เท็กโปรเซสซิ่งโคโฮเนนนิวรอลเน็ตเวิร์คโดยใช้กระบวนการเรียนรู้แนวใหม่

TEXT PROCESSING KOHONEN NEURAL NETWORKS WITH A NEW LEARNING ALGORITHM

ทรงพล ชุติพงศ์พัฒนกุล
SONGPOL CHUTIPONGPATTANAKUL

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต
สาขาวิชาเทคโนโลยีสารสนเทศ
บัณฑิตวิทยาลัย
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
พ.ศ. 2546

ISBN 974-324-583-9

เท็กโปรเซสซิ่งโคโฮเนนนิวรอลเน็ตเวิร์คโดยใช้กระบวนการเรียนรู้แนวใหม่

TEXT PROCESSING KOHONEN NEURAL NETWORKS WITH A NEW LEARNING ALGORITHM

ทรงพล ชุติพงศ์พัฒนกุล

SONGPOL CHUTIPONGPATTANAKUL

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต

สาขาวิชาเทคโนโลยีสารสนเทศ

บัณฑิตวิทยาลัย

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ. 2546

ISBN 974-324-583-9

TEXT PROCESSING KOHONEN NEURAL NETWORKS WITH A NEW LEARNING ALGORITHM

SONGPOL CHUTIPONGPATTANAKUL

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

2003

ISBN 974-324-583-9

บัณฑิตวิทยาลัย

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ใบรับรองวิทยานิพนธ์

หัวข้อวิทยานิพนธ์ เท็กโปรเซสซิ่ง โคโฮเนนนิวรอลเน็ตเวิร์คโดยใช้กระบวนการเรียนรู้แนวใหม่
TEXT PROCESSING KOHONEN NEURAL NETWORKS WITH A NEW LEARNING ALGORITHM

ชื่อนักศึกษา นายทรงพล ชุติพงศ์พัฒนกุล

รหัสประจำตัว 41067011

ปริญญา วิทยาศาสตรมหาบัณฑิต

สาขาวิชา เทคโนโลยีสารสนเทศ

อาจารย์ผู้ควบคุมวิทยานิพนธ์ ผศ.ดร.วรพจน์ กรีสุระเดช

คณะกรรมการสอบวิทยานิพนธ์		ลายมือชื่อ
ผศ.ดร.วรพจน์	กรีสุระเดช	
รศ.ดร.วิเชียร	เปรมชัยสวัสดิ์	
ผศ.ดร.อาริต	ธรรมโน	
ผศ.ดร.โชติพัชร์	ภรณวลัย	

วัน/เดือน/ปี ที่สอบ 21 พฤษภาคม 2546 **เวลา** 13.00 น. **เป็นต้นไป**

สถานที่สอบ ณ ห้อง M03 (ชั้นลอย) อาคารเรียนรวมและปฏิบัติการคณะเทคโนโลยีสารสนเทศ

บัณฑิตวิทยาลัยรับรองแล้ว

(รศ.ดร.บุญวัฒน์ อัตชู)

คณบดีบัณฑิตวิทยาลัย

วันที่ 29 **เดือน** มิถุนายน **พ.ศ.** 2546

หัวข้อวิทยานิพนธ์	เท็กโปรเซสซิ่งโคโฮเนนนิวรอลเน็ตเวิร์คโดยใช้กระบวนการเรียนรู้แนวใหม่
นักศึกษา	นายทรงพล ชุติพงศ์พัฒนกุล
รหัสประจำตัว	41067011
ปริญญา	วิทยาศาสตร์มหาบัณฑิต
สาขาวิชา	เทคโนโลยีสารสนเทศ
พ.ศ.	2546
อาจารย์ผู้ควบคุมวิทยานิพนธ์	ผศ. ดร.วรพจน์ กรีสุระเดช

บทคัดย่อ

วัตถุประสงค์ของการทำ clustering(การจัดกลุ่ม) คือ การแยกข้อมูลออกเป็นกลุ่มๆ โดยข้อมูลที่อยู่ภายในกลุ่มเดียวกันจะมีค่าความเหมือนกันมากกว่าข้อมูลที่อยู่ต่างกลุ่มกัน เทคนิคในการ clustering สำหรับวัตถุที่มีค่าของคุณสมบัติเป็นค่าเชิงตัวเลข (numerical values) เป็นที่รู้จักกันอย่างแพร่หลาย เช่น K-mean, nearest neighbor, decision tree และ neural networks วิธีเหล่านี้สามารถทำการ clustering ข้อมูลที่มีคุณสมบัติเป็นค่าเชิงตัวเลขได้เป็นอย่างดี แต่ในปัจจุบัน เริ่มมีความสนใจในการทำ clustering ข้อมูลที่เป็นประเภทข้อความหรือเอกสารมากขึ้น แม้ว่าวิธีในการ clustering แบบเดิมๆจะสามารถใช้ในการ clustering เอกสารหรือข้อความได้ แต่ต้องมีการนำข้อมูลไปผ่านขั้นตอนการแปลงรูปข้อมูลคุณสมบัติของเอกสารเหล่านั้นให้เป็นค่าเชิงตัวเลขเสียก่อน จึงจะนำมาใช้ได้ วิธีนี้แม้จะเป็นที่ยอมรับกัน แต่การแปลงข้อมูลที่เป็นข้อความให้เป็นค่าเชิงตัวเลขนั้นอาจทำให้ข้อมูลสูญเสียความหมายในตัวมันเองไป นอกจากนี้ยังทำให้เสียเวลาค่อนข้างมากในการแปลงและกำหนดสัญลักษณ์เพื่อแทนข้อมูลเหล่านั้นในกรณีที่ข้อมูลมีการกระจายตัวกันมากด้วย

Text Processing Kohonen Neural Networks เป็นนิวรอลเน็ตเวิร์คที่ขยายความสามารถของ Self-Organizing Feature Maps ขึ้นเพื่อการทำ clustering ข้อมูลที่มีลักษณะเป็นข้อความได้โดยตรง โดยไม่ต้องผ่านกระบวนการในการแปลงรูปข้อมูล โดยประยุกต์แนวคิดเรื่องการเปรียบเทียบความแตกต่างของข้อมูลแบบ symbolic เข้าไปในส่วนของการทำ competition และ Synaptic Adaptation ของนิวรอลเน็ตเวิร์คแบบ Self-Organizing Feature Maps. นิวรอลเน็ตเวิร์คใหม่สามารถรับข้อมูลที่เป็นข้อความได้โดยตรงและทำการ clustering ข้อมูลที่คุณสมบัติมีค่าเป็นข้อความได้เป็นอย่างดี

Thesis Title	Text Processing Kohonen Neural Networks with a New Learning Algorithm
Student	Mr. Songpol Chutipongpattanakul
Student ID.	41067011
Degree	Master of Science
Programmed	Information Technology
Year	2003
Thesis Advisor	Asst.Prof.Dr. Worapoj Kreesuradej

ABSTRACT

Clusterings are methods to group a set of objects into clusters such that objects within the same cluster have a high degree of similarity, while objects belonging to different clusters have a high degree of dissimilarity. Several clustering techniques for objects whose feature values are numerical values are well known, Several technique such as k-mean algorithm, nearest neighbor, decision trees and neural networks are proposed for clustering such objects. Recently, clustering problems are extends for document clustering. To cluster documents by using conventional techniques each document has to be mapped onto some representations that have quantitative features. This mapping use several time in processing and may cause data to loose their real meaning.

The Text Processing Kohonen Neural Network is extened ability of Self-Organizing Feature Maps. The Text Processing Kohonen Neural Network works directly on textual information without mapping documents onto some representation that has quantitative features and can assigns cluster labels to the objects. By modifing competition and Synaptic Adaptation process of the proposed neural network with the concept of dissimilarity measure for symbolic objects, The proposed neural network can directly receive a qualitative value without mapping the qualitative value into numerical value and can cluster the objects.

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงได้อย่างดีจากความกรุณาของ ผศ.ดร.วรพจน์ กรีสุระเดช ซึ่งเป็นอาจารย์ควบคุมวิทยานิพนธ์ฉบับนี้ ที่คอยให้คำปรึกษา คำแนะนำและแนะแนวทางแก้ปัญหาของงานวิจัยรวมทั้งวิชาการต่าง ๆ ที่เกี่ยวข้อง ซึ่งข้าพเจ้าซาบซึ้งในความอนุเคราะห์ของท่านและกราบขอบพระคุณเป็นอย่างสูง

กราบขอบพระคุณบิดา-มารดา และญาติพี่น้องของข้าพเจ้า ซึ่งให้กำลังใจและช่วยเหลือโดยตลอด

ขอขอบคุณ เพื่อน ๆ พี่ ๆ รวมทั้งน้อง ๆ ทุกคนในคณะเทคโนโลยีสารสนเทศ ที่ให้กำลังใจและคำปรึกษาตลอดมา

ขอขอบคุณบัณฑิตวิทยาลัย สถาบันเทคโนโลยีเจ้าคุณทหารลาดกระบัง ที่ให้ทุนสนับสนุนการทำวิทยานิพนธ์ครั้งนี้

สุดท้ายขอขอบคุณคณาจารย์และเจ้าหน้าที่คณะเทคโนโลยีสารสนเทศที่ให้ความรู้และช่วยเหลือในทุกๆด้าน

คุณค่าและประโยชน์ที่พึงได้จากวิทยานิพนธ์นี้ ข้าพเจ้าขอมอบแด่ผู้มีพระคุณทุกท่าน

ทรงพล ชุติพงศ์พัฒนกุล

สารบัญ

หน้า

สารบัญ (ต่อ)

หน้า

สารบัญ (ต่อ)

หน้า

สารบัญตาราง

ตารางที่ หน้า

สารบัญรูป

รูปที่ หน้า

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

การทำ clustering (การจัดกลุ่ม) คือ การแยกข้อมูลออกเป็นกลุ่มๆ โดยข้อมูลที่อยู่ภายในกลุ่มเดียวกันจะมีค่าความเหมือนกันมากกว่าข้อมูลที่อยู่ต่างกลุ่มกัน ซึ่งเราสามารถนำผลลัพธ์จากการ clustering มาใช้ในลักษณะของการแบ่งกลุ่มของข้อมูลหรืออาจทำไปใช้ในลักษณะของการแยกย่อยข้อมูลเพื่อเลือกเอาข้อมูลเฉพาะกลุ่มที่เราสนใจเอาไปทำการวิเคราะห์ด้วยวิธีการอื่นๆต่อไป การทำเช่นนี้ทำให้สามารถลดปริมาณการใช้ทรัพยากรของระบบลงและยังทำให้ได้ผลลัพธ์ที่เฉพาะเจาะจงมากขึ้น เทคนิคในการ clustering [1],[2]สำหรับวัตถุที่มีค่าของคุณสมบัติเป็น numerical values(ค่าเชิงตัวเลข) เป็นที่รู้จักกันอย่างแพร่หลาย เช่น Agglomerative Hierarchical clustering เป็นการ clustering ในแบบ Hierarchical clustering โดยจะสร้าง cluster จากการจัดให้กลุ่มที่ใกล้เคียงกันมารวมกันในแต่ละรอบของการทำงาน K-mean, nearest neighbor เป็นโมเดลในแบบของ partition clustering ซึ่งมีแนวคิดจากการหาตัวแทนของข้อมูลจำนวน k ตัว จากข้อมูลทั้งหมดโดยผลรวมของค่าความแตกต่างภายในกลุ่มเดียวกันจะน้อยที่สุด Fuzzy C-Mean เป็นโมเดลการ clustering หนึ่งในแบบของ fuzzy clustering ซึ่งมีความแตกต่างจากการ clustering ในแบบอื่นๆคือข้อมูลข้อมูลหนึ่งอาจเป็นสมาชิกของ cluster 2 cluster ในเวลาเดียวกันได้

แม้ว่าวิธีเหล่านี้สามารถทำการ clustering ข้อมูลที่มีคุณสมบัติเป็นค่าเชิงตัวเลขได้เป็นอย่างดี แต่ในปัจจุบัน ความสำคัญของข้อมูลที่เป็นประเภทข้อความเริ่มมีมากขึ้น ข้อมูลที่เป็นเอกสารข้อความมีจำนวนมากขึ้น แม้ว่าวิธีในการ clustering แบบเดิมๆจะสามารถใช้ในการ clustering เอกสารเหล่านี้ได้ แต่ก็ต้องมีการนำข้อมูลไปผ่านขั้นตอนการแปลงรูปข้อมูลคุณสมบัติของเอกสารเหล่านั้นให้เป็นค่าเชิงตัวเลขเสียก่อน จึงจะนำมาใช้ได้ วิธีการนี้แม้จะให้ผลลัพธ์ในระดับที่ยอมรับได้ แต่การแปลงข้อมูลที่เป็นข้อความให้เป็นค่าเชิงตัวเลขก่อนนั้น อาจทำให้ข้อมูลสูญเสียความหมายในตัวมันเองไป นอกจากนี้ยังทำให้เสียเวลาค่อนข้างมากในการแปลงและกำหนดสัญลักษณ์เพื่อแทนข้อมูลเหล่านั้นในกรณีที่ข้อมูลมีการกระจายตัวกันมากด้วย

Text Processing Kohonen Neural Networks เป็นโครงข่ายประสาทเทียม (neural network) ที่ขยายความสามารถของ Self-Organizing Feature Maps เพื่อการทำ clustering ข้อมูลที่มีลักษณะเป็นข้อความได้โดยตรง ไม่ต้องผ่านกระบวนการในการแปลงรูปข้อมูล โดยประยุกต์แนวคิดเรื่องการเปรียบเทียบความแตกต่างของ symbolic data เข้าไปในส่วนของการทำ competition และ Synaptic Adaptation ของโครงข่ายประสาทเทียมแบบ Self-Organizing Feature Maps. โครงข่าย

ประสาทเทียมแบบใหม่สามารถรับข้อมูลที่เป็นข้อความได้โดยตรงและทำการ clustering ข้อมูลที่คุณสมบัติมีค่าเป็นข้อความได้เป็นอย่างดี

1.2 วัตถุประสงค์ของงานวิจัย

1.2.1 เพื่อศึกษาวิธีการในการทำ clustering

1.2.2 เพื่อศึกษากระบวนการการทำงานของ Kohonen Self Organizing Feature Maps

1.2.3 เพื่อพัฒนาอัลกอริธึมในการทำ clustering ข้อมูลประเภทข้อความ โดยสามารถจัดการกับข้อมูลประเภทข้อความได้โดยตรง

1.3 ขอบเขตของงานวิจัย

งานวิจัยนี้ได้เสนออัลกอริธึมในการทำ clustering ข้อมูลประเภทข้อความ ให้สามารถจัดการกับข้อมูลประเภทข้อความได้โดยตรง ไม่ต้องทำการแปลงข้อมูลนั้นให้อยู่ในรูปของข้อมูลเชิงตัวเลขก่อน โดยประยุกต์แนวคิดในการหาค่าความแตกต่างของ symbolic data เข้ามาใช้ในประบวนการการทำ clustering ของ Kohonen Self Organizing Feature Maps และ conscience algorithm

1.4 ขั้นตอนการทำวิจัย

1.4.1 ศึกษาทฤษฎีและบทความต่างๆ ที่มีความเกี่ยวข้องกับงานวิจัยนี้

1.4.2 ศึกษาปัญหาในการทำ clustering ข้อมูลที่เป็นข้อความ

1.4.3 เขียนโปรแกรมเพื่อทำการ clustering เอกสารในกรณีที่คุณสมบัติที่ใช้บ่งบอกถึงตัวเอกสารมีค่าข้อมูลเป็นqualitative value

1.4.4 ทดลองกับข้อมูลตัวอย่าง พร้อมทั้งแก้ไขข้อผิดพลาดของโปรแกรม

1.4.5 รวบรวมผลการทดลองที่ได้จากโปรแกรม

1.4.6 วิเคราะห์และสรุปผลการดำเนินงาน

1.4.7 จัดทำเอกสารประกอบวิทยานิพนธ์

1.5 ประโยชน์ที่คาดว่าจะได้รับ

1.5.1 เข้าใจขั้นตอนและวิธีการต่างๆ ในการทำ clustering

1.5.2 เข้าใจกระบวนการวิจัยอย่างมีขั้นตอน

1.5.3 อัลกอริธึมใหม่ที่ได้สามารถนำไปประยุกต์ใช้ในกระบวนการจัดหมวดหมู่เอกสารแบบอัตโนมัติ

1.5.4 อัลกอริธึมใหม่ที่ได้สามารถพัฒนาต่อเพื่อให้สามารถครอบคลุมข้อมูลที่มีคุณสมบัติเป็น symbolic data ได้

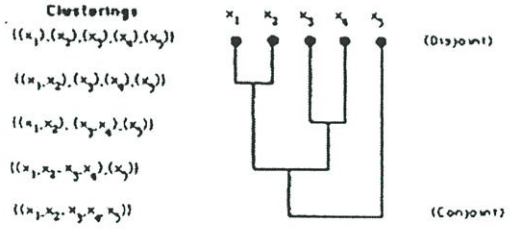
บทที่ 2
ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

2.1 วิธีการจัดกลุ่มข้อมูล(Clustering Methods)

การจัดกลุ่มข้อมูล(Clustering) เป็นวิธีการในการแยกข้อมูลออกเป็นกลุ่มย่อย(cluster) ตามลักษณะที่เหมือนกันของข้อมูล โดยข้อมูลที่อยู่ในกลุ่มเดียวกันจะมีความเหมือนกันมากกว่าข้อมูลที่อยู่ต่างกลุ่มกัน เราสามารถแบ่ง clustering methods ออกเป็น 4 ประเภทใหญ่ๆคือ hierarchical clustering, partitional clustering ,fuzzy clustering ,Self-Organizing Map Clustering

2.1.1 Hierarchical clustering

Hierarchical clustering เป็นการแบ่งข้อมูลออกเป็นส่วนๆ โดยสามารถแสดงผลลัพธ์ด้วยแผนภาพโครงสร้างแบบต้นไม้ "dendrogram" ดังรูปที่ 2.1 ซึ่งช่วยสร้างภาพที่เข้าใจได้ง่ายของ hierarchical clustering โดย dendrogram ประกอบด้วยชั้นของ nodes ซึ่งแสดงถึงการ clustering ในชั้นนั้นๆ แต่ละ cluster เส้นที่เชื่อมระหว่าง node แสดงถึงการรวมกันของ clusters เป็น cluster ใหม่อีก cluster หนึ่ง และถ้าเราตัดแผนภาพ dendrogram ตามขวางในแต่ละระดับชั้น เราจะได้ผลของการทำ clustering ในระดับชั้นนั้นๆ วิธีที่นิยมใช้ในการทำ Hierarchical Clustering ได้แก่ วิธีแบบ agglomerative และ divisive

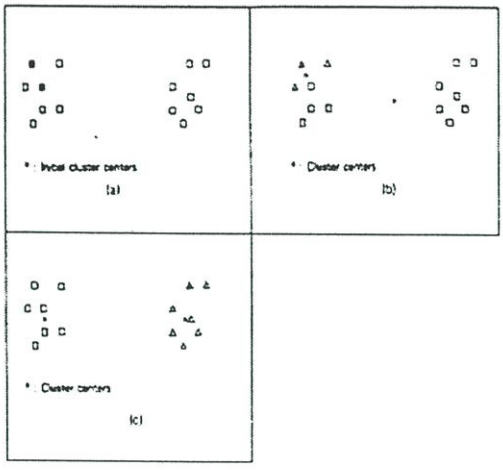


รูปที่ 2.1 แสดงตัวอย่างแผนภาพ dendrogram

2.1.2 Partitional clustering

ในทางตรงกันข้ามกับวิธีแบบ hierarchical clustering Partitional clustering ไม่ได้มีส่วนเกี่ยวข้องกับการสร้างโครงสร้างแบบต้นไม้(dendogram) เป็นเพียงการกำหนดวัตถุหรือข้อมูลที่เราสนใจลงใน cluster โดยที่ต้องระบุจำนวนของ cluster ที่ต้องการสร้างขึ้นไว้ก่อน ดังนั้น clustering แบบ 6 cluster ไม่ได้เกิดจากการรวมกันของ cluster 2 cluster จาก clusteringแบบ 7 cluster หลักการทำงานคร่าวๆ ในขั้นแรก เริ่มจากการกำหนด cluster seed ซึ่งเป็นจุดศูนย์กลางของ cluster จากนั้น

วัตถุหรือข้อมูลที่เราสนใจที่มีระยะห่างจาก cluster seed ไม่เกินจากที่กำหนดจะถูกระบุให้เป็นสมาชิกของ cluster นั้น ต่อมาก็กำหนด cluster seed ขึ้นมาอีกและทำการกำหนดสมาชิกให้กับ cluster ที่เกิดขึ้น ทำไปเรื่อยๆจนกระทั่งวัตถุหรือข้อมูลที่เราสนใจทั้งหมดเป็นสมาชิกของ cluster แล้ว วัตถุหรือข้อมูลที่เราสนใจนั้นสามารถเปลี่ยนจากสมาชิกของ cluster หนึ่งไปยังอีก cluster หนึ่งได้ถ้าวัตถุหรือข้อมูลที่เราสนใจนั้นอยู่ใกล้กับ cluster อื่นมากกว่า cluster เดิมที่มันเคยอยู่ โดยหลังจากที่ย้ายไปแล้ว จะมีการคำนวณจุดที่เป็นศูนย์กลางของ cluster นั้นใหม่และทำไปเรื่อยๆจนกว่าจะไม่มีการเปลี่ยนแปลงการเป็นสมาชิกของวัตถุหรือข้อมูลที่เราสนใจในแต่ละ cluster อีก วิธีที่นิยมใช้ในการทำ Partitional clustering ได้แก่ square-error clustering, clustering by graph theory และ nearest-neighbor clustering ตัวอย่างการทำงานของ partitional clustering : K-mean clustering แสดงได้ดังรูปที่ 2.2

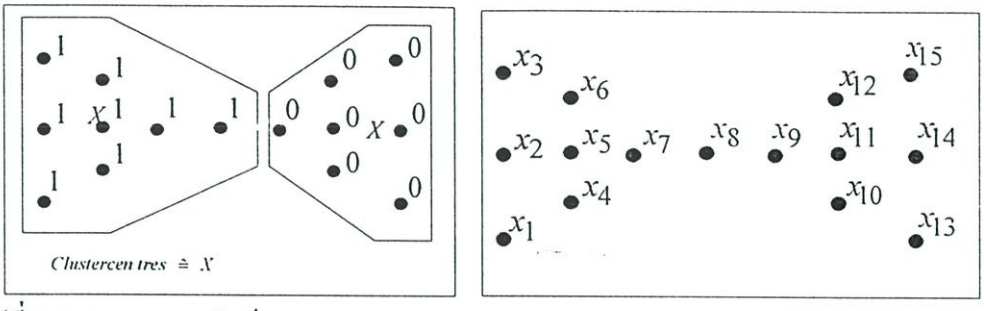


รูปที่ 2.2 แสดงการทำงานของ K-mean clustering : (a) initial data; (b) cluster membership after first loop; (c) cluster membership after second loop [1]

2.1.3 Fuzzy clustering

ในการทำ clustering ด้วย 2 วิธีข้างต้น cluster แต่ละ cluster มีสมาชิกแบ่งแยกกันอย่างชัดเจน สมาชิกใน cluster หนึ่งจะต้องเป็นสมาชิกของ cluster นั้นเพียง cluster เดียวเท่านั้น ซึ่งการแบ่งแยกเช่นนี้บางครั้งทำให้เราสูญเสียความหมายของข้อมูลที่มี ประโยชน์ไป ดังเช่น ถ้าเรากำหนดระดับของความสูงเป็น 2 ระดับ คือ สูงและทั่วไป โดยให้ระดับสูงคือผู้ที่มีความสูงมากกว่า 170 cm. ขึ้นไปและระดับทั่วไปคือผู้ที่มีความสูงต่ำกว่า 170 cm. ถ้ามีคนๆหนึ่งสูง 169 cm. เขาจะถูกจัดอยู่ในกลุ่มระดับทั่วไปทันที ทั้งๆที่ถ้ามองจากสภาพความเป็นจริงแล้วความสูง 169 cm. ถือว่าใกล้เคียงกับระดับสูงมาก แต่ด้วยการ clustering ตามวิธีข้างต้น เราจะต้องสูญเสียความหมายส่วนนั้นไป fuzzy clustering ถูกนำมาใช้แก้ปัญหาลักษณะนี้ โดยนำเอาคุณสมบัติของ fuzzy set มาใช้ ซึ่งตัว fuzzy set

จะเพิ่มข้อมูลในส่วนของ ค่าความเป็นสมาชิก (degree of membership) เข้ามา วิธีที่นิยมใช้ในการทำ Fuzzy clustering ได้แก่ fuzzy C-mean clustering ตัวอย่างเปรียบเทียบผลลัพธ์ในการ clustering แบบดั้งเดิมกับแบบ fuzzy clustering แสดงได้ดังรูปที่ 2.3

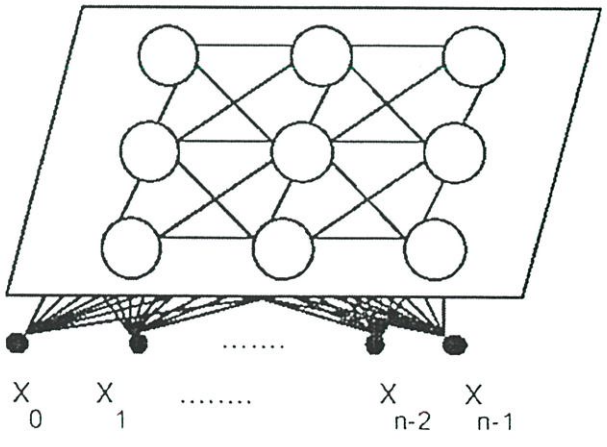


รูปที่ 2.3(a) แสดงผลลัพธ์ของการ clustering แบบ crisp

รูปที่ 2.3 (b) แสดงผลลัพธ์ของการ clustering แบบ fuzzy clustering [2]

2.1.4 Self-Organizing Maps Clustering

Self-Organizing Maps Clustering เป็นเทคนิคที่ใช้นิวรอลเน็ตเวิร์คในการแบ่งกลุ่ม cluster โดยมีข้อดีคือสามารถแบ่งกลุ่มของข้อมูลที่มีการกระจายของกลุ่มจำนวนมากแต่มีจำนวนข้อมูลน้อยๆได้ จุดหนึ่งที่ไม่เหมือนกับการ clustering ในแบบอื่นๆของ Self-Organizing Feature Maps(SOMs) คือการเปรียบเทียบความต่างกันของข้อมูล โดย SOMs ไม่ได้ใช้การเปรียบเทียบระหว่างข้อมูลด้วยกันเองแต่เป็นการเปรียบเทียบข้อมูลกับ neural output โครงสร้างการทำงานของ Self-Organizing Maps Clustering แสดงได้ดังรูปที่ 2.. 4

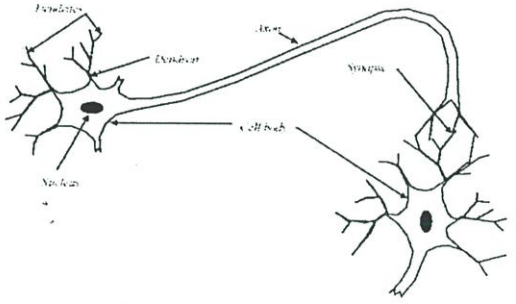


รูปที่ 2.4 แสดงโครงสร้างการทำงานของ Self-Organizing Maps Clustering

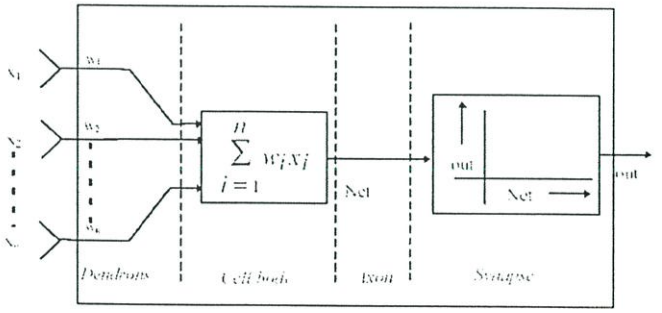
ในการทำงานของ SOMs จะเริ่มจากการหา neural output โหนดที่ใกล้เคียงกับข้อมูลที่เข้ามามากที่สุด จากนั้นจึงทำการปรับค่าของนิวรอลโหนดนั้นให้ใกล้เคียงกับข้อมูลเข้ามากขึ้น ซึ่งเมื่อทำไปเรื่อยๆ จนถึงจุดๆหนึ่ง นิวรอลโหนดในชั้น output ของ SOMs ก็จะเป็นตัวแทนของข้องมูลในแต่ละ cluster โดยการทำงานอย่างละเอียดของ SOMs จะกล่าวถึงในหัวข้อต่อไป

2.2 โครงข่ายประสาทเทียม (Neural Network)

โดยทั่วไปแล้ว โครงสร้างตามธรรมชาติของ nervous system (ระบบเส้นประสาท)ของมนุษย์ก็คือโครงข่ายประสาทที่ซับซ้อนมากๆนั่นเอง โดยมีสมองเป็นจุดศูนย์กลางการทำงาน nervous system ของมนุษย์ ประกอบด้วยเซลล์ประสาทจำนวน 10^{10} เซลล์ประสาท ซึ่งจะเชื่อมต่อกับเซลล์ประสาทอื่นๆทางโครงข่ายย่อยๆ แต่ละเซลล์ประสาทในสมองจะประกอบด้วยส่วน body , Axon(แกนของเซลล์ประสาทที่นำส่งกระแสประสาท), และ dendrites(ส่วนของเซลล์ประสาทที่มีลักษณะคล้ายกิ่ง) จำนวนมาก ดังแสดงในรูปที่ 2.5



รูปที่ 2.5(a) แสดงลักษณะของเซลล์ประสาท



รูปที่ 2.5(b) แสดงลักษณะของเซลล์ประสาทเทียม

การทำงานของเซลล์ประสาทประกอบด้วย Axon ซึ่งนำส่งกระแสสัญญาณเอาต์พุต โดยเอาต์พุตนี้จะส่งสัญญาณจาก neural หนึ่งไปยังอีก neural หนึ่ง ผ่าน Axon โดยที่จะมีเซลล์ประสาทรับสัญญาณเรียกว่า dendrites เป็นเซลล์ที่ใช้รับสัญญาณเข้ามาให้กับ neural โดย neural เป็นจุดกลางที่ใช้ประมวลผลเมื่อประมวลผลเสร็จ ก็ส่งเอาต์พุตออกไปให้กับ dendrites ของอีกเซลล์หนึ่ง ช่วงต่อระหว่าง Axon กับ dendrites จะมีช่องว่างเล็กๆเรียกว่า synapse(ส่วนที่มาบรรจบกันของเซลล์

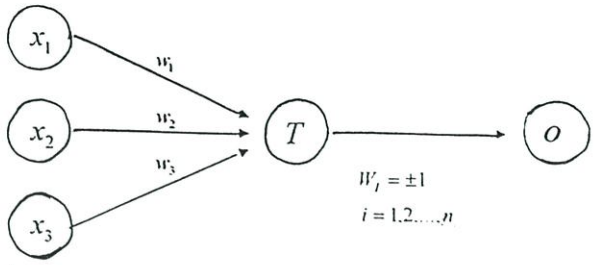
ประสาท) โดย synapse จะทำหน้าที่หลั่งสารเคมีออกมาเมื่อต้องการส่งสัญญาณ สารนี้ก็จะเข้าไปสู่ dendrites ของอีก neural หนึ่ง

โครงข่ายประสาทเทียมเป็นวิธีหนึ่งที่ช่วยให้คอมพิวเตอร์มีความสามารถมากขึ้น โดนเฉพาะการประมวลผลข่าวสารที่มีความยุ่งยากซับซ้อน เช่นการทำให้คอมพิวเตอร์สามารถรู้จำตัวอักษร ซึ่งต้องมีการตัดสินใจที่ต้องอาศัยความรู้และประสบการณ์ โดยถ้าใช้เทคนิคทางคณิตศาสตร์ธรรมดาในการแก้ปัญหาก็จะมีความซับซ้อนมาก แต่ถ้าใช้ระบบโครงข่ายประสาทเทียมก็จะช่วยลดความยุ่งยากลงได้มาก ซึ่งระบบที่สร้างขึ้นมานี้ถูกเรียกว่าระบบแบบจำลองโครงข่ายประสาทเทียม (Artificial Neural Network System: ANNS) ที่ทำให้คอมพิวเตอร์มีความสามารถในการเรียนรู้และตัดสินใจให้ระบบได้ โดยจะถอดแบบมาจากการทำงานของระบบสมองของมนุษย์

แบบจำลองโครงข่ายประสาทเทียมที่ใช้ในการประมวลผลโดยเครื่องคอมพิวเตอร์เพื่อนำไปใช้ควบคุมรักษาสมดุลต่างๆนั้น ระบบแรกถูกนำเสนอโดย McCulloch และ Pitt ในปีค.ศ. 1943 ซึ่งเป็นแบบจำลองของเซลล์ประสาทดังรูปที่ 2.6 อินพุต x_i (สำหรับ $i = 1, 2\ldots\ldots, n$) จะมีค่าเป็น {0, 1} ซึ่งจะขึ้นอยู่กับสัญญาณอินพุตจากเซลล์อื่นในขณะนั้นว่าจะมีหรือไม่มีสัญญาณ ส่วนสัญญาณที่จะส่งต่อไปยังเซลล์ถัดไปซึ่งเป็นเซลล์ผลลัพธ์ (จะแทนด้วย o) และ Firing Level ของแบบจำลองนี้ถูกกำหนดโดย

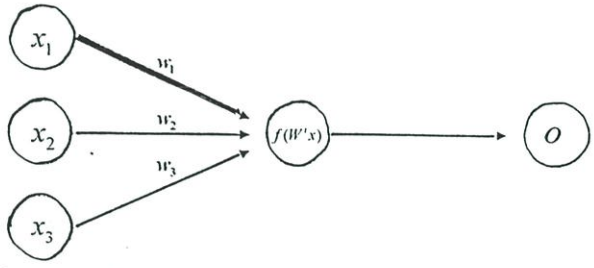
$$o^{k+1} = \begin{cases} 1 & if \sum_{i=1}^{n} w_i x_i^k \geq T \\ 0 & Otherwise \end{cases}$$

โดยที่ $k = 0,1,2,\ldots$ เป็นช่วงเวลาแบบไม่ต่อเนื่อง w_i เป็นค่าถ่วงน้ำหนัก ที่เชื่อมต่อกับอินพุตที่ I ซึ่งถ้า $w_i = +1$ แสดงถึงการกระตุ้นของ synapse และถ้า $w_i = -1$ synapse จะยับยั้งการส่งผ่านสัญญาณ และ T เป็นค่าความต่างศักดิ์เทรสโฮลด์หรือขีดเริ่มเปลี่ยนซึ่งถ้าค่าผลรวมของผลคูณระหว่างค่าถ่วงน้ำหนักกับสัญญาณอินพุตจะต้องมากกว่า T จึงจะมีสัญญาณผ่านไปยังเซลล์อื่นได้



รูปที่ 2.6 แบบจำลองเซลล์ประสาทของ McCulloch-Pitts

โครงข่ายอีกแบบหนึ่งซึ่งคล้ายกับแบบจำลองของ McCulloch-Pitts แต่แตกต่างกันตรงที่ค่าของตัวแปรต่างๆที่ใช้ในแบบจำลองโครงข่ายประสาทเทียม เป็นเลขจำนวนจริงและมีค่าถ่วงน้ำหนักจะได้จากการเรียนรู้ของระบบ ซึ่งแบบจำลองนี้แสดงในรูปที่ 2.7



รูปที่ 2.7 แบบจำลองเซลล์ประสาทเทียม

จากรูปที่ 2.7 แสดงโครงข่ายการเชื่อมต่อของแบบจำลองเซลล์ประสาทที่สามารถสอนให้โครงข่ายตัดสินใจได้ โดยมี x_i เป็นสัญญาณอินพุต และ w เป็นค่าถ่วงน้ำหนักที่ได้จากการสอนโครงข่าย และแต่ละโนดในโครงข่ายจะใช้แทนเซลล์ประสาทแต่ละเซลล์ ซึ่งบางครั้งจะเรียกว่าหน่วยประมวลผลพื้นฐาน (Process Element Unit) และมี synapse ซึ่งจะเชื่อมต่อโนดเพื่อใช้ในการส่งสัญญาณการกระตุ้นหรือยับยั้งสัญญาณจะขึ้นอยู่กับค่าถ่วงน้ำหนัก w_i และสำหรับสัญญาณเอาต์พุตสามารถคำนวณได้ดังนี้

$$o = f(W'x)$$

โดยที่ W เป็นเวคเตอร์ของค่าถ่วงน้ำหนักซึ่งสามารถกำหนดได้ดังนี้

$$W = [w_1, w_2, \ldots, w_n]'$$

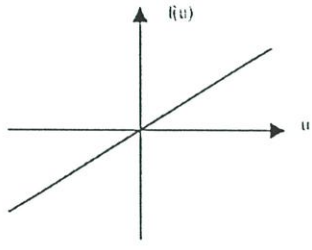
และ X เป็นเวกเตอร์อินพุต $X = [x_1, x_2, ..., x_n]'$ เมื่อ t เป็นตัวดำเนินการทรานสโพสต์ (Transpose)ของเมตริกซ์ ฟังก์ชันกำหนดสัญญาณเอาต์พุตในสมการที่ต่อไปนี้ถูกเรียกว่าฟังก์ชันการเร่งเร้าหรือแอกติเวชันฟังก์ชัน (Activation Function) และกำหนดให้

$$net = W'X = \sum_{i=1}^{n} w_i x_i$$

ฟังก์ชันการเร่งเร้ามีคุณสมบัติคล้ายกับกราฟของศักย์ไฟฟ้าขณะทำงาน มีด้วยกันสองชนิด คือ ชนิดที่เป็นเชิงเส้นและชนิดที่ไม่เป็นเชิงเส้น การเลือกใช้ฟังก์ชันการเร่งเร้าใดใน ANNS จะต้องพิจารณาให้เหมาะสมกับปัญหานั้นๆ แอกติเวชันฟังก์ชันที่นิยมใช้กันประกอบด้วย

1. Linear Function

   Linear function เป็นฟังก์ชันหนึ่งที่นิยมใช้กันมาก กราฟของฟังก์ชันสามารถแสดงดังรูปที่ 2.8



รูปที่ 2.8 Linear Activation Function

สมการทางคณิตศาสตร์สำหรับฟังก์ชันนี้สามารถเขียนได้ดังนี้

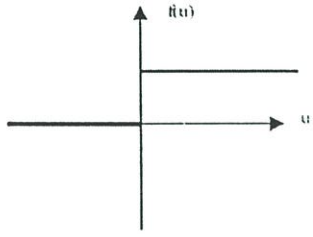
$$y = f(u) = \alpha.u$$

โดย α เป็นค่าความชัน (slope) ของ linear function ถ้าค่าความชัน α เป็น 1 แล้ว linear activation function นี้จะถูกเรียกว่า identify function โดย output (y) ของ identify function จะเท่ากับ input function (u) ถึงแม้ว่าฟังก์ชันนี้อาจดูเป็นกรณีที่ไม่มีสาระนัก แต่กระนั้นมันก็มีประโยชน์อย่างยิ่งในบางกรณีเช่น ในขั้นตอนสุดท้ายของ multilayer neural network

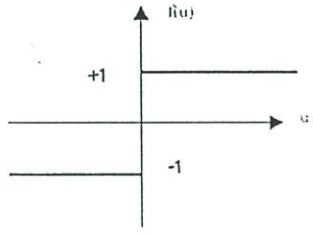
2. Threshold Function

Threshold (hard-limiter) activation function เป็นฟังก์ชันที่ให้ผลลัพธ์เป็นแบบ binary (ระบบเลขฐาน 2 คือ 0 และ 1) หรือ bipolar (มี 2 ขั้ว) ดังแสดงในรูปที่ 2.9 และ 2.10 ตามลำดับ ผลลัพธ์ของแบบ binary threshold function สามารถเขียนได้ดังนี้

$$y = f(u) = \begin{cases} 0 & if \quad u < 0 \\ 1 & if \quad u \geq 0 \end{cases}$$



รูปที่ 2.9 Binary Threshold Activation Function



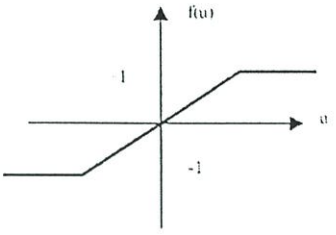
รูปที่ 2.10 Bipolar Threshold Activation Function

โครงข่ายประสาทเทียมที่ใช้ฟังก์ชันการเร่งเร้าแบบนี้ได้แก่ แบบจำลองของ McCulloch-Pitts

3. Piecewise Linear Function

ฟังก์ชันการเร่งเร้าแบบนี้ถูกเรียกอีกอย่างว่า saturating linear function และสามารถให้ค่าได้ทั้งแบบ binary หรือ bipolar อย่างใดอย่างหนึ่ง สมการทางคณิตศาสตร์ของ symmetric saturation function (รูปที่ 2.11) แสดงได้ดังนี้

$$y = f(u) = \begin{cases} -1 & if \quad u < -1 \\ u & if \quad -1 \leq u \leq 1 \\ 1 & if \quad u \geq 1 \end{cases}$$



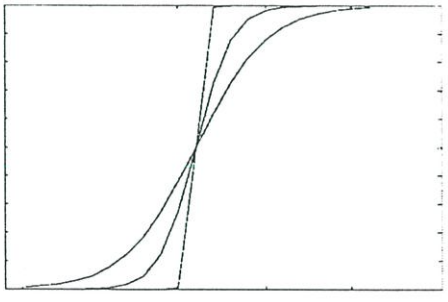
รูปที่ 2.11 Piecewise Linear Activation Function

4. Sigmoidal (S Shaped) Function

ฟังก์ชันแบบไม่เชิงเส้นนี้เป็นฟังก์ชันการเร่งเร้าที่ใช้โดยทั่วไปในการสร้างโครงข่ายประสาทเทียม เนื่องจากมีลักษณะที่ดีในเชิงคณิตศาสตร์ สามารถคำนวณความเปลี่ยนแปลงได้และเป็นฟังก์ชันเพิ่ม(increasing function)ที่ดี sigmoidal transfer function(ฟังก์ชันการแปลงแบบ Sigmoid) สามารถเขียนในรูปสมการเชิงคณิตศาสตร์ได้ดังนี้

$$f(x) = \frac{1}{1+e^{-\alpha x}}, \quad 0 \le f(x) \le 1$$

โดย α เป็น shape parameter ของ sigmoid function โดยการกำหนดค่าต่างๆของ shape parameter เราจะได้รูปกราฟดังแสดงในรูปที่ 2.12

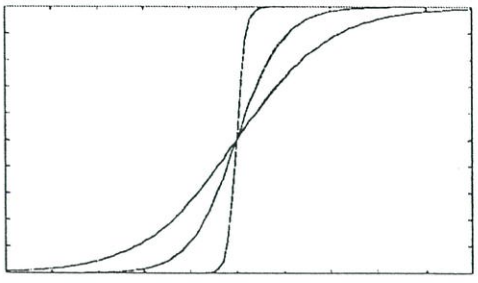


รูปที่ 2.12 A Sigmoid Activation Function

5. Tangent hyperbolic Function

ฟังก์ชันการแปลง(transfer function)แบบ Tangent hyperbolic Function (รูปที่ 2.13) นี้สามารถเขียนในรูปสมการทางคณิตศาสตร์ได้ดังนี้

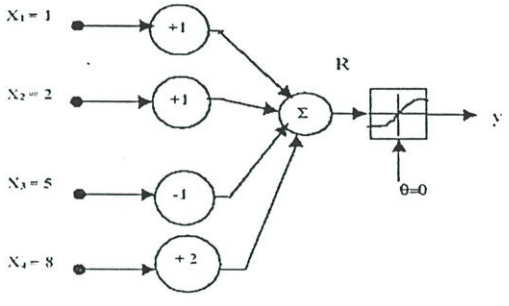
$$f(x) = \frac{e^{\alpha x} - e^{-\alpha x}}{e^{\alpha x} + e^{-\alpha x}} \qquad -1 \le f(x) \le 1$$



รูปที่ 2.13 Tangent Hyperbolic Activation Function

ตัวอย่างการทำงานของ activation function

โครงข่ายประสาทเทียมนี้ประกอบด้วยโหนดข้อมูลเข้าจำนวน 4 โหนดและมีค่าถ่วงน้ำหนัก (weight) ดังรูป ที่ 2.14



รูปที่ 2.14 Neuron Structure

ผลลัพธ์ R ของโครงข่ายก่อนที่จะผ่านขั้นตอนของ activation function สามารถคำนวณได้ดังนี้

$$R = W^T.X = [1 \quad 1 \quad -1 \quad 2].\begin{bmatrix}1\\2\\5\\8\end{bmatrix} = 14$$

ด้วย binary activation function และ sigmoid function ผลลัพธ์ของโครงข่ายประสาทเทียมนี้จะเป็นไปตามลำดับดังนี้

$$y(Threshold) = 1$$

$$y(Sigmoid) = 1.5 * 2^{-8}$$

2.3 การเรียนรู้ของโครงข่ายประสาทเทียม

การเรียนรู้ของโครงข่ายประสาทเทียมจะมีประสิทธิภาพเพียงใดขึ้นอยู่กับค่าถ่วงน้ำหนักของโครงข่าย ซึ่งการฝึกสอน (training) โครงข่าย ก็คือการหาค่าถ่วงน้ำหนักที่เหมาะสมให้แก่โครงข่ายนั้นๆ วิธีการสอนโครงข่ายประสาทเทียม มีดังนี้

2.3.1 การเรียนรู้แบบชี้นำ (Supervised Learning)

การสอน โดยวิธีนี้จะกำหนดเซตของการสอนให้กับโครงข่าย ซึ่งเซตนี้ประกอบด้วยอินพุตและเอาต์พุตที่ต้องการ เมื่อป้อนอินพุตให้กับโครงข่าย โครงข่ายจะมีการประมวลผลจนได้คำตอบและค่าถ่วงน้ำหนักออกมาชุดหนึ่ง สำหรับคำตอบที่ได้จากโครงข่ายจะถูกนำมาคำนวณค่าความผิดพลาดโดยวัดเป็นระยะทางว่ามีความห่างจากคำตอบที่ต้องการของอินพุตในชุดเดียวกันมากน้อยเพียงใด ถ้ายังมีความผิดพลาดสูงอยู่ก็จะมีการปรับค่าถ่วงน้ำหนัก และทำการสอนต่อไปจนกว่าค่าความผิดพลาดระหว่างคำตอบโครงข่ายกับเอาต์พุตที่ต้องการมีค่าน้อยพอที่จะยอมรับได้จึงจะหยุดการสอน และค่าถ่วงน้ำหนักที่ได้จะเป็นเหมือนฟังก์ชันที่ใช้ในการแปลงข้อมูล ตัวอย่างโครงข่ายที่มีการเรียนรู้แบบ Supervised Learning ได้แก่ Perceptrons, Radial-Basis Function Networks เป็นต้น

2.3.2 การเรียนรู้แบบไม่มีการชี้นำ (Unsupervised Learning)

การสอนโดยวิธีนี้จะป้อนอินพุตเข้าสู่โครงข่ายและภายในโครงข่ายจะมีเอาต์พุตโนดอยู่หลายโนดด้วยกัน โดยแต่ละโนดแทนกลุ่มของข้อมูลที่มีคุณสมบัติเหมือนกัน เมื่อป้อนอินพุตเข้าสู่โครงข่าย โครงข่ายจะคำนวณค่าความสัมพันธ์ที่มีอยู่ภายในเซตของอินพุตโดยอาศัยค่าถ่วงน้ำหนักเป็นตัวแยกความแตกต่างของอินพุตไปเก็บไว้ในโนดเอาต์พุตของโครงข่าย การสอนโดยวิธีนี้ จะไม่สามารถระบุได้ว่าเอาต์พุตโนดใดเป็นของข้อมูลกลุ่มไหน ผู้ใช้จะต้องกำหนดเอง ซึ่งแตกต่างจากการสอนแบบชี้นำที่โครงข่ายสามารถระบุกลุ่มเอาต์พุตได้อย่างแน่นอน ตัวอย่างโครงข่ายที่มีการเรียนรู้แบบ Unsupervised Learning ได้แก่ Self-Organizing Feature Maps (SOMs), ART1,ART2 เป็นต้น

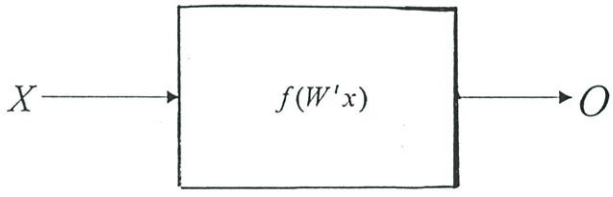
การสอนโครงข่ายเซลล์ประสาทเป็นการหาฟังก์ชันการแปลงและฟังก์ชันการแปลงที่ได้จะมีคุณสมบัติไม่เป็นเชิงเส้น ซึ่งฟังก์ชันการแปลงของโครงข่ายประสาทเทียมในที่นี้คือเซตของค่าถ่วงน้ำหนักของโครงข่าย ดังนั้นฟังก์ชันการแปลงจะมีศักยภาพมากน้อยเพียงใดนั้นจะขึ้นอยู่กับค่าถ่วงน้ำหนักของโครงข่ายนั้นๆว่ามีเสถียรภาพมากน้อยเพียงใดและค่าถ่วงน้ำหนักคำนวณได้จากการสอนโครงข่าย ซึ่งการสอนโครงข่ายมีหลายแบบด้วยกัน เช่น กฎการสอนของ Hebb(Hebbian Learning) กฎการสอนแบบเดลต้า(Delta Rule or Error-Correction Learning) กฎการเรียนรู้แบบแข่งขัน(Competitive Learning) เป็นต้น

2.4 แบบจำลองการทำงานของโครงข่ายประสาทเทียม

โครงข่ายประสาทเทียมและโครงข่ายของเซลล์ประสาทจริงของมนุษย์ จะมีการเชื่อมต่อกันของโนดในลักษณะของโครงข่ายอย่างแน่นหนา เพื่อให้โครงข่ายสามารถเรียนรู้และสามารถจดจำสิ่งที่ได้เรียนรู้มาแล้วได้ ซึ่งโครงสร้างการทำงานของโครงข่ายจะมีดังนี้

2.4.1 โครงข่ายที่ส่งสัญญาณไปข้างหน้า (Feedforward Networks)

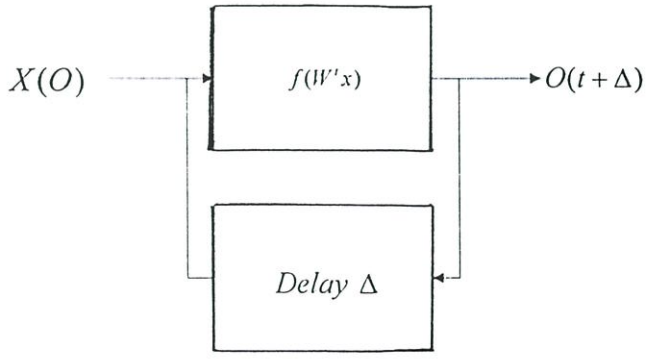
โครงข่ายชนิดนี้จะประกอบด้วยชั้นต่างๆของโครงข่าย โดยชั้นแรกจะเป็นอินพุตและชั้นสุดท้ายจะเป็นชั้นของเอาต์พุตส่วนระหว่างชั้นอินพุตและเอาต์พุตอาจจะมีหรือไม่มีชั้นที่แทรกอยู่ภายในก็ได้ ซึ่งจะขึ้นอยู่กับอัลกอริธึมที่ใช้ในการสอนโครงข่าย เช่นถ้าเป็นโครงข่าย Perceptron แบบหลายชั้น (Multilayer Perceptron) ก็จะมีชั้นที่อยู่ระหว่างระหว่างชั้นอินพุตและเอาต์พุตอีก ซึ่งอาจมีมากกว่าหนึ่งชั้นก็ได้ ส่วนโครงข่าย Self-Organizing Map ของ kohonen จะมีเพียงชั้นของอินพุตและเอาต์พุตเท่านั้น การเชื่อมต่อระหว่างโครงข่ายแบบ Feedforward จะมีค่าถ่วงน้ำหนักเป็นตัวเชื่อมและสัญญาณอินพุตที่เข้ามาจะถูกส่งไปตามทิศทางของลูกศรจนถึงชั้นเอาต์พุตโดยไม่มีการป้อนกลับ ดังรูปที่ 2.15



รูปที่ 2.15 บล็อกไดอะแกรมของโครงข่าย Feedforward

2.4.2 โครงข่ายที่มีการป้อนกลับ (Feedback Networks)

ในส่วนแรกของโครงข่ายนี้จะเป็นโครงข่าย Feedforward เหมือนกับแบบแรก และส่วนที่เพิ่มเข้ามาคือส่วนของการป้อนกลับซึ่งมีการหน่วงเวลาไปจากเวลาเดิมเท่ากับ Δ ดังรูปที่ 2.16



รูปที่ 2.16 บล็อกไดอะแกรมของโครงข่าย Feedback

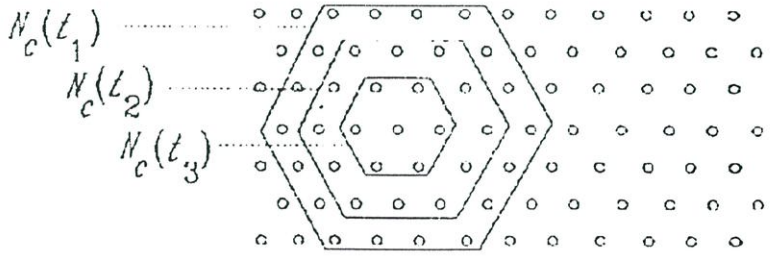
2.5 Self-Organizing feature Maps

Self-Organizing feature Maps(รูปที่ 2.4) [3],[4] ถูกนำเสนอโดย Tuevo Kohonen ในปี 1982 เป้าหมายของ algorithm นี้คือการแปลงข้อมูลที่เข้ามาให้อยู่ในรูปของตารางหนึ่งหรือสองมิติที่กำหนดไว้ โดยไม่ต้องอาศัยการสอนหรือชี้นำใดๆ

ขั้นตอนการทำงานของ algorithm สามารถแบ่งได้เป็น 3 ส่วนสำคัญคือ Competition, Cooperation และ Synaptic Adaptation โดยมีรายละเอียดดังนี้

1. Competition เป็นส่วนที่ทำการเปรียบเทียบความเหมือนกันระหว่างข้อมูลเข้ากับโนดต่างๆใน output layer ซึ่งจัดเรียงกันในรูปของตาราง 2 มิติดังรูปที่ 2.4 โดยใช้กฎการเรียนรู้แบบแข่งขัน (competitive learning) ซึ่งโนดที่มีค่าความเหมือนกันกับข้อมูลเข้ามากที่สุดจะเป็นโนดที่ได้รับการคัดเลือกเพียงโนดเดียว เราเรียกโนดนี้ว่า "wining neural"
2. Cooperation เป็นขั้นตอนในการหาโนดที่จะทำการ update ไปพร้อมๆกับ wining neural โดยผ่าน neighborhood function ซึ่ง จำนวนของ neighborhood เริ่มจากจำนวนที่ครอบคลุมแทบทุกโนดใน output layer จากนั้นจะค่อยๆลดลงตามจำนวนรอบของการเทรนนิ่ง และในรอบท้ายๆ จำนวนของ neighborhood จะเป็น 0 ซึ่งหมายความว่าการปรับค่าของ weight vector ในขั้นตอนที่ 3 จะกระทำเฉพาะ wining neural เท่านั้น ลักษณะการทำงานของ neighborhood function แสดงได้ในรูปที่ 2.17
3. Synaptic Adaptation เป็นขั้นตอนในการ update wining neural และโนดอื่นๆ ตามที่คำนวณได้จาก neighborhood function โดยการ update นี้จะทำให้โนดเหล่านี้มีค่าใกล้เคียงกับข้อมูลเข้ามากขึ้น

อัลกอริธึมของ Self-Organizing Feature Maps แสดงดังรูปที่ 2.18



รูปที่ 2.17 แสดงการทำงานของ neighborhood function เมื่อจำนวนรอบของการเทรนนิ่งเพิ่มขึ้น

1. เริ่มจากการสุ่มค่าเพื่อกำหนดเป็นค่าเริ่มต้นให้กับ weight vector $w_j(0)$. โดยมีข้อแม้ว่าค่าเริ่มต้นนี้จะต้องไม่ซ้ำกันสำหรับ J= 1,2,3....N โดย N เป็นจำนวนของ neural ใน lattice วิธีหนึ่งในการกำหนดค่าของ $w_j(0)$ คือการสุ่มค่า $w_j(0)$ จากข้อมูลเข้า
2. ทำการเลือกข้อมูลเข้า x ด้วยการสุ่ม
3. เปรียบเทียบข้อมูลเข้า x เพื่อหา winning neuron $i(x)$ โดยใช้ minimum-distance Euclidean criterion:

$$i(x) = arg_j\ min\|x(n) - w_j\|, \qquad j = 1,2,\ldots,N$$

4. ปรับค่า synaptic weight vectors โดยใช้

$$w_j(n+1) = \begin{cases} w_j(n) + \eta(n)[x(n) - w_j(n)], & j \in \wedge_{i(x)}(n) \\ w_j(n) & otherwise \end{cases}$$

โดย $\eta(n)$ เป็นค่า learning-rate และ $\wedge_{i(x)}(n)$ เป็น neighborhood function ซึ่งใช้ในการหา neighborhood ของwining neuron $i(x)$ ทั้ง $\eta(n)$ และ $\wedge_{i(x)}(n)$ จะมีค่าเปลี่ยนแปลงไปในแต่ละรอบของการเรียนรู้ ทั้งนี้เพื่อผลลัพธ์ที่ดี

5. เริ่มทำตั้งแต่ข้อ 2 ใหม่จนกว่าจะไม่มีความเปลี่ยนแปลงที่มากเกินกว่าค่าที่ยอมรับได้ใน feature map เกิดขึ้น

รูปที่ 2.18 อัลกอริธึมของ Self-Organizing Feature Maps

2.6 Conscience Algorithm

แนวคิดของ Conscience algorithm ซึ่งถูกนำเสนอโดย Duane Desieno [6] คือการกระจายโอกาสการชนะการแข่งขันในกระบวนการเรียนรู้แบบ competitive learning ให้กับทุกๆโนดของ output neural ใน feature map อย่างเท่าเทียมกันมากที่สุด โดยมีหลักการดังนี้คือ

"เมื่อ neural ใดๆได้รับการเลือกให้ชนะในการแข่งขันมากครั้งเข้า neural ตัวนั้นจะค่อยๆลดโอกาสการได้รับเลือกให้ชนะของตนลงไปเรื่อยๆกล่าวคือ ยิ่งชนะมากครั้งขึ้นเท่าใด โอกาสที่จะชนะในครั้งต่อไปก็จะน้อยลงเท่านั้น"

การทำงานของ Conscience algorithm เป็นดังรูปที่ 2.19

1. Find the synaptic weight vector W_i closet to input vector X :

$$\|X - W_i\| = \min_j \|X - W_j\|, \qquad j = 1,2,\ldots,N$$

2. Keep a running total of the fraction of time, p_j ,that neuron j wins the competition:

$$p_j^{new} = p_j^{old} + B(y_i - p_j^{old})$$

where $0 < B \leq 1$ and

$$y_i = \begin{cases} 1 & if \quad neuron \quad j \quad is \quad the \quad wining \quad neuron \\ 0 & Otherwise \end{cases}$$

the p_j is a initialized to zero at the beginning of the algorithm.

3. Find the new wining neuron using the conscience mechanism

$$\|X - W_i\| = \min_j (\|X - W_j\| - b_j)$$

where b_j is a bias term introduce to modify the competition; it is defined by

$$b_j = C(\frac{1}{N} - p_j)$$

where C is a bias factor and N is the total number of neurons in the network.

รูปที่ 2.19 แสดง Conscience algorithm

2.7 การเปรียบเทียบความเหมือนกันของเอกสาร

แนวคิดการเปรียบเทียบความเหมือนกันของเอกสารที่ใช้ในงานวิจัยนี้ประยุกต์มาจากงานวิจัยของ El-Sonbaty Y.A.[10] โดยในงานวิจัยนี้ได้แบ่งการเปรียบเทียบความแตกต่างของข้อมูล symbolic data ออกเป็น 2 ส่วนคือ การเปรียบเทียบความแตกต่างของข้อมูลชนิด qualitative และการเปรียบเทียบความแตกต่างของข้อมูลชนิด quantitative ซึ่งงานวิจัยนี้นำเอาแนวคิดการเปรียบเทียบความแตกต่างของข้อมูลชนิด qualitative มาใช้ในเชิงของการเปรียบเทียบความแตกต่างของเอกสาร โดยเอกสารต่างๆประกอบด้วยคุณสมบัติที่สามารถใช้แทนตัวเอกสารนั้นๆ เช่น เอกสารฉบับหนึ่งอาจประกอบด้วยชื่อหัวข้อของเอกสาร ชื่อผู้แต่ง คำที่พบมากในเอกสารนั้นๆ เป็นต้น ซึ่งเราสามารถเขียนแทนตัวเอกสารหนึ่งเอกสารซึ่งมีคุณสมบัติ n คุณสมบัติได้ดังนี้

$Doc = D_1 * D_2 * D_3 * \ldots\ldots * D_n$ n เป็นจำนวนคุณสมบัติของเอกสาร

ตัวอย่างเช่น

$Book = Title * Author * keyword$ Title feature คือคำที่ใช้อธิบายชื่อหัวข้อเรื่องของเอกสารหรือหนังสือ

Author feature คือคำที่บ่งถึงชื่อผู้แต่ง

Keyword feature คือคำที่พบบ่อยๆในเอกสารนั้นๆ

ความแตกต่างของเอกสาร 2 เอกสาร A และ B นิยามตามแนวคิดของ El-Sonbaty [6] ได้ดังนี้

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

การเปรียบเทียบ $D(A_k, B_k)$ สามารถแบ่งเป็น 2 ส่วนย่อยคือ การเปรียบเทียบในเชิง span. $D_s(A, B)$ และ การเปรียบเทียบในเชิง content, $D_c(A, B)$ ซึ่งนิยามได้ดังนี้

การเปรียบเทียบในเชิง content

$$D_c(A_k, B_k) = \frac{|Length\ of\ A_k + Lenght\ of\ B_k - 2 * Lenght\ of\ \text{intersection of}\ A_k\ and\ B_k|}{Span\ Length\ of\ A_k\ and\ B_k}$$

การเปรียบเทียบในเชิง span

$$D_s(A_k, B_k) = \frac{|Length\ of\ A_k - Lenght\ of\ B_k|}{Span\ Length\ of\ A_k\ and\ B_k}$$

$Length\ of\ A_k$ คือ จำนวนค่าต่างๆในคุณสมบัติ A_k

$Span\ Length\ of\ A_k\ and\ B_k$ คือจำนวนของค่าต่างๆในที่เกิดในการ union ของคุณสมบัติ A_k และ B_k

$Length\ of\ Intersection\ of\ A_k\ and\ B_k$ คือจำนวนของค่าต่างๆในที่เกิดในการ intersection ของคุณสมบัติ A_k และ B_k

โดยความแตกต่างรวมของคุณสมบัติ 2 คุณสมบัติ A_k และ B_k จะหาได้จาก

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

ความแตกต่างของเอกสาร 2 เอกสาร A และ B คือ

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

ตัวอย่างการหาความแตกต่างของเอกสาร

เอกสาร Doc ประกอบด้วยคุณสมบัติ 2 คุณสมบัติดังนี้

$$Doc = Title \quad X \quad Keyword$$

ตัวอย่างข้อมูลของเอกสาร Doc เป็นดังตารางที่ 2.1

ตารางที่ 2.1 แสดงตัวอย่างข้อมูลของเอกสาร Doc

เอกสาร	Title	Keyword
Doc1	a,c,d,g	c,d,i,j
Doc2	g,k,m,n	i,m,n

ความแตกต่างของเอกสาร $Doc1$ และ $Doc2$ หาได้จาก

ผลรวมของความแตกต่างของ feature แต่ละ feature ในเอกสาร นั่นคือ

$$D(Doc1,Doc2)=D(Doc1_{Title},Doc2_{Title})+D(Doc1_{Keyword},Doc2_{Keyword})$$

โดยความแตกต่างของแต่ละ feature หาได้จากผลรวมของความแตกต่างในเชิง span และความแตกต่างในเชิง content นั่นคือ

$$D(Doc1_{Title},Doc2_{Title})=D_s(Doc1_{Title},Doc2_{Title})+D_c(Doc1_{Title},Doc2_{Title})$$

และ

$$D(Doc1_{Keyword},Doc2_{Keyword})=D_s(Doc1_{Keyword},Doc2_{Keyword})+D_c(Doc1_{Keyword},Doc2_{Keyword})$$

ดังนั้น

$$D(Doc1_{Title},Doc2_{Title})=\frac{|4-4|}{7}+\frac{|4+4-2*(1)|}{7}=\frac{6}{7}$$

และ

$$D(Doc1_{Keyword},Doc2_{Keyword})=\frac{|4-3|}{6}+\frac{|4+3-2*(1)|}{6}=1$$

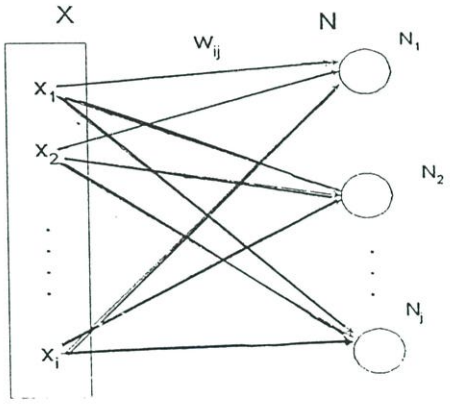
ดังนั้น

$$D(Doc1, Doc2) = 1.857$$

โครงสร้างการทำงานของโครงข่ายประสาทเทียมแบบ Self-Organizing Feature Maps(SOMs) และแนวคิดเรื่องการเปรียบเทียบความเหมือนกันของเอกสารรวมทั้งกลไกการทำงานของ conscienсve algorithm จะถูกนำมาใช้ในการทำงานของ Text Processing Kohonen Neural Networks(TPKNN) ซึ่งจะกล่าวถึงในรายละเอียดในบทที่ 3

บทที่ 3
เท็กโปรเซสซิ่งโคโฮเนนนิวรอลเน็ตเวิร์ค

เท็กโปรเซสซิ่งโคโฮเนนนิวรอลเน็ตเวิร์ค(Text Processing Kohonen Neural Networks) เป็น นิวรอลเน็ตเวิร์คที่ปรับปรุงมาจาก kohonen self-organizing feature map โดยนำกลไกการทำงานของ conscience algorithm และแนวคิดเกี่ยวกับการเปรียบเทียบความแตกต่างระหว่าง symbolic object มาประยุกต์เพื่อให้นิวรอลเน็ตเวิร์คนี้สามารถรับข้อมูลที่เป็นข้อความได้โดยตรง ทำให้สามารถลดขั้นตอนในการปรับเปลี่ยนข้อมูลให้เป็นข้อมูลเชิงปริมาณแต่ยังคงสามารถรักษาความหมายของข้อมูลให้คงเดิมอยู่ได้ โครงสร้างของ นิวรอลเน็ตเวิร์คนี้แสดงได้ดังรูปที่ 3.1



รูปที่ 3.1 แสดงโครงสร้างการทำงานของ Text Processing Kohonen Neural Networks

3.1 เท็กโปรเซสซิ่งโคโฮเนนนิวรอลเน็ตเวิร์ค(Text Processing Kohonen Neural Networks)

Text Processing Kohonen Neural Networks ซึ่งต่อไปนี้จะเรียกย่อว่า TPKNN ประกอบด้วยสองส่วนหลักคือส่วนของ input unit X นิยามได้ดังนี้

$$X = (x_1, x_2, ..., x_d)^T$$

d เป็นจำนวนคุณสมบัติของ input unit X และส่วนของ output unit นิยามได้ดังนี้

$$W_i = \{w_{i1}, w_{i2}, ..., w_{ik}\}$$

w_{ik} = เป็นค่า Weight vector ของ neural W_i คุณสมบัติที่ k
ข้อมูลเข้า X ใน input unit ซึ่งเชื่อมต่ออย่างสมบูรณ์กับโหนดทุกโหนดใน output unit W เส้นที่เชื่อมระหว่าง input unit X กับ output unit W คือ weight vector ซึ่งนิยามได้ดังนี้

$$w_{ik} = \{(A_{ik1}, e_{ik1}), (A_{ik2}, e_{ik2}), ... (A_{ikp}, e_{ikp})\}$$

A_{ikp} คือค่า qualitative value ของ weight w_{ik}
e_{ikp} คือ ค่าแสดงความสมาชิกของ A_{ikp} e_{ikp} มีค่าระหว่าง 0 ถึง 1 โดย $e_{ikp} = 0$ ถ้า qualitative value A_{ikp} ไม่ได้เป็นส่วนหนึ่งของข้อมูลเข้า i เลย ในขณะที่ $e_{ikp} = 1$ ถ้า qualitative value A_{ikp} เป็นสมาชิกของข้อมูลเข้า i อย่างสมบูรณ์

3.2 กระบวนการเรียนรู้ของอัลกอริธึม

กระบวนการเรียนรู้ของ TPKNN เป็นการขยายความสามารถของ Kohonen Self-Organizing Map ในส่วนของ competitive learning โดยเพิ่มเติมแนวคิดของ conscience algorithm และแนวคิดเกี่ยวกับการเปรียบเทียบคุณสมบัติที่มีค่าของข้อมูลเป็นแบบ qualitative เข้าไป ซึ่งทำให้อัลกอริธึมนี้สามารถจัดการกับข้อมูลแบบ qualitative ได้โดยตรง ขั้นตอนการทำงานของอัลกอริธึมสามารถแสดงได้ดังรูปที่ 3.2

3.3 การทำงานของ TPKNN algorithm

การทำงานของ TPKNN algorithm ประกอบด้วยส่วนสำคัญ 2 ส่วน คือ

3.3.1 ส่วนของการทำ competitive learning

เป็นส่วนที่ประยุกต์แนวคิดของ conscience algorithm และแนวคิดเรื่องการหาค่าความแตกต่างของเอกสารเข้ากับกระบวนการหา wining neural ของ competitive learning โดยประกอบด้วยขั้นตอนสำคัญ 3 ขั้นตอนดังนี้คือ

ขั้นตอนที่ 1 เป็นหา neural output W_i ที่มีความเหมือนกันกับข้อมูลเข้า input X มากที่สุด โดยการหาความแตกต่างรวมของ feature แต่ละ feature ของ input X กับ neural output W_i โดยความแตกต่างของแต่ละ feature จะหาได้จากผลรวมของความแตกต่างในเชิง span และความแตกต่างในเชิง content

$$\|X - W_i\| = \sum_{k=1}^{d} D(x_k, w_{ik}) * E$$

1: Initialize weight w_{ik} in each neural output W_i. Each weight can be initializing from the training data arbitrarily.

$W_i = \{w_{i1}, w_{i2}, \ldots, w_{ik}\}$

$w_{ik} = \{(A_{ik1}, e_{ik1}), (A_{ik2}, e_{ik2}), \ldots, (A_{ikp}, e_{ikp})\}$

w_{ik} = Weight of neural W_i feature k

A_{ikp} = Member p^{th} of w_{ik}

e_{ikp} = Degree of membership of A_{ikp}

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

3: Draw a sample X from the input distribution with a certain probability. For each input vector $X = (x_1, x_2, \ldots, x_d)^T$, Do step 4-6

4: For each output unit 'i', Compute

$$\|X - W_i\| = \sum_{k=1}^{d} D(x_k, w_{ik}) * E$$

d = number of feature and E is defined by

$$E = 1 - \frac{\sum e_{ikp}}{D}$$

where $A_{ikp} \in w_{ik} \cap x_k$, e_{ikp} is degree of membership of A_{ikp} and $D = \sum e_{ik}$

Finds index 'i' such that $\|X - W_i\|$ is minimum.

รูปที่ 3.2 แสดงอัลกอริธึมของ TPKNN

ซึ่งการหาค่าความแตกต่างของ feature x_k กับ weight vector w_{ik} หาได้จาก

$$D(x_k, w_{ik}) = D_s(x_k, w_{ik}) + D_c(x_k, w_{ik})$$

โดยค่าความแตกต่างในเชิง span ของ feature x_k กับ weight vector w_{ik} หาได้จาก

$$D_s(x_k, w_{ik}) = \frac{|Length\ \ of\ \ x_k\ \ - Lenght\ \ of\ \ w_{ik}|}{Span\ \ Lengtl:\ \ of\ \ x_k\ \ and\ \ w_{ik}}$$

5: Keep a running total of the fraction of time, p_i that neural W_i win the competition

$$p_i^{new} = p_i^{old} + B(y_i - p_i^{old})$$

where $0 < B \leq 1$ and

$$y_i = \begin{cases} 1 & if \quad neuron \quad i \quad is \quad wining \quad neuron \\ 0 & otherwise \end{cases}$$

6: Find the wining neuron using conscience mechanism

$$\|X - W_i\| = \min_I (\sum_{k=1}^{d} D(x_k, w_{ik}) * E - b_i)$$

where b_i is defined by

$$b_i = C(\frac{1}{N} - p_i)$$

where C is bias factor and N is total number of neurons in the network

7: For all weights that connect to the wining node I and its' neighborhood $(\wedge_I)$.

$$w_{ik}^{(new)} = \begin{cases} w_{ik}^{(old)} \cup x_k & if \quad i \in \wedge_I \\ w_{ik}^{(old)} & Otherwise \end{cases}$$

And

$$e_{ikp}^{(new)} = \begin{cases} f(e_{ikp}^{old} + \eta) & if \quad A_{ikp} \in w_{ik} \cap x_k \\ f(e_{ikp}^{old} - \eta) & if \quad A_{ikp} \notin w_{ik} \cap x_k \\ 2 * \eta & if \quad A_{ikp} \in x_j - (w_{ik} \cap x_k) \end{cases}$$

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

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

8: Continue with step 2-7 until the stopping condition is true.

รูปที่ 3.2 (ต่อ)

และค่าความแตกต่างในเชิง content ของ feature x_k กับ weight vector w_{ik} หาได้จาก

$$D_c(x_k, w_{ik}) = \frac{|Length \quad of \quad x_k \quad + Lenght \quad of \quad w_{ik} - 2 * Lenght \quad of \quad intersection \quad of \quad x_k \quad and \quad w_{ik}|}{Span \quad Length \quad of \quad x_k \quad and \quad w_{ik}}$$

neural output W_i ที่มีค่าความแตกต่างน้อยที่สุดจะเป็นโหนดที่ถูกเลือก

ขั้นตอนที่ 2 เมื่อเราได้ neural output W_i ที่มีความเหมือนกันกับข้อมูลเข้า input X มากที่สุดแล้ว ในขั้นตอนต่อไป เราจะทำการหาค่า bias term b_i โดยเริ่มจากการหาค่าสัดส่วนของการได้รับเลือก (total of the fraction of time) p_i ซึ่งนิยามได้ดังนี้

$$p_i^{new} = p_i^{old} + B(y_i - p_i^{old})$$

โดย B มีค่าระหว่าง 0 ถึง 1 และ y_i มีค่าเป็น 1 เมื่อ i เป็น index ของ neural output W_i ที่มีความเหมือนกันกับข้อมูลเข้า input X มากที่สุดและเป็น 0 ในกรณีอื่นๆ

จากค่า p_i ที่ได้ เราสามารถคำนวณหาค่า bias term b_i จาก

$$b_i = C(\frac{1}{N} - p_i)$$

โดย C เป็นค่า bias factor และ N เป็นจำนวนของ neural output ทั้งหมด

ค่า b_i ที่ได้จะถูกนำมาใช้ในการหา wining neural โดย

$$\|X - W_i\| = \min_I(\sum_{k=1}^{d} D(x_k, w_{ik}) * E - b_i)$$

โดย wining neural W_I ที่ได้จากขั้นตอนนี้รวมทั้ง neighborhood ของมัน ซึ่งหาได้จาก neighborhood function (Λ) ใน 3.3.2 จะได้รับการปรับปรุงค่าของ weight vector ให้มีความใกล้เคียงกับ input X มากขึ้น

3.3.2 ส่วนของการหา neighborhood ของ wining neural

ในส่วนนี้เป็นการหา neighborhood ของ wining neural ซึ่งจำนวนของ neighborhood จะเริ่มจากทุกๆโนดใน output layer และค่อยๆลดลงตามจำนวนรอบของการเทรนนิ่ง โดย มี neighborhood function เป็นตัวกำหนด

3.3.3 ส่วนของการ update weight vector ของ wining neural และ neighborhood

ในส่วนนี้เป็นการปรับปรุง weight vector ของ wining neural และ neighborhood ของมัน ให้มีความใกล้เคียงกับ input X มากขึ้น โดยการปรับปรุงนั้นอาจแบ่งได้เป็น 2 ส่วนย่อย คือ

1. ส่วนของการปรับปรุง w_{ik} ในส่วนของ A_{ikp}

ส่วนของการปรับปรุง w_{ik} ในส่วนของ A_{ikp} เพื่อรับสมาชิกใหม่เข้ามาใน w_{ik} โดย

$$w_{ik}^{(new)} = \begin{cases} w_{ik}^{(old)} \cup x_k & if \quad i \in \wedge_I \\ w_{ik}^{(old)} & Otherwise \end{cases}$$

2. ส่วนของการปรับปรุง w_{ik} ในส่วนของ e_{ikp}

โดยแบ่งการปรับปรุงในส่วนนี้ออกเป็น 3 กลุ่มคือ

$$e_{ikp}^{(new)} = \begin{cases} f(e_{ikp}^{old} + \eta) & if \quad A_{ikp} \in w_{ik} \cap x_k \\ f(e_{ikp}^{old} - \eta) & if \quad A_{ikp} \notin w_{ik} \cap x_k \\ 2 * \eta & if \quad A_{ikp} \in x_j - (w_{ik} \cap x_k) \end{cases}$$

กลุ่มที่ 1 เป็นกลุ่มของ e_{ikp} ที่มี A_{ikp} มีค่าซ้ำกับสมาชิกใน x_k ค่าของ e_{ikp} ในกลุ่มนี้จะถูกปรับให้เพิ่มขึ้น

กลุ่มที่ 2 เป็นกลุ่มของ e_{ikp} ที่มี A_{ikp} มีค่าไม่ซ้ำกับสมาชิกใน x_k ค่าของ e_{ikp} ในกลุ่มนี้จะถูกปรับให้ลดลง

กลุ่มที่ 3 เป็นกลุ่มของ A_{ikp} ที่เพิ่งรับเข้ามาใหม่จากขั้นตอนในส่วนแรก ค่าของ e_{ikp} จะถูกกำหนดขึ้นตามความเหมาะสม

บทที่ 4

การทดลองและผลการทดลอง

ในการทดลองเพื่อทดสอบประสิทธิภาพของ TPKNN อัลกอริธึม ซึ่งประยุกต์การทำงานของ SOMs และการหาค่าความแตกต่างระหว่างข้อมูลที่เป็น symbolic โดยใช้โปรแกรม MATLAB version 6.5 ของ บริษัท mathworks จำกัด ในการจำลองการทำงานของอัลกอริธึม โดยมีรายละเอียดของการทดลองดังนี้

4.1 การวัดประสิทธิภาพของอัลกอริธึม

ในการวัดประสิทธิภาพของอัลกอริธึม เพื่อทำความเข้าใจต่อประสิทธิภาพการทำงานของ TPKNN ในแต่ละแง่มุมนั้น ได้เลือกใช้ตัววัดที่ใช้ในการวิจัยด้าน Information retrieval คือตัววัด F-measure และตัววัด Entropy โดยตัววัด F measure เป็นตัววัดที่ใช้ในการวัดความถูกต้องของการ clustering ส่วนตัววัด Entropy เป็นตัววัดเพื่อวัดการซ้อนทับกันของกลุ่ม cluster ใน neural output โดยรายละเอียดของตัววัดทั้ง 2 มีดังนี้

4.1.1 ตัววัด F measure

เป็นตัววัดใช้วัดประสิทธิภาพของ hierarchical clustering ที่เกิดจากผลรวมของค่า 2 ค่าคือ precision และ recall โดยเราสามารถคำนวนค่า precision และ recall ของ cluster j และ class i ได้ดังนี้

$$\text{Re}call(i,j) = \frac{n_{ij}}{n_i}$$

$$\text{Pr}\ ecision(i,j) = \frac{n_{ij}}{n_j}$$

โดย n_{ij} เป็นจำนวนของสมาชิกใน *class* i ใน *cluster* j และ

n_j เป็นจำนวนของสมาชิกใน *cluster* j

n_i เป็นจำนวนของสมาชิกใน *class* i

F measure ของ cluster j และ class i สามารถหาได้จาก

$$F(i,j) = \frac{2 * \text{Re}call(i,j) * \text{Pr}\ ecision(i,j)}{\text{Re}call(i,j) + \text{Pr}\ ecision(i,i)}$$

สำหรับค่า F measure ของทุกๆ class หาได้จาก

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

โดย n เป็นจำนวนของเอกสารทั้งหมด

ค่า F measure มีค่าระหว่าง 0 ถึง 1 โดยจะมีค่ามากถ้าผลการ clustering มีคุณภาพดี

4.1.2 ตัววัด Entropy

ตัววัด Entropy ถูกใช้เพื่อวัดคุณภาพของ cluster ที่ได้ โดยค่าของ entropy จะดีที่สุดคือเป็น 0 เมื่อผลลัพธ์ของแต่ละ cluster ประกอบด้วยข้อมูลจาก class เดียวเท่านั้น โดยการหาค่า Entropy เริ่มจากการคำนวนค่า p_{ij} โดย p_{ij} คือค่าความน่าจะเป็นที่สมาชิกของ *cluster* j อยู่ใน *class* i หลังจากนั้นค่า Entropy ของแต่ละ *cluster* j หาได้จาก

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

ค่า Entropy ของทั้งหมดหาได้จาก

$$E_{CS} = \sum_{j=1}^{m} \frac{n_j * E_j}{n}$$

n_j เป็นขนาดของ *cluster* j, m เป็นจำนวนของ clusters, n เป็นจำนวนของข้อมูลทั้งหมด ค่า Entropy จะมีค่าน้อยเมื่อผลการ clustering มีการซ้อนทับกันน้อยและจะมีค่ามากขึ้นถ้าผลการ clustering มีการซ้อนทับกันมาก

4.2 ข้อมูลที่ใช้ในการทดลอง

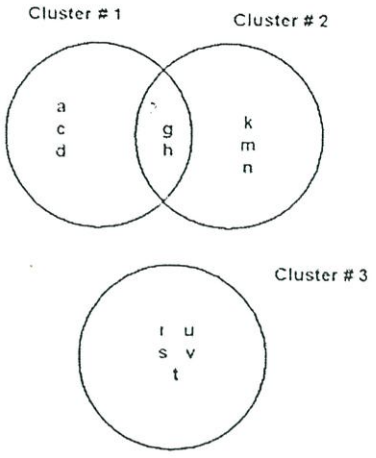
ข้อมูลที่นำมาใช้ในการทดสอบประสิทธิภาพของอัลกอริธึม TPKNN ประกอบด้วย ข้อมูลที่สร้างขึ้นโดยใช้การสุ่มตัวอักษรภาษาอังกฤษซึ่งในที่นี่เรียกว่า 'ข้อมูลชุดตัวอักษร' และข้อมูลจากข่าว reuter-21578 โดยรายละเอียดของข้อมูลแต่ละชุดมีดังนี้

4.2.1 ข้อมูลชุดตัวอักษร

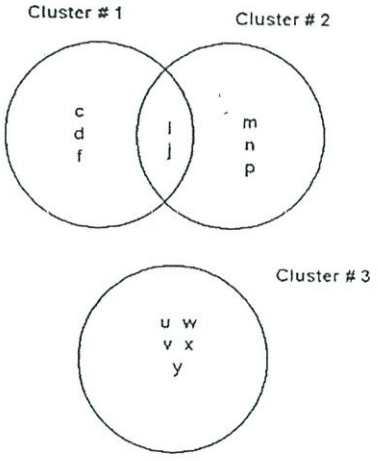
ข้อมูลชุดตัวอักษรนี้สร้างจากกลุ่มของตัวอักษรจำนวน 3 กลุ่มซึ่งใช้เป็นตัวแทนของข้อมูลที่มีค่าของคุณสมบัติเป็นข้อความ ข้อมูลแต่ละตัวประกอบด้วยคุณสมบัติ 2 คุณสมบัติคือ Title และ keyword

$$Doc = Title * Keyword$$

กลุ่มของตัวอักษรที่ใช้สร้างคุณสมบัติ Title และ keyword แสดงได้ดังรูปที่ 4.1 และ 4.2



รูปที่ 4.1 ชุดตัวอักษรที่ใช้สร้างข้อมูลในคุณสมบัติ Title



รูปที่ 4.2 ชุดตัวอักษรที่ใช้สร้างข้อมูลในคุณสมบัติ keyword

ตัวอย่างของข้อมูลที่สร้างขึ้นจากชุดของตัวอักษรในภาพที่ และ แสดงได้ดังตารางที่ 4.1

ตารางที่ 4.1 แสดงข้อมูลบางส่วนที่ใช้ในการเทรนนิ่ง

ลำดับที่	ค่าของคุณสมบัติ	
	Title	Keyword
1	c,d,g,h	d,j
2	n	i,m,np
3	r,s,t,v	u,w,y
4	h,k,n	j,p

โดยข้อมูลที่นำมาใช้ในการเทรนนิ่งประกอบด้วยข้อมูลจำนวน 100 ข้อมูลโดยเป็นสมาชิกของ cluster 1 จำนวน 34 ข้อมูล ข้อมูลที่เป็นสมาชิกของ cluster 2 จำนวน 35 ข้อมูล ข้อมูลที่เป็นสมาชิกของ cluster 3 จำนวน 35 ข้อมูล สำหรับข้อมูลที่นำมาทดสอบความถูกต้องของโมเดลที่ได้จากการเทรนนิ่งประกอบด้วยชุดข้อมูลที่ได้จากการสุ่มตัวอักษรจากกลุ่มของตัวอักษรเดียวกันกับชุดข้อมูลที่ใช้ในการเทรนนิ่งจำนวน 5 ชุด โดยมีรายละเอียดดังตารางที่ 4.2

ตารางที่ 4.2 แสดงรายละเอียดของชุดข้อมูลที่นำมาใช้ในการทดสอบความถูกต้องของโมเดล

ข้อมูล	จำนวน	cluster 1	cluster 2	cluster 3
data1k01	1000	328	331	341
data1k02	1000	333	322	345
data1k03	1000	342	321	337
data1k04	1000	322	337	341
data1k05	1000	345	333	322

4.1.2 ข้อมูลข่าว reuter-21578

ข้อมูลข่าว reuter-21578 เป็นชุดของข้อมูลที่นิยมใช้ในการทำวิจัยด้าน information retrival และ การทำ text clustering ประกอบด้วยข่าวจำนวน 22173 ข่าวและมีหัวข้อข่าว 135 หัวข้อ ข้อมูลข่าว reuter-21578 นั้นจะอยู่ในรูปของแฟ้มข้อมูล SGML ซึ่งในการนำมาใช้จะต้องมีการเตรียมข้อมูลก่อน โดยอาจแบ่งขั้นตอนการเตรียมข้อมูลเป็น 2 ขั้นตอนหลักคือ

ขั้นตอนที่ 1 เป็นขั้นตอนในการแยกข่าว reuter-21578 เพื่อเลือกเอาเฉพาะหัวข้อข่าวและเนื้อความของข่าวมาใช้

ขั้นตอนที่ 2 เป็นขั้นตอนของการตัดคำนำหน้านาม (article) เครื่องหมายวรรคตอนต่างๆ, ตัวเลข และคำทั่วไปต่างๆ ออก

ขั้นตอนที่ 3 เป็นขั้นตอนของการหาคำที่พบบ่อยในข่าวนั้นๆ ซึ่งคำที่พบบ่อยนี้จะถูกนำไปใช้เป็นค่าใน feature ของข่าวแต่ละข่าว

โดยในที่นี้ เราเลือกเฉพาะข่าวจาก 5 กลุ่มข่าวซึ่งมีสมาชิกมากพอมาทำการทดลองซึ่งมีรายละเอียดดังตารางที่ 4.3

ตารางที่ 4.3 แสดงรายละเอียดของหัวข้อข่าว reuter-21578 ที่นำมาทดลอง

กลุ่มข่าว	จำนวนข่าวทั้งหมด	จำนวนข่าวที่ใช้ train
earn	3183	355
acq	1932	331
crude	496	221
grain	467	215
money-fx	596	239

ซึ่งในการทดลองจะแบ่งข้อมูลเพื่อใช้ในการเทรนนิ่งโมเดลดังตารางที่ โดยทำการแบ่งข้อมูล 5 กลุ่มนี้เป็นชุดข้อมูล 2 ชุดคือ reuter1 เป็นชุดข้อมูลของข่าวในกลุ่ม {acq,crude,grain}และ reuter2 เป็นชุดข้อมูลของข่าวในกลุ่ม {acq,crude,grain,money-fx,earn} ส่วนข้อมูลข่าวที่ใช้ในการทดสอบความถูกต้องของโมเดลที่ได้จากการเทรนนิ่งนั้นจะใช้ข้อมูลของข่าวทั้งหมดในกลุ่มที่ได้ใช้ไปในการเทรนนิ่งนั้นๆ

4.3 ผลการทดลอง

ในการทดสอบประสิทธิภาพของ TPKNN อัลกอริธึมนี้แบ่งการทดลองออกเป็น 2 ส่วนคือ การทดลองเบื้องต้นโดยใช้ข้อมูลที่สังเคราะห์ขึ้นเองจากชุดตัวอักษรภาษาอังกฤษ (ชุดข้อมูลตัวอักษร)และการทดลองกับข้อมูลในชีวิตจริงจากชุดข้อมูลข่าว reuter-21578 โดยมีกระบวนการทดสอบและผลการทดสอบดังนี้

4.3.1 ชุดข้อมูลตัวอักษร

ชุดข้อมูลตัวอักษรประกอบด้วยข้อมูลดังหัวข้อที่ 4.2.1 โดยในการเทรนนิ่งนี้ประกอบด้วยนิวรอล (neural) ใน output layer จำนวน 4 โหนด แต่ละโหนดประกอบด้วย weight(w_{ij}) ซึ่งเป็นตัวแทนของ feature 2 feature คือ title feature และ keyword feature ผลลัพธ์ที่ได้หลังจากการทำเทรนนิ่งแล้วเป็นดังตารางที่ 4.4

ตารางที่ 4.4 ผลลัพธ์ที่ได้จากการเทรนนิ่ง

Neural ลำดับที่	ผลลัพธ์จากการเทรนนิ่ง		แสดงความเป็นตัวแทนของ cluster
	Title	Keyword	
1	c,d,g,h,s,u,v	d,f,i,j,w,y	1
2	d,g,h,k,m,n,s,t,v	c,d,f,i,j,p,w,y	Other
3	a,g,h,k,n	d,f,I,j,m,n,p	2
4	s,t,u,v	u,v,w,x,y	3

หลังจากทำการเทรนนิ่งแล้ว เราจะได้ค่า weight(w'_{ij}) ของนิวรอลซึ่งเป็นตัวแทนของ title feature และ keyword feature ของข้อมูลที่นำมาทำการเทรนนิ่ง โดยนิวรอลลำดับที่ 1 เป็นตัวแทนของ cluster 1 นิวรอลลำดับที่ 3 เป็นตัวแทนของ cluster 2 และนิวรอลลำดับที่ 4 เป็นตัวแทนของ cluster 3

ในการทดสอบความถูกต้องของนิวรอลเน็ตเวิร์คที่ได้จากการเทรนนิ่ง เราใช้ข้อมูลที่สร้างขึ้นจากชุดตัวอักษรเดียวกันกับข้อมูลที่ใช้ในการเทรนนิ่งจำนวน 1000 ตัวจำนวน 5 ชุดดังแสดงรายละเอียดในตารางที่ นำมาใช้เพื่อตรวจสอบความถูกต้องของโมเดลนิวรอลเน็ตเวิร์คที่ได้ โดยผลลัพธ์จากการทดสอบเป็นดังตารางที่ 4.5

ตารางที่ 4.5 แสดงผลลัพธ์ของการทดสอบโมเดลนิลรอลเน็ตเวิร์คของข้อมูลทดสอบ

ข้อมูล	จำนวนสมาชิก			ผลการทดลอง			F measure	Entropy
	cluster1	cluster2	cluster3	cluster1	cluster2	cluster3		
1	328	331	341	285	289	341	0.916	0.271
2	333	322	345	279	291	345	0.916	0.269
3	342	321	337	307	282	337	0.927	0.244
4	322	337	341	282	305	341	0.929	0.240
5	345	333	322	298	287	322	0.909	0.289
ค่าเฉลี่ย F measure : 0.919								
ค่าเฉลี่ย Entropy :0.262								

จากการทดลองเบื้องต้นกับข้อมูลชุดตัวอักษร แสดงให้เห็นว่าอัลกอริธึม TPKNN สามารถทำการแยกกลุ่มข้อมูลได้เป็นอย่างดี โดยมีค่าเฉลี่ยความถูกต้องในการแยกกลุ่ม F measure เท่ากับ 0.919 และมีค่าเฉลี่ยการซ้อนทับกันของข้อมูล Entropy เท่ากับ 0.262

4.3.2 ชุดข้อมูล reuter-21578

ชุดข้อมูลตัวอักษรประกอบด้วยข้อมูลดังหัวข้อที่ 4.2.2 โดยในการเทรนนิ่งนี้ประกอบด้วย นิวรอล (neural) ใน output layer จำนวน 9 โหนด แต่ละโหนดประกอบด้วย weight(w_{ij}) ซึ่งเป็นตัวแทนของ feature 2 feature คือ title feature และ keyword feature โดยโมเดลที่ได้หลังจากการเทรนนิ่งจะถูกนำไปทดสอบกับข้อมูลที่เป็นสมาชิกของกลุ่มข่าวที่ใช้ในการเทรนนิ่งทั้งหมด นั่นคือ ข้อมูลชุด reuter1 จะถูกทดสอบด้วยข้อมูลจากกลุ่มข่าว {acq,crude,grain } ทั้งหมด และข้อมูล reuter2 จะถูกทดสอบด้วยข้อมูลจากกลุ่มข่าว {acq,crude,grain,money-fx,earn} ทั้งหมด ซึ่งผลลัพธ์ของการทดสอบเป็นดังตารางที่ 4.6 และ 4.7

ตารางที่ 4.6 ผลลัพธ์จากการทดสอบโมเดลที่ได้จากเทรนนิ่งข่าว reuter1

กลุ่มข่าว	จำนวนข่าวที่นำมาทดสอบ	ผลลัพธ์ของข่าวที่ทำการ clustering ได้ถูกต้อง
acq	1932	1546
crude	496	323
grain	467	308
F measure = 0.80		
Entropy =0.48		

ตารางที่ 4.7 ผลลัพธ์จากการทดสอบโมเดลที่ได้จากเทรนนิ่งข่าว reuter2

กลุ่มข่าว	จำนวนข่าวที่นำมาทดสอบ	ผลลัพธ์ของข่าวที่ทำการ clustering ได้ถูกต้อง
acq	1932	1171
crude	496	292
grain	467	331
money-fx	596	255
earn	3183	2036
F measure = 0.67		
Entropy =0.73		

จากผลการทดลองกับชุดข้อมูลตัวอักษรแสดงให้เห็นว่าตัวอัลกอริธึม TPKNN สามารถทำงานได้อย่างถูกต้องและมีประสิทธิภาพ โดยสามารถทำการ clustering ข้อมูลสังเคราะห์นี้ได้เป็นอย่างดี มีค่า F measure วัดได้ 0.919 และค่า Entropy วัดได้ 0.262 ซึ่งแสดงถึงผลของการ clustering ที่มีคุณภาพดี มีการซ้อนทับกันของผลการ clustering ในแต่ละกลุ่มน้อย ในขณะที่เมื่อทำการทดลองกับข้อมูล reuter-21578 ผลการทดลองลดต่ำลง โดยในชุดข้อมูล reuter1 มีค่า F measure วัดได้ 0.80 และค่า Entropy วัดได้ 0.48 และในชุดข้อมูล reuter2 มีค่า F measure วัดได้ 0.67 และค่า Entropy วัดได้ 0.73 ซึ่งแสดงถึงผลของการ clustering ที่ดี แต่ยังมีการการซ้อนทับกันของผลการ clustering ในแต่ละกลุ่มพอสมควรในข้อมูลชุด reuter1 และค่อนข้างสูงในชุดข้อมูล reuter2

เมื่อเปรียบเทียบผลลัพธ์ของการ clustering ข้อมูล reuter-21578 โดยใช้อัลกอริธึมแบบ TPKNN กับการ clustering ด้วยวิธี Centroid Similarity และ Cluster Similarity จากงานวิจัยของ Michael Steinbach และคณะ [11] ซึ่งวัดค่า F measure ได้ 0.50 และ 0.53 ตามลำดับ พบว่าค่า F measure ที่วัดได้จากการใช้อัลกอริธึมแบบ TPKNN มีค่าใกล้เคียงและสูงกว่าผลการ clustering ด้วยวิธีทั้งสองข้างต้น ซึ่งแสดงให้เห็นถึงประสิทธิภาพของ TPKNN อัลกอริธึมได้เป็นอย่างดี

บทที่ 5
สรุปผลและแนวทางการพัฒนาในอนาคต

Text Processing Kohonen Neural Networks(TPKNN) เป็นอัลกอริธึมที่พัฒนาขึ้นเพื่อให้สามารถทำการ clustering ข้อมูลประเภทข้อความได้โดยตรง ไม่ต้องทำการแปลงข้อมูลให้เป็นข้อมูลเชิงตัวเลขก่อน ซึ่งช่วยให้สามารถลดเวลาในการแปลงและกำหนดสัญลักษณ์เพื่อแทนข้อมูล และยังสามารถคงความหมายของข้อมูลเดิมไว้ โดยการประยุกต์แนวคิดการหาความเหมือนกันของข้อมูลแบบ symbolic data เข้ากับความสามารถในการทำ clustering ของ Self-Organizing Feature Maps โดยประกอบด้วยการทำงานใน 3 ขั้นตอนหลักคือ

1. ขั้นตอนของการหา wining neural เป็นขั้นตอนที่ประยุกต์แนวแนวคิดการหาความเหมือนกันของข้อมูลแบบ symbolic data ในส่วนของข้อมูลที่เป็น qualitative เข้าไปเพื่อวัดความแตกต่างกันของข้อมูลที่เข้ามากับนิวรอลในชั้นของ output layer

2. ขั้นตอนของการหา neighborhood เป็นขั้นตอนของการหา neighborhood ของ wining neural ที่ได้จากขั้นตอนแรกโดยใช้ neighborhood function

3. ขั้นตอนของการปรับค่า weight vector ของ wining neural และ neighborhood ของมันเพื่อให้มีความใกล้เคียงกับข้อมูลเข้ามากขึ้นตามลำดับ

เพื่อทำการวัดประสิทธิภาพของโมเดลที่ได้จากการเทรนนิ่งโครงข่ายประสาทเทียมแบบ TPKNN ในงานวิจัยนี้ได้เลือกตัววัด F measure เพื่อใช้ในการวัดค่าความถูกต้องของการ clustering โดยค่า F measure จะมากถ้าผลการ clustering มีคุณภาพดีและตัววัด Entropy เพื่อใช้วัดการซ้อนทับของผลการ clustering โดยค่า Entropy จะน้อยถ้ามีการซ้อนทับกันเกิดขึ้นน้อย

ผลการทดลองซึ่งประกอบด้วยการทดลองกับชุดข้อมูลตัวอักษร (ตารางที่ 4.5)และการทดลองกับข้อมูลข่าว reuter-21578 ซึ่งประกอบด้วยชุดข้อมูล reuter1 และชุดข้อมูล reuter2 (ตารางที่ 4.6 , 4.7) ตามลำดับ

จากผลการทดลองกับชุดข้อมูลตัวอักษรแสดงให้เห็นว่าตัวอัลกอริธึม TPKNN สามารถทำงานได้อย่างถูกต้องและมีประสิทธิภาพ ในขณะที่เมื่อทำการทดลองกับข้อมูล reuter-21578 ผลการทดลองลดต่ำลง โดยในชุดข้อมูล reuter1 มีผลของการ clustering ที่ดี แต่ยังมีการซ้อนทับกันของผลการ clustering ในแต่ละกลุ่มพอสมควร ส่วนข้อมูลชุด reuter2 การซ้อนทับกันของผลการ clustering ในแต่ละกลุ่มมีค่อนข้างสูงทำให้ได้ค่า F measure ที่ต่ำลง แต่เมื่อเปรียบเทียบผลการ clustering กับการ clustering ด้วยวิธีอื่นๆใน [11] พบว่า TPKNN อัลกอริธึมยังคงสามารถทำการ clustering ได้อย่างมีประสิทธิภาพเมื่อเทียบกับการ clustering แบบอื่นๆ

การลดลงของค่า F measure ในการทดลองกับชุดข้อมูล reuter-21578 เกิดจากการกระจายของคำที่พบบ่อยในตัวข้อมูลเอง ซึ่งทำให้ค่าของ weight vector ในแต่ละโนดของ neural output ไม่สามารถเก็บค่าที่เป็นตัวแทนของข้อมูลแต่ละกลุ่มได้อย่างแท้จริง ซึ่งแสดงให้เห็นจากผลการทดลองของข้อมูลชุด reuter1 เทียบกับชุดข้อมูล reuter2 โดยชุดข้อมูล reuter1 ซึ่งประกอบด้วยกลุ่มข่าวจำนวนน้อยกว่าได้ค่า Entropy ซึ่งแสดงถึงการซ้อนทับกันของผลการ clustering น้อยกว่าชุดข้อมูล reuter2 อย่างเห็นได้ชัด และเมื่อเทียบกับชุดข้อมูลสังเคราะห์ซึ่งมีการกระจายกันของคำน้อย ผลการ clsutering ที่ได้จึงมีค่า Entropy น้อยด้วย แนวทางการแก้ไขอาจทำได้โดยการจำกัดจำนวนของสมาชิกใน weight vector ,ลดค่า learning rate ลง และเพิ่มระยะเวลาในการเรียนรู้ให้มากขึ้น เพื่อให้ neural output แต่ละโนดสามารถเก็บค่าที่เป็นตัวแทนของกลุ่มข้อมูลได้ดียิ่งขึ้น

เอกสารอ้างอิง

[1] Anil K. Jain and Richard C. Dubes. **Algorithms for Clustering Data**. Prentice Hall Englewood Cliffs, New Jersey 07632. ISBN: 0-13-022278-X

[2] Eric Backer. **Computer-assisted Reasoning in Cluster Analysis**. Prentice Hall International (UK) Limited Campus 400, Maylands Avenue Hemel Hempstead Hertfordshire, HP2 7EZ ISBN: 0-13-341884-7

[3] 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.

[4] Kohonen T. **Self-Organization and Associative Memory**. Berlin: Springer-Verlag 1989.

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

[6] DeSieno, D. "Adding a conscience to competitive learning" Neural Networks, 1988., IEEE International Conference on , 24-27 Jul 1988 pp. 117 -124 vol.1

[7] 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.

[8] Dinesh M.S., Gowda K.C., Ravi T.V. "Classification of symbolic data using fuzzy set theory" First International Conference on Knowledge-Based Intelligent Electronic Systems, 1997., vol. 2, pp. 383 -386

[9] Ravi, T.V. and Gowda K.C. "Clustering of symbolic objects using gravitational approach" IEEE Trans. On Syst., Man, Cybern., vol. 29, no. 6, pp.888 -894, 1999.

[10] 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.

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

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

ภาคผนวก
ผลงานวิจัยที่ได้รับการตีพิมพ์

1. Worapoj Kreesuradej and Songpol Chutipongpattanakul. "Document Clustering Using Text Processing Kohonen Neural Networks." ISCIT2001, November 2001. pp.41-43.

Document Clustering Using Text Processing Kohonen Neural Networks

Worapoj Kreesuradej,Ph.D. and Songpol Chutipongpattanakul

Worapoj Kreesuradej,Ph.D
Advanced Computer Application Group and
Design Research Group,
Facluty of Information Technology,
King Mongkut's Institute of Technology
Ladkrabang,
Ladkrabang, Bangkok 10520 Thailand
Tel: (662) 737-2551-4 ext. 522
Fax: (662) 3269074
Email: worapoj@it.kmitl.ac.th

Songpol Chutipongpattanakul
Advanced Computer Application Group and
Design Research Group,
Facluty of Information Technology,
King Mongkut's Institute of Technology
Ladkrabang,
Ladkrabang, Bangkok 10520 Thailand
Tel: (662) 737-2551-4 ext. 522
Fax: (662) 3269074
Email: s1067011@kmitl.ac.th

ABSTRACT

This paper proposes document clustering using Kohonen neural network. This algorithm works directly on textual information without mapping documents into some representation that has quantitative features. The input level of the proposed neural network can directly receive a qualitative value without mapping the qualitative value into numerical value. Then, based on Kohonen self organize feature maps and the concept 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 SOFM neural network are proposed for clustering such objects. Recently, clustering problems are extends 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 conventional methods, this paper propose a Text Processing Kohonen Neural Network. The proposed algorithm works directly on textual information without mapping documents onto some representation that has quantitative features and can assigns cluster labels to the objects. The results obtained from the proposed algorithm are shown in section 5.

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 [4]

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

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 [4], 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 define as below:

$$D_s(A_k, B_k) = \frac{|Length\ of\ A_k - Lenght\ of\ B_k|}{Span\ Length\ of\ A_k\ and\ B_k}$$

$$D_c(A_k, B_k) = \frac{|Length\ of\ A_k + Lenght\ of\ B_k - 2 * Lenght\ of\ intersection\ of\ A_k\ and\ B_k|}{Span\ Length\ of\ A_k\ and\ B_k}$$

Where the length of feature is number of its elements and the span length of two features value is define 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 KOHONEN NEURAL NETWORKS

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

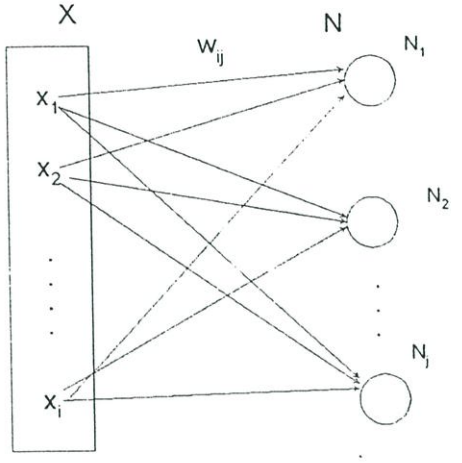


Figure 1. The architecture of Text Processing Kohonen Neural Network

Basically, the proposed network consists of two layers: input unit and output unit. Input unit each of which is fully connected to a set of output unit. These output units are arranged in two-dimensional grid. Unlike conventional neural networks, input units of the proposed neural network can receive qualitative values. The weight of output unit $'i'$ feature $'j'$, 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 $'i'$. The value of e_{pij} is 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 $'i'$.

Learning Algorithm

The Text Processing Kohonen Neural Network algorithm is based on Kohonen Self Organize Feature Maps and the concept of qualitative symbolic dissimilarity measure as describe in the prior section. The algorithm is summarized as below:

1: Initialize weight w_{ij} in each neural W_i. Each weight can be initialize from the training data arbitrarily.

$$W_i = \{w_{i1}, w_{i2}, \ldots, w_{ij}\}$$

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

w_{ij} = Weight of neural W_i feature j

A_{pij} = Member p^{th} of w_{ij}

e_{pij} = Degree of membership of A_{pij}

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

3: Draw a sample X from the input distribution with a certain probability. For each input vector

$X = (x_1, x_2, \ldots, x_d)^T$, Do step 4-6

4: For each output unit 'i', Compute

$$\|X - W_i\| = \sum_{k=1}^{d} D(x_k, w_{ik})$$

d= number of input and output feature

Finds index 'I' such that $\|X - W_I\|$ is minimum.

5: For all weights that connect to the wining node I and its' neighborhood $(\wedge_I)$.

$$w_{ij}^{(new)} = \begin{cases} w_{ij}^{(old)} \bigcup x_j & if \quad i \in \wedge_I \\ w_{ij}^{old} & Otherwise \end{cases}$$

And

$$e_{ijp}^{(new)} = \begin{cases} f(e_{ijp}^{(old)} + \eta) & if \quad A_{ijp} \in w_{ij} \cap x_j \\ f(e_{ijp}^{(old)} - \eta) & if \quad A_{ijp} \notin w_{ij} \cap x_j \end{cases}$$

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

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

6: Continue with step 2-5 until the stopping condition is true.

5. EXPERIMENTAL RESULT

In this section, experimental results of a document clustering based on the proposed neural network are represented. Here, a training set of 100 documents that consist of 3 cluster is synthesized. Each document in the training set can be represented by title feature and keyword feature, i.e.

$$Doc = Title * Keyword.$$

Each feature of a document is qualitative value. For this experiment, some English alphabets are used to represent value of each feature.The values of title feature and keyword feature of each cluster are shown in figure 2 and 3 respectively.

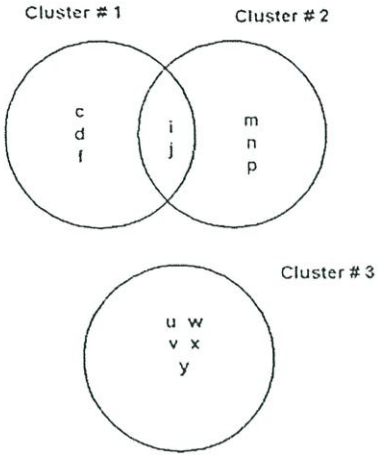


Fig. 2 Title feature profile

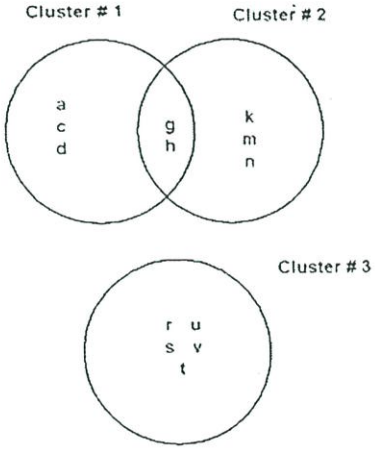


Fig. 3 Keyword feature profile

As an example, some documents that are generated according to data profile in figure 2 and figure 3 are shown in Table 1.

Table 1. Some documents of the training data

No.	Data value	
	Title	Keyword
1	c,d,g,h	d,j
2	n	i,m,np
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 documents from 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

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

Where c is a number of documents that are incorrectly assigned to a wrong cluster. According to the testing set, the proposed network gives the accuracy 91.4 %. This shows that the network have well performance in clustering text data.

6. CONCLUSIONS

In the future, some experiment results that are conducted on the Reuter-21578 news articles [5] 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 H., Mercel Dekker, New York, 1998.

[2] Kohonen T. " Self-Organization and Associative Memory" Berlin: Springer-Verlag 1989.

[3] 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.

[4] 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.

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

ประวัติผู้เขียน

ชื่อ-นามสกุล	นายทรงพล ชุติพงศ์พัฒนกุล
วันที่เกิด	29 มิถุนายน พ.ศ.2518
สถานที่เกิด	จังหวัดฉะเชิงเทรา
วุฒิการศึกษา	วิทยาศาสตรบัณฑิต (วทบ.)
	สาขาคณิตศาสตร์ประยุกต์
	คณะวิทยาศาสตร์
	สถาบันเทคโนโลยีเจ้าคุณทหารลาดกระบัง