การจดจำอักษรพิมพ์ภาษาไทยโดยโครงข่ายประสาทเทียม แบบจำลองอะแดปทีฟรีโซแนนซ์เทียรี่

PRINTED THAI CHARACTER RECOGNITION USING ART NEURAL NETWORK MODEL

เพ็ญพรรณ ใช้ฮวดเจริญ

PENPUN CHAIHAUDJAROEN

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต
สาขาวิชาวิทยาการคอมพิวเตอร์และเทคโนโลยีสารสนเทศ
บัณฑิตวิทยาลัย
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
พ.ศ. 2541

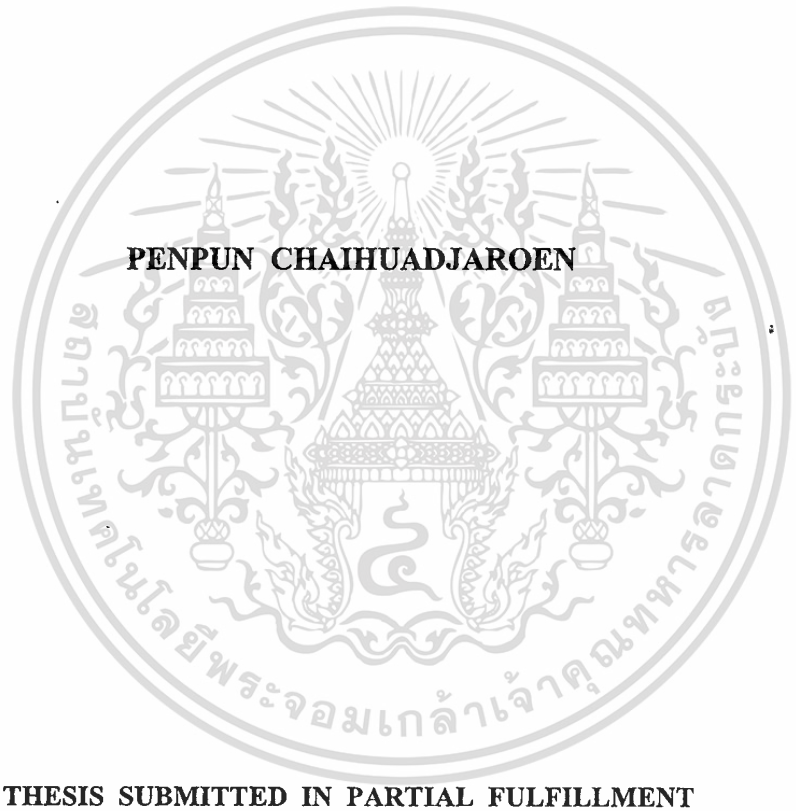
ISBN 974-622-167-1

เลขหมู่ ว. พ ... ท 2541
เลขทะเบียน 31029
วัน, เดือน, ปี 8 ก.ย. 2541

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PRINTED THAI CHARACTER RECOGNITION BY ADAPTIVE RESONANCE THEORY NEURAL NETWORK

PENPUN CHAIHUADJAROEN



THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE
MASTER OF SCIENCE IN COMPUTER SCIENCE
AND INFORMATION TECHNOLOGY
SCHOOL OF GRADUATE STUDIES
KING MONGKUT'S INSTITUE OF TECHNOLOGY LADKRABANG
1998

ISBN 974-622-167-1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

COPYRIGHT 1998

SCHOOL OF GRADUATE STUDIES

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

เอกสารนี้สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อวิทยานิพนธ์	การจดจำอักษรพิมพ์ภาษาไทยโดยโครงข่ายประสาทเทียมแบบจำลองอะแดปทีฟรีโซแนนซ์เทียรี่
นักศึกษา	นางสาวเพ็ญพรรณ ใช้ฮวดเจริญ
อาจารย์ผู้ควบคุมวิทยานิพนธ์	ผศ.ดร.บุญธีร์ เครือตราชู
หลักสูตร	วิทยาศาสตรมหาบัณฑิต
สาขาวิชา	สาขาวิชาวิทยาการคอมพิวเตอร์ และเทคโนโลยีสารสนเทศ
พ.ศ.	2541

บทคัดย่อ

งานวิจัยนี้เป็นการทดสอบแบบจำลองโครงข่ายประสาทเทียมอะแดปทีฟรีโซแนนซ์เทียรี (ART) โดยนำมาประยุกต์ในงานจดจำตัวอักษรพิมพ์ภาษาไทย และได้ทำการปรับปรุงแบบจำลองในส่วนของการเรียนรู้ และการเปลี่ยน Threshold เพื่อลดเวลาในขั้นตอนของการเรียนรู้ และลดความผิดพลาดในการจดจำตัวอักษร งานวิจัยนี้ได้แสดงให้เห็นถึงปัญหาและวิธีการแก้ไขโครงสร้างของ ART เพื่อเพิ่มประสิทธิภาพในการรู้จำ งานวิจัยนี้ได้ทำการฝึกสอนโครงข่ายด้วยตัวอักษรภาษาไทย 75 แบบ แบบละ 20 ตัว และทำการทดสอบด้วยตัวอักษรภาษาไทย 75 แบบ แบบละ 200 ตัว รวมทั้งหมด 15,000 ตัว ซึ่งพิมพ์ด้วยเครื่องพิมพ์เลเซอร์ โดยใช้ฟอนต์อังศนายูพีซีขนาด 12 จุดต่อนิ้ว จะได้ผลความถูกต้องประมาณ 99.13% ใช้เวลาในการฝึก 20 นาที และจำนวนรอบในการฝึก 6 รอบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Thesis Title	Printed Thai Character Recognition using ART Neural Network Model
Student	Miss Penpun Chaihuadjaroen
Thesis Advisor	Assist.Prof.Dr. Boontee Kruatrachue
Degree	Master of Science Program in Computer Science and Information Technology
Year	1998

ABSTRACT

This research presents an application of ART neural network model on Thai printed character recognition. To reduce the training time and the percentage error of recognition, the original ART model is modified. Additionally, the comparition between ART , Self-Organizing Maps, Back Propagation and CPN in terms of training time, number of node used and reconition efficiency is reported. This improveing ART model is obtained by training 6 times with 1,500 characters of 75 Thai Character patterns. With 15,000 samples of 75 different Thai character pattern of 12 point Angsana UPC font, the percent recognition of this model is 99.13%. The overall training time is 20 minutes.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ขอขอบพระคุณ ดร.บุญธีร์ เครือตราชู อาจารย์ผู้ควบคุมวิทยานิพนธ์เป็นอย่างสูง ที่ได้ให้คำปรึกษาตลอดจนให้คำแนะนำต่างๆ รวมทั้งให้กำลังใจในการทำวิทยานิพนธ์จนสำเร็จลุล่วงไปได้ด้วยดี

ขอขอบคุณ เพื่อนๆ รุ่นพี่ และรุ่นน้องทุกๆคนที่ได้ให้ความช่วยเหลือและเป็นกำลังใจตลอดมา และขอขอบคุณภาควิชาอิเล็กทรอนิกส์ มหาวิทยาลัยเทคโนโลยีมหานครที่ได้เอื้อเฟื้ออุปกรณ์และเครื่องคอมพิวเตอร์ในช่วงเวลาที่ทำการวิจัย

ขอขอบคุณคุณจักรภาษณ์ วิศวกุล ที่ได้ช่วยเหลือทางด้านอุปกรณ์ และให้กำลังใจแก่ผู้วิจัยมาตลอด รวมทั้งช่วยแนะนำวิธีการแก้ไขปัญหาในบางจุดที่ผู้วิจัยติดปัญหานั้น

สุดท้ายขอขอบพระคุณ บิดา-มารดา พี่สาวและน้องสาวของข้าพเจ้าที่ได้คอยเป็นกำลังใจให้ข้าพเจ้าตลอดมา

เพ็ญพรรณ ใช้ฮวดเจริญ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

หน้า

บทคัดย่อภาษาไทย I

บทคัดย่อภาษาอังกฤษ II

กิตติกรรมประกาศ III

สารบัญ IV

สารบัญตาราง VI

สารบัญภาพ VII

อภิธานศัพท์.....IX

บทที่ 1 บทนำ.....1

1.1 ความเป็นมาของงานวิจัย.....1

1.2 ผลงานวิจัยที่ผ่านมา..... 3

1.3 แนวทางของวิทยานิพนธ์.....4

1.4 วัตถุประสงค์ของงานวิจัย..... ...4

1.5 โครงสร้างของวิทยานิพนธ์..... 4

บทที่ 2 ทฤษฎีเบื้องต้นของโครงข่ายประสาทเทียม..... 6

2.1 ทฤษฎีเซลประสาทเบื้องต้น.....7

2.2 ตัวแบบของเซลประสาทเทียม.....8

2.3 สถาปัตยกรรมของโครงข่ายประสาทเทียม.....12

2.4 การฝึกสอนโครงข่ายประสาทเทียม.....13

2.5 การแบ่งแบบเชิงเส้น..... 14

บทที่ 3 โครงข่ายประสาทเทียมแบบ ART.....18

3.1 ทฤษฎีของ ART..... 18

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้า

3.2 โครงสร้างของโครงข่าย ART.....19

3.2 การปรับค่าน้ำหนักของ ART.....20

3.3 อัลกอริทึมที่ 1 ใช้ฝึกสอน ART.....22

3.4 Supplemental Units.....24

3.5 การทดลองใช้ ART เรียนรู้ตัวอักษรภาษาไทย.....26

3.6 ผลการทดลอง.....27

3.7 ปัญหาและการปรับปรุงเพื่อนำ ART มาใช้แยกตัวอักษรภาษาไทย.....28

บทที่ 4 ARTS และปัญหาของ ARTS ที่ใช้ในการจดจำอักษรพิมพ์ภาษาไทย.....29

4.1 อัลกอริทึมที่ใช้สอน ARTS.....29

4.2 การทดลองแยกภาษาไทยโดยใช้ ARTS.....31

4.3 ปัญหาการใช้ ARTS แยกตัวอักษรภาษาไทย.....35

4.4 การใช้ค่าเทรสโฮลแก้ปัญหา ARTS.....36

4.5 อัลกอริทึมที่ใช้สอน ARTS แบบมีการปรับเปลี่ยนค่า ρ.....36

บทที่ 5 ผลการทดลองใช้ค่าเทรสโฮลแก้ปัญหา ARTS.....39

บทที่ 6 บทสรุปและแนวทางการพัฒนาในอนาคต.....54

6.1 สรุปผลการทดลอง.....54

6.2 แนวทางการพัฒนาในอนาคต.....55

6.2.1 การเปลี่ยนวิธีหาค่า ratio.....55

6.2.2 การเรียงลำดับ ratio.....57

6.3 บรรณานุกรม.....58

6.4 ภาคผนวก.....59

6.5 ภาคผนวก ก บทความทางวิชาการ.....60

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้า

6.6 ภาคผนวก ข ตัวโปรแกรมของระบบการรู้จำ..... 70

6.7 ประวัติผู้เขียน.....94

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่ **หน้า**

3.1 แสดงการเรียนผิดโหนด ซึ่งเป็นผลของการฝึกตัวอักษรไทยโดยใช้ ART.....26

4.1 ประสิทธิภาพจำทดสอบโดยใช้ ARTS 31

4.2 แสดงตัวอักษรที่แยกแยะผิดโดยใช้ ARTS 33

4.3 แสดงการเรียงลำดับของโหนดชนะของตัวอักษรเลข 1.....34

5.1 แสดงสิทธิภาพจำทดสอบโดยใช้ ARTS ทดสอบกับตัวอักษรที่แสกนจาก เครื่องพิมพ์เลเซอร์ ความสว่าง 100 % 40

5.2 แสดงจำนวนอักษรที่แยกแยะผิดโดยใช้ ARTS 41
15,000 ตัว ทดสอบกับตัวอักษรที่สแกนจากกระดาษที่พิมพ์ด้วยเครื่อง

5.3 แสดงค่า ρ ของโหนดทั้งหมด ซึ่งเป็นผลที่ได้จากการฝึก.....43

5.4 แสดงการเรียงลำดับของโหนดชนะของตัวอักษร ไ1.....44

5.5 แสดงการเรียงลำดับของโหนดชนะของตัวอักษร ไ14.....47

5,6 ประสิทธิภาพการจำ ทดสอบกับตัวอักษรที่ถ่ายเอกสารจากการดาษที่พิมพ์ ด้วยเครื่องพิมพ์เลเซอร์ ความสว่าง 50 % โดยใช้การปรับเปลี่ยนค่า ρ.....49

5.7ประสิทธิภาพการจำทดสอบกับตัวอักษรที่ถ่ายเอกสารจากกระดาษที่พิมพ์ ด้วยเครื่องพิมพ์เลเซอร์ ความสว่าง 100 % โดยใช้การปรับเปลี่ยนค่า ρ 50

5.8 ประสิทธิภาพการจำ ทดสอบกับตัวอักษรที่ถ่ายเอกสาร จากกระดาษถ่าย เอกสารความสว่าง 50 % โดยใช้การปรับเปลี่ยนค่า ρ.....51

5.9 เปรียบเทียบแบบจำลองชนิดต่างๆ ที่นำมาใช้ในการรู้จำตัวอักษรไทย ใน ด้านจำนวนโหนดที่ใช้, เวลาที่ใช้ในการฝึกโครงข่าย, เวลาที่ใช้ในการ แยกแยะตัวอักษรต่อ 1 ตัว และประสิทธิภาพในการรู้จำ.....52

5.10 เปรียบเทียบประสิทธิภาพของแบบจำลองชนิดต่างๆ ในการแยกแยะตัว อักษรไทย โดยข้อมูลที่มีความชัดเจนแตกต่างกัน.....53

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ

ภาพที่ หน้า

1.1 แสดงการทำงานของเครื่องอ่านตัวอักษรด้วยแสง.....2

2.1 แสดงลักษณะของเซลประสาททางชีววิทยา.....8

2.2 แสดงตัวแบบของเซลประสาทเทียม..... 9

2.3 แสดงลักษณะของ Binary Step โดยใช้ค่า Threshold..... 10

2.4 แสดงลักษณะของ Binary Sigmoid 11

2.5 แสดงลักษณะของ Bipolar Sigmoid..... 11

2.6 แสดงตัวอย่างของโครงข่ายแบบชั้นเดียว.....12

2.7 แสดงตัวอย่างของโครงข่ายแบบหลายชั้น.....13

2.8 แสดงความสัมพันธ์ของข้อมูลเชิงเรขาคณิตของตัวอย่างเพอร์เซพตรอน.....15

2.9 แสดงกราฟเส้นแบ่งของเขตที่ได้จากสอนเพอร์เซพตรอน..... 16

2.10 (a) แสดงปัญหาของ XOR 16

2.10 (b) แสดงการเพิ่มชั้นของเพอร์เซปตรอน 17

2.10 (c) แสดงเส้นแบ่งที่ได้จากสอนรูปที่ 2.10(a) 17

3.1 สถาปัตยกรรมของ ART.....18

3.2 (a) แสดงการเรียนอักษร ด ของชุดน้ำหนัก t.....21

3.2 (b) แสดงชุดน้ำหนัก b ของ ART Node ซึ่งเก็บ pattern ของตัวอักษร ด.....21

3.3 (c) แสดงชุดน้ำหนักใหม่เมื่อผ่านการเรียนรู้.....22

3.4 แสดงการทำงานของ Supplemental Unit (G1,G2) 24

3.5 แสดงตัวอักษรที่มีความคล้ายคลึงกันมากคือ ดตคฅ.....28

3.6 แสดงการเรียนรู้ตัวอักษร ฅ โดยใช้ ART 29

4.1 ผลจากการฝึกโหนดของตัวอักษร ฅ โดยใช้ ARTS 32

4.2 แสดงอินพุทตัวอักษร 1 และโหนดชนะของตัวอักษรเลข ๖ และจุดที่ intersect 35

5.1 แสดงโหนดของตัวอักษร ก 40

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ (ต่อ)

ภาพที่ **หน้า**

5.2 แสดงโหนด ใ และ ไ ซึ่งเป็นโหนดลำดับที่ 84, 240 และ 85, 179 42

5.3 แสดงอินพุท ไ1 และโหนดชนะของตัวอักษร ไ ลำดับที่ 85 และจุดที่ intersect 45

5.4 แสดงอินพุท ไ1 และโหนดชนะของตัวอักษร ไ ลำดับที่ 179 และจุดที่ intersect 46

5.5 อินพุท ไ14 และโหนดชนะของตัวอักษร ใ ลำดับที่ 84 และจุดที่ intersect 48

6.1 แสดงโหนดชนะคือตัวอักษร ฤ และอินพุทคือตัวอักษร ถ.....49

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อภิธานศัพท์

Adaline เซลประสาทเทียมชนิดหนึ่งซึ่งถูกพัฒนาโดย Bernard Winrow ในปี 1960

ART network โครงข่ายประสาทเทียมที่พัฒนาโดยใช้ทฤษฎี adaptive resonance โดย Capenter และ Grossberg

Self-Organizing Map (SOM) network โครงข่ายประสาทเทียมชนิดหนึ่งที่พัฒนาโดย Teuvo Kohonen ซึ่งในการสอนโครงข่ายจะไม่มีเป็นหมายให้กับโครงข่าย

Perceptron โครงข่ายชนิดมีเพียงชั้นเดียว และสามารถแก้ปัญหาได้เพียงปัญหาที่เป็นแบบเชิงเส้นเท่านั้น (linear separability)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

1.1 ความเป็นมาของงานวิจัย

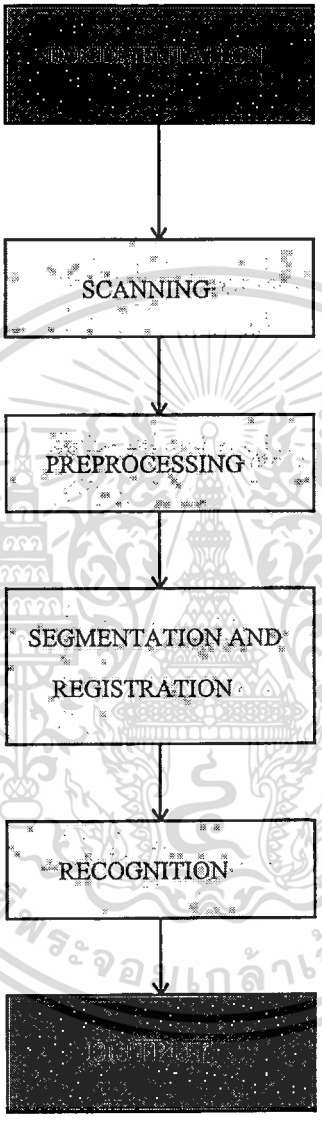
ปัจจุบันคอมพิวเตอร์ได้เข้ามามีบทบาทที่สำคัญในชีวิตประจำวันมากขึ้นเรื่อยๆ เทคโนโลยีทางด้านนี้ได้มีการแข่งขันและพัฒนาไปอย่างรวดเร็วทั้งทางฮาร์แวร์ และซอฟแวร์ ซึ่งงานส่วนใหญ่ที่ใช้คอมพิวเตอร์คืองานจัดเก็บข้อมูล เพื่อนำข้อมูลไปประมวลผล หรือการจัดเก็บข้อมูลเพื่อนำไปใช้ต่อไป

ได้มีการศึกษาและทำการวิจัยเกี่ยวกับความพยายามในการสอนหรือฝึกสอนให้คอมพิวเตอร์มีความสามารถหรือมีความรู้ในการพิจารณาแยกแยะอักษรต่างๆให้ถูกต้องกันอย่างมากมายทั้งในและต่างประเทศ โดยได้มีการเรียนรู้และสร้างทฤษฎีของการรู้จำตัวอักษร ซึ่งในแต่ละวิธีก็จะมีความเหมาะสมกับปัญหาที่เกิดขึ้นเฉพาะด้าน โดยสามารถแบ่งวิธีการรู้จำได้โดยการพิจารณาลักษณะอักษรในรูปแบบที่แตกต่างกัน เช่นการรู้จำอักษรตัวพิมพ์และการรู้จำอักษรลายมือเขียน หรืออาจแบ่งโดยการพิจารณาจากวิธีในการรู้จำตัวอักษรโดยวิธีการซ้อนทับกัน (Matching) การรู้จำตัวอักษรโดยวิธีการพิจารณาโครงสร้างทางกายภาพของตัวอักษร ซึ่งแต่ละวิธีก็มีขีดความสามารถของการเรียนรู้จำที่แตกต่างกันไป

เครื่องอ่านตัวอักษรด้วยแสง (Optical Character Reader) เป็นอุปกรณ์หนึ่ง ซึ่งช่วยให้การจัดเก็บข้อมูลตัวอักษรลงในคอมพิวเตอร์ได้อย่างมีประสิทธิภาพ และรวดเร็วมากขึ้น โดยมีหลักการทำงานดังรูปที่ 1.1 คือ การฉายแสงลงบนแผ่นกระดาษแล้วนำปริมาณแสงที่สะท้อนกลับมาเปลี่ยนเป็นสัญญาณไฟฟ้าและสัญญาณไฟฟ้าจะถูกนำไปเปลี่ยนเป็นรหัสข้อมูลทาง digital แล้วนำไปเก็บไว้ในหน่วยความจำของคอมพิวเตอร์ เพื่อส่งผลให้คอมพิวเตอร์ทำการวิเคราะห์ต่อไปว่าเป็นตัวอักษรอะไร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 1.1 แสดงการทำงานของเครื่องอ่านตัวอักษรด้วยแสง



ในต่างประเทศมีการวิจัยระบบรู้จำตัวอักษรภาษาต่างๆมากมาย แต่วิธีต่างๆที่มีการประยุกต์ใช้ในภาษาต่างประเทศ ก็จะเหมาะสมกับคุณลักษณะของภาษานั้นๆโดยเฉพาะ ยังไม่สามารถนำวิธีการเดียวกันนั้นมาใช้ในการจดจำอักษรภาษาไทยได้โดยตรงทันที จึงมีการศึกษาวิจัยให้คอมพิวเตอร์สามารถที่จะรับรู้และเข้าใจตัวอักษรภาษาไทยกันอย่างกว้างขวาง ซึ่งงานวิจัยเหล่านี้ล้วนเป็นแนวทางในการพัฒนาระบบให้ดียิ่งๆขึ้นไป งานวิจัยที่เสนอในวิทยานิพนธ์นี้จึงเน้นการปรับปรุงและพัฒนาประสิทธิภาพให้คอมพิวเตอร์สามารถแยกแยะและจดจำตัวอักษรให้ถูกต้อง ลด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ความผิดพลาดในการจดจำตัวอักษรและลดเวลาในขั้นตอนของการเรียนรู้โดยการนำโครงข่ายประสาทเทียมแบบจำลองอะแดปทีฟรีโซแนนซ์เทียรี (Adaptive Resonance Theory Neural Network) มาประยุกต์ใช้ในการแก้ปัญหา

1.2 ผลงานวิจัยที่ผ่านมา

1. ผลงานวิจัยของนายทวี เปรมรัตนชัย , รศ. ดร. ชม กิ้มปาน และผศ.ดร.บุญธีร์ เครือตราชู [5] เรื่องการรู้จำอักขระตัวพิมพ์ภาษาไทยด้วยวิธีโครงข่ายนิวรอนแบบแบคพรอบพาเกชั่น เสนอการฝึกสอนโครงข่ายนิวรอนด้วยอัลกอริทึมแบบแบคพรอบพาเกชันโดย รับข้อมูลจากตัวอักษรที่สแกนจากเครื่องสแกนภาพและนำข้อมูลนั้นมาผ่านโปรแกรมแปลงภาพไฟล์กราฟฟิกไปเป็นไบนารีไฟล์ จากนั้นจะผ่านโปรแกรมแยกอักขระเพื่อนำมาใช้เป็นข้อมูลอินพุทของโปรแกรมนิวรอนที่ใช้ฝึก และทำการทดสอบประสิทธิภาพการรู้จำ

2. ผลงานวิจัยของนายอภิรักษ์ จิรายุสกุล และ ผศ.ดร.บุญธีร์ เครือตราชู [7] เรื่องการวิเคราะห์ตัวพิมพ์อักษรภาษาไทยโดยใช้ CPN โดยนำเสนอการนำโครงข่ายประสาทเทียม Counterpropagation Network(CPN) มาประยุกต์ใช้ในการจดจำตัวอักษรภาษาไทย และได้มีการเปลี่ยนแปลงโครงสร้างหลักของ CPN 2 ส่วน คือส่วนวิธีการเรียนรู้โดยใช้วิธีการแยกสอนตัวอย่างข้อมูลทีละกลุ่ม ส่วนที่ 2 คือให้โครงสร้างชั้นในของคอมเพททิทิพมีความสามารถในการเพิ่มจำนวนโหนด และทำการฝึกสอนโครงข่ายประสาทเทียมและทดสอบประสิทธิภาพการรู้จำ

3. ผลงานวิจัยของนางสาวอัญชลี วานิชทวีวัฒน์ และผศ.ดร. บุญธีร์ เครือตราชู[6] เรื่องการจดจำอักขระภาษาไทยตัวพิมพ์โดยโครงข่ายประสาทเทียมแบบเซลฟ์ออร์แกนไนซิงแมปซ์ โดยนำเสนอการนำโครงข่ายประสาทเทียมแบบ Self-Organizing Maps (SOM) และ Adaptive Self-Organizing Map มาประยุกต์ใช้ในการจดจำตัวอักษรภาษาไทย และได้มีการปรับเปลี่ยนโครงสร้างบางส่วน เพื่อให้เหมาะสมในการจดจำตัวอักษรภาษาไทย และทำการฝึกสอนโครงข่ายประสาทเทียมและทดสอบประสิทธิภาพการรู้จำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.3 แนวทางของวิทยานิพนธ์

วิทยานิพนธ์ฉบับนี้ นำเสนอวิธีการวิเคราะห์ตัวอักษรพิมพ์ภาษาไทย โดยใช้โครงข่ายประสาทเทียม อะแดปทีฟรีโซแนนซ์เทียรี(ART) ซึ่งเป็นวิธีที่ง่ายและการฝึกโครงข่ายใช้เวลาน้อยมาก หลักการของโครงข่ายจะทำการเรียนรู้ในลักษณะที่ไม่มีผู้ฝึกสอน (Unsupervised Learning) โดยขณะทำการฝึกตัวโครงข่ายจะทำการปรับตัวเอง สร้างค่าน้ำหนักเพื่อประเมินว่าค่าอินพุทที่เข้ามานั้นควรจะแยกแยะให้อยู่ในกลุ่มใด และนำความรู้จากค่าน้ำหนักที่เกิดจากการฝึกมาใช้ในการแยกแยะอินพุทในการใช้งานต่อไป งานวิจัยนี้ ได้ทำการปรับปรุงโครงสร้างของอัลกอริทึม ในส่วนของการเรียนรู้ คือเปลี่ยนวิธีการเรียนรู้ให้อยู่ในรูปแบบของการเรียนรู้แบบไม่มีผู้ฝึกสอน (Supervised Learning) และทำการเปลี่ยนแปลงโครงสร้างในส่วนการปรับค่า Threshold เพื่อเพิ่มประสิทธิภาพในการจดจำ ลดจำนวนโหนด และลดระยะเวลาในการฝึกโครงข่าย

1.4 วัตถุประสงค์ของงานวิจัย

1. เพื่อศึกษาแนวทางการนำโครงข่ายประสาทเทียมมาใช้ในการวิเคราะห์ตัวอักษรภาษาไทย
2. เพื่อศึกษาคุณสมบัติและขั้นตอนการนำ ART มาประยุกต์ใช้งาน
3. เพื่อศึกษาปัญหาที่เกิดขึ้นจากการนำตัวแบบ ART มาใช้ในการวิเคราะห์ภาษาไทย และหาแนวทางการแก้ปัญหาที่เกิดขึ้น
4. เพื่อศึกษาประสิทธิภาพในการแยกแยะตัวอักษรภาษาไทยของ ART

1.5 โครงสร้างของวิทยานิพนธ์

งานวิจัยในวิทยานิพนธ์นี้ได้เน้นวิจัยเฉพาะ ส่วนของการวิเคราะห์ตัวอักษรภาษาไทย โดยนำเอาโครงข่ายประสาทเทียมแบบจำลอง Adaptive Resonance Theory มาประยุกต์ใช้ในงานจดจำภาษาไทย แบ่งโครงสร้างวิทยานิพนธ์ออกเป็น 6 ส่วน คือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2 กล่าวถึงทฤษฎีเบื้องต้นของโครงข่ายประสาทเทียม โดยอธิบายลักษณะการทำงานของสมองมนุษย์ และอธิบายถึงลักษณะการเรียนรู้และความสามารถในการจดจำของโครงข่ายประสาทเทียม

บทที่ 3 โครงข่ายประสาทเทียมแบบ ART อธิบายถึงโครงสร้างของโครงข่ายประสาทเทียมของ ART วิธีการปรับค่าน้ำหนัก และอัลกอริทึมที่ใช้ในการฝึกสอน และทดลองนำ ART มาใช้ในการเรียนรู้ตัวอักษรภาษาไทย แสดงผลการทดลอง ตลอดจนปัญหา และแนวทางในการแก้ไขปัญหา ซึ่งแนวทางในการแก้ไขปัญหาคือการปรับเปลี่ยนวิธีการเรียนรู้จากวิธีการเรียนรู้ในลักษณะที่ไม่มีผู้ฝึกสอน (Unsupervised learning) เป็นการเรียนรู้ในลักษณะที่มีผู้ฝึกสอน(Supervised learning)

บทที่ 4 โครงข่ายประสาทเทียมแบบ ARTS อธิบายถึงวิธีการเปลี่ยนโครงสร้างของอัลกอริทึมของ ART ให้การเรียนรู้ในลักษณะที่มีผู้ฝึกสอน(Supervised learning) เรียกวิธีนี้ว่า ARTS และทดลองนำ ARTS มาใช้ในการเรียนรู้ตัวอักษรภาษาไทย และแสดงผลการทดลอง ตลอดจนแสดงให้เห็นถึงปัญหา และแนวทางในการแก้ไขปัญหา ซึ่งแนวทางในการแก้ไขปัญหาคือการนำ ARTS มาปรับเปลี่ยนวิธีการปรับค่า ρ

บทที่ 5 กล่าวถึงผลการทดลองใช้ค่าเทรสโฮสแก้ปัญหา ARTS ในการแยกแยะตัวอักษรภาษาไทย และแสดงให้เห็นผลการเปรียบเทียบระหว่างการใช้โครงข่ายประสาทเทียมแบบจำลองชนิดต่างๆ ในด้านประสิทธิภาพการรู้จำและ จำนวนโหนดที่ใช้

บทที่ 6 สรุปผลและแนวทางในการพัฒนาในอนาคต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

ทฤษฎีเบื้องต้นของโครงข่ายประสาทเทียม

ปัจจุบันคอมพิวเตอร์ได้เข้ามามีบทบาทในการทำงานเป็นอย่างมาก ซึ่งงานบางอย่างคอมพิวเตอร์จะสามารถทำได้รวดเร็วกว่ามนุษย์หลายเท่า ตัวอย่างงานเช่น การจัดการด้านฐานข้อมูล, การควบคุมเครื่องจักรกล, งานประมวลด้านการคำนวณเป็นต้น แม้ว่าคอมพิวเตอร์มีความสามารถในการประมวลผลได้สูงและถูกนำไปใช้ในงานหลายๆด้านได้เป็นอย่างดี แต่งานบางอย่างที่ทำด้วยคอมพิวเตอร์เมื่อนำผลมาเปรียบเทียบผลกับมนุษย์แล้ว มนุษย์สามารถทำงานได้ดีกว่าและเร็วกว่ามาก เช่นการจดจำหน้าภาพหน้าคน การจดจำเสียง ซึ่งถ้าเป็นคอมพิวเตอร์แล้วอาจจะไม่สามารถจดจำได้ อีกทั้งเวลาที่ใช้ในการประมวลผลก็ใช้เวลานานมากด้วย

ความแตกต่างระหว่างมนุษย์และคอมพิวเตอร์ในการทำงานจากตัวอย่างที่กล่าวข้างต้นเนื่องมาจากสมองมนุษย์มีความแตกต่างจากคอมพิวเตอร์ กล่าวคือสมองมนุษย์ประกอบด้วยเซลประสาทจำนวนมากมายเชื่อมต่อกัน ซึ่งเซลเหล่านี้ทำหน้าที่เป็นหน่วยประมวลผลให้กับสมองมนุษย์ การส่งสัญญาณของเซลประสาทระหว่างเซลหนึ่งสู่อีกเซลหนึ่งใช้เวลาเร็วมากคือประมาณสิบส่วนล้านวินาที ซึ่งรูปแบบการส่งสัญญาณทำในลักษณะขนานกัน และวิธีการแก้ปัญหาของสมองมนุษย์ใช้ประสบการณ์ที่เกิดจากการเรียนรู้ในครั้งอดีตนำมาวิเคราะห์เพื่อแก้ปัญหานั้นๆ ขณะที่สถาปัตยกรรมของคอมพิวเตอร์ประกอบด้วยหน่วยประมวลเพียงหน่วยเดียว และการทำงานใช้ชุดคำสั่งสั่งงานเป็นแบบลำดับขั้น อีกทั้งวิธีการแก้ปัญหาต่างๆนั้นจะต้องทราบลำดับขั้นตอนการทำงานที่แน่นอนของปัญหานั้น

ดังนั้น เมื่อเราต้องการให้คอมพิวเตอร์สามารถจัดการกับปัญหาในลักษณะที่กล่าวข้างต้น จึงได้มีการนำสถาปัตยกรรมของสมองมนุษย์มาเป็นตัวแบบในการประมวลผลด้วยคอมพิวเตอร์ ซึ่งเรียกว่า Artificial Neural Systems หรือโครงข่ายประสาทเทียม (Neural Network)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โครงข่ายประสาทเทียมประกอบด้วยหน่วยประมวลผลแบบง่ายๆจำนวนมหาศาลที่เชื่อมต่อกันซึ่งเรียกหน่วยนี้ว่า นิวรอน (Neurons) เซล (Cell) หรือ โหนด (Nodes) และเรียกส่วนที่เชื่อมต่อกันระหว่างโหนดนี้ว่า ส่วนเชื่อมต่อ (Interconnection) ส่วนเชื่อมต่อจะเก็บความรู้ ซึ่งใช้ในการแก้ปัญหานั้นๆ ของโครงข่าย ซึ่งความรู้ที่ใช้ในการปัญหานั้นจะได้มาจากการเรียนรู้ตัวอย่างของปัญหา เราสามารถเปรียบเทียบโครงข่ายประสาทเทียมกับสมองมนุษย์ได้ในแง่ของการเก็บข้อมูลต่างๆในรูปของแพตเทิร์นของส่วนเชื่อมต่อ (Pettern of Interconnections) และลักษณะการแก้ปัญหาด้วยวิธีการเรียนตัวอย่างซึ่งเทียบได้กับการสะสมประสพการณ์ของมนุษย์

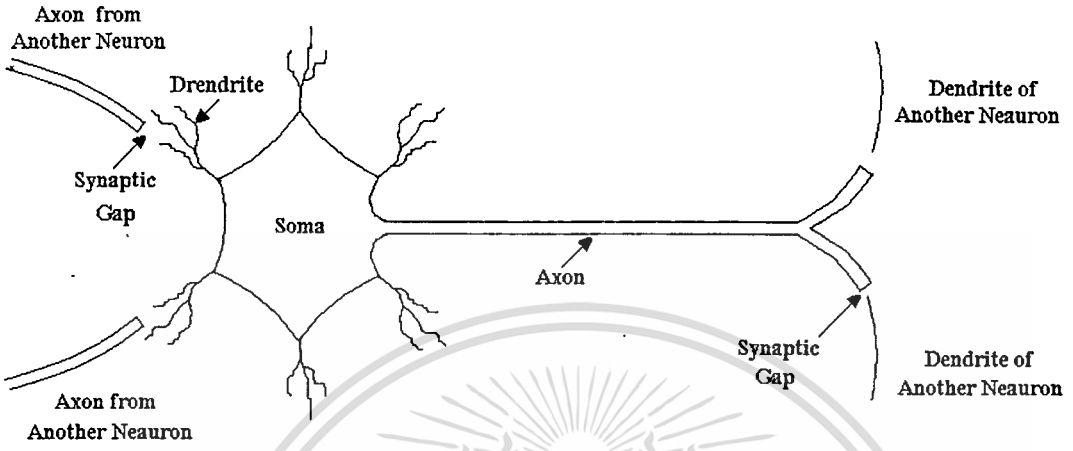
โครงข่ายประสาทเทียมได้ถูกนำมาประยุกต์ในงานด้านต่างๆหลายด้าน เช่น การเก็บข้อมูล(Storing), การเรียกข้อมูลมาใช้งาน (Recall data) และ การแบ่งกลุ่มแพตเทิร์นที่มีลักษณะเหมือนกัน (pattern Classification) เป็นต้น โครงข่ายประสาทเทียมมีคุณสมบัติซึ่งเรียกว่า Generalization กล่าวคือ โครงข่ายสามารถวิเคราะห์หรือสร้างคำตอบสำหรับปัญหาซึ่งมีลักษณะที่เหมือนหรือคล้ายกับตัวอย่างที่นำมาสอนได้อย่างมีประสิทธิภาพ โดยที่ปัญหานั้นไม่เคยถูกนำมาสอนให้กับโครงข่ายเลย

2.1 ทฤษฎีเซลประสาทเบื้องต้น

ระบบประสาทมนุษย์ประกอบด้วยเซลประสาทประมาณ 10^{11} เซล และมีการเชื่อมต่อกันระหว่างเซลประมาณ 10^{15} จุด ลักษณะเซลประสาททางชีวภาพประกอบด้วย 3 ส่วนหลักคือ เดนไดร์ (Dendrites), โซมา (Soma) หรือตัวเซล (Cell body) และแอคซอน (Axon) ส่วนที่เชื่อมระหว่างตัวเซลกับแอคซอนของเซลอื่นๆที่อยู่รอบๆข้างเรียกว่า ไซแนปส์ (Synapse) โดยตัวเดนไดรซ์จะรับสัญญาณจากเซลประสาทที่อยู่รอบๆข้างผ่านทางไซแนปส์ ด้วยปฏิกิริยาทางเคมี ซึ่งสัญญาณที่เข้ามานี้คือ ประจุอิเล็กตรอนโดยปฏิกริยาเคมีที่เกิดขึ้นจะทำการปรับเปลี่ยนความถี่ของสัญญาณที่เข้ามา ตัวเซลทำหน้าที่รวมสัญญาณที่เข้ามาแล้วทำการส่งสัญญาณออกให้กับแอคซอนเพื่อผ่านต่อให้กับเซลอื่นๆ ซึ่งสัญญาณที่ออกจากตัวเซลมีลักษณะเป็นแบบสัญญาณกระตุ้นให้กับเซลอื่น กล่าวคือ ถ้าค่าสัญญาณที่ส่งออกจากตัวเซลมีจำนวน 100 ครั้งต่อวินาทีถือว่าค่าสัญญาณที่ส่งออกเป็นสถานะการกระตุ้น (Fire) โดยทั่วไปเราสนใจสัญญาณที่ออกจากเซลประสาทในรูปของสถานะการกระตุ้นและไม่กระตุ้น (Not Fire) ให้กับเซลที่อยู่รอบข้างมากกว่าสนใจเงื่อนไขที่เกี่ยวข้องที่ทำให้เซลประสาทเกิดสถานะนี้ขึ้นมา รูปที่ 2.1 แสดงลักษณะของเซลประสาททางชีววิทยา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2.1 แสดงลักษณะของเซลทางประสาทชีววิทยา



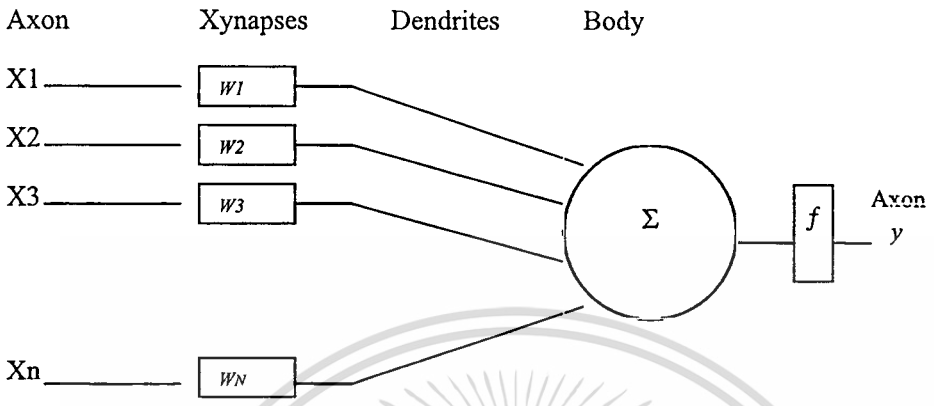
แสดงลักษณะของนิวรอนเซล (Neural Cell) ของเซลประสาท

2.2 ตัวแบบของเซลประสาทเทียม (Aritificial Neural Model)

ตัวเซลประสาทเทียมที่ใช้ใน Artificial Neural Network มีคุณลักษณะพื้นฐานเหมือนกับเซลประสาททางชีววิทยา รูปที่ 2.1 แสดงลักษณะของตัวแบบเซลประสาทเทียมที่ใช้กันโดยทั่วไป ซึ่งใช้พื้นฐานจากตัวแบบที่เสนอโดย MCCulloch และ Pitt ในปี 1943 จากรูปที่ 2.2 เซลประสาทเทียมรับสัญญาณ X_1-X_n จากเซลรอบข้าง โดยไซแนปส์หรือส่วนเชื่อมต่อจะทำการปรับเปลี่ยนหรือถ่วงน้ำหนักค่าสัญญาที่เข้ามาด้วย w_1-w_n สมการที่ 2.1 แสดงการหาค่าผลรวมสัญญาณที่เข้าสู่เซลประสาทเทียมโหนดที่ 1 ซึ่งเรียกว่าค่า Net Input และ Y คือค่าสัญญาณที่ออกจากเซลประสาทเทียม ซึ่งจะปรากฏที่แอคซอนโ ดยค่าสัญญาณนี้จะส่งผ่านทางฟังก์ชั่นกระตุ้น (Activation Function) $f()$ ซึ่งฟังก์ชันนี้ทำหน้าที่เลือกผ่านค่าสัญญาณให้กับแอคซอน โดยจะทำการย่อขนาดของผลรวมสัญญาณที่ออกจากตัวเซลให้อยู่ในช่วง 0 ถึง 1 แต่ในบางครั้งอาจไม่มีการใช้ฟังก์ชั่นการกระตุ้นก็ได้ สมการที่ 2.2 แสดงการหาค่าฟังก์ชั่นกระตุ้นของสมการที่ 2.1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2.2 แสดงตัวแบบของเซลประสาทเทียม



$$net_i = \sum x_j w_{ij} \quad \text{.....(2.1)}$$

$$y_i = f_i(net_i) \quad ,i = 1,2, ..., m \quad \text{.....(2.2)}$$

โดยที่ x_j : ค่าสัญญาณลำดับที่ j ที่เข้าสู่เซลประสาทเทียมโหนดที่ i

W_{ij} : ค่าน้ำหนักสัญญาณลำดับที่ j ที่เข้าสู่เซลประสาทเทียมโหนดที่ i

net_i : ค่า Net Input ของเซลประสาทเทียมโหนดที่ i

$f_i()$: ฟังก์ชั่นกระตุ้นของเซลประสาทเทียมโหนดที่ i

y_i : ค่าสัญญาณที่ออกจากเซลประสาทเทียมโหนดที่ i

m : จำนวนเซลประสาทเทียมของโครงข่าย

ค่าของส่วนเชื่อมต่อของเซลประสาทเทียมอาจมีค่าเป็นบวก ลบ หรือศูนย์ก็ได้ ดังนั้นในการปรับแต่งค่าสัญญาณที่เข้าสู่เซลประสาทด้วยค่าน้ำหนักของส่วนเชื่อมต่อ ถ้าข้อมูลที่เข้าสู่เซลประสาทเทียมมีค่าเป็นบวก และค่าของส่วนเชื่อมต่อเป็นบวกด้วยก็จะก่อให้เกิดสภาพเสริมกัน (Excitatory) และถ้าค่าของส่วนเชื่อมต่อมีค่าเป็นลบ ก็จะลดค่าความเข้มของข้อมูลที่เข้ามานั้น ส่วนในกรณีที่ค่าของส่วนเชื่อมต่อเท่ากับศูนย์หมายถึง ส่วนเชื่อมต่อตรงส่วนนั้นไม่มีการทำงาน

ค่าผลลัพธ์ที่ได้จากฟังก์ชั่นกระตุ้นนี้ทำให้เซลประสาทเทียมมีลักษณะเหมือนกับเซลประสาททางชีววิทยา กล่าวคือค่าผลลัพธ์ที่ได้จากฟังก์ชั่นกระตุ้นเป็นค่าที่อยู่ในช่วง 0 ถึง 1 ถ้าค่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สัญญาณที่ออกจากเซลประสาททางชีววิทยามีลักษณะเป็นสถานะของการกระตุ้นเราจะสามารถแทนความหมายสถานะการกระตุ้นของเซลประสาทด้วยค่า 1 และการไม่กระตุ้น 0 รูปแบบฟังก์ชั่นกระตุ้นที่นิยมใช้มีดังต่อไปนี้

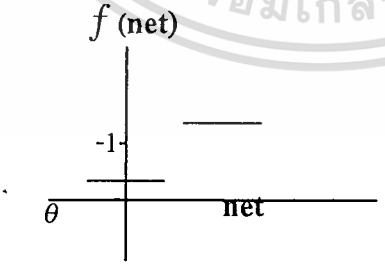
1. Step Function ฟังก์ชั่นนี้นิยมใช้ในโครงข่ายแบบชั้นเดียว (Single Layer) ซึ่งใช้ในการแปลงค่า Net Input ที่อยู่ในรูปของค่าต่อเนื่องให้อยู่ในรูปของไบนารี (Binary) คือ 1 และ 0 หรือไบโพลล่า (Bipolar) คือ 1 และ -1 ฟังก์ชั่นนี้จะใช้ค่าเทรสโฮลด์ (Theshold) θ ในการกำหนดการแปลงค่า ซึ่งเรียกฟังก์ชั่นนี้ว่า Threshold Function หรือ Heavisde Function ตัวอย่างโครงข่ายที่ใช้ฟังก์ชั่นนี้ได้แก่ Adaline สมการที่ 2.3 แสดงสมการของ Threshold Function

$$f(\text{net}) = \begin{cases} 1 & \text{if net} \geq \theta \\ 0 & \text{if net} < \theta \end{cases} \quad \text{.....(2.3)}$$

โดยที่ net : ค่า Net Input ที่ของเซลประสาทเทียม

θ : ค่าเทรสโฮลด์ที่ใช้ในการแปลงค่าของเซลประสาทเทียม

รูปที่ 2.3 แสดงลักษณะของ Binary Step โดยใช้ค่า Threshold

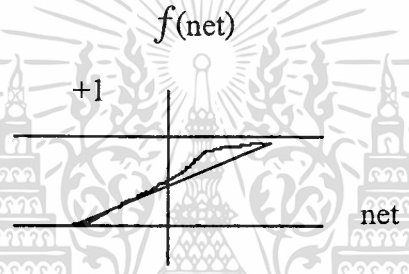


2. Sigmoid Function (*S*-shaped curves) ฟังก์ชั่นนี้ปกติแล้วนิยมนำมาใช้ในการแปลค่า Net Input ให้อยู่ในรูปของช่วง 0 ถึง 1 ซึ่งเรียกว่า Binary Sigmoid ฟังก์ชั่น หรืออยู่ในช่วง -1 ถึง 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

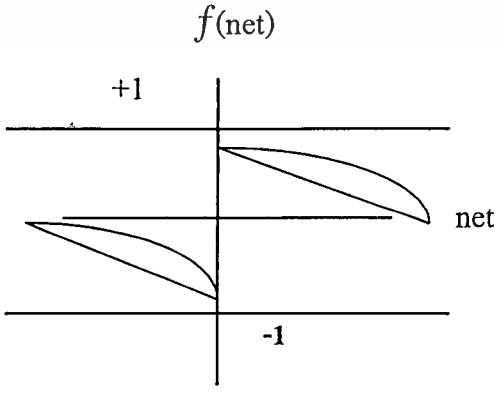
ซึ่งเรียกว่า Bipolar Sigmoid ฟังก์ชั่นที่นำมาใช้ในการแผลงค่าเป็น Sigmoid Function โดยทั่วไปแล้ว Logistic Function และ Hyperbolic Tangent Function แสดงในสมการที่ 2.4 และ 2.5 ตามลำดับ ซึ่งฟังก์ชั่นเหล่านี้มีประโยชน์อย่างมากในการสอนโครงข่ายแบบแพร่ย้อนกลับ (Backpropagation Network) รูปที่ 5 และ 6 แสดงตัวอย่างของ Binary Sigmoid และ Bipolar Sigmoid ด้วย Logistic Function และ Hyperbolic Tangent Function ตามลำดับ

รูปที่ 2.4 แสดงลักษณะของ Binary Sigmoid



$$f(\text{net}) = 1/(1+e^{-\text{net}}) \quad \text{.....(2.4)}$$

รูปที่ 2.5 แสดงลักษณะของ Bipolar Sigmoid



$$f(\text{net}) = (e^{\text{net}}-e^{-\text{net}}) / (e^{\text{net}}+e^{-\text{net}}) \quad \text{.....(2.5)}$$

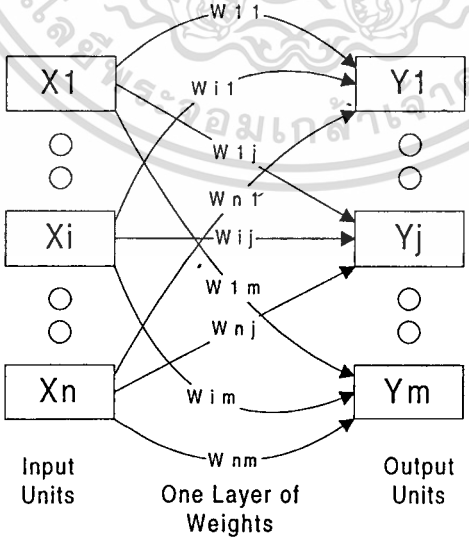
โดยที่ net : **ค่า Net Input ที่ของเซลประสาทเทียม**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3 สถาปัตยกรรมของโครงข่ายประสาทเทียม (Neural Network Architecture)

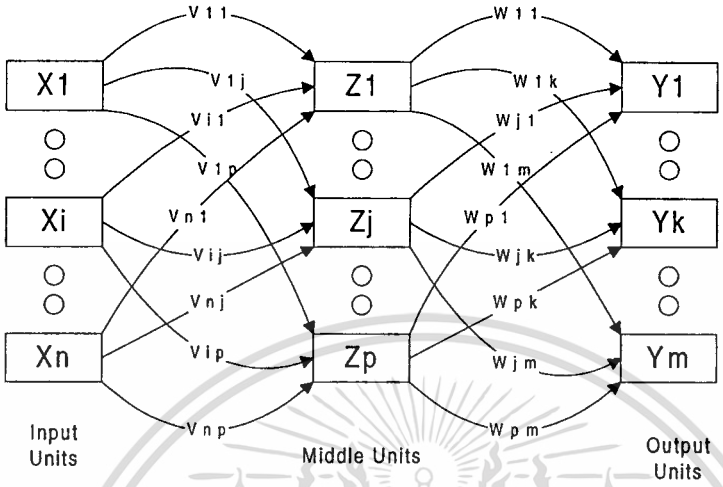
โครงข่ายประสาทเทียมประกอบด้วยเซลประสาทเทียมหรือโหนดจำนวนมาก ที่เชื่อมต่อกัน ซึ่งการเชื่อมต่อจะแบ่งออกเป็นกลุ่มย่อยเรียกว่า ชั้น (Layer) โดยทั่วไปแล้ว แบ่งออกเป็น 2 แบบคือ โครงข่ายแบบชั้นเดียว (Single Layer) และโครงข่ายแบบหลายชั้น (Multilayer) การกำหนดจำนวนชั้นของโครงข่ายนี้ส่วนที่เป็นหน่วยรับข้อมูลเข้า (Input Unit) จะไม่ถูกนับด้วยเนื่องจากเป็นส่วนที่ไม่มีการคำนวณ (หนังสือบางเล่มถือว่าหน่วยข้อมูลเข้านี้เป็นชั้นของโครงข่ายด้วย) ดังนั้นเราอาจกล่าวได้ว่าจำนวนชั้นของโครงข่ายคือ จำนวนชั้นของส่วนเชื่อมต่อที่ถูกถ่วงน้ำหนัก โดยปกติแล้ว โครงข่ายแบบหลายชั้นสามารถแก้ปัญหาที่มีความซับซ้อนได้ดีกว่าโครงข่ายแบบชั้นเดียว และในบางปัญหานั้นไม่สามารถแก้ปัญหาด้วยโครงข่ายแบบชั้นเดียวได้ รูปที่ 2.6 แสดงลักษณะการต่อของโครงข่ายชั้นเดียว ซึ่งตัวแบบที่สถาปัตยกรรมแบบนี้ได้แก่ BAM, Hopfield ส่วนรูปที่ 2.7 แสดงสถาปัตยกรรมแบบหลายชั้นซึ่งตัวแบบได้แก่ Backpropagation, Self Organizing Maps, Couterpropagation

รูปที่ 2.6 แสดงตัวอย่างของโครงข่ายแบบชั้นเดียว



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2.7 แสดงตัวอย่างของโครงข่ายแบบหลายชั้น



2.4 การฝึกสอนโครงข่ายประสาทเทียม

โครงข่ายประสาทเทียมประกอบด้วยเซลประสาทหลายๆเซล ทำงานร่วมกันเพื่อแก้ปัญหาต่างๆ โดยโครงข่ายประสาทเทียมจะถูกฝึกให้เรียนรู้จากตัวอย่างเพื่อจะได้มีความรู้ที่จะนำไปแก้ปัญหาต่อไป แบบจำลองต่างๆ จะมีขั้นตอนในการฝึกที่แตกต่างกัน การฝึกโครงข่ายประสาทเทียมสามารถแบ่งตามลักษณะของการเรียนรู้ได้ 2 ประเภทคือ

1. การเรียนรู้แบบมีผู้สอน (Supervised Learning) ข้อมูลจะต้องประกอบด้วยตัวอย่างข้อมูลที่เราต้องการสอน และผลลัพธ์ที่เราต้องการให้โครงข่ายสร้างเมื่อมีการนำข้อมูลเข้าในลักษณะเดียวกันมาเป็นข้อมูลเข้า (Input) ขณะสอน โครงข่ายชนิดนี้จะกำหนดค่าผลลัพธ์เป้าหมายให้กับข้อมูลเข้าแต่ละตัว โครงข่ายจะนำค่าที่ผิดพลาดที่ได้จากการคำนวณโครงข่ายกับผลลัพธ์ที่ต้องการให้โครงข่ายสร้างนั้นมาเป็นค่าน้ำหนัก เพื่อปรับให้ได้ค่าข้อมูลออกตรงตามผลลัพธ์เป้าหมายที่ต้องการ ตัวอย่างแบบจำลองนี้เช่น Backpropagation, Adaline, Perceptron

2. การเรียนรู้แบบไม่มีผู้สอน (Unsupervised Learning) จะสอนโครงข่ายด้วยการป้อนแบบข้อมูลเข้าอย่างต่อเนื่องเพียงอย่างเดียว โดยไม่มีการส่งค่าผลลัพธ์เป้าหมายให้กับข้อมูลเข้าแต่ละตัว การปรับค่าน้ำหนักใช้ข้อมูลที่นำมาสอนเป็นตัวปรับค่า โดยค่าน้ำหนักจะปรับตามกลุ่มที่ข้อมูลเข้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มีรูปแบบคล้ายๆ กัน ตัวอย่างแบบจำลองนี้เช่น Counterpropagation (CPN) และ Adaptive Resonance Theory Neural Network (ART)

การวิจัยในวิทยานิพนธ์ฉบับนี้จะเป็นแบบจำลองโครงข่ายประสาทเทียม ART ซึ่งมีโครงสร้างแบบ มัลติเลเยอร์ (Multilayer) ซึ่งมีรูปแบบในการเรียนรู้แบบ Unsupervised Learning และใช้อัลกอริทึมแบบคลัสเตอร์ นำมาทดสอบการรู้จำตัวอักษรภาษาไทย และได้ทำการเปลี่ยนแปลงโครงสร้างโดยเปลี่ยนรูปแบบการเรียนรู้ให้เป็นแบบ Supervised และเปลี่ยนขั้นตอนของการเพิ่มและลดขนาดของ Threshold ในช่วงของการฝึก

2.5 การแบ่งแบบเชิงเส้น (Linear Separability)

ปกติทั่วไปแล้ว ผลลัพธ์ที่ออกจากเซลประสาทเทียมเราต้องการคำตอบในรูปของตรรกศาสตร์ คือใช่หรือไม่ใช่ กล่าวคือคำตอบใช่เมื่อข้อมูลที่เข้าสู่เซลประสาทเทียมเป็นสมาชิกของกลุ่มนั้น และไม่ใช่เมื่อข้อมูลนั้นไม่ได้เป็นสมาชิก ซึ่งเทียบกับผลลัพธ์ที่ได้จากเซลประสาทเทียมคือ 1 และ -1 (0) การที่เราต้องการให้โครงข่ายสร้างคำตอบในลักษณะนี้ต้องใช้ฟังก์ชั่นกระตุ้นแบบ Step Function เนื่องจากผลลัพธ์ที่ออกจากฟังก์ชั่นนี้มีสองสถานะเช่นกัน

จากตัวอย่างการสอนเพอร์เซปตรอนข้างต้น เราสามารถแสดงความสัมพันธ์ของกลุ่มข้อมูลที่นำมาฝึกสอนด้วยกราฟรูปที่ 2.8 โดยจุดดำแสดงเวกเตอร์ (1,1) ซึ่งให้ผลลัพธ์คือ 1 และจุดขาว แสดงเวกเตอร์ (1,-1), (-1,1) และ (-1,-1) ซึ่งให้ผลลัพธ์ -1 ซึ่งจากตัวอย่างนี้สามารถเขียนสมการหาค่า Net Input ได้ดังสมการที่ 2.6

$$\text{net} = \sum x_i w_i - w_0 \qquad \text{เมื่อ } x_0 \text{ เท่ากับ -1} \quad \text{.....(2.6)}$$

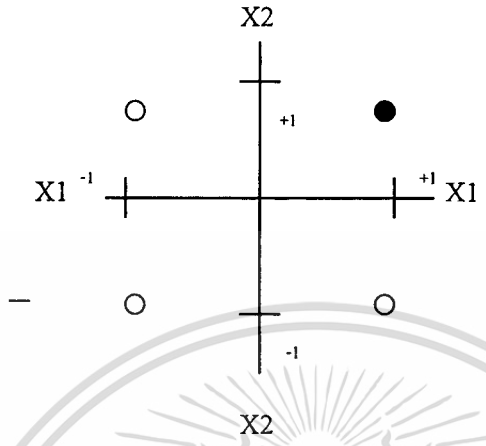
โดยที่

$$f(\text{net}) = +1 \text{ ถ้า net} \geq 0$$

$$= -1 \text{ ถ้า net} < 0$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2.8 แสดงความสัมพันธ์ของข้อมูลเชิงเรขาคณิตของตัวอย่างเพอร์เซปตรอน



และเรียกขอบเขตระหว่างพื้นที่ $f(\text{net}) > 0$ และ $f(\text{net}) < 0$ ว่าขอบเขตการตัดสินใจ (Decision Boundary) ซึ่งเราสามารถหาเส้นแบ่งได้จากสมการที่ 2.7

$$\sum_{i=1}^{n} x_i w_i - w_0 \quad \text{.....(2.7)}$$

ซึ่งสมการที่ 2.7 เป็นสมการเส้นตรง (line) เพลน (plane) หรือ ไฮเปอร์เพลน (Hyperplane)

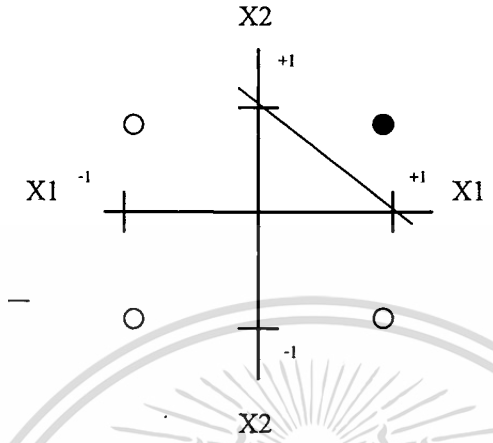
เนื่องจากข้อมูลที่เข้าสู่เพอร์เซปตรอนมีลักษณะเป็นคู่คือ (x_1, x_2) และค่าน้ำหนักที่ได้จากการสอนครั้งสุดท้ายคือ $w_0 = 1$, $w_1 = 1$ และ $w_0 = 1$ เมื่อนำมาแทนค่าในสมการของเส้นขอบเขตการตัดสินใจแสดงได้ดังสมการ 2.8 นี้

$$x_1 + x_2 = 1 \quad \text{.....(2.8)}$$

จากการแทนค่าสมการของเส้นแบ่งขอบเขตการตัดสินใจข้างบนเมื่อนำวาดเส้นตรงแสดงดังรูปที่ 2.9 จากรูปที่ 2.9 จะเห็นว่าเส้นตรงนั้นสามารถแบ่งขอบเขตระหว่างข้อมูลทั้งสองกลุ่มได้ ซึ่งเราเรียกลักษณะแบ่งแบบนี้ว่า การแบ่งแบบเชิงเส้น (Linear Separability)

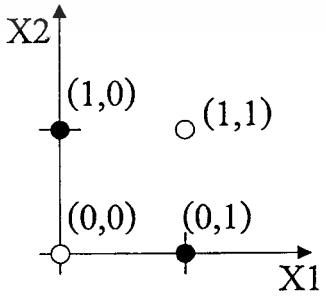
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2.9 กราฟเส้นแบ่งขอบเขตที่ได้จากสอนเพอร์เซปตรอน



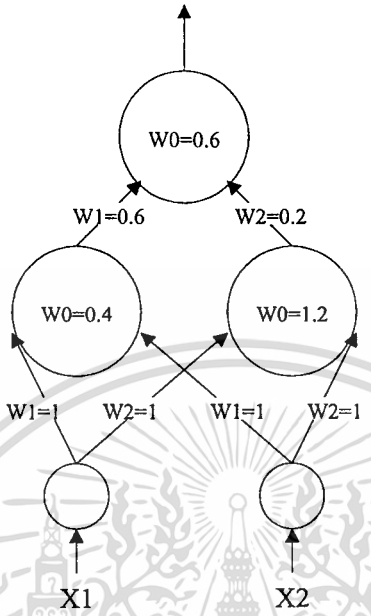
จากตัวอย่างปัญหาของเพอร์เซปตรอนข้างต้นเป็นการทำโอเปอร์เรชัน (Operation) AND ซึ่งมีลักษณะเป็นแบบเชิงเส้น แต่กรณีที่ลักษณะของปัญหาเป็นแบบไม่เป็นเชิงเส้น (Nonlinear) เช่น XOR แสดงดังกราฟรูปที่ 2.10(a) ซึ่งเพอร์เซปตรอนไม่สามารถแก้ปัญหานี้ได้ ดังนั้นเพื่อให้สามารถแก้ปัญหานี้ได้จึงมีการเพิ่มชั้นของโครงข่ายแสดงดังรูปที่ 2.10(b) รูปที่ 2.10(c) แสดงเส้นแบ่งที่ได้จากการสอนเพอร์เซปตรอนที่เพิ่มชั้นเพื่อใช้แยกข้อมูลปัญหา XOR

รูปที่ 2.10(a) แสดงกราฟของปัญหา XOR

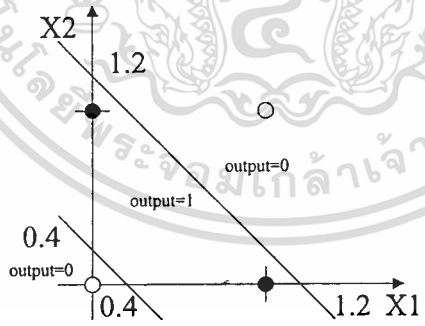


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2.10(b) แสดงการเพิ่มชั้นของเพอร์เซปตรอน



รูปที่ 2.10(c) แสดงเส้นแบ่งที่ได้จากสอนรูปที่ 2.10(a)



แสดงการแก้ปัญหาของ XOR ด้วยเพอร์เซปตรอนที่เพิ่มชั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

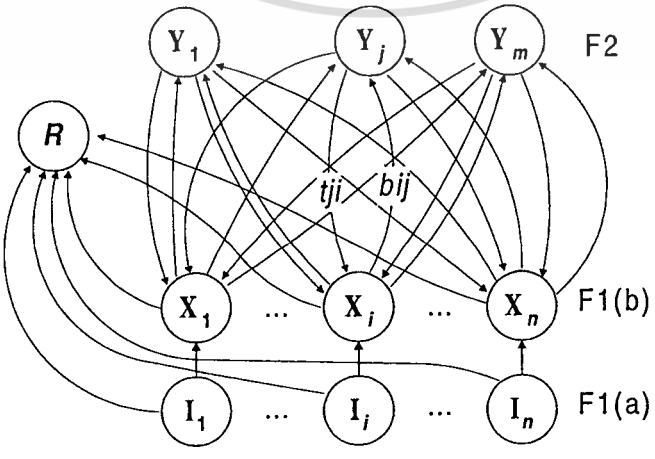
บทที่ 3

โครงข่ายประสาทเทียมแบบ ART

3.1 ทฤษฎีของ ART

โครงข่ายประสาทเทียมอะแดปทีฟรีโซแนนซ์เทียรี หรือ ART ถูกพัฒนาขึ้นโดย Carpenter และ Grossberg ART เป็นแบบจำลองโครงข่ายประสาทเทียมที่มีโครงสร้างแบบมัลติเลเยอร์ (multilayer) ประกอบด้วย อินพุทเลเยอร์ F1(a), อินเตอร์เฟสเลเยอร์ F1(b) และคลัสเตอร์เลเยอร์ F2 รูปแบบในการเรียนรู้เป็นแบบการเรียนรู้โดยไม่มีผู้สอน (Unsupervised learning) และใช้อัลกอริทึมแบบคลัสเตอร์ โครงข่าย ART มีข้อดีคือใช้เวลาในการฝึกโครงข่ายน้อย และไม่มีปัญหาในเรื่อง local minima และ memory washout [1]

รูปที่ 3.1 สถาปัตยกรรมของ ART



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 โครงสร้างของโครงข่าย ART

แบบจำลอง ART ได้พัฒนาเป็น 2 ประเภทคือ แบบจำลอง ART ซึ่งรับค่าอินพุทเป็นค่าไบนารี(ARTI) และแบบจำลองที่รับค่าอินพุทในลักษณะ continuous value (ARTII) งานวิจัยนี้ได้ใช้ ART ตัวแบบแรกซึ่งเป็นแบบจำลองที่รับค่าอินพุทไบนารี ART มีโครงสร้างที่สำคัญ 3 ส่วนแสดงดังรูปที่ 1 คือ เลเยอร์ F1, เลเยอร์ F2 และหน่วย R ซึ่งเป็น Threshold ทำหน้าที่ตัดสินว่าค่าอินพุทที่เข้ามาควรจะอยู่โหนดใด ซึ่งเป็นค่าที่ user สามารถกำหนดได้ตามความเหมาะสม

เลเยอร์ F1 ประกอบด้วย 2 เลเยอร์ย่อยคืออินพุทเลเยอร์ F1(a) และอินเตอร์เฟสเลเยอร์ F1(b) F1(a) รับข้อมูลเป็นไบนารี มีค่า 0,1 และส่งข้อมูลไปให้ F1(b) ระหว่างเลเยอร์ F1 และ F2 จะมีค่าน้ำหนักที่ใช้เชื่อมต่ออยู่ 2 ชุด คือชุดน้ำหนัก b (bottom-up weight) และ t (top-down weight) ชุดน้ำหนัก b จะเชื่อมอย่างทั่วถึงระหว่าง เลเยอร์F1(b) และ F2 ส่วนชุดน้ำหนัก t จะเชื่อมอย่างทั่วถึงระหว่างชั้น F2 และ F1(b) การตัดสินว่าโหนดใดควรเป็นโหนดที่จะเรียนรู้ ทำได้โดยการหาโหนดชนะดังสมการที่ 3.1

$$Y_j = \sum_{i=1}^{n} b_{ij} I_i \qquad \text{..... (3.1)}$$

$$\mathbf{Y_J} = \text{Max } Y_j \qquad \text{..... (3.2)}$$

เมื่อได้โหนดชนะ J ที่มีค่า Y_j มากที่สุดแล้ว การหาโหนดที่จะใช้ฝึก(โหนดเรียน) ต้องพิจารณาความคล้ายของอินพุท I ที่เข้ามาฝึกกับค่าน้ำหนักของโหนดชนะที่ได้ฝึกไว้แล้ว การวัดความคล้ายทำโดยใช้น้ำหนัก t_{ji} ค่าน้ำหนัก t_{ji} จะใช้เก็บค่าจุดที่ซ้ำกันระหว่างอินพุทใหม่ที่เข้ามาฝึก กับอินพุทเดิมที่ได้ฝึกไว้แล้ว ดังนั้น ความคล้ายของโหนดกับอินพุท สามารถวัดได้โดยการหาส่วนจุดซ้ำระหว่าง t_{ji} กับอินพุท I_i และนับจำนวนว่ามีจุดซ้ำกี่จุด แล้วจึงนำมา normalize กับขนาดของอินพุทที่มีค่าเป็น 1 เรียกว่า ratio

$$\text{ratio} = \sum_{i=1}^{n} t_{ji} I_i / \sum_{i=1}^{n} I_i \quad \text{ที่ } I_i \text{ เป็น 1} \qquad \text{..... (3.3)}$$

โหนดที่จะเรียนรู้ต้องมีความคล้ายมากกว่าหรือเท่ากับ ρ ในกรณีที่โหนดชนะมีค่าความคล้ายน้อยกว่าค่า ρ จะทำการหาค่าโหนดชนะใหม่โดยนำค่า Y_j ที่มีค่าสูงสุดรองลงไปมาเป็นโหนดชนะ และ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทดสอบความคล้ายจนกว่าจะพบโหนดที่มีความคล้ายเกินค่า ρ และทำการปรับน้ำหนักดังสมการที่ 3.4-3.7

$$X_i = t_{ji} * I_i \quad \text{..... (3.4)}$$

$$||X|| = \sum_{i=1}^{n} t_{ji} * I_i \quad \text{..... (3.5)}$$

$$t_{ji} = X_i \quad \text{..... (3.6)}$$

$$b_{ij} = L*X_i / (L-1+||X||) \quad \text{..... (3.7)}$$

กรณีที่ไม่มีโหนดดังกล่าวก็จะทำการสร้างโหนดใหม่เก็บ Pattern I และให้ค่าน้ำหนักเริ่มต้นของ t_{ji} เป็นค่าอินพุท I_i และ initial weight b_{ij} ดังสมการที่ 3.8 และ 3.9

$$t_{ji}\,(NEW) = I_i \quad \text{..... (3.8)}$$

$$b_{ij}\,(NEW) = L*I_i / (L-1+\sum_{i=1}^{n} I_i) \quad \text{..... (3.9)}$$

3.3 การปรับค่าน้ำหนักของ ART

ART มีชุดน้ำหนัก 2 ชุดคือ b และ t การปรับชุดน้ำหนักของ t และ b จะปรับน้ำหนักเฉพาะจุดที่ซ้ำกัน (Intersection) ระหว่าง t_{ji} กับค่าของอินพุท X_n โดยจุดที่จุดซ้ำของชุดน้ำหนัก t จะให้ค่าเป็น 1 และจุดที่ไม่ซ้ำจะให้ค่าเป็น 0 ดังรูปที่ 3.2 ส่วนชุดน้ำหนัก b จุดซ้ำจะให้ค่าน้ำหนักดังสมการที่ (3.7) และค่า 0 สำหรับจุดที่ไม่ซ้ำ จากตัวอย่างดังรูปที่ 3.2 ค่าน้ำหนักใหม่ของ b_{ij} กำหนดได้ดังนี้

L คือค่าพารามิเตอร์ = 2

X_i คือจุดที่ intersect กันระหว่างจุดของอินพุท I กับจุดของโหนดเรียน J = 0 หรือ 1

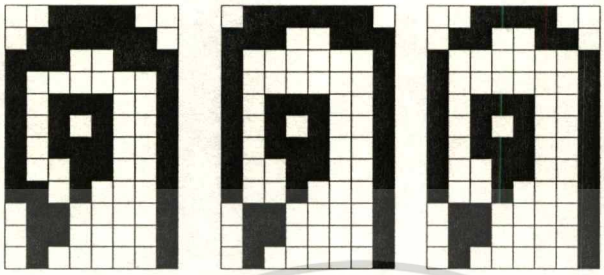
||X|| คือจำนวนจุดที่ intersect กันระหว่างจุดของอินพุท I กับจุดของโหนดเรียน J = 41

$b_{ij} = L*X_i / (L-1+||X||)$

$$b_{ij} = \begin{cases} 0 \\ 0.047 \end{cases}$$ ขึ้นกับว่าเป็นจุดซ้ำ หรือไม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.2(a) แสดงการเรียนตัวอักษร ด ของชุดน้ำหนัก t



Art Node โหนดเรียน $X= t_{ji}*X_i$

** สีขาวคือค่า 0

** สีดำคือค่า 1

จะเห็นว่าการปรับค่าน้ำหนักใหม่ของ b จะไม่สนใจค่าน้ำหนักเดิม แต่ค่าจะเปลี่ยนแปลงขึ้นกับจำนวนจุดซ้ำ ระหว่างอินพุทกับโหนดเรียน ค่าน้ำหนักของ b จะน้อยเมื่อจำนวนจุดซ้ำมาก แต่จะมีค่ามากถ้าจำนวนจุดซ้ำน้อย ดังรูปที่ 3.2(b) ผลจากการการปรับค่าน้ำหนักในลักษณะนี้จะทำให้ pattern ที่เรียนรู้ในรอบแรกซึ่งตกที่โหนดหนึ่ง อาจจะเปลี่ยนไปตกที่โหนดอื่นได้ในรอบถัดไป

รูปที่ 3.2(b) แสดงชุดน้ำหนัก b ของART Node ซึ่งเก็บ pattern ของตัวอักษร ด

0	0	0.0400	0.0400	0.0400	0.0400	0	0
0	0.0400	0.0400	0.0400	0.0400	0.0400	0.0400	0
0.0400	0.0400	0.0400	0	0	0.0400	0.0400	0.0400
0.0400	0	0	0	0	0	0	0.0400
0.0400	0	0.0400	0.0400	0.0400	0	0	0.0400
0.0400	0	0.0400	0	0.0400	0	0	0.0400
0.0400	0	0.0400	0.0400	0.0400	0	0	0.0400
0.0400	0	0	0.0400	0.0400	0	0	0.0400
0.0400	0.0400	0	0.0400	0.0400	0	0	0.0400
0	0.0400	0.0400	0.0400	0	0	0	0.0400
0	0.0400	0.0400	0	0	0	0	0.0400
0	0.0400	0	0	0	0	0	0.0400

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.2(c) แสดงชุดน้ำหนักใหม่เมื่อผ่านการเรียนรู้

0	0	0.0476	0.0476	0.0476	0.0476	0	0
0	0.0476	0.0476	0	0	0.0476	0.0476	0
0.0476	0	0	0	0	0	0	0.0476
0.0476	0	0	0	0	0	0	0.0476
0.0476	0	0.0476	0.0476	0.0476	0	0	0.0476
0.0476	0	0.0476	0	0.0476	0	0	0.0476
0.0476	0	0.0476	0.0476	0.0476	0	0	0.0476
0.0476	0	0	0.0476	0.0476	0	0	0.0476
0.0476	0	0	0.0476	0	0	0	0.0476
0	0.0476	0.0476	0	0	0	0	0.0476
0	0.0476	0.0476	0	0	0	0	0.0476
0	0.0476	0	0	0	0	0	0.0476

3.4 อัลกอริทึมที่ 1 ที่ใช้ฝึกสอน ART

The notation

n number of components in the input vector.

m number of cluster that can be formed.

b_{ij} bottom-up weights (From F1(b) unit Xi to F2 unit Yj).

t_{ji} top-down weights (From F1(b) unit Yj to F1 unit Xi) .

ρ Threshold value

I binary input vector (an n-tuple).

X activation vector for F1(b) layer.

||X|| norm of vector X, defined as the sum of the components Xi.

Step 0 Initialize parameter L and Threshold (ρ)

$L > 1$, $\rho = 0.75$

Initialize weight b and t

$0<b_{ij}(0)<L/(L-1+n)$, $t_{ji}(0)=1$

Step 1 While stop condition is false

Step 2 For each training input

Step 3 Sending input signal I from F1(a) to

F1(b) and compute the norm of I ||I||

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$X_i = I_i \quad , \|I\| = \sum_{i=1}^{n} I_i$$

Step 4 Find winning Node Y_J

$$Y_j = \sum_{i=1}^{n} b_{ij} X_i$$

$Y_J \geq Y_j$ For all nodes j

Step 5 Recompute activation X of F1(b)

$$X_i = t_{ji} * I_i \quad , \|X\| = \sum_{i=1}^{n} t_{ji} * I_i$$

Step 6 Test the resemble of Input and Winning Node

ratio = $\|X\| / \|I\|$

IF ratio > ρ

THEN

goto step8

ENDIF

Step 7 Find another winning node Y_J with

max Y_j and ratio > ρ

IF possible

THEN

goto Step 8

ELSE

create new node to learn this pattern with

$t_{ji} = I_i$

b_{ij} (NEW) = $L*I_i / (L-1+\|I\|)$

ENDIF

Step 8 Update the weight for Node J

b_{ij} (NEW) = $L*X_i / (L-1+\|X\|)$

t_{ji} (NEW) = X_i

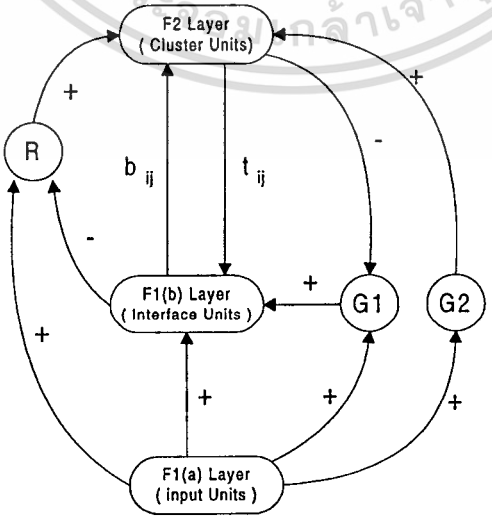
Step 9 Test for stopping condition

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5 Supplemental Units

อัลกอริทึมที่ใช้สอน ART ที่ได้กล่าวมา อธิบายถึงการประมวลผลของโครงข่ายประสาทเทียม (neural network process) ในเทอมของเลเยอร์ F1, F2 และ หน่วย R ซึ่งในสภาวะการทำงานจริง บางเลเยอร์จะมีสถานะในการทำงานหลายสถานะ และเซลประสาทจะมีการตอบรับแตกต่างกันในสถานะที่แตกต่างกัน ตัวอย่างที่แสดงให้เห็นการตอบรับที่แตกต่างกัน เช่น เมื่อเลเยอร์ F1(b) ได้รับสัญญาณอินพุทจากเลเยอร์ F1(a) ซึ่งในขณะนั้นเลเยอร์ F2 ยังไม่แอคทีฟ ผลจากการกระตุ้นก็คือเซลประสาทในเลเยอร์ F1(b) ซึ่งรับค่าอินพุทมาจากเลเยอร์ F1(a) จะ on ในหน่วยที่ได้รับสัญญาณจากเลเยอร์ F1(a) แต่ขณะที่เลเยอร์ F2 แอคทีฟ เมื่อเลเยอร์ F1(b) ได้รับสัญญาณกระตุ้น เซลประสาทในเลเยอร์ F2 จะ on เฉพาะหน่วยที่ได้รับสัญญาณจากทั้งเลเยอร์ F2 และ Input Unit เท่านั้น อย่างไรก็ตามในการทำงานจริง เซลประสาททางชีววิทยา (bioneural) โดยตัวมันเองไม่สามารถตัดสินใจได้ว่าสถานะที่เกิดเป็นสถานะแบบใดและควรทำอย่างไร หน่วยที่ใช้ควบคุมการตัดสินใจส่วนนี้คือ Supplemental Units(Gain Control Units) ซึ่งประกอบด้วยหน่วย G1 และ G2 ดังรูปที่ 3.3

รูปที่ 3.3 แสดงการทำงานของ Supplemental Unit (G1, G2)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กำหนดให้สัญญาณกระตุ้นคือ + สัญญาณยับยั้งคือ - จากภาพจะเห็นว่าเซลประสาทในเลเยอร์ F1(b) สามารถรับสัญญาณได้ 3 ทางคือ จาก F1(a) , G1 และ จากเลเยอร์ F2 โดยผ่านชุดน้ำหนัก t_{ji} ส่วนเซลประสาทในเลเยอร์ F2 สามารถรับสัญญาณได้ 3 ทางคือ จาก F1(b) โดยผ่านชุดน้ำหนัก t_{ji} , G2 และ หน่วย R การ on ในเลเยอร์ใดๆ จะเกิดขึ้นก็ต่อเมื่อได้รับสัญญาณ 2 ใน 3 ทาง ตามกฎ 2 ใน 3 (two-third rules)

เมื่อมีอินพุทเข้ามา ในขณะที่ F2 ยังไม่แอคทีฟ สัญญาณจะถูกส่งไปให้เซลประสาทในเลเยอร์ F1(b) หน่วย G1 และหน่วย G2 ซึ่งในกรณีนี้พบว่าเซลประสาทในเลเยอร์ F1(b) ได้รับสัญญาณ 2 ทิศทางคือจาก F1(a) และ G1 จึงอยู่ในสภาวะ on แต่ขณะที่ F2 แอคทีฟ F2 จะส่งสัญญาณกระตุ้นให้ F1(b) และส่งสัญญาณ ยับยั้งให้กับ G1 F1(b) จึงได้รับสัญญาณจาก F2 โดยผ่านชุดน้ำหนัก t_{ji} และได้รับสัญญาณจากเลเยอร์ F1(a) F1(b) จึง on เฉพาะหน่วยที่ได้รับสัญญาณทั้งจาก F2 และ F1(a)

3.6 การทดลองใช้ ART เรียนรู้ตัวอักษรภาษาไทย

งานวิจัยนี้ ได้นำ ART มาประยุกต์ใช้ในงานแยกแยะตัวอักษรพิมพ์ภาษาไทย โดยข้อมูลที่นำเข้าสู่โครงข่ายเป็นข้อมูลภาพของตัวอักษรที่ประกอบด้วยจุดภาพ อินพุทเมตริก (input metrix) โดยอินพุทจะเป็นตัวอักษรพิมพ์ภาษาไทย 75 แบบประกอบด้วยตัวพยัญชนะไทย 44 ตัว และเลขไทย เลขอารบิค สระ า เ ใ ๆ า ฤ ฦ ไม้หันอากาศ โดยใช้แบบตัวอักษรอังศนายูพีซี ขนาด 12 จุด พิมพ์จากเครื่องพิมพ์เลเซอร์ จากนั้นจะทำการแกนตัวอักษรโดยใช้ความละเอียดในการสแกนเท่ากับ 300 จุดต่อนิ้ว (dpi) และ scale100 และทำการสแกนตัวอักษร โดยทำการแปลงภาพตัวอักษร (*.tiff) ให้เป็นไบนารี โดยกำหนดให้ค่า 0 แทนพื้นตัวอักษรและค่า 1 แทนเนื้อตัวอักษร จากนั้นตัดตัวอักษรออกมาทีละตัวจากไฟล์ไบนารี และวางตัวอักษรลงในเมตริก 2 มิติ ซึ่งในงานวิจัยนี้ให้เมตริกของตัวอักษรมีขนาด 27*37 โดยวางตัวอักษรที่เป็นข้อมูลเข้าที่ตำแหน่งชิดบนด้านซ้ายของเมตริก

ในการฝึกสอนโครงข่ายจะใช้ชุดฝึกประกอบด้วยตัวอักษรไทยตัวละ 20 แบบรวมทั้งหมด 1500 ส่วนตัวอักษรชุดทดสอบโครงข่ายจะใช้ตัวอักษรไทยตัวละ 200 แบบรวมทั้งหมด 15000 ตัว

กำหนดค่า parameter ของโครงข่ายประสาทเทียมชุดนี้ดังนี้

n = 999

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$m = 550$

$L = 2$

$\rho = 0.75$

$b_{ij} = 2/(2-1+55)$

$t_{ji} = 1$

ตารางที่ 3.1 แสดงการเรียนผิดโหนดซึ่งเป็นผลของการฝึกตัวอักษรไทยโดยใช้ ART

โหนดที่	ตัวอักษรที่เรียนผิดโหนด	โหนดที่	ตัวอักษรที่เรียนผิดโหนด	โหนดที่	ตัวอักษรที่เรียนผิดโหนด	โหนดที่	ตัวอักษรที่เรียนผิดโหนด
3	ก ถ	148	ฮ 6	260	ด ๑	402	อ 9
14	ค ศ ต	158	ฌ 9	268	ฐ 8	416	0 6
57	ฒ 9	160	ฐ 9	269	ฐ 8	418	ฮ 6
71	ก 0	164	ฌ 9	271	ท ห	420	ร ะ
72	ฉ 9	166	ง า	288	ฮ 7	431	ฮ 8
74	ท ห	167	ฐ 8	289	ฮ 9	433	ธ 7
80	ท ห	176	ฒ 9	291	ฮ 8	448	ฐ 8
83	ท พ	177	ค ศ	321	โ ๅ	451	ธ 7
85	บ ษ	178	ค ศ	331	ฮ 8	454	ใ ไ
107	ย 6	182	ธ 7	332	ฐ 8	459	เ เ
109	ย 6	185	ฉ 9	360	ด ต	459	เ เ
120	ค ศ	212	ค ศ	369	ฐ 8	466	โ ๅ
129	บ ษ	214	ค ศ	374	ฮ 8	469	โ ๅ
130	ภ ฦ	221	ด ๑	375	ด ต	472	ฮ 9
146	ธ 7	239	น ห	386	ฮ 8	485	ว า
						490	ฌ 9

แสดงการเรียนผิดโหนด ผลการฝึกรอบที่ 80 จำนวนโหนดทั้งหมด 499 โหนด $\rho = 0.75$ พบว่ายังไม่สามารถแยกตัวอักษรได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.7 ผลการทดลอง

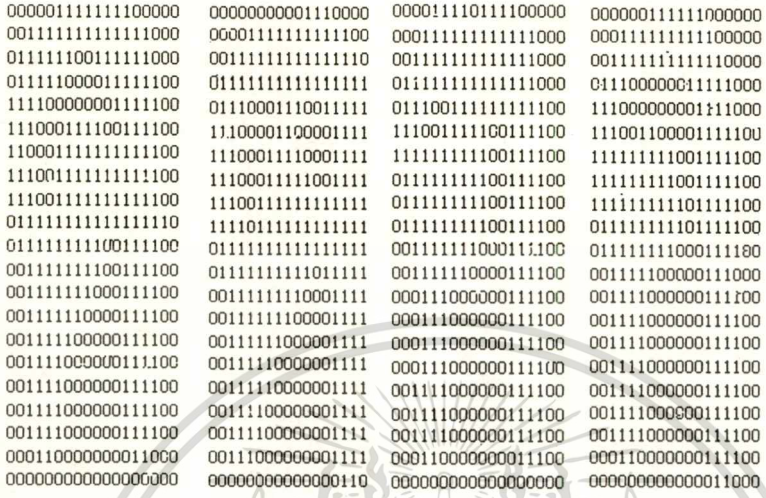
ผลจากการทดลองตามอัลกอริทึมที่ 1 ที่ใช้สอน ART โครงข่ายจะเริ่มเสถียรในรอบที่ 80 คือไม่มีการสร้างโหนดใหม่เพิ่มขึ้นอีก การฝึกจึงสิ้นสุดลง ผลการทดลองพบว่าโครงข่ายยังไม่สามารถแยกตัวอักษรได้ บางโหนดจะพบว่ามีตัวอักษรมากกว่า 1 ตัวตกอยู่โหนด เดียวกัน ตัวอย่างเช่น ก ถ จะตกโหนดเดียวกันเสมอ โครงข่ายนี้จึงไม่สามารถนำมาใช้ได้ เพราะในการทดสอบการแยกแยะตัวอักษร หากมีอินพุทตกในโหนดลักษณะนี้ จะไม่สามารถแยกแยะได้ว่าอินพุทที่เข้ามานั้นเป็นตัวอักษรแบบใด ผลการทดสอบโดยใช้ค่า $\rho = 0.75$ จะสิ้นสุดการฝึกในรอบที่ 80 โดยได้จำนวนโหนดทั้งสิ้น 499 โหนด มีจำนวนโหนดที่จำผิดพลาด 61 โหนด แสดงโหนดุที่ผิดพลาดในตารางที่ 3.1

3.8 ปัญหาและการปรับปรุงเพื่อนำ ART มาใช้แยกตัวอักษรภาษาไทย

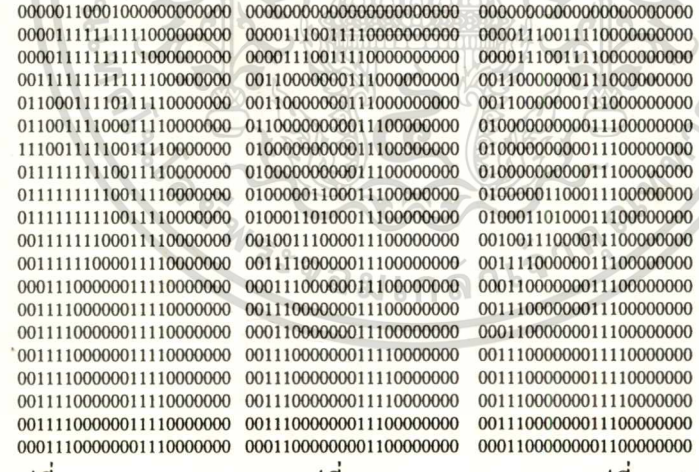
ปัญหาขณะฝึกโครงข่ายไม่สามารถแยกตัวอักษรไทยได้ การเรียนของ ART เป็นแบบ Unsupervised โดยจะเลือกทำการปรับน้ำหนักของโหนดชนะเพียงโหนดเดียว และทำการปรับชุดน้ำหนักทั้ง b และ t เนื่องจากลักษณะของอักษรไทยบางตัวจะมีความคล้ายกันมาก เช่น ดตคฅ เมื่อผ่านการสแกนจะทำให้ภาพของตัวอักษรมีความคล้ายกันมากยิ่งขึ้นดังตัวอย่างรูปที่ 3.4 จะเห็นได้ว่าความแตกต่างระหว่างตัวอักษรแต่ละตัวในชุดนี้น้อยมาก กรณีเช่นนี้จึงทำให้มีการเรียนผิดโหนดได้ ดังรูปที่ 3.5 การแก้ไขปัญหานี้โดยให้ค่า ρ มีค่าสูงๆ จะแก้ปัญหาการเรียนผิดโหนดได้ แต่ว่าจะเกิดโหนดจำนวนมาก ตัวอย่างเช่นอักษร ด แต่ละแบบก็จะกลายเป็นแต่ละโหนดไป จึงไม่สามารถแก้ไขโดยวิธีนี้ได้ แต่สามารถการแก้ไขได้โดยการเปลี่ยนวิธีการเรียนรู้จาก Unsupervise ART ให้เป็น Supervise ART

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.4 แสดงตัวอักษรชุดที่มีความคล้ายคลึงกันมากคือ ด ต ค ศ



รูปที่3.5 แสดงการเรียนรู้ตัวอักษร ศ โดยใช้ ART



รูปที่ 3.5(a) รูปที่ 3.5(b) รูปที่ 3.5(c)

รูปที่ 3.5(a) โหนดของตัวอักษร ศ รูปที่ 3.5(b) โหนดของตัวอักษร ค หลังการเรียน 1 รอบด้วยตัวอักษร 1500 ตัว ค่า ρ เท่ากับ 0.75 โดยตัวอักษรที่ฝึกโหนดในรอบนี้คือ ศ15 ศ16 ศ17 ศ 18 ศ20 ด1 ด2 ด4 ด6 ด13 ด19 รูปที่ 3.5(c) โหนดของตัวอักษร ศ หลังการเรียนรอบที่ 2 ค่า ρ เท่ากับ 0.75 โดยตัวอักษรที่ทำการฝึกโหนดในรอบนี้คือ ศ2 ล7 ล10 ล12 ล13 ล15 ล19 ด1 ด3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

ARTS และปัญหาของ ARTS ที่ใช้ในการจดจำอักษรพิมพ์ภาษาไทย

Supervised ART (ARTS) เป็นการฝึกให้แบบจำลองมีการปรับน้ำหนักในลักษณะเดียวกับ ART แต่มีข้อแตกต่างคือในช่วงการฝึก จะเพิ่มการทดสอบว่าอินพุทกับโหนดชนะที่มีค่าความคล้ายเกิน ρ นั้นเป็นตัวอักษรเดียวกันหรือไม่ ARTS จะทำการปรับน้ำหนักเฉพาะโหนดชนะที่มีค่าความคล้ายเกินค่า ρ และ pattern ของอินพุทจะต้องเป็นตัวเดียวกับ pattern ในโหนดเรียน ดังนั้นการเรียนของโหนดชนะจะเป็นการปรับน้ำหนักเฉพาะตัวอักษรแบบเดียวกันเท่านั้น โหนดจะไม่ถูกปรับค่าอย่างผิดๆ เหมือนลักษณะการเรียนรู้แบบ unsupervised ART

4.1 อัลกอริทึมที่ใช้สอน ARTS

Step 0 Initialize parameter L and Threshold (ρ)

$L > 1$, $\rho = 0.75$

Initialize weight b and t

$0 < b_{ij}(0) < L/(L-1+n)$, $t_{ji}(0)=1$

Step 1 While stop condition is false

Step 2 For each training input

Step 3 Sending input signal I from F1(a) to F1(b) and compute the norm of I $||I||$

$$X_i = I_i \quad , ||I|| = \sum_{i=1}^{n} I_i$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Step 4 Find winning Node Y_J

$$Y_j = \sum_{i=1}^{n} b_{ij} X_i$$

$Y_J \geq Y_j$ For all nodes j

Step 5 Recompute activation X of F1(b)

$$X_i = t_{ji} * I_i \quad , \|X\| = \sum_{i=1}^{n} t_{ji} * I_i$$

Step 6 Test the resemble of Input and Winning Node

ratio = $\|X\| / \|I\|$

IF ratio > ρ

THEN

IF Input-Pattern = Node-Pattern

goto step8

ELSE

goto step7

ENDIF

ELSE

go to step7

ENDIF

Step 7 Find another winning node Y_J with max Y_j and ratio > ρ

IF possible

THEN

goto Step 8

ELSE

create new node to learn this pattern with

$t_{ji} = I_i$

b_{ij} (NEW) = $L*I_i / (L-1+\|I\|)$

ENDIF

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Step 8 Update the weight for Node J

$$b_{ij}(NEW) = L*X_i / (L-1+||X||)$$

$$t_{ji}(NEW) = X_i$$

Step 9 Test for stopping condition

4.2 การทดลองแยกตัวอักษรภาษาไทยโดยใช้ ARTS

การทดลองแยกตัวอักษรภาษาไทยโดยใช้ ARTS ซึ่งอัลกอริทึมปรับเปลี่ยนเป็น ARTS สามารถแก้ปัญหาการตกผิดของอินพุทโหนด ขณะฝึก การเรียนของโหนดชนะจะเป็นการปรับน้ำหนักเฉพาะตัวอักษรแบบเดียวกันเท่านั้น โหนดจะไม่ถูกปรับค่าอย่างผิดๆ เหมือนลักษณะการเรียนรู้แบบ Unsupervised ART เมื่อนำโครงข่ายมาทดสอบประสิทธิภาพของการรู้จำโดยกำหนดขนาด ρ ขณะทดสอบให้มีค่าเดียวกับขณะฝึก และได้สรุปผลการทดลองดังตารางที่ 4.1 ผลการทดลองแสดงให้เห็นจากรูป 4.1 แสดงให้เห็นถึงการเรียนตัวอักษร ๆ จะเห็นว่าผลจากการเรียนรู้จะได้โหนดที่ไม่ผิดเพี้ยนเหมือนลักษณะโหนดที่เรียนโดยใช้ ART

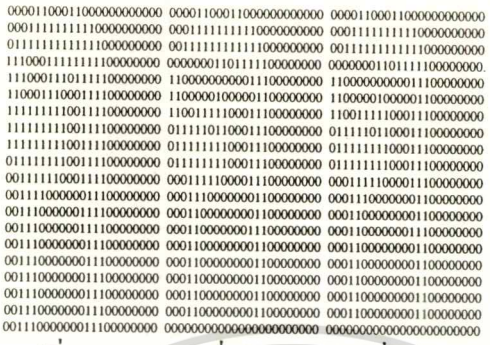
ตารางที่ 4.1 แสดงประสิทธิภาพการจำทดสอบโดยใช้ ARTS

ρในช่วงการฝึก	จำนวนรอบ	จำนวนโหนดทั้งหมด	จำนวนโหนดที่ใช้	ประสิทธิภาพการจำ
0.55	4	158	158	97.0666
0.60	5	193	193	97.9466
0.65	5	258	258	97.6666
0.70	6	352	351	97.5066
0.75	6	468	465	96.5400
0.80	6	598	599	95.4733
0.85	6	797	796	94.4800
0.90	4	1000	1000	92.4272

ประสิทธิภาพการจำทดสอบ โดยใช้ตัวอักษรไทยจำนวน 15000 ตัว โดยใช้ค่า ρ เริ่มต้นขณะทำการฝึกแตกต่างกัน

ลิขสิทธิ์มหาวิทยาลัยเชียงใหม่
Copyright© by Chiang Mai University
All rights reserved
อนุญาตให้ใช้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.1 ผลจากการฝึกโหนด ๓ โดยใช้ ARTS



รูปที่ 4.1(a) รูปที่ 4.1(b) รูปที่ 4.1(c)

รูปที่ 4.1(a) โหนดของตัวอักษร ๓ รูปที่ 4.1(b) โหนดเดิม หลังการเรียน 1 รอบด้วยตัวอักษร 1500 ตัว ค่า ρ เท่ากับ 0.75 โดยตัวอักษรที่เข้าฝึกโหนดในรอบนี้คือ ๓1 ๓2 ๓3 ๓4 รูปที่ 4.1(c) โหนดเดิม หลังการเรียนรอบที่ 2 ค่า ρ เท่ากับ 0.75 โดยตัวอักษรที่ทำการฝึกโหนดในรอบนี้คือ ๓4

จากตารางที่ 4.2 แสดงตัวอักษรที่แยกแยะผิดซึ่งเป็นผลผลการแยกแยะตัวอักษรที่แสดงเฉพาะตัวอักษรที่ตกผิดโหนดที่ค่า ρ เท่ากับ 0.60 ซึ่งให้ประสิทธิภาพในการแยกแยะตัวอักษรที่ดีที่สุดคือ 97.94 % จะเห็นได้ว่าตัวอักษรบางตัวเช่น เลข 1 ไปตกที่โหนดของตัวอักษร ๖ เมื่อพิจารณาดูแล้วจะเห็นว่ารูปร่างของตัวอักษรไม่มีความคล้ายคลึงกันเลย แต่เมื่อพิจารณาดูจากตารางที่ 4.3 จะเห็นได้ว่าค่า Y ซึ่งเป็นค่าน้ำหนัก b ของ ART โหนดคูณกับค่าของอินพุทที่เข้ามานั้นโหนดชนะที่ให้ค่า Y สูงสุดคือโหนดที่ 131 ซึ่งเป็นโหนดของตัวอักษรเลข 1 นั่นเอง แต่เมื่อพิจารณาค่า ratio แล้วพบว่าค่าที่ได้น้อยกว่าค่า ρ โหนดที่ 131 จึงไม่สามารถเรียนตัวอักษรตัวนี้ได้ จนกระทั่งค่า Y ลำดับที่ 20 ซึ่งเป็นโหนดของตัวอักษรเลข ๖ ค่า ratio จึงมีค่ามากกว่าค่า ρ จึงทำการเรียนรู้โหนดนี้ เหตุที่ค่า ratio ของโหนดนี้มีค่ามากเป็นเพราะ ART โหนดซึ่งเป็นโหนดของเลข ๖ มีขนาดใหญ่ขณะที่อินพุทโหนดมีขนาดเล็กว่ามาก เนื่องจากว่าค่า ratio เกิดจากการนำค่าของจำนวนจุดที่ซ้ำกันระหว่างอินพุทกับ ART โหนดที่เป็นโหนดชนะส่วนด้วยจำนวนทั้งหมดของอินพุท ดังนั้นถ้า ART โหนดมีขนาดใหญ่ โอกาสที่โหนดชนะจะเป็นโหนดเรียนจึงมีค่าสูง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.2 แสดงตัวอักษรที่แยกแยะผิดโดยใช้ ARTS

ตัวอักษร	จำนวนการแยกแยะผิด	ตัวอักษร	จำนวนการแยกแยะผิด	ตัวอักษร	จำนวนการแยกแยะผิด	ตัวอักษร	จำนวนการแยกแยะผิด
ก	-	ข	ช(2)	ฃ	-	ค	ก (3)
ฅ	ค(24) ต(1)	ฆ	-	ง	-	จ	ฉ(3)
ฉ	-	ช	ข(5)	ซ	-	ฌ	ญ(1)
ญ	-	ฎ	ฏ(1)	ฏ	ฎ(5)	ฐ	พ(10)
ฑ	-	ฒ	-	ณ	ฌ(3)	ด	ต(11)
ต	-	ถ	-	ท	ห(8)	ธ	-
น	ห(18) ษ(1)	บ	-	ป	-	ผ	-
ฝ	-	พ	ห(4)	ฟ	ฎ(9)	ภ	-
ม	-	ย	-	ร	-	ฤ	ถ (9)0(4)
ล	ถ(7)	ฦ	ภ(70)	ว	-	ศ	ด(2) ต(8) ส(1)
ษ	บ(3)	ส	-	ห	ท(2)	ฬ	พ(1)
อ	ก(1) ถ(4)ค(1)	ฮ	-	ฯ	-	ะ	-
า	-	0	ก(1) อ(1)	1	ม(2) ๖(1) ฟ(1)	2	-
3	-	4	-	5	-	6	๘ (1)
7	-	8	ฮ(5)	9	-	เ	-
โ	ๅ(1)	ไ	โ(2)	ๅ	ไ(39)	ๆ	-
ๆ	-	๐	-	๑	-	๒	-
๓	-	๔	๕(3)	๕	-	๖	-
๗	-	๘	๕(1) ห(1)	๘	ก(1) อ(1)		

แสดงจำนวนอินพุทที่ตกผิดโหนด ทดสอบโดยใช้ตัวอักษรไทยจำนวน 15,000 ตัว ทดสอบกับตัวอักษรที่สแกนจากกระดาษที่พิมพ์ด้วยเครื่องพิมพ์เลเซอร์ ความสว่าง 100% โดยใช้ค่า $\rho = 0.60$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.3 แสดงการเรียงลำดับของโหนดชนะของตัวอักษรเลข 1

โหนดที่	ตัวอักษร	Y	ratio
131	1	1.7704	0.58
129	1	1.7692	0.49
180	1	1.4864	0.59
130	1	1.2692	0.35
95	เ	1.1463	0.50
90	ะ	1.0000	0.26
91	ั	0.9393	0.33
116	๖	0.9176	0.34
94	เ	0.9122	0.27
118	๖	0.8852	0.58
47	ธ	0.8643	0.54
117	๖	0.8631	0.44
49	บ	0.8440	0.49
61	ม	0.8208	0.59
133	2	0.8169	0.31
59	ภ	0.8113	0.46
135	2	0.8048	0.35
134	2	0.7792	0.32
65	ร	0.7608	0.37
176	๖	0.7604	0.78

แสดงการเรียงลำดับของโหนดชนะ และค่า ration ซึ่งอินพุทเป็นตัวอักษรเลข 1 อารบิค โดยใช้ค่า $\rho = 0.60$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.2 แสดงอินพุทตัวอักษร 1 และโหนดชนะของตัวอักษรเลข ๖ และจุดที่ intersect ระหว่างอินพุทกับโหนดชนะ

000001000	11000000000000000000000000	## +
000111100	11110000000000000000000000	###1+++
011111100	11111000000000000000000000	#1111++
011111100	01111000000000000000000000	1111++
000111100	00111000000000000000000000	#11++
000111100	00111100000000000000000000	#111+
000111100	00011100000000000000000000	111+
000111100	00011111000000000000000000	1111#
000111100	00011111111111000000000000	1111##
000111000	00011111111111100000000000	111###
000111000	00011111111111100000000000	111###
000111000	00011111111111110000000000	111###
000111000	00011110000111111000000000	111#
000111000	00011110000011111000000000	111#
000111100	00011110000001111100000000	1111
000111100	00011111000001111100000000	1111#
000111100	00011111000001111100000000	1111#
000111100	00011111000001111100000000	1111#
000111100	00011111000001111100000000	1111#
001111100	00011111000011111000000000	+1111#
011111110	00001111111111111000000000	+++1111#
111111111	00001111111111111000000000	++++11111

อินพุทคือ 1 ART NODE คือโหนดของตัวอักษร ๖ ค่า Y = 0.7604, Threshold = 0.78 และ ratio = 0.75 จะเห็นได้ว่าค่า ratio น้อยกว่าค่า Threshold โหนดนี้จึงไม่สามารถใช้เรียนรู้ได้

** 1 หมายถึงจุดที่ซ้ำกันระหว่างอินพุทกับโหนดชนะ

** # หมายถึงจุดที่มีใน ART โหนดแต่ไม่มีในอินพุทโหนด

** + หมายถึงจุดที่มีในอินพุทโหนดแต่ไม่มีใน ART โหนด

** 0 หมายถึงจุดที่ไม่มีทั้งในอินพุทโหนดและ ART โหนด

4.3 ปัญหาการใช้ ARTS แยกตัวอักษรภาษาไทย

ปัญหาการแยกแยะโหนดผิดขณะทดสอบการจำ ปัญหาที่ผ่านมาในการไม่สามารถแยกแยะโหนดได้นั้น เป็นปัญหาที่เกิดขึ้นขณะเรียน แต่ปัญหาที่เกิดขึ้นในข้อนี้เป็นปัญหาที่เกิดขึ้นขณะทดสอบการจำ เนื่องจากตัวอักษรไทยบางกลุ่มมีความคล้ายคลึงกันมากเช่น คดตฅ ดังนั้นในขณะทดสอบการจำตัวอักษรหากกำหนดค่า ρ ให้มีค่าน้อย ตัวอักษรอื่นที่คล้ายคลึงกันก็อาจจะตก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผิดกลุ่มได้ดังผลการทดลองที่ผ่านมา จากตารางที่ 3 จะเห็นได้ว่าค่า ratio ของตัวอักษรเลข 1 ของโหนดลำดับที่ 131 ที่มีค่าน้ำหนักสูงที่สุด มีค่าน้อยกว่าค่า ρ จึงถือว่าการทดสอบความคล้ายระหว่างโหนดชนะกับจำนวนจุดของอินพุทพบว่าค่าความคล้ายไม่ผ่าน ดังนั้นหากกำหนดค่า ρ ที่ไม่เหมาะสมเช่นมากเกินไปก็จะเกิดปัญหาจำผิดพลาดได้ดังตารางที่ 4.3 แต่ในกรณีตรงกันข้าม ถ้าเรากำหนดค่า ρ ให้น้อยก็จะทำให้อินพุทบางตัวที่มีความคล้ายคลึงกับโหนดตกผิดโหนดได้

การให้ค่าความคล้าย ρ คงที่เพียงค่าเดียวในการทดสอบกับตัวอักษรทั้งชุดซึ่งมีทั้งกลุ่มที่มีความคล้ายกันมากและกลุ่มที่ไม่มีความคล้ายกับตัวอักษรอื่นเลย ไม่สามารถแก้ปัญหานี้ได้ การปรับให้แบบจำลองสามารถปรับค่าความคล้ายได้ตามกลุ่มของตัวอักษรจะสามารถแก้ปัญหานี้ได้ จากปัญหาการใช้ ARTS แยกตัวอักษรภาษาไทย

4.4 การใช้ค่าเทรสโฮลแก้ปัญหา ARTS

งานวิจัยนี้จึงใช้วิธีการปรับค่า ρ ของความคล้ายระหว่างอินพุทกับโหนดที่จะเรียนในช่วงการฝึกโครงข่ายแสดงโดยอัลกอริทึมที่ใช้สอน ARTS แบบที่มีการปรับเปลี่ยนค่า ρ ให้มีการเพิ่มและลดไม่เท่ากัน ขณะทำการฝึกจะตรวจสอบว่าอินพุทกับโหนดที่จะเรียนเป็น pattern ตัวอักษรเดียวกันหรือไม่ หากมีการผิดโหนดจะไม่ยอมให้ปรับค่าน้ำหนัก แต่จะเพิ่มค่า ρ ของโหนดนั้นให้มากขึ้นเพราะถือว่าเป็นตัวอักษรที่มีความคล้ายกับอักษรอื่นทำให้มีการตกผิดโหนด และหาค่าโหนดชนะตัวถัดไป ทดสอบจนกว่าจะพบโหนดชนะที่มีค่าความคล้ายเกินค่า ρ และเป็นตัวอักษร pattern เดียวกันกับอินพุท หรือถ้าไม่พบโหนดเช่นนี้ก็จะทำการสร้างโหนดใหม่ กรณีครบรอบของกาารเรียนรู้ด้วยการฝึกตัวอักษรครบ 1500 ตัวแล้ว' ถ้าหากว่าโหนดใดไม่มีอินพุทแบบอื่นมาตกผิดที่โหนดนั้นเลยแสดงว่าโหนดนั้นไม่คล้ายกับโหนดอื่นก็จะลดค่า ρ ให้แก่โหนดนั้น ดังนั้น ρ ของ ART โหนด จะมีค่าไม่เท่ากัน ขึ้นอยู่กับว่าโหนด นั้นเป็นโหนดของตัวอักษรในกลุ่มที่มีความคล้ายคลึงกับตัวอักษรอื่นมากหรือไม่ การทำเช่นนี้จะลดจำนวนโหนดลงด้วย

4.5 อัลกอริทึมที่ใช้สอน ARTS แบบมีการปรับเปลี่ยนค่า ρ

Step 0 Initialize parameter L and Threshold (ρ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$L > 1$, $\rho = 0.75$

Initialize weight b and t

$0<b_{ij}(0)<L/(L-1+n)$, $t_{ji}(0)=1$

Step 1 While stop condition is false

Step 2 For each training input

Step 3 Sending input signal I from F1(a) to F1(b)

and compute the norm of I ||I||

$$X_i = I_i \quad , ||I|| = \sum_{i=1}^{n} I_i$$

Step 4 Find winning Node Y_j

$$Y_j = \sum_{i=1}^{n} b_{ij} X_i$$

$Y_J \geq Y_j$ For all nodes j

Step 5 Recompute activation X of F1(b)

$$X_i = t_{ji} * I_i \quad , ||X|| = \sum_{i=1}^{n} t_{ji} * I_i$$

Step 6 Test the resemble of Input and Winning Node

ratio = $||X||/||I||$

IF ratio $> \rho$

THEN

IF Input-Pattern = Node-Pattern

goto step8

ELSE

$\rho = \rho - \alpha$

goto step7

ENDIF

ELSE

$\rho = \rho - \alpha$

go to step7

ENDIF

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Step 7 Find another winning node Y_j with max

Y_j and ratio > ρ

IF possible

THEN

goto Step 8

ELSE

create new node to learn

this pattern with

$t_{ji} = I_i$

$b_{ij}(NEW)=L*I_i / (L-1+||I||)$

ENDIF

Step 8 Update the weight for Node J

b_{ij} (NEW) = $L*X_i / (L-1+||X||)$

t_{ji} (NEW) = X_i

Step 9 Test for stopping condition

Step 10 If stop condition is False increase ρ of

unerringly training nodes

FOR (I = 1 TO N)

IF NODE I_i IS UNERRINGLY TRAING NODE

$\rho_I = \rho + \beta$

ENDIF

ENDFOR

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

ผลการทดลองใช้ค่าเทรสโฮลแก้ปัญหา ARTS

จากการทดลองด้วยอัลกอริทึมที่ใช้สอน ARTS แบบปรับเปลี่ยนค่า ρ โดยฝึกกับตัวอักษรไทย จำนวน 1500 ตัว โดยใช้ค่า ρ ต่างๆ กัน และทดสอบประสิทธิภาพการจำโดยใช้ตัวอักษรไทย 15,000 ตัว ทดสอบกับตัวอักษรที่สแกนจากกระดาษที่พิมพ์ด้วยเครื่องพิมพ์เลเซอร์ ความสว่าง 100% ค่า ρ จะเป็นค่าที่ไม่เท่ากันซึ่งเป็นผลจากการปรับค่าขณะฝึก ผลการทดสอบแสดงดังตารางที่ 5.1 เมื่อเปรียบเทียบผลการทดลองระหว่าง ARTS แบบมีการปรับเปลี่ยนค่า ρ กับแบบไม่มีการมากกว่าแบบไม่มีการปรับค่า ρ แต่ในการทดสอบการแยกแยะตัวอักษรโดยใช้ตัวอักษร 15,000 ตัว โครงข่ายที่ใช้การแก้ปัญหาด้วยการใช้ค่าเทรสโฮลที่ไม่เท่ากันสามารถแยกแยะตัวอักษรได้ดีกว่า ผลจากการแยกแยะที่ผิดพลาดซึ่งเป็นอินพุทที่ตกผิดโหนดแสดงในตาราง 5.2 จากตารางแสดงให้เห็นว่าอักษรตัวใดตกผิดไปที่โหนดใดบ้าง ตัวอักษรที่สามารถแยกแยะได้ 100% มีจำนวน 54 ตัว จากตัวอักษรทั้งหมด 75 ตัว เมื่อพิจารณาโหนดของตัวอักษรที่สามารถแยกแยะได้ถูกต้อง 100 % ซึ่งในตัวอย่างแสดงดังภาพ 5.1 เป็นภาพของโหนดตัวอักษร ก ซึ่งเป็นโหนดที่ได้จากการฝึกมี 4 โหนด จะเห็นว่าโหนดของตัวอักษรที่ได้มีความชัดเจนมากกว่าการฝึกทั้งแบบ ART และแบบ ARTS แต่เมื่อพิจารณาตัวอักษรที่มีการแยกแยะผิดมากที่สุด จะพบว่าตัวอักษร ไ จะไปตกที่โหนดของตัวอักษร ใ โดยตกผิดทั้งหมด 24 ตัวจากอินพุท ไ แบบต่างๆๆ 200 ตัว รูปที่ 5.2 แสดงภาพโหนดของตัวอักษร ไ และ ใ พบว่าโหนดของ ไ มี 2 โหนด และโหนดของ ใ มี 2 โหนด ซึ่งโหนดทั้ง 2 แบบมีความคล้ายคลึงกันมากซึ่งเป็นผลมากจากตัวอักษรทั้ง 2 แบบนี้มีความคล้ายคลึงกันอยู่แล้ว

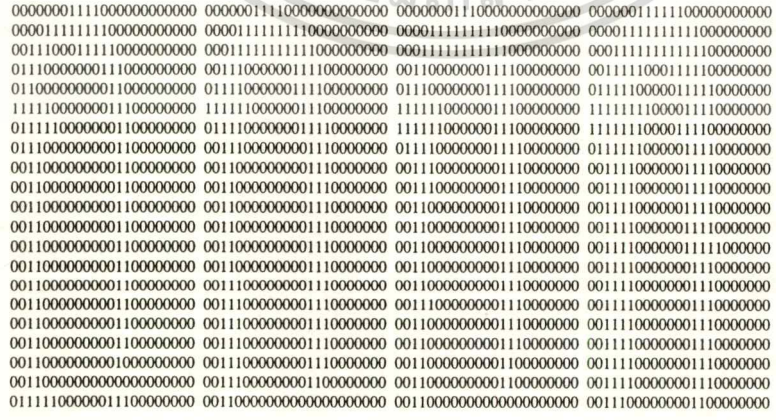
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 5.1 แสดงประสิทธิภาพการจำทดสอบโดยใช้ ARTS ทดสอบกับตัวอักษรที่สแกนจากกระดาษที่พิมพ์ด้วยเครื่องพิมพ์เลเซอร์ ความสว่าง 100%

ค่า ρ เริ่มต้นในการสอน	จำนวนรอบ	จำนวนโหนด ทั้งหมด	จำนวนโหนด ที่ใช้	ประสิทธิภาพการจำ
0.50	7	180	178	98.6466
0.55	6	218	212	99.1333
0.60	7	241	239	99.0133
0.65	6	310	304	98.7600
0.70	6	380	369	98.5133
0.75	6	495	481	98.6533
0.80	6	559	544	98.3866
0.85	6	770	759	98.1666
0.90	6	1000	983	96.7933

ประสิทธิภาพการจำทดสอบ โดยใช้ตัวอักษรไทยจำนวน 15,000 ตัว ทดสอบกับตัวอักษรที่สแกนจากกระดาษที่พิมพ์ด้วยเครื่องพิมพ์เลเซอร์ ความสว่าง 100% โดยใช้การปรับเปลี่ยนค่า ρ โดยกำหนด $\beta = 0.01$ และ $\alpha = 0.03$

รูปที่ 5.1 แสดงโหนดของตัวอักษร ก



แสดงโหนดของตัวอักษร ก ซึ่งเป็นโหนดลำดับที่ 0, 1, 133 และ 188

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

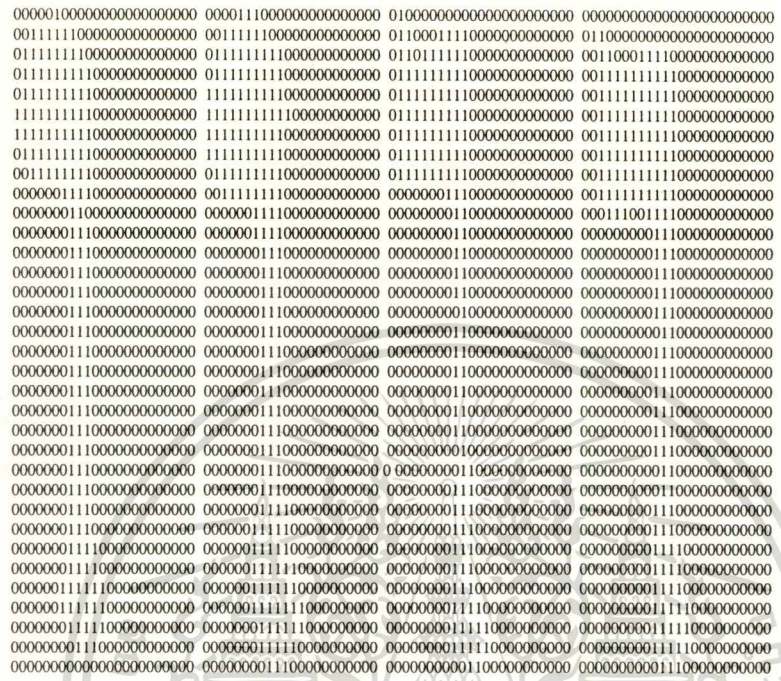
ตารางที่ 5.2 แสดงตัวอักษรที่แยกแยะผิดโดยใช้ ARTS

ตัวอักษร	จำนวนการแยกแยะผิด	ตัวอักษร	จำนวนการแยกแยะผิด	ตัวอักษร	จำนวนการแยกแยะผิด	ตัวอักษร	จำนวนการแยกแยะผิด
ก	-	ข	ช(1)	ฃ	-	ค	ต(5),ด(2),ก(7),ต(5)
ฅ	ด(4),ค(1)	ฆ	ฑ(9)	ง	-	จ	ฉ(1)
ฉ	-	ช	-	ซ	-	ฌ	ฌ(1)
ญ	-	ฎ	-	ฏ	ฏ(4)	ฐ	-
ฑ	-	ฒ	-	ณ	ฌ(10)	ด	ต(5)
ต	ด(1)	ถ	-	ท	-	ธ	-
น	-	บ	-	ป	-	ผ	-
ฝ	-	พ	ห(5)	ฟ	-	ภ	-
ม	-	ย	-	ร	-	ฤ	ถ(19)
ล	ถ(1)	ฦ	ภ(13)	ว	-	ศ	-
ษ	บ(2)	ส	-	ห	-	ฬ	-
อ	ก(1),ค(1)	ฮ	-	ฯ	-	ะ	-
า	-	0	-	1	-	2	-
3	-	4	-	5	-	6	-
7	2(1)	8	-	9	-	เ	-
โ	า(1)	ใ	ไ(1)	ไ	ใ(24), า(5)	ๅ	-
ๆ	-	๐	-	๑	-	๒	-
๓	-	๔	-	๕	-	๖	-
๗	-	๘	-	๙	-		

แสดงจำนวนโหนดที่ตกผิด ทดสอบโดยใช้ตัวอักษรไทยจำนวน 15,000 ตัว ทดสอบกับตัวอักษรที่สแกนจากกระดาษที่พิมพ์ด้วยเครื่องพิมพ์เลเซอร์ ความสว่าง 100% โดยใช้การปรับเปลี่ยนค่า ρ กำหนด $\beta = 0.01$ และ $\alpha = 0.03$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 5.2 แสดงโหนดของตัวอักษร ใ และ ไ



โหนดของตัวอักษร ใ และ ไ ซึ่งเป็นโหนดลำดับที่ 84, 210 และ 85, 179

ตารางที่ 5.3 แสดงค่า ρ ของโหนดต่างๆ ที่ปรับค่าแล้ว จะเห็นได้ว่าโหนดใดที่มีอินพุทตกผิดบ่อยๆ โหนดนั้นจะมีค่า ρ สูง แต่อย่างไรก็ตามค่า ρ ก็จะไม่สูงเกินกว่า 1 เพราะว่าเมื่ออินพุทมาตกผิดที่โหนดนั้นมากขึ้น ค่าความคล้ายหรือค่า ρ จะถูกปรับให้สูงขึ้นเรื่อยๆ จนกระทั่งไม่มีอินพุทใดสามารถจะตกผิดที่โหนดนี้ได้อีก ถึงแม้ว่าอินพุทนั้นจะเป็นโหนดชนะก็ตามเพราะ ค่า ρ ของโหนดนั้นจะสูงมากเกินกว่าที่โหนดชนะนั้นจะตกผิดได้

เมื่อพิจารณาตัวอักษร ไ ซึ่งเป็นตัวอักษรที่แยกแยะผิดมากที่สุด ตารางที่ 5.4 แสดงให้เห็นการเรียงลำดับของโหนดชนะของตัวอักษร ไ ซึ่งเป็นอินพุทตัวแรก (ไ1) ซึ่งเป็นอินพุทที่ตกถูกโหนด รูปที่ 5.3 จะแสดงอินพุทคือตัวอักษร ไ1 กับโหนดชนะลำดับที่ 1 ซึ่งเป็นโหนดของตัวอักษร ไ ลำดับที่ 85 ซึ่งพบว่าค่า ratio น้อยกว่าค่า Threshold โหนดลำดับที่ 85 นี้จึงไม่สามารถใช้เป็นโหนดเรียนได้ ส่วนรูปที่ 5.4 แสดงอินพุทคือตัวอักษร ไ1 กับโหนดชนะลำดับที่ 2 เป็นโหนดลำดับที่ 179 ซึ่งเป็นโหนดของตัวอักษร ไ เช่นกัน จะพบว่าค่า ratio มากกว่าค่า Threshold โหนดนี้จึงเป็นโหนดที่เรียนรู้ตัวอักษร ไ1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 5.3 แสดงค่า ρ ของโหนดทั้งหมดซึ่งเป็นผลที่ได้จากการฝึก

โหนด ที่	ตัวอักษร	ค่า ρที่ปรับแล้ว	โหนด ที่	ตัวอักษร	ค่า ρที่ปรับแล้ว	โหนด ที่	ตัวอักษร	ค่า ρที่ปรับแล้ว	โหนด ที่	ตัวอักษร	ค่า ρที่ปรับแล้ว
0	ก	0.53	43	ธ	0.60	86	ๆ	0.49	129	ธ	0.49
1	ก	0.68	44	น	0.56	87	ๆ	0.49	130	ธ	0.60
2	ข	0.59	45	บ	0.49	88	ๆ	0.56	131	๙	0.49
3	ข	0.86	46	บ	0.60	89	๐	0.49	132	๙	0.56
4	ฃ	0.53	47	ป	0.49	90	๐	0.69	133	ก	0.60
5	ฃ	0.74	48	ป	0.53	91	๑	0.49	134	ข	0.67
6	ค	0.70	49	ผ	0.53	92	๑	0.69	135	ฃ	0.54
7	ค	0.77	50	ฝ	0.49	93	๒	0.49	136	ค	0.72
8	ค	0.53	51	พ	0.76	94	๒	0.78	137	ค	0.67
9	ฅ	0.72	52	ฟ	0.49	95	๓	0.49	138	ฆ	0.60
10	ฆ	0.56	53	ภ	0.53	96	๔	0.49	139	ง	0.50
11	ฆ	0.65	54	ภ	0.69	97	๔	0.59	140	จ	0.60
12	ง	0.53	55	ม	0.53	98	๕	0.49	141	ฉ	0.66
13	จ	0.53	56	ย	0.56	99	๕	0.69	142	ช	0.54
14	จ	0.71	57	ร	0.49	100	๖	0.49	143	ซ	0.69
15	ฉ	0.56	58	ร	0.65	101	๖	0.53	144	ฌ	0.60
16	ฉ	0.86	59	ฤ	0.53	102	๗	0.49	145	ญ	0.60
17	ช	0.56	60	ล	0.53	103	๗	0.60	146	ฎ	0.57
18	ช	0.81	61	ล	0.72	104	๘	0.49	147	ฎ	0.73
19	ซ	0.49	62	ฦ	0.56	105	๘	0.49	148	ฏ	0.60
20	ซ	0.66	63	ฦ	0.72	106	๙	0.49	149	ฐ	0.67
21	ฌ	0.59	64	ว	0.49	107	๙	0.49	150	ฑ	0.54
22	ฌ	0.87	65	ว	0.68	108	0	0.49	151	ฒ	0.70
23	ญ	0.49	66	ศ	0.64	109	0	0.53	152	ณ	0.60
24	ญ	0.62	67	ษ	0.53	110	1	0.49	153	ด	0.61
25	ฎ	0.53	68	ษ	0.81	111	1	0.49	154	ต	0.54
26	ฎ	0.80	69	ส	0.49	112	1	0.53	155	ต	0.79
27	ฏ	0.53	70	ส	0.69	113	2	0.49	156	ถ	0.66
28	ฏ	0.88	71	ห	0.49	114	2	0.49	157	ท	0.57
29	ฐ	0.71	72	ห	0.62	115	2	0.49	158	ธ	0.57
30	ฑ	0.53	73	อ	0.49	116	3	0.49	159	น	0.69
31	ฑ	0.77	74	อ	0.56	117	3	0.49	160	บ	0.61
32	ฒ	0.49	75	ฮ	0.59	118	3	0.53	161	ผ	0.67
33	ฒ	0.69	76	ฯ	0.49	119	4	0.49	162	ฝ	0.61
34	ณ	0.49	77	ะ	0.49	120	4	0.53	163	พ	0.70
35	ณ	0.63	78	ะ	0.49	121	5	0.49	164	ภ	0.70
36	ด	0.53	79	า	0.49	122	5	0.49	165	ย	0.57
37	ด	0.78	80	า	0.60	123	5	0.49	166	ร	0.57
38	ต	0.61	81	เ	0.56	124	6	0.49	167	ฤ	0.64
39	ต	0.78	82	โ	0.57	125	6	0.49	168	ล	0.54
40	ถ	0.56	83	โ	0.59	126	7	0.49	169	ฦ	0.67
41	ถ	0.78	84	ใ	0.74	127	7	0.49	170	ศ	0.79
42	ท	0.67	85	ไ	0.61	128	7	0.49	171	ษ	0.58

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โหนดที่	ตัวอักษร	ที่ปรับแล้ว	โหนดที่	ตัวอักษร	ที่ปรับแล้ว	โหนดที่	ตัวอักษร	ที่ปรับแล้ว	โหนดที่	ตัวอักษร	ที่ปรับแล้ว
172	ห	0.70	183	ธ	0.61	194	ฎ	0.61	205	ล	0.65
173	ฬ	0.67	184	๐	0.50	195	ฐ	0.65	206	ศ	0.71
174	อ	0.67	185	๒	0.50	196	ฒ	0.65	207	ษ	0.61
175	ฮ	065	186	๗	0.50	197	ณ	0.59	208	ส	0.62
176	เ	0.54	187	๙	0.60	198	ค	0.71	209	อ	0.55
177	โ	0.61	188	ก	0.80	199	ต	0.62	210	ไ	0.62
178	โ	0.64	189	ค	0.70	200	ถ	0.64	211	ๅ	0.58
179	ไ	065	190	ค	0.77	201	ท	0.58	212	๖	0.55
180	ๅ	0.50	191	ฌ	067	202	ป	0.51	213	๒	0.51
181	๑	0.70	192	ญ	0.64	203	พ	0.71	214	ค	0.60
182	๘	0.60	193	ฏ	0.64	204	ฤ	0.62	215	ฒ	0.56
									216	ต	0.52
									217	โ	0.52

แสดงค่า ρ ที่ได้ทำการปรับเปลี่ยนค่าหลังจากฝึกสอนโครงข่ายโดยชุดฝึกเป็นตัวอักษรไทยจำนวน 1500 ตัว และใช้อัลกอริทึมแบบ ARTS ที่มีการปรับเปลี่ยนค่า ρ กำหนดค่า ρ เริ่มต้น = 0.55 และ $\beta = 0.01$ และ $\alpha = 0.03$

ตารางที่ 5.4 แสดงการเรียงลำดับของโหนดชนะของตัวอักษร ไ1

โหนดที่	Y	ρ	ratio
85 (ไ)	1.9694	0.61	0.59
179(ไ)	1.9378	0.65	0.71
84 (ไ)	1.8823	0.74	0.66
210(ไ)	1.8555	0.62	0.76
82 (โ)	1.8043	0.57	0.38

แสดงการเรียงลำดับของโหนดชนะ และค่า ration ซึ่งอินพุทคือตัวอักษร ไ1 โดยใช้ค่า ρ แบบปรับเปลี่ยนค่าได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 5.3 แสดงอินพุท ไ1 และโหนดชนะของตัวอักษร ไ ลำดับที่ 85 และจุดที่ intersect ระหว่างอินพุทกับโหนดชนะ

1000000000000	0000000000000	+
1110000000000	0100000000000	+1+
1110011111000	0110001111000	+11 +1111
1111111111000	0110111111000	+11+111111
1111111111100	0111111111000	+111111111+
1111111111100	0111111111000	+111111111+
1111111111100	0111111111000	+111111111+
1111111111100	0111111111000	+111111111+
1111111111100	0111111111000	+111111111+
1111111111100	0111111111000	+111111111+
1111111111100	0000000111000	+++++++111+
0111101111100	0000000011000	++++ ++11+
0000000111100	0000000011000	+11+
0000001111100	0000000011000	++11+
0000001111100	0000000011000	++11+
0000001111100	0000000011000	++11+
0000001111100	0000000001000	+++1+
0000000111100	0000000011000	+11+
0000000111100	0000000011000	+11+
0000000111100	0000000011000	+11+
0000000111100	0000000011000	+11+
0000000111100	0000000011000	+11+
0000001111100	0000000011000	++11+
0000001111100	0000000001000	+++1+
0000000111100	0000000001100	++11
0000000111100	0000000011100	+111
0000000111100	0000000011100	+111
0000001111100	0000000011100	++111
0000001111110	0000000011100	++111+
0000001111110	0000000011110	++1111
0000001111110	0000000011111	++1111#
0000001111111	0000000011111	++11111
0000001111110	0000000011110	++1111
0000000111110	0000000000110	++++11
0000000011100	0000000000000	+++
อินพุท ไ1	โหนดชนะ (ไ)	ส่วนที่ intersect

อินพุทคือ ไ1 ART NODE คือโหนดของตัวอักษร ไ ซึ่งเป็นโหนดลำดับที่ 85 ค่า Y = 1.9694 Threshold = 0.6100 และ ratio = 0.5917 จะเห็นได้ว่าค่า ratio น้อยกว่าค่า Threshold โหนดนี้จึงไม่สามารถใช้เรียนรู้ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 5.4 แสดงอินพุท ไ1 และโหนดชนะของตัวอักษร ไ ลำดับที่ 179 และจุดที่ intersect ระหว่างอินพุทกับโหนดชนะ

1000000000000	0000000000000	+
1110000000000	1100000000000	11+
1110011111000	0110001111000	+11 +1111
1111111111000	0111111111100	+111111111#
1111111111100	0111111111100	+1111111111
1111111111100	0111111111100	+1111111111
1111111111100	0111111111100	+1111111111
1111111111100	0111111111100	+1111111111
1111111111100	0111111111100	+1111111111
1111111111100	0111111111100	+1111111111
1111111111100	0011100111100	++111++1111
0111101111100	0000000011100	++++ ++111
0000000111100	0000000011100	+111
0000001111100	0000000011100	++111
0000001111100	0000000011100	++111
0000001111100	0000000011100	++111
0000001111100	0000000001100	+++11
0000000111100	0000000011100	+111
0000000111100	0000000011100	+111
0000000111100	0000000011100	+111
0000000111100	0000000011100	+111
0000000111100	0000000011100	+111
0000001111100	0000000011100	++111
0000001111100	0000000001100	+++11
0000000111100	0000000001100	++11
0000000111100	0000000011100	+111
0000000111100	0000000011100	+111
0000001111100	0000000011110	++111#
0000001111110	0000000011110	++1111
0000001111110	0000000011110	++1111
0000001111110	0000000011111	++1111#
0000001111111	0000000011111	++11111
0000001111110	0000000011111	++1111#
0000000111110	0000000001110	++111
0000000011100	0000000000000	+++
อินพุท ไ1	โหนดชนะ (ไ)	ส่วนที่ intersect

อินพุทคือ ไ1 ART NODE คือโหนดของตัวอักษร ไ ซึ่งเป็นโหนดลำดับที่ 179 ค่า Y = 1.9378 Threshold = 0.6500 และ ratio = 0.7158 จะเห็นได้ว่าค่า ratio มากกว่าค่า Threshold โหนดนี้จึงเป็นโหนดที่เรียนตัวอักษร ไ1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 5.5 แสดงให้เห็นถึงการเรียนโหนดผิดระหว่างอินพุทคือ ไ14 กับโหนดชนะซึ่งเป็นโหนดลำดับที่ 84 เป็นโหนดของตัวอักษร ใ รูปที่ 5.5 แสดงให้เห็นถึงความคล้ายคลึงกันระหว่างอินพุทกับ ART โหนด เนื่องจากตัวอักษรทั้ง 2 ตัวนี้มีความคล้ายคลึงกันมากค่า ratio จึงมีค่าสูงเกินกว่าค่า ρ ซึ่งเป็นตัวัดความคล้ายคลึงระหว่างตัวอักษรและอินพุท

ตารางที่ 5.6 เป็นผลการทดสอบประสิทธิภาพการจำโดยใช้ตัวอักษรไทย 15,000 ตัว ที่มีสัญญาณรบกวนซึ่งเป็นตัวอักษรที่สแกนจากกระดาษที่พิมพ์ด้วยเครื่องพิมพ์เลเซอร์ความสว่าง 50% พบว่าประสิทธิภาพการจำจะดีในช่วงการกำหนดความคล้ายระหว่างอินพุทกับโหนดชนะประมาณ 35 % ได้ประสิทธิภาพในการจำประมาณ 90% จะเห็นว่าจำนวนโหนดมากไม่มีผลต่อประสิทธิภาพการจำ และกรณีที่อินพุทเป็นตัวอักษรที่ไม่ชัดเจน การใช้ค่าความคล้าย ρ เริ่มต้นในการฝึกต่ำจะให้ประสิทธิภาพในการจำสูงกว่า

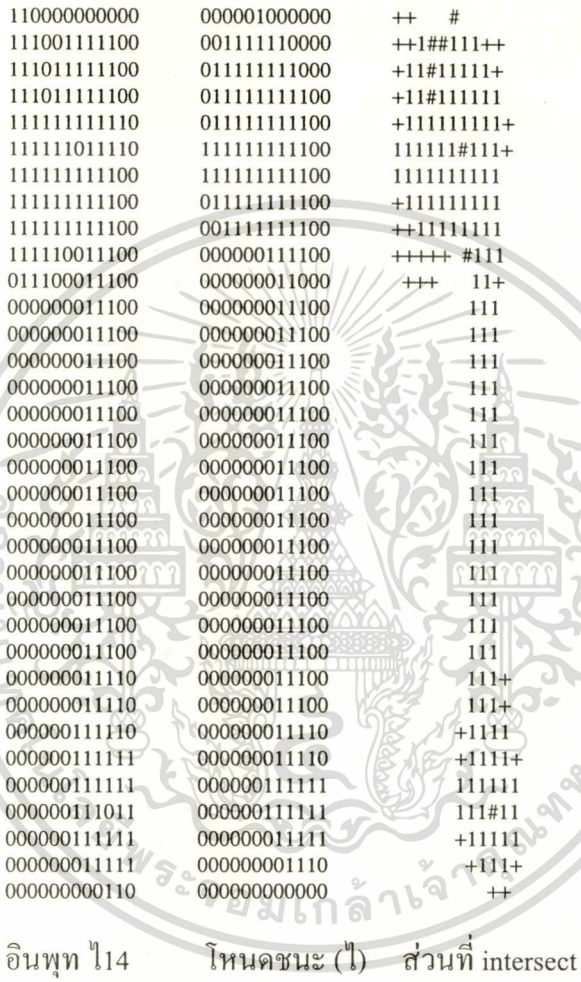
ตารางที่ 5.5 แสดงการเรียงลำดับของโหนดชนะของตัวอักษร ไ14

โหนดที่	Y	ρ	ratio
84(ใ)	1.8823	0.74	0.8121
86(า)	1.8806	0.49	0.3575
85(ไ)	1.8625	0.61	0.6906
82(โ)	1.8478	0.57	0.4861
217(โ)	1.8103	0.52	0.5966

แสดงการเรียงลำดับของโหนดชนะ และค่า ration ซึ่งอินพุทเป็นตัวอักษร ไ14 โดยใช้ค่า ρ แบบปรับเปลี่ยนค่าได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 5.5 แสดงอินพุท ไ14 และโหนดชนะของตัวอักษร ใ ลำดับที่ 84 และจุดที่ intersect ระหว่างอินพุทกับโหนดชนะ



อินพุท ไ14	โหนดชนะ (ไ)	ส่วนที่ intersect
110000000000	000001000000	++ #
111001111100	001111110000	++1##111++
111011111100	011111111000	+11#11111+
111011111100	011111111100	+11#111111
111111111110	011111111100	+111111111+
111111011110	111111111100	111111#111+
111111111100	111111111100	1111111111
111111111100	011111111100	+111111111
111111111100	001111111100	++11111111
111110011100	000000111100	+++++ #111
011100011100	000000011000	+++ 11+
000000011100	000000011100	111
000000011100	000000011100	111
000000011100	000000011100	111
000000011100	000000011100	111
000000011100	000000011100	111
000000011100	000000011100	111
000000011100	000000011100	111
000000011100	000000011100	111
000000011100	000000011100	111
000000011100	000000011100	111
000000011100	000000011100	111
000000011100	000000011100	111
000000011100	000000011100	111
000000011110	000000011100	111+
000000011110	000000011100	111+
000000111110	000000011110	+1111
000000111111	000000011110	+1111+
000000111111	000000111111	111111
000000111011	000000111111	111#11
000000111111	000000011111	+11111
000000011111	000000001110	+111+
000000000110	000000000000	++

อินพุทคือ ไ14 ART NODE คือโหนดของตัวอักษร ใ ซึ่งเป็นโหนดลำดับที่ 84 ค่า Y = 1.8823, Threshold = 0.7400 และ ratio = 0.8121 จะเห็นได้ว่าค่า ratio มากกว่าค่า Threshold โหนด นี้จึงเป็นโหนดที่เรียนตัวอักษร ไ14

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 5.6 ประสิทธิภาพการจำ ทดสอบกับตัวอักษรที่ถ่ายเอกสารจากกระดาษที่พิมพ์ด้วยเครื่องพิมพ์เลเซอร์ ความสว่าง 50% โดยใช้การปรับเปลี่ยนค่า ρ

ค่าρ เริ่มต้นในการสอน	จำนวนรอบ	จำนวนโหนดทั้งหมด	จำนวนโหนดที่ใช้	ประสิทธิภาพการจำ
0.35	7	134	134	90.5533
0.40	6	142	141	87.5733
0.45	7	159	158	85.9866
0.50	7	180	178	88.1866
0.55	6	218	212	83.4133
0.60	7	241	239	85.3266
0.65	6	310	304	86.9466
0.70	6	380	369	76.5666
0.75	6	495	481	74.1133
0.80	6	559	544	69.1400
0.85	6	770	759	64.6866
0.90	6	1000	983	48.2400

ประสิทธิภาพการจำทดสอบ โดยใช้ตัวอักษรไทยจำนวน 15,000 ตัว ทดสอบกับตัวอักษรที่สแกนจากกระดาษที่พิมพ์ด้วยเครื่องพิมพ์เลเซอร์ ความสว่าง 50% โดยใช้การปรับเปลี่ยนค่า ρ โดยกำหนด $\beta = 0.01$ และ $\alpha = 0.03$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 5.7 ประสิทธิภาพการจำ ทดสอบกับตัวอักษรที่ถ่ายเอกสารจากกระดาษที่พิมพ์ด้วยเครื่องพิมพ์เลเซอร์ ซึ่งสแกนด้วยความสว่าง 100% โดยใช้การปรับเปลี่ยนค่า ρ

ค่าρ เริ่มต้นในการสอน	จำนวนรอบ	จำนวนโหนดทั้งหมด	จำนวนโหนดที่ใช้	ประสิทธิภาพการจำ
0.35	7	134	134	97.6000
0.40	6	142	141	97.6800
0.45	7	159	158	96.8733
0.50	7	180	178	97.9133
0.55	6	218	212	97.9266
0.60	7	241	239	97.2466
0.65	6	310	304	97.1133
0.70	6	380	369	97.4200

ประสิทธิภาพการจำทดสอบ โดยใช้ตัวอักษรไทยจำนวน 15,000 ตัว ทดสอบกับตัวอักษรที่สแกนจากกระดาษที่พิมพ์ด้วยเครื่องพิมพ์เลเซอร์ ความสว่าง 50% โดยใช้การปรับเปลี่ยนค่า ρ โดยกำหนด $\beta = 0.01$ และ $\alpha = 0.03$

ตารางที่ 5.7 และตารางที่ 5.8 แสดงให้เห็นถึงผลการทดสอบประสิทธิภาพการจำโดยใช้ ARTS แบบมีการปรับเปลี่ยนค่า ρ โดยทดสอบกับตัวอักษรที่ไม่ชัดเจนซึ่งเป็นตัวอักษรชุดทดสอบ 15,000 ตัวชุดเดิม นำมาถ่ายเอกสารความสว่าง 100 % และ 50 % พบว่ากรณีที่ข้อมูลไม่ชัดเจน ประสิทธิภาพในการแยกแยะตัวอักษรจะแยกแยะได้ดีเมื่อใช้ ค่า ρ ต่ำ ในที่นี้พบว่ากรณีที่ใช้กระดาษถ่ายเอกสารความสว่าง 100 % จะแยกแยะตัวอักษรได้ดีที่สุดที่ค่า ρ = 55 % ส่วนกรณีที่ข้อมูลเป็นตัวอักษรจากกระดาษถ่ายเอกสารความสว่าง 50 % ประสิทธิภาพในการแยกแยะตัวอักษรจะแยกแยะได้ดีที่สุดที่ค่า ρ = 35 % จะพบกว่ากรณีที่ข้อมูลไม่ชัดเจนประสิทธิภาพในการแยกแยะตัวอักษรของ ARTS แบบปรับเปลี่ยนค่า ρ นี้จะค่อนข้างต่ำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 5.8 ประสิทธิภาพการจำ ทดสอบกับตัวอักษรที่ถ่ายเอกสารจากกระดาษที่พิมพ์ด้วยเครื่องพิมพ์เลเซอร์ โดยสแกนที่ความสว่าง 50% โดยใช้การปรับเปลี่ยนค่า ρ

ค่าρ เริ่มต้นในการสอน	จำนวนรอบ	จำนวนโหนดทั้งหมด	จำนวนโหนดที่ใช้	ประสิทธิภาพการจำ
0.35	7	134	134	85.6733
0.40	6	142	141	81.6800
0.45	7	159	158	79.4133
0.50	7	180	178	81.6066
0.55	6	218	212	77.5600
0.60	7	241	239	78.1533
0.65	6	310	304	77.8466
0.70	6	380	369	71.6400

ประสิทธิภาพการจำทดสอบ โดยใช้ตัวอักษรไทยจำนวน 15,000 ตัว ทดสอบกับตัวอักษรที่สแกนจากกระดาษที่พิมพ์ด้วยเครื่องพิมพ์เลเซอร์ ความสว่าง 50% โดยใช้การปรับเปลี่ยนค่า ρ โดยกำหนด $\beta = 0.01$ และ $\alpha = 0.03$

การเปรียบเทียบผลการทดลอง ART กับโครงข่ายประสาทเทียมตัวแบบ SOM, Back Propagatoin และ CPN แสดงดังตารางที่ 5.9 โดยในแต่ละโครงข่ายใช้ตัวอักษรฝึกชุดเดียวกัน (1,500 ตัว) และทดสอบการวิเคราะห์ตัวอักษรด้วยชุดทดสอบเดียวกัน (15,000 ตัว) จากการเปรียบเทียบโครงข่ายทั้ง 4 แบบพบว่าประสิทธิภาพในการแยกแยะตัวอักษรโครงข่ายแบบ Back Propagation จะให้ประสิทธิภาพในการรู้จำดีที่สุดแต่เวลาที่ใช้ในการฝึกค่อนข้างนาน และการแยกแยะตัวอักษรก็ใช้เวลามากกว่าโครงข่ายแบบอื่น ซึ่งในการใช้งานจริงนั้น ความเร็วในการแยกแยะตัวอักษรเป็นสิ่งที่สำคัญ เมื่อเปรียบเทียบโครงข่าย CPN , SOM และ ART ที่ให้ประสิทธิภาพในการแยกแยะต่ำกว่าแบบ Back Propagation จะเห็นได้ว่า ART ใช้เวลาในการฝึกและการแยกแยะตัวอักษรค่อนข้างเร็วเมื่อเปรียบเทียบกับโครงข่ายแบบอื่น แต่ประสิทธิภาพในการแยกแยะตัวอักษรยังน้อยกว่าโครงข่ายแบบอื่นเล็กน้อย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 5.9 เปรียบเทียบแบบจำลองชนิดต่างๆ ที่นำมาใช้ในการรู้จำอักษรไทย ในด้านจำนวนโหนดที่ใช้ , เวลาที่ใช้ในการฝึกโครงข่าย, เวลาที่ใช้ในการแยกแยะตัวอักษรต่อ 1 ตัว และประสิทธิภาพการรู้จำ

แบบจำลอง	จำนวนโหนดที่ใช้	เวลาที่ใช้ในการฝึกโครงข่าย	เวลาที่ใช้ในการแยกแยะตัวอักษร 1 ตัว (วินาที)	ประสิทธิภาพในการรู้จำ
ART	212	20 นาที (pentium 166)	0.12	99.13
SOM	218	6 ชั่วโมง(HP 9000/827)	0.216	99.86
Back Propagation	87	13 วัน (HP 9000/827)	1.50	99.96
CPN	174	30 นาที(HP 9000/827)	0.166	99.86

เปรียบเทียบแบบจำลองชนิดต่างๆ ที่นำมาใช้ในการรู้จำอักษรไทย โดยใช้ข้อมูลชุดฝึกและชุดทดสอบเดียวกัน ชุดฝึกเป็นอักษรไทยจำนวน 15,000 ตัว และข้อมูลชุดทดสอบใช้ตัวอักษรไทยจำนวน 15,000 ตัว ซึ่งเป็นตัวอักษรที่สแกนจากกระดาษที่พิมพ์ด้วยเครื่องพิมพ์เลเซอร์ ความสว่าง 100%

** ART ทำการฝึกโดยใช้ไมโครคอมพิวเตอร์ซีพียู pentium ความเร็ว 166 เมกกะเฮิตซ์

** SOM, Back propagation และ CPN ทำการฝึกโดยใช้เครื่องมินิคอมพิวเตอร์ HP 9000/827

ตารางที่ 5.10 แสดงผลการเปรียบเทียบแบบจำลองชนิดต่างๆ โดยใช้ข้อมูลชุดฝึกที่มีสัญญาณรบกวนแตกต่างกัน จะเห็นได้ว่ากรณีที่ข้อมูลมีสัญญาณรบกวนมาก ประสิทธิภาพในการแยกแยะของ ART จะค่อนข้างต่ำ เมื่อเปรียบเทียบกับโครงข่ายแบบอื่นถึงแม้ว่าเวลาที่ใช้ในการแยกแยะตัวอักษรแต่ละตัวจะน้อยมากก็ตาม โครงข่ายแบบ Back propagaion จะสามารถแยกแยะตัวอักษรที่มีสัญญาณรบกวนได้ดี แต่อย่างไรก็ตามเวลาที่ใช้ในการแยกแยะตัวอักษรยังค่อนข้างมากเมื่อเทียบกับโครงข่ายแบบอื่น โครงข่ายแบบ Counter propagation หรือ Self-Organizing Maps หรือ Counter-propagaion จะเหมาะสมสำหรับข้อมูลที่มีสัญญาณรบกวนมากกว่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 5.10 เปรียบเทียบประสิทธิภาพของแบบจำลองชนิดต่างๆ ในการแยกแยะตัวอักษรไทย โดยข้อมูลที่มีความชัดเจนแตกต่างกัน

ชนิดกระดาษที่ทดสอบ (% ความสว่าง)	Back propagation	Counter propagation	Self-Organizing Maps	Adaptive Resonance Theory
พิมพ์เลเซอร์ (100%)	99.96	99.86	99.86	99.13
พิมพ์เลเซอร์(50%)	98.90	97.34	97.32	90.55
ถ่ายเอกสาร(100%)	99.74	99.69	99.68	97.91
ถ่ายเอกสาร(50%)	96.23	95.69	96.50	85.67
เวลาในการแยกแยะตัวอักษรต่อ 1 ตัว (วินาที)	1.5	0.166	0.216	0.12

เปรียบเทียบแบบจำลองชนิดต่างๆ ที่นำมาใช้ในการรู้จำอักษรไทย โดยใช้ข้อมูลชุดฝึกและชุดทดสอบเดียวกัน ชุดฝึกเป็นอักษรไทยจำนวน 15,000 ตัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 6

บทสรุปและแนวทางการพัฒนาในอนาคต

6.1 สรุปผลการทดลอง

การนำ ART ในแบบจำลองเดิมมาใช้ในการจดจำตัวอักษรไทย ลักษณะการเรียนรู้จะใช้ชุดน้ำหนัก 2 ชุดคือ b และ t โดยจะเรียนรู้เฉพาะจุดที่ซ้ำกันเท่านั้นซึ่งจะเกิดปัญหามากถ้ามีการจดจำโหนดที่เรียนผิดเพราะจะทำให้โหนดที่เรียนผิดเพี้ยนไปจนไม่มีอินพุทใดสามารถตกที่โหนดนี้ได้ เมื่อทำการฝึกเสร็จแล้วพบว่า โครงข่ายไม่สามารถแยกแยะตัวอักษรได้ จึงไม่สามารถนำไปทดสอบประสิทธิภาพการจำได้ ดังนั้นงานวิจัยนี้ได้นำเอาวิธีการเรียนรู้แบบการเรียนรู้แบบมีผู้สอนมาใช้กับ ART ซึ่งพบว่าโครงข่ายสามารถแยกแยะตัวอักษรได้ ประสิทธิภาพในการแยกแยะตัวอักษรไทยได้ 99.13 % จำนวนโหนด 212 โหนด และได้นำวิธีการปรับค่าความคล้าย ρ ที่ไม่เท่ากันในแต่ละโหนดมาใช้ เพื่อให้โครงข่ายมีประสิทธิภาพในการแยกแยะตัวอักษรสูงขึ้น โดยเพิ่มขั้นตอนการลดและเพิ่มค่า ρ ในขณะที่เรียนรู้ ขณะที่ทำการเรียนรู้หากว่าโหนดใดก็ตามมีตัวอักษรอื่นที่ไม่ใช่ตัวอักษรของโหนดนั้นมาตก ก็จะทำการเพิ่มค่า ρ ในขนาดที่เท่ากับค่า α และไม่ให้โครงข่ายทำการเรียนรู้ตัวอักษรนั้น แต่ให้หาโหนดชนะใหม่ และเมื่อครบรอบของการเรียนรู้หากโหนดใดไม่มีตัวอักษรอื่นมาตกผิดเลยก็จะลดค่า ρ ในขนาดที่เท่ากับค่า β ผลการทดลองพบว่าประสิทธิภาพในการแยกแยะตัวอักษรไทยได้ค่าสูงขึ้น แต่จำนวนโหนดก็เพิ่มขึ้นด้วย

เมื่อนำแบบจำลองมาทดสอบกับตัวอักษรชุดที่มีสัญญาณรบกวน จะพบว่าแบบจำลองจะแยกแยะตัวอักษรได้ดีเมื่อใช้ค่า ρ ต่ำ ตัวอักษรชุดทดสอบที่พิมพ์ด้วยเครื่องพิมพ์เลเซอร์ ความสว่าง 50 % และชุดทดสอบจากกระดาษถ่ายเอกสารความสว่าง 50 % จะให้ผลการทดสอบที่ดีที่สุดเมื่อใช้ค่า ρ ประมาณ 35 % แต่อย่างไรก็ตาม จะเห็นได้ว่าเมื่อใช้ชุดทดสอบที่มีสัญญาณรบกวนมากโครงข่ายแยกแยะตัวอักษรผิดพลาดค่อนข้างมาก ชุดตัวอักษรที่พิมพ์ด้วยเครื่องพิมพ์เลเซอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ความสว่าง 50% จะแยกแยะตัวอักษรได้ประมาณ 90 % ชุดตัวอักษรถ่ายเอกสารความสว่าง 100 % จะสามารถแยกแยะตัวอักษรได้ประมาณ 97 % และชุดตัวอักษรถ่ายเอกสารความสว่าง 50 % จะแยกแยะตัวอักษรได้ประมาณ 80%

เวลาที่ใช้ในการเรียนรู้ของ ART ใช้เวลาน้อยมาก จากการทดลองข้างต้นใช้เวลาในการฝึกประมาณ 20 นาที เวลาที่ใช้ในการแยกแยะตัวอักษรใช้เวลาประมาณ 0.002 วินาทีต่อตัวอักษร โดยใช้เครื่องไมโครคอมพิวเตอร์ที่ใช้ซีพียู pentium ความเร็ว 166 เมกกะเฮิตซ์ จากตัวอย่างข้อมูลที่ใช้ฝึก 1500 ตัว และจำนวนข้อมูลที่ใช้ทดสอบประสิทธิภาพการแยกแยะตัวอักษร 15000 ตัวได้ประสิทธิภาพในการจำประมาณ 99% แต่อย่างไรก็ตามการกำหนดค่า ρ เริ่มต้นของตัวอักษรไม่ควรให้ค่าสูงเกินไป เพราะจะทำให้จำนวนโหนดมาก และอัลกอริทึมจะทำการปรับค่า ρ ให้เอง เมื่อเปรียบเทียบแบบจำลอง ART กับแบบจำลองอื่น ที่นำมาใช้ในการจดจำอักษรไทย พบว่า ART ใช้เวลาในการฝึกและการแยกแยะตัวอักษรค่อนข้างน้อยเมื่อเปรียบเทียบกับแบบจำลองอื่น แต่จำนวนโหนดและประสิทธิภาพในการรู้จำยังไม่ดีนักเมื่อเปรียบเทียบกับแบบจำลองอื่น

6.2 แนวทางการพัฒนาในอนาคต

การปรับปรุงโครงสร้างของแบบจำลอง ทำให้ประสิทธิภาพในการแยกแยะตัวอักษรสูงขึ้น แต่อย่างไรก็ตาม การปรับปรุงโครงสร้างอาจจะทำเพิ่มเพื่อให้ผลของการแยกแยะตัวอักษรได้ประสิทธิภาพดีขึ้นกว่าเดิมอีกนั้น อาจจะทำเพิ่มได้อีก 2 วิธีดังนี้

6.2.1 การเปลี่ยนวิธีหาค่า **ratio** การปรับโครงสร้างของแบบจำลองโดยเปลี่ยนแปลงวิธีการเรียนรู้จากการเรียนรู้แบบมีผู้สอนเป็นการเรียนรู้แบบไม่มีผู้สอน และนำวิธีการปรับค่าความคล้าย ρ ที่ไม่เท่ากันในแต่ละโหนดมาใช้กับโครงข่าย เป็นวิธีที่ช่วยให้โครงข่ายมีประสิทธิภาพในการแยกแยะตัวอักษรสูงขึ้น การที่โครงข่ายจะยอมรับว่าอินพุทที่เข้ามากับโหนดชนะมีความคล้ายคลึงกันในระดับที่สามารถยอมรับให้โหนดชนะเป็นโหนดเรียนได้หรือไม่นั้น จะขึ้นกับค่า ratio ซึ่งเป็นค่าที่ได้จากอัตราส่วนของ ขนาดของจุดที่ซ้ำส่วนด้วยขนาดของอินพุท ดังสมการที่ 6.1 ว่าค่านี้มีขนาดมากกว่า ρ หรือไม่

$$\text{ratio} = \sum_{i=1}^{n} t_{ji} I_i / \sum_{i=1}^{n} I_i \quad \text{ที่ } I_i \text{ เป็น } 1 \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad (6.1)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โหนดชนะ คือโหนดที่มีค่า Y_j สูงสุด ซึ่งค่าที่ได้นี้เป็นผลมากจากการนำอินพุทคูณกับค่าน้ำหนักของชุดน้ำหนัก b ทุกโหนด ค่านี้จะมากเมื่อจุดของโหนดที่เป็นอินพุทโหนดซ้ำกับจุดของ ART โหนดมาก และถ้า ART โหนดมีขนาดเล็กค่าน้ำหนักทุกจุดของชุดน้ำหนัก b จะมากกว่าค่าน้ำหนักแต่ละจุดของชุดน้ำหนัก b ของ ART โหนดที่มีขนาดใหญ่ ตัวอย่างเช่น ART ชุดน้ำหนัก b ของ ART โหนดที่เก็บตัวอักษร เ กับ ป นั้น ค่าน้ำหนัก b แต่ละจุดของ เ จะมากกว่า ป กรณีที่โหนดชนะมีขนาดใหญ่แต่อินพุทมีขนาดเล็ก ถึงแม้ว่าโหนดชนะจะเป็นโหนดของตัวอักษรคนละตัวกับอินพุท โหนดชนะจะเป็นโหนดเรียนได้ทันที ตัวอย่างเช่นอินพุทคือ ถ และโหนดชนะคือ ฤ ดังรูปที่ 6.1

รูปที่ 6.1 แสดงโหนดชนะ คือตัวอักษร ฤ และอินพุทคือตัวอักษร ถ

ART โหนด	อินพุทโหนด
0000011111100000	0000011111100000
0000111111110000	0001111111111000
0001100011111000	0011000001111100
0110000000011110	1111000000011110
1111100000011110	1111100000001110
1111110000011110	1111110000001110
0111111000001110	1111100000001110
0011111000000111 0	0111100000001110
0011110000000111 0	0111100000001110
0011110000000111 0	0111100000001110
0011110000000111 0	0111100000001110
0011110000000111 0	0111100000001110
0001110000000111 0	0111100000001110
0001110000000111 0	0111100000001110
0001110000000111 1	0111111000001110
0011111110000111 1	0111111110001110
0001111110000111 1	0111111110001111
0001111111000111 1	0011111110001111
0001111111000111 1	0011111100001110
0001111110000111 1	0001111100001110
0000000000000111 1	0000000000000000
0000000000000111 1	0000000000000000
0000000000000111 1	0000000000000000
0000000000000111 0	0000000000000000
0000000000000111 0	0000000000000000
0000000000000111 0	0000000000000000
0000000000000111 0	0000000000000000
0000000000000001 10	0000000000000000

การแก้ปัญหาในลักษณะเช่นนี้อาจจะทำได้โดยการเปลี่ยนขั้นตอนการหาค่า ratio ระหว่างโหนดชนะกับอินพุท โดยเปลี่ยนจากการหาอัตราส่วนระหว่างจำนวนจุดของโหนดซ้ำส่วนด้วยจำนวนจุดของอินพุทโหนด เป็นการหาอัตราส่วนระหว่างจุดที่ไม่ซ้ำระหว่างอินพุทกับโหนดทุกโหนด ส่วนด้วยจำนวนจุดของ ART โหนด โดยหาอัตราส่วนเช่นนี้กับ ART โหนดทุกโหนด และเลือกเอา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อัตราส่วนที่น้อยที่สุดเป็นโหนดเรียนแทน โหนดที่จะเป็นโหนดเรียน จะเป็นโหนดที่มีความแตกต่างระหว่างอินพุทกับ ART โหนดน้อยที่สุด กรณีนี้จะแก้ปัญหาในลักษณะนี้ได้

6.2.2 **การเรียงลำดับ ratio** การเปลี่ยนแปลงโครงสร้างในลักษณะนี้จะใช้ชุดน้ำหนักเพียงค่าเดียวคือค่า t โดยทำการหาค่า ratio ของอินพุทกับโหนดทุกโหนด จากนั้นทำการเรียงลำดับและเลือกเอาโหนดที่มีค่า ratio สูงสุดเป็นโหนดเรียนแทน

งานวิจัยนี้มุ่งเน้นในเรื่องการแยกแยะตัวอักษรภาษาไทยโดยใช้โครงข่ายประสาทเทียม ART ซึ่งตัวแบบที่ใช้สามารถใช้ได้เฉพาะแบบฟอนต์เดียว (single font) แต่ในปัจจุบัน รูปแบบตัวอักษรที่ใช้กันทั่วไปมีหลายแบบ หากนำโครงข่ายนี้ไปใช้วิเคราะห์ตัวอักษรในงานจริง โครงข่ายนี้ควรจะสามารถแยกแยะตัวอักษรแบบหลายฟอนต์ได้ หรือขนาดไม่ควรจะกำหนดให้มีขนาดเดียว น่าจะสามารถยืดหยุ่นในเรื่องขนาดของฟอนต์ด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

1. Capenter, G. A. and Grossberg S. "A Massively Parallel Architecture for a Self-Organizing Neuron Pattern Recognition Machine." Computer Vision, Graphics, and Image Processing. vol.37(1987) pp. 54-115.
2. Fausett, Laurene. Fundamentals of Neauron Networks Architectures, Algorithms and Applications. New Jersey: Prentice-Hall International Inc, 1994.
3. Healy, Micheal J. "A logical Architecture of Supervisedd Learning." IEEE Conference paper. (1993) pp. 190-195.
4. Khan, Emdadur R. "Distortion invariant hand-written digit recognition using adaptive resonance theory(ART)neuron network." IECON'90 Annual conference of IEEE Industrial Electronics Society. (1990) pp.1266-1270.
5. ชม กิ้มปาน. การรู้จำรูปแบบอักษรพิมพ์ภาษาไทยของคอมพิวเตอร์. วิทยานิพนธ์วิศวกรรมศาสตรดุษฎีบัณฑิต. กรุงเทพฯ : บัณฑิตวิทยาลัย สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง. 2529.
6. ประสาร ตังตสานนท์. การจดจำรูปแบบคัดลายมือภาษาไทย โดยวิธีแยกแยะลักษณะเด่น. วิทยานิพนธ์วิศวกรรมศาสตรมหาบัณฑิต. กรุงเทพฯ : บัณฑิตวิทยาลัย สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง. 2529.
7. สุรพันธ์ เอื้อไพบูลย์. การจดจำอักษรลายมือเขียนภาษาไทยโดยการพิจารณาหัวตัวอักษร. วิทยานิพนธ์วิศวกรรมศาสตรมหาบัณฑิต. กรุงเทพฯ : บัณฑิตวิทยาลัย สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง. 2531.
8. สุรสิทธิ์ ราตรี. การรู้จำอักษรตัวพิมพ์ภาษาไทยโดยวิธีค้นหาลักษณะโครงลักษณะโครงสร้างลายเส้น. วิทยานิพนธ์วิศวกรรมศาสตรมหาบัณฑิต. กรุงเทพฯ : บัณฑิตวิทยาลัย สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง. 2532.
9. ทวี เปรมรัตนชัย. การรู้จำอักขระตัวพิมพ์ภาษาไทยด้วยวิธีโครงข่ายนิวรอนแบบแบคพรอบพาเกชั่น. วิทยานิพนธ์วิทยาศาสตรมหาบัณฑิต. กรุงเทพ : บัณฑิตวิทยาลัย สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง. 2529.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

10. อัญชลี วานิชทวีวัฒน์. การจดจำอักขระภาษาไทยตัวพิมพ์โดยโครงข่ายประสาทเทียมแบบเซลฟ์ออร์แกนไนซิงแมป วิทยานิพนธ์วิทยาศาสตรมหาบัณฑิต. กรุงเทพ : บัณฑิตวิทยาลัย สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง. 2540.

11. อภิรักษ์ จิรายุสกุล. การวิเคราะห์ตัวพิมพ์อักษรภาษาไทยโดยใช้ CPN วิทยานิพนธ์วิทยาศาสตรมหาบัณฑิต. กรุงเทพ : บัณฑิตวิทยาลัย สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง. 2537.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก.

บทความทางวิชาการ

การจดจำอักษรพิมพ์ภาษาไทยโดยโครงข่ายประสาทเทียมแบบจำลอง ART

Printed Thai Character Recognition using ART Neural Network Model

เพ็ญพรรณ ใช้ฮวดเจริญ* บุญธีร์ เครือตราชู**

* ภาควิชาคณิตศาสตร์ สาขาวิชาวิทยาการคอมพิวเตอร์และสารสนเทศน์
สถาบันเทคโนโลยี พระจอมเกล้าเจ้าคุณทหารลาดกระบัง

** ภาควิชาวิศวกรรมคอมพิวเตอร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

บทคัดย่อ

งานวิจัยนี้เป็นการทดสอบแบบจำลองโครงข่ายประสาทเทียมทฤษฎีอะแดปทีฟรีโซแนนซ์ (ART) โดยนำมาประยุกต์ในงานจดจำตัวอักษรพิมพ์ภาษาไทย และได้ทำการปรับปรุงแบบจำลองในส่วนของการเรียนรู้ และการเปลี่ยน Threshold เพื่อลดเวลาในขั้นตอนของการเรียนรู้ และลดความผิดพลาดในการจดจำตัวอักษร โดยแสดงให้เห็นถึงปัญหาและวิธีการแก้ไขโครงสร้างของ ART เพื่อเพิ่มประสิทธิภาพในการรู้จำ งานวิจัยนี้ได้ทำการฝึกสอนโครงข่ายด้วยตัวอักษรภาษาไทย 75 แบบ แบบละ 20 ตัว และทำการทดสอบด้วยตัวอักษรภาษาไทย 75 แบบ แบบละ 200 ตัว รวมทั้งหมด 15,000 ตัว ซึ่งพิมพ์ด้วยเครื่องพิมพ์เลเซอร์ โดยใช้ฟอนต์อังศนายูพีซีขนาด 12 จุด ต่อนิ้ว จะได้ผลความถูกต้องประมาณ 99.13% ใช้เวลาในการฝึก 20 นาที และจำนวนรอบในการฝึก 6รอบ

Abstract

This research presents an application of ART neuron network model on Thai printed character recognition. To reduce the training time and the percentage error of recognition, the original ART model is modified.. This improving ART model is obtained by training 6 times with 1,500 characters of 75 Thai Character patterns. With 15,000 samples of 75 different Thai character patterns of 12 point Angsana UPC font, the percent recognition of this model is 99.13%. The overall training time is 20 minutes.

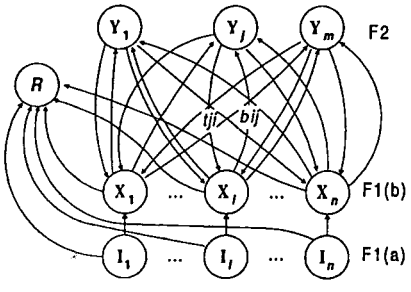
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. บทนำ

ปัจจุบันมีการศึกษาโครงข่ายประสาทเทียม และได้นำไปประยุกต์ใช้งานอย่างกว้างขวาง แบบจำลองโครงข่ายประสาทเทียมแต่ละโมเดลจะมีความสามารถที่แตกต่างกันทั้งในด้านความเร็ว, การเรียนรู้, ขนาดของหน่วยความจำที่ใช้ และความสามารถในการจดจำรูปแบบ

อะแดปทีฟรีโซแนนซ์เทียรี(ART) เป็นแบบจำลองโครงข่ายประสาทเทียม มีโครงสร้างแบบมัลติเลเยอร์(multilayer) ประกอบด้วย เลเยอร์ F1(a), เลเยอร์ F1(b) และเลเยอร์ F2 รูปแบบในการเรียนรู้ เป็นการเรียนรู้แบบไม่มีผู้สอน (Unsupervised learning) และใช้อัลกอริทึมแบบคลัสเตอร์ ข้อดีของ ART คือความเร็วในการฝึก และไม่มีปัญหาในเรื่อง local minima และ memory washout [1] ส่วนข้อจำกัดของแบบจำลองนี้พบว่า เนื่องจากวิธีการเรียนรู้เป็นแบบไม่มีผู้สอน ดังนั้น ถ้าอินพุทที่เข้ามาไม่สมบูรณ์ จะทำให้เกิดความผิดพลาดในการแยกแยะตัวอักษรได้ง่าย

การทดลองนี้ได้ศึกษาแบบจำลอง ART และนำมาทดสอบการรู้จำตัวอักษรภาษาไทย โดยได้ทำการเปลี่ยนแปลงโครงสร้างของรูปแบบการเรียนรู้ให้เป็นการเรียนรู้แบบมีผู้สอน (Supervised learning) และเปลี่ยนขั้นตอนการเพิ่มและลดขนาดของ Threshold ในช่วงของการฝึก นอกจากนี้จะรายงานการเปรียบเทียบระหว่างแบบจำลอง ART, Self-Organizing Maps(SOM), Back Propagation และ CPN ในแง่ของเวลาที่ใช้ฝึก,จำนวนโหนดที่ใช้ และประสิทธิภาพการรู้จำ



ภาพที่ 1: สถาปัตยกรรมของ ART

2. สถาปัตยกรรมของ ART

สถาปัตยกรรมของ ART แสดงในภาพที่ 1 และอัลกอริทึมในข้อ 3 โดยมีโครงสร้างที่สำคัญ 3 ส่วน คือ เลเยอร์ F1, เลเยอร์ F2 และ หน่วย R ซึ่งเป็นตัวตัดสินว่าค่าอินพุทที่เข้ามาควรจะอยู่โหนดใด เลเยอร์ F1 ประกอบด้วย 2 เลเยอร์ย่อยคือ อินพุทเลเยอร์ F1(a) และอินเตอร์เฟสเลเยอร์ F1(b) F1(a) รับข้อมูลเป็นไบนารีมีค่า 0,1 และส่งข้อมูลไปให้ F1(b) ระหว่าง เลเยอร์ F1 และ F2 จะมีค่าน้ำหนักที่ใช้เชื่อมต่ออยู่ 2 ชุด คือ ชุดน้ำหนัก b (bottom-up weight) และ t (top-down weight) ชุดน้ำหนัก b จะเชื่อมอย่างทั่วถึงระหว่างเลเยอร์ F1(b) และ F2 ส่วนชุดน้ำหนัก t จะเชื่อมอย่างทั่วถึงระหว่างเลเยอร์ F2 และ F1(b) การตัดสินว่าโหนดใดควรเป็นโหนดที่จะเรียนรู้ ทำได้โดยการหาโหนดชนะดังสมการที่ 1

$$Y_j = \sum_{i=1}^{n} b_{ij} I_i \qquad (1)$$

$$Y_J = \text{Max } Y_j \qquad (2)$$

เมื่อได้โหนดชนะ J ที่มีค่า Y_j มากที่สุดแล้ว การหาโหนดที่จะใช้ฝึก(โหนดเรียน) ต้องพิจารณาความคล้ายของอินพุท I ที่เข้ามาฝึก กับค่าน้ำหนักของโหนดชนะที่ได้ฝึกไว้แล้ว การวัดความคล้ายทำโดยใช้น้ำหนัก t_{ji} ค่าน้ำหนัก t_{ji} จะใช้เก็บจุดที่ซ้ำกัน ระหว่างอินพุทใหม่ที่เข้ามาฝึกกับโหนดเดิมที่ได้ฝึกไว้แล้ว ดังนั้น ความคล้ายของโหนดกับอินพุท สามารถวัดได้โดยการหาส่วนที่ซ้ำกัน ระหว่าง t_{ji} กับอินพุท I_i และนับจำนวน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ว่ามีจุดซ้ำกี่จุด แล้วจึงนำมา normalize กับขนาดของอินพุทที่มีค่าเป็น 1 เรียกว่า ratio

$$\text{ratio} = \sum_{i=1}^{n} t_{ji} I_i / \sum_{i=1}^{n} I_i \quad \text{ที่ } I_i \text{ เป็น } 1 \qquad (3)$$

โหนดที่จะเรียนรู้ต้องมีความคล้ายมากกว่าหรือเท่ากับ ρ ในกรณีที่โหนดชนะมีค่าความคล้ายน้อยกว่าค่า ρ จะทำการหาค่าโหนดชนะใหม่โดยนำค่า Y_j ที่มีค่าสูงสุดรองลงไปมาเป็นโหนดชนะ และทดสอบความคล้ายจนกว่าจะพบโหนดที่มีความคล้ายเกินค่า ρ และทำการปรับน้ำหนักดังสมการที่ 4-7

$$X_i = t_{ji} * I_i \qquad (4)$$

$$\|X\| = \sum_{i=1}^{n} t_{ji} * I_i \qquad (5)$$

$$t_{ji} = X_i \qquad (6)$$

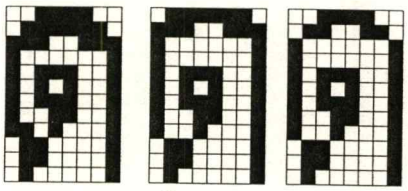
$$b_{ij} = L*X_i / (L-1+\|X\|) \qquad (7)$$

กรณีที่ไม่มีโหนดดังกล่าวก็จะทำการสร้างโหนดใหม่เก็บ Pattern I และให้ค่าน้ำหนักเริ่มต้นของ t_{ji} เป็นค่าอินพุท I_i และ initial weight b_{ij} ดังสมการที่ 8 และ 9

$$t_{ji}\ (\text{NEW}) = I_i \qquad (8)$$

$$b_{ij}\ (\text{NEW}) = L*I_i / (L-1+\sum_{i=1}^{n} I_i) \qquad (9)$$

จากสมการที่ 6, 7 การปรับชุดน้ำหนักของ t และ b จะปรับน้ำหนักเฉพาะจุดที่ซ้ำกัน (Intersection) ระหว่าง t_{ji} กับค่าของอินพุท X_n โดยจุดที่ซ้ำของชุดน้ำหนัก t จะให้ค่าเป็น 1 และจุดที่ไม่ซ้ำจะให้ค่าเป็น 0 ดังภาพที่ 2(a) ส่วนชุดน้ำหนัก b จุดที่ซ้ำจะให้ค่าน้ำหนักดังสมการที่ (7) และค่า 0 สำหรับจุดที่ไม่ซ้ำ ส่วนค่าน้ำหนักของ b จะน้อยเมื่อจำนวนจุดซ้ำมาก แต่จะมีค่ามากถ้าจำนวนจุดซ้ำน้อย ดังภาพที่ 2(b)



ภาพที่ 2(a): แสดงการเรียนตัวอักษร ด ของชุดน้ำหนัก t
** สีขาวคือค่า 0
** สีดำคือค่า 1

0	0	0.0400	0.0400	0.0400	0.0400	0	0
0	0.0400	0.0400	0.0400	0.0400	0.0400	0.0400	0
0.0400	0.0400	0.0400	0	0	0.0400	0.0400	0.0400
0.0400	0	0	0	0	0	0	0.0400
0.0400	0	0.0400	0.0400	0.0400	0	0	0.0400
0.0400	0	0.0400	0	0.0400	0	0	0.0400
0.0400	0	0.0400	0.0400	0.0400	0	0	0.0400
0.0400	0	0	0.0400	0.0400	0	0	0.0400
0.0400	0.0400	0	0.0400	0.0400	0	0	0.0400
0	0.0400	0.0400	0.0400	0	0	0	0.0400
0	0.0400	0.0400	0	0	0	0	0.0400
0	0.0400	0	0	0	0	0	0.0400

ภาพที่ 2(b) : แสดงชุดน้ำหนัก b ของ ART Node ซึ่งเก็บ pattern ของตัวอักษร ด

0	0	0.0476	0.0476	0.0476	0.0476	0	0
0	0.0476	0.0476	0	0	0.0476	0.0476	0
0.0476	0	0	0	0	0	0	0.0476
0.0476	0	0	0	0	0	0	0.0476
0.0476	0	0.0476	0.0476	0.0476	0	0	0.0476
0.0476	0	0.0476	0	0.0476	0	0	0.0476
0.0476	0	0.0476	0.0476	0.0476	0	0	0.0476
0.0476	0	0	0.0476	0.0476	0	0	0.0476
0.0476	0	0	0.0476	0	0	0	0.0476
0	0.0476	0.0476	0	0	0	0	0.0476
0	0.0476	0.0476	0	0	0	0	0.0476
0	0.0476	0	0	0	0	0	0.0476

ภาพที่ 2(c):คือชุดน้ำหนัก b ของโหนดเดิมเมื่อผ่านการเรียนรู้

3. อัลกอริทึมที่ใช้สอน ART

Step 0 Initialize parameter L and Threshold (ρ)

$L > 1$, $\rho = 0.75$

Initialize weight b and t

$0 < b_{ij}(0) < L/(L-1+n)$, $t_{ji}(0) = 1$

Step 1 While stop condition is fasle

Step 2 For each training input

Step 3 Sending input signal I from F1(a) to

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

F1(b) and compute the norm of I ||I||

$$X_i = I_i \quad , \|I\| = \sum_{i=1}^{n} I_i$$

Step 4 Find winning Node Y_j

$$Y_j = \sum_{i=1}^{n} b_{ij} X_i$$

$\mathbf{Y_J} \geq Y_j$ For all nodes j

Step 5 Recompute activation X of F1(b)

$$X_i = t_{ji} * I_i \quad , \|X\| = \sum_{i=1}^{n} t_{ji} * I_i$$

Step 6 Test the resemble of Input and Winning Node

ratio = ||X|| / ||I||

IF ratio > ρ

THEN

goto step8

ENDIF

Step 7 Find another winning node Y_j with max Y_j and ratio > ρ

IF possible

THEN

goto Step 8

ELSE

create new node to learn this pattern with

$t_{ji} = I_i$

b_{ij} (NEW) = L*I_i / (L-1+||I||)

ENDIF

Step 8 Update the weight for Node J

b_{ij} (NEW) = L*X_i / (L-1+||X||)

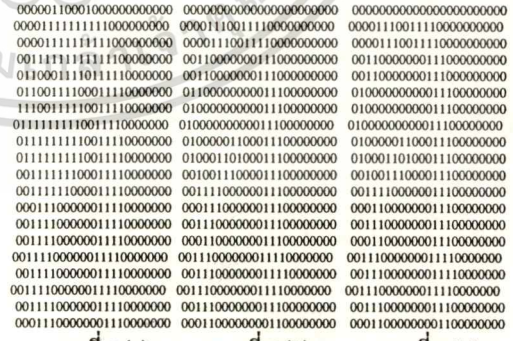
t_{ji} (NEW) = X_i

Step 9 Test for stopping condition

4. การทดลองแยกตัวอักษรภาษาไทยโดยใช้ ART

การทดลองนี้ใช้ตัวอักษรพิมพ์ภาษาไทย 75 แบบ ประกอบด้วยตัวพยัญชนะไทย 44 ตัว และเลขไทย เลขอารบิค สระ า เ ใ ๅ า ฤ ฦ ไม้หันอากาศ โดยใช้แบบตัวอักษรอังศนายูพีซี ขนาด 12 จุด ความละเอียดในการสแกนเท่ากับ 300dpi และ scale100 ในการสอนและทดสอบการจำ จำนวนชุดในฝึกประกอบด้วยตัวอักษรไทยตัวละ 20 แบบรวมทั้งหมด 1500 ตัว เมตริกซ์ของอินพุทที่ใช้ในการทดลองมีขนาด 27x37 และวางตัวอักษรในตำแหน่งมุมซ้ายบนของอินพุทเมตริกซ์

จากการทดลอง โครงข่ายจะเริ่มเสถียรในรอบที่ 80 คือไม่มีการสร้างโหนดใหม่เพิ่มขึ้นอีก การฝึกจึงสิ้นสุดลง ผลการทดลองพบว่าโครงข่ายยังไม่สามารถแยกตัวอักษรได้ ในรอบสุดท้ายยังพบว่าบางโหนดมีตัวอักษรมากกว่า 1 ตัวตกอยู่โหนดเดียวกัน ตัวอย่างเช่น ก ถ จะตกโหนดเดียวกันเสมอ โครงข่ายนี้จึงไม่สามารถนำมาใช้ได้ เพราะในการทดสอบการแยกแยะตัวอักษรหากมีอินพุทตกในโหนดลักษณะนี้ จะไม่สามารถแยกแยะได้ว่าอินพุทที่เข้ามานั้นเป็นตัวอักษรแบบใด ผลการทดสอบโดยใช้ค่า ρ = 0.75 จะสิ้นสุดการฝึกในรอบที่ 80 โดยได้จำนวนโหนดทั้งสิ้น 499 โหนด มีจำนวนโหนดที่จำผิดพลาด 61 โหนด แสดงโหนดที่ผิดพลาดในตารางที่ 1



ภาพที่ 3(a) ภาพที่ 3(b) ภาพที่ 3(c)

ภาพที่ 3: ภาพที่ 3(a)โหนดของตัวอักษร ค ภาพที่3(b) โหนดเดิม หลังการเรียน 1 รอบด้วยตัวอักษร 1500 ตัว ค่า ρ เท่ากับ 0.60 โดยตัวอักษรที่เข้าฝึกโหนดในรอบนี้คือ ค15 ค16 ค17 ค20 ด1 ด4 ด6 ด13 ด19 ภาพที่ 3 (c) โหนด เดิมหลังการเรียนรอบที่ 2 ค่า ρ เท่ากับ 0.75 โดยตัวอักษรที่ทำการฝึกโหนดในรอบนี้คือ ค9 ด1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โหนดที่	ตัวอักษรที่เรียนผิดโหนด	โหนดที่	ตัวอักษรที่เรียนผิดโหนด	โหนดที่	ตัวอักษรที่เรียนผิดโหนด	โหนดที่	ตัวอักษรที่เรียนผิดโหนด
3	ก ถ	148	ฮ 6	260	ด ๑	402	อ 9
14	ค ฅ ต	158	ฌ 9	268	ฐ 8	416	๐ 6
57	ฒ 9	160	ฐ 9	269	ฐ 8	418	ฮ 6
71	ก 0	164	ฌ 9	271	ท ห	420	ร ะ
72	ฉ 9	166	ง า	288	ฮ 7	431	ฮ 8
74	ท ห	167	ฐ 8	289	ฮ 9	433	ธ 7
80	ท ห	176	ฒ 9	291	ฮ 8	448	ฐ 8
83	ท พ	177	ค ศ	321	โ ๅ	451	ธ 7
85	บ ษ	178	ค ฅ	331	ฮ 8	454	ไ ไ
107	ย 6	182	ธ 7	332	ฐ 8	459	เ 1
109	ย 6	185	ฉ 9	360	ด ต	459	เ 1
120	ค ฅ	212	ค ฅ	369	ฐ 8	466	โ ๅ
129	บ ษ	214	ค ศ	374	ฮ 8	469	โ ๅ .
130	ภ ฦ	221	ด ๑	375	ด ต	472	ฮ 9
146	ธ 7	239	น ห	386	ฮ 8	485	ว า
						490	ฌ 9

ตารางที่ 1: แสดงการเรียนผิดโหนด ผลการฝึกรอบที่ 80 จำนวนโหนดทั้งหมด 499 โหนด $\rho = 0.75$ พบว่ายังไม่สามารถแยกตัวอักษรได้

5. ปัญหาและการปรับปรุงเพื่อนำ ART มาใช้แยกตัวอักษรภาษาไทย

ปัญหาที่เกิดขึ้นคือ ขณะฝึกโครงข่ายไม่สามารถแยกตัวอักษรไทยได้ เนื่องจากการเรียนของ ART เป็นแบบการเรียนโดยไม่มีผู้ฝึกสอน ART จะเลือกทำการปรับน้ำหนักโหนดชนะเพียงโหนดเดียว และทำการปรับชุดน้ำหนักทั้ง b และ t เนื่องจากลักษณะของอักษรไทยบางตัวจะมีความคล้ายกันมาก เช่น ดตคฅ เมื่อผ่านการสแกนจะทำให้ภาพของตัวอักษรมีความคล้ายกันมากยิ่งขึ้น กรณีเช่นนี้จึงทำให้มีการเรียนผิดโหนดได้ ดังภาพที่ 3 การแก้ไขโดยการใช้ค่า ρ ให้มีค่าสูงๆ จะแก้ปัญหาการเรียนผิดโหนดได้แต่ว่าจะเกิดโหนดจำนวนมาก ตัวอย่างเช่น อักษร ก แต่ละแบบก็จะกลายเป็นแต่ละโหนดไปจึงไม่สามารถแก้ไขโดยวิธีนี้ได้ การแก้ไขทำได้โดยการเปลี่ยนวิธีการเรียนรู้ของ ART จากวิธีการเรียนรู้แบบไม่มีผู้สอน (Unsupervised ART) ให้เป็น ART ที่มีวิธีการเรียนรู้แบบมีผู้สอน (Supervised ART)

6. Supervised ART

Supervised ART(ARTS) เป็นการฝึกให้แบบจำลองมีการปรับน้ำหนักในลักษณะเดียวกับ ART แต่มีข้อแตกต่างคือในช่วงการฝึก จะเพิ่มการทดสอบว่าอินพุทกับโหนดชนะที่มีค่าความคล้ายเกิน ρ นั้นเป็นตัวอักษรเดียวกันหรือไม่ แสดงการเปลี่ยนแปลงอัลกอริทึมจากอัลกอริทึมที่ 1 ดังนี้

Step6 Test the resemble of Input and Winning Node

ratio = ||XII/ IIII

IF ratio > ρ

THEN

IF Input-Pattern = Node-Pattern

goto step8

ELSE

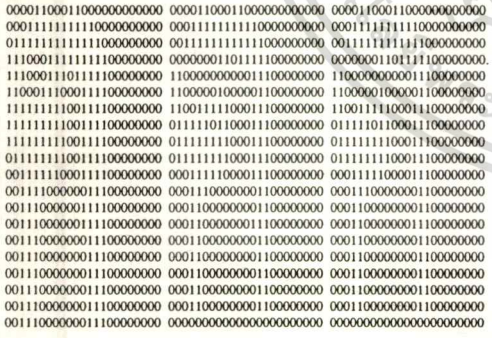
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

goto step7
ENDIF
ELSE
go to step7
ENDIF

โดย ARTS จะทำการปรับน้ำหนักเฉพาะโหนดชนะที่มีค่าความคล้ายเกินค่า ρ และ pattern ของอินพุทจะต้องเป็นตัวเดียวกับ pattern ในโหนด เรียน ดังนั้นการเรียนของโหนดชนะจะเป็นการปรับน้ำหนักเฉพาะตัวอักษรแบบเดียวกันเท่านั้น โหนดจะไม่ถูกปรับค่าอย่างผิดๆ เหมือนลักษณะการเรียนรู้แบบไม่มีผู้สอน

7. การทดลองแยกตัวอักษรภาษาไทยโดยใช้ ARTS

การทดลอง ARTS โดยใช้อัลกอริทึมที่ปรับเปลี่ยนเป็นแบบมีผู้สอน สามารถแก้ปัญหาการตกผิดของอินพุทโหนด ขณะฝึก การเรียนของโหนดชนะจะเป็นการปรับน้ำหนักเฉพาะตัวอักษรแบบเดียวกันเท่านั้น โหนดจะไม่ถูกปรับค่าอย่างผิดๆ เหมือนลักษณะการเรียนรู้ของ ART แบบไม่มีผู้สอน



ภาพที่ 4(a) ภาพที่ 4(b) ภาพที่ 4(c)

ภาพที่ 4: ภาพที่ 4(a): โหนดของตัวอักษร ศ ภาพที่ 4(b): โหนดเดิม หลังการเรียน 1 รอบด้วยตัวอักษร 1500 ตัว ค่า ρ เท่ากับ 0.75 โดยตัวอักษรที่เข้าฝึกโหนดในรอบนี้คือ ศ1 ศ2 ศ3 ศ4 ภาพที่ 4(c): โหนดเดิม หลังการเรียนรอบที่ 2 ค่า ρ เท่ากับ 0.75 โดยตัวอักษรที่ทำการฝึกโหนดในรอบนี้คือ ศ4

ρในช่วงการฝึก	จำนวนรอบ	จำนวนโหนดทั้งหมด	จำนวนโหนดที่ใช้	ประสิทธิภาพการจำ
0.55	4	158	158	97.0666
0.60	5	193	193	97.9466
0.65	5	258	258	97.6666
0.70	6	352	351	97.5066
0.75	6	468	465	96.5400
0.80	6	598	599	95.4733
0.85	6	797	796	94.4800
0.90	4	1000	1000	92.4272

ตารางที่ 2: ประสิทธิภาพการจำทดสอบ โดยใช้ตัวอักษรไทยจำนวน 15,000 ตัวโดยใช้ค่า ρ ขณะทำการทดสอบ = 0.75

ผลการทดลองแสดงให้เห็นดังภาพที่ 4 เมื่อนำโครงข่ายมาทดสอบประสิทธิภาพของการรู้จำโดยกำหนดขนาด ρ ขณะทดสอบให้มีค่าเดียวกับขณะฝึก และได้สรุปผลการทดลองดังตารางที่ 2

8. ปัญหาและการปรับปรุงเพื่อนำ ARTS มาใช้แยกตัวอักษรภาษาไทย

ปัญหาที่พบในข้อ 4 เป็นปัญหาที่เกิดขึ้นขณะเรียน แต่ปัญหาที่เกิดขึ้นในข้อนี้เป็นปัญหาการแยกแยะโหนดผิดที่เกิดขึ้นขณะทดสอบการจำ เนื่องจากตัวอักษรไทยบางกลุ่มมีความคล้ายคลึงกันมาก เช่น คดตฅ ดังนั้นในขณะทดสอบการจำตัวอักษรหากกำหนดค่า ρ ให้มีค่าน้อย ตัวอักษรอื่นที่คล้ายคลึงกันก็อาจจะตกผิดกลุ่มได้ เช่น ด ต หรือ ค ฅ กำหนดให้อินพุทขณะทำการทดสอบคือ ฅ พบว่าค่า Y_j ของโหนดชนะจะเป็นโหนดที่ 8 ซึ่งเป็นโหนดของ ฅ ที่มีค่าน้ำหนักสูงที่สุด แต่เมื่อนำมาทดสอบความคล้ายระหว่างโหนดชนะกับจำนวนจุดของอินพุทพบว่าค่าความคล้ายไม่ผ่าน โดยกำหนด ρ = 0.75 ดังนั้นหากกำหนดค่า ρ น้อยเกินไปก็จะเกิดปัญหาจำผิดพลาดได้ดังตารางที่ 3 แต่ในกรณีตรงกันข้าม ถ้าเรากำหนดค่า ρ ให้มากเพื่อกันไม่ให้ตัวอักษรที่มีความคล้ายกันตกผิดโหนด ก็จะปัญหาไม่สามารถจำได้ ตัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อย่างเช่นอักษร ก จะต้องมีความเหมือนกันจนเกือบจะเป็นตัวอักษรเดียวกันจึงจะยอมรับว่าอินพุทที่เข้ามาตกในโหนดนี้ ไม่เช่นนั้นก็จะไม่ยอมรับ การให้ค่าความคล้าย ρ คงที่เพียงค่าเดียวในการทดสอบ

โหนดลำดับที่	โหนดชนะเรียงลำดับจากมากไปน้อย	Yj	ratio
8	ศ	0.891432	0.7410
47	ด	0.843279	0.7520
9	ศ	0.811133	0.8354
144	ศ	0.731223	0.8966
408	ด	0.689567	0.7856

ตารางที่ 3: แสดงโหนดชนะเรียงลำดับตามค่า Y_j ของอินพุท ศ ขณะทำการทดสอบการจำ โดยใช้ค่า $\rho = 0.75$ โหนดชนะที่มีค่า Y_j สูงสุดคือ ศ ลำดับโหนดที่ 8

กับตัวอักษรทั้งชุดซึ่งมีทั้งกลุ่มที่มีความคล้ายกันมากและกลุ่มที่ไม่มีความคล้ายกับตัวอักษรอื่นเลย ไม่สามารถแก้ปัญหานี้ได้ การปรับให้แบบจำลองสามารถปรับค่าความคล้ายได้ตามกลุ่มของตัวอักษรจะสามารถแก้ปัญหานี้ได้จากปัญหาการใช้ ARTS แยกตัวอักษรภาษาไทย

9. การใช้ค่าเทรสโฮลแก้ปัญหา ARTS

งานวิจัยนี้จึงใช้วิธีการปรับค่า ρ ของความคล้ายระหว่างอินพุทกับโหนดที่จะเรียนในช่วงการฝึกโครงข่าย แสดงโดยอัลกอริทึมในตารางที่ 3 โดยปรับค่า ρ ให้มีการเพิ่มและลดไม่เท่ากัน ขณะทำการฝึกจะตรวจสอบว่าอินพุทกับโหนดที่จะเรียนเป็น pattern ตัวอักษรเดียวกันหรือไม่ หากมีการผิดโหนดจะไม่ยอมให้ปรับค่าน้ำหนัก แต่จะเพิ่มค่า ρ ของโหนดนั้นให้มากขึ้นเพราะถือว่าเป็นตัวอักษรที่มีความคล้ายกับอักษรอื่นทำให้มีการตกผิดโหนด และหาค่าโหนดชนะตัวถัดไป ทดสอบจนกว่าจะพบโหนดชนะที่มีค่าความคล้ายเกินค่า ρ และเป็นตัวอักษร pattern เดียวกันกับอินพุท หรือถ้าไม่พบโหนดเช่นนี้ก็จะทำการสร้างโหนดใหม่ กรณีครบรอบของการเรียนรู้ด้วยการฝึกตัวอักษรครบ 1500 ตัวแล้ว ถ้าหากว่าโหนดใดไม่มีอินพุทแบบอื่นมาตกผิดที่โหนดนั้นเลยแสดงว่าโหนดนั้นไม่คล้ายกับโหนดอื่นก็จะลดค่า ρ ให้แก่โหนดนั้น ดังนั้น ρ ของ ART โหนดจะมีค่าไม่เท่ากัน ขึ้นอยู่กับว่าโหนดนั้นเป็นโหนดของตัวอักษรในกลุ่มที่มีความคล้ายคลึงกับตัวอักษรอื่นมากหรือไม่ การทำเช่นนี้จะลดจำนวนโหนดลงด้วย

10. อัลกอริทึมที่ใช้สอน ARTS

Step 0 Initialize parameter L and Threshold (ρ)

$L > 1$, $\rho = 0.75$

Initialize weight b and t

$0<b_{ij}(0)<L/(L-1+n)$, $t_{ji}(0)=1$

Step 1 While stop condition is fasle

Step 2 For each training input

Step 3 Sending input signal I from F1(a) to F1(b) and compute the norm of I $\|I\|$

$$X_i = I_i \quad , \|I\| = \sum_{i=1}^{n} I_i$$

Step 4 Find winning Node Y_j

$$Y_j = \sum_{i=1}^{n} b_{ij} X_i$$

$Y_j \geq Y_j$ For all nodes j

Step 5 Recompute activation X of F1(b)

$$X_i = t_{ji} * I_i \quad , \|X\| = \sum_{i=1}^{n} t_{ji} * I_i$$

Step 6 Test the resemble of Input and Winning Node

ratio = $\|X\| / \|I\|$

IF ratio > ρ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

THEN

IF Input-Pattern = Node-Pattern

THEN

goto step8

ELSE

$\rho = \rho + \alpha$

goto step7

ENDIF

1. ELSE

$\rho = \rho + \alpha$

go to step7

ENDIF

Step 7 Find another winning node Y_j with max Y_j and ratio > ρ

IF possible

THEN

goto Step 8

ELSE

create new node to learn this pattern with

$t_{ji} = I_i$

$b_{ij}(NEW) = L*I_i / (L-1+\|I\|)$

ENDIF

Step 8 Update the weight for Node J

$b_{ij}(NEW) = L*X_i / (L-1+\|X\|)$

$t_{ji}(NEW) = X_i$

Step 9 Test for stopping condition

Step 10 If stop condition is False increase ρ of unerringly training nodes

FOR (I = ๙ TO N)

IF NODE I_i IS UNERRINGLY TRAING NODE

$\rho_i = \rho - \beta$

ENDIF

ENDFOR

11. การทดลองใช้ค่าเทรสโฮลแก้ปัญหา ARTS

จากการทดลองด้วยอัลกอริทึมในข้อที่ 10 โดยฝึกกับตัวอักษรไทยจำนวน 1500 ตัว ใช้ค่า ρ ต่างๆ กัน และทดสอบประสิทธิภาพการจำโดยใช้ตัวอักษรไทย 15,000 ตัว ทดสอบกับตัวอักษรที่สแกนจากกระดาษที่พิมพ์ด้วยเครื่องพิมพ์เลเซอร์ ความสว่าง 100% โดยค่า ρ เป็นค่าที่ไม่เท่ากันซึ่งเป็นผลจากการปรับค่าขณะฝึก ผลการทดสอบแสดงดังตารางที่ 4 จะพบว่ามีจำนวนโหนดที่ใช้จริงมากกว่าการฝึกแบบ ARTS ในการทดสอบกับตัวอักษร 15,000 ตัวพบ

ค่าρ เริ่มต้นในการสอน	จำนวนรอบ	จำนวนโหนดทั้งหมด	จำนวนโหนดที่ใช้	ประสิทธิภาพการจำ
0.50	7	180	178	98.6466
0.55	6	218	212	99.1333
0.60	7	241	239	99.0133
0.65	6	310	304	98.7600
0.70	6	380	369	98.5133
0.75	6	495	481	98.6533
0.80	6	559	544	98.3866
0.85	6	770	759	98.1666
0.90	6	1000	983	96.7933

ตารางที่ 4: ประสิทธิภาพการจำทดสอบ โดยใช้ตัวอักษรไทยจำนวน 15,000 ตัว ทดสอบกับตัวอักษรที่สแกนจากกระดาษที่พิมพ์ด้วยเครื่องพิมพ์เลเซอร์ ความสว่าง 100% โดยใช้การปรับเปลี่ยนค่า ρ โดยกำหนด β = 0.01 และ α = 0.03

ค่าρ เริ่มต้นในการสอน	จำนวนรอบ	จำนวนโหนดทั้งหมด	จำนวนโหนดที่ใช้	ประสิทธิภาพการจำ
0.50	7	180	178	88.1866
0.55	6	218	212	83.4133
0.60	7	241	239	85.3266
0.65	6	310	304	86.9466
0.70	6	380	369	76.5666
0.75	6	495	481	74.1133
0.80	6	559	544	69.1400
0.85	6	770	759	64.6866
0.90	6	1000	983	48.2400

ตารางที่ 5: ประสิทธิภาพการจำทดสอบ โดยใช้ตัวอักษรไทยจำนวน 15,000 ตัว ทดสอบกับตัวอักษรที่สแกนจากกระดาษที่พิมพ์ด้วยเครื่องพิมพ์เลเซอร์ ความสว่าง 50%

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยใช้การปรับเปลี่ยนค่า ρ โดยกำหนด β = 0.01 และ α = 0.03
ว่าโครงข่ายที่ใช้การแก้ปัญหาด้วยการใช้ค่าเทรสโฮล ไม่เท่ากันสามารถแยกแยะตัวอักษรไทยได้ดีกว่า แต่จำนวนรอบที่ใช้ในการฝึกมากกว่า ส่วนในตารางที่ 5 เป็นการทดสอบประสิทธิภาพการจำโดยใช้ตัวอักษรไทย 15,000 ตัว ที่มีสัญญาณรบกวนซึ่งเป็นตัวอักษรที่สแกนจากกระดาษที่พิมพ์ด้วยเครื่องพิมพ์เลเซอร์ ความสว่าง 50% พบว่าประสิทธิภาพการจำจะดีในช่วงการกำหนดความคล้ายระหว่างอินพุทกับโหนดชนะระหว่าง 50-65% จะเห็นว่าจำนวนโหนดมากไม่มีผลต่อประสิทธิภาพการจำ

12. สรุปผลการทดลอง

การนำ ART ในแบบจำลองเดิมมาใช้ในการจดจำตัวอักษรไทย ลักษณะการเรียนรู้จะใช้ชุดน้ำหนัก 2 ชุดคือ b และ t โดยจะเรียนรู้เฉพาะจุดที่ซ้ำกันเท่านั้นซึ่งจะเกิดปัญหามากถ้ามีการจดจำโหนด ที่เรียนผิดเพราะจะทำให้โหนดผิดเพี้ยนไปจนไม่มีอินพุทใดสามารถตกที่โหนดนี้ได้ ดังนั้นงานวิจัยนี้ได้นำเอาวิธีการเรียนรู้แบบการเรียนรู้แบบมีผู้สอนมาใช้กับ ART และได้นำวิธีการปรับค่าความคล้าย ρ ที่ไม่เท่ากันในแต่ละโหนดมาใช้ เพื่อให้โครงข่ายมีประสิทธิภาพในการแยกแยะตัวอักษรสูงขึ้น เวลาที่ใช้ในการเรียนรู้ของ ART ใช้เวลาน้อยมาก จากการทดลองข้างต้นใช้เวลาในการฝึกประมาณ 20 นาที โดยใช้เครื่องไมโครคอมพิวเตอร์ที่ใช้ซีพียู pentium ความเร็ว 166 เมกกะเฮิตซ์ จากตัวอย่างข้อมูลที่ใช้ฝึก 1500 ตัว ได้ประสิทธิภาพในการจำประมาณ 99% แต่อย่างไรก็ตามการกำหนดค่า ρ เริ่มต้นของตัวอักษรไม่ควรให้ค่าสูงเกินไป เพราะจะทำให้จำนวนโหนดมากและอัลกอริทึมจะทำการปรับค่า ρ ให้เอง

เมื่อเปรียบเทียบแบบจำลอง ART กับแบบจำลองอื่น ที่นำมาใช้ในการจดจำอักษรไทย พบว่า ART ใช้เวลาค่อนข้างน้อยเมื่อเปรียบเทียบกับแบบจำลองอื่น แต่จำนวนโหนดประสิทธิภาพในการรู้จำยังไม่ดีนักเมื่อเปรียบเทียบกับแบบจำลองอื่น

13. เอกสารอ้างอิง

1. Khan R. Emdadur, "Distortion invariant hand-written digit recognition using adaptive resonance theory(ART)neuron network", IECON'90 Annual conference of IEEE Industrial Electronics Society,pp.1266-1270,1990.

2. Micheal J.Healy, "A logical Architecture of Supervisedd Learning", IEEE Conference paper, pp. 190-195, 1993.

3. G. A. Carpenter S. Grossberg,"A Massively Parallel Architecture for a Self-Organizing Neural Pattern Recognition Machine", Computer Vision, Graphics, and Image Processing, vol.37, 54-115, 1987.

4.Laurene Fausett,"Adaptive Resonance Theory", Fundamentals of Neural Networks Architectures, Algorithms and Applications, Prentice-Hall, pp.218-246,1993.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ข.

ตัวโปรแกรมของระบบการรู้จำ

1. โปรแกรม PJ7.C

โปรแกรมที่ใช้ในการรู้จำตัวอักษร

```
/*************************************************************
Program trining ART model
Programmer Penpun C.
*************************************************************/
#include <stdio.h>
#include <stdlib.h>
#ifdef __TURBOC__
#include <alloc.h>
#else
#include <malloc.h>
#endif
#include "art.h"

struct putstruct{
        int putcluster;
   int wrongcluster;
   char putpattern;
        float  putthres;
};

long  TotalAllocate = 0;
void *mymalloc(size_t size, char *s);
FILE *myfopen(char *s, char *mode);
float (*b)[MAXNUMBER][MAXCLUSTER];
int (*t)[MAXNUMBER][MAXCLUSTER];
struct  putstruct (*put1)[MAXCLUSTER];

main()
{
FILE *infile, *outfile;
```

เอกสารนี้มีลิขสิทธิ์การที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
FILE *boutfile, *toutfile, *dinfile;
FILE *putinfile, *putoutfile;
char buf[BUFSIZE];
char pattern[5];
int (*s)[MAXNUMBER],(*posarr)[MAXCLUSTER];
float (*y)[MAXCLUSTER],(*max)[MAXCLUSTER];
int count,count1,i,j,norms,normx,maxindex,posindex,flag=0;
int loop = 1;
float temp, value = 0;
float *rj;
int number,k,valuex,valuey,valueindex=0;
float threshold;
int dflag=0;
float newratio;
float ratio;
int inwalk = 0,flagb=0;
float myvalue=0;
float myratio=0,rb=0;
int mynormx=0;

int tempput1;
char tempput2;
float tempput3;
int Pflag = 0;

int walk=0;
float sumb=0,sums = 0;
int sumt=0, suminter=0;

        b = mymalloc(sizeof(*b), "b");
        t = mymalloc(sizeof(*t), "t");
        y = mymalloc(sizeof(*y), "y");
        max = mymalloc(sizeof(*max), "max");
        s = mymalloc(sizeof(*s), "s");
        put1 = mymalloc(sizeof(*put1),"put1");
        posarr = mymalloc(sizeof(*posarr), "posarr");

        printf("\nTotal allocate = %8u\n", TotalAllocate);
        printf("Max cluster    = %8d\n", MAXCLUSTER);
        printf("Max number     = %8d\n", MAXNUMBER);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/* exit(1); */
binit();
tinit();

infile=fopen(INPUT_FILE,"r");
if(infile == NULL){
        printf("Can't read input file!\n");
        exit(1);
}
else
        printf("Open INPUTFILE for read\n");

outfile=fopen(OUTPUT_FILE,"w");
if(     outfile== NULL){
        printf("Can't write output file\n");
        exit(1);
}
else
        printf("Open pj.out for write\n");

boutfile=fopen("binit.out","w");
if(     boutfile== NULL){
        printf("Can't write boutput file\n");
        exit(1);
}
else
        printf("Open binit.out for write\n");

toutfile=fopen("tinit.out","w");
if(     toutfile== NULL){
        printf("Can't write toutput file\n");
        exit(1);
}
else
        printf("Open tinit.out for write\n");

putinfile=fopen("putin.dat","r");
if( putinfile==NULL){
        printf("Can't read putin.dat \n");
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            exit(1);
}
else
            printf("Open putin.dat for read\n");

putoutfile=fopen("putout.out","w");
if( putoutfile==NULL){
            printf("Can't write putout.out \n");
            exit(1);
}
else
            printf("Open putout.out \n");

/*.....
read from putin.dat in to put1 array
if it has no data (999) in the first, set default to it.
.....*/

fscanf(putinfile,"%d %c %f",&tempput1,&tempput2,&tempput3);
i=0;

if(tempput1!=999){
             printf("r1 -> %d\n", i);
            rewind(putinfile);
            while( tempput1 != 999){
                        fscanf(putinfile,"%d %c %f",&tempput1,&tempput2,&tempput3);
                        printf("%d %c %f\n",tempput1,tempput2,tempput3);
                        (*put1)[i].putcluster = tempput1;
                        (*put1)[i].wrongcluster = 0;
                        (*put1)[i].putpattern = tempput2;
                        (*put1)[i].putthres  = tempput3;
                        i++;
            }
      printf("r2 -> %d\n", i);
      i--;

}
else
{
            for (i=0; i< (MAXCLUSTER -1) ; i++){
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                (*put1)[i].putcluster=i;
                (*put1)[i].wrongcluster=0;
                (*put1)[i].putpattern='0';
                (*put1)[i].putthres=THRESHOLD;
        }

}

 printf("r3 -> %d\n", i);
 (*put1)[i].putcluster = 999;

 fgets(buf,BUFSIZE-1,infile);
 while( !feof(infile)){

        while( !feof(infile) && flag==0){
                switch(buf[0]){

                        case 'p':
                                count=0;
                                valueindex=0;
                                loop=1;
                                pattern[0]=buf[44];
                                pattern[1]=buf[45];
                                if((buf[46] == ' ')|| buf[46]=='\n') {
                                        pattern[2]='\0';
                                }
                                else{
                                        pattern[2]=buf[46];
                                }
                                pattern[3]='\0';
                                break;
                        case '0':
                        case '1':
                                i=0;

                                while(buf[i]!='\n'){
                                        (*s)[count]=buf[i]-'0';
                                        i++;
                                        count++;
                                }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                                        number=loop*COL-count;
                                        for(k=0;k<number;k++){
                                            (*s)[count]=0;
                                            count++;
                                        }

                                        loop++;
                                        break;

                                case 'C':
                                        flag=1;
                                        break;
                                }
                        fgets(buf,BUFSIZE-1,infile);

        } /*end second while */
/* append 0 from count until MAXNUMBER */
        count1=count;
        for(i=count;i<MAXNUMBER;i++){
                (*s)[count]=0;
                count++;
        }

/* calculate norms */
        norms=0;
        for(i=0;i<count;i++){
                norms=norms+(*s)[i];
        /* calculate y=s*b and find max of (*y)[i] */
        for (j=0;j<MAXCLUSTER;j++){
                (*y)[j] = 0;
                for( i=0; i< MAXNUMBER; i++){
                        temp=(*s)[i]*(*b)[i][j];
                        (*y)[j]=(*y)[j]+temp;
                }
        }

        mysort(y,MAXCLUSTER,posarr);

        maxindex=0;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
posindex=0;

do{
    /* Calculate normx=t*s */

        normx=0;
        maxindex=(*posarr)[posindex];
        for(i=0;i<count;i++){
                normx =    normx+(*s)[i]*(*t)[i][maxindex];
        }

        ratio= (float)normx/norms;
        value = (L*1)/((L-1)+normx);
        /*************************************************/
        /*              new threshold          */
        /*  read from diff.out in to newthreshold array.  */
        /*************************************************/
        Pflag = 0;
        threshold = (*put1)[maxindex].putthres;

        /**** first cluster *****/
        if ( (ratio == 1.0) && ((*put1)[maxindex].putpattern == '0')) {
                (*put1)[maxindex].putpattern = pattern[0];
                (*put1)[maxindex].putcluster = maxindex;
                (*put1)[maxindex].wrongcluster = 2;
                        /* change weight for bottomup and topdown */
                        for(i=0;i<MAXNUMBER;i++)
                                (*t)[i][maxindex]=(*s)[i]*(*t)[i][maxindex];

                        for(i=0;i<MAXNUMBER;i++){
                            value = (L*(*t)[i][maxindex])/((L-1)+normx);
                            ( *b)[i][maxindex]=value;
                        }
                break;
        }

        if ( ratio > threshold ){

                /***** same pattern ******/
            if( (*put1)[maxindex].putpattern == pattern[0] ){
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/* change weight for bottomup and topdown */
if( (*put1)[maxindex].wrongcluster ==0 || (*put1)[maxindex].wrongcluster==2){
        (*put1)[maxindex].wrongcluster =2;
 }
        for(i=0;i<MAXNUMBER;i++)
                (*t)[i][maxindex]=(*s)[i]*(*t)[i][maxindex];
        for(i=0;i<MAXNUMBER;i++){
           value = (L*(*t)[i][maxindex])/((L-1)+normx);
                        ( *b)[i][maxindex]=value;
        }
        break;
        }

        else{   /***** difference pattern ******/
                (*put1)[maxindex].wrongcluster = 1;
                (*put1)[maxindex].putthres = threshold + INCRESE;
                Pflag = 1;

        }
}
else{
                posindex++;
}

if (Pflag == 1)
        posindex++;

Pflag = 0;
/* can't accenpt in every weight */
if(maxindex == MAXCLUSTER -1){
        maxindex=-1;
        break;
}
}while(maxindex < MAXCLUSTER);
if(pattern[2]!='\0'){
        printf("%c%c%c %d\n",pattern[0],pattern[1],pattern[2],maxindex);
        fprintf(outfile,"%c%c%c %d\n",pattern[0],pattern[1],pattern[2], maxindex);
}else{
        printf("%c%c %d\n",pattern[0],pattern[1],maxindex);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                fprintf(outfile,"%c%c %d\n",pattern[0],pattern[1], maxindex);
            }
            flag=0;
        }  /* end first while */

        for(i=0;i<MAXCLUSTER;i++){
            for(j=0;j<MAXNUMBER;j++){
                fprintf(boutfile,"%f ",(*b)[j][i]);
                fprintf(toutfile,"%d ",(*t)[j][i]);
                if (j%10==9)
                  fprintf(boutfile, "\n");
            }
            fprintf(boutfile,"\n");
            fprintf(toutfile,"\n");
        }

        /*.....
            save array put1 in to putout.out file
        .....*/
 for( i=0;i < (MAXCLUSTER-1) ;i++){
    if ((*put1)[i].wrongcluster == 2){
       (*put1)[i].putthres = (*put1)[i].putthres - DECRESE;
    }
    fprintf(putoutfile,"%d %c %f\n",(*put1)[i].putcluster,(*put1)[i].putpattern,(*put1)[i].putthres);
    printf("%d %c %f\n",(*put1)[i].putcluster,(*put1)[i].putpattern,(*put1)[i].putthres);
        }
printf("i --> %d\n",i);


 fprintf(putoutfile,"999");
 printf("after savearray \n");
 printf("Before falut\n");

        fclose(infile);
        printf("Close INPUTFILE\n");
        fclose(outfile);
        printf("Close pj.out\n");
        fclose(boutfile);
        printf("Close tinit.out\n");
        fclose(toutfile);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            printf("Close binit.out\n");
            fclose(dinfile);
            printf("Close diff.out\n");
    close(putoutfile);
    printf("Close putout.out\n");
    free(b);
    free(t);
    free(y);
    free(max);
    free(s);
    free(posarr);
}

int ShellSort(f, Element, pos)
float *f;
int *pos;
int Element;
{
  float Max, tmp;
  int i, j, MaxIdx, SwpCount;

  for (i=0;i<Element;i++)
    pos[i] = i;

  SwpCount = 0;
  for (i=0;i<Element;i++) {
    Max = -9999.0;
    MaxIdx = 0;

    /* Search for the maximum */
    for (j=i;j<Element;j++) {
      if (f[j] > Max) {
            Max = f[j];
            MaxIdx = j;
      }
    } /* for j */

    /* Exchange value */
    if (MaxIdx != i) {
      tmp = f[MaxIdx];
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    f[MaxIdx] = f[i];
    f[i] = tmp;

    tmp = pos[i];
    pos[i] = pos[MaxIdx];
    pos[MaxIdx] = tmp;

    SwpCount++;
   }
  }

 return SwpCount;
}

binit()
{
FILE *binfile,*tinfile;
int i,j,tvalue;
float bvalue;
        binfile=fopen("binit.dat","r");
  tinfile=fopen("tinit.dat","r");

  if(binfile==NULL){
                printf("tinfile\n");
                exit(1);
        }
  if(tinfile==NULL){
    printf("binfile\n");
    exit(1);
  }
        for(i=0;i<MAXCLUSTER;i++){
                for(j=0;j<MAXNUMBER;j++){
                        fscanf(binfile,"%f",&bvalue);
      (*b)[j][i]=bvalue;
      fscanf(tinfile,"%d",&tvalue);
      (*t)[j][i]=tvalue;
                }
        }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    fclose(binfile);
    fclose(tinfile);
}

void *mymalloc(size_t size, char *s) {
  void *p;

  printf("Allocate %15s: %8u\t", s, size);

  p = malloc(size);
  if (p == NULL) {
              printf("(fail)\n", s);
              exit(1);
  } else {
             printf("(ok)\n", s);
             TotalAllocate += size;
  }

  return p;
}

FILE *myfopen(char *s, char *mode) {
  FILE *fp;

  fp = fopen(s, mode);
  if(fp == NULL){
                        fprintf(stderr,"Can't open file %s! (%s)\n", s, mode);
                        exit(1);
  }

  return fp;
}
```

2. โปรแกรม ART.H

โปรแกรมที่ใช้เก็บค่าตัวแปรต่างๆ

```
#define INPUT_FILE     "t1.x1"
#define THRESHOLD      0.55
#define MAXNUMBER      999  /*900 */
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
#define MAXCLUSTER     550  /*500*/
#define PATTERNNUMBER   200

#define OUTPUT_FILE    "pj.out"
#define BUFSIZE        80
#define NEWRATIO       0.99
#define REDUCE                  0.01
#define INCRESE                         0.03
#define DECRESE             0.01
#define L              2.0
#define COL            27
#define ROW            37
#define FIRSTLOOP                       1

#ifdef __TURBOC__
 #define MAXNUMBER    999
 #define MAXCLUSTER   5    /*50*/
#endif
```

3. โปรแกรม LAST75.C

โปรแกรมที่ใช้แยกแยะตัวอักษร

```
/*************************************************************
For recognition data
Programmer Penpun C.
*************************************************************/

#include <stdio.h>
#include <stdlib.h>
#ifdef __TURBOC__
#include <alloc.h>
#else
#include <malloc.h>
#endif

#include "last75.h"

struct tstruct
{
        int putcluster;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        char putpattern;
        float  putthres;
};
struct  tstruct (*put1)[900];

long  TotalAllocate = 0;
void *mymalloc(size_t size, char *s);
FILE *myfopen(char *s, char *mode);
float (*b)[MAXNUMBER][MAXCLUSTER];
int (*t)[MAXNUMBER][MAXCLUSTER];



main()
{
FILE *infile, *outfile;
FILE *boutfile, *toutfile, *dinfile;
FILE *putinfile, *putoutfile, *errfile;
FILE *finalfile;
char buf[BUFSIZE];
char pattern[5];
int (*s)[MAXNUMBER],(*posarr)[MAXCLUSTER];
float (*y)[MAXCLUSTER],(*max)[MAXCLUSTER];
int count,count1,i,j,norms,normx,maxindex,posindex,flag=0;
int loop = 1;
float temp, value = 0;
float *rj;
int number,k,valuex,valuey,valueindex=0;
float threshold;
int dflag=0;
float newratio;
float ratio;
int ccount=0;
float cpercent=0,tpercent=0;
int tempput1;
char tempput2;
float tempput3;
int Pflag = 0;
int lpercent=1;
        b = mymalloc(sizeof(*b), "b");
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
t = mymalloc(sizeof(*t), "t");
y = mymalloc(sizeof(*y), "y");
 max = mymalloc(sizeof(*max), "max");
s = mymalloc(sizeof(*s), "s");
put1 = mymalloc(sizeof(*put1),"put1");
posarr = mymalloc(sizeof(*posarr), "posarr");
printf("\nTotal allocate = %8u\n", TotalAllocate);
printf("Max cluster    = %8d\n", MAXCLUSTER);
printf("Max number     = %8d\n", MAXNUMBER);

binit();
infile=fopen(INPUT_FILE,"r");
if(infile == NULL){
        printf("Can't read input file!\n");
        exit(1);
}
else
        printf("Open INPUTFILE for read\n");

outfile=fopen(OUTPUT_FILE,"w");
if(     outfile== NULL){
        printf("Can't write output file\n");
        exit(1);
}
else
        printf("Open pj.out for write\n");

boutfile=fopen("binit.out","w");
if(     boutfile== NULL){
        printf("Can't write boutput file\n");
        exit(1);
}
else
        printf("Open binit.out for write\n");

toutfile=fopen("tinit.out","w");
if(     toutfile== NULL){
        printf("Can't write toutput file\n");
        exit(1);
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
else
        printf("Open tinit.out for write\n");

putinfile=fopen("putin.dat","r");
if( putinfile==NULL){
        printf("Can't read putin.dat \n");
        exit(1);
}
else
        printf("Open putin.dat for read\n");

putoutfile=fopen("putout.out","w");
if( putoutfile==NULL){
        printf("Can't write putout.out \n");
        exit(1);
}
else
        printf("Open putout.out \n");

errfile=fopen("errfile.out","w");
if( errfile==NULL){
        printf("Can't write errfile.out \n");
        exit(1);
}
else
        printf("Open errfile.out \n");

  finalfile=fopen("final.out","w");
  if( finalfile==NULL){
     printf("Can't write final.out \n");
        exit(1);
  }
  else
  printf("Open finalfile.out \n");

}

/*.....
read from putin.dat in to put1 array
if it has no data (999) in the first, set default to it.
.....*/
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
fscanf(putinfile,"%d %c %f",&tempput1,&tempput2,&tempput3);
i=0;

if(tempput1!=999){
        printf("r1 -> %d\n", i);
        rewind(putinfile);
        while( tempput1 != 999){
                fscanf(putinfile,"%d %c %f",&tempput1,&tempput2,&tempput3);
                (*put1)[i].putcluster = tempput1;
                 (*put1)[i].putpattern = tempput2;
                (*put1)[i].putthres  = tempput3;
                i++;
        }
        printf("r2 -> %d\n", i);
        i--;

}

else
{
        for (i=0; i<549; i++){
                (*put1)[i].putcluster=i;
                (*put1)[i].putpattern='0';
                (*put1)[i].putthres=THRESHOLD;
        }

}

printf("r3 -> %d\n", i);
(*put1)[i].putcluster = 999;

fgets(buf,BUFSIZE-1,infile);
while( !feof(infile)){

        while( !feof(infile) && flag==0){
                switch(buf[0]){

                        case 'p':
                                count=0;
                                valueindex=0;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                loop=1;
                pattern[0]=buf[44];
                pattern[1]=buf[45];
                if((buf[46] == ' ')|| buf[46]=='\n') {
                        pattern[2]='\0';
                }
                else{
                        pattern[2]=buf[46];
                }
                pattern[3]='\0';
                break;
                case '0':
                case '1':
                        i=0;

                while(buf[i]!='\n'){
                        (*s)[count]=buf[i]-'0';
                        i++;
                        count++;
                }

                 number=loop*COL-count;
                for(k=0;k<number;k++){
                (*s)[count]=0;
                count++;
                }

                loop++;
                        break;


                case 'C':
                flag=1;
                break;
        }
    fgets(buf,BUFSIZE-1,infile);

} /*end second while */
* append 0 from count until MAXNUMBER */
count1=count;
 for(i=count;i<MAXNUMBER;i++){
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            (*s)[count]=0;
            count++;
}


/* calculate norms */
norms=0;
for(i=0;i<count;i++)
            norms=norms+(*s)[i];

/* calculate y=s*b and find max of (*y)[i] */
for (j=0;j<MAXCLUSTER;j++){
            (*y)[j] = 0;
            for( i=0; i< MAXNUMBER; i++){
                        temp=(*s)[i]*(*b)[i][j];
                        (*y)[j]=(*y)[j]+temp;
            }
}

 mysort(y,MAXCLUSTER,posarr);


maxindex=0;
posindex=0;

do{
/* Calculate normx=t*s */
            normx=0;
            maxindex=(*posarr)[posindex];

            for(i=0;i<count;i++){
                        normx =    normx+(*s)[i]*(*t)[i][maxindex];
            }
            ratio= (float)normx/norms;
            /*                  new threshold           */
            /*  read from diff.out in to newthreshold array.  */
            /*************************************************/
            Pflag = 0;
            threshold = (*put1)[maxindex].putthres;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                if ( ratio > threshold ){
                                        break;
                }
                else{
                                        posindex++;
                }


                /* can't accenpt in every weight */
                if(maxindex == MAXCLUSTER -1){
                        maxindex=-1;
                        break;
                }
}while(maxindex < MAXCLUSTER);


if (pattern[0] == (*put1)[maxindex].putpattern){
        cpercent=cpercent+1.0;
}
ccount++;


if(pattern[2]!='\0'){
        if(pattern[0] != (*put1)[maxindex].putpattern)
        fprintf(errfile,"%c%c%c %d %c\n",pattern[0],pattern[1],pattern[2],
        maxindex,(*put1)[maxindex].putpattern);
}
else{
if(pattern[0] != (*put1)[maxindex].putpattern)
        fprintf(errfile,"%c%c %d %c\n",pattern[0],pattern[1],
        maxindex,(*put1)[maxindex].putpattern);
}
flag=0;
/*****************************************************
                        Check percent
*****************************************************/


        if(ccount == 200){
        fprintf(errfile,"Pattern %c  percent %f\n",pattern[0],cpercent/2.0);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                printf("Pattern %c  percent %f\n",pattern[0],cpercent/2.0);
                tpercent=tpercent+cpercent;
                lpercent++;
                printf("cpercent %f tpercent %f lpercent %d\n",cpercent,tpercent,lpercent);
                ccount = 0;
                cpercent = 0.0;
            }
        }  /* end first while */

        for(i=0;i<MAXCLUSTER;i++){
            for(j=0;j<MAXNUMBER;j++){
                fprintf(boutfile,"%f ",(*b)[j][i]);
                fprintf(toutfile,"%d ",(*t)[j][i]);
                if (j%10==9)
                 fprintf(boutfile, "\n");
            }
            fprintf(boutfile,"\n");
            fprintf(toutfile,"\n");
        }

        /*.....
        save array put1 in to putout.out file
        .....*/
    for( i=0;i < 549;i++){
      fprintf(putoutfile,"%d %c %f\n",(*put1)[i].putcluster,(*put1)[i].putpattern,(*put1)[i].putthres);
            }
            fprintf(putoutfile,"999");
    printf(errfile,"total percent = %f\n",tpercent/150.0);
    fprintf(errfile,"total percent = %f\n",tpercent/150.0);

free(b);
free(t);
free(y);
free(max);
free(s);
free(posarr);
fclose(infile);
printf("Close INPUTFILE\n");
fclose(outfile);
printf("Close pj.out\n");
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
fclose(boutfile);
printf("Close tinit.out\n");
fclose(toutfile);
printf("Close binit.out\n");
fclose(dinfile);
printf("Close diff.out\n");
}

int ShellSort(f, Element, pos)
float *f;
int *pos;
int Element;
{
 float Max, tmp;
 int i, j, MaxIdx, SwpCount;

 for (i=0;i<Element;i++)
  pos[i] = i;

 SwpCount = 0;
 for (i=0;i<Element;i++) {
  Max = -9999.0;
  MaxIdx = 0;

  /* Search for the maximum */
  for (j=i;j<Element;j++) {
   if (f[j] > Max) {
          Max = f[j];
          MaxIdx = j;
   }
  } /* for j */

  /* Exchange value */
  if (MaxIdx != i) {
   tmp = f[MaxIdx];
   f[MaxIdx] = f[i];
   f[i] = tmp;

   tmp = pos[i];
   pos[i] = pos[MaxIdx];
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
      pos[MaxIdx] = tmp;

      SwpCount++;
    }
  }

  return SwpCount;
}

binit()
{
FILE *binfile,*tinfile;
int i,j,tvalue;
float bvalue;
        binfile=fopen("binit.dat","r");
   tinfile=fopen("tinit.dat","r");

   if(binfile==NULL){
                printf("tinfile\n");
                exit(1);
        }
   if(tinfile==NULL){
      printf("binfile\n");
      exit(1);
   }
        for(i=0;i<MAXCLUSTER;i++){
                for(j=0;j<MAXNUMBER;j++){
                        fscanf(binfile,"%f",&bvalue);
        (*b)[j][i]=bvalue;
        fscanf(tinfile,"%d",&tvalue);
        (*t)[j][i]=tvalue;
                }
        }
        fclose(binfile);
   fclose(tinfile);
}

tinit()
{
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
void *mymalloc(size_t size, char *s) {
 void *p;

 printf("Allocate %15s: %8u\t", s, size);

 p = malloc(size);
 if (p == NULL) {
            printf("(fail)\n", s);
            exit(1);
 } else {
           printf("(ok)\n", s);
           TotalAllocate += size;
 }

 return p;
}

FILE *myfopen(char *s, char *mode) {
 FILE *fp;

 fp = fopen(s, mode);
 if(fp == NULL){
                    fprintf(stderr,"Can't open file %s! (%s)\n", s, mode);
                    exit(1);
 }

 return fp;
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประวัติผู้เขียน

ชื่อผู้เขียน	**นางสาวเพ็ญพรรณ ใช้ฮวดเจริญ**
วัน เดือน ปีเกิด	วันที่ 16 มกราคม 2511
สถานที่เกิด	จังหวัดชลบุรี
วุฒิการศึกษาระดับปริญญาตรี	วท.บ. สาขาวิชาสถิติประยุกต์
สถานที่สำเร็จการศึกษา	สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีที่สำเร็จการศึกษา	2532

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้