithm performed to satisfy,thai is a average accuracy of clustering equal 81.5 percent at entropy = 0.5017 and f-measure = 0.8593.

7. Conclusion

This paper presented a new algorithm for text clustering and the experimental result on reuter-21578 data set. The experimental result show that the performance of this clustering algorithm is vary well and the total accuracy is 81.5 %. It have some error on clustering because some news data is member of more than one topic and training data on some topic is not cover feasible issue in test data. In the future, some experiment results that are conducted on the Reuter-21578 news articles [4] more than 5 topics will be reported. The result of the selecting corpus will be that the labeling is reflect some semantic coherence that can be trusted.

8. References

[1] Merkl D., "Text Data Mining," A Handbook of Natural Language Processing: Techniques and Applications for The Processing of Language as Text, Edited by Dale R., Moisl H., and Somers H., Mercel Dekker, New York, 1998.

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

[3] 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, 1998.

[4] Lewis D.D. Reuter-21578 text categorization test collection distribution 1.0. http://www.research.att. com/lewis , 1999

[5] M. Steinbach, G. Karypis, and V. Kumar. A comparison of document clustering techniques. In KDD Workshop on Text Mining, 2000.

ประวัติผู้เขียน

นายสมคิด แสนเสนาะ เกิดเมื่อวันที่ 27 มกราคม 2518 ที่จังหวัดนครศรีธรรมราช สำเร็จการศึกษาวิศวกรรมศาสตรบัณฑิต (วิศวกรรมไฟฟ้ากำลัง) จากมหาวิทยาลัยเทคโนโลยีมหานคร ปีการศึกษา 2541

ปี พ.ศ. 2542 ศึกษาต่อปริญญาโท สาขาเทคโนโลยีสารสนเทศ คณะเทคโนโลยีสารสนเทศ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง