สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

การรู้จำอักขระตัวพิมพ์ภาษาไทยโดยวิธีโครงข่ายนิวรอน แบบแบคพรอบพาเกชั้น

THAI PRINTED CHARACTER RECOGNITION USING BACK PROPAGATION NEURAL NETWORK

นายทวี เปรมรัตนชัย

MR. THAWEE PREMRATANACHAI

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต

สาขาวิชาวิทยาการคอมพิวเตอร์และเทคโนโลยีสารสนเทศ

บัณฑิตวิทยาลัย

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ.2539

ISBN 974-621-594-9

ลิขสิทธิ์ของบัณฑิตวิทยาลัย สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เลขหมู่.....
เลขทะเบียน 27118
วัน, เดือน, ปี 11 ก.พ. 2540

THAI PRINTED CHARACTER RECOGNITION USING BACK PROPAGATION NEURAL NETWORK

THAWEE PREMRATANACHAI

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE
MASTER OF SCIENCE PROGRAM IN COMPUTER SCIENCE
AND INFORMATION TECHNOLOGY SCHOOL OF
GRADUATE STUDIES
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG
1996

ISBN 974-621-594-9

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อวิทยานิพนธ์	การรู้จำอักขระตัวพิมพ์ภาษาไทยด้วยวิธีโครงข่ายนิวรอนแบบแบคพรอบพาเกชั่น
นักศึกษา	นาย ทวี เปรมรัตนชัย
อาจารย์ผู้ควบคุมวิทยานิพนธ์	รศ.ดร. ชม กิ้มปาน
	ดร. บุญธีร์ เครือตราชู
ระดับการศึกษา	วิทยาศาสตร์มหาบัณฑิต สาขาวิชาวิทยาการคอมพิวเตอร์และเทคโนโลยีสารสนเทศ
ภาควิชา	คณิตศาสตร์และวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์
พ.ศ.	2538

บทคัดย่อ

วิทยานิพนธ์ฉบับนี้ทำการวิจัยเรื่องการรู้จำอักขระตัวพิมพ์ภาษาไทยโดยวิธีโครงข่ายนิวรอนแบบแบคพรอบพาเกชั่น (Thai Printed Character Recognition By Using Backpropagation Neural Network)โดยมีอินพุตคืออักขระภาษาไทยที่ได้จากการแสกนข้อความของหน้ากระดาษเอกสารซึ่งใช้อักขระภาษาไทยรูปแบบอังสานายูพีซี (Angsana Upc) 87 ตัวอักษร ขนาด 12 จุดต่อนิ้วในการฝึกสอนโครงข่ายนิวรอนให้สามารถรู้จำก่อนการใช้งานจริงและ เมื่อใช้งานจริงจะทำการแสกนหน้ากระดาษเอกสารด้วยอัตราส่วนResolution=300,Scale=100%,Brightnes s=100 แล้วจึงผ่านโปรแกรมแปลงภาพไฟล์กราฟฟิกไปเป็นไบนารีไฟล์ จากนั้นจึงผ่านโปรแกรมแยกอักขระออกมาที่ละตัวอักษรจากข้อความของหน้าเอกสารพร้อมบันทึกตำแหน่งของอักขระนั้นเพื่อใช้เป็นอินพุตให้แก่โครงข่ายนิวรอนในการรู้จำ และเป็นข้อมูลในการจัดลำดับอักขระกลับสู่รูปประโยคเดิมหลังการรู้จำแล้วเสร็จ โดยผลลัพธ์หลังการรู้จำและการจัดลำดับอักขระจะทำการจัดเก็บลงบนไฟล์ซึ่งสามารถนำไปใช้งานได้จริง การวิจัยนี้จะทำการจำลองโครงข่ายนิวรอนโดยใช้อัลกอลิทึมแบบแบคพรอบพาเกชั่นซึ่งโครงข่ายนิวรอนที่จำลองมีขนาด 4 เลเยอร์ คือ 1 อินพุตเลเยอร์ขนาด 1,600 โหนด , 2 ฮิดเด่นเลเยอร์ขนาด 230 โหนด , 1 เอาท์พุตเลเยอร์ขนาด 87 โหนด โดยทำการจำลองโครงข่ายนิวรอนบนเครื่องมินิคอมพิวเตอร์ รุ่น HP 9000/827 ซึ่งมีระบบปฏิบัติการ คือ ยูนิกซ์รุ่น HP-UX 9.04 ผ่านระบบเน็ตเวิร์กเพื่อใช้ในการฝึกสอนโครงข่ายนิวรอนและ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การรู้จำ ซึ่งผลการวิจัยพบว่าการฝึกสอนโครงข่ายนิวรอนให้รู้จำอักขระภาษาไทยต้นแบบ 87 ตัว ใช้เวลา 13 วัน และผลการรู้จำอักษรถูกต้องเฉลี่ย 98% จากอักขระที่ใช้ในการทดสอบทั้งหมด 60,000 ตัว และเปอร์เซนต์ในการรู้จำหน้ากระดาษทดสอบเมื่อทำการกำจัดสิ่งรบกวนจะมีความถูกต้องเฉลี่ย 98% โดยความเร็วของการรู้จำ 1 ตัวอักษรเฉลี่ยประมาณ 1.5 วินาที

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Thesis Title	Thai Printed Character Recognition By Using Back-Propagation Neural Network
Student	Mr Thawee Premratanachai
Thesis Advisor	Assoc.Prof.Dr Chom Kimpan
Thesis Co-advisor	Dr.Boontee Kruatrachu
Level of Study	Master of Science Program in Computer Scierce and Information Technology
Department	Mathematic and Computer Science, Faculty of Science
Year	1996

Abstract

This thesis is a study of Thai Printed Character Recognition by Using Bacprogpagation Neural Network. Thai Characters of page document scanned from 87 Angsana Upc with 12 point per inch are used as inputs. After train Neural Network recognizes with these characters, page document is scanned with the ratio of resolution=300, scale=100%, brightness=100%. Consequently file graphic is obtained and it is converted to be a binary file using the tif2b.exe program on microcomputer. When each character in the binary file is separated, its recorded its position in order to be the input for Neural Network to recognize and to be data for converting to the original position after Neural Network recognize. The recognition results and the characters position are saved on file. In the study, Neural Network is simulated by using Back-propagation Algorithm. There are four layers in the simulated Neural Network. One input layer is 1,600 nodes, two hidden layers are 230 nodes, one output layer is 87 nodes. HP 9000/827 Minicomputer with Unix HP-UX 9.04 is used for the simulation. From the study, it is found that it took 13 days for traing Neural Network to recognition these 87 Thai characters with the average of 98% when 60,000 tested characters are used. Without any noise, the percentage of the accuracy of the tested paper is 98 with the recognition speed of 1.5 seconds per character.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ขอขอบพระคุณ รองศาสตราจารย์ ดร.ชม กิ้มปาน ได้ให้ความอนุเคราะห์ ให้คำปรึกษาตลอดจนแนะนำการจัดทำวิทยานิพนธ์ฉบับนี้จนสำเร็จลุล่วงได้ดี

ขอขอบพระคุณ ดร.บุญธีร์ เครือตราชู ได้ให้ความรู้และคำปรึกษาตลอดจนข้อแนะนำในการทำวิจัยจนกระทั่งสำเร็จด้วยดี

ขอขอบพระคุณ มูลนิธิเพื่อการศึกษาคอมพิวเตอร์และการสื่อสารได้ให้ความอนุเคราะห์ให้อุดหนุนในการศึกษาและวิจัยฯ

ขอขอบพระคุณ คณาจารย์ เจ้าหน้าที่คณะวิทยาการคอมพิวเตอร์และเทคโนโลยีสารสนเทศที่ได้ให้ความช่วยเหลือ

สุดท้ายต้องขอขอบพระคุณ บิดา-มารดา ของข้าพเจ้าที่อบรมสั่งสอนข้าพเจ้าด้วยดีตลอดมา

ทวี เปรมรัตนชัย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

หน้า

บทคัดย่อภาษาไทย.....I

บทคัดย่อภาษาอังกฤษ.....III

กิตติกรรมประกาศ.....IV

สารบัญ.....V

สารบัญตาราง.....IX

สารบัญภาพ.....X

บทที่

1 บทนำ.....1

1.1 เนื้อหาโดยรวมของวิทยานิพนธ์.....1

1.2 การแปลงภาพข้อมูลจากเครื่องตรวจกวาดภาพเป็นภาพไบนารี.....3

1.3 ลักษณะของข้อมูล.....4

2 วัตถุประสงค์ ประวัติความเป็นมาและขั้นตอนการรู้จำตัวอักษร.....11

2.1 วัตถุประสงค์ของงานวิจัย.....12

2.2 ประวัติความเป็นมาของงานวิจัย.....13

2.3 ขั้นตอนการรู้จำตัวอักษร.....14

2.3.1 การแยกตัวอักษร.....14

2.3.2 การรู้จำตัวอักษร.....16

2.3.3 การเรียงรูปประโยคหลังการรู้จำ.....16

3 การแยกตัวอักษร.....18

3.1 ทฤษฎีที่นำมาใช้ในการแยกตัวอักษร.....19

3.2 ปัญหาและข้อจำกัดของทฤษฎีที่นำมาใช้.....22

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้า

บทที่

4 การรู้จำตัวอักษร.....30

4.1 ทฤษฎีพื้นฐานของโครงข่ายนิวรอน(Neural Network).....30

4.1.1 ลักษณะของโครงข่ายนิวรอน.....30

4.1.2 โครงข่ายนิวรอนในลักษณะ 1 ชั้น.....33

(Single Layer Neural Network)

4.1.3 โครงสร้างของนิวรอนในกรณีที่มีลักษณะหลายๆชั้น.....34

(Multilayer Neural Network)

4.2 การฝึกสอน(Training) ของโครงข่ายนิวรอน.....35

4.2.1 เพอเซปตรอน(Perceptron).....36

4.2.2 โครงข่ายนิวรอนแบบแบคพรอบพาเกชั่นที่ใช้ในวิทยานิพนธ์ฉบับนี้

(Backpropagation Neural Network Algorithm in this thesis)

4.2.2.1 ลักษณะโครงสร้างของโครงข่าย.....40

(Network Configuration)

5 ขั้นตอนการทำงานและการใช้ของโปรแกรมระบบ.....60

5.1 ขั้นตอนการทำงานของโปรแกรม.....60

5.2 ขั้นตอนการใช้งานโปรแกรมที่พัฒนาขึ้น.....61

5.2.1 โปรแกรมสำเร็จรูปสำหรับเก็บภาพ

จากเครื่องตรวจกวาดภาพ.....61

5.2.2 โปรแกรมที่พัฒนาขึ้นสำหรับการแปลง

ภาพหน้าเอกสารจากไฟล์กราฟฟิกไปเป็น

ไฟล์ข้อมูลไบนารีบนเครื่องไมโครคอมพิวเตอร์.....62

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้า

บทที่

5.2.3 โปรแกรมที่พัฒนาขึ้นสำหรับการแยกตัวอักขระและระบุตำแหน่งของข้อมูลในแต่ละบรรทัดจากภาพหน้าเอกสารบนเครื่องไมโครคอมพิวเตอร์.....64

5.2.4 โปรแกรมที่พัฒนาขึ้นสำหรับการรู้จำตัวอักษรโดยโครงข่ายนิวรอนและการจัดเรียงอักขระกลับสู่ตำแหน่งเดิมในประโยคข้อความหลังการรู้จำ.....65

6 การทดสอบและการประเมินผล.....66

6.1 การทดสอบและประเมินผลความถูกต้องในการจดจำตัวอักษรภาษาไทยชุดทดสอบของโครงข่ายนิวรอนแบบแบคพรอบพาเกชั่นและแบบเคาท์เตอร์พรอบพาเกชั่น.....66

6.2 การทดสอบและเปรียบเทียบความถูกต้องในส่วนของการนำเอาโครงข่ายนิวรอนที่ฝึกสอนโดยอัลกอลิทึมแบบแบคพรอบพาเกชั่นและ แบบเคาท์เตอร์พรอบพาเกชั่นไปประยุกต์งานจริงกับตัวอย่างข้อมูลหน้าเอกสาร.....76

7 บทสรุป.....80

7.1 ข้อสรุปและเปรียบเทียบของการรู้จำ 3 วิธีคือ การรู้จำโดยวิธีโครงข่ายนิวรอน, การรู้จำโดยวิธีแมตซิ่ง , การรู้จำโดยวิธีวิเคราะห์โครงสร้างทางกายภาพ.....80

7.2 ปัญหาและข้อจำกัดในการทำงาน.....85

7.1.1 ปัญหาและข้อจ

หน้าเอกสาร.....85

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน้า

บทที่

7.1.2 ปัญหาและข้อจำกัดของการทำงานในส่วนการรู้จำตัว
อักษรของโครงข่ายนิวรอนแบบแบคพรอบพาเกชั่น.....86
7.3 ข้อเสนอแนะสำหรับแนวทางการพัฒนาระบบการรู้จำของโครงข่าย
นิวรอนแบบแบคพรอบพาเกชั่นในอนาคต.....87

เอกสารอ้างอิง.....89
ภาคผนวก ก แฟ้มข้อมูลภาพกราฟฟิกรูปแบบ Tif File Version 5.4.....92
ภาคผนวก ข ตารางรหัสแทนอักขระ.....99
ภาคผนวก ค การใช้งานโปรแกรมเครื่องตรวจกวาดภาพ.....103
ภาคผนวก ง ตัวโปรแกรมการรู้จำอักษรคัดลายมือ.....107
ประวัติผู้เขียน.....203

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

หน้า

ตารางที่

5. การจดจำอักขระชุดทดสอบ 15,000 ตัว ด้วยโครงข่ายนิวรอนแบบแบคพรอบพาเกชั่น....67
6. การจดจำอักขระอินพุตชุดทดสอบ 15,000 ตัว ด้วยโครงข่ายนิวรอนแบบแบคพรอบพาเกชั่น....68
7. การจดจำอักขระโดยการเลื่อนภาพอักขระลง 1 แถว โดยวิธีแบคพรอบพาเกชั่นนิวรอน....71
8. การจดจำอักขระโดยการเลื่อนภาพอักขระไปทางขวา 1 แถว โดยวิธีแบคพรอบพาเกชั่นนิวรอน....72
9. การจดจำอักขระโดยการเลื่อนภาพอักขระไปทางขวาและลงล่างอย่างละ 1 แถว โดยวิธีแบคพรอบพาเกชั่นนิวรอน....72
10. การจดจำอักขระโดยการเลื่อนภาพอักขระไปทางขวา 2 แถว โดยวิธีแบคพรอบพาเกชั่นนิวรอน....73
11. การจดจำอักขระโดยการเลื่อนภาพอักขระลง 1 แถว โดยวิธีเคาท์เตอร์พรอบพาเกชั่นนิวรอน....74
12. การจดจำอักขระโดยการเลื่อนภาพอักขระไปทางขวา 1 แถว โดยวิธีเคาท์เตอร์พรอบพาเกชั่น....74
13. การจดจำอักขระโดยการเลื่อนภาพอักขระไปทางขวาและลงล่างอย่างละ 1 แถว โดยวิธีเคาท์เตอร์พรอบพาเกชั่นนิวรอน....75
14. การจดจำอักขระโดยการเลื่อนภาพอักขระไปทางขวาและลงล่างอย่างละ 2 แถว โดยวิธีเคาท์เตอร์พรอบพาเกชั่นนิวรอน....75

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ

หน้า

1-1 บล๊อกไดอะแกรมของขั้นตอนการทำงานของระบบการรู้จำอักขระภาษาไทยโดยโครงข่ายนิวรอนแบบแบคพรอบพาเกชั่น....2

1-2 รูปแสดงหลักการทำงานของเครื่องตรวจกวาดภาพ....3

1-3 รูปแสดงเครื่องตรวจกวาดที่ใช้ในงานวิจัยนี้....4

1-4 รูปแสดงตัวอย่างของภาพบิทแมพ แพทเทิร์น(Bitmap Pattern)....5

1-5 รูปแสดง Bitmap Pattern ในรูปของตัวเลข 0 กับ 1....5

1-6 รูปแสดง Bitmap Data ด้วยเลขฐาน 16....5

1-7 รูปแสดงโครงสร้าง Bitmap ความกว้างเป็นผลคูณของ 8....6

1-8 รูปแสดงโครงสร้าง Bitmap ความกว้างไม่เป็นผลคูณของ 8....7

1-9 รูปแสดงตัวอย่างภาพตัวอักษรในลักษณะเลขไบนารี....8

1-10 รูปตัวอย่างภาพหน้าเอกสารจากเครื่องตรวจกวาดภาพหน้าเอกสาร....9

1-11 รูปแสดงตัวอย่างของชุดตัวอักษรไทยรูปแบบ Angsana UPC ขนาด 12 dpi ที่ใช้ในงานวิจัย....10

2-1 รูปแสดงลักษณะโครงข่ายนิวรอนที่ใช้ในงานวิจัย....12

2-2 รูปแสดงองค์ประกอบของนิวรอนเซล....13

3-1 รูปแสดงทิศทางการเปลี่ยนรหัส "1" ที่ติดกันทั้ง 8 ทิศ....19

3-2 รูปแสดงขั้นตอนการเปลี่ยนเนื้อรหัส "1" ที่ติดกันของอักษรนั้น....20

3-3 รูปแสดงโฟลว์ชาร์ทการแยกอักขระ 1 ตัวออกจากข้อความที่เป็นบรรทัด....21

3-4 รูปแสดงภาพหน้ากระดาษเอกสารจากการตรวจกวาดภาพที่มีสิ่งรบกวนบนหน้ากระดาษ....22

3-5 รูปแสดงขั้นตอนการแยกภาพอักษรหนึ่งหน้ากระดาษออกเป็นบรรทัด....24

3-6 รูปแสดงข้อความในแต่ละบรรทัดหลังจากทำการแยกบรรทัดจากหน้าเอกสาร....25

3-7 ภาพแสดงไฟล์ซึ่งทำการแยกอักขระแต่ละตัวออกจากบรรทัดนั้น ๆ....27

3-8 รูปแสดงขั้นตอนการแยกภาพตัวอักษรออกจากภาพบรรทัด....28

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน้า

4-1 ลักษณะของนิวรอนเซล (neuron cell) ของประเซลประสาท....31
4-2 นิวรอน (neuron) ของโครงข่ายนิวรอน (neuron network)....31
4-3 สมการเชิงเส้นอย่างง่าย....32
4-4 สมการ threshold....32
4-5 ลักษณะของนิวรอน (neuron) 1 หน่วย (1 cell) ที่ประกอบด้วยฟังก์ชั่นการกระตุ้น (Actiuation Function)....33
4-6 ลักษณะของโครงข่ายนิวรอนในลักษณะ 1 ชั้น (Single Layer)....33
4-7 ลักษณะโครงข่ายนิวรอนในลักษณะ 2 ชั้น (Two-Layer Neural Network)....34
4-8 ลักษณะของเพอเซปตรอน (Perceptron)....36
4-9 เพอเซปตรอน (Perceptron) ที่ใช้แก้ปัญหา....36
4-10 กราฟความสัมพันธ์ X-Y....37
4-11 ตารางเอ็กซ์คลูซีฟออ....37
4-12 ลักษณะนิวรอนของโครงข่ายแบบแบคพรอบพาเกชั่น....40
4-13 กราฟของฟังก์ชั่นการกระตุ้น (Activation Function)....41
4-14 โครงข่ายนิวรอนแบบแบคพรอบพาเกชั่น 2 ฮิดเด่นเลเยอร์ที่ใช้ในงานวิจัยนี้ (OUT=87,H=200,I=1,600)....42
4-15 ตัวอย่างโครงข่ายนิวรอนขณะทำการป้อนอินพุตและค่าเป้าหมาย....48
5-1 ภาพแสดงการทำงานของระบบทั้งหมด....60
5-2 ภาพโปรแกรม Photo Styler....61
5-3 ภาพขณะรันโปรแกรม Tif2b.exe....62
5-4 ภาพแสดงไอคอนของ "FTP" บนวินโดวส์และภาพขณะทำการถ่ายโอนไฟล์โดยโปรแกรม FTP....63
5-5 ภาพขณะทำการถ่ายโอนไฟล์โดยโปรแกรม "FTP" บนวินโดวส์ฟอร์เวิร์กกรุ๊ป....63
5-6 ภาพแสดงขั้นตอนการเข้าระบบการทำงานของเครื่องไดมอนด์โดยโปรแกรมเทลเน็ต....64

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

<u>เนื้อหาโดยรวมของวิทยานิพนธ์</u>

ในปัจจุบันนี้คอมพิวเตอร์เข้ามามีบทบาทต่อชีวิตมนุษย์เป็นอย่างมาก สามารถช่วยมนุษย์เราในการทำงานที่ยุ่งยากซับซ้อนให้เสร็จภายในเวลาอันสั้น ทำให้มนุษย์เรามีความสะดวกสบายยิ่งขึ้นกว่าแต่ก่อน แต่ความสามารถของเครื่องคอมพิวเตอร์ในปัจจุบันนี้ ไม่สามารถจะนำไปใช้งานได้ดีในงานที่มีลักษณะอาศัยความฉลาดของมนุษย์ในการตัดสินใจ ซึ่งมีความไม่แน่นอนในการแก้ปัญหาเข้ามาเกี่ยวข้องด้วย รวมทั้งความเร็วในการแก้ปัญหาของเครื่องคอมพิวเตอร์ต่อปัญหาในลักษณะเช่นนี้ จะใช้เวลาในการแก้ปัญหานานเป็นอันมาก

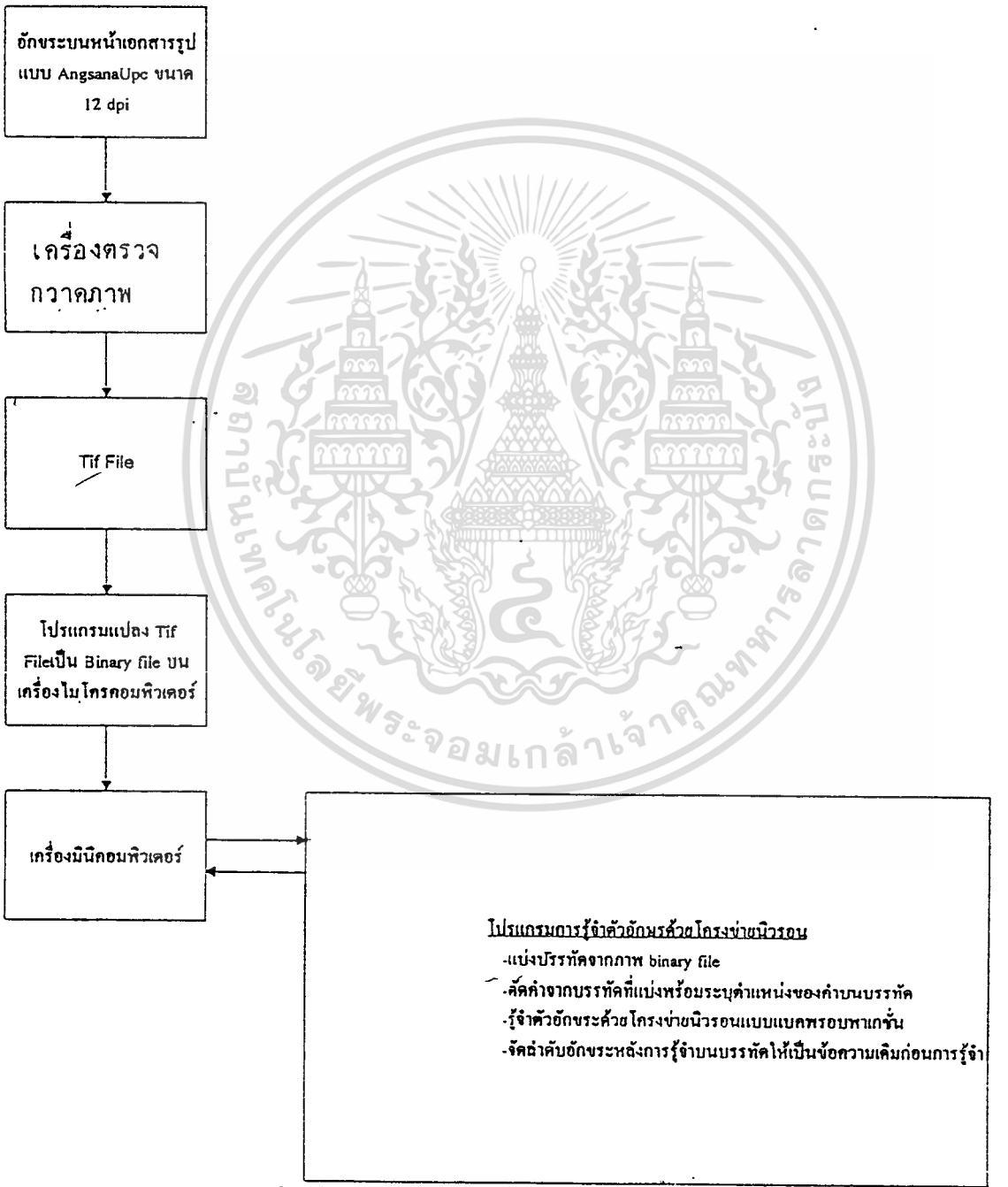
เทคโนโลยีทางด้านโครงข่ายนิวรอน (Neural Network) สามารถช่วยเราในการแก้ปัญหาดังกล่าวได้ โครงข่ายนิวรอนเป็นเทคโนโลยีหนึ่งทางด้านปัญญาประดิษฐ์ (Artificial Intelligence) ที่อาศัยรูปแบบทางโครงสร้างและการทำงานของสมองมนุษย์ มาทำการสร้างเครื่องคอมพิวเตอร์ที่มีความฉลาด มีความรู้อยู่ภายในตัวมันเอง มีความสามารถในการเรียนรู้ต่อความรู้ที่เข้ามาใหม่ มีความสามารถในการแก้ปัญหาที่มีลักษณะไม่แน่นอน และมีความเร็วในการแก้ปัญหาสูง

สำหรับงานวิจัยที่ศึกษา คือ การนำโครงข่ายนิวรอนแบบแบคพรอบพาเกชั่น (Back-propagation Neural Network) มาใช้ในการรู้จำอักขระตัวพิมพ์ภาษาไทย โครงข่ายนิวรอนแบบนี้เป็นรูปแบบหนึ่งของโครงข่ายนิวรอน (Neural Network)ที่นิยมใช้กันอย่างแพร่หลายและประสบความสำเร็จเป็นอย่างมากในทางปฏิบัติ โครงข่ายนิวรอน (Neural Network) รูปแบบนี้ มีความสามารถในการจดจำ (recognize) รูปแบบในการแก้ปัญหาโดยทำการฝึกสอน (training) โครงข่ายด้วยอินพุตชุดฝึกสอน ซึ่งเป็นอักขระที่ได้จากเครื่องตรวจกวาดภาพ (Image scanner) โดยอักขระบนหน้าเอกสารที่ใช้ในการฝึกสอนเป็นอักขระรูปแบบ AngsanaUPC , ขนาด 12 dpi และทำการแสกนหน้าเอกสารด้วยอัตราส่วนการแสกนคือ 300 dpi ความสว่าง 100% ซึ่งอักขระชุดฝึกสอนมีจำนวน 87x20 = 1,740 ตัวอักษร ตามจำนวนอักษรไทยและสระ(67) , เลขไทย (10) และเลขอารบิก (10) เมื่อได้อักขระชุดฝึกสอนหลังจากแยกตัวอักษรจากหน้าเอกสารแล้วเราจึงป้อนให้แก่ตัวโครงข่ายนิวรอนซึ่งทำการจำลองด้วยการเขียนโปรแกรมบนเครื่องมินิคอมพิวเตอร์ด้วยภาษา C. เมื่อตัวโครงข่ายนิวรอนทำการฝึกสอนด้วยชุดฝึกสอนแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะมีความสามารถในการแก้ปัญหาต่อรูปแบบของปัญหาที่เหมือนหรือใกล้เคียงกับรูปแบบของปัญหาที่ทำการฝึกสอนได้

สำหรับงานวิจัยนี้จะเน้นถึงการจำลองโครงข่ายนิวรอนให้รู้จำอักขระภาษาไทย , เลขไทย และเลขอารบิก โดยขั้นตอนการทำงานของระบบการรู้จำอักขระภาษาไทยนั้น สามารถเขียนได้ดังแผนผังการทำงานดังนี้



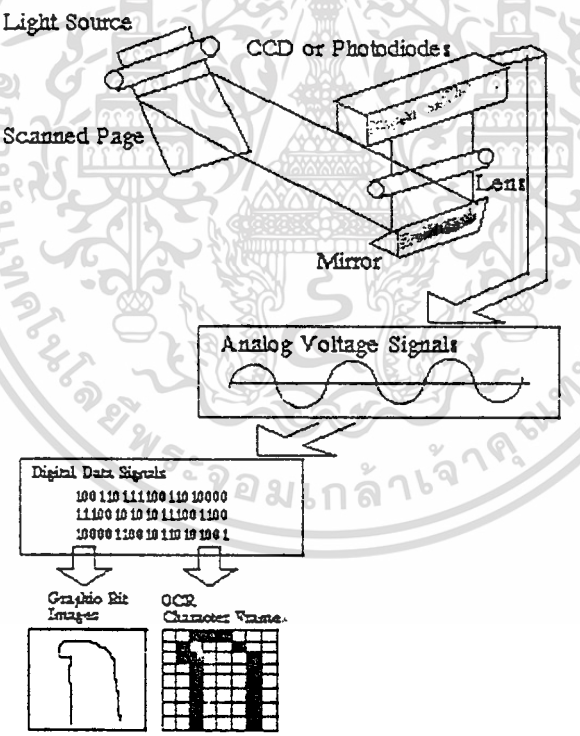
รูปที่ 1-1 บล๊อกไดอะแกรมของขั้นตอนการทำงานของระบบการรู้จำอักขระภาษาไทยโดยโครงข่ายนิวรอนแบบแบคพรอบพาเกชั่น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. <u>การแปลงภาพข้อมูลจากเครื่องตรวจกวาดภาพเป็นภาพ</u>

<u>เลขไบนารี (biniary data : 0 หรือ 1)</u>

ความละเอียด (Resolution) ของเครื่องตรวจกวาดภาพจะวัดกันด้วยค่าจุดต่อนิ้ว (Dot per inch, DPI) ซึ่งเป็นค่าตัวเลขแน่นอนค่าหนึ่งที่ได้จากความสัมพันธ์ของจำนวนเซลในหนึ่งอาร์เรย์และพื้นที่ทั้งหมดที่ตรวจกวาด ยกตัวอย่างเช่น อาร์เรย์ประกอบด้วย เซลรับแสงจำนวน 2,400 หน่วย สามารถครอบคลุมพื้นที่ในแนวตรงได้ 8 นิ้ว จะได้ค่าความละเอียดของเครื่องตรวจกวาดภาพ (Hardware resolution) เป็น 300 dpi ซึ่งเมื่อนำมาใช้ตรวจกวาดภาพ ก็จะได้ภาพที่มีความละเอียดของภาพ (Image Resolution) ขนาด 300 dpi เครื่องตรวจกวาดภาพทั่ว ๆ ไป ยังสามารถตรวจกวาดภาพและให้ความละเอียดที่ต่ำกว่าหรือสูงกว่าได้ โดยอาศัยซอฟต์แวร์ช่วยในการจัดการกำหนดความละเอียดของภาพ การทำงานของเครื่องตรวจกวาดภาพสามารถแสดงได้ดังรูป

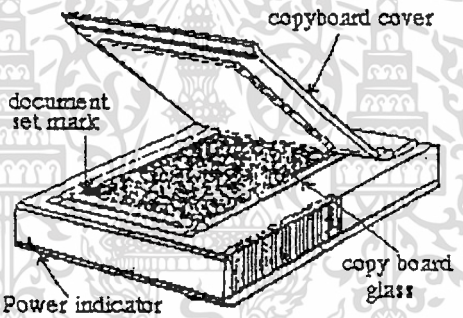


รูปที่ 1-2 รูปแสดงหลักการทำงานของเครื่องตรวจกวาดภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.1 เครื่องตรวจกวาดภาพที่ใช้ในการวิจัย

เครื่องตรวจกวาดภาพที่ใช้ในงานวิจัยนี้เป็นเครื่องตรวจกวาดภาพชนิดตั้งโต๊ะผลิตโดยบริษัท Hewlett Packard รุ่น Scan Jet สามารถตรวจกวาดภาพเอกสารที่มีขนาดใหญ่ได้ถึงขนาดกระดาษ A4 และสามารถเลือกระดับความละเอียดของภาพ (Scaning -resolution) ได้ตั้งแต่ 38 ถึง 600 จุดต่อนิ้ว (Dots per inch) สามารถปรับค่าความละเอียดของภาพได้โดยอาศัยซอฟแวร์ รูปภาพหรือข้อความที่ตรวจกวาดได้จะถูกจัดเก็บลงแฟ้มข้อมูลกราฟฟิกในฟอร์แมต TIFF ข้อมูลที่จัดเก็บมีลักษณะเป็น Packed binary data (มีค่าเป็นตัวเลขค่า 0 กับ 1) โดยที่ข้อมูลที่มีค่าเป็น 1 จะแทนส่วนที่เป็นพื้นดำที่เป็นข้อความหรือรูปภาพ และข้อมูลที่มีค่าเป็น 0 จะแทนส่วนที่เป็นพื้นสว่างที่เป็นช่องว่างบนแผ่นกระดาษ



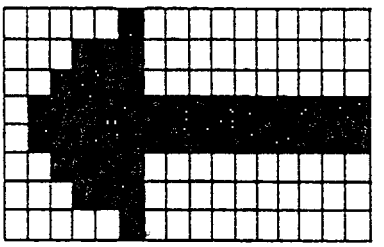
รูปที่ 1-3 รูปแสดงเครื่องตรวจกวาดภาพที่ใช้ในงานวิจัยนี้

2. ลักษณะของข้อมูล

2.1 ลักษณะของข้อมูลภาพกราฟฟิก

เนื่องจากข้อมูลที่จัดเก็บได้จากเครื่องตรวจกวาดภาพจะอยู่ในแฟ้มข้อมูลภาพกราฟ-ฟิก นั่นคือเป็นตัวเลขค่าไบนารี่คือ เลข 0 กับ 1 ของจุดภาพหนึ่ง ๆ ถูกย่อรวมกันเป็นจุดเก็บอยู่ในรูปตัวเลขฐาน 16 ก่อนที่จะนำข้อมูลดังกล่าวมาป้อนให้กับเครื่องคอมพิวเตอร์เพื่อทำการประมวลผลต่อไป จะต้องแปลงข้อมูลกราฟฟิกในรูปเลขฐาน 16 นี้กลับให้เป็นข้อมูลไบนารี่ในรูปตัวเลข 0 กับ 1 เสียก่อน แฟ้มข้อมูลกราฟฟิก ที่ได้จากเครื่องตรวจกวาดภาพที่ใช้ในงานวิจัยนี้เป็นแฟ้มข้อมูลรูปแบบ TIFF มีลักษณะเป็นบิทแม็พ (Bitmap pattern) ซึ่งก็คือกลุ่มของจุดสี่เหลี่ยมอาจจะเป็นสีขาวหรือสีดำมาเรียงต่อกันเป็นตารางซึ่งจะต่อรวมกันเป็นแสดงออกมาเป็นรูปภาพตัวอย่างของรูปภาพแบบบิตแม็พ แสดงได้ดังรูปที่ 1-4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 1-4 แสดงตัวอย่างของภาพ Bitmap Pattern

Bitmap pattern ที่แสดงดังรูปบนสามารถที่จะแสดงให้อยู่ในรูปของตัวเลข 0 กับ 1 ได้ดังรูปที่1-5 และBitmap data ซึ่งเป็นข้อมูลที่เก็บบันทึกจริงในแฟ้มข้อมูลกราฟฟิกจะมีลักษณะเป็นตัวเลขฐาน 16 ดังแสดงในรูปที่1-6

0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0
0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0
0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0
0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0

รูปที่ 1-5 รูปแสดง Bitmap pattern ในรูปของตัวเลข 0 กับ 1

0x20(byte0)	0x00(byte1)
0x38(byte2)	0x00(byte3)
0x3c(byte4)	0x00(byte5)
0xfe(byte6)	0xff(byte7)
0xfe(byte8)	0xff(byte9)
0x3c(byte10)	0x00(byte11)
0x38(byte12)	0x00(byte13)
0x20(byte14)	0x00(byte15)

รูปที่ 1-6 รูปแสดง Bitmap data ด้วยเลขฐาน 16

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2 การเปลี่ยนข้อมูลบิตแม็พเป็นข้อมูลไบนารี่

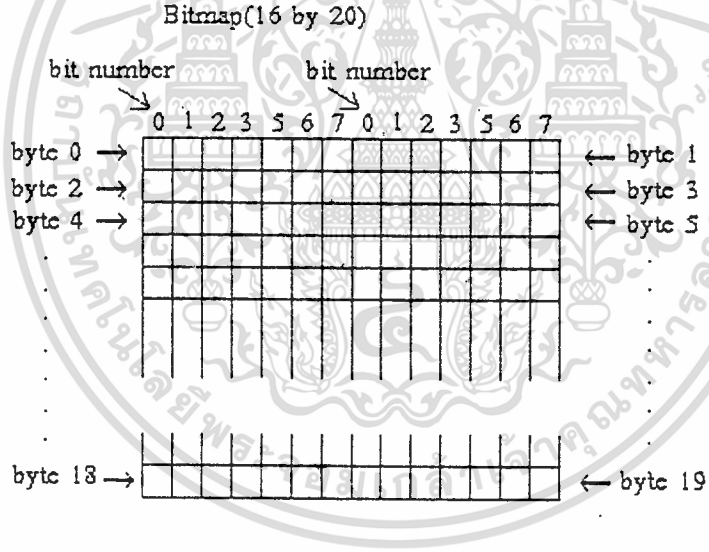
ข้อมูลภาพกราฟฟิกแบบ Bitmap มีอยู่ 2 ลักษณะ โดยพิจารณาจากความกว้างของภาพ และนำมาใช้เป็นหลักเกณฑ์ในการเปลี่ยนข้อมูลที่อยู่ในลักษณะเลขฐาน 16 เป็นเลขฐาน 2 คือ

1. ถ้าความกว้างเป็นผลคูณของ 8 มีเงื่อนไขของโครงสร้างดังนี้

   (ก) ข้อมูลในรูปของตัวเลข 0 หรือ 1 จะถูกจับเป็นกลุ่มโดยเรียงกันไปจำนวน 8 บิท นั่นก็คือ 1 ไบท์

   (ข) ลำดับของไบท์ของข้อมูลจะนับจากบนลงล่าง และซ้ายไปขวา

   (ค) ลำดับของบิทในแต่ละไบท์จะนับจากซ้ายไปขวา และบิททางซ้ายมือสุด (leftmost bit) นับเป็นบิทที่ 0 และบิททางขวามือสุด (Rightmost bit) นับเป็นบิทที่ 7



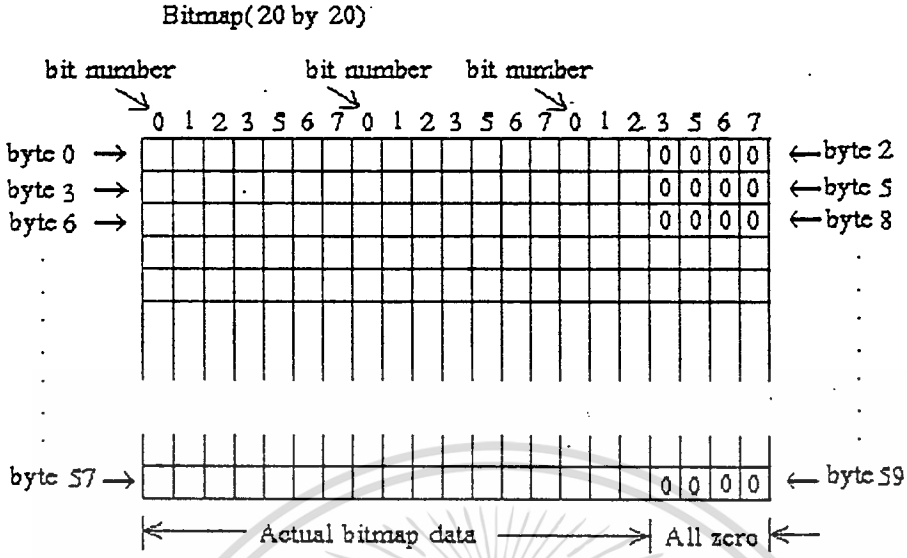
รูปที่ 1-7 รูปแสดงโครงสร้าง Bitmap ความกว้างเป็นผลคูณของ 8

2. ถ้าความกว้างไม่เป็นผลคูณของ 8 จะมีเงื่อนไขดังนี้

   (ก) จะนับความกว้างเป็นผลคูณของ 8 ที่น้อยที่สุดที่มีค่ามากกว่าความกว้างที่แท้จริงของรูป

   (ข) บิทที่เหลืออยู่นอกเหนือจากความกว้างที่แท้จริง จะถูกเติมด้วยตัวเลข 0

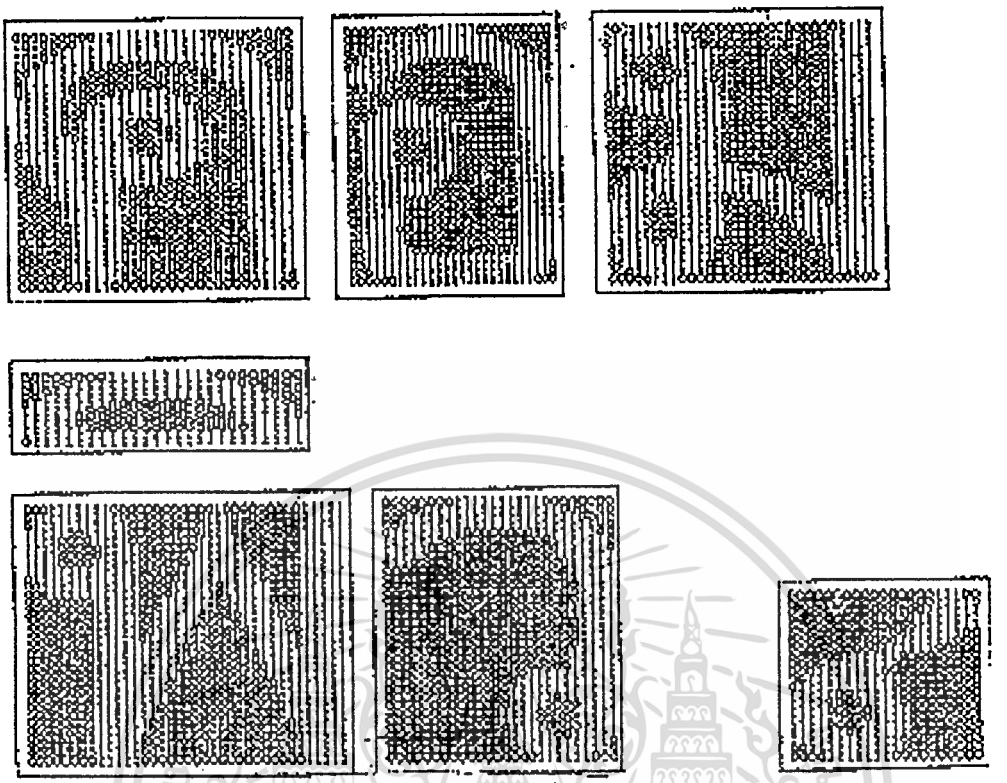
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 1-8 รูปแสดงโครงสร้าง Bitmap ความกว้างไม่เป็นผลคูณของ 8

เมื่อทราบรูปแบบของ Bitmap แล้ว ก็ทำการเปลี่ยนข้อมูลภาพที่มีลักษณะเป็นเลขฐาน 16 ให้เป็นเลขฐาน 2 โดยหนึ่งบิตของเลขฐาน 2 จะแทนจุดภาพหนึ่งจุด บิตที่มีค่าเป็น 0 ก็จะแทนจุดสว่างหรือจุดที่เป็นพื้นกระดาษสีขาว และบิตที่มีค่าเป็น 1 ก็จะแทนจุดดำมืดทึบ ซึ่งก็คือจุดที่อาจเป็นตัวอักษรหรือจุดดำของรูปภาพ ก็จะได้ข้อมูลภาพตัวอักษรที่ถูกต้องและพร้อมที่จะส่งไปประมวลผลในขั้นตอนต่อไป ตัวอย่างของภาพตัวอักษรที่ถูกเปลี่ยนจุดภาพให้เป็นค่าไบนารี่ แสดงได้ดังรูปที่ 1-9

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

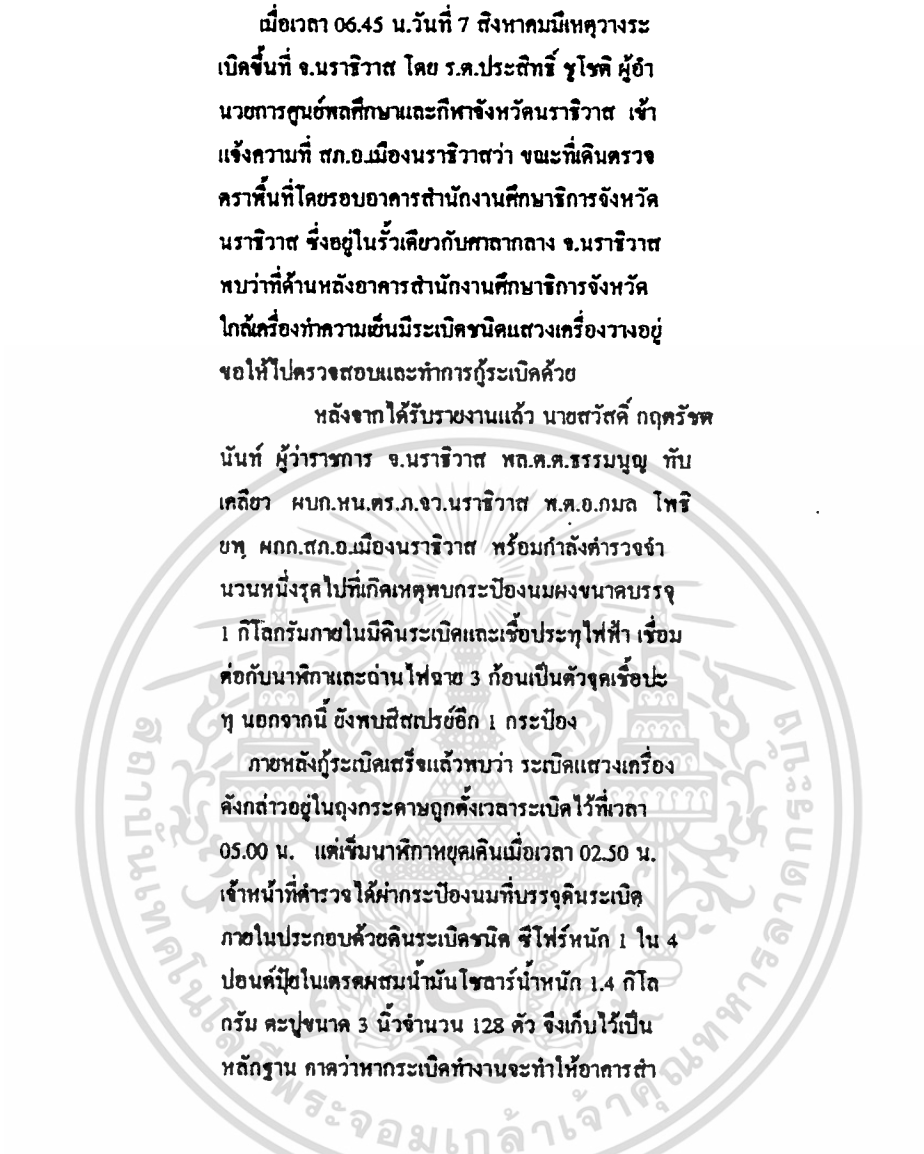


รูปที่ 1-9 แสดงตัวอย่างภาพตัวอักษรในลักษณะเลขไบนารี่

2.3 ตัวอย่างข้อมูลที่ใช้ในงานวิจัย

ข้อมูลที่ใช้ในงานวิจัยนี้จะมีลักษณะเป็นภาพหน้าเอกสารหนึ่งหน้าขนาดไม่เกินขนาดกระดาษ A4 บนหน้ากระดาษจะประกอบด้วย ข้อความที่มีตัวอักษรภาษาไทย , วรรณยุกต์ไทย และเลขอารบิก ซึ่งข้อความที่ใช้ในงานวิจัยจะใช้ข้อความที่มีรูปแบบคือ Font AngsanaUPC ขนาด 12 dpi แบบเดียวเท่านั้น โดยภาพรูปที่ 1-10 จะแสดงตัวอย่างของภาพหน้าเอกสารในรูปแบบภาพกราฟฟิกซึ่งได้จากการจัดเก็บจากเครื่องตรวจกวาดภาพด้วยอัตราส่วนการการแสกนหน้าเอกสารคือ (Resolution = 300 dpi , Brightness = 100% , Seale = 100%)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


เมื่อเวลา 06.45 น.วันที่ 7 สิงหาคมมีเหตุวางระ
เบิดขึ้นที่ จ.นราธิวาส โดย ร.ต.ประสิทธิ์ ชูโชติ ผู้อำ
นวยการศูนย์พลศึกษาและกีฬาจังหวัดนราธิวาส เข้า
แจ้งความที่ สภ.อ.เมืองนราธิวาสว่า ขณะที่คนตรวจ
ตราพื้นที่โดยรอบอาคารสำนักงานศึกษาธิการจังหวัด
นราธิวาส ซึ่งอยู่ในรั้วเดียวกับศาลากลาง จ.นราธิวาส
พบว่าที่ด้านหลังอาคารสำนักงานศึกษาธิการจังหวัด
ใกล้เครื่องทำความเย็นมีระเบิดชนิดแสวงเครื่องวางอยู่
ขอให้ไปตรวจสอบและทำการกู้ระเบิดด้วย

หลังจากได้รับรายงานแล้ว นายสวัสดิ์ กฤตรัชต
นันท์ ผู้ว่าราชการ จ.นราธิวาส พล.ต.ต.ธรรมนูญ ทับ
เคลียว ผบก.ทน.ตร.ภ.จว.นราธิวาส พ.ต.อ.ภมล โพธิ
ขพุ ผกก.สภ.อ.เมืองนราธิวาส พร้อมกำลังตำรวจจำ
นวนหนึ่งรุดไปที่เกิดเหตุพบกระป๋องนมผงขนาดบรรจุ
1 กิโลกรัมภายในมีดินระเบิดและเชื้อประทุไฟฟ้า เชื่อม
ต่อกับนาฬิกาและถ่านไฟฉาย 3 ก้อนเป็นตัวจุดเชื้อประ
ทุ นอกจากนี้ ยังพบสิ่งประหลาดอีก 1 กระป๋อง

ภายหลังกู้ระเบิดเสร็จแล้วพบว่า ระเบิดแสวงเครื่อง
ดังกล่าวอยู่ในถุงกระดาษถูกตั้งเวลาระเบิดไว้ที่เวลา
05.00 น. แต่เจ้าหน้าที่หยุดเดินเมื่อเวลา 02.50 น.
เจ้าหน้าที่ตำรวจได้ผ่ากระป๋องนมที่บรรจุดินระเบิด
ภายในประกอบด้วยดินระเบิดชนิด ซีโฟร์หนัก 1 ใน 4
ปอนด์ปนในเตรดผสมน้ำมันโซดาร์น้ำหนัก 1.4 กิโล
กรัม ตะปูขนาด 3 นิ้วจำนวน 128 ตัว จึงเก็บไว้เป็น
หลักฐาน คาดว่าหากระเบิดทำงานจะทำให้อาคารสำ

รูปที่ 1-10 รูปตัวอย่างภาพหน้าเอกสารจากเครื่องตรวจกวาดภาพหน้าเอกสาร

ตัวอย่างภาพตัวอักษรที่ใช้ในการรู้จำ ประกอบด้วยตัวอักษรภาษาไทย 44 ตัว สระและวรรณยุกต์ไทย จำนวน 19 ตัว เลขไทย 10 ตัว และตัวเลขอารบิก 10 ตัว รูปแบบ (Font) ของตัวอักษรที่ใช้คือ 1 รูปแบบ คือ Angsana UPC ขนาด 12 dpi ได้แสดงดังรูปที่ 1-11

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ก ข ฃ ค ฅ ฆ ง จ ฉ ช ซ ฌ ญ ฎ ฏ ฐ ฑ ฒ ณ ด ต ถ

ท ธ น บ ป ผ ฝ พ ฟ ภ ม ย ร ล ว ศ ษ ส ห ฬ อ ฮ

0 1 2 3 4 5 6 7 8 9

๐ ๑ ๒ ๓ ๔ ๕ ๖ ๗ ๘ ๙

ฯ ั า ิ ี ึ ื ุ ู เ แ โ ใ ไ ๅ ็ ่ ้ ๊ ๋ ์ ๎ ๏

รูปที่ 1-11 รูปแสดงตัวอย่างของชุดตัวอักษรไทย Angsana UPC ขนาด 12 dpi ที่ใช้ในงานวิจัย

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

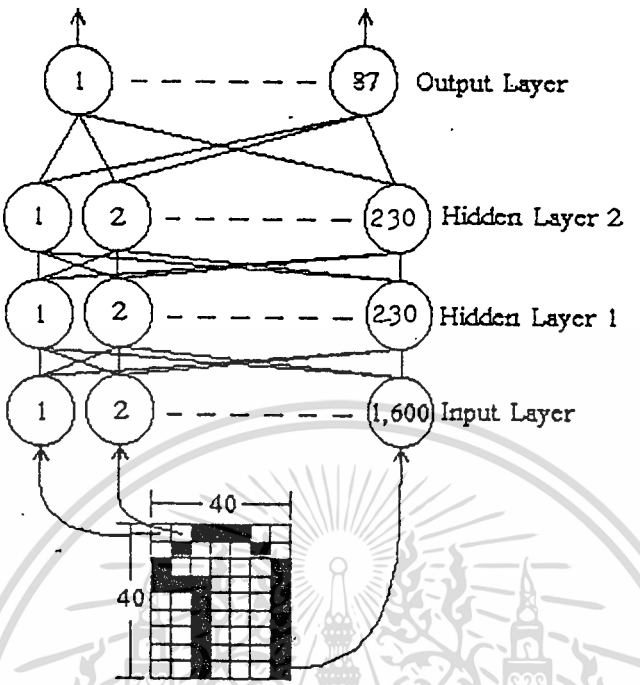
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

วัตถุประสงค์ ประวัติความเป็นมาและขั้นตอนการรู้จำตัวอักษร

จากความพยายามในการสอนหรือฝึกสอนให้คอมพิวเตอร์มีความสามารถหรือมีความรู้เพื่อใช้ในการพิจารณาแยกแยะอักษรต่าง ๆ ได้อย่างถูกต้องนั้น ได้มีการศึกษาและทำการวิจัยในหลาย ๆ วิธี โดยได้มีการเรียนรู้และสร้างทฤษฎีของการรู้จำตัวอักษรกันอย่างมากมาย ซึ่งในแต่ละวิธีก็จะมีความเหมาะสม และใช้แก้ปัญหาที่เกิดขึ้น ดังรายงานการวิจัยทั้งในและต่างประเทศ ซึ่งสามารถแบ่งวิธีการรู้จำได้โดยการพิจารณาลักษณะอักษรในรูปแบบที่แตกต่างกัน เช่น การรู้จำอักษรตัวพิมพ์และการรู้จำอักษรลายมือเขียน หรืออาจแบ่งโดยการพิจารณาจากวิธีในการรู้จำ เช่นการรู้จำตัวอักษรโดยวิธีการซ้อนทับกัน (Matching) การรู้จำตัวอักษรโดยวิธีการพิจารณาโครงสร้างทางกายภาพของตัวอักษร ซึ่งแต่ละวิธีก็มีขีดความสามารถของการรู้จำที่แตกต่างกันไป และใช้แก้ปัญหาได้แตกต่างกันออกไป ส่วนของงานวิจัยในวิทยานิพนธ์ฉบับนี้ คือการรู้จำอักษรไทยโดยโครงข่ายนิวรอนแบบ แบคพรอบพาเกชั่นโดยทำการจำลองโครงข่ายนิวรอน 4 ชั้น ด้วยกัน คือ ชั้นฮิดเดนจะใช้ 2 ชั้น ด้วยจำนวนโหนดเท่ากับ 230โหนด ในแต่ละชั้น และชั้นอินพุตจะใช้จำนวนโหนดเท่ากับ 1,600 โหนด ตามขนาดของอินพุต ซึ่งมีขนาดความกว้างxยาว เท่ากับ 40x40 และชั้นเอาท์พุตจะมีจำนวนโหนด 87 โหนด ตามจำนวนตัวอักษรเป้าหมายที่ต้องการรู้จำ ซึ่งโครงข่ายนิวรอนรูปแบบนี้มีความสามารถในการจดจำรูปแบบของตัวอักษรที่คล้าย ๆ กันได้เป็นอย่างดี อีกทั้งมีความสามารถจดจำอักขระที่มีรูปร่างที่เลอะเลือนจากอักขระต้นแบบได้ค่อนข้างดี ในส่วนของการฝึกสอนโครงข่ายให้รู้จำอักษรต้นแบบนั้น จะทำการป้อนอักขระต้นแบบให้โครงข่ายจดจำจนกว่าค่าผิดพลาดโดยรวมซึ่งเกิดจากการรู้จำอักขระต้นแบบทั้งหมดมีค่าต่ำกว่าค่าผิดพลาดเป้าหมายที่ตั้งไว้ จึงจะหยุดการฝึกสอนโครงข่าย และนำเวต(weight) ของโครงข่ายนิวรอนซึ่งเก็บความรู้ที่ได้ฝึกสอนไว้ ไปใช้งานการรู้จำได้ต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2-1 รูปลักษณะโครงข่ายนิวรอนที่ใช้ในงานวิจัย

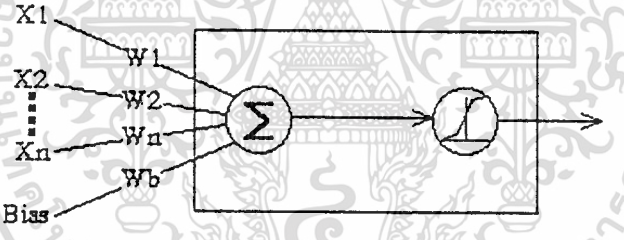
2.1 วัตถุประสงค์ของงานวิจัย

วิทยานิพนธ์ฉบับนี้มีวัตถุประสงค์เพื่อทำความเข้าใจเกี่ยวกับโครงข่ายนิวรอนแบบ แบคพรอบพาเกชั่นว่ามีการทำงานแบบใด และทำการโปรแกรมจำลอง (simulated) การทำงานของโครงข่ายนิวรอนแบบแบคพรอบพาเกชั่น (Backpropagation Neural Network) บนเครื่องมินิคอมพิวเตอร์ โดยอาศัยระบบเน็ตเวิร์คของสถาบัน เป็นตัวเชื่อมโยงในการเข้าไปปฏิบัติบนเครื่องมินิคอมพิวเตอร์ที่ชื่อ diamond อุปกรณ์ที่ต้องใช้อีกชนิดหนึ่งคือเครื่องตรวจกวาดภาพ (Image scanner) เพื่อแปลงภาพหน้ากระดาษเอกสารไปเป็นภาพกราฟฟิก (Tif file format) จากนั้นจึงทำการแปลงภาพไฟล์กราฟฟิก (Tif file) ไปเป็นภาพไบนารี่ (0 , 1) (โดยแทนรหัส 0 ด้วยพื้นกระดาษขาวของหน้าเอกสารและแทนรหัส 1 ด้วยเนื้อตัวอักษรของหน้าเอกสาร) ซึ่งการแปลงภาพเป็นรหัส 0, 1 ทำโดยโปรแกรมที่สร้างขึ้นบนเครื่องไมโครคอมพิวเตอร์ แล้วจึงนำภาพไบนารี่ (0 , 1) ที่ได้มาทำการแยกออกทีละบรรทัด จากนั้นจึงทำการแยกอักขระของแต่ละบรรทัด พร้อมระบุหรือบันทึกตำแหน่งของอักขระบนบรรทัด เพื่อช่วยในการวิเคราะห์การเรียงอักขระกลับสู่รูปประโยคเดิม หลังการรู้จำอักขระทุกตัวบนบรรทัดเรียบร้อยแล้ว โครงข่ายนิวรอนที่ทำการจำลองขึ้นเพื่อใช้ในการรู้จำอักขระภาษาไทยนี้ สามารถทำการปรับปรุงหรือเปลี่ยนแปลงเพื่อจดจำอินพุตอื่น ๆ ได้อีก เช่นการรู้จำลายนิ้วมือหรืออื่น ๆ อีกได้ต่อไป

เอกสารนี้มีไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2 ประวัติความเป็นมา

ประวัติความเป็นมาของโครงข่ายนิวรอนนั้น เกิดจากแนวความคิดที่ต้องการให้คอมพิวเตอร์มีความสามารถเหมือนมนุษย์ คือสามารถตัดสินปัญหาต่าง ๆ ที่เข้ามาได้โดยตัวมันเอง จึงมีแนวความคิดที่จะจำลองการทำงานสมองของมนุษย์ โดยโครงข่ายนิวรอน (Neural Network) คือการจำลองรูปแบบทางชีววิทยาทางโครงสร้างสมองของมนุษย์ ซึ่งเป็นการจำลองเอาบางส่วนที่จำเป็นมาพร้อมผสมกับรูปแบบการจัดการโครงสร้างที่ถูกคิดขึ้น เพื่อทำให้เกิดฟังก์ชั่นที่ต้องการ โดยทั่วไปแล้วลักษณะโครงสร้างของโครงข่ายนิวรอนมีองค์ประกอบพื้นฐานคือ ตัวนิวรอนเซล ซึ่งเป็นเซลทางสมองของมนุษย์ โดยโครงข่ายนิวรอน (Neural Network) จำลองแบบมา ลักษณะโดยทั่วไปของนิวรอนเซลจะประกอบไปด้วย ส่วนของอินพุตที่เข้ามา พร้อมกับค่าน้ำหนัก (weight) ที่สามารถปรับค่าได้ ผ่านเข้ามาตรงส่วนที่ทำหน้าที่รวมสัญญาณอินพุต เมื่อรวมสัญญาณอินพุตเรียบร้อยแล้วจึงค่อยผ่านเข้าสู่ฟังก์ชั่นการกระตุ้น (Activation Function) ซึ่งจะทำการแปลงสัญญาณให้อยู่ในระหว่างค่า 0 และ 1 ดังรูปข้างล่าง



รูปที่ 2-2 รูปแสดงองค์ประกอบของนิวรอนเซล

2.2.1 เทคนิคและวิธีการในการฝึกสอนโครงข่ายนิวรอนในการฝึกสอนโครงข่ายนิวรอนให้เรียนรู้สิ่งต่าง ๆ ทำได้ 2 วิธีด้วยกัน ดังต่อไปนี้

1) ซุปเปอร์ไวส์เลินนิ่งอัลกอลิทึม (Supervised Learning Algorithm) ตัวอย่างของอัลกอลิทึมที่ใช้วิธีการนี้คือ แบคพร็อบพาเกชั่น อัลกอลิทึก (Backpropagation Algorithm) วิธีการของอัลกอลิทึมนี้คือมีการระบุค่าเป้าหมายให้กับอินพุตที่ใช้ในการฝึกสอนทุกตัว โดยหลักการของวิธีการนี้ประกอบด้วย 2 ขั้นตอนคือ ขั้นตอนไปข้างหน้า (Forward Phase) เพื่อให้ได้มาซึ่งค่าเอาท์พุตทุกตัวบนเอาท์พุตเลเยอร์ จากนั้นนำค่าเอาท์พุตมาเปรียบเทียบกับค่าเป้าหมายเพื่อหาค่าผิดพลาดที่เกิดขึ้นไว้ใช้ในการปรับเวต (weight) ในกระบวนการย้อนกลับ (Backward Phase) โดยหลักการเรียนรู้แบบซุปเปอร์ไวส์เลินนิ่งจะมีความยืดหยุ่นในการจดจำอินพุตที่มีลักษณะคล้าย ๆ กันได้ดี แต่เสียเวลามากต่อการฝึกสอนโครงข่ายนิวรอนให้รู้จำอินพุตทุกตัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2) อันซุปเปอร์ไวส์เลินนิ่งอัลกอลิทึม (Unsupervised learning Algorithm) ตัวอย่างของอัลกอลิทึมที่ใช้วิธีการนี้เช่น เคาท์เตอร์พรอบพาเกชั่น อัลกอลิทึม (Counterpropagation Algorithm) วิธีการของอัลกอลิทึมนี้จะไม่มีการระบุค่าเป้าหมายให้แก่อินพุตที่ใช้ในการฝึกสอนแต่จะให้โครงข่ายนิวรอนเรียนรู้กับตัวอย่างของอินพุตที่ต้องการจดจำเพื่อให้โครงข่ายปรับตัวในลักษณะการสร้างค่าเฉลี่ยของเวต (weight) ต่ออินพุตลักษณะกลุ่มหนึ่ง ๆ เพื่อใช้ในการเปรียบเทียบกับตัวอย่างของอินพุตแต่ละกลุ่มนั้น โดยหลักการเรียนรู้ของวิธีนี้จะต้องป้อนอินพุตเป็นกลุ่ม ๆ เช่น ต้องการให้โครงข่ายนิวรอนจดจำอินพุตกลุ่มอักขระ "ก" ก็จะป้อนอักขระกลุ่ม "ก" จนกว่าโครงข่ายนิวรอนจะสร้างเวต (weight) เฉลี่ยของอักขระกลุ่ม "ก" ไว้ใช้ในการจดจำอักขระ "ก" จึงจะทำการป้อนอักขระกลุ่มอื่น ๆ เพื่อให้โครงข่ายนิวรอนสามารถจดจำอักขระกลุ่มอื่น ๆ ต่อไป ซึ่งหลักการเรียนรู้แบบอันซุปเปอร์ไวส์เลินนิ่งจะมีปัญหาในการจดจำอักขระซึ่งมีลักษณะใกล้เคียงกันมาก ๆ แต่มีความสะดวกในการฝึกสอนเพราะเสียเวลาในการฝึกสอนโครงข่ายไม่นานกว่าแบบแรก

ในส่วนของวิทยานิพนธ์ฉบับนี้ได้ศึกษาและเลือกใช้วิธีการแบบซุปเปอร์ไวส์เลินนิ่งอัลกอลิทึมและใช้อัลกอลิทึมแบบแบคพรอบพาเกชั่นในการจดจำอักขระตัวพิมพ์ภาษาไทยเพราะอัลกอลิทึมนี้มีความสามารถที่ดีต่อการจดจำอินพุตที่มีลักษณะคล้าย ๆ กันได้ดี เช่น "ค" "ต" "ด" เป็นต้น

2.3 **ขั้นตอนการรู้จำตัวอักษร** สามารถแบ่งออกเป็น 3 ขั้นตอนใหญ่ ๆ ด้วยกันได้คือ

1) การแยกตัวอักษรแต่ละตัวออกมาจากข้อมูลหน้ากระดาษ (Segmentation)

2) การรู้จำตัวอักษร (Recognition)

3) การจัดเรียงรูปประโยค กลับสู่รูปแบบประโยคเดิมหลังการรู้จำ (Rearange character to Statement)

2.3.1 การแยกตัวอักษรแต่ละตัวออกมาจากข้อมูลหน้ากระดาษ (Segmentation) การแยกตัวอักษรแต่ละตัวออกมาจากข้อมูลหน้ากระดาษเป็นกระบวนการในการเตรียมข้อมูลเบื้องต้นก่อนเข้าสู่กระบวนการรู้จำตัวอักษรโดยโครงข่ายนิวรอน โดยเมื่อป้อนแฟ้มข้อมูลภาพที่ได้จากเครื่องตรวจกวาดภาพที่ต้องทำการแปลงข้อมูลที่ได้ให้อยู่ในรูปข้อมูลเพื่อการประมวลผลเสียก่อน โดยข้อมูลต้องอยู่ในรูปรหัส 0 และ 1 คือ ใช้ 0 แทนพื้นขาวของหน้ากระดาษ และใช้ 1 แทนเนื้อตัวอักษร และจากนั้นจึงค่อยจัดลำดับการเข้าประมวลผล โดยมีขั้นตอนดังต่อไปนี้

1) หลังจากแปลงข้อมูลให้อยู่ในรูปรหัส 0 และ 1 ซึ่งแทนข้อมูลภาพอักษรหนึ่งหน้ากระดาษแล้วขั้นตอนต่อไปคือการแบ่งข้อมูลภาพอักษรหนึ่งหน้ากระดาษออกจากกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เป็นบรรทัด ๆ (ตามแนวนอน) โดยอาศัยช่องว่างของข้อมูลตามแนวนอนระหว่างบรรทัดเป็นตัวพิจารณา ซึ่งช่องว่างของหน้ากระดาษจะมีลักษณะคงที่เพราะในการวิจัยนี้ได้กำหนดรูปแบบและขนาดของอักขระไว้ 1 รูปแบบ คือ อักขระรูปแบบ AngsanaUPC ขนาด 12 dpi ทำให้ระยะห่างของอักขระระหว่างบรรทัดมีค่าไม่เกินกว่าค่าคงที่ที่เลือกไว้คือ 12 ซึ่งจากข้อมูลภาพหนึ่งหน้ากระดาษซึ่งแทนรหัสในส่วนของเนื้อตัวอักษรด้วย 1 และส่วนของพื้นอักษรด้วย 0 นั้น ในการแยกแต่ละบรรทัดออกจากกันก็โดยการพิจารณาหารหัสที่เป็น 0 ที่มีความต่อเนื่องกันตามแนวนอนเท่ากับความยาวของค่าคงที่ซึ่งแทนช่องว่างระหว่างบรรทัดเมื่อพบช่องว่างระหว่างบรรทัดแล้วก็ทำการตัดข้อความทีละบรรทัดออกจากหน้ากระดาษและทำการพิจารณาต่อเนื่องกันไปจนกว่าจะสิ้นสุดข้อมูลภาพหนึ่งหน้ากระดาษ

2) หลังจากทำการแบ่งข้อมูลออกเป็นแต่ละบรรทัดแล้ว จึงทำการแยกข้อมูลเฉพาะอักษรแต่ละตัวออกจากบรรทัด โดยอาศัยเทคนิคการค้นหารหัส 1 ซึ่งแทนเนื้อตัวอักษรที่ติดกันออกทีละ 1 ตัวอักษร โดยการเปลี่ยนรหัสจาก 1 เป็นรหัส 2 เพื่อแสดงเนื้อที่ติดกัน เมื่อทำการแปลงเนื้อของ 1 ตัวอักษรที่ติดกันจาก 1 เป็น 2 หมดแล้วเราก็จะได้เนื้ออักขระ 1 ตัว ที่มีเฉพาะเลข 2 ซึ่งถ้าเราแยกอักขระที่มีรหัสเฉพาะ 2 ออกมาจากบรรทัด เราก็จะได้อักขระ 1 ตัว ออกมา เมื่อขณะทำการแยกอักขระออกจากบรรทัดเราต้องหาตำแหน่ง Xcentroid , Ycentroid ของอักขระบนบรรทัดพร้อมกันไปด้วย โดยมีสูตรคือ

$$Ix = \frac{\sum_j \sum_i i * F(i,j)}{\sum_j \sum_i F(i,j)} \qquad (2.1)$$

$$Iy = \frac{\sum_i \sum_j j * F(i,j)}{\sum_i \sum_j F(i,j)} \qquad (2.2)$$

เมื่อ F(i,j) คือ จุดภาพใด ๆ ของภาพตัวอักษรที่มีค่าเป็น 0 หรือ 1

i , j คือ ตำแหน่งพิกัดของจุดภาพในแนวแถวและคอลัมน์ตามลำดับ

Ix , Iy คือ ตำแหน่งพิกัด (Coordinate) ของจุดศูนย์ถ่วง

เมื่อเราทราบจุดศูนย์ถ่วงของตัวอักษรเราก็สามารถใช้ค่านี้ในการจัดเรียงรูปประโยคเป็นรูปประโยคเดิมหลังการรู้จำได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.2 การรู้จำตัวอักษร (Recognition) เป็นกระบวนการหลังจากการแยกตัวอักษรออกจากบรรทัดของหน้ากระดาษ (Segmentation) โดยเมื่อทำการแยกอักษรแต่ละตัวบนบรรทัดพร้อมระบุตำแหน่ง Ix , Iy ของอักขระบนบรรทัดแล้ว ก็จะได้ไฟล์ซึ่งมีอักขระเรียงพร้อมตำแหน่ง Ix , Iy ของอักขระพร้อมสัญลักษณ์ระบุการสิ้นสุดบรรทัดบนไฟล์

เมื่อได้ไฟล์ดังกล่าวเราก็จะทำการป้อนอินพุตไฟล์ที่ได้เข้าสู่โครงข่ายนิวรอนที่ทำการฝึกสอนให้รู้จำอินพุตต้นแบบไว้แล้ว เพื่อให้โครงข่ายนิวรอนระบุว่าอินพุตที่ป้อนเข้าไปมีค่าเป้าหมายอะไร โดยผลลัพธ์ของการรู้จำจะเก็บไว้บนไฟล์พร้อมค่าเป้าหมายของอักขระดังกล่าวซึ่งคือรหัสแอสกี้ของอักขระนั้นๆตามด้วยค่าตำแหน่งของอักขระนั้นบนหน้ากระดาษ

2.3.3 การจัดเรียงรูปประโยคกลับสู่ประโยครูปแบบเดิมหลังการรู้จำ

เมื่อผ่านกระบวนการที่ 2 มาแล้วเราจะต้องทำการสลับตำแหน่งของอักขระให้อยู่ในตำแหน่งเดิมก่อนการแยกตัวอักษรออกจากบรรทัด ซึ่งตัวอักษรแต่ละตัวที่ถูกแยกออกจากภาพประโยค ก่อนที่จะส่งผ่านไปทำการวิเคราะห์และรู้จำจะถูกบันทึกค่าตำแหน่งพิกัด Ix , Iy ซึ่งก็จะนำไปแทนตำแหน่งของตัวอักษรในภาพประโยคได้เมื่อภาพตัวอักษรถูกส่งไปทำการรู้จำ และได้ผลลัพธ์ออกมาเป็นรหัสแอสกี้ของตัวอักษรนั้น เราก็จะนำผลลัพธ์ของการรู้จำที่ได้มาเรียงลำดับอักขระที่ถูกต้องก่อนการบันทึกลงไฟล์ ซึ่งการจัดเรียงตัวอักษรของแต่ละบรรทัดจะยึดตามค่า Ix (ตำแหน่งของอักขระในแนวแกน x) จากค่าน้อยไปหาอักขระที่มีค่า Ix มาก บนบรรทัดแต่ละบรรทัด เพื่อจัดเรียงตัวอักษรให้อยู่ในตำแหน่งที่ใกล้เคียงกับส่วนประกอบที่ประกอบขึ้นเป็นคำจากนั้นจึงทำการหาอักขระที่อยู่บนแนวคอลัมน์เดียวกัน และเรียงลำดับอักขระบนคอลัมน์ให้ถูกตำแหน่ง ซึ่งตัวอย่างของการจัดเรียงอักขระบนแนวคอลัมน์เดียวกัน สามารถแสดงได้ด้วยตัวอย่างของคำ “รู้” ซึ่งขั้นแรกเราต้องทำการหาความสูงของอักขระบนแนวคอลัมน์เดียวกัน ว่าอักขระตัวใดมีค่าสูงสุดก็จะจัดเรียงอักขระนั้นเป็นตัวแรกในการจัดเรียง จากนั้นเรียงอักขระตัวต่อมาด้วยอักขระที่มีค่า Iy น้อยกว่าตัวแรก ซึ่งจากตัวอย่างคือ ◌้ (ไม้โท) และตามด้วยอักขระที่มีค่า Iy มากกว่าอักขระตัวแรก (ร) ซึ่งผลคือ ◌ู (สระอู) ดังนั้นผลลัพธ์ของการจัดเรียงอักขระบนแนวคอลัมน์นี้จะได้ “ร ◌้ ◌ู” เป็นต้น

จากกระบวนการของการรู้จำอักษรโดยโครงข่ายนิวรอนแบบแบค็พรอบพาเกชั่นที่ใช้งานวิจัยในวิทยานิพนธ์ฉบับนี้ สามารถชี้ให้เห็นลักษณะเด่นของวิธีการรู้จำแบบแบคพรอบพาเกชั่นได้คือ สามารถแยกแยะอักขระที่มีลักษณะเหมือนกันได้ดีและ เวลาของการจดจำโดยโครงข่ายนั้นใช้เวลาสั้น และสามารถทำให้โครงข่ายนิวรอนเรียนรู้อินพุตรูปแบบใหม่ ๆ ได้เอง โดยทำการสร้างชุดฝึกสอนขึ้นใหม่แล้วนำไปฝึกสอนให้แก่โครงข่ายนิวรอนพร้อมระบุค่าเป้าหมายลงไปในอินพุต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชุดฝึกสอนนั้นทุก ๆ ตัว โดยจุดเด่นของงานวิจัยนี้ได้ทำการพัฒนาโครงข่ายนิวรอนบนเครื่องมินิคอมพิวเตอร์โดยผ่านเน็ตเวิร์คทำให้ผู้ใช้ที่ต้องการใช้งานสามารถใช้งานได้สะดวกโดยเข้าสู่ในระบบเน็ตเวิร์ก พร้อมทั้งขอเข้าสู่เครื่องมินิคอมพิวเตอร์บนระบบเน็ตเวิร์คที่ชื่อ diamond ก็สามารถใช้งานโปรแกรมโครงข่ายนิวรอนเพื่อการรู้จำอักขระได้อีกทั้งสามารถใช้งานได้หลาย ๆ ผู้ใช้ขณะเดียวกันได้เพราะระบบเน็ตเวิร์คเอื้อประโยชน์ต่อการใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

การแยกตัวอักษร

ขั้นตอนการทำงานที่มีส่วนสำคัญของการรู้จำตัวอักษรภาษาไทย ส่วนหนึ่งก็คือขั้นตอนในการแยกตัวอักษร (Segmentation) คือเป็นขั้นตอนการดึงภาพเฉพาะ 1 ตัวอักษรออกมาจากประโยคในภาพหน้าเอกสารของแต่ละบรรทัด พร้อมทั้งบันทึกตำแหน่งของตัวอักษรที่แยกออกมาเพื่อนำไปเข้าสู่กระบวนการรู้จำตัวอักษรที่ทำการประมวลผลการรู้จำได้ทีละ 1 ตัวอักษรเท่านั้น ซึ่งจากข้อมูลที่ใช้ในการทำงานจริงจะมีลักษณะเป็นหน้าเอกสารหนึ่งหน้า หรือเป็นข้อความของบรรทัดหลายบรรทัด ดังนั้นจึงต้องทำการแยกตัวอักขระทีละหนึ่งตัวออกจากอักขระกลุ่มนั้น ๆ ออกมาเป็นข้อมูลอักขระเดี่ยว ๆ เสียก่อนจึงทำการรู้จำตัวอักษร

จากการพิจารณาลักษณะของอักขระภาษาไทยนั้น เราสามารถสรุปได้ว่าระดับของอักขระภาษาไทยแบ่งออกเป็น 4 ระดับ คือ

ระดับที่ 1 เป็นระดับบนสุด ประกอบไปด้วย วรรณยุกต์ และตัวการันต์

ระดับที่ 2 เป็นระดับกลางบน ประกอบไปด้วย สระระดับบน เช่น

ระดับที่ 3 เป็นระดับกลางล่าง ประกอบไปด้วย พยัญชนะ และสระระดับกลาง เช่น า , ไ , โ , ฯลฯ

ระดับที่ 4 เป็นระดับล่างสุด ประกอบไปด้วย สระระดับล่าง เช่น ฯลฯ

ลักษณะของอักขระที่ทำให้เกิดปัญหาในการแยกตัวอักษรและการรู้จำตัวอักษร คือ

1) อักษรที่ติดกันหรือ อักษรที่ขาดหาย ทำให้เกิดความไม่สมบูรณ์ของตัวอักษรในการรู้จำ

2) อักษรที่เป็นอักขระที่เขียนแยกกัน เช่น แ หรือ ะ เป็นต้น

จากลักษณะของอักขระดังกล่าวเพื่อให้การรู้จำตัวอักษรเป็นไปได้อย่างถูกต้องเราจึงกำหนดให้หน้ากระดาษที่จะทำการรู้จำต้องมีสิ่งรบกวน (noise) ไม่มากจนทำให้เกิดปัญหาของการแยกตัวอักษรออกเป็นตัวอักษรเดี่ยว ๆ ได้ ซึ่งจากผลการทดลองพบว่า หน้ากระดาษซึ่งพิมพ์ด้วยรูป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

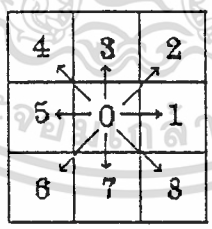
แบบ Angsana UPC ขนาด 12 dpi ขนาด Regular พิมพ์ด้วยเครื่องพิมพ์เลเซอร์สามารถให้อักขระที่ชัดเจนพอต่อการแยกอักขระและอัตราส่วนขณะทำการสแกนหน้าเอกสารด้วยเครื่องตรวจกวาดภาพ (Image Scanner) นั้นต้องใช้อัตราส่วนดังนี้คือ

Resolution = 300 dpi , Scale = 100% , Brightness = 100%

จึงทำให้หน้าเอกสารมีความชัดเจนพอต่อการแยกตัวอักษรเดี่ยว ๆ ออกจากหน้าเอกสารเพื่อนำไปใช้ในการรู้จำตัวอักษรได้

3.1) ทฤษฎีที่นำมาใช้ในการแยกตัวอักษร

ในส่วนของทฤษฎีที่นำมาใช้ในการแยกภาพตัวอักษรออกเป็นตัวอักษรเดี่ยว ๆ ก็คือทฤษฎีติดตามเนื้อตัวอักษรที่มีรหัส "1" เหมือนกันหรือติดต่อกัน โดยหลักการของทฤษฎีนี้คล้าย ๆ กับหลักการติดตามรอยขอบของภาพของทฤษฎีเซลลูล่าร์ออโตมาต้า (Cellular Automata) โดยหลักการของทฤษฎีนี้ก็คือการแยกเนื้อรหัส "1" ที่ติดกันออกจากกลุ่มอักขระในบรรทัด คือจะทำการเปลี่ยนรหัสที่ติดกันจากรหัส "1" ไปเป็นรหัส "2" ทั้งหมดของอักขระ 1 ตัว ที่ต้องการแยกออกจากกลุ่มอักขระบนบรรทัด โดยทิศทางการเปลี่ยนรหัส "1" ที่ติดกันจะพิจารณาใน 8 ทิศทางของรหัส "1" ที่ต้องการแพร่รหัสจาก "1" ไปเป็น "2" ของอักขระ 1 ตัว ดังภาพ

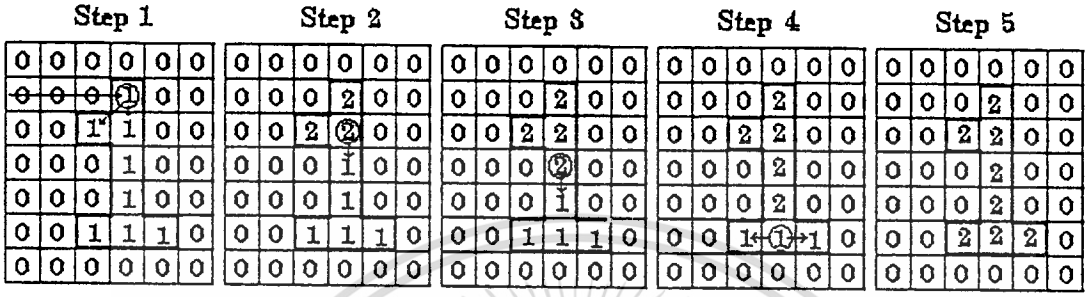


รูปที่ 3-1 แสดงทิศทางการเปลี่ยนรหัส "1" ที่ติดกันทั้ง 8 ทิศ

เมื่อแพร่รหัส "1" ที่ติดกันของเนื้ออักขระ 1 ตัว จนหมด 1 ตัวอักษร คือไม่มีรหัส 1 ให้เปลี่ยนได้อีกจึงค่อยทำการแยกอักขระบนบรรทัดซึ่งมีเฉพาะรหัส "2" ออกมา เราก็จะได้อักขระ 1 ตัว จากการแยกพร้อมบันทึกตำแหน่งของอักขระบนบรรทัดนั้น ๆ ด้วย เราจะทำกับอักขระทุกตัวบนบรรทัดนั้น ๆ จนกว่าจะแยกอักขระได้ครบทุกตัวบนบรรทัด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

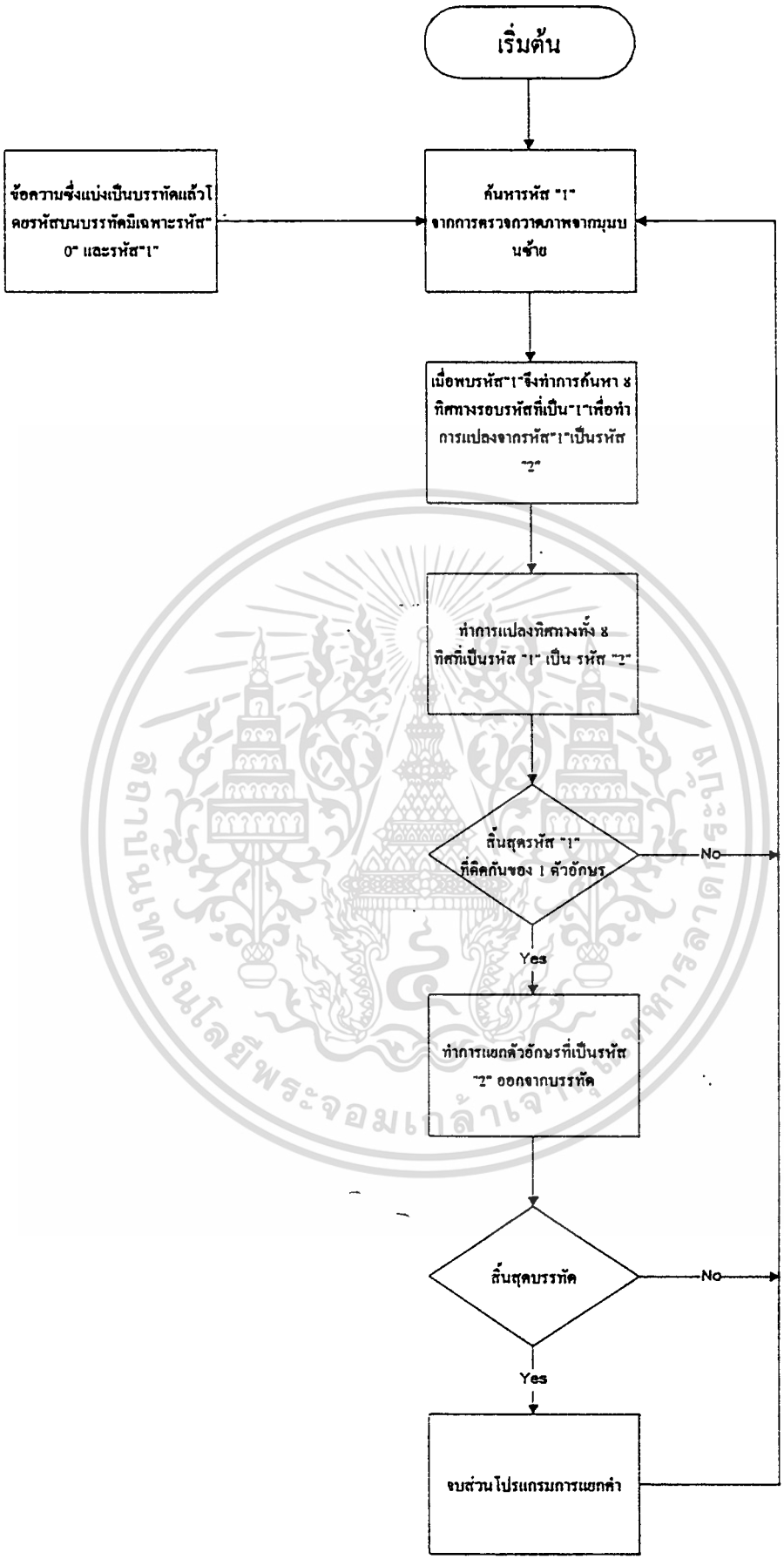
จากรูปที่ 3-2 แสดงขั้นตอนการเปลี่ยนเนื้อรหัส "1" ที่ติดกันของอักขระหนึ่งตัวเพื่อการแยกอักขระเลขหนึ่ง "1"



รูปที่ 3-2 รูปแสดงขั้นตอนการเปลี่ยนเนื้อรหัส "1" ที่ติดกันของอักขระนั้น

เราสามารถแทนการค้นหารหัส "1" ที่ติดกันของเนื้ออักขระ 1 ตัว เพื่อทำการแยกอักขระ 1 ตัว ด้วยโฟลว์ชาร์ทได้คือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



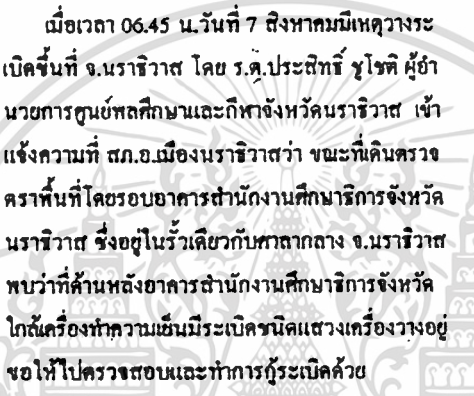
รูปที่ 3-3 รูปแสดงโฟลว์ชาร์ทการแยกอักขระ 1 ตัว ออกจากข้อความที่เป็นบรรทัด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2) ปัญหาและข้อจำกัดของทฤษฎีที่นำมาใช้ในการแยกภาพตัวอักษรออกจากข้อความ

ปัญหาและข้อจำกัดของการแยกภาพตัวอักษร 1 ตัว ออกจากข้อความบนบรรทัดโดยการใช้ทฤษฎีติดตามเนื้อตัวอักษรที่มีรหัส "1" เหมือนกันหรือติดกันนี้จะไม่สามารถแยกภาพอักขระ 2 ตัว ที่ติดกันจากสาเหตุสิ่งรบกวน (noise) ได้ และไม่สามารถแบ่งแยกภาพอักขระที่พิมพ์ห่างกันแต่มีความหมายเป็นอักขระตัวเดียวกันได้คือ ญ , แ , ะ ให้เป็นตัวอักขระเดียวกันได้

จากภาพข้างล่างคือภาพอักขระตัวอย่างซึ่งมีสิ่งรบกวนของภาพทำให้ไม่สามารถใช้ทฤษฎีติดตามเนื้อตัวอักษรที่มีรหัส "1" ติดกันเพื่อแยกอักขระทีละหนึ่งตัวออกจากภาพประโยคได้


เมื่อเวลา 06.45 น.วันที่ 7 สิงหาคมมีเหตุวางระ
เบิดขึ้นที่ จ.นราธิวาส โดย ร.ต.ประสิทธิ์ ชูโชติ ผู้อำ
นวยการศูนย์พลศึกษาและกีฬาจังหวัดนราธิวาส เข้า
แจ้งความที่ สภ.อ.เมืองนราธิวาสว่า ขณะที่คนตรวจ
ตราพื้นที่โดยรอบอาคารสำนักงานศึกษาธิการจังหวัด
นราธิวาส ซึ่งอยู่ในรั้วเดียวกับศาลากลาง จ.นราธิวาส
พบว่าที่ด้านหลังอาคารสำนักงานศึกษาธิการจังหวัด
ใกล้เครื่องทำความเย็นมีระเบิดชนิดแสวงเครื่องวางอยู่
ขอให้ไปตรวจสอบและทำการกู้ระเบิดด้วย

รูปที่ 3-4 รูปแสดงภาพหน้ากระดาษเอกสารจากการตรวจกวาดภาพที่มีสิ่งรบกวนบนหน้ากระดาษ

3.3) ขั้นตอนการแยกตัวอักขระ

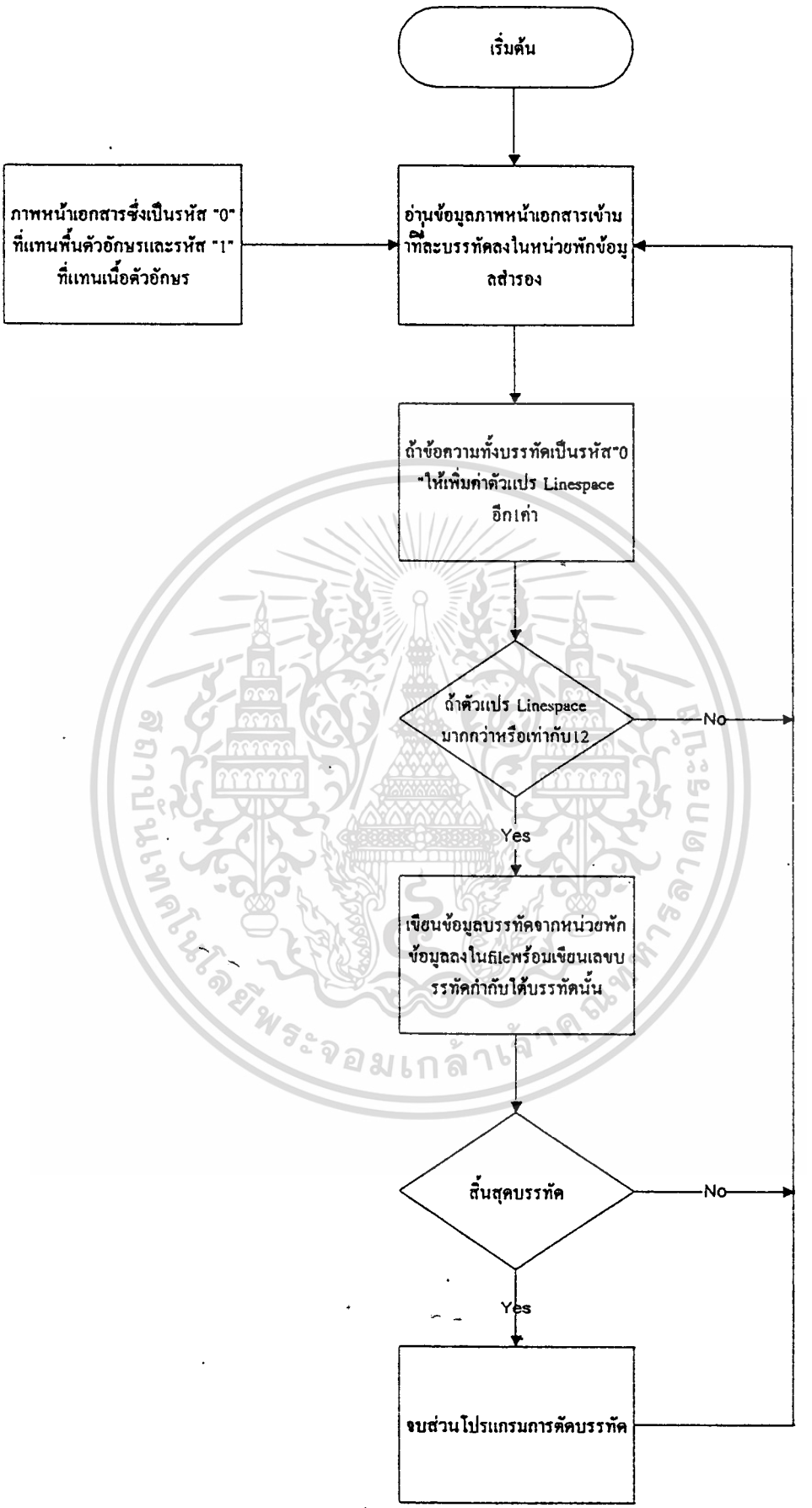
3.3.1) การแยกแยะภาพอักษร หนึ่งหน้ากระดาษออกเป็นบรรทัด

เมื่อรับข้อมูลภาพที่ได้จากเครื่องกวาดภาพ (Image Scanner) แล้วต้องทำการเปลี่ยนข้อมูลที่อยู่ในรูปของ Tiff file format ให้อยู่ในรูปแบบที่ใช้งานประมวลผลได้ โดยต้องอ่านข้อมูลหลักจากโครงสร้างของแฟ้มข้อมูล Tiff file ที่ได้กล่าวไว้แล้วในตอนต้น แล้วดำเนินการแปลงข้อมูลให้อยู่ในรูปรหัส 0 และ 1 เมื่อได้ข้อมูลที่เป็นรหัสแทนทั้งในส่วนที่เนื้อตัวอักษรซึ่งแทนด้วยรหัส "1" และส่วนที่เป็นพื้นตัวอักษรซึ่งแทนด้วยรหัส 0 ทั้งหนึ่งหน้ากระดาษแล้ว จึงทำการสแกนหาช่องว่างระหว่างบรรทัดหรือสแกนหารหัส 0 ที่ต่อเนื่องกันตลอดความกว้างของข้อมูลภาพนั่นเอง โดยรูปแบบของอักขระที่เลือกใช้ไว้ 1 รูปแบบคือ. รูปแบบ Angsanna UPC ขนาด 12 dpi รูปแบบ Regular ซึ่งจะทำให้ช่องว่างระหว่างบรรทัดนั้นอยู่ในช่วงค่า ๆ หนึ่ง ซึ่งจากการทดลองพบว่าช่อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ว่างระหว่างบรรทัดที่ทำการสแกนมาอยู่ที่บรรทัดที่เป็นรหัส “0” ที่ต่อเนื่องกัน 12 บรรทัด ซึ่งเป็นระยะขอบเขตด้านบน และขอบเขตด้านล่างของแต่ละบรรทัด ซึ่งเป็นการแยกแยะตัวอักษรออกเป็นแต่ละบรรทัด โดยขั้นตอนการแยกแยะภาพอักษรหนึ่งหน้ากระดาษออกเป็นบรรทัด สามารถเขียนเป็นลำดับการทำงานของโฟลว์ชาร์ทได้คือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3-5 รูปแสดงขั้นตอนการแยกภาพอักษรหนึ่งหน้ากระดาษออกเป็นบรรทัด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.2) การแยกภาพตัวอักษรออกจากบรรทัดพร้อมระบุตำแหน่งของภาพตัวอักษร

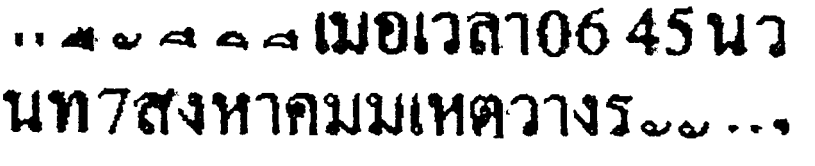
เมื่อได้รหัสข้อมูลภาพที่แทนด้วยรหัส 0 และ 1 ในแต่ละบรรทัดแล้วเราจึงดำเนินการพิจารณาแยกแยะตัวอักษรในขั้นตอนต่อไป โดยวิธีการติดตามเนื้อตัวอักษรที่มีรหัส "1" ติดต่อกัน โดยการสแกนหารหัส "1" จากมุมบนซ้ายเพื่อพิจารณารหัสที่เป็น "1" ที่มีค่าติดกันของอักขระ 1 ตัวอักษร และทำการเปลี่ยนรหัสที่ติดกันของเนื้อตัวอักษรจากรหัส "1" เป็นรหัส "2" ตามทิศทาง 8 ทิศ รอบเนื้อตัวอักษรที่เป็น 1 นั้นจนกว่าจะหมดรหัส 1 ของเนื้อตัวอักษรนั้น 1 ตัวอักษร จากนั้นจึงทำการแยกรหัส "2"ของเนื้อตัวอักษรนั้นออกจากบรรทัดนั้นพร้อมทั้งระบุตำแหน่ง Ix , Iy (จุดศูนย์ถ่วงของอักขระนั้น) , ค่าพิกัด Xmin , Xmax , Ymin, Ymax ของอักขระตัวนั้น ๆ และดำเนินการสแกนหาอักขระแต่ละตัวจนหมดระยะของบรรทัดนั้น ๆ ดังแสดงในรูปข้างล่าง

เมื่อเวลา 06.45 น.วันที่ 7 สิงหาคมมีเหตุวางระ
Line1
เบิดขึ้นที่ จ.นราธิวาส โดย ร.ต.ประสิทธิ์ รูโรหิ ผู้อำ
Line2
นวยการศูนย์พลศึกษาและกีฬาจังหวัดนราธิวาส เข้า
Line3
แจ้งความที่ สภ.อ.เมืองนราธิวาสว่า ขณะที่ดินตรวจ
Line4
ตราพื้นที่โดยรอบอาคารสำนักงานศึกษาธิการจังหวัด
Line5
นราธิวาส ซึ่งอยู่ในรั้วเดียวกับสากลาง จ.นราธิวาส
Line6
พบว่าที่ด้านหลังอาคารสำนักงานศึกษาธิการจังหวัด
Line7
ใกล้เครื่องทำความเย็นมีระเบิดชนิดแสวงเครื่องวางอยู่
Line8
จอให้ไปตรวจสอบและทำการกู้ระเบิดด้วย
Line9
<End of page>

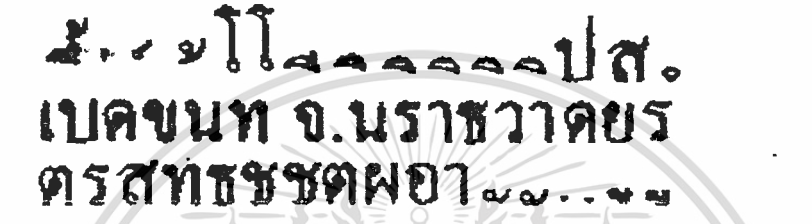
รูปที่ 3-6 รูปแสดงข้อความในแต่ละบรรทัดหลังจากทำการแยกบรรทัดจากหน้าเอกสาร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

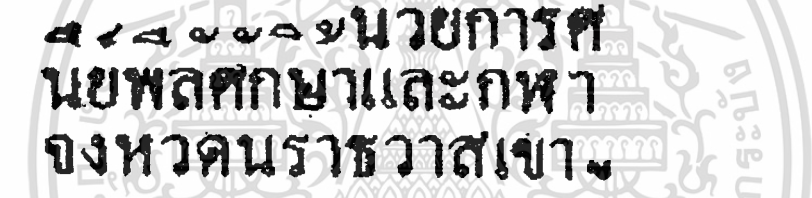
ซึ่งจากภาพรูปที่ 3-6 เราสามารถแยกตัวอักษรออกจากรูปประโยคข้อความได้ตามลำดับดังรูปที่3-7



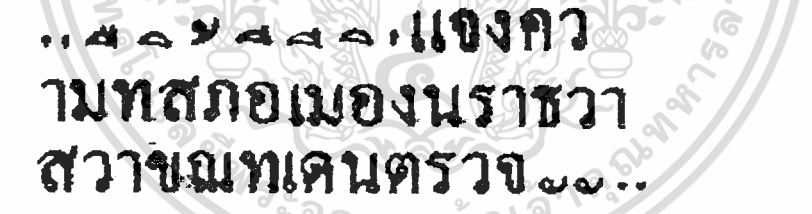
Line 1



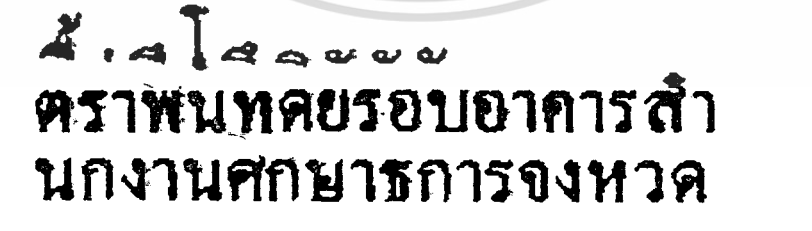
Line 2



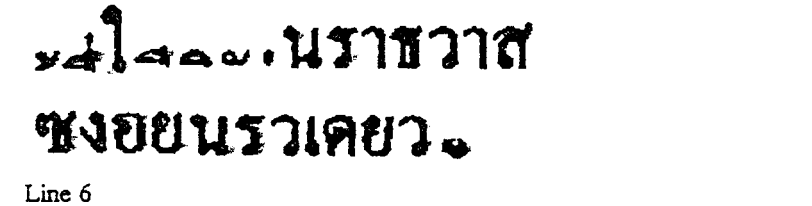
Line 3



Line 4

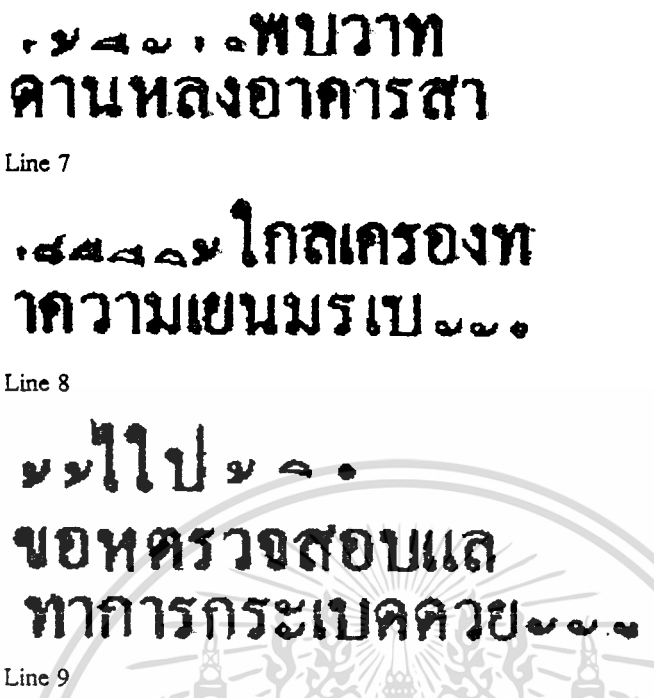


Line 5



Line 6

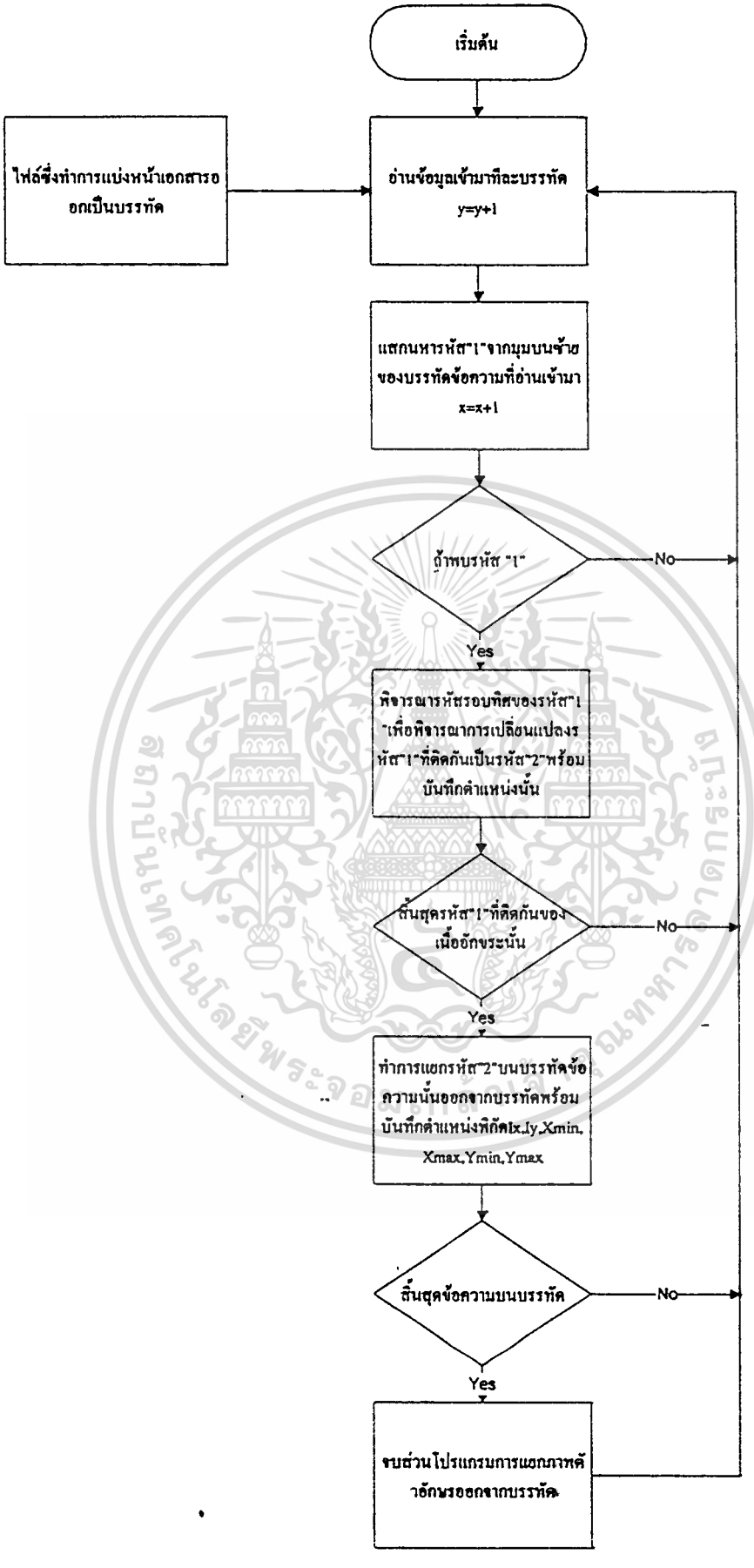
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3-7 ภาพแสดงไฟล์ซึ่งทำการแยกอักขระแต่ละตัวออกจากบรรทัดนั้น ๆ

ซึ่งขั้นตอนการแยกภาพตัวอักษร 1 ตัว ออกจากบรรทัดพร้อมระบุตำแหน่งของภาพตัวอักษรนั้นสามารถเขียนเป็น ขั้นตอนการทำงานของโฟลว์ชาร์ท ได้ดือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

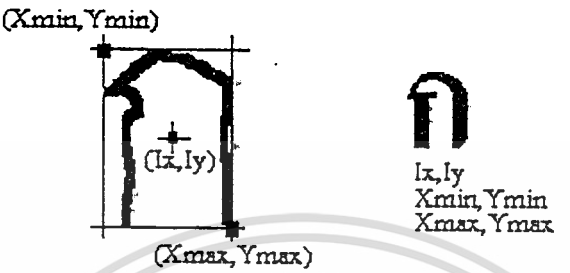


รูปที่ 3-8 รูปแสดงขั้นตอนการแยกภาพตัวอักษรออกจากภาพบรรทัด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.3) การจัดเรียงรูปประโยคหลังการรู้จำ

ตัวอักษรแต่ละตัวที่ถูกแยกออกจากภาพประโยค ก่อนที่จะถูกส่งผ่านไปทำการรู้จำจะถูกบันทึกค่าตำแหน่งพิกัด Xmin , Xmax , Ymin , Ymax และจุดศูนย์ถ่วงของแต่ละตัวอักษรไว้ดังภาพ



ซึ่งค่าตำแหน่งพิกัดของตัวอักษรแต่ละตัวจะนำมาใช้ในการจัดเรียงรูปประโยคหลังการรู้จำก่อนที่จะถูกบันทึกลงไปในไฟล์ผลลัพธ์ของการรู้จำ โดยการจัดเรียงอักขระแต่ละตัวหลังการรู้จำ ขั้นตอนการจัดเรียงรูปประโยคกลับสู่ตำแหน่งเดิมของอักขระแต่ละตัวนั้น มี 3 ขั้นตอนด้วยกันคือ

ขั้นตอนที่ 1 จัดเรียงอักขระแต่ละตัวตามค่า Ix จากน้อยไปหาอักขระที่มีค่า Ix มากในแต่ละบรรทัด เพื่อจัดเรียงตัวอักษรให้อยู่ในตำแหน่งที่ใกล้เคียงกับรูปประโยคเดิม

การทำงานการรู้จำ -> การท.างาน การะรุ.จา

ขั้นตอนที่ 2 หาอักขระที่มีแนวคอลัมน์เดียวกันแล้วจัดเรียงตามลำดับค่า Iy ของอักขระในแนวคอลัมน์เดียวกันจากค่าน้อยไปหาอักขระที่มีค่า Iy มาก ซึ่งจากตัวอย่างอักขระที่มีแนวคอลัมน์เดียวกันคือ ท,รู้,จ

การ.ท างาน การะรุ.จา

ขั้นตอนที่ 3 จัดเรียงอักขระที่มีแนวคอลัมน์เดียวกันตามลำดับของรูปประโยคคือ วางตำแหน่งของอักขระตามลำดับดังนี้

ระดับที่ 3 + ระดับที่ 2 + ระดับที่ 1 + ระดับที่ 4

ซึ่งอักขระในระดับที่ 3 จะมีความสูงสูงที่สุด จึงสามารถใช้เป็นตัวอ้างอิงในการแยกอักขระระดับที่ 2 และอักขระระดับที่ 1 ออกจากอักขระระดับล่างคือ อักขระระดับที่ 4 ได้

การท า.งาน การระะรุ.จ.า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

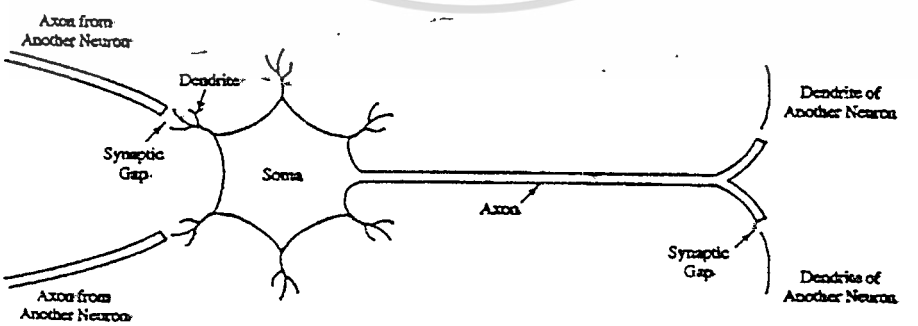
การรู้จำตัวอักษร

4.1 ทฤษฎีพื้นฐานของโครงข่ายนิวรอน (Neural Network)

โครงข่ายนิวรอน (Neural Network) เป็นแนวความคิดหนึ่งทาง AI (Artificial Intelligence) ในการพยายามสร้างเครื่องจักร (machine) ที่จะสามารถเรียนรู้และจดจำ ซึ่งมีกระบวนการ (process) คล้ายคลึงกระบวนการ (process) ทางสมองของมนุษย์

4.1.1 ลักษณะของโครงข่ายนิวรอน

โครงข่ายนิวรอน (Neural Network) มีลักษณะเป็นการจำลองรูปแบบทางชีววิทยาทางโครงสร้างสมองของมนุษย์ โดยการจำลองเอาบางส่วนที่จำเป็นมาใช้ผสมกับรูปแบบการจัดโครงสร้างที่ถูกคิดขึ้น เพื่อทำให้เกิดฟังก์ชั่นที่ต้องการ โดยทั่วไปแล้ว ลักษณะโครงสร้างของโครงข่ายนิวรอน (Neural Network) มีอยู่หลายรูปแบบแต่ลักษณะทางโครงสร้างพื้นฐานจะเหมือนกัน กล่าวคือ ลักษณะทางโครงสร้างพื้นฐานจะประกอบไปด้วยนิวรอน (Neuron) ซึ่งเป็นเซลทางสมองของมนุษย์ซึ่งโครงข่ายนิวรอน (Neural Network) จำลองแบบมา ลักษณะโดยทั่วไปของนิวรอนเซล (neuron cell) มีลักษณะดังรูปข้างล่าง

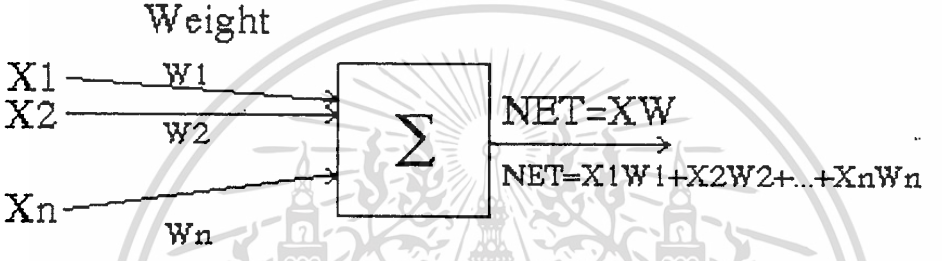


รูปที่ 4-1 ลักษณะของนิวรอนเซล (neuron cell) ของเซลประสาท

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กล่าวคือจะประกอบไปด้วยส่วนของเดนไดร์ท (dendrites) ซึ่งยื่นออกมาจากเซลร่างกาย (cell body) ไปเชื่อมต่อยังนิวรอนเซล (neuron cell) อื่น ๆ โดยที่จุดเชื่อมต่อเหล่านี้ จะมีส่วนของไซแนบด์ (synapse) ซึ่งสามารถปรับตัวได้เมื่อ input ได้ถูกส่งผ่านไซแนบด์ (synapse) มา แล้วจะไปรวมตัวกันที่เซลร่างกาย (cell body) และนิวรอลเซล (neuron cell) จะส่ง output ออกทางแอกซอน (axon) ตามลักษณะการรวมตัวกันของ input ที่เซลร่างกาย (cell body)

สำหรับลักษณะของนิวรอน (neuron) ของโครงข่ายนิวรอน (Neural Network) จะมีลักษณะดังรูปที่ 4.2



รูปที่ 4.2 นิวรอน (Neuron) ของโครงข่ายนิวรอน (Neural Network)

จะมีลักษณะคล้ายกับนิวรอนเซล (Neuron cell) กล่าวคือ ส่วนของอินพุต (input) ที่เข้ามาจะมีค่าน้ำหนัก (weight) ซึ่งเปรียบเสมือนส่วนของไซแนบด์ (Synapse) ซึ่งสามารถปรับค่าได้ ผ่านเข้ามาที่ส่วนที่ทำหน้าที่รวม อินพุท(input) ที่เข้ามาและออกเป็นค่าเอาท์พุท(output)ค่าหนึ่ง สำหรับค่า เอาท์พุท(output) ที่ออกมาสามารถคำนวณตามสมการดังนี้คือ

$$NET = X_1W_1+X_2W_2+.....+X_NW_N \quad (4.1)$$

หรืออาจเขียนในรูปแบบของเวกเตอร์ได้ดังนี้

$$NET = XW \quad (4.2)$$

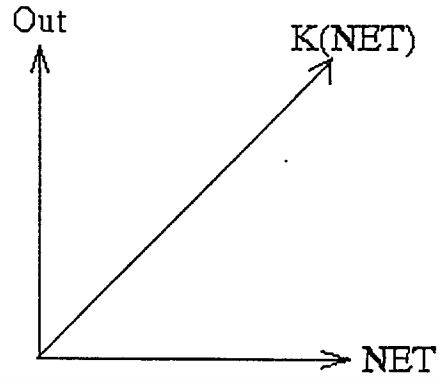
โดย X = เวกเตอร์ อินพุท(input)

W = เวกเตอร์ น้ำหนัก(weight)

NET = เวกเตอร์ เอาท์พุท(output)

สำหรับโดยทั่วไปแล้ว ส่วนเอาท์พุตที่ออกมา จะถูกกระทำต่อไปโดยส่วนที่เรียกว่า ฟังก์ชั่นการกระตุ้น (Activation Function) ซึ่งมีลักษณะของฟังก์ชั่นนี้มีอยู่หลายรูปแบบซึ่งอาจจะมีลักษณะเป็นสมการเชิงเส้นอย่างง่าย ๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

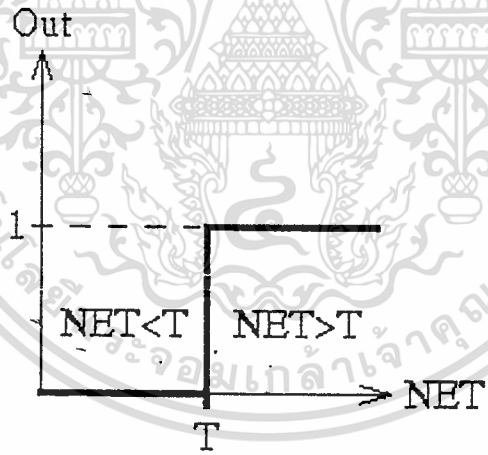


$$OUT = K(NET) \qquad (4.3)$$

ซึ่ง K = ค่าคงที่

รูปที่ 4-3 สมการเชิงเส้นอย่างง่าย

หรือในรูปของฟังก์ชั่น threshold



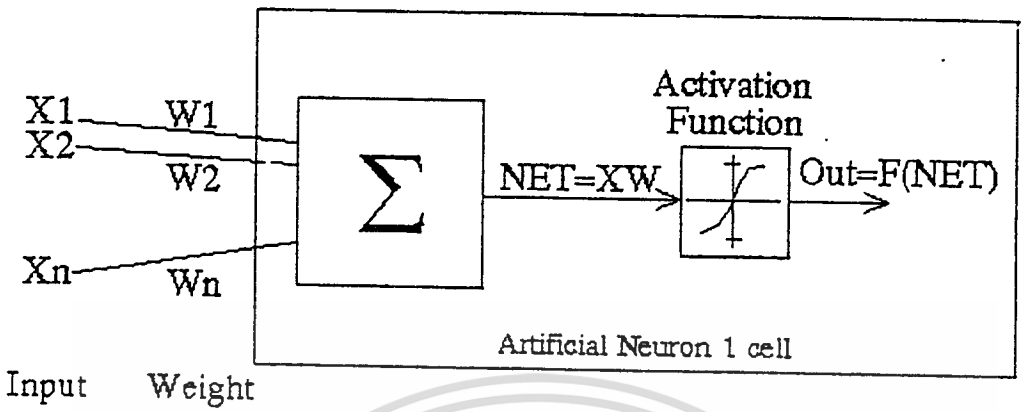
$$OUT = 1 \text{ ถ้า } NET > T \qquad (4.4)$$

$$OUT = 0 \text{ กรณีอื่น} \qquad (4.5)$$

รูปที่ 4-4 สมการ threshold

หรืออาจอยู่ในรูปฟังก์ชั่นอื่นซึ่งมีลักษณะไม่เชิงเส้น (nonlinear function) ในรูปที่ 4-5 แสดงลักษณะของนิวรอน (neuron) ที่ได้รวมเอาฟังก์ชั่นการกระตุ้น (Activation Function) ไว้แล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

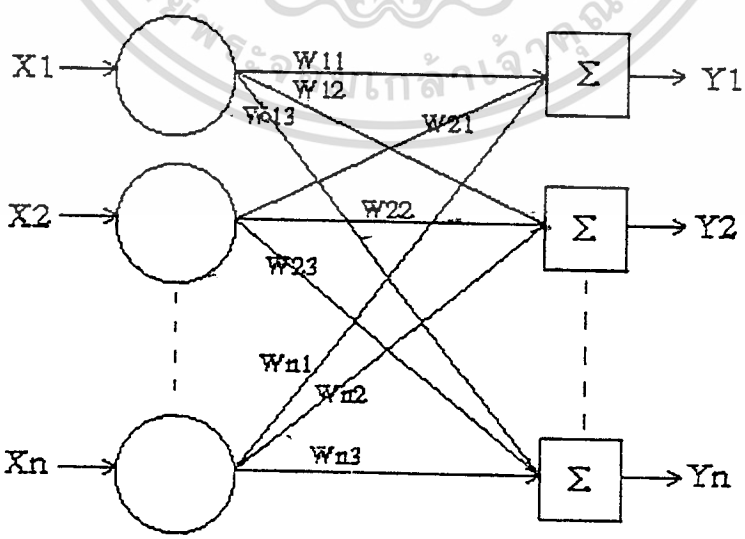


รูปที่ 4-5 ลักษณะของนิวรอน (neuron) 1 หน่วย (1 cell) ที่ประกอบด้วยฟังก์ชั่นการกระตุ้น (Actiuation Function) ที่ใช้ในงานวิจัย

4.1.2 โครงข่ายนิวรอนในลักษณะ 1 ชั้น (Single Layer Neural Network)

ถึงแม้ว่าลักษณะของนิวรอน (Neuron) เพียง 1 นิวรอน (Neuron) จะสามารถทำฟังก์ชั่นบางอย่างได้ แต่ประสิทธิภาพที่เพิ่มขึ้นจากการทำงานมาจากการรวมนิวรอน (Neuron) หลาย ๆ นิวรอน (Neuron) เข้าด้วยกัน

ลักษณะรูปแบบอย่างง่ายที่สุดของโครงข่ายนิวรอน (Neural Network) จะจัดกลุ่มของนิวรอน (Neuron) มีลักษณะ 1 ชั้น (Single Layer) ดังแสดงในรูปที่ 5.4



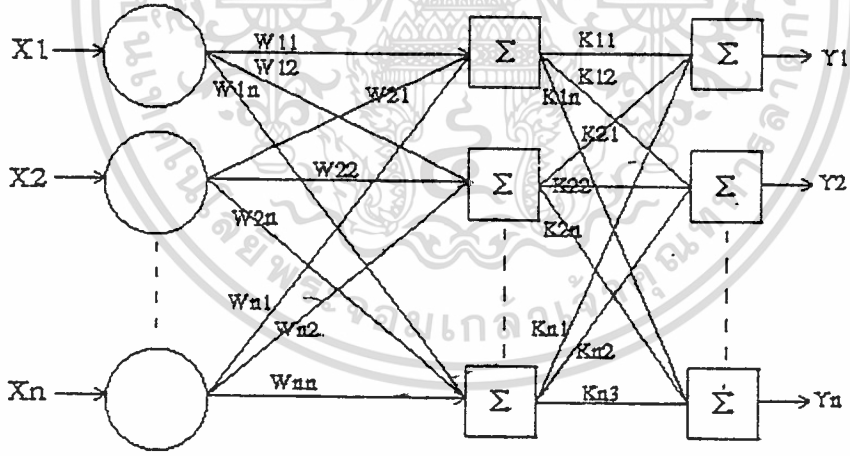
รูปที่ 4-6 ลักษณะของโครงข่ายนิวรอนในลักษณะ 1 ชั้น (Single Layer Neural Network)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับในส่วนของวงกลมจะเป็นเพียงส่วนที่แบ่งกระจายอินพุตไปยังนิวรอน (Neuron) ต่าง ๆ สำหรับลักษณะโครงสร้างในรูปที่ 4-6 อาจจะมีการเชื่อมโยงบางส่วนถูกตัดทิ้งไปหรืออาจจะมีการป้อนกลับจาก output ไปยัง input ได้เรียกว่ามีลักษณะย้อนกลับ (Backward Learning) เพื่อส่งสัญญาณผิดพลาดกลับไปปรับแต่งเวต (weight) ให้ดีขึ้นเพื่อลดสัญญาณผิดพลาดที่เกิดขึ้นในการเรียนรู้กลุ่มอินพุตชุดหนึ่ง ๆ โดยโครงสร้างอย่างง่ายนี้สามารถฝึกสอนกลุ่มอินพุตที่มีลักษณะง่าย ๆ ได้แต่ถ้าต้องการนำไปประยุกต์ใช้กับอินพุตที่มีรูปร่างยาก ๆ หรือคล้าย ๆ กันต้องเพิ่มชั้นของโครงข่ายนิวรอนอีกเพื่อขยายเวต (weight) ให้มากขึ้นในการเรียนรู้

4.1.3 โครงสร้างของนิวรอนในกรณีที่มีลักษณะหลาย ๆ ชั้น (Multilayer Neural Network)

ลักษณะของโครงข่ายนิวรอนในลักษณะเพียง 1 ชั้น (Single Layer Neural Network) ถึงแม้ว่าจะทำงานได้ดีขึ้นจากนิวรอน (Neuron) เพียง 1 นิวรอน (Neuron) แต่ประสิทธิภาพในการทำงานจะเพิ่มมากขึ้นถ้าเราจัดรูปแบบโครงสร้างในลักษณะหลาย ๆ ชั้น (Multilayer) ในรูปที่ 4-7 แสดงลักษณะโครงข่ายนิวรอนในลักษณะ 2 ชั้น (Two Layer Neural Network)



รูปที่ 4-7 ลักษณะโครงข่ายนิวรอนในลักษณะ 2 ชั้น (Two-Layer Neural Network)

สำหรับประสิทธิภาพในการทำงานของโครงข่ายนิวรอนในลักษณะหลาย ๆ ชั้นนี้ (Multilayer Neural Network) จะไม่เพิ่มขึ้นหรือไม่แตกต่างจากโครงข่ายนิวรอนในลักษณะ 1 ชั้น (Single Layer Neural Network) เลย ถ้าในส่วนของฟังก์ชั่นการกระตุ้น (Activation Function) มี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ลักษณะเป็นสมการเชิงเส้น (Linear Function) ซึ่งเราสามารถแสดงให้เห็นได้โดยพิจารณาสมการดังนี้คือ

ถ้าให้ X เป็นอินพุตเวกเตอร์ (Input vector)

W_1 เป็นค่าน้ำหนัก (weight) ในรูปเมตริกซ์ (Matrix) ในชั้นที่ 1

W_2 เป็นค่าน้ำหนัก (weight) ในรูปเมตริกซ์ (Matrix) ในชั้นที่ 2

เพราะฉะนั้นค่า Output ที่ออกมาจะอยู่ในรูป

$$\text{Output} = (XW_1)W_2 \qquad (4.6)$$

เนื่องจากเราสามารถจัดกลุ่มการคูณได้ ดังนั้นอาจเขียนได้เป็น

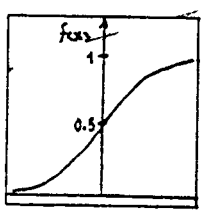
$$\text{Output} = X(W_1W_2) \qquad (4.7)$$

ซึ่งแสดงว่าลักษณะของโครงข่ายนิวรอนในลักษณะ 2 ชั้น (Two Layer Neural Network) จะเหมือนกับโครงข่ายนิวรอนในลักษณะ 1 ชั้น (Single Layer Neural Network) ที่มีค่าเมตริกซ์ (Matrix) ของค่าน้ำหนัก (weight) เท่ากับผลคูณของเมตริกซ์ (Matrix) ของค่าน้ำหนัก (weight) ในชั้น (layer) ที่ 1 คูณกับชั้น (Layer) ที่ 2

ดังนั้นในการเพิ่มประสิทธิภาพในการทำงานของโครงข่ายนิวรอนในลักษณะหลาย ๆ ชั้น (Multilayer Neural Network) เพิ่มจากโครงข่ายนิวรอนในลักษณะ 1 ชั้น (Single Layer Neural Network) จึงควรกำหนดฟังก์ชั่นการกระตุ้น (Activation Function) ในลักษณะไม่เชิงเส้น โดยในวิทยานิพนธ์ฉบับนี้ใช้ฟังก์ชั่นการกระตุ้นที่มีสมการของฟังก์ชั่น คือ

$$f(x) = 1/(1+\exp(-x)) \qquad (4.8)$$

ซึ่งมีรูปคือ



4.2 การฝึกสอน (Training) ของโครงข่ายนิวรอน

การฝึกสอน(Training)เป็นลักษณะที่จะพยายามสอนให้โครงข่ายนิวรอน (Neural Network) มีความรู้ในลักษณะที่เราต้องการ โดยในลักษณะการฝึกสอน (Training) ของโครงข่ายนิวรอน (Neural Network) จะเป็นการป้อนชุด input ให้แก่โครงข่ายนิวรอน (Neural Network)

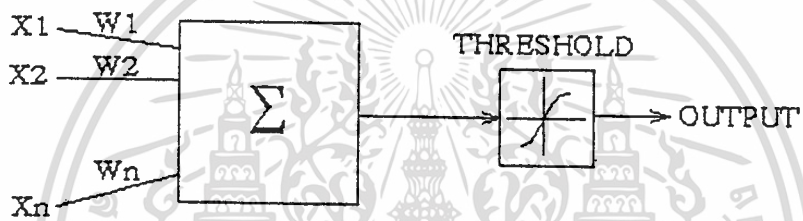
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และพยายามปรับค่าน้ำหนักของตัวโครงข่ายไปเรื่อย ๆ เพื่อให้มีค่าลู่เข้า (converge) ค่าของชุด output ที่เราต้องการให้มีค่าออกมา

โดยทั่วไปแล้ว เราจะเรียกวิธีการฝึกสอน (Training) เพื่อให้ตัวโครงข่ายสามารถเรียนรู้ (learning) เพื่อให้ได้ค่า output ที่ต้องการออกมาว่า Training Algorithm

4.2.1 เพอเซปตรอน (Perceptron)

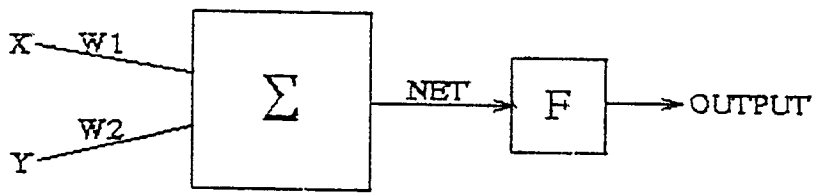
ลักษณะของนิวรอน (Neuron) ในยุคแรก ๆ ของโครงข่ายนิวรอน (Neural Network) มีชื่อเรียกว่าเพอเซปตรอน (Perceptron) ซึ่งมีลักษณะดังรูปที่ 4-8



รูปที่ 4-8 ลักษณะของเพอเซปตรอน (Perceptron)

ลักษณะโดยทั่วไปจะเหมือนกับนิวรอน (Neuron) ที่ได้กล่าวมาแล้ว แต่มีลักษณะเฉพาะคือในส่วนของฟังก์ชั่นการกระตุ้น (Activation Function) จะเป็นฟังก์ชั่น threshold ซึ่งจะให้ค่า 1 ออกมา ถ้าค่าผลรวมที่เข้ามามีค่ามากเกินกว่าค่าที่กำหนดค่าหนึ่งและจะให้ค่า 0 ออกมา ถ้าค่าผลรวมมีค่าน้อยกว่าค่าที่กำหนด

สำหรับการฝึกสอน (training) จะกระทำกันในลักษณะเพียง 1 ชั้น (Single Layer) ซึ่งประสบผลสำเร็จกันมากในช่วงแรก ๆ แต่ต่อมาได้ประสบปัญหาบางอย่างซึ่งลักษณะของเพอเซปตรอน(Perceptron) เพียง 1 ชั้น (Single Layer) ไม่สามารถแก้ปัญหาได้ ซึ่งปัญหาตัวอย่างที่ประสบปัญหากันซึ่งไม่สามารถแก้ปัญหาได้เป็นปัญหาเกี่ยวกับการเอกคลูซีฟออร์ (XOR problem) พิจารณาจากรูปที่ 4-9



รูปที่ 4-9 เพอเซปตรอน (Perceptron) ที่ใช้แก้ปัญหา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

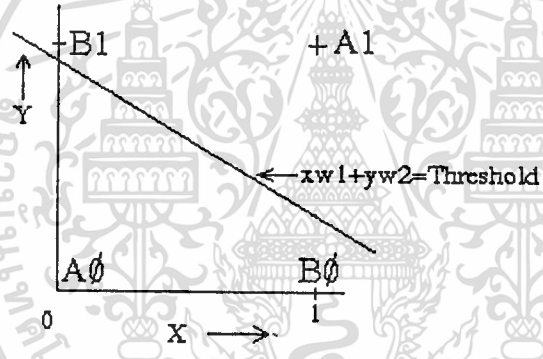
ถ้า x และ y เป็นอินพุตที่จะทำการ XOR เพราะฉะนั้นค่า NET ที่ออกมาจึงมีค่าเท่ากับ

$$NET = XW_1 + YW_2 \qquad (4.9)$$

ถ้าเรากำหนดค่า threshold ของฟังก์ชั่น F ให้มีค่า ๆ หนึ่งและกำหนดค่า W_1 และ W_2 เป็นค่า ๆ หนึ่งแล้วเราจะได้สมการเป็น

$$Threshold = XW_1 + YW_2 \qquad (4.10)$$

ซึ่งถ้าทำการ plot กราฟในพิกัด X-Y จะได้กราฟเส้นตรง ดังรูปที่ 4-10



รูปที่ 4-10 กราฟความสัมพันธ์ X-Y

พิจารณาตาราง XOR ในรูปที่ 5.9

Point	Xvalue	Yvalue	Desired Output
A0	0	0	0
B0	1	0	1
B1	0	1	1
A1	1	1	0

รูปที่ 4-11 ตารางเอ็กซ์คลูซีฟออ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ถ้าเราทำการ plot จุดต่างๆของตารางลงไปในกราฟความสัมพันธ์ X-Y จะได้ดังรูปที่ 4-10

ถ้าเราพิจารณาสมการที่ให้ค่า NET ออกมาใหม่อีกที่จะเห็นว่าค่า NET ที่มีค่ามากกว่าค่า threshold จะมีค่าอยู่ทางซีกใดซีกหนึ่งของเส้นตรงส่วนค่า NET ที่มีค่าน้อยกว่าจะอยู่ทางอีกซีกหนึ่ง

จากความจริงที่ว่าเพอเซปตรอน (Perceptron) นี้จะสามารถแก้ปัญหานี้ได้ ถ้าจุด A0 , A_1 อยู่ในซีกหนึ่งและจุด B0 , B_1 อยู่ในอีกซีกหนึ่ง โดยใช้เส้นตรงที่มีค่า NET เท่ากับค่า threshold เป็นตัวแบ่งแต่จากรูปไม่ว่าจะแบ่งด้วยสมการเส้นตรงรูปแบบใดก็ไม่สามารถที่จะแบ่งได้ ซึ่งหมายถึงไม่ว่าเราจะกำหนดค่า threshold , W1,W2 มีค่าเช่นไร ก็ไม่สามารถให้ค่า output ออกมาถูกต้องทุกค่า ซึ่งปัญหาเช่นนี้เราเรียกว่า ปัญหาการแบ่งแยกเชิงเส้น (linear separability problem) ซึ่งเกิดขึ้นมากมายหลายปัญหานอกจากปัญหาของการ XOR

ปัญหาของการ XOR นี้ สามารถแก้ปัญหาได้ โดยการใช้ลักษณะของการใช้เพอเซปตรอนหลาย ๆ ชั้น (Multilayer Perceptron) แต่วิธีการที่ให้ได้มาซึ่งค่าน้ำหนัก (weight) ของตัวโครงข่าย (Network) เป็นลักษณะของการกำหนดค่าโดยใช้ความคิดขึ้นมาเอง ไม่ใช่ลักษณะของการฝึกสอน (training) เนื่องมาจากในสมัยนั้น ยังไม่มีการคิดวิธีการฝึกสอน (training) ที่ได้ผลในส่วนของเพอเซปตรอนหลาย ๆ ชั้น (Multilayer Perceptron) ดังนั้นปัญหาเหล่านี้จึงเป็นปัญหาที่ยืดเยื้อมานานหลายปี สาเหตุเนื่องมาจากการขาดการฝึกสอน (training) ตัวโครงข่าย (network) ในลักษณะหลาย ๆ ชั้น (multilayer) นั่นเอง

กฎของเดลตา (The Delta Rule)

กฎที่สำคัญกฎหนึ่ง ซึ่งใช้ในการฝึกสอน (training) เพอเซปตรอน (perceptron) และต่อมาได้ถูกขยายเพื่อนำไปใช้ในการฝึกสอน (training) แก่โครงข่ายนิวรอน (Neural Network) ที่มีลักษณะหลายชั้น (multilayer) ด้วย มีชื่อเรียกว่า กฎของเดลตา (Dalta Rule) โดยกฎของเดลตา (Delta Rule) นี้จะถูกใช้ในการปรับน้ำหนัก (weight) ของเพอเซปตรอน (Perceptron) ได้อย่างสมเหตุสมผล , ลักษณะของกฎจะประกอบด้วยสูตรการคำนวณต่าง ๆ ดังนี้

$$\delta = (T - A) \qquad (4.11)$$

โดย T เป็นค่า output ที่เราต้องการให้เกิดผลออกมาจากเพอเซปตรอนนั้น (perceptron)

A เป็นค่า output ที่ออกมาจริงตามการป้อนอินพุตเข้าไป

δ เป็นค่าผลต่างระหว่างค่า 2 ค่าข้างบน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หลังจากได้ค่า δ แล้ว ค่านี้จะถูกนำไปคูณกับค่า input แต่ละ input เพื่อใช้ในการปรับค่าน้ำหนัก (weight) ที่เหมาะสมของแต่ละ input นอกจากนั้นยังต้องถูกนำไปคูณกับค่าอัตราการเรียนรู้ (learning rate) เพื่อเพิ่มประสิทธิภาพในการฝึกสอน (training) ให้เร็วขึ้นหรือช้าลง ซึ่งสามารถเขียนเป็นสูตรได้ดังนี้คือ

$$\Delta i = n \, \delta \, Xi \qquad (4.12)$$

ซึ่งค่า Δi หมายถึงค่าที่จะต้องถูกนำไปใช้ในการปรับค่าน้ำหนัก (weight) ของแต่ละค่าน้ำหนัก (weight)

ค่า Xi คือค่า input vector

n คืออัตราการเรียนรู้ (learning rate)

ดังนั้นสูตรการปรับค่าน้ำหนักสามารถเขียนได้ดังนี้ คือ

$$wi\,(n+1) = wi(n) + \Delta i \qquad (4.13)$$

โดย Δi เป็นค่าน้ำหนักที่ต้องปรับของแต่ละ input ของค่า input x ใด ๆ ในสมการ

wi(n+1)เป็นค่าน้ำหนัก หลังการปรับ

wi(n) เป็นค่าน้ำหนัก ก่อนการปรับ

4.2.2 โครงข่ายนิวรอนแบบแบคพรอบพาเกชั่นที่ใช้ในวิทยานิพนธ์ฉบับนี้ **(Backpropagation Neural Network Algorithm in this thesis)**

หลังจากประสบปัญหาเกี่ยวกับการฝึกสอนให้แก่โครงข่ายนิวรอนในลักษณะหลายๆ (multilayer) มาเป็นเวลานานต่อมาได้มีการคิดอัลกอลิทึม (Algorithm) หนึ่งขึ้นมา เพื่อฝึกสอน (training) ให้แก่โครงข่ายนิวรอน (Neural Network) ในลักษณะหลายๆชั้น (multilayer) ซึ่งได้ผลดีอย่างมากในการฝึกสอนโครงข่ายนิวรอน ซึ่งอัลกอลิทึม (Algorithm) นี้ ใช้กับโครงข่ายนิวรอน (Neural Network) ที่มีรูปแบบแน่ชัดตรงกับอัลกอลิทึม (Algorithms) ของโครงข่ายนิวรอนแบบนี้ว่าโครงข่ายนิวรอนแบบแบคพรอบพาเกชั่น (Backpropagation Neural Network)

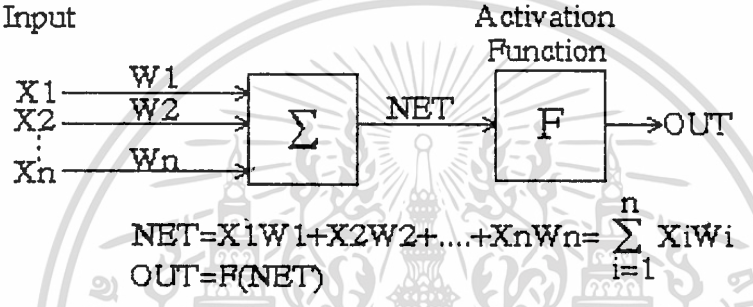
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.2.1 ลักษณะโครงสร้างของโครงข่าย (Network Configuration)

1. นิวรอน (The Neuron)

ลักษณะนิวรอน (Neuron) ของโครงข่ายแบบ Backpropagation (Backpropagation Network) มีลักษณะดังรูปที่ 4-12

โดยขั้นตอนแรกต้องทำการสร้างเวต (weight) จากการสุ่มเลขระหว่าง +0.001ถึง-0.001



รูปที่ 4-12 ลักษณะนิวรอนของโครงข่ายแบบแบคพรอบพาเกชั่น

และลักษณะทั่ว ๆ ไป จะคล้ายกับลักษณะทั่ว ๆ ไปของนิวรอนที่ได้เคยกล่าวมาแล้ว แต่ส่วนหนึ่งที่แตกต่างกันคือ เพิ่มฟังก์ชั่นการกระตุ้นที่มีลักษณะไม่เชิงเส้น (nonlinear function) ซึ่งสมการที่นิยมใช้กันสำหรับฟังก์ชั่นการกระตุ้น (Activation Function) นี้จะอยู่ในรูปแบบสมการคือ

$$OUT=F(NET)=1/(1+\exp(-NET)) \qquad (4.14)$$

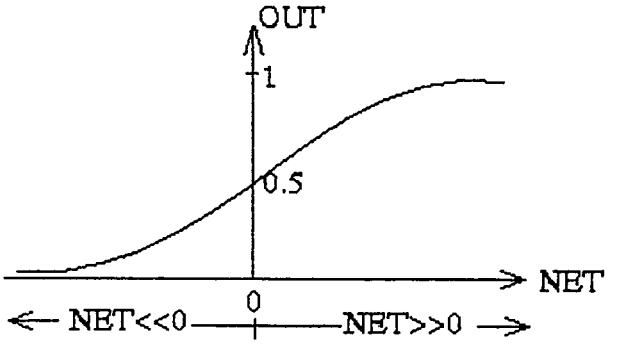
ซึ่งสมการนี้อธิบายได้ว่า ถ้าค่า NET = 0 แล้ว OUT = 0.5

ถ้าค่า NET >> 0 แล้ว OUT = 1

ถ้าค่า NET << 0 แล้ว OUT = 0

ซึ่งกราฟของสมการนี้สามารถดูได้ดังรูปที่ 4-13

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4-13 กราฟของฟังก์ชั่นการกระตุ้น (Activation Function) ที่มีสมการ

$$[OUT=1/(1+\exp(-NET))] \qquad (4.15)$$

ซึ่งฟังก์ชั่นลักษณะนี้เรามักเรียกว่า Sigmoid Function เนื่องมาจากเส้นโค้งของกราฟมีลักษณะคล้ายรูปตัว S ซึ่งค่า derivative ของฟังก์ชั่นนี้จะมีค่าดังนี้

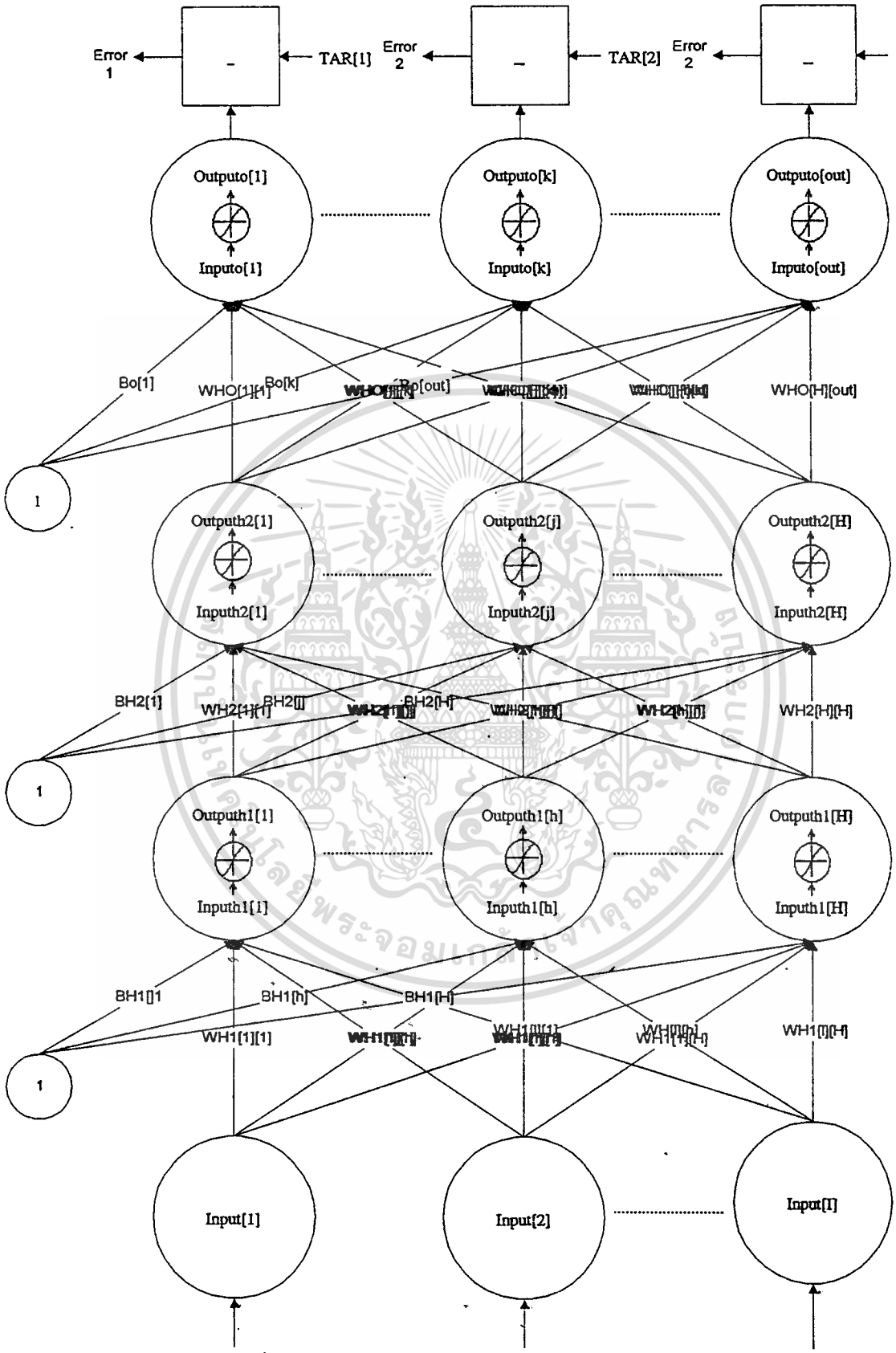
$$dOUT/dNET = OUT(1-OUT) \qquad (4.16)$$

ซึ่งค่านี้จะใช้ในการปรับแต่งน้ำหนัก(weight)โดยจะมีผลในการทำให้เกิดการลู่เข้าของค่า outputที่ต้องการได้เป็นอย่างดี ดังจะกล่าวถึงต่อไป

2. ลักษณะของโครงข่ายหลาย ๆ ชั้น (Multilayer Network)

ในรูปที่ 5.12 แสดงลักษณะของโครงข่ายนิวรอนแบบ แบคพรอบพาเกชั่น (Backpropagation Neural Network) ในลักษณะหลาย ๆ ชั้น (multilayer) ซึ่งประกอบขึ้นจากนิวรอน (Neuron) ที่ได้กล่าวข้างต้น มีส่วนเฉพาะที่เพิ่มเข้าในตอนท้ายของ output ซึ่งใช้ในการหาค่าความแตกต่างระหว่างค่า output ที่ต้องการกับค่า output ที่เกิดขึ้นจริง ซึ่งค่านี้จะถูกย้อนกลับเข้าไปในตัวโครงข่าย (wetwork)เพื่อใช้ในการปรับค่าน้ำหนัก (weight) ภายในตัวโครงข่าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4-14 โครงข่ายนิวรอนแบบแบคพรอบพาเกชั่น 2 ฮิดเด่นเลเยอร์ ที่ใช้ในงานวิจัยนี้ (OUT = 87 , H = 230 , I = 1,600)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คำนิยาม

Input [1,..., 1,600] = อินพุตเวกเตอร์ที่ใช้ในการฝึกสอนหรือป้อนให้แก่โครงข่าย

TAR [1,..., 87] = เอ้าท์พุตเป้าหมายที่ต้องการซึ่งเกิดจากการเซตด้วยตัวเราเองให้ตรงกับอินพุตเป้าหมายที่ต้องการ

deltao [K] = ตำแหน่งที่ใช้ในการปรับค่าผิดพลาดของเวตให้ถูกต้องต่อเวต WHO[j][K] เพื่อทำให้ผลของค่าผิดพลาด ที่เอาท์พุตยูนิต Outputo[K] และค่า deltao[K] นี้จะเป็นข้อมูลในการบอกค่าผิดพลาดกับขึ้นไปสู่ hidden units ซึ่งเป็นชั้นอยู่ก่อนหน้าที่ส่งข้อมูลมาสู่ยูนิต Inputo[K]

deltah2 [j] = สัดส่วนที่ใช้ในการปรับค่าผิดพลาดของเวตให้ถูกต้องสำหรับเวต WH2 [h] [j] ซึ่งทำให้เกิดค่าผิดพลาดที่เอาท์พุตยูนิต Outputh2 [j] และค่า deltah2 [j] นี้จะเป็นข้อมูลในการบอกค่าผิดพลาดกับขึ้นไปสู่เลเยอร์ที่อยู่ก่อนหน้าที่ส่งสัญญาณเข้ามาสู่ยูนิต Inputh2 [j]

deltah1 [h] = สัดส่วนที่ใช้ในการปรับค่าผิดพลาดของเวตให้ถูกต้องสำหรับเวต WH1[i][h] ซึ่งทำให้เกิดค่าผิดพลาดที่เอาท์พุตยูนิต Output1 [I] และค่า deltah1[h] นี้จะเป็นข้อมูลที่ใช้ในการบอกค่าผิดพลาดกับขึ้นไปสู่เลเยอร์ที่อยู่ก่อนหน้าที่ส่งสัญญาณเข้ามาสู่ยูนิต Inputh1[i]

n = อัตราการเรียนรู้ ซึ่งในการทำงานใช้ค่านี้เท่ากับ 0.1

alpha = โมเมนตัมของการปรับเวต ซึ่งในการทำงานใช้ค่านี้เท่ากับ 0.5

Input[i] = อินพุตยูนิต i

BH1[h] = ค่าไบแอสบนฮิดเด่นยูนิต ตัวที่ h บนฮิดเด่นเลเยอร์ ที่ 1

BH2 [j] = ค่าไบแอสบนฮิดเด่นยูนิต ตัวที่ j บนฮิดเด่นเลเยอร์ ที่ 2

BO [K] = ค่าไบแอสบนเอาท์พุตยูนิต ตัวที่ k บนเอาท์พุตเลเยอร์

Inputh2 [j] = อินพุตยูนิต j บนฮิดเด่นเลเยอร์ที่ 2

ค่าผลรวมอินพุตของ Inputh2 [j] ซึ่งมีสมการในการคำนวณ คือ

$$\text{Inputh2 [j]} = \text{BH2[j]} + \sum_{i}^{H} Outputh1[i] * WH2[i][j] \qquad (4.17)$$

ค่าเอาท์พุตที่ออกจาก Inputh2[j] เข้าสู่ Activation Function คือ

$$\text{Outputh2 [j]} = 1.0/(1.0+ \exp(-\text{Inputh2[j]})) \qquad (4.18)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Inputh1[h] = อินพุตยูนิต h บนฮิดเด่นเลเยอร์ที่ 1

ค่าผลรวมอินพุตของ Input1 [h] ซึ่งมีสมการในการคำนวณคือ

$$\text{Inputh1[h]} = \text{BH1[h]} + \sum_{i}^{I} Input[i] * WH1[i][h] \qquad (4.19)$$

ค่าเอาท์พุตที่ออกจาก Inputh1[h] เข้าสู่ Activation Function คือ

$$\text{Outputh1[h]} = 1.0/(1.0+\exp(-\text{Inputh1[h]})) \qquad (4.20)$$

Outputo[K] = เอาท์พุตยูนิต K บนเอาท์พุตเลเยอร์

ค่าผลรวมอินพุตของ Inputo [K] ซึ่งมีสมการในการคำนวณคือ

$$\text{Inputo[K]} = \text{BO[K]} + \sum_{i}^{H} Outputh2[i] * WHO[i][k] \qquad (4.21)$$

ค่าเอาท์พุตที่ออกจาก Inputo[K] เข้าสู่ Activation Function คือ

$$\text{Outputo[K]} = 1.0/(1.0+\exp(-\text{Inputo[K]})) \qquad (4.22)$$

อัลกอลิทึมของการฝึกสอน

<u>ขั้นตอนที่ 0</u> เริ่มกำหนดค่าตัวแปรเวต (weight) ทั้งหมดเป็นเลขระหว่าง -0.001ถึง +0.001

<u>ขั้นตอนที่ 1</u> ทำขั้นตอนที่ 2 ถึง 13 ขณะที่ข้อความนี้เป็นจริง

While((errorb >0.005) && ([errorb(new)-errorb(old)] > 0.0002 (4.23)

<u>ขั้นตอนที่ 2</u> สำหรับทุกคู่ของการฝึกสอนให้แก่โครงข่ายนิวรอนให้ทำ step 3 ถึง 11

กระบวนการดำเนินงานไปข้างหน้า (Forward Phase)

<u>ขั้นตอนที่ 3</u> สำหรับแต่ละยูนิตอินพุต (Input[1 to 1600] ให้รับสัญญาณอินพุตInput[i] และกระจายสัญญาณไปสู่ยูนิตในชั้นบนซึ่งเป็นยูนิตฮิดเด่นเลเยอร์ในฮิดเด่นเลเยอร์ชั้นที่ 1

<u>ขั้นตอนที่ 4</u> สำหรับแต่ละฮิดเด่นยูนิต (Inputh1[h]) ให้หาผลรวมของการคูณกันของคู่ weight และอินพุตที่เข้ามา โดยมีสมการคือ

$$\text{Inputh1[h]} = \text{BH1[h]} + \sum_{i}^{1600} Input[i] * WH1[i][h] \qquad (4.24)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากนั้นทำการนำผลลัพธ์ Inputh1[h] เข้าสู่ฟังก์ชั่นแอคติเวชั่นเพื่อหาสัญญาณเอาท์พุต Outputh1[h] ซึ่งมีสมการคือ

Outputh1[h] = 1.0/(1.0+exp(-Inputh1[h])) (4.25)

แล้วนำสัญญาณ Outputh1[h] กระจายไปทุกยูนิตของเลเยอร์บนซึ่งคือทุก ๆ ยูนิตบน hidden Unit ใน layer2

ขั้นตอนที่ 5 สำหรับแต่ละยูนิตบนฮิดเด่นชั้นที่ 2 (Inputh2[j]) ให้หาผลรวมของการคูณกันของคู่ weight และอินพุตที่ส่งสัญญาณเข้ามาหาโดยมีสมการในการหาคือ

$$\text{Inputh2[j]} = \text{BH2[j]} + \sum_{i}^{H} Inputh1[i] * WH2[i][j] \quad (4.26)$$

จากนั้นทำการนำผลลัพธ์ Inputh2[j] เข้าสู่ฟังก์ชั่นแอคติเวชั่นเพื่อหาสัญญาณเอาท์พุต Outputh2[j] ซึ่งมีสมการคือ

Outputh2[j] = 1.0/(1 + exp (-Inputh2[j])) (4.27)

แล้วนำสัญญาณ Outputh2[j] กระจายไปทุกยูนิตของเลเยอร์บนซึ่งคือทุก ๆ ยูนิตบน Output Layer

ขั้นตอนที่ 6 สำหรับแต่ละยูนิตบน Output เลเยอร์ (Out=1, ... , 87) เราสามารถหาอินพุตของ Output โหนดแต่ละตัว (Inputo[OUT] ได้ตามสมการ

$$\text{Inputo[K]} = \text{BO[K]} + \sum_{j=1}^{H} Outputh2[j] * WO[j][k] \quad (4.28)$$

และจากนั้นทำการหาผลลัพธ์เข้าสู่ฟังก์ชั่นแอคติเวชั่นเพื่อหาสัญญาณเอาท์พุต Outputo[K] ซึ่งมีสมการคือ

Outputo[K] = 1.0 /(1.0+exp(-Inputo[K])) (4.29)

ขั้นตอนที่ 7 $$\text{errorp} = (1/2) * \sum_{i}^{OUT} (TAR[i] - OUTPUTO[i])^2 \quad (4.30)$$

กระบวนการย้อนกลับเพื่อหาค่าปรับสัญญาณผิดพลาด (Backward or Backpropagation of error)

ขั้นตอนที่ 8 สำหรับแต่ละเอาท์พุตยูนิต (Outputo[K] , k=1, ..., OUT) รับค่าเป้าหมายของแต่ละอินพุตแพตเทิร์นมาเพื่อคำนวณหาค่าผิดพลาด (deltao[K]) ซึ่งมีสมการในการคำนวณคือ

delao[K] = (TAR[K]-Outputo[K]) (1-Outputo[K])Outputo[K] (4.31)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทำการคำนวณค่าเวตที่ต้องปรับปรุงเพื่อปรับค่าเวต WHO[j][K] ภายหลัง , ซึ่งสมการของเวตที่ต้องปรับปรุงคือ ΔWHO [j][K]

$$\Delta WHO[j][K]_{new} = (n*deltao[K]*Outputh2[j])+(alpha*\Delta WHO[j][K]_{old})$$

ทำการคำนวณค่าไบแอส (Bias) ที่ต้องปรับปรุงเพื่อปรับค่า BO[K] ในภายหลัง

$$\Delta BO[k]_{new} = (n*deltaO[K]) + (alpha*BO[K]_{old}) \quad (4.32)$$

และทำการส่งสัญญาณ deltao[K] ไปสู่ยูนิตในเลเยอร์ทุกตัวที่ต่ำกว่า 1 เลเยอร์

ขั้นตอนที่ 9 สำหรับแต่ละยิดเด่นยูนิตในเลเยอร์ที่ 2 (Inputh2[j] , j = 1 , ... (H = 200)) หาผลรวมของสัญญาณ delta ซึ่งส่งมาจากเลเยอร์ข้างบน ซึ่งมีสมการคือ

$$deltah2[j] = [\sum_{k=1}^{OUT} (deltao[K] * WHO[j][k])]*(1-Outputh2[j])*Outputh2[j] \quad (4.33)$$

ทำการคำนวณค่าเวตที่ต้องปรับปรุงเพื่อใช้ในการปรับเวต (WH2[h][j]

$$\Delta WH2[h][j]_{new} = (n*deltah2[j]*Outputh1[h])+(alpha*\Delta WH2[h][j]_{old})$$

ทำการคำนวณค่าไบแอส (Bias) ที่ต้องปรับปรุงเพื่อปรับค่า BH2[j] ให้ถูกต้องขึ้น

$$\Delta BH2[j]_{new} = (n*deltah2[j]) + (alpha*BH2[j]_{old}) \quad (4.34)$$

และทำการส่งสัญญาณ deltah2[j] ไปสู่ยูนิตในเลเยอร์ที่ต่ำกว่าทุก node

ขั้นตอนที่ 10 สำหรับแต่ละยิดเด่นยูนิตในเลเยอร์ที่ 1 (Inputh1[h] , h=1,... , (H=200) ทำการหาผลรวมของสัญญาณ delta ซึ่งส่งมาจากเลเยอร์ข้างบนซึ่งมีสมการคือ

$$deltah1[h]=[\sum_{j=1}^{H}(deltah2[j]*WH2[h][j])](1-Outputh1(h))*Outputh1[h] \quad (4.35)$$

ทำการคำนวณค่าเวตที่ต้องปรับปรุงเพื่อทำการปรับเวต (WH1[i][h]) ซึ่งมีสมการ

$$\Delta WH1[i][h]_{new}=(n*deltah1[h]*Input[i])+(alpha*\Delta WH1\ [i][h]_{old}) \quad (4.36)$$

ทำการคำนวณค่าไบแอส (Bias) ที่ต้องปรับปรุงเพื่อปรับค่า BH1[h] ให้ถูกต้องขึ้น

$$\Delta BH1[h]_{new}=(n*deltah1[h])+(alpha*\Delta BH1[h]_{old}) \quad (4.37)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และทำการส่งสัญญาณ deltah1[h] ไปสู่ยูนิตในเลเยอร์ที่ต่ำกว่าทุก node

ขั้นตอนการปรับ weight และ Bias

ขั้นตอนที่ 11 สำหรับแต่ละ output ยูนิต Outputo[K] (K=1, ..., OUT) ทำการปรับค่า bias และค่าเวต (weight) ทุก ๆ โหนด (j = 1,....., (H=200))

$$WHO[j][K]_{new} = WHO[j][K]_{old} + \Delta WHO[j][K] \quad (4.38)$$

$$BO[K]_{new} = BO[K]_{old} + \Delta BO[K] \quad (4.39)$$

สำหรับแต่ละ hidden ยูนิต ในเลเยอร์ที่ 2 Outputh2[j] (j=1, ..., H) ทำการปรับค่า bias และ weights (h=1, ..., H)

$$WH2[h][j]_{new} = WH2[h][j]_{old} + \Delta WH2[h][j] \quad (4.40)$$

$$BO2[j]_{new} = BH2[j]_{old} + \Delta BH2[j] \quad (4.41)$$

สำหรับแต่ละอิดเด่นยูนิตในเลเยอร์ที่ 1 (Outputh1[h] , (h=1, ...,H)) ทำการปรับค่า bias และค่าเวต (weight) โดยสมการ

$$WH1[i][h]_{new} = WH1[i][h]_{old} + \Delta WH1[i][h] \quad (4.42)$$

$$BH1[h]_{new} = BH1[h]_{old} + \Delta BH1[h] \quad (4.43)$$

เมื่อ (i=1, ..., I)

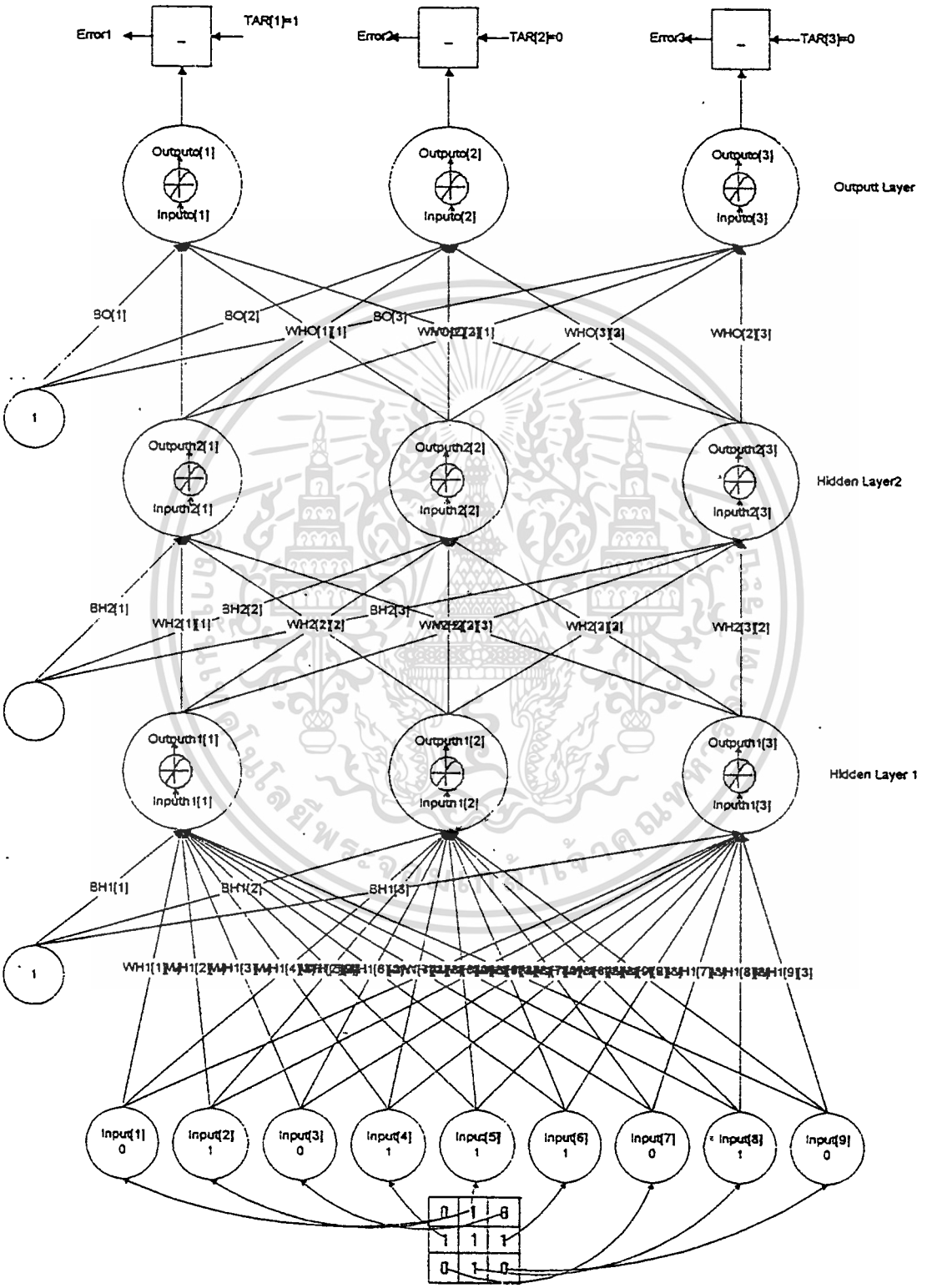
ขั้นตอนที่ 12 อ่านอินพุตจากไฟล์ชุดฝึกสอนจนกว่าจบไฟล์ แล้วทำการปรับค่า

$$errorb = errorb + errorp \quad (4.44)$$

ขั้นตอนที่ 13 ทดสอบเงื่อนไขการสิ้นสุดการฝึกสอนคือถ้า (errorb > 0.05) และ ($errorb_{new}$ - $errorb_{old}$) > 0.0002 เป็นเท็จอันใดอันหนึ่งให้หยุดการฝึกสอนโครงข่ายนิวรอน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่างการคำนวณของโครงข่ายนิวรอนแบบแบคพรอบพาเกชั่น (ซึ่งประกอบด้วย 2 Hidden Layer , 1 Input Layer , 1 Output Layer ขณะทำการป้อนอินพุตเป้าหมายเท่ากับ 1



เมื่ออินพุตที่ป้อนมีค่าเป้าหมายคือ TAR[1]=1

รูปที่ 4.15 ตัวอย่างโครงข่ายนิวรอนขณะทำการป้อนอินพุตและค่าเป้าหมาย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นตอนการคำนวณขณะทำการฝึกสอนอินพุตเป้าหมาย 1 ให้แก่โครงข่ายนิวรอนด้วยวิธีแบคพรอบพาเกชั่น

ขั้นตอนที่0 เริ่มกำหนดค่าตัวแปรเวต(weight) และค่าbiasทั้งหมดเป็นเลข +0.1,-0.1

ขั้นตอนที่ 1 ทำขั้นตอนที่ 2 ถึง 13 ขณะข้อความนี้เป็นจริง

While((errorb>0.05)&&((errorb(new)-errorb(old))>0.0002))

ขั้นตอนที่ 2 สำหรับทุกคู่ของการฝึกสอนให้โครงข่ายนิวรอนดำเนินการขั้นตอนที่ 3 ถึง 11

กระบวนการดำเนินการคำนวณไปข้างหน้า (Forward Phase)

ขั้นตอนที่ 3 Input[1] =0 , Input[2] = 1 , Input[3] =0 , Input[4] = 1 , Input[5] = 1

Input[6] = 1 , Input[7] =0 , Input[8] = 1 , Input[9] =0

ขั้นตอนที่ 4 $\text{Inputh1}[1] = 0.1 + \sum_{j=1}^{9}((0*0.1)+(0.1*1)+(0*1.0)+(1*0.1)+(1*0.1)+(1*0.1)+(0*0.1)+(1*0.1)+(0*0.1))$

$= 0.6$

$\text{Inputh1}[2] = 0.1 + \sum_{j=1}^{9}((0*0.1)+(1*0.1)+(0*1.0)+(1*0.1)+(1*0.1)+(1*0.1)+(0*0.1)+(1*0.1)+(0*0.1))$

$= 0.6$

$\text{Inputh1}[3] = 0.1 + \sum_{j=1}^{9}((0*0.1)+(1*0.1)+(0*1.0)+(1*0.1)+(1*0.1)+(1*0.1)+(0*0.1)+(1*0.1)+(0*0.1))$

$= 0.6$

คำนวณค่า Outputh1[1] , Output1[2] , Output1[3] จากการผ่านแอคติเวชั่นฟังก์ชั่น

Outputh1[1] = 1.0/(1+exp(-0.6)) = 0.6456

Outputh1[2] = 1.0/(1+exp(-0.6)) = 0.6456

Outputh1[3] = 1.0/(1+exp(-0.6)) = 0.6456

แล้วนำสัญญาณ Outputh1[1->3] กระจายไปทุกยูนิตของเลเยอร์บน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นตอนที่ 5 $\text{Inputh2[1]} = 0.1 + \sum_{i=1}^{3} [(0.1x0.6456) + (0.1x0.6456) + (0.1x0.6456)]$

$= 0.29368$

$\text{Inputh2[2]} = 0.1 + \sum_{i=1}^{3} [(0.1x0.6456) + (0.1x0.6456) + (0.1x0.6456)]$

$= 0.29368$

$\text{Inputh2[3]} = 0.29368$

คำนวณค่า Outputh2[1] , Outputh[2] , Outputh2[3] จากการผ่านแอคติเวชั่น

$\text{Outputh2[1]} = 1.0/(1+\exp(-0.29368)) = 0.5728$

$\text{Outputh2[2]} = 1.0/(1+\exp(-0.29368)) = 0.5728$

$\text{Outputh2[3]} = 1.0/(1+\exp(-0.29368)) = 0.5728$

ขั้นตอนที่ 6 สำหรับแต่ละยูนิตบน Output เลเยอร์ (OUT=1,2,3) ทำการหา

Inputo(1,2,3)

$\text{Inputo[1]} = 0.1+\sum_{i=1}^{3}[(0.1x0.5728)+(0.1x0.5728)+(0.1x0.5728)] = 0.27$

$\text{Inputo[2]} = 0.1+\sum_{i=1}^{3}[(0.1x0.5728)+(0.1x0.5728)+(0.1x0.5728)] = 0.27$

$\text{Inputo[3]} = 0.1+\sum_{i=1}^{3}[(0.1x0.5728)+(0.1x0.5728)+(0.1x0.5728)] = 0.27$

คำนวณค่า Outputo[1] , Outputo[2] , Outputo[3] จากการผ่านแอคติเวชั่นฟังก์ชั่น

$\text{Outputo[1]} = 1.0/(1+\exp(-0.27)) = 0.567$

$\text{Outputo[2]} = 1.0/(1+\exp(-0.27)) = 0.567$

$\text{Outputo[3]} = 1.0/(1+\exp(-0.27)) = 0.567$

ขั้นตอนที่ 7 $\text{errorp} = 1/2*\sum_{i=1}^{3}(\text{TAR[i]} - \text{Outputo[i]})^2$

$= 1/2[(1-0.567)^2 + (0-0.567)^2 + (0-0.567)^2] = 0.41523$

กระบวนการย้อนกลับเพื่อหาตัวปรับสัญญาณผิดพลาด (Backward or Backpropagation of error)

ขั้นตอนที่ 8 ทำการคำนวณค่า deltao[1] , deltao[2] , deltao[3] เพื่อคำนวณตัวปรับค่าผิดพลาด

$\text{deltao[1]} = (\text{TAR[1]}-\text{Outputo[1]})\ (1-\text{Outputo[1]})\ \text{Outputo[1]}$

$= (1-0.567)(1-0.567)(0.567) = 0.491022$

$\text{deltao[2]} = (\text{TAR[2]}-\text{Outputo[2]})\ (1-\text{Outputo[2]})\ \text{Outputo[2]}$

$= (0-0.567)(1-0.567)(0.567) = -0.1392$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

deltao[3] = (TAR[3]-Outputo[3]) (1-Outputo[3]) Outputo[3]

= (0-0.567)(1-0.567)(0.567) = -0.1392

ทำการคำนวณค่าเวตที่ต้องปรับปรุงเพื่อปรับค่าเวต WHO[1,2,3][1,2,3]

ΔWHO[1][1]=(n*deltao[1]*Outputh2[1])+(alpha*ΔWHO[1][1])

=(0.1x0.491022*0.5728)+(0.5*0) = 0.0281

ΔWHO[1][2] =(n*deltao[2]*Outputh2[1])+(alpha*ΔWHO[1][2])

= (0.1x0.1392*0.5728)+(0.5*0) = $-7.97x10^{-3}$

ΔWHO[1][3] = (n*deltao[3]*Outputh2[1])+(alpha*ΔWHO[1][3])

= (0.1*-0.1392*0.5728)+(0.5*0) = $-7.97x10^{-3}$

ΔWHO[2][1] =(n*deltao[1]*Outputh2[2])+(alpha*ΔWHO[2][1])

= (0.1x*0.491*0.5728)+(0.5*0) = 0.0281

ΔWHO[2][2] =(n*deltao[2]*Outputh2[2])+(alpha*ΔWHO[2][2])

= (0.1x-0.1392x0.5728)+(0.5*0) = $-7.97x10^{-3}$

ΔWHO[2][3] =(n*deltao[3]*Outputh2[2])+(alpha*ΔWHO[2][3])

= (0.1*-0.1392x0.5728)+(0.5*0) = $-7.97x10^{-3}$

ΔWHO[3][1] =(n*deltao[1]*Outputh2[3])+(alpha*ΔWHO[3][1])

= (0.1*0.491*0.5728)+(0.5*0) = 0.0281

ΔWHO[3][2] =(n*deltao[2]*Outputh2[3])+(alpha*ΔWHO[3][2])

= (0.1*-0.1392x0.5728)+(0.5*0) = $-7.97x10^{-3}$

ΔWHO[3][3] = (n*deltao[3]*Outputh2[3])+(alpha*ΔWHO[3][3])

= (0.1*-0.1392x0.5728)+(0.5*0) = $-7.97x10^{-3}$

ทำการคำนวณค่าไบแอส(Bias) ที่ต้องทำการปรับปรุงเพื่อปรับค่า BO[1],BO[2],BO[3]

$\Delta BO[1]_{new}$ = (n*deltao[1])+(alpha*$\Delta BO[1]_{old}$)

= (0.1*0.491)+(0.5*0) = 0.0491

$\Delta BO[2]_{new}$ = (0.1*-0.1392)+(0.5*0) = -0.01392

$\Delta BO[3]_{new}$ = (0.1*-0.1392)+(0.5*0) = -0.01392

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทำการส่งสัญญาน deltao[1],deltao[2],deltao[3]ไปสู่ยูนิตในเลเยอร์ทุกตัวที่ต่ำกว่า1เลเยอร์

ขั้นตอนที่ 9 ทำการคำนวณค่า deltah2[1] , deltah2[2] , deltah2[3] เพื่อคำนวณตัวปรับค่า

$$deltah2[1]=[\sum_{K=1}^{3}(deltao[j]xWHO[1][k])](1-Outputh2[1])Outputh2[1]]$$

$$= [(0.491x0.1)+(-0.1392x0.1)+(-0.1392x0.1)]*$$

$$(1-0.5728)(0.5728) = 5.2x10^{-3}$$

$$deltah2[2]=[\sum_{K=1}^{3}(deltao[k]xWHO[2][k])](1-Outputh2[2])\ Outputh2[2]$$

$$= [(0.491x0.1)+(-0.1392x0.1)+(-0.1392x0.1)]*$$

$$(1-0.5728)(0.5728) = 5.2x10^{-3}$$

$$deltah2[3]=[\sum_{K=1}^{3}(deltao[k]xWHO[1][k])](1-Outputh2[3])\ Outputh2[3]$$

$$= [(0.491x0.1)+(-0.1392x0.1)+(-0.1392x0.1)]*$$

$$(1-0.5728)(0.5728) = 5.2x10^{-3}$$

ทำการคำนวณค่าเวตที่ต้องปรับปรุงเพื่อปรับค่าเวต WH2[1->3][1->3]

$$\Delta WH1[1][1] =(nxdeltah2[1]xOutputh1[1] + (alpha\ x\Delta WH2[1][1])$$

$$= (0.1x0.0052x0.6456) + (0.5x0) =3.35*10^{-4}$$

$$\Delta WH1[1][2] =(nxdeltah2[2]xOutputh1[1] + (alpha\ x\Delta WH2[1][2])$$

$$= (0.1x0.0052x0.6456) + (0.5x0) =3.35*10^{-4}$$

$$\Delta WH1[1][3] =(nxdeltah2[3]xOutputh1[1] + (alpha\ x\Delta WH2[1][3])$$

$$= (0.1x0.0052x0.6456) + (0.5x0) =3.35*10^{-4}$$

$$\Delta WH1[2][1] =(nxdeltah2[1]xOutputh1[2] + (alpha\ x\Delta WH2[2][1])$$

$$= (0.1x0.0052x0.6456) + (0.5x0) =3.35*10^{-4}$$

$$\Delta WH1[2][2] =(nxdeltah2[2]xOutputh1[2] + (alpha\ x\Delta WH2[2][2])$$

$$= (0.1x0.0052x0.6456) + (0.5x0) =3.35*10^{-4}$$

$$\Delta WH1[2][3] =(nxdeltah2[3]xOutputh1[2] + (alpha\ x\Delta WH2[2][3])$$

$$= (0.1x0.0052x0.6456) + (0.5x0) =3.35*10^{-4}$$

$$\Delta WH1[3][1] =(nxdeltah2[1]xOutputh1[3] + (alpha\ x\Delta WH2[3][1])$$

$$= (0.1x0.0052x0.6456) + (0.5x0) =3.35*10^{-4}$$

$$\Delta WH1[3][2] =(nxdeltah2[2]xOutputh1[3] + (alpha\ x\Delta WH2[3][2])$$

$$= (0.1x0.0052x0.6456) + (0.5x0) =3.35*10^{-4}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$\Delta WH1[3][3] = (nxdeltah2[3]xOutputh1[3] + (alpha\ x\Delta WH2[3][3])$$

$$= (0.1x0.0052x0.6456) + (0.5x0) = 3.35*10^{-4}$$

ทำการคำนวณค่าไบแอส (Bias ที่ต้องปรับปรุงเพื่อปรับค่า BH2[1] , BH2[2] , BH2[3] ให้ถูกต้องขึ้น

$$\Delta BH2[1]_{new} = (\ n\ x\ deltah2[1])+(alpha*\Delta BH2[1]_{old})$$

$$= (0.1x0.0052)+(0.5x0)$$

$$= 5.2*10^{-4}$$

$$\Delta BH2[2]_{new} = (\ n\ x\ deltah2[2])+(alpha*\Delta BH2[2]_{old})$$

$$= (0.1*0.0052)+(0.5*0)$$

$$= 5.2*10^{-4}$$

$$\Delta BH2[3]_{new} = 5.2*10^{-4}$$

<u>ขั้นตอนที่ 10</u> ทำการคำนวณค่า deltah1[1] , deltah1[2] , deltah1[3] เพื่อคำนวณตัวปรับค่า

$$deltah1[1] = [\sum_{j=1}^{3} (deltah2[j]\ x\ WH2[1][j])]\ (1-Outputh1[1]$$

$$= [(0.0052x0.1)+(0.0052x0.1)+(0.0052x0.1)]*$$

$$(1-0.6456) = 5.52x10^{-4}$$

$$deltah1[2] = [\sum_{j=1}^{3} (deltah2[j]\ x\ WH2[2][j])]\ (1-Outputh2[2]$$

$$= [(0.0052x0.1)+(0.0052x0.1)+(0.0052x0.1)]*$$

$$(1-0.6456) = 5.52x10^{-4}$$

$$deltah1[3] = [\sum_{j=1}^{3} (deltah2[j]\ x\ WH2[3][j])]\ (1-Outputh2[3]$$

$$= [(0.0052x0.1)+(0.0052x0.1)+(0.0052x0.1)]*$$

$$(1-0.6456) = 5.52x10^{-4}$$

ทำการคำนวณค่าเวตที่ต้องปรับปรุงเพื่อปรับค่าเวต WH1[1->3][1->3]

$$\Delta WH1[1][1] = (nxdeltah1[1]\ x\ Input[1] + (alpha\ x\Delta WH1[1][1])$$

$$= (0.1x0.00052x0) + (0.5x0) = 0$$

$$\Delta WH1[1][2] = (nxdeltah1[2]\ x\ Input[1] + (alpha\ x\Delta WH1[1][2])$$

$$= (0.1x0.00052x0) + (0.5x0) = 0$$

$$\Delta WH1[1][3] = (nxdeltah1[3]\ x\ Input[1] + (alpha\ x\Delta WH1[1][3])$$

$$= (0.1x0.00052x0)+(0.5x0)=0$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$\Delta WH1[2][1] = (nxdeltah1[1] \times Input[2] + (alpha \times \Delta WH1[2][1])$

$= (0.1x0.00052x1) + (0.5x0) = 0.000052$

$\Delta WH1[2][2] = (nxdeltah1[2] \times Input[2] + (alpha \times \Delta WH1[2][2])$

$= (0.1x0.00052x1) = 0.000052$

$\Delta WH1[2][3] = (nxdeltah1[3] \times Input[2] + (alpha \times \Delta WH1[2][3])$

$= 0.000052$

$\Delta WH1[3][1] = (nxdeltah1[1] \times Input[3] + (alpha \times \Delta WH1[3][1])$

$= (0.1x0.000052x0) = 0$

$\Delta WH1[3][2] = (nxdeltah1[2] \times Input[3] + (alpha \times \Delta WH1[3][2])$

$= (0.1x0.000052x0) + (0.5x0) = 0$

$\Delta WH1[3][3] = (nxdeltah1[3] \times Input[3] + (alpha \times \Delta WH1[3][3])$

$\Delta WH1[4][1] = (nxdeltah1[1] \times Input[4] + (alpha \times \Delta WH1[4][1])$

$= (0.1x0.00052x1) = 0.000052$

$\Delta WH1[4][2] = (nxdeltah1[2] \times Input[4] + (alpha \times \Delta WH1[4][1])$

$= (0.1x0.00052x1) = 5.2x10^{-5}$

$\Delta WH1[4][3] = (nxdeltah1[3] \times Input[4] + (alpha \times \Delta WH1[4][3])$

$= 5.2x10^{-5}$

$\Delta WH1[5][1] = (nxdeltah1[1] \times Input[5] + (alpha \times \Delta WH1[5][1])$

$= 5.2x10^{-5}$

$\Delta WH1[5][2] = (nxdeltah1[2] \times Input[5] + (alpha \times \Delta WH1[5][2])$

$= 5.2x10^{-5}$

$\Delta WH1[5][3] = (nxdeltah1[3] \times Input[5] + (alpha \times \Delta WH1[5][3])$

$= 5.2x10^{-5}$

$\Delta WH1[6][1] = (nxdeltah1[1] \times Input[6] + (alpha \times \Delta WH1[6][1])$

$= 5.2x10^{-5}$

$\Delta WH1[6][2] = (nxdeltah1[2] \times Input[6] + (alpha \times \Delta WH1[6][2])$

$= 5.2x10^{-5}$

$\Delta WH1[6][3] = (nxdeltah1[3] \times Input[6] + (alpha \times \Delta WH1[6][3])$

$= 5.2x10^{-5}$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$\Delta WH1[7][1] = (nxdeltah1[1] \times Input[7] + (alpha \times \Delta WH1[7][1])$

$= 0$

$\Delta WH1[7][2] = (nxdeltah1[2] \times Input[7] + (alpha \times \Delta WH1[7][2])$

$= 0$

$\Delta WH1[7][3] = (nxdeltah1[3] \times Input[7] + (alpha \times \Delta WH1[7][3])$

$= 0$

$\Delta WH1[8][1] = (nxdeltah1[1] \times Input[8] + (alpha \times \Delta WH1[8][1])$

$= 5.2x10^{-5}$

$\Delta WH1[8][2] = (nxdeltah1[2] \times Input[8] + (alpha \times \Delta WH1[8][2])$

$= 5.2x10^{-5}$

$\Delta WH1[8][3] = (nxdeltah1[3] \times Input[8] + (alpha \times \Delta WH1[8][3])$

$= 5.2x10^{-5}$

$\Delta WH1[9][1] = (nxdeltah1[3] \times Input[9] + (alpha \times \Delta WH1[9][1])$

$= 0$

$\Delta WH1[9][2] = (n \times deltah1[2] \times Input[9] + (alpha \times \Delta WH1[9][2])$

$= 0$

$\Delta WH1[9][3] = (n \times deltah1[3] \times Input[9] + (alpha \times \Delta WH1[9][3])$

$= 0$

ทำการคำนวณค่าไบแอส (Bias ที่ต้องปรับปรุงเพื่อปรับค่า BH1(1) , BH1(2) , BH1[3] ให้ถูกต้องขึ้น

$\Delta BH1[1]_{new} = (n \times deltah1[1]) + (alpha \times \Delta BH1[1]_{old})$

$= (0.1x0.00052) + (0.5x\ 0) = 0.000052$

$\Delta BH1[2]_{new} = (0.1 \times 0.00052) + (0.5x0\ ) = 0.000052$

$\Delta BH1[3]_{new} = (nxdeltah1[3]) + (alpha \times \Delta BH1\ [3]_{old}) = 0.000052$

และทำการส่งสัญญาณผิดพลาด deltah1[1] , deltah1[2] , deltah1[3] ไปสู่ยูนิตที่อยู่ในเลเยอร์ที่ต่ำกว่าทุก node

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นตอนการปรับเวต (weight) และ Bias

ขั้นตอนที่ 11 สำหรับแต่ละ output ยูนิต Outputo[K] , (K=1,2,3) ทำการปรับค่า bias และค่าเวต (weight) ทุก ๆ โหนด (j=1,2,3,)

$WHO[1][1] = WHO[1][1]_{old} + \Delta WHO[1][1] = 0.1 + 0.0281 = 0.1281$

$WHO[1][2] = 0.1 + (-7.79 \times 10^{-3}) = 0.092$

$WHO[1][3] = 0.1 + (-7.79 \times 10^{-3}) = 0.092$

$WHO[2][1] = 0.1 + (0.0281) = 0.1281$

$WHO[2][2] = 0.1 + (-7.79 \times 10^{-3}) = 0.092$

$WHO[2][3] = 0.1 + (-7.79 \times 10^{-3}) = 0.092$

$WHO[3][1] = 0.1 + (0.0281) = 0.1281$

$WHO[3][2] = 0.1 + (-7.79 \times 10^{-3}) = 0.092$

$WHO[3][3] = 0.1 + (-7.79 \times 10^{-3}) = 0.092$

$BO[1]_{new} = BO[1]_{old} + \Delta BO[1] = 0.1 + 0.0491 = 0.1491$

$BO[2] = 0.1 - 0.01392 = 0.086$

$BO[3] = 0.1 - 0.01392 = 0.086$

สำหรับแต่ละ hidden unit ในเลเยอร์ที่ 2 , Outputh2[j] , (j=1,2,3)

$WH2[1][1]_{new} = WH2[1][1]_{old} + \Delta WH2[1][1] = 0.1 + 3.35 \times 10^{-4}$

$WH2[1][2] = 0.1 + 3.35 \times 10^{-4} = 0.1$

$WH2[1][3] = 0.1 + 3.35 \times 10^{-4} = 0.1$

$WH2[2][1] = 0.1 + 3.35 \times 10^{-4} = 0.1$

$WH2[2][2] = 0.1 + 3.35 \times 10^{-4} = 0.1$

$WH2[2][3] = 0.1 + 3.35 \times 10^{-4} = 0.1$

$WH2[3][1] = 0.1 + 3.35 \times 10^{-4} = 0.1$

$WH2[3][2] = 0.1 + 3.35 \times 10^{-4} = 0.1$

$WH2[3][3] = 0.1 + 3.35 \times 10^{-4} = 0.1$

ทำการปรับค่า bias

$BH2[1]_{new} = BH2[1]_{old} + \Delta BH[1] = 0.1 + 5.2 \times 10^{-4}$

$BH2[2] = 0.1 + 5.2 \times 10^{-4} = 0.10052$

$BH2[3] = 0.1 + 5.2 \times 10^{-4} = 0.10052$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับแต่ละฮิดเด่นยูนิตในเลเยอร์ที่ 1 (Outputh1[h] , (h=1,2,3))

$WH1[1][1] = WH1[1][1]_{old} + \Delta WH1[1][1] = 0.1+0 = 0.1$

WH1[1][2] = 0.1

WH1[1][3] = 0.1

WH1[2][1] = 0.100052

WH1[2][2] = 0.100052

WH1[2][3] = 0.100052

WH1[3][1] = 0.1

WH1[3][2] = 0.1

WH1[3][3] = 0.1

WH1[4][1] = 0.100052

WH1[4][2] = 0.100052

WH1[4][3] = 0.100052

WH1[5][1] = 0.100052

WH1[5][2] = 0.100052

WH1[5][3] = 0.100052

WH1[6][1] = 0.100052

WH1[6][2] = 0.100052

WH1[6][3] = 0.100052

WH1[7][1] = 0.1

WH1[7][2] = 0.1

WH1[7][3] = 0.1

WH1[8][1] = 0.100052

WH1[8][2] = 0.100052

WH1[8][3] = 0.100052

WH1[9][1] = 0.1

WH1[9][2] = 0.1

WH1[9][3] = 0.1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทำการปรับค่า bias

$$BH1[1]_{new} = BH1[1]_{old} + \Delta BH1[1] = 0.100052$$

$$BH1[2]_{new} = BH1[2]_{old} + \Delta BH1[2] = 0.100052$$

$$BH1[3]_{new} = BH1[3]_{old} + \Delta BH2[3] = 0.100052$$

ขั้นตอนที่ 12 อ่านอินพุตจากไฟล์ชุดฝึกสอนจนกว่าจบไฟล์ คือนำอินพุตชุดฝึกสอนสอนให้แก่โครงข่ายนิวรอนครบทุกตัวค่อยทำการปรับค่า errorb จากสูตร

$$errorb_{new} = errorb_{old} + errorp$$

(แต่จากตัวอย่างเป็นการสอนโครงข่ายนิวรอนด้วยอินพุต 1 ตัวจึงไม่ทำขั้นตอนที่ 12)

ขั้นตอนที่ 13 ทดสอบเงื่อนไขการสิ้นสุดการฝึกสอนคือถ้า (error > 0.05) และ

$(errorb_{new} - errorb_{old}) > 0.0002$ เป็นเท็จอันใดอันหนึ่งให้หยุดการฝึกสอนโครงข่ายนิวรอน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

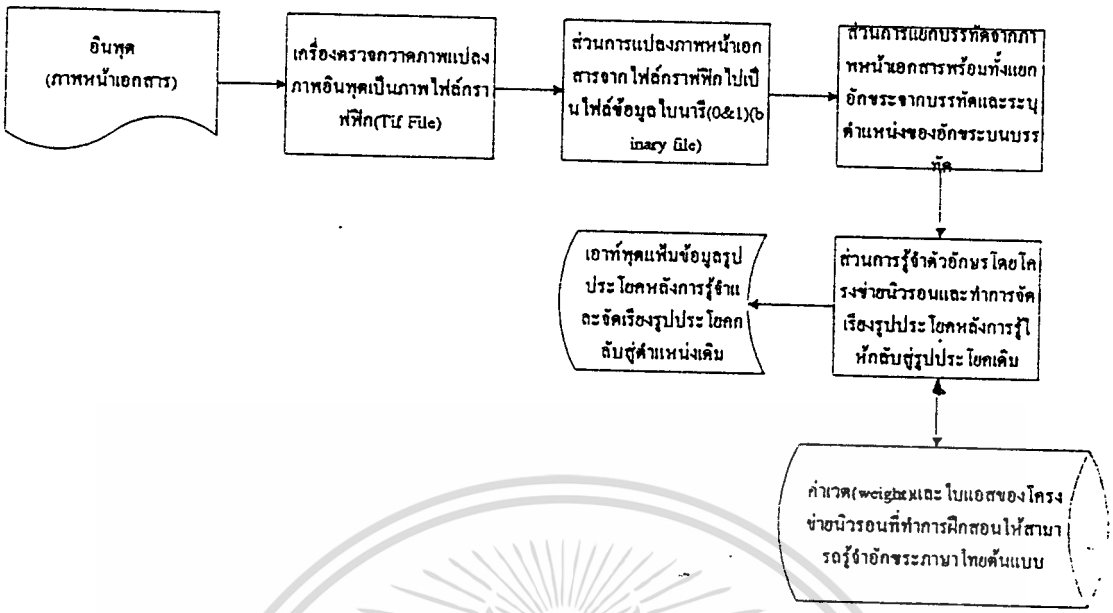
บทที่ 5

ขั้นตอนการทำงานและการใช้งานของโปรแกรมระบบ

5.1 ขั้นตอนการทำงานของโปรแกรม

ขั้นตอนการทำงานของระบบทั้งหมดนั้นประกอบด้วยขบวนการทำงานในแต่ละขั้นตอนตามลำดับการแสดงในรูปที่ 5.1 โดยเริ่มต้นจากการตรวจกวาดภาพหน้าเอกสารโดยเครื่องตรวจกวาดภาพซึ่งจะทำการแปลงอินพุตภาพหน้าเอกสารไปเป็นภาพไฟล์กราฟฟิกของหน้าเอกสารซึ่งอยู่ในรูปแบบทิฟไฟล์ (Tif file) หลังจากนั้นจึงทำการแปลงภาพไฟล์กราฟฟิกของภาพหน้าเอกสารไปเป็นภาพไฟล์ข้อมูลไบนารี (0,1) โดยโปรแกรมบนเครื่องไมโครคอมพิวเตอร์ที่ทำการจำลองขึ้นโดยแทนส่วนพื้นกระดาษขาวของหน้าเอกสารด้วยข้อมูล 0 และแทนส่วนเนื้อตัวอักษรของภาพหน้าเอกสารด้วยข้อมูล 1 หลังจากนั้นจึงทำการถ่านโอนข้อมูลภาพไฟล์ไบนารีลงบนเครื่องมินิคอมพิวเตอร์ ซึ่งมีชื่อว่า "diamond" ซึ่งอยู่บนเน็ตเวิร์คที่มีรหัสของที่อยู่คือ 161.246.4.3 = "diamond.ce.kmitl.ac.th" เพื่อทำการเรียกโปรแกรมในการแยกบรรทัดของภาพหน้าเอกสารพร้อมทั้งแยกอักขระออกจากข้อมูลในแต่ละบรรทัดและระบุตำแหน่งของอักขระบนบรรทัดซึ่งทำการแยกมา โดยผลลัพธ์จะถูกเก็บไว้ในไฟล์เพื่อทำการนำไฟล์ผลลัพธ์ไปรู้จำตัวอักษรโดยโครงข่ายนิวรอนที่ทำการจำลองขึ้นหลังการรู้จำจะนำผลการรู้จำ ไปเข้าสู่ส่วนการจัดเรียงรูปประโยค เพื่อทำการจัดเรียงรูปประโยคให้กลับสู่ตำแหน่งเดิม โดยผลลัพธ์สุดท้ายที่ได้คือแฟ้มข้อมูลรหัสแอสกี้ของอักขระที่เรียงเป็นรูปประโยคเดิม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5-1 ภาพแสดงการทำงานของระบบทั้งหมด

ในการทำงานของระบบจะมีทั้งโปรแกรมคอมพิวเตอร์ที่นำมาใช้และที่พัฒนาขึ้นเองบนเครื่องไมโครคอมพิวเตอร์และบนเครื่องมินิคอมพิวเตอร์สำหรับการทำงานในแต่ละขั้นตอน โดยแบ่งโปรแกรมที่ใช้งานได้เป็น 4 ส่วนดังนี้

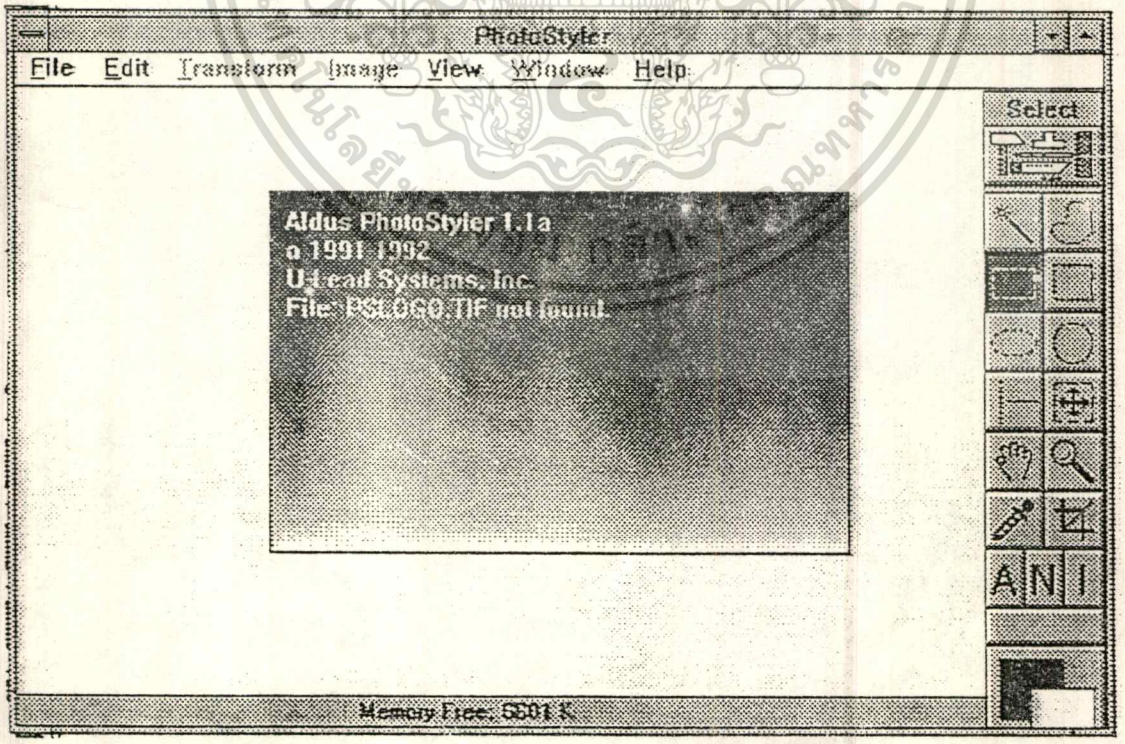
1) โปรแกรมสำเร็จรูปสำหรับเก็บภาพจากเครื่องตรวจกวาดภาพบนเครื่องไมโครคอมพิวเตอร์
2) โปรแกรมที่พัฒนาขึ้นสำหรับการแปลงภาพหน้าเอกสารจากไฟล์กราฟฟิกไปเป็นไฟล์ข้อมูลไบนารีบนเครื่องไมโครคอมพิวเตอร์
3) โปรแกรมที่พัฒนาขึ้นสำหรับการแยกอักขระและระบุตำแหน่งของข้อมูลในแต่ละบรรทัดจากภาพหน้าเอกสาร บนเครื่องไมโครคอมพิวเตอร์
4) โปรแกรมที่พัฒนาขึ้นสำหรับการรู้จำตัวอักษรโดยโครงข่ายนิวรอน และการจัดเรียงรูปประโยคของอักขระหลังการรู้จำให้อยู่ตำแหน่งเดิมของประโยค บนเครื่องมินิคอมพิวเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.2 ขั้นตอนการใช้งานโปรแกรมที่พัฒนาขึ้นทั้งหมด

5.2.1) <u>โปรแกรมสำเร็จรูปสำหรับเก็บภาพจากเครื่องตรวจกวาดภาพ</u>

โปรแกรมสำเร็จรูปสำหรับเก็บภาพหน้าเอกสารที่เลือกใช้มีชื่อว่า PhotoStyler version 1.1a ซึ่งประกอบด้วยดีไวซ์ไดรเวอร์สามารถควบคุมการทำงานของเครื่องตรวจกวาดภาพยี่ห้อ Hewlett Package รุ่น ScanJet ที่เป็นเครื่องตรวจกวาดภาพที่ใช้ในงานวิจัยนี้ได้เป็นอย่างดี โดยผู้ใช้สามารถเลือกระดับความละเอียดของการตรวจกวาด (Scanning resolution) ได้ตั้งแต่ 38-600 dpi (Dot per inch) และขนาดของหน้าเอกสารที่สามารถตรวจกวาดได้มีขนาดใหญ่ได้ถึงขนาด 8.5x14 นิ้ว ของข้อมูลอินพุตในลักษณะของภาพหน้าเอกสารสามารถเก็บบันทึกไว้ในแฟ้มข้อมูลกราฟฟิกได้ในหลาย ๆ รูปแบบ เช่น Tif , Bmp , Pcx หรือ Gif สำหรับงานวิจัยนี้ได้เลือกรูปแบบคือ Tif format ซึ่งเป็นรูปแบบแฟ้มข้อมูลที่เหมาะสมกับการจัดเก็บข้อมูลภาพจากเครื่องตรวจกวาดภาพทั้งนี้จากข้อได้เปรียบของโครงสร้างของแฟ้มข้อมูลแบบนี้ซึ่งมีลักษณะการจัดเก็บข้อมูลที่ไม่แน่นอนตายตัวดังนั้นจึงเป็นการง่ายสำหรับการเพิ่มเติมรายละเอียดของข้อมูลเมื่อมีการแก้ไขหรือปรับแต่งข้อมูลภาพภายหลัง (รายละเอียดของโครงสร้างของแฟ้มข้อมูลภาพ Tif กล่าวไว้ในส่วนภาคผนวก ก. ตอนท้ายของวิทยานิพนธ์ฉบับนี้)



รูปที่ 5-2 ภาพโปรแกรม PhotoStyler

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.2.2) <u>โปรแกรมที่พัฒนาขึ้นสำหรับการแปลงภาพหน้าเอกสารจากไฟล์กราฟฟิกไปเป็นไฟล์ข้อมูลไบนารีบนเครื่องไมโครคอมพิวเตอร์</u>

โปรแกรมคอมพิวเตอร์ที่พัฒนาขึ้นในการแปลงภาพหน้าเอกสารจากไฟล์กราฟฟิก (Tif file) ไปเป็นไฟล์ข้อมูลไบนารี่นี้ได้พัฒนาขึ้นด้วยภาษา C โดยใช้ตัวคอมไพล์เลอร์ชื่อ Borland C++ Version 3.1 ของบริษัท Borland International ทำงานบนสภาวะของระบบปฏิบัติการดอส (Dos Environment) บนเครื่องไมโครคอมพิวเตอร์ตระกูล IBM ทั่ว ๆ ไป

การทำงานทั้งหมดของโปรแกรมนี้ มีขั้นตอนการสั่งงานจากดอสพร้อมท์ โดยการคีย์คำสั่งตามรูปแบบดังนี้คือ

C:\> TIF2B \S(Input tif filename) \D(Output binary filename)

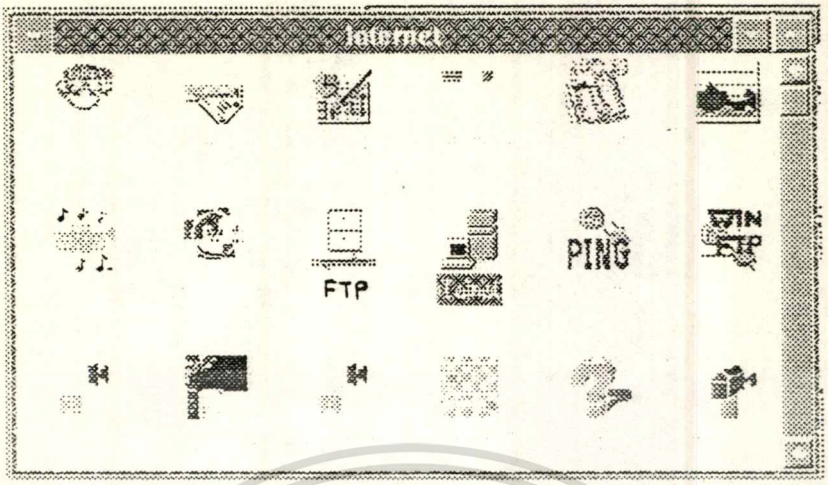
โดยเราต้องป้อนชื่ออินพุตไฟล์กราฟฟิก (Tif file) ตามหลังสัญลักษณ์ "\S" และป้อนชื่อเอาท์พุตไฟล์ไบนารีหลังสัญลักษณ์ "\D" จากนั้นกดแป้น [Enter] โปรแกรมจะทำการแปลงภาพไฟล์กราฟฟิกไปเป็นไฟล์ไบนารีโดยการแทนรหัส 0 ตรงส่วนพื้นกระดาษขาวหน้าเอกสารและแทนรหัส 1 ตรงส่วนเนื้อตัวอักษร โดยขณะที่โปรแกรมกำลังทำการประมวลผลอยู่จะขึ้นข้อความบนจอภาพ ดังรูปที่ 5.3

[C:\]tif2b /SInput.tif /DOutput.txt
Program convert TIFF file to binary 0-1 text file V 0.01a,Written By Thawee
programe is converting *

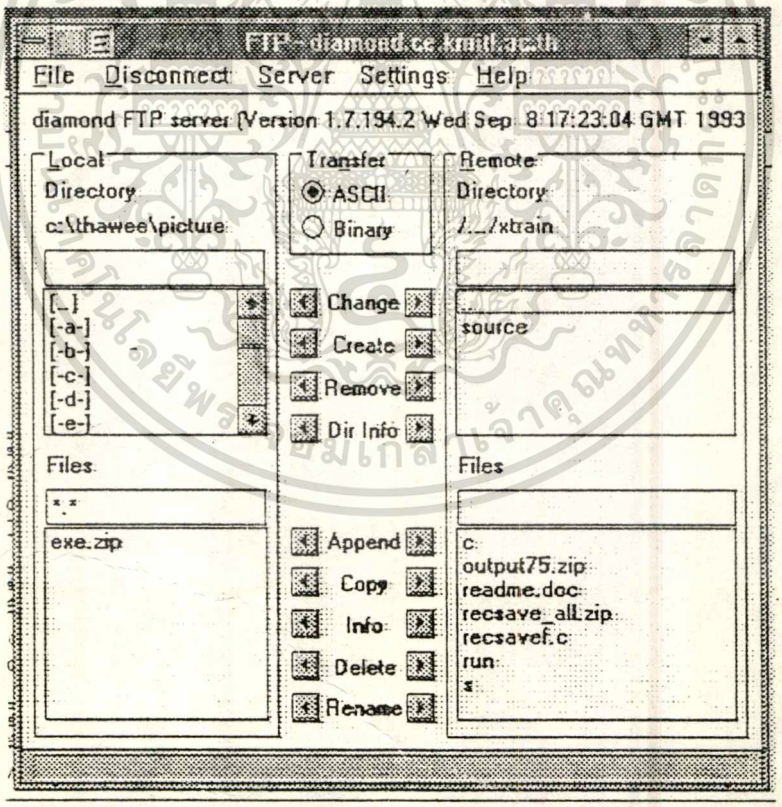
รูปที่5-3 ภาพขณะรันโปรแกรม Tif2b.exe

เมื่อโปรแกรมทำงานเสร็จแล้วบนจอภาพจะปรากฏเครื่องหมายพร้อมท์อีกครั้ง "C:\>" จากนั้นเมื่อได้ไฟล์ไบนารีของข้อมูลภาพหน้าเอกสารแล้ว เราจะนำไฟล์ที่ได้นี้ถ่ายโอนไปบนเครื่องมินิคอมพิวเตอร์ซึ่งมีชื่อว่า ไดมอนด์ (diamond) โดยมีรหัสที่อยู่บนเน็ตเวิร์คคือ 161.246.4.3 = "diamone.ce.kmitl.ac.th" ซึ่งการถ่ายโอนไฟล์นี้จะถ่ายโอนชนิด binary โดยโปรแกรมสำเร็จรูปบนวินโดวส์ซึ่งมีชื่อว่า "FTP" บนโปรแกรม ไมโครซอฟท์วินโดวส์ฟอร์เวอร์กกรุ๊ป (Microsoft Windows for Work Group 3.11) ซึ่งมีรูปดังแสดงในภาพ ตามรูปที่ 5.4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5-4 ภาพแสดงไอคอนของ "FTP" บนหน้าจอวินโดวส์และภาพขณะทำการถ่ายโอนไฟล์โดยโปรแกรม FTP ดังรูปที่ 5-5

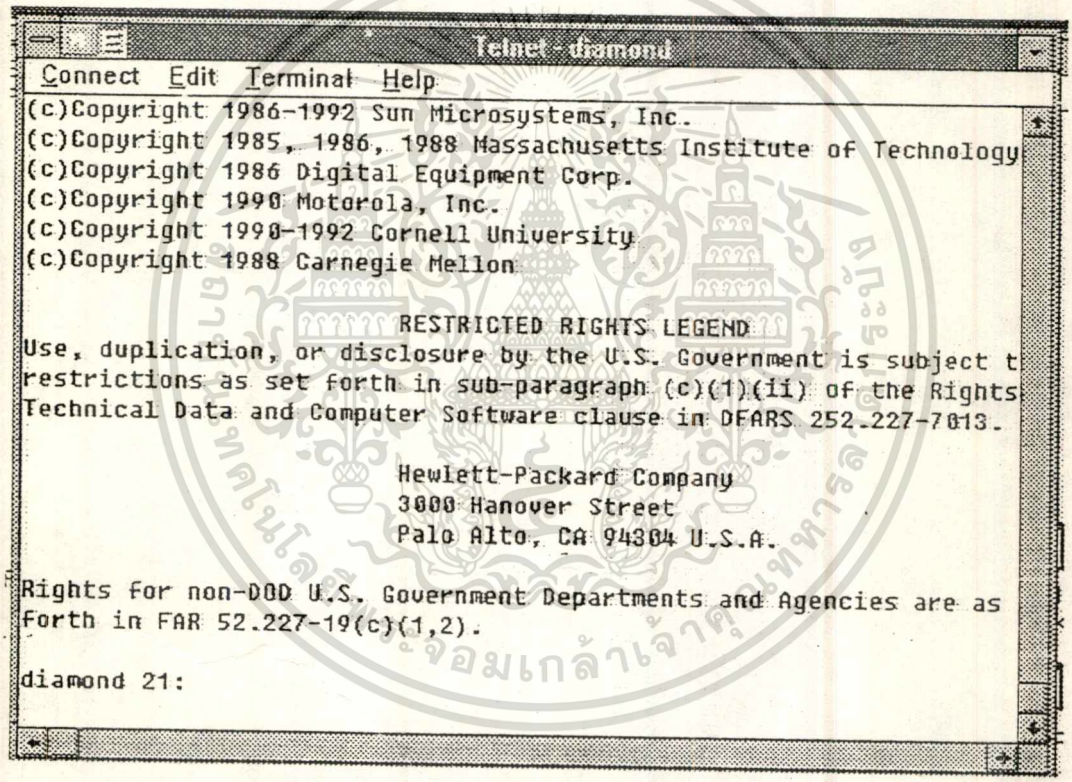


รูปที่ 5-5 ภาพขณะทำการถ่ายโอนไฟล์โดยโปรแกรม "FTP" บนวินโดวส์ฟอร์เวิร์กกรุป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.2.3) <u>โปรแกรมที่พัฒนาขึ้นสำหรับการแยกตัวอักขระและระบุตำแหน่งของข้อมูลในแต่ละบรรทัดจากภาพหน้าเอกสารบนเครื่องมินิคอมพิวเตอร์</u>

การทำงานของโปรแกรมนี้บนเครื่องมินิคอมพิวเตอร์ที่มีชื่อว่าไดมอนด์ (diamond) ซึ่งมีที่อยู่บนเน็ตเวิร์กคือ 161.246.4.3 = "diamond.ce.kmitl.ac.th" ซึ่งเราสามารถเข้าสู่ระบบเน็ตเวิร์กได้โดยผ่านโปรแกรมซีมูเลตเทอมินัล(simulate terminal) ซึ่งมีชื่อว่า เท็ลเน็ต (telnet) บนวินโดวส์ฟอร์เวิร์กกรุปบนเครื่องไมโครคอมพิวเตอร์ดังภาพไอคอนเท็ลเน็ตในรูปที่ 5-4 และภาพดังรูปที่ 5-6 แสดงขั้นตอนการเข้าระบบการทำงานของเครื่อง ไดมอนด์ (diamond) โดยผ่านโปรแกรมเท็ลเน็ต



รูปที่ 5-6 ภาพแสดงขั้นตอนการเข้าระบบการทำงานของเครื่องไดมอนด์ โดยโปรแกรมเท็ลเน็ต

เมื่อเราสามารถติดต่อเข้าสู่ระบบเครื่องไดมอนด์ได้แล้ว เราสามารถทำการใช้งานโปรแกรมที่พัฒนาขึ้นสำหรับการแยกอักขระและระบุตำแหน่งของข้อมูลในแต่ละบรรทัดจากไฟล์ ภาพหน้าเอกสาร ซึ่งเราทำการแปลงมาเป็นข้อมูลไบนารี (0&1) ได้โดยการสั่งงานตามรูปแบบคำสั่งดังนี้ คือ

$cutx3.out [Input binary file] [Output file name]<enter>

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยการพิมพ์ชื่อโปรแกรมตามด้วยชื่อไฟล์อินพุตที่เป็นไฟล์ภาพหน้าเอกสารซึ่งทำการแปลงเป็นรหัสไบนารี 0&1 ไว้แล้ว แล้วตามด้วยชื่อ ไฟล์ผลลัพธ์ ซึ่งเป็นไฟล์ที่เราทำการแยกอักขระออกจากบรรทัดของหน้าเอกสารพร้อมระบุตำแหน่งของอักขระแต่ละตัวบนบรรทัดนั้น ๆ เมื่อพิมพ์ตามขั้นตอนเสร็จแล้วให้กดแป้น Enter รอจนกว่าโปรแกรมจะทำงานเสร็จ เราจะได้ไฟล์ อักขระแต่ละตัวพร้อมตำแหน่งบนหน้าเอกสาร เพื่อนำไปรู้จำโดยโครงข่ายนิวรอนซึ่งทำการเรียนรู้ไว้แล้ว

5.2.4) <u>โปรแกรมที่พัฒนาขึ้นสำหรับการรู้จำตัวอักษรโดยโครงข่ายนิวรอนและการจัดเรียงอักขระกลับสู่ตำแหน่งเดิมในประโยคข้อความหลังการรู้จำ</u>

เมื่อได้ไฟล์ซึ่งประกอบด้วยอักขระและตำแหน่งของอักขระที่ทำการแยกออกมาแต่ละตัวจากหน้าเอกสารโดยโปรแกรม cutx3.out แล้วเราจึงใช้โปรแกรมคอมพิวเตอร์ที่พัฒนาขึ้นสำหรับการรู้จำโดยโครงข่ายนิวรอนและจัดเรียงอักขระกลับสู่ตำแหน่งเดิม ในประโยคหลังการรู้จำโดยโปรแกรมได้พัฒนาขึ้นโดยภาษา c บนเครื่องมินิคอมพิวเตอร์ที่ชื่อ diamond ตัวโปรแกรมจะประกอบด้วยการทำงาน 2 ขั้นตอน. คือ ขั้นตอนการรู้จำตัวอักษรโดยโครงข่ายนิวรอน และขั้นตอนการจัดเรียงอักขระกลับสู่รูปประโยคเดิมโดยเราสามารถเข้าสู่การใช้งานโปรแกรมนี้ได้โดยผ่านโปรแกรมเท็ลเน็ต ในโปรแกรมวินโดวส์ฟอร์เวิร์กกรุป 3.1 (Windows for Workgroup 3.11) บนเครื่องไมโครคอมพิวเตอร์

โปรแกรมที่พัฒนาขึ้นมีชื่อว่า recog.out ซึ่งอยู่ในซับไดเรกทอรี "/Users/stdgr, thawee/xtrain/recog.out โดยเราสามารถใช้งานโปรแกรมนี้ได้ โดยพิมพ์รูปแบบคำสั่งดังคำสั่งข้างล่างนี้ เมื่อพิมพ์คำสั่งการใช้งานตามรูปแบบข้างแล้วโปรแกรมจะดำเนินการจัดจำอักขระแต่ละตัวบนไฟล์อินพุตพร้อมจัดเรียงตำแหน่งของอักขระกลับสู่ตำแหน่งเดิมในบรรทัดของหน้าเอกสารนั้น ซึ่งผลลัพธ์ของการทำงานของโปรแกรมนี้จะเก็บไว้ในไฟล์ชื่อสุดท้ายของการพิมพ์คำสั่งทำงานของโปรแกรมนี้

แสดงคำสั่งการทำงานโปรแกรม recog.out

$ recog.out [input file name] [file weight] [1] [file output of recognition]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 6

การทดสอบและประเมิณผล

6.1 การทดสอบและประเมิณผลความถูกต้องในการจดจำตัวอักษรภาษาไทยชุดทดสอบกับโครงข่ายนิวรอนแบบแบคพรอบพาเกชั่นและเคาท์เตอร์พรอบพาเกชั่น

ในบทนี้จะกล่าวถึงการทดสอบความถูกต้องในการจดจำอักขระภาษาไทยของโครงข่ายนิวรอน 2 แบบคือแบบที่ฝึกสอน โดยอัลกอลิทึมแบบซุปเปอร์ไวส์ (คือแบคพรอบพาเกชั่น) และแบบที่ฝึกสอนโดยอัลกอลิทึมแบบอันซุปเปอร์ไวส์(คือเคาท์เตอร์พรอบพาเกชั่น) ซึ่งโครงข่ายนิวรอนแบบแบคพรอบพาเกชั่นที่ใช้มี 4 เลเยอร์คือ 1 อินพุตเลเยอร์ , 2 ฮิดเด่นเลเยอร์ , 1 เอาท์พุตเลเยอร์ โดยกำหนดจำนวนโหนดในแต่ละเลเยอร์คืออินพุตเลเยอร์เท่ากับ 1,600 โหนด , ฮิดเด่นเลเยอร์เท่ากับ 200 โหนด , เอาท์พุตเลเยอร์เท่ากับ 75 โหนด โดยเวลาที่ใช้ในการฝึกสอนโครงข่ายนิวรอนโครงนี้ใช้เวลา 13 วัน และจำนวนรอบของการฝึกสอนเท่ากับ 349 รอบ และโครงข่ายนิวรอนแบบแบคพรอบพาเกชั่นที่ใช้มี 2 เลเยอร์ คือ 1 อินพุตเลเยอร์ และ 1 คอมเพ็ตทิทีฟเลเยอร์ (Competitive Layer) โดยกำหนดในแต่ละเลเยอร์คืออินพุตเลเยอร์เท่ากับ 1,600 โหนด , คอมเพ็ดทิทีฟ เลเยอร์ (Competitive Layer) เท่ากับ 160 โหนด โดยนำอักขระชุดทดสอบมาทำการจดจำเพื่อเปรียบเทียบเปอร์เซนต์ความถูกต้องในการจดจำและความสามารถของการจดจำของ 2 อัลกอลิทึม

อักขระที่ใช้ในการจดจำได้จากกระดาษ 2 ชนิดคือ

1. อักขระจากกระดาษซึ่งพิมพ์จากเครื่องเลเซอร์พรินเตอร์
2. อักขระจากกระดาษซึ่งทำการถ่ายเอกสารจากกระดาษข้อแรก

และกำหนดรูปแบบการแสกนหน้าเอกสาร 2 รูปแบบคือ

3. การแสกนหน้ากระดาษเอกสารด้วยความสว่าง 100% ของกระดาษข้อ 1 และข้อ 2
4. การแสกนหน้ากระดาษเอกสารด้วยความสว่าง 50% ของกระดาษข้อ 1 และข้อ 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่างภาพอักขระของหน้าเอกสารจากกระดาษข้อ 1,2,3,4 แสดงในรูปที่ 6-1,6-2,6-3,6-4

อักขระชุดทดสอบประกอบด้วยอักขระพยัญชนะภาษาไทย , สระ ,วรรณยุกต์ , เลขอารบิก

รวมทั้งสิ้นเป็นจำนวน 75 ตัวอักษร โดยอักขระ 1 ตัวอักษรจะมีอักขระเพื่อทดสอบ 200 ตัว ดังนั้นอักขระชุดทดสอบจะมีจำนวนตัวอักษรทั้งสิ้น (75*200=15,000 ตัวอักษร) โดยผลการทดสอบการรู้จำตัวอักษรชุดทดสอบ ของโครงข่ายนิวรอนแบบแบคพรอบพาเกชั่น 1 ตัวอักษรกินเวลาการรู้จำเท่ากับ 1.5 วินาที และการจดจำแบบเคาท์เตอร์พรอบพาเกชั่นกินเวลาการรู้จำต่อ 1 ตัวอักษรเท่ากับ (5)วินาที และผลการทดสอบการรู้จำของโครงข่ายนิวรอนทั้ง 2 แบบแสดงผลดังตารางที่ 5 และ 6

ตารางที่ 5 การจดจำอักขระชุดทดสอบ 15,000 ตัวด้วยโครงข่ายนิวรอนแบบแบคพรอบพาเกชั่น

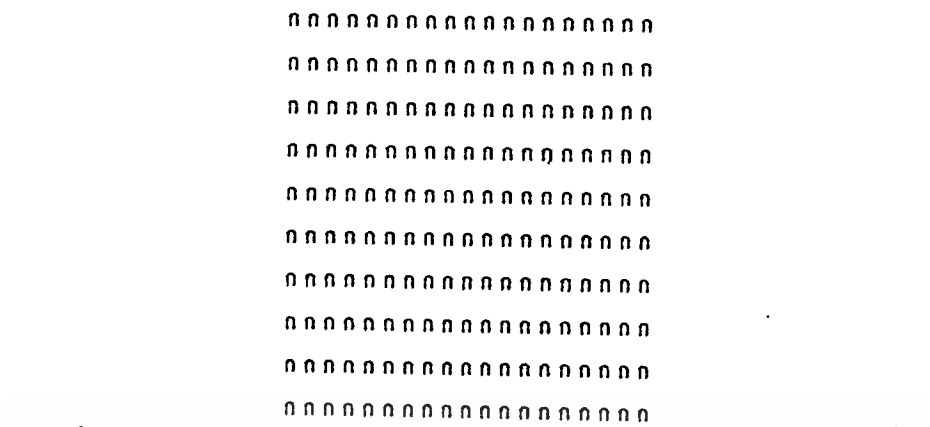
ชนิดกระดาษที่ทดสอบ (เปอร์เซนต์ความสว่างของการแสกน)	จำนวนตัวอักษรที่ไม่สามารถรู้จำจากอักขระ 15,000 ตัวอักษร	เปอร์เซนต์ความถูกต้องโดยรวมของการรู้จำจากอักขระ 15,000ตัวอักษร
อักขระจากการกระดาษเครื่องพิมพ์เลเซอร์ (100%)	7	99.96%
อักขระจากการกระดาษเครื่องพิมพ์เลเซอร์ (50%)	176	98.90%
อักขระจากการกระดาษเครื่องถ่ายเอกสาร (100%)	36	99.73%
อักขระจากการกระดาษเครื่องถ่ายเอกสาร (50%)	560	96.23%

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

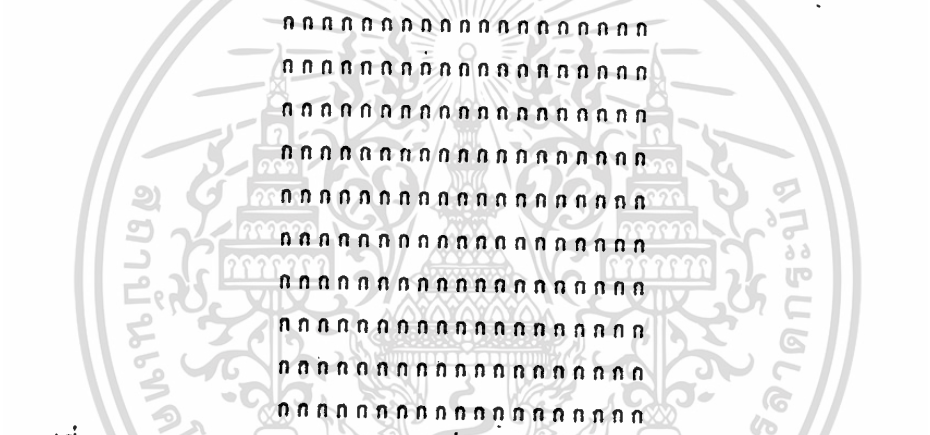
ตารางที่ 6 การจดจำอักขระอินพุตชุดทดสอบ 15,000 ตัว ด้วยโครงข่ายนิวรอนแบบแบคพรอบพาเกชั่น

ชนิดกระดาษที่ทดสอบ (เปอร์เซนต์ความสว่างของการแสกน)	จำนวนตัวอักษรที่ไม่สามารถรู้จำจากอักขระ15,000ตัวอักษร	เปอร์เซนต์ความถูกต้องโดยรวมของการรู้จำจากอักขระ 15,000ตัวอักษร
อักขระจากกระดาษเครื่องพิมพ์เลเซอร์(100%)	21	99.86%
อักขระจากกระดาษเครื่องพิมพ์เลเซอร์(50%)	398	97.34%
อักขระจากกระดาษเครื่องถ่ายเอกสาร(50%)	48	99.68%
อักขระจากกระดาษเครื่องพิมพ์เลเซอร์(100%)	646	95.69%

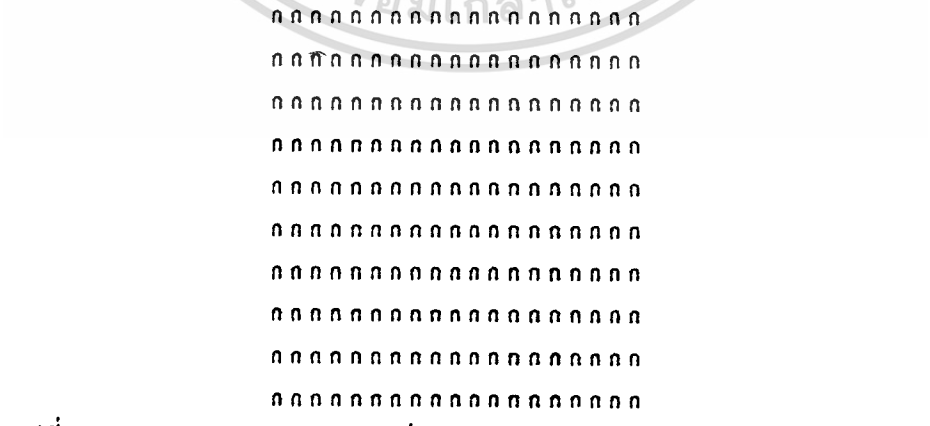
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่6-1 ภาพอักขระจากกระดาษเครื่องพิมพ์เลเซอร์ด้วยการแสกนในอัตราส่วน Resolution=300,Scale=100%,Brightness=100%

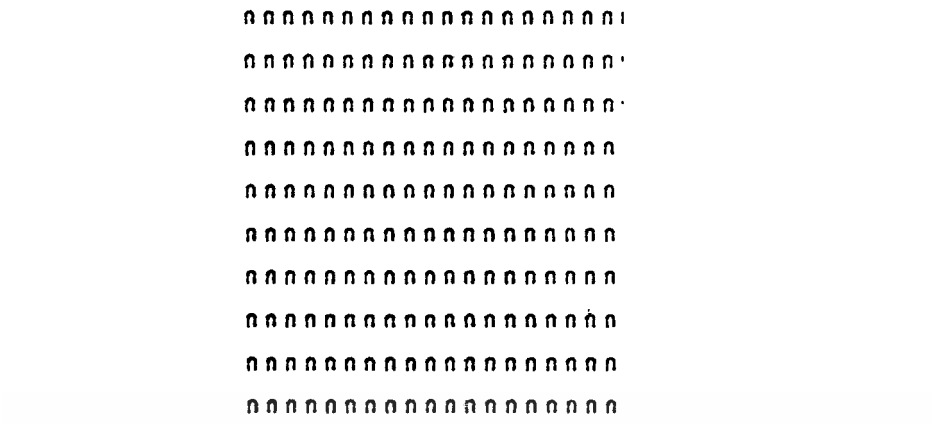


รูปที่6-2 ภาพอักขระจากกระดาษเครื่องพิมพ์เลเซอร์ด้วยการแสกนในอัตราส่วน Resolution=300,Scale=100%,Brightness=50%



รูปที่6-3 ภาพอักขระจากกระดาษซึ่งทำการถ่ายเอกสารกระดาษเลเซอร์และการแสกนด้วยอัตราส่วนResolution=300,Scale=100%,Brightness=100%

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่6-4 ภาพอักขระจากกระดาษซึ่งทำการถ่ายเอกสารกระดาษเลเซอร์และการแสกนด้วยอัตราส่วนResolution=300,Scale=100%,Brightness=50%

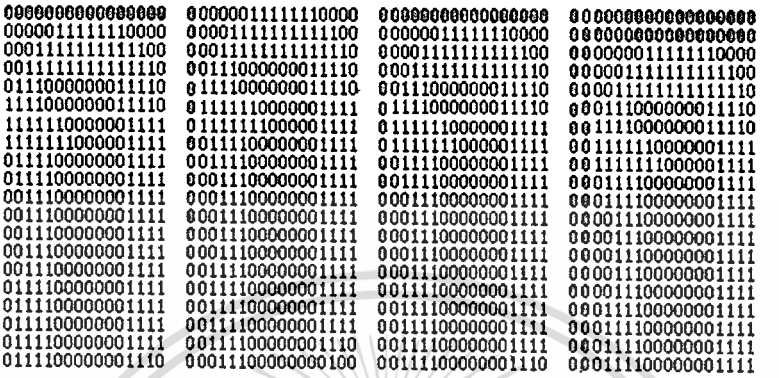
จากผลการทดลองการจดจำอักขระชุดทดสอบด้วยโครงข่ายนิวรอน2แบบพบว่าเปอร์เซนต์ความถูกต้องโดยรวมของโครงข่ายนิวรอนแบบแบคพรอบพาเกชั่นจะสูงกว่าเปอร์เซนต์ความถูกต้องโดยรวมของโครงข่ายนิวรอนแบบเคาท์เตอร์พรอบพาเกชั่น ในกระดาษทดสอบทุกชนิดซึ่งสามารถอธิบายได้ว่าการฝึกสอนแบบอันซุปเปอร์ไวส์(อัลกอลิทึมแบบแบคพรอบพาเกชั่น) นั้นจะใช้เวลาในการปรับปรุงเวตเพื่อจดจำอักขระชุดฝึกสอนอย่างช้า ๆ และปรับปรุงเวตให้ได้เปอร์เซนต์ความผิดพลาดของการรู้จำอักขระโดยรวมน้อยที่สุดเมื่อพบอักขระชุดทดสอบที่ใกล้เคียงกัน แต่การฝึกสอนโครงข่ายนิวรอนแบบซุปเปอร์ไวส์ (อัลกอลิทึมแบบเคาท์เตอร์พรอบพาเกชั่น) นั้นจะใช้เวลาในการจดจำอักขระอินพุตโดยใช้เวต 1 กลุ่มต่อการจดจำอักขระอินพุต 1 รูปแบบ ซึ่งใช้เวลาในการเรียนรู้สั้นแต่ไม่สามารถสร้างเวตที่แยกอักขระที่ใกล้เคียงกันมากๆได้เหมือนแบคพรอบพาเกชั่น ซึ่งเป็นจุดด้อยของอัลกอลิทึมนี้

ซึ่งในการจดจำอักขระต่อการใช้งานจริงจะพบว่าขณะทำการแสกนหน้าเอกสารนั้นจะเกิดสิ่งรบกวน (noise) ต่อภาพอักขระนั้นๆ ทำให้อักขระที่ได้มีลักษณะของภาพที่เกิดการเลื่อนบิตจากด้านบนและด้านข้างจากอินพุตแม่แบบที่ใช้ในการฝึกสอนอยู่ประมาณ 1 ถึง 2 บิต จึงได้ทำการทดสอบการเลื่อนบิตกับกลุ่มอักขระชุดทดสอบ 15,000 ตัว โดยแบ่งการทดสอบการจดจำเป็น 4 แบบคือ

1. การทดสอบการจดจำอักขระโดยการเลื่อนภาพอักขระลง 1 แถว ดังภาพที่ 6-5
2. การทดสอบการจดจำอักขระโดยการเลื่อนภาพอักขระไปทางขวา 1 แถว ดังภาพที่ 6-6
3. การทดสอบการจดจำอักขระโดยการเลื่อนภาพอักขระไปทางขวาและลงข้างล่างอย่างละ 1 แถว ดังภาพที่ 6-7

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. การทดสอบการจดจำอักขระโดยการเลื่อนภาพอักขระไปทางขวาและลงข้างล่างอย่างละ 1 แถว ดังภาพที่ 6-8



รูปที่6-5 รูปที่6-6 รูปที่6-7 รูปที่6-8

โดยผลการทดสอบการรู้จำอักขระ 4 แบบดังกล่าวแสดงในตารางที่ 7,8,9,10 โดยผลการทดลองทั้ง 4 ตารางนี้ไม่รวมค่าผิดพลาดจากขั้นตอน Segmentation

ตารางที่7 การจดจำอักขระโดยการเลื่อนภาพอักขระลง 1 แถว โดยวิธีแบคพรอบพาเกชั่นนิวรอน

ชนิดกระดาษที่ทดสอบ (เปอร์เซนต์ความสว่างในการแสกน)	จำนวนตัวอักษรที่จดจำผิดจากอักขระทั้งหมด 15,000ตัว	เปอร์เซนต์ความถูกต้องของการรู้จำอักขระ 15,000ตัว
อักขระจากกระดาษเลเซอร์ (100%)	242	98.38
อักขระจากกระดาษเลเซอร์ (50%)	2616	82.56
อักขระจากกระดาษซีรอกซ์ (100%)	333	97.78
อักขระจากกระดาษซีรอกซ์ (50%)	2378	84.14

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่8 การจดจำอักขระโดยการเลื่อนภาพอักขระไปทางขวา 1 แถว โดยวิธีแบคพรอบพาเกชั่นนิวรอน

ชนิดกระดาษที่ทดสอบ (เปอร์เซนต์ความสว่างในการแสกน)	จำนวนตัวอักษรที่จดจำผิดจากอักขระทั้งหมด 15,000ตัว	เปอร์เซนต์ความถูกต้องของการรู้จำอักขระ 15,000ตัว
อักขระจากกระดาษเลเซอร์ (100%)	1542	89.72
อักขระจากกระดาษเลเซอร์ (50%)	4129	72.47
อักขระจากกระดาษซีรอกซ์ (100%)	1307	91.28
อักขระจากกระดาษซีรอกซ์ (50%)	4019	73.20

ตารางที่9 การจดจำอักขระโดยการเลื่อนภาพอักขระไปทางขวาและลงล่างอย่างละ 1 แถว โดยวิธีแบคพรอบพาเกชั่นนิวรอน

ชนิดกระดาษที่ทดสอบ (เปอร์เซนต์ความสว่างในการแสกน)	จำนวนตัวอักษรที่จดจำผิดจากอักขระทั้งหมด 15,000ตัว	เปอร์เซนต์ความถูกต้องของการรู้จำอักขระ 15,000ตัว
อักขระจากกระดาษเลเซอร์ (100%)	2704	81.97
อักขระจากกระดาษเลเซอร์ (50%)	6252	58.32
อักขระจากกระดาษซีรอกซ์ (100%)	2371	84.19
อักขระจากกระดาษซีรอกซ์ (50%)	6448	57.01

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่10 การจดจำอักขระโดยการเลื่อนภาพอักขระไปทางขวา 2 แถว โดยวิธีแบคพรอบพาเกชั่นนิวรอน

ชนิดกระดาษที่ทดสอบ (เปอร์เซนต์ความสว่างในการแสกน)	จำนวนตัวอักษรที่จดจำผิดจากอักขระทั้งหมด 15,000ตัว	เปอร์เซนต์ความถูกต้องของการรู้จำอักขระ 15,000ตัว
อักขระจากกระดาษเลเซอร์ (100%)	11,264	24.90
อักขระจากกระดาษเลเซอร์ (50%)	12,882	14.12
อักขระจากกระดาษซีรอกซ์ (100%)	11,182	25.45
อักขระจากกระดาษซีรอกซ์ (50%)	13,055	12.96

จากผลการทดลองดังตารางที่ 7,8,9 พบว่าเปอร์เซนต์ความถูกต้องในการจดจำอักขระในการเลื่อนบิตภาพลง 1 แถว , ไปทางขวา 1 แถว , ลงและไปทางขวาอย่างละ 1 แถว นั้น เปอร์เซนต์ความถูกต้องยังสูงเกิน 50% ซึ่งทำให้สามารถนำไปใช้งานจริงได้ แต่ภาพที่มีการเลื่อนบิตลงและไปทางขวาอย่างละ 2 บิต ดังตารางที่10นั้นพบว่าเปอร์เซนต์ความถูกต้องในการจดจำภาพอักขระจะน้อยกว่า 50 % ทำให้รู้ว่าโครงข่ายนิวรอนแบบแบคพรอบพาเกชั่นไม่สามารถจดจำภาพอักขระที่มีการเลื่อนบิตไปทางขวาและลงล่างเกิน 1 แถวได้สูงเกิน 50% ของการรู้จำทั้งหมด

และเพื่อการทดสอบการใช้งานโครงข่ายนิวรอนแบบแบคพรอบพาเกชั่นกับโครงข่ายนิวรอนแบบอื่นว่ามีประสิทธิภาพแตกต่างกันอย่างไรจึงได้ทำการจดจำอักขระ 4 แบบ จากภาพที่ 1,2,3,4 กับโครงข่ายนิวรอนแบบเคาท์เตอร์พรอบพาเกชั่น เพื่อเปรียบเทียบประสิทธิภาพต่อการรู้จำอักขระจริงจากหน้าเอกสารดังตารางที่ 11,12,13,,14 โดยผลการทดลองทั้ง 4 ตารางนี้ไม่รวมค่าผิดพลาดจากขั้นตอน Segmentation

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่11 การจดจำอักขระโดยการเลื่อนภาพอักขระลง 1 แถว โดยวิธีเคาท์เตอร์พรอบพาเกชั่นนิวรอน

ชนิดกระดาษที่ทดสอบ (เปอร์เซนต์ความสว่างในการแสกน)	จำนวนตัวอักษรที่จดจำผิดจากอักขระทั้งหมด 15,000ตัว	เปอร์เซนต์ความถูกต้องของการรู้จำอักขระ 15,000ตัว
อักขระจากกระดาษเลเซอร์ (100%)	246	98.36
อักขระจากกระดาษเลเซอร์ (50%)	1783	88.11
อักขระจากกระดาษซีรอกซ์ (100%)	320	97.86
อักขระจากกระดาษซีรอกซ์ (50%)	646	95.69

ตารางที่12 การจดจำอักขระโดยการเลื่อนภาพอักขระไปทางขวา 1 แถว โดยวิธีเคาท์เตอร์พรอบพาเกชั่นนิวรอน

ชนิดกระดาษที่ทดสอบ (เปอร์เซนต์ความสว่างในการแสกน)	จำนวนตัวอักษรที่จดจำผิดจากอักขระทั้งหมด 15,000ตัว	เปอร์เซนต์ความถูกต้องของการรู้จำอักขระ 15,000ตัว
อักขระจากกระดาษเลเซอร์ (100%)	2441	83.72
อักขระจากกระดาษเลเซอร์ (50%)	4903	67.31
อักขระจากกระดาษซีรอกซ์ (100%)	1943	87.04
อักขระจากกระดาษซีรอกซ์ (50%)	5411	63.92

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่13 การจดจำอักขระโดยการเลื่อนภาพอักขระไปทางขวาและลงล่างอย่างละ 1 แถว โดยวิธีเคาท์เตอร์พรอบพาเกชั่นนิวรอน

ชนิดกระดาษที่ทดสอบ (เปอร์เซนต์ความสว่างในการแสกน)	จำนวนตัวอักษรที่จดจำผิดจากอักขระทั้งหมด 15,000ตัว	เปอร์เซนต์ความถูกต้องของการรู้จำอักขระ 15,000ตัว
อักขระจากกระดาษเลเซอร์ (100%)	3354	77.63
อักขระจากกระดาษเลเซอร์ (50%)	7492	50.05
อักขระจากกระดาษซีรอกซ์ (100%)	2875	80.83
อักขระจากกระดาษซีรอกซ์ (50%)	7944	47.04

ตารางที่14 การจดจำอักขระโดยการเลื่อนภาพอักขระไปทางขวาและลงล่างอย่างละ 2 แถว โดยวิธีเคาท์เตอร์พรอบพาเกชั่นนิวรอน

ชนิดกระดาษที่ทดสอบ (เปอร์เซนต์ความสว่างในการแสกน)	จำนวนตัวอักษรที่จดจำผิดจากอักขระทั้งหมด 15,000ตัว	เปอร์เซนต์ความถูกต้องของการรู้จำอักขระ 15,000ตัว
อักขระจากกระดาษเลเซอร์ (100%)	12,541	16.39
อักขระจากกระดาษเลเซอร์ (50%)	13,402	10.65
อักขระจากกระดาษซีรอกซ์ (100%)	12,101	19.32
อักขระจากกระดาษซีรอกซ์ (50%)	13,276	11.49

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากผลการทดลองดังตารางที่ 11,12,13,14 พบว่าเปอร์เซนต์ความถูกต้องโดยรวมในการจดจำของโครงข่ายนิวรอนแบบเคาท์เตอร์พรอบพาเกชั่นนั้นมีเปอร์เซนต์ความถูกต้องเฉลี่ยน้อยกว่าแบบโครงข่ายนิวรอนแบบแบคพรอบพาเกชั่นอยู่เล็กน้อย และเปอร์เซนต์ความถูกต้องโดยรวมของการเลื่อนบิตภาพลง 1 และ 2 แถว การเลื่อนบิตภาพไปทางขวา 1 แถว , การเลื่อนบิตภาพไปทางขวา 1 แถว , การเลื่อนบิตภาพไปทางขวาและลงล่าง 1 แถว ยังคงมีเปอร์เซนต์ความถูกต้องสูงเกิน 50% ซึ่งทำให้สามารถนำไปใช้งานจริงได้แต่ภาพที่มีการเลื่อนบิตลงและไปทางขวา 2 แถว จะพบว่าเปอร์เซนต์ความถูกต้องในการจดจำภาพอักขระจะน้อยกว่า 50% ทำให้ทราบว่าโครงข่ายนิวรอนแบบเคาท์เตอร์พรอบพาเกชั่นซึ่งมีการฝึกสอนโครงข่ายนิวรอนแบบอันซุปเปอร์ไวส์(Unsupervised Training) ไม่สามารถจดจำภาพอักขระที่มีการเลื่อนบิตภาพลงพร้อมทั้งไปทางขวา 2 แถวได้เกินกว่า 50% ของอักขระโดยรวมทั้งหมด

6.2 การทดสอบและเปรียบเทียบความถูกต้องในส่วนของการนำเอาโครงข่ายนิวรอนที่ฝึกสอนโดยอัลกอลิทึมแบบ แบคพรอบพาเกชั่น และ แบบเคาท์เตอร์พรอบพาเกชั่นไปประยุกต์ใช้งานจริงกับตัวอย่างข้อมูลหน้าเอกสาร

เพื่อให้เกิดการทดสอบโครงข่ายนิวรอนแบบแบคพรอบพาเกชั่นต่อการใช้งานจริง เราจะทดสอบโครงข่ายนิวรอนแบบแบคพรอบพาเกชั่นกับโครงข่ายนิวรอนแบบแบคพรอบพาเกชั่นด้วยภาพข้อความจากหน้าเอกสารซึ่งทำการพิมพ์ด้วยเครื่องพิมพ์เลเซอร์โดยอักขระมีขนาดและรูปแบบเดียวกัน คือ อังสานายูพีซี (Angsana Upc) ขนาด 12 dpi และทำการแสกนหน้ากระดาษเอกสารด้วยอัตราส่วนคือ Resolution=300%,Scale=100%,Brightness=100% พร้อมกำจัดสิ่งรบกวนระหว่างบรรทัด โดยภาพหน้าเอกสารแสดงในภาพที่ 9 พร้อมผลลัพธ์การรู้จำหน้ากระดาษเอกสารโดยโครงข่ายนิวรอนแบบแบคพร็อบพาเกชั่นแสดงในภาพที่10 และผลลัพธ์การรู้จำของโครงข่ายนิวรอนแบบเคาท์เตอร์พรอบพาเกชั่นแสดงในภาพที่ 11 โดยจากภาพที่ 10 และ11 ของการรู้จำหน้าเอกสารโดยโครงข่ายนิวรอนแบบแบคพรอบพาเกชั่นมีค่าเท่ากับ 98% และเปอร์เซนต์ความถูกต้องของการจดจำหน้ากระดาษเอกสารของโครงข่ายนิวรอนแบบเคาท์เตอร์พรอบพาเกชั่นมีค่าเท่ากับ 96% โดยเราสามารถสรุปได้ว่าวิธีการรู้จำของโครงข่ายนิวรอนแบบแบคพรอบพาเกชั่นมีประสิทธิภาพสูงกว่าวิธีการรู้จำแบบเคาท์เตอร์พรอบพาเกชั่น และอักขระที่จดจำไม่ได้ของทั้งสองวิธีคืออักขระที่มีขนาดเล็ก เช่น (ไม้เอก), (จุด),(สระอำ),(สระอี),(สระอึ) โดยจำนวนบิตที่ต่างกันมีจำนวนบิตที่น้อยจึงทำให้โอกาสในการรู้จำมีเปอร์เซนต์ลดลง เพราะภาพอักขระมีความแตกต่างกันเพียงไม่กี่บิต ความผิดพลาดของการรู้จำอีกส่วนหนึ่งคือสิ่งรบกวน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(noise) ระหว่างคำทำให้ไม่สามารถแยกอักขระระหว่างคำหรือการแยกอักขระระหว่างบรรทัดได้ จึงทำให้อักขระอินพุตที่ได้มีลักษณะติดกันจนไม่สามารถรู้จำได้ และส่งผลให้การเรียงกลับสู่รูปประโยคเดิมหลังการรู้จำผิดพลาดตามไปด้วย ในการเพิ่มประสิทธิภาพการรู้จำของโครงข่ายนิวรอนให้มีประสิทธิภาพสูงขึ้นสามารถทำได้โดยทำการลบสิ่งรบกวนก่อนการประมวลผลภาพระหว่างคำและระหว่างบรรทัด จะทำให้เปอร์เซนต์การรู้จำของโครงข่ายนิวรอนสูงขึ้นตามไปด้วย

เมื่อเวลา 06.45 น.วันที่ 7 สิงหาคมมีเหตุวางระเบิดขึ้นที่ จ.นราธิวาส โดย ร.ต.ประสิทธิ์ ชูโชติ ผู้อำนวยการศูนย์พลศึกษาและกีฬาจังหวัดนราธิวาส เข้าแจ้งความที่ สภ.อ.เมืองนราธิวาสว่า ขณะที่ตรวจตราพื้นที่โดยรอบอาคารสำนักงานศึกษาธิการจังหวัดนราธิวาส ซึ่งอยู่ในรั้วเดียวกับศาลากลาง จ.นราธิวาส พบว่าที่ด้านหลังอาคารสำนักงานศึกษาธิการจังหวัดใกล้เครื่องทำความเย็นมีระเบิดชนิดแสวงเครื่องวางอยู่ ขอให้ไปตรวจสอบและทำการกู้ระเบิดด้วย

หลังจากได้รับรายงานแล้ว นายสวัสดิ์ กฤตรัชตนันท์ ผู้ว่าราชการ จ.นราธิวาส พล.ต.ต.ธรรมนูญ ทับเกลี้ยว ผบก.ทน.ตร.ภ.จว.นราธิวาส พ.ต.อ.กมล ไพริยพ ผกก.สภ.อ.เมืองนราธิวาส พร้อมกำลังตำรวจจำนวนหนึ่งรุดไปที่เกิดเหตุพบกระป๋องนมผงขนาดบรรจุ 1 กิโลกรัมภายในมีดินระเบิดและเชื้อประทุไฟฟ้า เชื่อมต่อกับนาฬิกาและถ่านไฟฉาย 3 ก้อนเป็นตัวจุดเชื้อประทุ นอกจากนี้ ยังพบสีสเปรย์อีก 1 กระป๋อง

ภายหลังกู้ระเบิดเสร็จแล้วพบว่า ระเบิดแสวงเครื่องดังกล่าวอยู่ในถุงกระดาษถูกตั้งเวลาระเบิดไว้ที่เวลา 05.00 น. แต่เข็มนาฬิกาหยุดเดินเมื่อเวลา 02.50 น. เจ้าหน้าที่ตำรวจได้ผ่ากระป๋องนมที่บรรจุดินระเบิดภายในประกอบด้วยดินระเบิดชนิด ซีโฟร์หนัก 1 ใน 4 ปอนด์ปนในเตรดผสมน้ำมันไซลาร์น้ำหนัก 1.4 กิโลกรัม ตะปูขนาด 3 นิ้วจำนวน 128 ตัว จึงเก็บไว้เป็นหลักฐาน คาดว่าหากระเบิดทำงานจะทำให้อาคารสำ

รูปที่9 แสดงภาพข้อความจากการแสกนหน้าเอกสารซึ่งทำการพิมพ์จากกระดาษเลเซอร์ด้วยอัตราส่วนการแสกนคือ Resolution=300%,Scale=100%,Brightness=100%

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อเวลา0645น่วันที่7สิงหาคมมีเหตุวางระ
เบิดขึ้นที่ฉ่นราธิวาสโดยรถประสิทธิ์ชูโชติผู้อุ
นวยการศูนย์พลศึกษาและกีฬาจังหวัดนราธิวาสเข้า
แจ้งความที่สภ่อเมืองนราธิวาสว่าขณะที่เดินตรวจ
ตราพื้นที่โดยรอบอาคารสุานักงานศึกษาธิการฉังหวัด
นราธิวาสซึ่งอยู่ในรั้วเดียวกับศาลากลางฉ่นราธิวาส
พบว่าที่ด้านหลังอาคารสุานักงานศึกษาธิการฉังหวัด
ธใกล้เครื่องทุาความเย็นมีระเบิดชนิดแสวงเครื่องวางอยู่
ขอใฝ่ไปตรวจสอบและทุาการกู้ระเบิดด้วย
หลังฉากได้รับรายงานแล้วนายสวัสดิ์กฤตรัชต
นันท์ผู้ว่าราชการจ่นราธิวาสพลตต่ธรรมนูญทับ
เคลียวผบก่หน่ตร์ภ่จว่นราธิวาสพ่ต่อ่กมลโพธิ
รยพผกก่สภ่อ่เมืองนราธิวาสพร้อมกุาลังตุารวจฉุ
นวนหนึ่งรุดไปที่เกิดเหตุพบกระป๋องนมผงขนาดบรรจ
1กิโลกรัมภายในมีดินระเบิดและเชื่อประทุไฟฟ้าเชื่อ
ต่อกับนาฬิกาและถ่านไฟฉาย3ก้อนเป็นตัวจุดเชื่อประ
ทุนอกจากนี้ยังพบฟีสเปรย์อีก1กระป๋อง
ภายหลังกู้ระเบิดเสร็จแล้วพบว่าระเบิดแสวงเครื่อง
ดังกล่าวอยู่ในถุงกระดาษถูกตั้งเวลาระเบิดไว้ที่เวลา
05๋00น่แต่เข็มนาฬิกาหยุดเดินเมื่อเวลา02๋50น
เจ้าหน้าที่ตุารวจได้ผ่ากระป๋องนมที่บรรจุดินระเบิด
ภายในประกอบด้วยดินระเบิดชนิดซีโฟร์หนัก1ใน
ปอนด์ปัยในเตรดผสมน้ำมันโซลาร์น้ำหนัก14กิโล
กรัมตะปูขนาด3นิ้วจุานวน128ตัวจึงเก็บไว้เป็น
หลักฐานคาดว่าหากระเบิดทุางานฉะทุาให้อาคารสุ

<u>รูปที่10</u> ภาพผลลัพธ์ของการรู้จำข้อความจากการป้อนอินพุตด้วยภาพที่ 1 โดยโครงข่ายนิวรอนแบบแบคพรอบพาเกชั่น (อินพุตเลเยอร์=1,600โหนด,ฮิดเด่นเลเยอร์=230 โหนด,เอาท์พุตเลเยอร์=87 โหนด)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อเวลา 0645 นวันที่ 7 สิงหาคมมีเหตุวางระ
เบิดขึ้นที่ จ นราธิวาสโดยร ต ประสิทธิ์ชูโชติ ผู้อำ
นวยการศูนย์พลศึกษาและกีฬาจังหวัดนราธิวาส เข้า
แจ้งความที่ สภ อเมืองนราธิวาสว่า ขณะที่เดินตรวจ
ตราพื้นที่โดยรอบอาคารสำนักงานศึกษาธิการจังหวัด
นราธิวาส ซึ่งอยู่ในรั้วเดียวถับศาลากลาง จ นราธิวาส
พบว่าที่ด้านหลังอาคารสำนักงานศึกษาธิการจังหวัด
ใกล้เครื่องทำความเย็นมีระเบิดชนิดแสวงเครื่องวางอยู่
ขอให้ไปตรวจสอบและทำการกู้ระเบิดด้วย
หลังจากได้รับรายงานแล้ว นายสวัสดิกฤดรัชต
นันท์ ผู้ว่าราชการ จ นราธิวาส พล ต ตธรรมนูญ ทับ
เสลียว ผบก หน ตร ภจว นราธิวาส พ ต อ กมล โพธิ
ยพ ผกก สภ อเมืองบราธิวาส พร้อมกำลังตำรวจจำ
นวนหนึ่งรุดไปที่เกิดเหตุพบกระป๋องนมผงขนาดบรรจุ
1 กิโลกรัมภายในมีดินระเบิดและเชื้อประทุไฟฟ้าเชื่อม
ต่อกับนาฬิกาและถ่านไฟฉาย 3 ก้อนเป็นตัวจุดเชื้อปะ
ทุ นอกจากนี้ยังพบฟีสเปรย์อีก1 กระป๋อง
ภายหลังกู้ระเบิดเสร็จแล้วพบว่าระเบิดแสวงเครื่อง
ดังกล่าวอยู่ในถุงกระดาษถูกตั้งเวลาระเบิดไว้ที่เวลา
0500 น แต่เจ็บนาฬิกาหยุดเดินเมื่อเวลา 0250 น
เจ้าพนักงานตำรวจได้ผ่ากระป๋องนมที่บรรจุดินระเบิด
ภายในประกอบด้วยดินระเบิดชนิด ซีโฟร์หนัก1 ใน 4
ปอนด์บุ๋ยในเตรตผสมน้ำมันโซลาร์น้ำหนัก14 กิโล
กรัม ตะปูขนาด 3 นิ้วจำนวน128 ตัว จึงเก็บไว้เป็น
หลักฐาน คาดว่าหากระเบิดทำงานจะทำให้อาคารสำ

<u>รูปที่11</u> ภาพผลลัพธ์ของการรู้จำข้อความจากการป้อนอินพุตด้วยภาพที่ 1 โดยโครงข่ายนิวรอนแบบเคาท์เตอร์พรอบพาเกชั่น (2 เลเยอร์ คือ อินพุตเลเยอร์ = 1064 โหนด, คอมเพ็ททิทีฟเลเยอร์(Competitive Layer)=165 โหนด)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 7

บทสรุป

7.1 ข้อสรุปและเปรียบเทียบของการรู้จำ 3 วิธีคือ การรู้จำโดยวิธีโครงข่ายนิวรอน,การรู้จำโดยวิธีแมตชิ่ง (Matching) , การรู้จำโดยวิธีวิเคราะห์โครงสร้างทางกายภาพ

ในการศึกษาทฤษฎีการรู้จำตัวอักขระนั้น สามารถสรุปทฤษฎีการรู้จำหลักๆได้คือ การรู้จำโดยวิธีโครงข่ายนิวรอน เช่น แบคพรอบพาเกชั่น , การรู้จำโดยวิธีแมตชิ่ง(Matching) เช่น เคาท์เตอร์พรอบพาเกชั่น โดยนำผลการวิจัยจากวิทยานิพนธ์ของ นายอภิรักษ์ จิรายุสกุล เรื่องการวิเคราะห์ตัวพิมพ์อักษรภาษาไทยโดยวิธี เคาท์เตอร์ พรอบพาเกชั่น และ การวิเคราะห์โครงสร้างทางกายภาพ เช่น การพิจารณาโครงสร้างทางกายภาพโดยการเปรียบเทียบตารางQ-codeโดยนำผลการวิจัยจากวิทยานิพนธ์ของนายสมศักดิ์ วลัยรัชต์ เรื่อง การวิเคราะห์และระบุส่วนประกอบของหน้าเอกสารและการรู้จำตัวอักษร ซึ่งวิธีการรู้จำโดยวิธีแมตชิ่งมีหลักการทำงานที่สามารถสรุปได้คือ ทฤษฎีที่นำอักขระแม่แบบมาเก็บไว้ในหน่วยความจำแล้วนำอักขระอินพุตที่ต้องการรู้จำเข้าไปเปรียบเทียบกับอักขระแม่แบบ แบบบิทต่อบิทที่มีตำแหน่งเหมือนกันระหว่างอักขระแม่แบบและอักขระอินพุตที่ต้องการรู้จำถ้าพบว่าบิทที่มีตำแหน่งเหมือนกันมีข้อมูลเหมือนกันเป็นจำนวนมาก จะสรุปได้ว่าอักขระอินพุตและอักขระแม่แบบนั้นมีรูปร่างเหมือนกันคือมีเปอร์เซนต์ความใกล้เคียงกันสูง โดยการเปรียบเทียบต้องนำอักขระอินพุตไปเปรียบเทียบกับอักษรแม่แบบทุกตัวในหน่วยความจำ ถ้ามีอักขระหลายรูปแบบ เช่น อักขระภาษาไทยมีรูปแบบพยัญชนะ และสระ (67 รูปแบบ) , เลขไทย (10 รูปแบบ) , เลขอารบิก (มี 10 รูปแบบ) จะต้องนำอินพุตที่ต้องการรู้จำไปเปรียบเทียบกับอักขระแม่แบบทุกตัว เพื่อหาเปอร์เซนต์ความใกล้เคียงของอักขระอินพุตกับอักขระแม่แบบทุกตัว และถ้าเปอร์เซนต์ความเหมือนของอักขระอินพุตสูงสุดที่อักขระแม่แบบตัวใดจะถือว่ามีอักขระอินพุตเหมือนกับอักขระแม่แบบตัวนั้นมากที่สุด

และวิธีการรู้จำโดยวิธีวิเคราะห์โครงสร้างทางกายภาพ คือวิธีการค้นหาคุณลักษณะเด่นทางกายภาพของอักขระ โดยทำการพิจารณาคุณสมบัติทางโทโพโลยีของภาพอักขระแล้วนำมาหารหัสภาพของโทโพโลยีนั้นๆ ซึ่งขอยกตัวอย่างวิธีการหาคุณสมบัติเด่นของอักขระคือ วิธีฟีเจอร์คอนเซนเทรชั่น ซึ่งวิธีนี้มีการทำงานสามารถสรุปได้คร่าว ๆ คือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1) หาจุดศูนย์ถ่วงของภาพอักขระอินพุต

2) แบ่งอักขระออกเป็น9ส่วนตามตำแหน่งจุดส่วนถ่วง

3) หารหัสโครงสร้างของภาพทั้ง9ส่วนโดยเทียบจากภาพสัญลักษณ์ของ Q-code

4) รวมรหัสของ Q-code จากภาพทั้ง 9 ส่วนก็จะได้รหัสของภาพอินพุตนั้นจากกรรมวิธีการรู้จำโดยวิธีวิเคราะห์ทางกายภาพนี้จะต้องทำการหาลักษณะทางกายภาพของอักขระแม่แบบทุกตัวก่อนแล้วจึงเก็บลงในฐานข้อมูลเพื่อใช้ในการเปรียบเทียบเมื่อมีอักขระอินพุตใหม่เข้ามาทำการเปรียบเทียบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางสรุปข้อดี ของการรู้จำ 3 วิธี คือ โครงข่ายนิวรอน , การรู้จำโดยวิธีแมตชิ่ง (Matching) การรู้จำโดยวิธีวิเคราะห์ทางกายภาพ

	การรู้จำโดยวิธีโครงข่ายนิวรอน	การรู้จำโดยวิธีแมตชิ่ง	การรู้จำโดยวิธีวิเคราะห์ทางกายภาพ
ข้อดี	1) มีประสิทธิภาพการรู้จำที่ดี	1) มีการประมวลผลที่ไม่ยุ่งยากซับซ้อน	1) มีการประมวลที่ไม่ยุ่งยากซับซ้อน และไม่มากนัก และสามารถทำการรู้จำได้รวดเร็ว
	2) ใช้เวลาฝึกสอนโครงข่ายนิวรอนให้รู้จำอักขระแม่แบบเพียงครั้งเดียว	2) สามารถเพิ่มอักขระแม่แบบใหม่ในชุดอักขระแม่แบบได้ง่าย	2) สามารถเพิ่มอักขระแม่แบบใหม่ในชุดอักขระแม่แบบได้ง่ายโดยทำการหารหัสของโครงสร้างทางกายภาพนั้นก่อน
	3) การรู้จำอักขระ 2 ตัว เสียเวลาประมาณ 1 วินาที	3) การรู้จำอักขระ 6 ตัว เสียเวลาประมาณ 1 วินาที	3) การรู้จำอักขระ 2 ตัว เสียเวลาประมาณ 1 วินาที
	4) โครงข่ายนิวรอนรูปแบบเดียวสามารถรู้จำอักขระแม่แบบได้ทุกชนิดที่มีลักษณะใกล้เคียงกัน	4) ไม่เสียเวลาฝึกสอนให้รู้จำอักขระแม่แบบเหมือนโครงข่ายนิวรอน หรือไม่ต้องสิ้นเปลืองเวลาในการหารหัสของโครงสร้างทางกายภาพเหมือนการรู้จำโดยวิธีวิเคราะห์ทางกายภาพ	4) ใช้หน่วยความจำในการเก็บรหัสของโครงสร้างทางกายภาพที่น้อยมาก
	5) สามารถเปลี่ยนขนาดของโครงข่ายได้ง่ายจากการเปลี่ยนขนาดของตัวแปรในโปรแกรม	5) ประสิทธิภาพในการรู้จำจะดีมากถ้าอักขระแม่แบบที่ต้องการรู้จำมีรูปร่างไม่ใกล้เคียงกันมากจนเป็นปัญหาในการรู้จำแบบแมตชิ่ง	5) สามารถเพิ่มอักขระแม่แบบได้ง่ายโดยการหารหัสโครงสร้างทางกายภาพของอักขระแม่แบบนั้นแล้วจัดเก็บลงในฐานข้อมูลไว้เปรียบเทียบกับอักขระอินพุตที่เข้ามาใหม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	6) สามารถรู้จำอักขระที่มีลักษณะใกล้เคียงกันได้ โดยฝึกสอนอักขระที่ใกล้เคียงกันนั้นให้โครงข่ายนิวรอนรู้จำก่อนใช้งาน	6) ใช้ในการรู้จำกรณีที่ต้องการการประมวลผลที่รวดเร็ว	6) ใช้ในการรู้จำที่ต้องการการประมวลที่เร็วและใช้ในหน่วยความจำที่ไม่มาก คือ เก็บเพียงรหัสของโครงสร้างทางกายภาพเท่านั้น
	7) ใช้ในการรู้จำที่ต้องการผลการทดลองรู้จำที่สูงในกรณีที่อักขระแต่ละตัวมีลักษณะใกล้เคียงกันมาก	7) สามารถประมวลผลบนเครื่องไมโครคอมพิวเตอร์ได้	7) สามารถประมวลผลบนเครื่องไมโครคอมพิวเตอร์ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางสรุปข้อเสีย ของการรู้จำ 3 วิธี คือ โครงข่ายนิวรอน , การรู้จำโดยวิธีแมตซิ่ง (Matching) การรู้จำโดยวิธีวิเคราะห์ทางกายภาพ

	การรู้จำโดยวิธีโครงข่ายนิวรอน	การรู้จำโดยวิธีแมตซิ่ง	การรู้จำโดยวิธีวิเคราะห์ทางกายภาพ
ข้อเสีย	1) เสียเวลาในการฝึกสอนอักขระชุดฝึกสอนนานในการวิจัยใช้เวลา 13 วัน	1) การรู้จำจะเริ่มใช้เวลานานถ้ามีอักขระแม่แบบเพิ่มมากขึ้นในการรู้จำเพราะเสียเวลาในการทำแมตซิ่งกับอักขระแม่แบบทุกตัว	1) สามารถรู้จำอักขระได้ถูกต้องถ้ามีสิ่งรบกวนในภาพอักขระไม่มากนักเพราะจำทำให้ได้รหัสของโครงสร้างทางกายภาพที่เปลี่ยนแปลงไปจากเดิม
	2) ต้องใช้เครื่องคอมพิวเตอร์ที่มีความสามารถจองหน่วยความจำได้ประมาณ 14 เมกะไบท์ในการประมวลผลการรู้จำของโครงข่ายนิวรอนขนาด 2 ฮิดเด่นเลเยอร์ (เลเยอร์ละ 230 โหนด)	2) เสียหน่วยความจำเพิ่มมากขึ้นตามจำนวนอักขระแม่แบบที่เพิ่มมากขึ้น	2) ไม่สามารถรู้จำอักขระที่มีลักษณะเอียงได้เพราะจะทำให้ได้รหัสโครงสร้างทางกายภาพที่เปลี่ยนแปลงไปจากรหัสของอักขระแม่แบบ
	3) ไม่สามารถรู้จำอักขระที่มีลักษณะเอียงได้	3) สามารถรู้จำอักขระได้ถูกต้องกรณีที่มีสิ่งรบกวนในภาพอักขระไม่มากนักได้	3) ไม่สามารถรู้จำอักขระที่มีลักษณะโครงสร้างทายกายภาพที่ใกล้เคียงกันมากได้เช่น (“ค”,” ”),(“ด”, “ต”) เป็นต้น
	4) ถ้ามีสิ่งรบกวนในภาพอักขระที่ต้องการรู้จำจะทำให้เปอร์เซนต์ความเหมือนของอักขระอินพุตต่ออักขระแม่แบบลดน้อยลงตามปริมาณสิ่งรบกวนของหน้าเอกสาร	4) ไม่สามารถรู้จำอักขระที่มีลักษณะเอียงจากแม่แบบได้	4) ถ้าภาพอักขระที่ต้องการรู้จำมีสิ่งรบกวนจะทำให้เกิดรหัสของโครงสร้างใหม่ต้องเสียเวลาในการให้ผู้ใช้ระบุค่าเป้าหมายของอักขระนั้น ๆ ใหม่ทุกครั้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	5) ถ้าต้องการเพิ่มอักขระแม่แบบใหม่ให้โครงข่ายนิวรอนรู้จักต้องเสียเวลาในการฝึกสอนอักขระนั้นให้โครงข่ายเรียนรู้ก่อนซึ่งใช้เวลาในการเรียนรู้ที่นาน	5) การรู้จำโดยวิธีแมตซิ่ง (Matching) จะไม่สามารถรู้จำอักขระที่มีลักษณะใกล้เคียงกันมาก ๆ ได้เช่น (“ค” , “ ”,(“ด”,“ ”)	5) ไม่สามารถรู้จำอักขระที่ขาดแหว่ง,หรือหมึกพิมพ์ซึมมาก จนทำให้ลายเส้นไม่สมบูรณ์และผิดเพี้ยนไปซึ่งทำให้คำนวณได้รหัสแทนตัวอักษรนั้นเป็นรหัสใหม่ที่ไม่มีในฐานข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7.2 ปัญหาและข้อจำกัดในการทำงานของระบบ

ในการวิจัยนี้ประกอบด้วยขั้นตอนการทำงานต่างๆหลายขั้นตอนซึ่งอาจเกิดปัญหาและข้อจำกัดอยู่ในแต่ละส่วนดังนี้คือ

7.2.1 ปัญหาและข้อจำกัดของข้อมูลอินพุตที่เป็นข้อความของหน้าเอกสาร

ข้อมูลอินพุตซึ่งเป็นภาพข้อความหน้าเอกสารขนาดกระดาษ A4 ซึ่งมีขนาดความกว้างประมาณ 8 นิ้วและยาวประมาณ 11 นิ้ว การจัดเก็บภาพหน้าเอกสารให้อยู่ในรูปแบบที่เครื่องคอมพิวเตอร์สามารถนำไปประมวลผลได้นั้นทำได้โดยการผ่านเครื่องตรวจกวาดภาพหน้าเอกสารเพื่อแปลงให้อยู่ในรูปข้อมูลไบนารี งข้อมูลหนึ่งไบท์คือหนึ่งพิกเซล(Pixel) ของภาพหน้าเอกสารนั้นเอง ดังนั้นภาพหน้าเอกสารบนกระดาษ A4 เมื่อตรวจกวาดมาด้วยความละเอียดของการตรวจกวาดภาพขนาด 300 จุดต่อนิ้ว ก็จะประกอบด้วยจุดซึ่งเป็นความกว้างของภาพเท่ากับ 2,400 จุด และความสูงของภาพเท่ากับ 3,300 จุดทำให้ภาพหน้าเอกสารนี้ประกอบด้วยจุดภาพทั้งสิ้น 7,920,000 ไบท์ด้วยเช่นกัน

การพิจาณาภาพหน้าเอกสารเพื่อแยกอักขระตัวอักษรแต่ละตัวออกจากภาพหน้าเอกสารนั้นอาศัยทฤษฎีติดตามเนื้อตัวอักษรที่มีรหัส “1” ติดกันซึ่งดัดแปลงมาจากหลักการของทฤษฎีหลักการติดตามรอยของภาพของ ทฤษฎีเซลลูลาร์ออโตมาต้า (Cellular Automata) ซึ่งการพิจารณาอักขระที่มีรหัส “1” ติดกันเพื่อแยกอักขระออกจากภาพหน้าเอกสารนั้นต้องคำนวณประมวลผลซ้ำ ๆ กันเป็นจำนวณมากและกินเวลาการทำงานมากด้วยเช่นเดียวกันถ้าภาพข้อมูลที่ใช้มีขนาดใหญ่มากเท่าไร ก็ยิ่งต้องใช้เวลาในการประมวลผลมากขึ้นเท่านั้น ซึ่งแนวทางการแก้ปัญหาในเรื่องของเวลาที่ใช้ในการประมวลผลสามารถทำได้ด้วย การเพิ่มประสิทธิ์ภาพทางฮารด์แวร์ให้กับเครื่องคอมพิวเตอร์ไม่ว่าจะเป็นการเพิ่มหน่วยความจำ(Ram) หรือเลือกใช้หน่วยประมวลกลาง(CPU) ที่มีความเร็วของการประมวลผลที่เร็วขึ้น

สาเหตุการรู้จำของโครงข่ายนิวรอนที่ทำให้เกิดความผิดพลาดมีสาเหตุคือ

1. เกิดจากภาพหน้าเอกสารที่มีอักขระแต่ละตัวติดกัน ซึ่งเกิดจากสิ่งรบกวนระหว่างอักขระมีมากจนส่งผลให้ภาพอักขระแต่ละตัวนั้นมีภาพติดกัน

2. อีกกรณีหนึ่งที่จะทำให้การรู้จำตัวอักษรผิดพลาดไปได้คือภาพหน้าเอกสารเอียงซึ่งเกิดจากการวางกระดาษบนเครื่องตรวจกวาดภาพไม่ตรงเมื่อทำการตรวจกวาดภาพหน้าเอกสารจะทำให้ได้อักขระที่เอียงมีลักษณะที่แตกต่างจากอักขระที่ใช้เป็นแม่แบบในขณะทำการฝึกสอนโครงข่ายนิวรอน ซึ่งก็จะทำให้ไม่สามารถรู้จำอักขระได้ถูกต้องได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7.2.2. ปัญหาและข้อจำกัดของการทำงานในส่วนการรู้จำตัวอักษรของโครงข่ายนิวรอนแบบแบคพรอบพาเกชั่น

เนื่องจากการรู้จำโดยโครงข่ายนิวรอนแบบแบคพรอบพาเกชั่นนั้นจะต้องทำการฝึกสอนโครงข่ายนิวรอนด้วย อักขระชุดฝึกสอนก่อนจึงต้องเสียเวลาในการฝึกสอนโครงข่ายนิวรอนกับชุดฝึกสอนนั้น ซึ่งกินเวลามากน้อยตามจำนวนรูปแบบที่จะทำให้โครงข่ายนิวรอนรู้จำและขนาดของโครงข่ายนิวรอนที่จะฝึกสอนให้รู้จำ คือถ้าอักขระชุดฝึกสอนมีจำนวนรูปแบบที่มากจะต้องใช้โครงข่ายนิวรอนที่มีจำนวนฮิดเด่นโหนด(hidden node) ที่มากตามไปด้วยทำให้สิ้นเปลืองเวลาการฝึกสอนโครงข่ายนิวรอนที่ใหญ่ตามขนาดของโครงข่าย และจำนวนรูปแบบที่จะฝึกสอน ซึ่งจากการทดลองในงานวิจัยนี้พบว่าถ้าอักขระที่จะใช้ฝึกสอนมีจำนวนรูปแบบเท่ากับอักขระภาษาไทย 87 รูปแบบ และใช้โครงข่ายนิวรอนขนาดอินพุตโหนดเท่ากับ 1,600 โหนด จำนวนฮิดเด่นเลเยอร์เท่ากับ 2 เลเยอร์ แต่ละเลเยอร์มีจำนวนโหนดเท่ากับ 200 โหนด เอาท์พุตโหนดเท่ากับ 75 โหนดตามจำนวนรูปแบบ ซึ่งพบว่ากินเวลาในการฝึกสอนเท่ากับ 13 วัน และเสียจำนวนรอบในการฝึกสอนเท่ากับ 349 รอบ ซึ่งสิ้นเปลืองเวลาการฝึกสอนที่นานมากและเวลาในการรู้จำอักขระ 1 ตัวภายหลังการฝึกสอนโครงข่ายนิวรอนกินเวลา 1.5 วินาที อีกทั้งโครงข่ายนิวรอนมีการทำงานที่ทั้งต้องอาศัยเนื้อที่หน่วยความจำมากในการทำงาน ทั้งขณะทำการฝึกสอนโครงข่ายนิวรอนและขณะทำการจดจำอักขระจึงต้องอาศัยเครื่องที่มีหน่วยประมวลผลที่เร็วและมีพื้นที่ของหน่วยความจำที่มากตามไปด้วย

7.3 ข้อเสนอแนะสำหรับแนวทางการพัฒนาระบบการรู้จำโครงข่ายนิวรอนแบบแบคพรอบพาเกชั่นในอนาคต

เนื่องจากในการวิจัยฉบับนี้มีการทำงานที่ไม่ครอบคลุมการทำงานบางส่วนจึงได้มีข้อเสนอแนะเพื่อเป็นแนวทางการพัฒนาระบบต่อไปในอนาคต คือ

1. เนื่องจากในเครื่องที่ใช้ในการฝึกสอนโครงข่ายนิวรอนจะต้องเป็นเครื่องที่สามารถประมวลผลได้เร็วและรูปแบบการทำงานของโครงข่ายนิวรอนมีการประมวลผลคำสั่งที่มีรูปแบบที่ซ้ำๆกัน ผู้วิจัยมีความเห็นว่า เหมาะสมกับเครื่องประมวลผลความเร็วสูงที่มีรูปแบบการประมวลผลแบบรีดิวซ์อินสตรัคชั่นเซต(RISC= Reduce Instruction Set) คือเครื่องจะจัดรูปแบบคำสั่งที่มีขั้นตอนการทำงานคล้ายๆ กันเป็นชุดๆ แล้วดำเนินการประมวลผลแบบขนานกับคำสั่งแต่ละชุดแบบต่อเนื่องกันจึงจะทำให้การฝึกสอนโครงข่ายนิวรอนมีการทำงานที่เร็วขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. เนื่องจากการฝึกสอนโครงข่ายนิวรอนจะต้องนำอักขระชุดฝึกสอนมาฝึกสอนกับโครงข่ายนิวรอน ดังนั้นการจัดกลุ่มอักขระที่มีลักษณะคล้ายๆกันให้อยู่ในกลุ่มเดียวกัน โนเซตอักขระชุดฝึกสอนจะส่งผลให้โครงข่ายนิวรอนสามารถแยกกลุ่มอักขระได้ง่ายขึ้น

3. อักขระที่มีรูปร่างใกล้กันมาก คือ ต่างกันไม่กี่พิกเซล ควรจัดให้มีค่าเป้าหมายเดียวกันแล้ว สร้างโครงข่ายนิวรอนใหม่ ที่มีจำนวนฮิดเด่นโหนดน้อยกว่าโครงข่ายแรก มาทำการแยกอักขระที่มีรูปร่างใกล้เคียงกันเหล่านั้น เพื่อลดเวลาในการฝึกสอน โครงข่ายนิวรอนในการจดจำอักขระที่ใกล้เคียงกันมากๆ

4. ถ้าฝึกสอนโครงข่ายนิวรอนกับอักขระชุดฝึกสอนนั้น ๆ ไม่สามารถทำให้ค่าผิดพลาดน้อยกว่า ค่าผิดพลาดเป้าหมายได้ คือ โครงข่ายนิวรอนมีจำนวนโหนดในฮิดเด่นเลเยอร์น้อยจนไม่สามารถจดจำอักขระแต่ละตัวได้ ควรทำการเพิ่มจำนวนโหนดบนฮิดเด่นเลเยอร์เพื่อเพิ่มประสิทธิภาพการรู้จำของโครงข่ายนิวรอนในการจำแนกอักขระที่มีลักษณะใกล้เคียงกัน

5. เนื่องจากการรู้จำในวิทยานิพนธ์ฉบับนี้เป็นการรู้จำอักขระภาษาไทย 87 อักขระซึ่งเป็นอักขระเพียง 1 รูปแบบเท่านั้นคือ อังสานายูพีซี (Angsana Upc) ขนาด 12 dpi ถ้าสามารถเพิ่มการฝึกสอนโครงข่ายนิวรอนกับอักขระรูปแบบอื่นก็จะทำให้สามารถรู้จำอักขระได้ครบทุกรูปแบบซึ่งจะทำให้เกิดการนำไปใช้งานได้จริง

6. ถ้าเพิ่มความสามารถในการย่อและขยายอักขระที่จะนำมารู้จำได้จะทำให้สามารถฝึกสอนโครงข่ายนิวรอนด้วยอักขระหนึ่งๆ ในอักขระแต่ละรูปแบบซึ่งเป็นการลดเวลาที่สูญเสียไปในการฝึกสอนโครงข่ายนิวรอนกับการรู้จำอักขระอื่นๆ ในอักขระรูปแบบเดียวกัน

7. ถ้าเพิ่มประสิทธิภาพในการกำจัดสิ่งรบกวน (noise) ของภาพหน้าเอกสารได้ดียิ่งขึ้นจะทำให้อักขระที่ได้มีความชัดเจนสูงขึ้น ซึ่งจะทำให้ผลการรู้จำอักขระมีความถูกต้องที่สูงขึ้นและทำให้ขั้นตอนการแยกอักขระออกจากภาพหน้าเอกสารจะมีประสิทธิภาพดียิ่งขึ้น

8. เนื่องจากปัญหาการรู้จำของภาพหน้าเอกสารมีความผิดพลาดเกิดขึ้นได้ถ้าการวางภาพหน้าเอกสารไม่ตรงดังนั้นถ้ามีการเพิ่มขั้นตอนการปรับหมุนภาพก่อนการรู้จำภาพอักขระได้จะเป็นข้อดีที่จะเพิ่มความสามารถในการรู้จำภาพอักขระให้มีความถูกต้องสูงขึ้น

9. ในการวิจัยควรเลือกอักขระแต่ละตัวที่จะใช้ในการฝึกสอนโครงข่ายนิวรอนนั้นควรเลือกภาพอักขระที่ชัดเจน และมีสิ่งรบกวนซึ่งเกิดจากการแสกนน้อยกว่าอักขระทั่วไป ซึ่งจะส่งผลให้การรู้จำมีประสิทธิภาพสูงขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารอ้างอิง

1. ผศ. ชม กิ้มปาน , "ทฤษฎีการรู้จำรูปแบบเบื้องต้น" , ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าวิทยาเขตเจ้าคุณทหารลาดกระบัง กรุงเทพ , พิมพ์ครั้งที่ 1 , กรกฎาคม 2525.

2. Kimpan,C.,A. Itoh and Prof. K. Kawanishi, "Recognition of Printed Thai Characters using a Matching Method" , IEE Proceeding, Vol 130, Pt.E, No.6, November 1983.

3. Kimpan,C., "Printed Thai Characters Recognition" ,Dissertion of Doctor of Engineering King Mongkut's Institute of Technology Chaokhuntaharn Ladkrabang, 1986.

4. Kimpan,C.,A.Itoh and Prof. K .Kawanshi,"Fine Classification of Printed Thai Character Recognition using the Karhunen-love Expansion" , IEE Proceeding , Vol 134,Pt.E, No.5, September 1987.

5. Kimpan,C., "Printed Thai Characters Recognition Using Topological Properties Method" ,Int.J.Electronics 1986,Vol.60,No.3,pp.303-329

6. ชาย เกษมอมรกุล , "การออกแบบพจนานุกรมสำหรับการเรียนรู้อักษรลายมือไทย-อังกฤษ อัตโนมัติบนเครื่องไมโครคอมพิวเตอร์" , วิทยานิพนธ์มหาบัณฑิต คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง,ปีการศึกษา 2532.

7. "Neural Network and Pc tools a Practical Guide" , Russel C. Eberhart and Roy W. Dobbins , Academic Press, Inc.

8. "Artificial Intelligence in Process Eng" , Schumberger Lab For Computer Science, Academic Press,Inc.

9. "An Introduction to Neural and Electrical Network" , Douglas L.Reilly and Leon N, Academic Press,Inc

10. "Handwritten Digit Recognition by Neural Network with Single-Layer Training, Stefan Knerr,Le'on Personnaz ,and Ge'rand Dreyfuz.

11. "An Introduction to Neural Computing" , Igor Aleksander, helen Morton, Chapman&Hall

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

12. สมศักดิ์ วลัยรัชต์ และ รศ.ดร. ชม กิ้มปาน , "การแยกและจัดกลุ่มตัวอักษรตัวพิมพ์หลายรูปแบบ" การประชุมวิชาการทางวิศวกรรมไฟฟ้า ครั้งที่14 , คณะวิศวกรรมศาสตร์ มหาวิทยาลัยสงขลานครินทร์ วิทยาเขตหาดใหญ่ พฤศจิกายน 2534

13. ศิวัฒน์ ศิวะบวร , "Cellular Automata กับปัญหาเขาวงกต" , คอมพิวเตอร์รีวิว ฉบับที่108 ประจำเดือนสิงหาคม 2536

14. Hewlett-Packard , "A Guide to the Tag Image File Format Version 5.0" , Hewlett Packard Company , August 1991 .

15. Hewlett-Packard, "Scanjet User's Guide" , Hewlett-Packard Company, April 1987.

16. Hewlett-Packard, "X Window System Programming: Xlib" , Hewlett-Packard Company October 1989.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

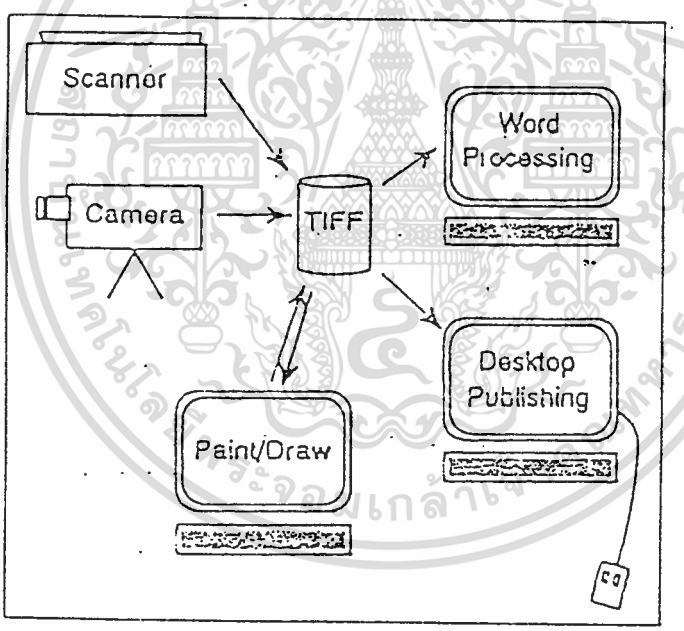
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก.

แฟ้มข้อมูลกราฟิกรูปแบบ TIFF Version 5.0

ภาพรวมของแฟ้มข้อมูลกราฟิกรูปแบบ TIFF

แฟ้มข้อมูล Tag Image File Format หรือเรียกสั้นๆ ว่า TIFF ถูกพัฒนาขึ้นโดยบริษัท Aldus Corporation เพื่อใช้จัดเก็บข้อมูลภาพแบบ Bit-map เราพบว่าโปรแกรมด้าน Desktop-publishing เกือบทั้งหมด รวมถึงโปรแกรมสำหรับสแกนเนอร์ต่างสนับสนุน TIFF ทั้งนั้น ความสามารถในการจัดเก็บรายละเอียดทั้งหมดของภาพไว้ได้ ทำให้ TIFF เป็นที่นิยมและถูกพัฒนาต่อมาขึ้นเป็นซูเปอร์เซ็ตของรูปแบบแฟ้มข้อมูลอื่นๆ ที่ใช้เก็บภาพ



รูปที่ 1 ผังงานของ Scanned images ในงาน Desktop Publishing

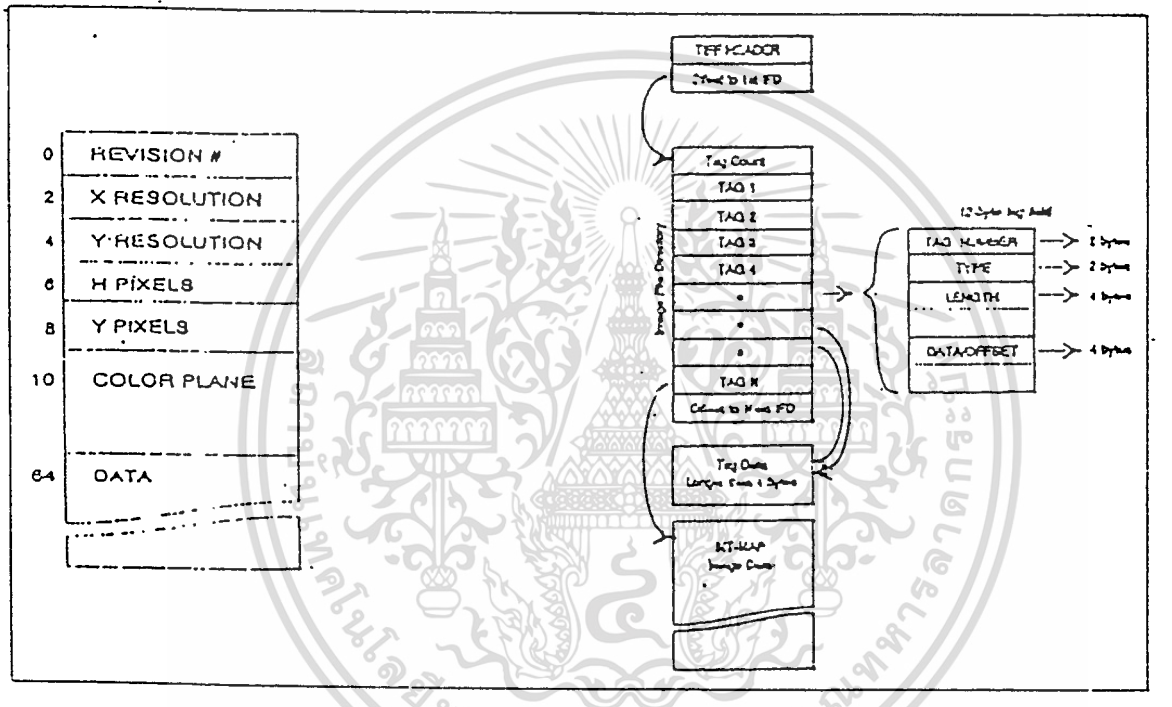
จัยแลกต่างหนึ่งที่ TIFF ต่างจากแฟ้มข้อมูลแบบอื่นๆ คือข้อมูลส่วนใหญ่ที่เก็บอยู่จะไม่ถูกกำหนดตำแหน่งแน่นอนตายตัว โดยมีเพียงข้อมูล 8 ไบต์แรกเท่านั้นที่เป็นตำแหน่งพิเศษและเก็บไว้ที่ตำแหน่งนี้เสมอ ส่วนอื่นๆ ที่เหลือจะถูกอ้างอิงกับส่วนเริ่มต้นของแฟ้มข้อมูล จึงเป็นการง่ายต่อการเพิ่มเติมข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เดิมทีแล้วแฟ้มข้อมูลของโปรแกรมภาคภาพทั่วๆ ไป ใช้รูปแบบแฟ้มข้อมูลแบบแน่นอน (Fixed-format organization) ซึ่งเราสามารถทราบโครงสร้างและตำแหน่งของข้อมูลได้ก่อนการเปิดอ่านแฟ้มข้อมูล ยกตัวอย่างเช่น X resolution จะเก็บอยู่ตำแหน่งไบท์ที่ 2 และ 3 และส่วนของข้อมูลจะเริ่มต้นที่ตำแหน่งไบท์ที่ 64 เป็นต้น

สำหรับแฟ้มข้อมูล TIFF ซึ่งใช้เทคนิคที่แตกต่างกัน โดยแบ่งเป็น Tags เก็บรายละเอียดของข้อมูล ในแต่ละ Tag จะมีความยาว 12 ไบท์อธิบายลักษณะแต่ละส่วนของภาพ ตัวอย่างเช่น ในแฟ้มข้อมูลจะประกอบด้วย Tag สำหรับ X and Y resolution และ Tag สำหรับตำแหน่งของข้อมูล (StripOffsets)

รูปที่ 2 ตัวอย่าง Fixed-file format



รูปที่ 3 ตัวอย่างโครงสร้างทั่วไปของ TIFF

หากพิจารณาถึงโครงสร้างของแฟ้มข้อมูล TIFF พบมีการจัดเก็บรายละเอียดหลายอย่างที่ยังไม่มีความจำเป็น แต่ในทางกลับกันพบว่ากลายเป็นข้อได้เปรียบกว่าแฟ้มข้อมูลแบบตายตัว โดยเมื่อต้องการเพิ่มเติม Tag ใหม่ ไม่จำเป็นต้องเขียนโปรแกรมในส่วนจัดเก็บใหม่ ตัวอย่างเช่น ถ้าเพิ่มเติม Tag ที่เก็บ DateTime ในขณะที่อีกโปรแกรมหนึ่งไม่ Support ข้อมูลของ Tag DateTime โอกาสที่จะเกิดข้อผิดพลาดก็จะไม่เกิดขึ้นในกรณีที่มี Non-support tag อยู่ในแฟ้มข้อมูล ต่างกับแฟ้มข้อมูลแบบตายตัวเมื่อเพิ่มเติมข้อมูลเข้าไปจะต้องเขียนโปรแกรมส่วนจัดการใหม่ และเมื่อโปรแกรมที่ใช้งานที่เป็นโปรแกรมรุ่นเก่า ซึ่งต้องการข้อมูลบางอย่างตรงตำแหน่งที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

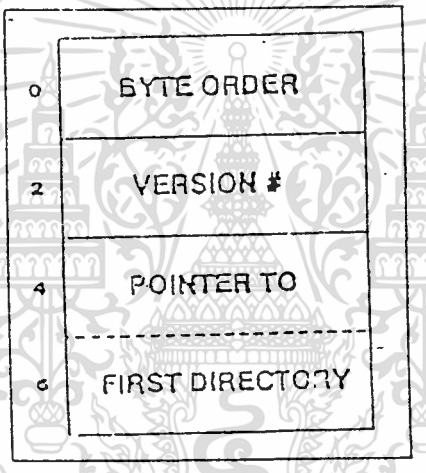
กำหนด กลับพบว่าข้อมูลในตำแหน่งนั้นเปลี่ยนไปหรือหาข้อมูลไม่พบเลยก็ได้ ผลลัพธ์ที่ได้ก็คือการทำงานที่ผิดพลาด

โครงสร้างทั่วไปของ TIFF

แฟ้มข้อมูลกราฟิก TIFF หนึ่งๆ จะประกอบด้วยส่วนประกอบหลัก 3 ส่วนดังต่อไปนี้

1. Header Field

ในส่วนหัวนี้จะมีความยาว 8 ไบท์ ประกอบด้วยข้อมูลดังแสดงในรูปที่ 4

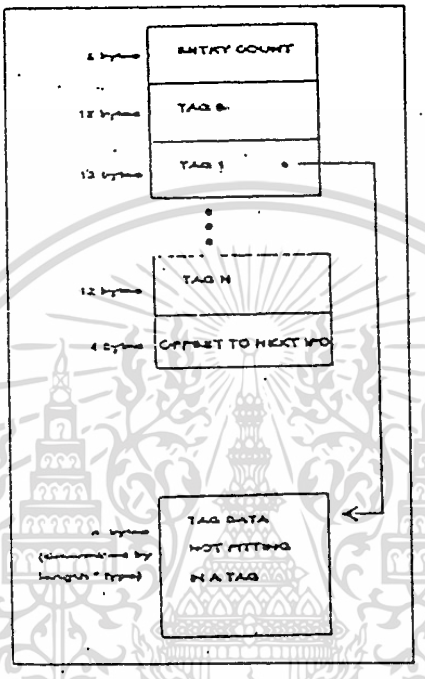


รูปที่ 4 The TIFF Header

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. The Image File Directory (IFD)

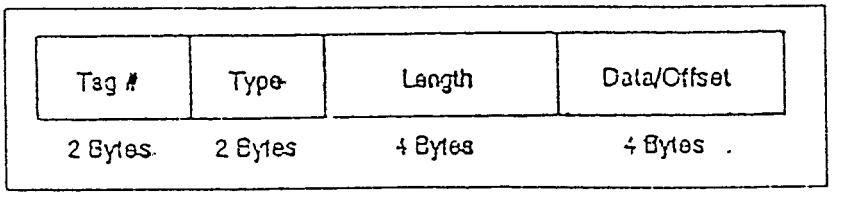
ในหนึ่ง IFD จะประกอบด้วย 3 ส่วนคือ Entry count, Tag ต่างๆ แต่ละอันยาว 12 ไบท์เรียงต่อกัน และส่วนที่จะชี้ไปยังตำแหน่งของ IFD ชุดต่อไป ส่วนนี้ยาว 4 ไบท์ แสดงในรูปที่ 5



รูปที่ 5 Image File Directoy

3.The Tags Field

สามารถที่ Tag ที่แตกต่างกันได้ถึง 50 Tags โดยแต่ละ Tag จะมีความยาว 12 ไบท์ ซึ่งจะประกอบด้วย 4 ส่วนคือ Tag number, Type of tag value, Number of values in tag และ Tag value of offset



รูปที่ 6 The 12 bytes of a TIFF tag

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่าง Tag ที่สำคัญและคำอธิบาย

1. The ImageWidth Tag

ImageWidth Tag = 256(100H)

Type = SHORT or LONG

Number of distinct values = 1

Value range : All positive integers in Type

ImageWidth บอกความกว้างของรูปภาพมีหน่วยเป็น Pixels ความกว้างจะยึดถือตามแกน X หรือตามแนวนอน ความกว้างของภาพอาจเรียกอีกอย่างว่าเป็นจำนวนคอลัมน์ในรูปภาพ Tag นี้เป็น Tag ที่สำคัญเพราะจะไม่สามารถอ่านรูปภาพได้เลยหากไม่ทราบค่าความกว้างของรูปเสียก่อน

2. The ImageLength Tag

ImageLength Tag = 257 (101H)

Type = SHORT or LONG

Number of distinct values = 1

Value range : All positive interger in Type

ImageLength บอกความยาวของรูปภาพมีหน่วยเป็น Pixels ความยาวจะยึดถือตามแกน Y หรือตามแนวตั้ง ความยาวของรูปภาพอาจเรียกอีกอย่างว่าเป็นจำนวนแถวในรูปภาพ สำหรับ Tag นี้ก็เป็น Tag ที่สำคัญเช่นเดียวกัน เพราะจะไม่สามารถอ่านรูปภาพได้เลย หากไม่ทราบความยาวของรูป

3. The StripOffsets Tag

StripOffsets Tag = 273 (111H)

Type = SHORT or LONG

Number of distinct values·

= one per strip for single plane bitmaps

= numbec of planes x strips/plane for multi-plane bitmaps

Vulue range : All positive integers in Type

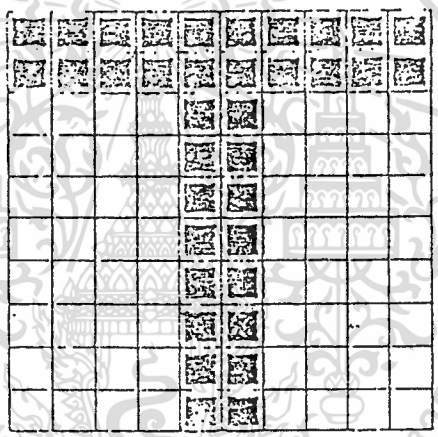
StripOffsets เป็น Tag ที่สำคัญและจำเป็นต้องทราบค่า เพราะ Tag นี้จะบอกตำแหน่งว่าส่วนของรูปภาพเริ่มต้นที่ตำแหน่งใด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในการเปิดอ่านแฟ้มข้อมูลกราฟิก TIFF นี้ ผู้ใช้จะต้องทราบค่าที่สำคัญอย่างน้อย 3 ค่าคือ ความกว้าง ความยาว และตำแหน่งของข้อมูลภาพที่เก็บในแฟ้มข้อมูลชนิดนี้ ซึ่งก็คือค่าต่างๆ ที่เก็บอยู่ใน Tag ทั้ง 3 ที่กล่าวมาแล้วข้างต้นนั่นเอง

ตัวอย่างแฟ้มข้อมูลแบบ TIFF

An Intel format 10 x 10 pixel Black and White TIFF Image



300 DPI Image
BINARY (Line art) image
No data compression
INTEL byte order

Used	Tag #	Tag Name	Value
__x__	0FEH	NewSubfileType	__1___
__x__	100H	ImageWidth	__10__
__x__	101H	ImageLength	__10__
__x__	102H	BitsPerSample	__1___
__x__	103H	Compression	__1___
__x__	106H	PhotometricInterp.	__0___
__x__	107H	Thresholding	__1___
__x__	111H	StripOffsets	_@198
__x__	115H	SamplesPerPixel	__1___
__x__	116H	RowsPerStrip	__10__
__x__	117H	StripByteCounts	__20__
__x__	11AH	XResolution	_300__
__x__	11BH	YResolution	_300__
__x__	128H	ResolutionUnit	__2___

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

I. Header

Offset	Value	
000	49 49	Note byte order is INTEL format (49 is ASCII 'I').
002	2A 00	Version #42; Note byte reversal.
004	08 00 00 00	Offset to first IFD.

II. Image File Directory

008 0E 00 14 tags in this directory

	Offset	Tag	Type	Length	Value/Offset	
Tag 1	00A	FE 00	04 00	01 00 00 00	00 00 00 00	(NewSubfileType = Full Res)
Tag 2	016	00 01	03 00	01 00 00 00	0A 00 00 00	(Image Width = 10 pixels)
Tag 3	022	01 01	03 00	01 00 00 00	0A 00 00 00	(Image Length = 10 pixels)
Tag 4	02E	02 01	03 00	01 00 00 00	01 00 00 00	(BitsPerSample = 1)
Tag 5	03A	03 01	03 00	01 00 00 00	01 00 00 00	(No compression, pack data)
Tag 6	046	06 01	03 00	01 00 00 00	00 00 00 00	(Black is 1)
Tag 7	052	07 01	03 00	01 00 00 00	01 00 00 00	(Thresholding type = Line Art)
Tag 8	05E	11 01	04 00	01 00 00 00	C6 00 00 00	(Offset to raster data)
Tag 9	06A	15 01	03 00	01 00 00 00	01 00 00 00	(1 sample/pixel)
Tag 10	076	16 01	04 00	01 00 00 00	0A 00 00 00	(10 Rows per Strip)
Tag 11	082	17 01	04 00	01 00 00 00	14 00 00 00	(20 Bytes per Strip)
Tag 12	08E	1A 01	05 00	01 00 00 00	B6 00 00 00	(Offset to X resolution)
Tag 13	09A	1B 01	05 00	01 00 00 00	BE 00 00 00	(Offset to Y resolution)
Tag 14	0A6	28 01	03 00	01 00 00 00	02 00 00 00	(Resolution units are inches)

082 00 00 00 00 (No more Image File Directories)

III. Extended tag information (as needed)

086 2C 01 00 00 01 00 00 00 (300 dpi X resolution)
08E 2C 01 00 00 01 00 00 00 (300 dpi Y resolution)

IV. Raster data (as needed)

0C6	FF C0	(10 black pixels)
0C8	FF C0	(10 black pixels)
0CA	0C 00	(2 black pixels)
0CC	0C 00	(2 black pixels)
0CE	0C 00	(2 black pixels)
0D0	0C 00	(2 black pixels)
0D2	0C 00	(2 black pixels)
0D4	0C 00	(2 black pixels)
0D6	0C 00	(2 black pixels)
0D8	0C 00	(2 black pixels)

Strip offset pointers (as needed)

None needed.

Strip byte counts (as needed)

None needed.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ข

รหัสแอสกี้ของอักขระในรหัส สมอ

-ตัวเลขอารบิก(0-9)

รหัสแอสกี้	อักขระ
48	0
49	1
50	2
51	3
52	4
53	5
54	6
55	7
56	8
57	9

-อักขระภาษาไทย(ก-ฮ)

รหัสแอสกี้	อักขระ
161	ก
162	ข
163	ฃ
164	ค
165	ฅ
166	ฆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รหัสแอสกี้	อักขระ
167	ง
168	จ
169	ฉ
170	ช
171	ซ
172	ฌ
173	ญ
174	ฎ
175	ฏ
176	ฐ
177	ฑ
178	ฒ
179	ณ
180	ด
181	ต
182	ถ
183	ท
184	ธ
185	น
186	บ
187	ป
188	ผ
189	ฝ
190	พ
191	ฟ
192	ภ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

193	ม
194	ย
195	ร
196	ฤ
197	ล
198	ฦ
199	ว
200	ศ
201	ษ
202	ส
203	ห
204	ฬ
205	อ
206	ฮ

-ตัวเลขไทย (0-๙)

รหัสแอสกี้	อักขระ
240	0
241	๑
242	๒
243	๓
244	๔
245	๕
246	๖
247	๗
248	๘
249	๙

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

-อักขระพิเศษ

รหัสแอสกี้	อักขระ
207	ๆ
208	ะ
209	ั
210	า
211	ำ
212	ิ
213	ี
214	ึ
215	ื
216	ุ
217	ู
224	เ
225	แ
226	โ
227	ใ
228	ไ
229	็
230	่
231	้
232	๊
233	๋
234	์
235	๎
236	๏

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ค

การใช้งานโปรแกรมเครื่องตรวจกวาดภาพ

(Photo Styler Version 1.1a)

การใช้งานเครื่องตรวจกวาดภาพ (Scanner Hewlett Packard) จะต้องประกอบด้วยเครื่องมือ 2 ส่วนใหญ่ๆ คือ

1.ส่วน Hardware ประกอบด้วย

- เครื่องตรวจกวาดภาพ (Hewlett Packard Scanner)
- เครื่องไมโครคอมพิวเตอร์ ที่มีขนาดหน่วยความจำไม่น้อยกว่า 640 Khyte และ Harddisk ขนาดไม่น้อยกว่า 10 Mbyte

2.ส่วน Software ประกอบด้วย

- โปรแกรม Photo Styler Version 1.1a ซึ่งทำงานบน Window
- โปรแกรม ไดรเวอร์ ของ Photo Styler ซึ่งมีชื่อว่า Sjdriver.sys

ขั้นตอนการเตรียมโปรแกรมเพื่อใช้งาน

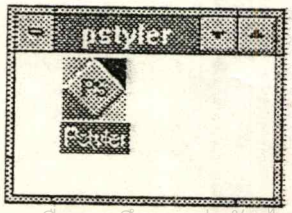
1.ต้องจัดเตรียมชุดโปรแกรม Phyto Styler Version 1.1a ลงบน Harddisk โดยการทำตามขั้นตอนการติดตั้งโปรแกรมของ โปรแกรม Photo Styler ซึ่งโปรแกรมจะทำการกำหนดข้อกำหนดต่างๆ ของอุปกรณ์ (ลักษณะเครื่องไมโครคอมพิวเตอร์ ชนิดของจอภาพ (Monitor) แป้นพิมพ์ (Keybord) เครื่องพิมพ์ (Printer) เครื่องจุดภาพ (plotter) อุปกรณ์เลื่อนตำแหน่ง (Mouse) ช่องทางการติดต่อ (Port) ให้ตรงกับการใช้งานจริง

2.หลังจากนั้นให้จัดการ Install Utility เพื่อจัดส่วนโปรแกรม Sjdriver.sys ให้กับเครื่องไมโครคอมพิวเตอร์ โดยทำการเติมบรรทัด DEVICE=C:\SJDRIVER.SYS ลงในบรรทัดของไฟล์Config.sys แล้วทำการ Reboot เครื่องเตรียมการใช้งานโปรแกรม

ขั้นตอนการใช้งานโปรแกรม Photo Styler Version 1.1a

1.ต้อง Boot เครื่องไมโครคอมพิวเตอร์โดยผ่านส่วนของ Sjdriver.sys จากการกำหนดจากไฟล์ Config.sys

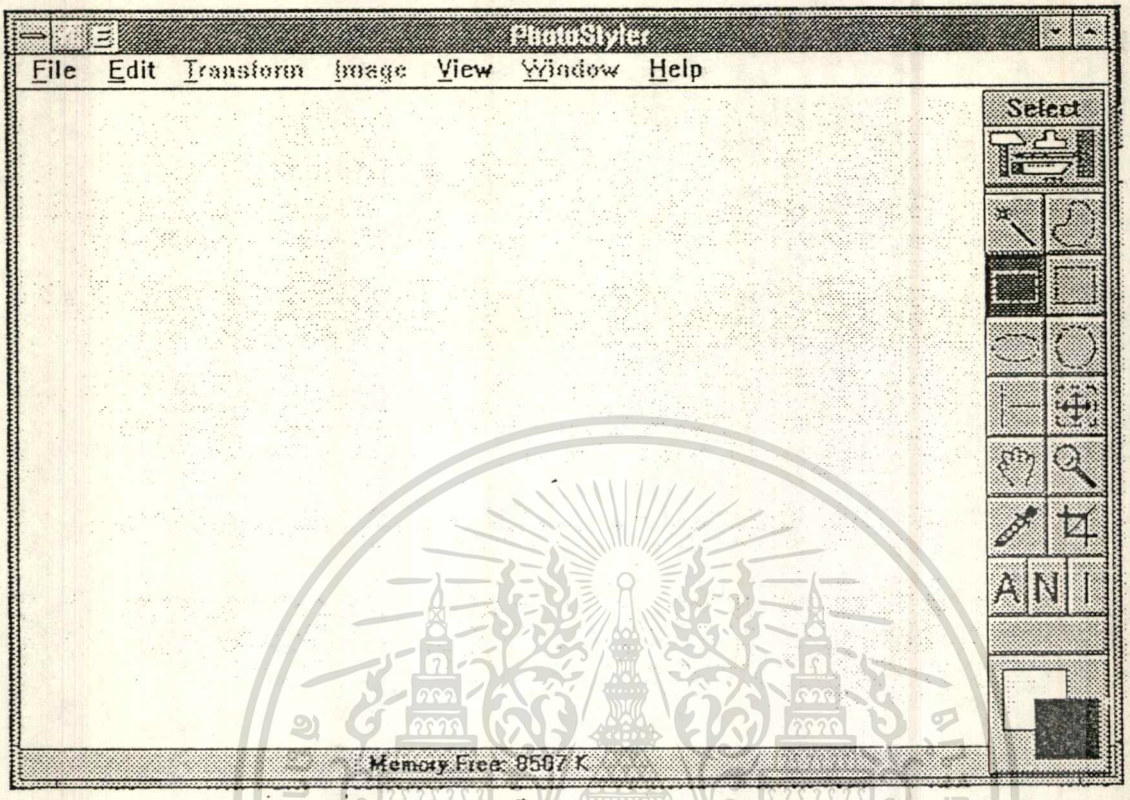
2.จากนั้นเรียกโปรแกรม Windows โดยคำสั่ง C:\>win <enter> และทำการเรียกโปรแกรม Photo Styler จากไอคอนดังรูปที่ ค-1



รูปที่ ค-1 แสดงไอคอนของโปรแกรม Photo Styler Version 1.1a

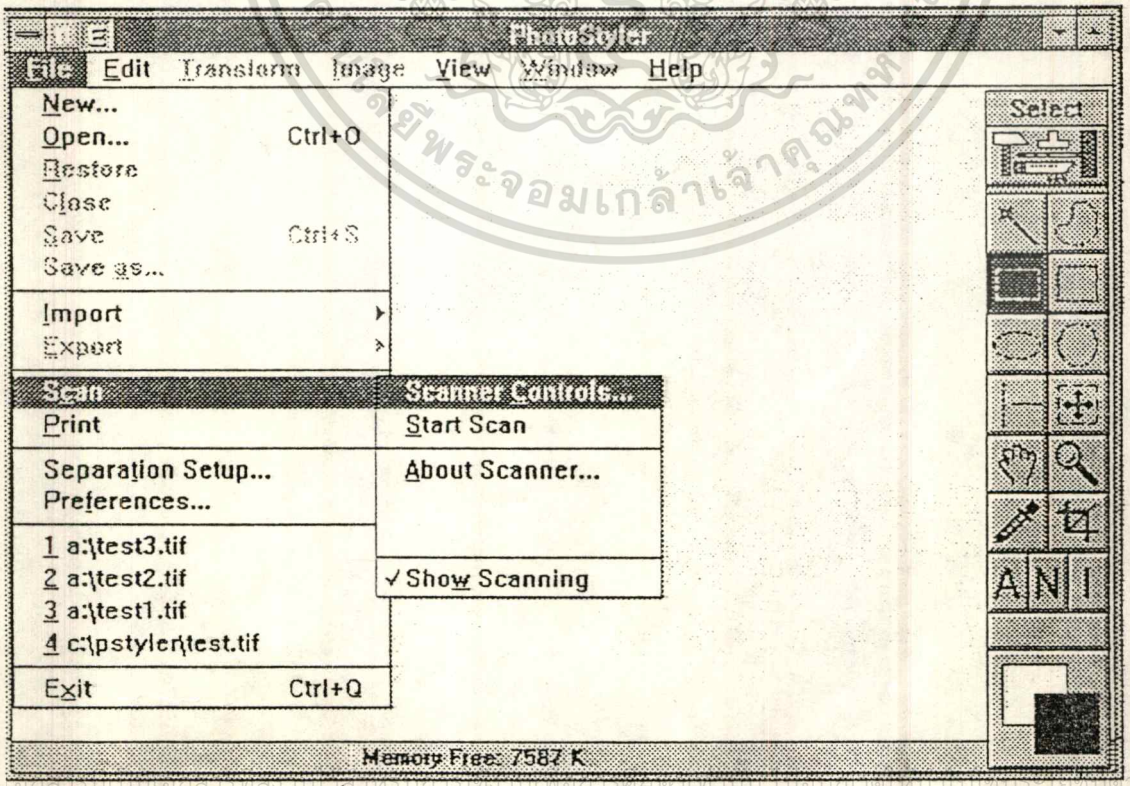
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.เมื่อโปรแกรม Photo Styler Version 1.1a ทำงานจะมีหน้าต่างการทำงานดังรูปที่ ค-2



รูปที่ ค-2 รูปแสดงหน้าต่างของโปรแกรม Photo Styler Version 1.1a

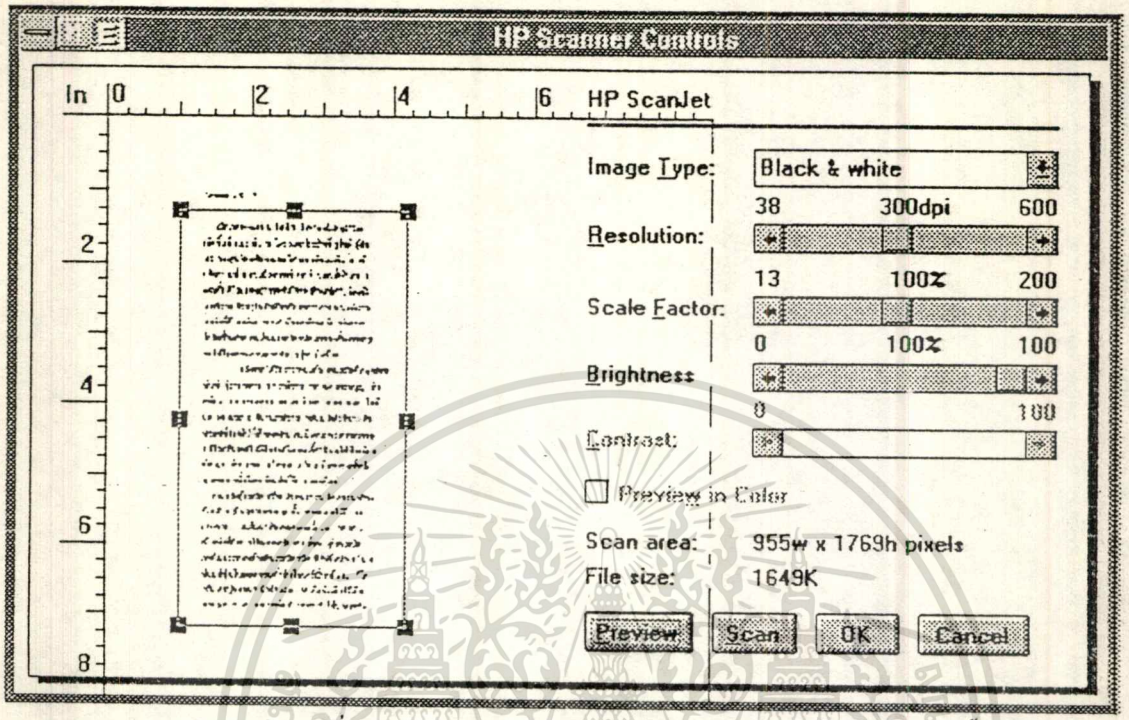
4.จากนั้นทำการแสกนหน้าเอกสารจากการวางหน้ากระดาษลงบนเครื่องตรวจกวาดภาพแล้ว เลือกเมนูเพื่อการแสกนหน้าเอกสารดังรูปที่ ค-3



รูปที่ ค-3 รูปแสดงเมนูการแสกนหน้ากระดาษเอกสาร

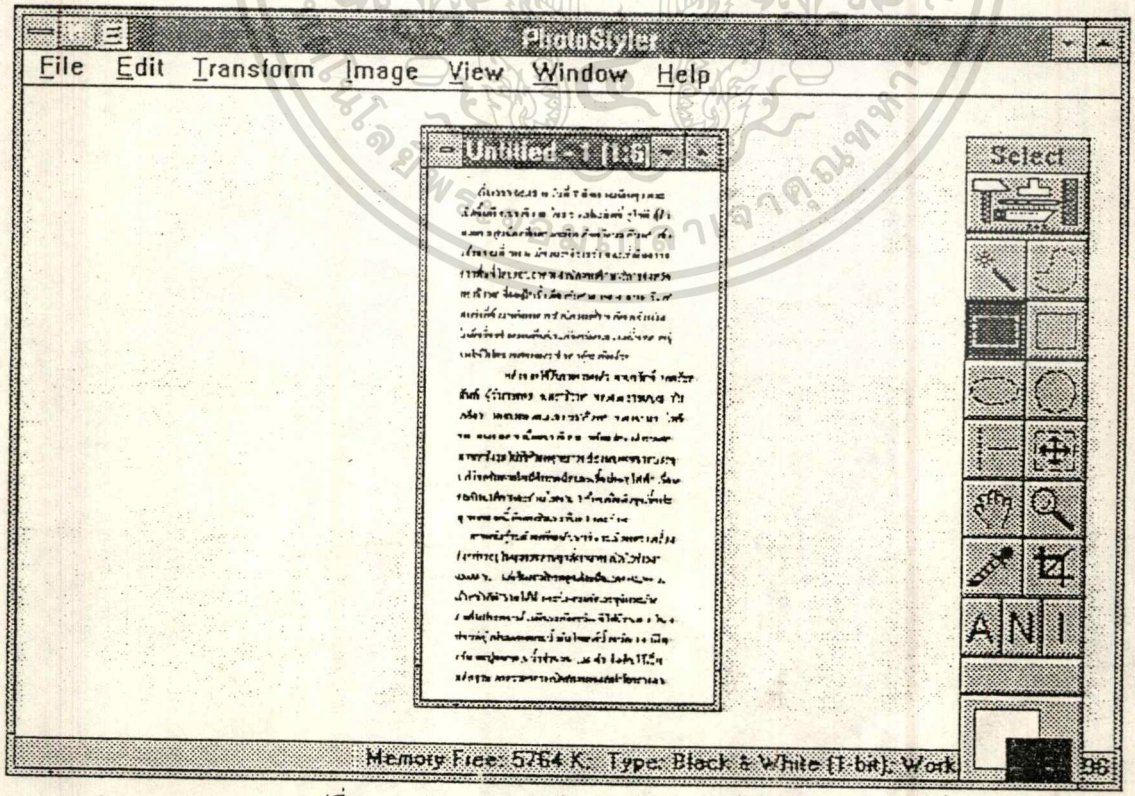
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.ซึ่งในการวิจัยนี้กำหนดอัตราส่วนการแสกนหน้าเอกสารดังรูปที่ ค-4



รูปที่ ค-4 รูปแสดงอัตราส่วนการแสกนหน้าเอกสารที่ใช้ในการวิจัยนี้

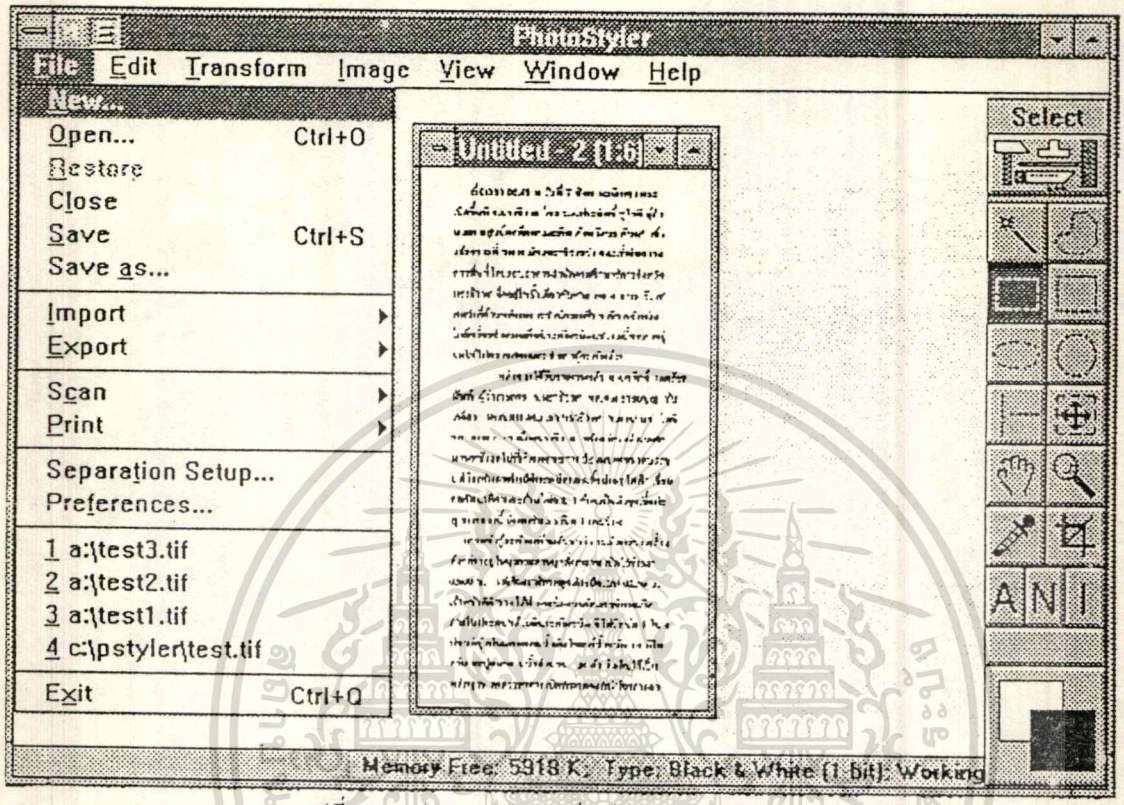
6.จากนั้นจะได้ภาพหน้าเอกสารหลังจากการแสกนตามอัตราส่วนดังรูปที่ ค-5



รูปที่ ค-5 รูปแสดงภาพหน้าเอกสารจากการแสกนตามอัตราส่วนที่กำหนด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7.จากนั้นทำการเก็บข้อมูลภาพหน้าเอกสารที่แสกนได้โดยการเลือกเมนูตามรูปที่ ค-6



รูปที่ ค-6 รูปแสดงเมนูการเก็บข้อมูลภาพหน้าเอกสารลงไฟล์

8.เมื่อทำการใช้งานโปรแกรม Photo Styler Version 1.1a แล้วเสร็จ เราสามารถออกจากการใช้งานโปรแกรม Photo Styler 1.1a ได้โดยการเลือกเมนู Exit หรือกดปุ่ม Ctrl+Q จากรูปที่ ค-6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ง

ตัวโปรแกรมของระบบการรู้จำ

1.ตัวโปรแกรม netsave.c

สำหรับฝึกสอนโครงข่ายนิวรอนค้วยอักขระต้นแบบบนเครื่องมินิคอมพิวเตอร์

```
#include<stdio.h>
#include<ctype.h>
#include<stdlib.h>
#include<math.h>
#include<sys/times.h>
#include<malloc.h>
#include<string.h>
#include<float.h>
#include<signal.h>

#define I    1600
#define H    230
#define OUT   87
#define LOOP  100 /*LOOP is integer for saving weight,bias,delta */

void saveb(char *argv[]);
void test(char *argv[]);
void freew(void);
void savew(char *argv[]);
void createb(void);
void initdel(void);
void loaddel(char *argv[]);
void savedel(char *argv[]);
void loadw(char *argv[]);
void loadb(char *argv[]);

float  n,alpha;
double errorb;
double bh1[(H+1)],bh2[(H+1)],bo[(OUT+1)];
double deltah1[(H+1)],deltah2[(H+1)],delto[(OUT+1)];
double dbh1[(H+1)],dbh2[(H+1)],dbo[(OUT+1)];
double Outputh1[(H+1)],Outputh2[(H+1)],Outputo[(OUT+1)];
double Inputh1[(H+1)],Inputh2[(H+1)],Inputo[(OUT+1)];
double *wh1[I+1],*dwh1[I+1],*wh2[H+1],*dwh2[H+1],*wo[H+1],*dwo[H+1];
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
int    count,ltimes,xpos,ypos;
int    ind,oxpos[561],oypos[561];
double Input[(I+1)];
char   str[51];/*str size must wider than the width of the widest
character */

main(int argc,char *argv[20])
{
int    load;
int    i,j,all,tempn1,tempn2;
int    col,row,Tar,T1,T2,over;
FILE   *fp;
char   ch;
double TAR[(OUT+1)],rms,diff,errorp,del,deltap;
double temph1,temph2,tempo;
double dtemph1,dtemph2,dtempo;
long   start,end;

            times(&start);
            if(argc!=13)
            {
              printf("\nYou enter wrong !");
              printf("\nEnter [Program name] + [file input.txt]+");
              printf("\n      [file wh1.txt] + [file  dwh1.txt]+");
              printf("\n      [file wh2.txt] + [file  dwh2.txt]+");
              printf("\n      [file wo.txt ] + [file  dwo.txt]+");
              printf("\n      [file bh1.txt] + [file  bh2.txt]+");
              printf("\n      [file bo.txt ] + [file delta.txt]");
              printf("\n              [0 or 1] /* 0 mean intial all, 1 mean load old*/");
              printf("\n");
              exit(1);
            }
            n=0.1;
            alpha=0.5;
            load=atoi(argv[12]);
 /* Created weight    */

            for(i=1;i<=I;i++)
            {
              if(( wh1[i]=(double * )malloc((H+1)*sizeof(double))) == NULL)
              {
                printf("\nOut of memory for allocating wh1[I][H]");
                exit(1);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  }
  for(j=1;j<=H;j++)
  {
    whl[i][j]=(rand() % 1000)*0.0001;
    if(rand() % 2)
     whl[i][j]=-1.0*whl[i][j];
  }
}
for(i=1;i<=I;i++)
{
  if(( dwhl[i]=(double * )malloc((H+1)*sizeof(double))) == NULL)
  {
    printf("\nOut of memory for allocating dwhl[I][H]");
    exit(1);
  }
  for(j=1;j<=H;j++)
  {
    dwhl[i][j]=0.0;
  }
}
/* THIS  add */
for(i=1;i<=H;i++)
{
  if(( wh2[i]=(double * )malloc((H+1)*sizeof(double))) == NULL)
  {
    printf("\nOut of memory for allocating wh2[H][H]");
    exit(1);
  }
  for(j=1;j<=H;j++)
  {
    wh2[i][j]=(rand() % 1000)*0.0001;
    if(rand() % 2)
     wh2[i][j]=-1.0*wh2[i][j];
  }
}
for(i=1;i<=H;i++)
{
  if(( dwh2[i]=(double * )malloc((H+1)*sizeof(double))) == NULL)
  {
    printf("\nOut of memory for allocating dwh2[I][H]");
    exit(1);
  }
  for(j=1;j<=H;j++)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    {
      [illegible];
    }
  }
  /* THIS */
  for(i=1;i<=H;i++)
  {
    if(( wo[i]=(double * )malloc((OUT+1)*sizeof(double))) == NULL)
    {
      printf("\nOut of memory for allocating wo[H][OUT]");
      exit(1);
    }
    for(j=1;j<=OUT;j++)
    {
      wo[i][j]=(rand() % 1000)*0.0001;
      if(rand() % 2)
       wo[i][j]=-1.0*wo[i][j];
    }
  }
  for(i=1;i<=H;i++)
  {
    if(( dwo[i]=(double * )malloc((OUT+1)*sizeof(double))) == NULL)
    {
      printf("\nOut of memory for allocating dwo[H][OUT]");
      exit(1);
    }
    for(j=1;j<=OUT;j++)
    {
      dwo[i][j]=0.0;
    }
  }
if(load==0)
{
  createb();
  initdel();
  errorb=0.0;
  times=0;
  delhmp=0.0;
}
else
{
  loadb(argv);
  loaddel(argv);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            loadw(argv);
            deltmp=errorb;
          }
            errorp=0.0;
            j=0;
            tempn1=I+4+sqrt(I);
            tempn2=I+sqrt(I);
all=0;
del=0.0;
count=0;
do
{
 errorb=0.0;
 if( (fp=fopen(argv[1],"rb")) == NULL)
 {
        printf("\nError in open file %s",argv[1]);
        exit(1);
 }
 j=1;
 while(!feof(fp))
 {
        for(i=1;i<=I;i++)
        {
                Input[i]=0.0;
        }
        j=1;
        while(((fgets(str,sizeof(str),fp))!=NULL)&(j<=sqrt(I)))
        {
                if(ferror(fp))
                {
                        printf("\nError in reading %s",argv[1]);
                        exit(1);
                }
                else
                {
                        if(str[0]!='T')
                        {
                                for(i=0;i<sqrt(I);i++)
                                {
                                    if((str[i]=='1')|(str[i]=='0'))
                                    {
                                        Input[((40*j)+i)-39]=(str[i]-48)*1.0;
                                    }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                                str[i]='0';
                                }
                        }
                        else
                        {
                                break;
                        }
                }
                j=j+1;/*while(fgets)*/
        }/*end while(fgets)*/
/*      printf("j=%d",j);*/
        /*set number target whether j more than I */
        if(j>sqrt(I))
        {
                do
                {
                        fgets(str,sizeof(str),fp);
                }while(str[0]!='T');
        }
        T1=(str[1]-48)*10;
        T2=(str[2]-48);
        Tar=T1+T2;
        /* end set number target */
        /*show Input[1->I]*/
/*      j=1;*/
/*      printf("\n");*/
/*      for(i=1;i<=sqrt(I);i++)*/
/*      {*/
/*              for(j=1;j<=sqrt(I);j++)*/
/*              {*/
/*                      printf("%1.0lf",Input[(40*(i-1))+j]);*/
/*              }*/
/*              printf("\n");*/
/*              getchar();*/
/*      }*/
/*      printf("\nTar == %d",Tar);*/
/*      getchar();*/
        /*end show*/
        for(i=1;i<=OUT;i++)
        {
                if(i==Tar)
                {
                        TAR[i]=1.0;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            }
            else
            {
                    TAR[i]=0.0;
            }
    }

/* This point can use Input[1-->I] */
        /* Forward && Backward */

        /* Forward phase */
        /*1Compute input for every node in hidden layer 1 */
                for(i=1;i<=H;i++) /* hidden layer 1 to H = col 1 to H hidden weight */
                {
                 temph1=0.0;
                 for(j=1;j<=I;j++)/* input value 1 to I of hidden layer */
                 {
                  temph1=temph1+(Input[j]*(wh1[j][i]));
                 }
                 Inputh1[i]=temph1+bh1[i];
                } -
        /* end 1 */
        /*2Compute output for every node in hidden layer */
                for(i=1;i<=H;i++)
                {
                        if( (Inputh1[i]>-88.00)&&(Inputh1[i]<88.00) )
                        {
                         Outputh1[i]=1.0/(1.0+exp(-Inputh1[i]));
                        }
                        else
                        {
                         if(Inputh1[i]>=88.00)
                            Outputh1[i]=1.000000;
                         else
                            if(Inputh1[i]<=-88.00)
                               Outputh1[i]=0.000000;
                        }
                }
        /* end 2 */
        /* THIS ADD */
        /* *Compute input for every node in hidden layer 2 */
                for(i=1;i<=H;i++) /* hidden layer 2 to H = col 1 to H hidden weight */
                {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        temph2=0.0;
        for(j=1;j<=H;j++)/* input value 1 to H of hidden layer */
        {
         temph2=temph2+(Outputh1[j]*(wh2[j][i]));
        }
        Inputh2[i]=temph2+bh2[i];
        }
/* *Compute output for every node in hidden layer 2 */
        for(i=1;i<=H;i++)
        {
                if( (Inputh2[i]>-88.00)&&(Inputh2[i]<88.00) )
                {
                 Outputh2[i]=1.0/(1.0+exp(-Inputh2[i]));
                }
                else
                {
                 if(Inputh2[i]>=88.00)
                    Outputh2[i]=1.000000;
                 else
                    if(Inputh2[i]<=-88.00)
                      Outputh2[i]=0.000000;
                }
        }
/* end 2 */
/* END THIS ADD */
/*3Compute input for every node in output layer */
        for(i=1;i<=OUT;i++) /* node 1 to OUT in output layer */
        {
         tempo=0.0;
         for(j=1;j<=H;j++) /* input value of node i.in output layer from hidden layer (node j=1 to j=H) */
         {
          tempo=tempo+(Outputh2[j]*(wo[j][i]));
         }
         Inputo[i]=tempo+bo[i];
        }
/* end 3 */
/*4Compute Output for every node in output layer */
        for(i=1;i<=OUT;i++)
        {
         if( (Inputo[i]>-88.00)&&(Inputo[i]<88.00) )
         {
                Outputo[i]=1.0/(1.0+exp(-Inputo[i]));
         }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            else
            {
                    if(Inputo[i]>=88.00)
                      Outputo[i]=1.000000;
                    else
                     if(Inputo[i]<=-88.00)
                      Outputo[i]=0.000000;
            }
           }
/* end 4 */
/*5Compute different between target and output value */
           ms=0.0;
           for(i=1;i<=OUT;i++)
           {
            diff=0.0;
            diff=TAR[i]-Outputo[i];
            ms=ms+(diff*diff);
           }
           ms=ms/2.00;
           erroep=ms;
/* end 5 */

/* Backward Phase */
          /*6Compute deltao[1-->OUT] in output layer */
          for(i=1;i<=OUT;i++)
          {
           deltao[i]=Outputo[i]*(1.0-Outputo[i])*(TAR[i]-Outputo[i]);
          }
          /* end 6 */
          /*7Compute dwo[H][OUT] that point from hidden2 to output */
           for(i=1;i<=OUT;i++)
           {
                    for(j=1;j<=H;j++)
                    {
                     dwo[j][i]=((n*deltao[i])*Outputh2[j])+(alpha*dwo[j][i]);
                     wo[j][i]=wo[j][i]+dwo[j][i];
                    }
           }
          /* end 7 */
          /* THIS ADD */
      /*8compute deltah2[1->H] in hidden layer 2 */
          for(i=1;i<=H;i++)
          {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                dtemph2=0.0;
                for(j=1;j<=OUT;j++)
                {
                        dtemph2=dtemph2+(deltao[j]*wo[i][j];
                }
                deltah2[i]=dtemph2*Outputh2[i]*(1.0-Outputh2[i]);
               }
/*          printf("\nEnd compute deltah2[1-->H] in module 8"); */
            /* end 8 */

            /*9compute dwh2[H][H] */
/*          printf("\nStart compute dwh2[H][H] in module 9"); */
            for(i=1;i<=H;i++)
            {
                for(j=1;j<=H;j++)
                {
                dwh2[j][i]=((n*deltah2[i])*Outputh1[j])+(alpha*dwh2[j][i]);
                wh2[j][i]=wh2[j][i]+dwh2[j][i];
                }
            }
/*          printf("\nEnd compute dwh2[H][H] in module 9"); */
            /* end 9 */
            /* END THIS */

            /*8compute deltah1[1->H] in hidden layer 1 */
            for(i=1;i<=H;i++)
            {
             dtemph1=0.0;
             for(j=1;j<=H;j++)
             {
                dtemph1=dtemph1+(deltah2[j]*wh2[i][j];
             }
             deltah1[i]=dtemph1*Outputh1[i]*(1.0-Outputh1[i]);
            }
/*          printf("\nEnd compute deltah1[1-->H] in module 8"); */
            /* end 8 */
            /*9compute dwh1[I][H] */
/*          printf("\nStart compute dwh1[i][H] in module 9"); */
            for(i=1;i<=H;i++)
            {
                for(j=1;j<=I;j++)
                {
                dwh1[j][i]=((n*deltah1[i])*Input[j])+(alpha*dwh1[j][i]);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                        wh1[j][i]=wh1[j][i]+dwh1[j][i];
                      }
                  }
/*          printf("\nEnd compute dwh1[I][H] in module 9"); */
                  /* end 9 */
                  /*10compute dbo[1->OUT] in output layer */
/*          printf("\nStart compute dbo[1-->OUT] in output layer in module 10"); */
                   for(i=1;i<=OUT;i++)
                   {
                        dbo[i]=(n*delta0[i])+(alpha*dbo[i]);
                        bo[i]=bo[i]+dbo[i];
                   }
/*          printf("\nEnd compute dbo[1-->OUT] in module 10"); */
                  /* end 10 */
                  /*11compute dbh1[1->H] in hidden layer 1 */
                   for(i=1;i<=H;i++)
                   {
                        dbh1[i]=(n*deltah1[i])+(alpha*dbh1[i]);
                        bh1[i]=bh1[i]+dbh1[i];
                   }
                  /* end 11 */
                  /*12compute dbh2[1->H] in hidden layer 2 */
                   for(i=1;i<=H;i++)
                   {
                        dbh2[i]=(n*deltah2[i])+(alpha*dbh2[i]);
                        bh2[i]=bh2[i]+dbh2[i];
                   }
                  /* end 12 */
          /* end backward phase */
          /* End Forward && Backward */
          j=0;
  errorb=errorb+errorp;
 }/* end while(fread()) */
 fclose(fp);
 ktimes=ktimes+1;
 del=errorb-delkmp;
 delkmp=errorb;
 if(del<0.0)
 {
   del=-1.0*del;
 }
 printf("\nErrorb = %lf ",errorb);
 printf("LOOP Times  = %d  ",ktimes);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    printf("Del = %lf",del);
    if(count>=LOOP)
    {
    savew(argv);/* This function will save wh[I][H],dwh[I][H],wo[H][OUT],dwo[H][OUT] to file  */
    saveb(argv);/* This function will save bh[H],bo[OUT] to file bh.txt,bo.txt                  */
    savedel(argv);
    count=0;
    }
    else
    {
    count=count+1;
    }
}while( (errorb>0.05)&&(del>0.0002) );
    fclose(fp); /* Close file input pattern */
    savew(argv);/* This function will save wh[I][H],dwh[I][H],wo[H][OUT],dwo[H][OUT] to file  */
    saveb(argv);/* This function will save bh[H],bo[OUT] to file bh.txt,bo.txt                  */
    savedel(argv);
    freew();
    times(&end);
    end-=start;
    printf("\nProcess took time = %02d:%02d:%02d\n",
            (end/60/60)%60,(end/60)%60,end%60);
    return(0);
}/* end main */

void freew(void)
{
int i;
 /* start  free memory of weight */
 for(i=0;i<=H;i++)
 {
  free(wh1[i]);
  free(dwh1[i]);
 }
 for(i=0;i<=H;i++)
 {
  free(wh2[i]);
  free(dwh2[i]);
 }
 for(i=0;i<=OUT;i++)
 {
  free(wo[i]);
  free(dwo[i]);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
 }
 /* end free allocating memory for weight && delta weight */
}

void savedel(char *argv[])
{
int         i;
double      del;
FILE        *fp;
/* start save delta to file */
            if((fp=fopen(argv[11],"wb"))==NULL)
            {
                        printf("\nError in creating file %s",argv[11]);
                        exit(1);
            }
            /* H1 */
            for(i=1;i<=H;i++)
            {
                        del=deltah1[i];
                        fwrite(&del,sizeof(double),1,fp);
            }
            for(i=1;i<=H;i++)
            {
                        del=Outputh1[i];
                        fwrite(&del,sizeof(double),1,fp);
            }
            for(i=1;i<=H;i++)
            {
                        del=dbh1[i];
                        fwrite(&del,sizeof(double),1,fp);
            }
            /*end H1*/
            /* H2 */
            for(i=1;i<=H;i++)
            {
                        del=deltah2[i];
                        fwrite(&del,sizeof(double),1,fp);
            }
            for(i=1;i<=H;i++)
            {
                        del=Outputh2[i];
                        fwrite(&del,sizeof(double),1,fp);
            }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        for(i=1;i<=H;i++)
        {
                del=dbh2[i];
                fwrite(&del,sizeof(double),1,fp);
        }
        /*end H2*/
        /*Out */
        for(i=1;i<=OUT;i++)
        {
                del=deltao[i];
                fwrite(&del,sizeof(double),1,fp);
        }
        for(i=1;i<=OUT;i++)
        {
                del=Outputo[i];
                fwrite(&del,sizeof(double),1,fp);
        }
        for(i=1;i<=OUT;i++)
        {
                del=dbo[i];
                fwrite(&del,sizeof(double),1,fp);
        }
        /*end Out */
        fwrite(&errorb,sizeof(double),1,fp);
        fwrite(&ltimes,sizeof(int),1,fp);
        fclose(fp);
}
void loaddel(char *argv[])
{
int     i;
double  del;
FILE    *fp;
/* start save delta to file. */
        if((fp=fopen(argv[11],"rb"))==NULL)
        {
                printf("\nError in reading file %s",argv[11]);
                exit(1);
        }
        /* HI */
        for(i=1;i<=H;i++)
        {
                fread(&del,sizeof(double),1,fp);
                deltah1[i]=del;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
}
for(i=1;i<=H;i++)
{
        fread(&del,sizeof(double),1,fp);
        Outputh1[i]=del;
}
for(i=1;i<=H;i++)
{
        fread(&del,sizeof(double),1,fp);
        dbh1[i]=del;
}
/*end H1*/
/* H2 */
for(i=1;i<=H;i++)
{
        fread(&del,sizeof(double),1,fp);
        deltah2[i]=del;
}
for(i=1;i<=H;i++)
{
        fread(&del,sizeof(double),1 fp);
        Outputh2[i]=del;
}
for(i=1;i<=H;i++)
{
        fread(&del,sizeof(double),1,fp);
        dbh2[i]=del;
}
/*end H2*/
/*Out */
for(i=1;i<=OUT;i++)
{
        fread(&del,sizeof(double),1,fp);
        deltao[i]=del;
}
for(i=1;i<=OUT;i++)
{
        fread(&del,sizeof(double),1,fp);
        Outputo[i]=del;
}
for(i=1;i<=OUT;i++)
{
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                fread(&del,sizeof(double),1,fp);
                dbo[i]=del;
        }
        /*end Out */
        fread(&errorb,sizeof(double),1,fp);
        fread(&ltimes,sizeof(int),1,fp);
        fclose(fp);
}

void savew(char *argv[])
{
int   i,j,fr,fo;
double weight;
FILE   *fph1,*fpdh1,*fph2,*fpdh2,*fpo,*fpdo;

 /* start save all weight to file */
   /* save wh1[I][H] to argv[2] */
   if((fph1=fopen(argv[2],"wb"))==NULL)
   {
     printf("\nError in creating file %s",argv[2]);
     exit(1);
   }
   for(i=1;i<=I;i++)
   {
     for(j=1;j<=H;j++)
     {
          weight=wh1[i][j];
          fwrite(&weight,sizeof(double),1,fph1);
          if(ferror(fph1))
          {
            printf("\nError in writing file %s",argv[2]);
            exit(1);
          }
     }
   }
   fclose(fph1);

   /* save dwh1[I][H] to argv[3] */
   if((fpdh1=fopen(argv[3],"wb"))==NULL)
   {
     printf("\nError in creating file %s",argv[3]);
     exit(1);
   }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
for(i=1;i<=I;i++)
{
  for(j=1;j<=H;j++)
  {
        weight=dwh1[i][j];
        fwrite(&weight,sizeof(double),1,fpdh1);
        if(ferror(fph1))
        {
          printf("\nError in writing file %s",argv[3]);
          exit(1);
        }
  }
}
fclose(fpdh1);
/* THIS ADD */
/* save wh2[H][H] to argv[4] */
if((fph2=fopen(argv[4],"wb"))==NULL)
{
  printf("\nError in creating file %s",argv[4]);
  exit(1);
}
for(i=1;i<=H;i++)
{
  for(j=1;j<=H;j++)
  {
        weight=wh2[i][j];
        fwrite(&weight,sizeof(double),1,fph2);
        if(ferror(fph2))
        {
          printf("\nError in writing file %s",argv[4]);
          exit(1);
        }
  }
}
fclose(fph2);

/* save dwh2[H][H] to argv[5] */
if((fpdh2=fopen(argv[5],"wb"))==NULL)
{
  printf("\nError in creating file %s",argv[5]);
  exit(1);
}
for(i=1;i<=H;i++)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
{
  for(j=1;j<=H;j++)
  {
        weight=dwh2[i][j];
        fwrite(&weight,sizeof(double),1,fpdh2);
        if(ferror(fph2))
        {
          printf("\nError in writing file %s",argv[5]);
          exit(1);
        }
  }
}
fclose(fpdh2);

/* THIS END ADD */

/* save wo[H][OUT] to argv[6] */
if((fpo=fopen(argv[6],"wb"))==NULL)
{
  printf("\nError in creating file %s",argv[6]);
  exit(1);
}
for(i=1;i<=H;i++)
{
  for(j=1;j<=OUT;j++)
  {
        weight=wo[i][j];
        fwrite(&weight,sizeof(double),1,fpo);
        if(ferror(fpo))
        {
          printf("\nError in writing file %s",argv[6]);
          exit(1);
        }
  }
}
fclose(fpo);

/* save dwo[H][OUT] to argv[7] */
if((fpdo=fopen(argv[7],"wb"))==NULL)
{
  printf("\nError in creating file %s",argv[7]);
  exit(1);
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    for(i=1;i<=H;i++)
    {
     for(j=1;j<=OUT;j++)
     {
          weight=dwo[i][j];
          fwrite(&weight,sizeof(double),1,fpdo);
          if(ferror(fpdo))
          {
           printf("\nError in writing file %s",argv[7]);
           exit(1);
          }
     }
    }
    fclose(fpdo);
 /* end save weight to file       */
}

void loadw(char *argv[])
{
int    i,j,fr,fc;
double weight;
FILE   *fph1,*fpdh1,*fph2,*fpdh2,*fpo,*fpdo;

 /* start save all weight to file */
    /* save wh1[I][H] to argv[2] */
    if((fph1=fopen(argv[2],"rb"))==NULL)
    {
     printf("\nError in reading file %s",argv[2]);
     exit(1);
    }
    for(i=1;i<=I;i++)
    {
     for(j=1;j<=H;j++)
     {
          fread(&weight,sizeof(double),1,fph1);
          wh1[i][j]=weight;
          if(ferror(fph1))
          {
           printf("\nError in reading file %s",argv[2]);
           exit(1);
          }
     }
    }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
fclose(fph1);

/* load dwh1[I][H] to argv[3] */
if((fpdh1=fopen(argv[3],"rb"))==NULL)
{
  printf("\nError in reading file %s",argv[3]);
  exit(1);
}
for(i=1;i<=I;i++)
{
  for(j=1;j<=H;j++)
  {
        fread(&weight,sizeof(double),1,fpdh1);
        dwh1[i][j]=weight;
        if(ferror(fpdh1))
        {
          printf("\nError in reading file %s",argv[3]);
          exit(1);
        }
  }
}
fclose(fpdh1);

/* THIS ADD */
/* load wh2[H][H] from argv[4] */
if((fph2=fopen(argv[4],"rb"))==NULL)
{
  printf("\nError in reading file %s",argv[4]);
  exit(1);
}
for(i=1;i<=H;i++)
{
  for(j=1;j<=H;j++)
  {
        fread(&weight,sizeof(double),1,fph2);
        wh2[i][j]=weight;
        if(ferror(fph2))
        {
          printf("\nError in reading file %s",argv[4]);
          exit(1);
        }
  }
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
fclose(fph2);

/* load dwh2[H][H] from argv[5] */
if((fpdh2=fopen(argv[5],"rb"))==NULL)
{
  printf("\nError in reading file %s",argv[5]);
  exit(1);
}
for(i=1;i<=H;i++)
{
  for(j=1;j<=H;j++)
  {
        fread(&weight,sizeof(double),1,fpdh2);
        dwh2[i][j]=weight;
        if(ferror(fpdh2))
        {
          printf("\nError in reading file %s",argv[5]);
          exit(1);
        }
  }
}
fclose(fpdh2);

/* THIS END ADD */

/* load wo[H][OUT] from argv[6] */
if((fpo=fopen(argv[6],"rb"))==NULL)
{
  printf("\nError in reading file %s",argv[6]);
  exit(1);
}
for(i=1;i<=H;i++)
{
  for(j=1;j<=OUT;j++)
  {
        fread(&weight,sizeof(double),1,fpo);
        wo[i][j]=weight;
        if(ferror(fpo))
        {
          printf("\nError in reading file %s",argv[6]);
          exit(1);
        }
  }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    }
    fclose(fpo);

    /* load dwo[H][OUT] from argv[7] */
    if((fpdo=fopen(argv[7],"rb"))==NULL)
    {
      printf("\nError in reading file %s",argv[7]);
      exit(1);
    }
    for(i=1;i<=H;i++)
    {
      for(j=1;j<=OUT;j++)
      {
            fread(&weight,sizeof(double),1,fpdo);
            dwo[i][j]=weight;
            if(ferror(fpdo))
            {
              printf("\nError in reading file %s",argv[7]);
              exit(1);
            }
      }
    }
    fclose(fpdo);
 /* end load weight from file      */

}

void loadb(char *argv[])
{
int    i,j,fn;
FILE   *fbh1,*fbh2,*fbo;
double bias;

 /* start load all weight from file */
    /* load bh1[H] from argv[8] */
    if((fbh1=fopen(argv[8],"rb"))==NULL)
    {
      printf("\nError in reading file %s",argv[8]);
      exit(1);
    }
    for(i=1;i<=H;i++)
    {
      fread(&bh1[i],sizeof(double),1,fbh1);
```

เอกสารนี้จัดทำขึ้นเพื่อการใช้งานสำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    if(ferror(fbh1))
    {
        printf("\nError in reading file %s",argv[8]);
        exit(1);
    }
  }
  fclose(fbh1);
  /* this add */
  /* load bh2[H] from argv[9] */
  if((fbh2=fopen(argv[9],"rb"))==NULL)
  {
    printf("\nError in reading file %s",argv[9]);
    exit(1);
  }
  for(i=1;i<=H;i++)
  {
    fread(&bh2[i],sizeof(double),1,fbh2);
    if(ferror(fbh2))
    {
        printf("\nError in reading file %s",argv[9]);
        exit(1);
    }
  }
  fclose(fbh2);
  /* end this add */

  /* load bo[OUT] to argv[10] */
  if((fbo=fopen(argv[10],"rb"))==NULL)
  {
    printf("\nError in reading file %s",argv[10]);
    exit(1);
  }
  for(i=1;i<=OUT;i++)
  {
    fread(&bo[i],sizeof(double),1,fbo);
    if(ferror(fbo))
    {
        printf("\nError in reading file %s",argv[10]);
        exit(1);
    }
  }
  fclose(fbo);
/* end load bias from file      */
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
}

void saveb(char *argv[])
{
int   i,j,fn;
FILE   *fbh1,*fbh2,*fbo;

/* start save all weight to file */
  /* save bh1[H] to argv[8] */
  if((fbh1=fopen(argv[8],"wb"))==NULL)
  {
    printf("\nError in creating file %s",argv[8]);
    exit(1);
  }
  for(i=1;i<=H;i++)
  {
    fwrite(&bh1[i],sizeof(double),1,fbh1);
    if(ferror(fbh1))
    {
          printf("\nError in writing file %s",argv[8]);
          exit(1);
    }
  }
  fclose(fbh1);
  /* this add */
  /* save bh2[H] to argv[9] */
  if((fbh2=fopen(argv[9],"wb"))==NULL)
  {
    printf("\nError in creating file %s",argv[9]);
    exit(1);
  }
  for(i=1;i<=H;i++)
  {
    fwrite(&bh2[i],sizeof(double),1,fbh2);
    if(ferror(fbh2))
    {
          printf("\nError in writing file %s",argv[9]);
          exit(1);
    }
  }
  fclose(fbh2);
  /* end this add */
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    /* save bo[OUT] to argv[10] */
    if((fbo=fopen(argv[10],"wb"))==NULL)
    {
      printf("\nError in creating file %s",argv[10]);
      exit(1);
    }
    for(i=1;i<=OUT;i++)
    {
      fwrite(&bo[i],sizeof(double),1,fbo);
      if(ferror(fbo))
      {
            printf("\nError in writing file %s",argv[10]);
            exit(1);
      }
    }
    fclose(fbo);
 /* end save bias to file        */
}

void initde1(void)
{
int i;

            /* H1 */
            for(i=1;i<=H;i++)
            {
             deltah1[i]=0.0;
             Outputh1[i]=0.0;
             bh1[i]=0.0;
            }
            /* H2 */
            for(i=1;i<=H;i++)
            {
             deltah2[i]=0.0;
             Outputh2[i]=0.0;
             bh2[i]=0.0;
            }
            /* OUT */
            for(i=1;i<=OUT;i++)
            {
             delta0[i]=0.0;
             Output0[i]=0.0;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            dbo[i]=0.0;
            }

}/* end initial() */

void createb(void)
{
int i;
            /* (H1)bias in hidden layer1 */
            for(i=1;i<=H;i++)
            {
             bh1[i]=(rand()%1000)/1000.00;
             if(rand()%2)
                        bh1[i]=-1.00*bh1[i];
            }
            /* (H2)bias in hidden layer2 */
            for(i=1;i<=H;i++)
            {
             bh2[i]=(rand()%1000)/1000.00;
             if(rand()%2)
                        bh2[i]=-1.00*bh2[i];
            }
            /* (OUT)bias in output layer */
            for(i=1;i<=OUT;i++)
            {
             bo[i]=(rand()%1000)/1000.00;
             if(rand()%2)
                        bo[i]=-1.00*bo[i];
            }
}/* end createb() */
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.ตัวโปรแกรม Tif2b.exe

สำหรับแปลงภาพไฟล์กราฟฟิกจากการแสกนหน้าเอกสารเป็นไฟล์ไบนารีบนเครื่องไมโครคอมพิวเตอร์

```
#include <process.h>
#include <string.h>
#include <stdio.h>
#include <conio.h>
#include <alloc.h>
#include <dos.h>
#include <graphics.h>


/* TIFF tag names */
#define SubfileType          255 /* 0xFF */
#define ImageWidth           256 /* 0x100 */
#define ImageLength          257 /* 0x101 */
#define RowsPerStrip         278 /* 0x116 */
#define StripOffsets         273 /* 0x111 */
#define StripByteCounts      279 /* 0x117 */
#define SamplesPerPixel      277 /* 0x115 */
#define BitsPerSample        258 /* 0x102 */
#define PlanarConfiguration  284 /* 0x11c */
#define Compression          259 /* 0x103 */
#define Group3Options        292 /* 0x124 */
#define Group4Options        293 /* 0x125 */
#define FillOrder            266 /* 0x10a */
#define Thresholding         263 /* 0x107 */
#define CellWidth            264 /* 0x108 */
#define CellLength           265 /* 0x109 */
#define MinSampleValue       280 /* 0x118 */
#define MaxSampleValue       281 /* 0x119 */
#define PhotometricInterp    262 /* 0x106 */
#define GrayResponseUnit     290 /* 0x122 */
#define GrayResponseCurve    291 /* 0x123 */
#define ColorResponseUnit    300 /* 0x12c */
#define ColorResponseCurves  301 /* 0x12d */
#define XResolution          282 /* 0x11a */
#define YResolution          283 /* 0x11b */
#define ResolutionUnit       296 /* 0x128 */
#define Orientation          274 /* 0x112 */
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
#define DocumentName        269 /* 0x10d */
#define PageName            285 /* 0x11d */
#define XPosition           286 /* 0x11e */
#define YPosition           287 /* 0x11f */
#define PageNumber          297 /* 0x129 */
#define ImageDescription    270 /* 0x10e */
#define Make                271 /* 0x10f */
#define Model               272 /* 0x110 */
#define FreeOffsets         288 /* 0x120 */
#define FreeByteCounts      289 /* 0x121 */

/* TIFF sizes */
#define TIFFbyte            1
#define TIFFascii           2
#define TIFFshort           3
#define TIFFlong            4
#define TIFFrational        5

/* TIFF compression type */
#define COMPnone            1
#define COMPhuff            2
#define COMPfax3            3
#define COMPfax4            4
#define COMPwrd1            0x8003
#define COMPmpnt            0x8005

#if DEBUG
   #define tag_count        37

   typedef struct {
         int number;
         char name[20];
         } TAGNAME;

   static TAGNAME tag_names[tag_count] = {
     { 255,"SubfileType"}, { 256,"ImageWidth"},
     { 257,"ImageLength"}, { 278,"RowsPerstrip"},
     { 273,"StripOffsets"}, { 279,"StripByteCounts"},
     { 277,"SamplesPerPixel"}, { 258,"BitsPerSample"},
     { 284,"PlanarConfiguration"}, { 259,"Compression"},
     { 292,"Group3Options"}, { 293,"Group4Options"},
     { 266,"FillOrder"}, { 263,"Threshholding"},
          { 264,"CellWidth"}, { 265,"CellLength"},
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    { 280,"MinSampleValue"}, { 281,"MaxSampleValue"},
    { 262,"PhotometricInterp"}, { 290,"GrayResponseUnit"},
    { 291,"GrayResponseCurve"}, { 300,"ColorResponseUnit"},
    { 301,"ColorResponseCurves"}, { 282,"XResolution"},
    { 283,"YResolution"}, { 296,"ResolutionUnit"},
    { 274,"Orientation"}, { 269,"DocumentName"},
    { 285,"PageName"}, { 286,"Xposition"},
    { 287,"YPosition"}, { 297,"PageNumber"},
    { 270,"ImageDescription"}, { 271,"Make"},
    { 272,"Model"}, { 288,"FreeOffsets"},
    { 289,"FreeByteCounts"},
    };

  char *tag_name(n)
  int n;
  {
    int i;

    for(i=0;i<tag_count;++i)
          {
      if(n==tag_names[i].number)
        return(tag_names[i].name);
    }
    return("BadTag");
  }

#endif

#define  PIXEL2BYTES(n)     ((n+7)/8)
#define ON                  1
#define OFF                 2
char *cFarPtr_(char *p,long l);
unsigned long cFGetLong_( FILE *fp,struct TiffImage *tiffpic );
int cGetTiffFile_( char *fnm,struct TiffImage *tiffpic );
int cUnpackTiffPict_( struct TiffImage *tiffpic,FILE *fp,FILE *dfp);
int cReadTiffLine_(char *p,FILE *fp,struct TiffImage *tiffpic );
int cFillFrame_( struct TiffImage *tiffpic );
long cDecodeTag_(FILE *fp,struct TiffImage *tiffpic );
int cPanTiffPict_( struct Window *w ,int wactv,int pcv_w,int c, int (*func)() );
int cFGetWord_( FILE *fp,struct TiffImage *tiffpic );
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
unsigned long FGetLong_( FILE *fp,struct TiffImage *tiffpic );
int cError_( char *s );

struct TiffImage {
    unsigned int pwidth,pdepth,pwbytes;    /* the trusty Image dimensions */
    unsigned int numberType;          /* number storage format */
    unsigned int TIFFversion;         /* TIFF Version */
    unsigned int TIFFentries;         /* number of IFD entries */
    unsigned int TIFFsubfile;         /* subfile type */
    unsigned int TIFFsamples;         /* number of samples */
    unsigned int TIFFplancfg;         /* planar configuration */
    unsigned int TIFFcompres;         /* compression type */
    unsigned int TIFFphotmet;         /* photometric interpretation */

    unsigned long TIFFoffset;        /* TIFF offset */
    unsigned long TIFFrowstrip;
        unsigned long TIFFstripoff;      /* Where the first one is */
    unsigned long TIFFstripcnt;      /* How many there are */
    unsigned long TIFFbytecnt;
    unsigned long TIFFImageStart;    /* First byte of Image */
    char fnm[20];
    int numtim;
};



long cDecodeTag_(FILE *fp, struct TiffImage *tiffpic )
{
  long length,offset;
  int tag,type;

  tag  = cFGetWord_(fp,tiffpic );
  type = cFGetWord_(fp,tiffpic );

  if(type == TIFFlong)
  {
        length = cFGetLong_(fp,tiffpic );
    offset = cFGetLong_(fp,tiffpic );
  }
  else
  {
    length = (unsigned long) cFGetWord_( fp,tiffpic );
    cFGetWord_(fp,tiffpic );
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    offset = (unsigned long) cFGetWord_(fp,tiffpic );
    cFGetWord_(fp,tiffpic );
  }
  switch(tag) {
    case SubfileType  :
                tiffpic->TIFFsubfile = (unsigned int) offset;
                break;
    case ImageWidth   :
                tiffpic->pwidth = (unsigned int) offset;
                tiffpic->pwbytes = PIXEL2BYTES(tiffpic->pwidth);
                break;
    case ImageLength  :
                tiffpic->pdepth = (unsigned int) offset;
                break;
   case RowsPerStrip :
                if(type==TIFFlong)
                   tiffpic->TIFFrowstrip = offset;
                else
                   tiffpic->TIFFrowstrip = offset & 0xFFFFL;
                break;
   case StripOffsets :
                if(type == TIFFlong)
                {
                   tiffpic->TIFFstripoff = offset;
                   tiffpic->TIFFstripcnt = length;
                }
                else
                {
                   tiffpic->TIFFstripoff = offset & 0xFFFFL;
                   tiffpic->TIFFstripcnt = length & 0xFFFFL;
                }
               break;
   case StripByteCounts :
                if(type == TIFFlong)
                   tiffpic->TIFFbytecnt = offset;
                else
                   tiffpic->TIFFbytecnt = offset & 0xFFFFL;
               break;
  case SamplesPerPixel :
               tiffpic->TIFFsamples = (unsigned int) offset;
               break;
  case BitsPerSample  :
              break;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
case PlanarConfiguration :
            tiffpic->TIFFplancfg = (unsigned int) offset;
            break;
case Compression    :
            tiffpic->TIFFcompres = (unsigned int) offset;
            break;
case Group3Options    :
            break;
case Group4Options    :
            break;
case FillOrder        :
            break;
case Thresholding     :
            break;
case CellWidth        :
            break;
case CellLength       :
            break;
case MinSampleValue   :
            break;
case MaxSampleValue   :
            break;
case PhotometricInterp:
            tiffpic->TIFFphotmet=(unsigned int) offset;
            break;
case GrayResponseUnit :
            break;
case GrayResponseCurve:
            break;
case ColorResponseUnit:
            break;
case ColorResponseCurves:
            break;
case XResolution      :
            break;
case YResolution      :
            break;
case ResolutionUnit   :
            break;
case Orientation      :
            break;
case DocumentName     :
            break;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        case PageName          :
                        break;
        case XPosition         :
                        break;
        case YPosition         :
                        break;
        case PageNumber        :
                        break;
        case ImageDescription  :
                        break;
        case Make              :
                        break;
        case Model             :
                        break;
        case FreeOffsets       :
                        break;
        case FreeByteCounts    :
                        break;
    }
    return(length);
}

int cFGetWord_( FILE *fp, struct TiffImage *tiffpic )
{
    if( tiffpic->numberType == 'II')
        return((fgetc(fp) & 0xff) + ((fgetc(fp) & 0xff) << 8));
    else
        return(((fgetc(fp) & 0xff) << 8) + (fgetc(fp) & 0xff));
}

unsigned long cFGetLong_( FILE *fp,struct TiffImage *tiffpic )
{
    if( tiffpic ->numberType == 'II')
        return((unsigned long) (fgetc(fp) & 0xFF) +
          ((unsigned long) (fgetc(fp) & 0xFF) << 8) +
          ((unsigned long) (fgetc(fp) & 0xFF) << 16) +
                    ((unsigned long) (fgetc(fp) & 0xFF) << 24));
    else
        return(((unsigned long) (fgetc(fp) & 0xFF) << 24) +
          ((unsigned long) (fgetc(fp) & 0xFF) << 16) +
          ((unsigned long) (fgetc(fp) & 0xFF) << 8) +
          (unsigned long) (fgetc(fp) & 0xFF));
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
char *cFarPtr_( char *p,long l )
{
   unsigned int seg,off;
   seg = FP_SEG(p);
   off = FP_OFF(p);
   seg += (off / 16);
   off &= 0x000f;
   off += (unsigned int)(l & 0x000fL);
   seg += (l / 16L);
   p = MK_FP(seg,off);
   return(p);
}

int cError_( char *s )
{
   puts("\n");
   puts(s);
   puts("\n");
   exit(1);
   return;
}

int cReadTiffLine_(char *p,FILE *fp,struct TiffImage *tiffpic) /* read and decode a line into p */
{
   int n=0,c,i;

   memset(p,0,tiffpic->pwbytes);
   if(tiffpic->TIFFcompress == COMPnone)
   {
            n=fread(p,1,tiffpic->pwbytes,fp);
//          if( tiffpic->TIFFphotmet == 1 )
            for(i=0; i< tiffpic->pwbytes;++i)
                     p[i] = ~p[i];
   }
   else
            if(tiffpic->TIFFcompress == COMPnpnt)
            {
                     do {
                                       c=fgetc(fp) &0xFF;
                                       if(c & 0x80)
                                       {
                                                if(c != 0x80)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                                        {
                                          i = ((~c) & 0xFF) + 2;
                                          c = fgetc(fp);
                                          if(tiffpic->TIFFphotmet == 1)
                                                  while(i--) p[n++] = ~c;
                                        }
                                }
                                else
                                {
                                        i=(c & 0xFF) + 1;
                                        if(tiffpic->TIFFphotmet == 1)
                                        while(i--) p[n++] = fgetc(fp);
                                 }
                        } while(n < tiffpic->pwbytes);
                }
                else
                {
                        printf("Undefined compression method %d\n",tiffpic->TIFFcompres);
                        exit(1);
                }
        return(n);
}

WrtTiff2Bin_( FILE *dfp,unsigned char *buff,int pwbytes);
WrtTiff2Bin_( FILE *dfp,unsigned char *buff,int pwbytes)
{
        int i,j;
        int old_x,old_y;
        unsigned char mark;

        old_x = wherex();
        old_y = wherey();

        cprintf( "programe is converting  " );
        for( i = 0 ; i < pwbytes ; i++ )
        {
                for( mark = 0x80,j = 0 ; j < 8 ; j++ )
                {
                        if( (buff[i] & mark) > 0 )
                                fputc('1',dfp);
                        else
                                fputc('0',dfp);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                        mark >>=1;
                    }

            }
//    fputc( 0xd,dfp);
    fputc( 0xa,dfp);
            gotoxy( old_x,old_y);
            cprintf( "programme is converting    ");
            gotoxy( old_x,old_y);
}


int cUnpackTiffPict_( struct TiffImage *tiffpic,FILE *fp,FILE *dfp);
int cUnpackTiffPict_( struct TiffImage *tiffpic,FILE *fp,FILE *dfp)
{
   long l,t;
   int i,n;
   unsigned char *buff;
   n = 0;

   buff =  farcalloc(tiffpic->pwbytes,sizeof(char) );

   /* if there is only one strip, the offset points to it */
   if( tiffpic->TIFFstripcnt == 1L )
   {
            fseek(fp,tiffpic->TIFFImageStart+tiffpic->TIFFstripoff,SEEK_SET);
            for(i=0; i < tiffpic->pdepth ;++i)
            {
                    n+=cReadTiffLine_( (char*)buff,fp,tiffpic );
                    WrtTiff2Bin_( dfp,buff,tiffpic->pwbytes);
            }
   }
   else
   {
   /* if there is more than one strip, the offset points to the offset */
   /* of the first strip - convoluted, you say? */
            for(l=0L ; l< tiffpic->TIFFstripcnt;++l)
            {
                    fseek(fp,tiffpic->TIFFImageStart+tiffpic->TIFFstripoff+(l*sizeof(long)),SEEK_SET);
                    t=cFGetLong_(fp,tiffpic );
                    fseek(fp,t,SEEK_SET);
                    n+=cReadTiffLine_( (char*)buff,fp,tiffpic );
                    WrtTiff2Bin_( dfp,buff,tiffpic->pwbytes);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    }
  }

//   fputc( 0xd,dfp);
   fputc( 0xa,dfp);
   farfree(buff);
   return(n);
}


int ConvTiff2Bin_( char *fnm,char *dfnm,struct TiffImage *tiffpic );
int ConvTiff2Bin_( char *fnm,char *dfnm,struct TiffImage *tiffpic )
{
  FILE *fp;
  FILE *dfp;
  int i;
  /* attempt to open the file */
  if((fp=fopen( fnm,"rb")) == NULL)
     cError_("Error Opening Tiff the file");

  if((dfp=fopen( dfnm,"wt")) == NULL)
     cError_("Error Opening Output the file");

  tiffpic->numtin       =1;

  tiffpic->numberType=0;        /* number storage format */
  tiffpic->TIFFversion=0;       /* TIFF Version */
  tiffpic->TIFFentries=0;       /* number of IFD entries */
  tiffpic->TIFFsubfile=0;       /* subfile type */
  tiffpic->TIFFsamples=0;       /* number of samples */
  tiffpic->TIFFplancfg=0;       /* planar configuration */
  tiffpic->TIFFcompres=1;       /* compression type */
  tiffpic->TIFFphotmet=1;       /* photometric interpretation */

  tiffpic->TIFFoffset=0L;       /* TIFF offset */
  tiffpic->TIFFrowstrip=0L;
  tiffpic->TIFFstripoff=0L;     /* Where the first one is */
  tiffpic->TIFFstripcnt=0L;     /* How many there are */
  tiffpic->TIFFbytecnt=0L;
  tiffpic->TIFFImageStart = 0L;

  /* read in the header */
  tiffpic->numberType= cFGetWord_(fp,tiffpic );
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    if(tiffpic->numberType != 'MM' && tiffpic->numberType != 'II')
      cError_("This isn't a TIFF file");

    /* discard the version. */
    cFGetWord_(fp,tiffpic );

    tiffpic->TIFFoffset=cFGetLong_(fp,tiffpic );

    fseek(fp,tiffpic->TIFFImageStart+tiffpic->TIFFoffset,SEEK_SET);

    tiffpic->TIFFentries=cFGetWord_(fp,tiffpic );
    for(i=0; i< tiffpic->TIFFentries;++i)
     cDecodeTag_(fp,tiffpic);

    /* fool the screen */
    tiffpic->TIFFphotmet ^= 1;

    /* philosophical value judgement */
    if( tiffpic->pwidth == 0 ||
            tiffpic->pdepth == 0 ||
       tiffpic->TIFFstripoff == 0L )
      cError_("This is a bad TIFF file");
    cUnpackTiffPict_( tiffpic ,fp,dfp);
    fclose(fp);
    return;
}
main( int argv,char *argc[] )
{
    struct TiffImage tiffpic;
          printf( "\nPrograme convert TIFF file to binary 0-1 text file V 0.01a,Writen By Thawee\n");
    if( argv == 3 )
    {
       ConvTiff2Bin_( argc[1]+2,argc[2]+2,&tiffpic );
       printf( "\nprograme converte sucess\n");
    }
    else
    {
       printf( "\nSynTax tif2b  /S[scr] /D[dest]"
            "\n  - scr      : image TIFF file  "
                                "\n  - dest     : text 0-1 destination file\n");

    }
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.ตัวโปรแกรม Cutx.c

สำหรับแยกอักขระบนภาพหน้าเอกสารที่เป็นไบนารีไฟล์พร้อมบันทึกตำแหน่ง
ของอักขระที่แยกได้ที่ละตัวบนเครื่องมินิคอมพิวเตอร์

```
#include<stdio.h>
#include<ctype.h>
#include<stdlib.h>
#include<math.h>

#define lineint  3
#define I        1600  /* number node of input layaer*/
#define sizexy   2000  /* array maximum for put extract character into*/
#define cpix     '2'  /* compute pixel*/
#define wpix     '0'
#define bpix     '1'

double  imagex(FILE *fp);
double  imagey(FILE *fp);
void    extract(FILE *fp,double xd,int target,char *argv[]);
void    header(void);
double   linenum;
char     *Line;
int      image[102][102];

FILE     *fp1,*fp2,*fp3,*fp;

void main(int argc,char *argv[30])
{
int      first,i,temp;
int      stop,linetemp,numline;
int      x,y,j,black,real;
char     ch,*line,mark1,mark2,mark3;
FILE     *fp1;
double   width,xd,yd,z;
char     s_tar[3];
         numline=0;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  mark1='1';
  mark2='0';
  mark3='\n';
if(argc!=3)
{
        printf("\nEnter [prog name] + [page.txt(input file)] +");
        printf("\n[Line.txt(output file)]\n");
        exit(1);
}
else/*-start else-*/
{
        /*[Cut space on header of page*/
        if((fp1=fopen(argv[1],"r+b")) == NULL)
        {
                printf("\nError in opening file %s for reading",argv[1]);
                exit(1);
        }
        if((fp2=fopen("temp.txt","w+b")) == NULL)
        {
                printf("\nError in opening file temp.txt for writing");
                exit(1);
        }
        widel=imagex(fp1);/*find page.txt's width*/
        if( (Line=(char *)malloc((widel+3)*sizeof(char))) == NULL)
        {
                printf("\nCan't malloc for char *Line");
                exit(1);
        }
        first=0;
        do
        {
                temp=0;
                fgets(Line,widel+3,fp1);
                for(i=0;i<widel;i++)
                {
                        if(Line[i]=='1')
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                {
                        temp=temp+1;
                }
        }
        if(temp>0)
        {
                first=1;
        }
        if(first==1)
        {
                fputs(Line,fp2);
        }
}while(!feof(fp1));
fclose(fp1);
fclose(fp2);
free(Line);
/*1End cut space on header of page */

/*2Start cut line and indicate line number and set line interval */
linenum=1.0;
if((fp1=fopen("temp.txt","r+b"))==NULL)
{
        printf("\nError in opening file temp.txt");
        exit(1);
}
if((fp2=fopen("temp1.txt","w+b"))==NULL)
{
        printf("\nError in opening file temp1.txt");
        exit(1);
}
if( (Line=(char *)malloc((widel+3)*sizeof(char))) == NULL)
{
        printf("\nCan't malloc for char *Line");
        exit(1);
}
do
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
{
        fgets(Line,widel+3,fp1);
        temp=0;
        for(i=0;i<widel;i++)
        {
                if(Line[i]=='1')
                {
                        temp=temp+1;
                }
        }
        if(temp==0)
        {
                linetemp=linetemp+1;
        }
        else
        {
                linetemp=0;
        }
        if(linetemp>lineint)
        {
                fputs("L",fp2);
                gcvt(linenum,sizeof(double),s_tar);
                fputs(s_tar,fp2);
                fputs("\n",fp2);
                linenum=linenum+1.0;
                stop=1;
                do
                {
                        fgets(Line,widel+3,fp1);
                        temp=0;
                        for(i=0;i<widel;i++)
                        {
                                if(Line[i]=='1')
                                {
                                        temp=temp+1;
                                }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                }
                if(temp>0)
                {
                        stop=0;
                }
        }while((stop==1)&&(!feof(fp1)));
        fputs(Line,fp2);
    }
    else
    {
        fputs(Line,fp2);
    }
}while(!feof(fp1));
fclose(fp1);
fclose(fp2);
system("rm temp.txt");
/*2End */

/*3extract character on temp1.txt and output to line.txt*/
if((fp1=fopen("temp1.txt","r+b"))==NULL)
{
    printf("\nError in opening file temp1.txt");
    exit(1);
}
if((fp2=fopen(argv[2],"a+b"))==NULL)
{
    printf("\nError in opening file");
    exit(1);
}
linenum=1.0;
do
{
    if((fp3=fopen("tmp.txt","w+b"))==NULL)
    {
        printf("\nError in opening file tmp.txt");
        exit(1);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                }
                z=0.0;
                do
                {
                        fgets(Line,widel+3,fp1);
                        if(Line[0]!='L')
                        {
                                z=z+1.0;
                                fputs(Line,fp3);
                        }
                }while((Line[0]!='L')&&(!feof(fp1)));
/*start extract3.c*/
numline=numline+1;
xd=widel*1.0;
yd=z;
printf("\nLine %d has [WIDE = %1.0lf] [DEPTH = %1.0lf]",numline,xd,yd);
j=1;
real=0;
do
{
        black=0;
        fseek(fp3,0L,0);
        do
        {
                ch=getc(fp3);
                if(ch==bpix)
                {
                        black=1;
                        real=1;
                }
        }while(!feof(fp3));
        if(black==1)
        {
                extract(fp3,xd,j,argv);
/*              printf("\n");*/
/*              printf("\nExtract character [%d] finished",j);*/
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/*                          printf("\nFinish put extract character [%d] on file %s",j,argv[2]);*/
                    }
                    j=j+1;
          }while(black==1);
          if(real==0)
          {
/*          printf("\nFile tmp.txt not have black pixel for extract");*/
          }
          else
          {
            printf("\nFinish cutline : %d",numline);
/*          printf("\n\nExtracting in file tmp.txt now complete !");*/
          }
                              /*end extract3.c*/
                              if(Line[0]=='L')
                              {
                                fputs(Line,fp2);
                              }
                              fclose(fp3);
                    }while(!feof(fp1));
                    fclose(fp1);
                    fclose(fp2);
                    free(Line);
                    system("rm temp1.txt");
          /*3End */
          }/*-end else-*/
          system("rm tmp.txt");
}/*end main*/

void extract(FILE *fp,double col,int target,char *argv[])
{
int       i,j,h,true,in;
double    corx[sizexy],cory[sizexy];
double    c,r,maxx,maxy,minx,miny,cenx,ceny;
double    tempx,tempy,statex,statey;
int       pos1,tail,tall;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
int     ir,ic,tempd;
char    *line;
char    ch,mark,T1,T2;
fpos_t  pos;
FILE    *fpnew2;
double  wide,tempw,tempd1,tempd2,k;

  wide=sqrt(I);
  /*initial corx[sizexy] cory[sizexy] to 0*/
  for(i=0;i<sizexy;i++)
  {
        corx[i]=0;
        cory[i]=0;
  }
  for(i=1;i<=100;i++)
  {
        for(j=1;j<=100;j++)
        {
              image[i][j]='0';
        }
  }

  /*first step is finding 1 in image first*/
  r=1.0;
  c=0.0;
  true=0;
  mark='2';
  fseek(fp,0L,0);
  do
  {
        do
        {
              fgetpos(fp,&pos);
              fread(&ch,sizeof(char),1,fp);
              c=c+1.0;
              if(ch==bpix)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                {
                        corx[0]=c;
                        cory[0]=r;
                        fsetpos(fp,&pos);
                        fwrite(&mark,sizeof(char),1,fp);
                        true=1;
                }
        }while( (ch!='\n')&&(true==0) );
        c=0.0;
        r=r+1.0;
  }while( (true==0)&&(!feof(fp)) );/*if true==1 mean already find out
'bpix' in file*/
  /*second step is searching 1 near first*/
        pos1=0;
        tail=1;
        do
        {
         pos1=pos1+1;
         statex=corx[pos1-1];
         statey=cory[pos1-1];
         for(i=1;i<=8;i++)
         {
                tempx=0.0;
                tempy=0.0;
                if( (i==1) )
                {
                        tempx=statex+1;
                        tempy=statey;
                }
                if( (i==2) )
                {
                        tempx=statex+1;
                        tempy=statey-1;
                }
                if( (i==3) )
                {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            tempx=statex;
            tempy=statey-1;
}
if( (i==4) )
{
            tempx=statex-1;
            tempy=statey-1;
}
if( (i==5) )
{
            tempx=statex-1;
            tempy=statey;
}
if( (i==6) )
{
            tempx=statex-1;
            tempy=statey+1;
}
if( (i==7) )
{
            tempx=statex;
            tempy=statey+1;
}
if( (i==8) )
{
            tempx=statex+1;
            tempy=statey+1;
}
k=((tempy-1)*(col+1))+(tempx-1);
if(k>(col+1))
{
  fseek(fp,0L,0);
  fseek(fp,k,0);
  fgetpos(fp,&pos);
  ch=getc(fp);
  if(ch==bpix)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น ทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                {
                        fsetpos(fp,&pos);
                        fwrite(&mark,sizeof(char),1,fp);
                        cory[tail]=tempy;
                        corx[tail]=tempx;
                        tail=tail+1;
                }
              }
        }/*end for (i=1) to (i=8)*/
      }while(pos1<sizexy);
      maxx=0;
      maxy=0;
      minx=corx[0];/* this minx is not real minx but is the first
                      black minx */
      miny=cory[0];
      i=0;
      for(i=0;i<sizexy;i++)
      {
        if(corx[i]>maxx)
                maxx=corx[i];
        if(cory[i]>maxy)
                maxy=cory[i];
        if( (corx[i]<minx)&&(corx[i]!=0) )
                minx=corx[i];
        if( (cory[i]<miny)&&(cory[i]!=0) )
                miny=cory[i];
      }
/*    printf("\nMax X=%1.0lf",maxx);*/
/*    printf("\nMin X=%1.0lf",minx);*/
/*    printf("\nMax Y=%1.0lf",maxy);*/
/*    printf("\nMin Y=%1.0lf",miny);*/
/*    getchar();*/
/*    printf("\nExtract finish !");*/
/*    width=maxx-minx+1;*/
/*    depth=maxy-miny+1;*/
      tempw=minx+wide;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        tempd1=miny+wide;
/*after this point will extract character*/
        fseek(fp,0L,0);
        if(maxy>=tempd1)
        {
            tempd2=tempd1-1;
            tall=0;
        }
        else/*if(maxy<tempd1)*/
        {
            tempd2=maxy;
            tall=1;
        }
        for(ir=0,ic=0,i=1;i<=tempd2;i++)/*each row*/
        {
                if((line=malloc((long)(col+3)*sizeof(char)))==NULL)
                {
                        printf("Out of memory for line");
                        exit(1);
                }
                fread(line,sizeof(char),(col+1),fp);

                if(i>=miny)
                {
                        ir=ir+1;
                        if(maxx<tempw)
                        {
                                ic=0;
                                for(j=(minx-1);j<maxx;j++)
                                {
                                        ic=ic+1;
                                        if(line[j]==cpix)
                                        {
                                                putc(bpix,fp2);
                                                image[ir][ic]='1';
                                        }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                                        if(line[j]==wpix)
                                        {
                                                putc(wpix,fp2);
                                        }
                                }
/*              while(j<(tempw-1))
                {
                        putc(wpix,fp2);
                        j=j+1;
                }*/
                putc('\n',fp2);
               }/*end if(maxx<tempw)*/
               else/*if(maxx>=tempw)*/
               {
                ic=0;
                for(in=(minx-1);in<(tempw-1);in++)
                {
                        ic=ic+1;
                        if(line[in]==cpix)
                        {
                                putc(bpix,fp2);
                                image[ir][ic]='1';
                        }
                        if(line[in]==wpix)
                        {
                                putc(wpix,fp2);
                        }
                }
                putc('\n',fp2);
               }
                }/*end if(i>=miny)*/
                free(line);
        }/*end for i=1 to i<=tempd2*/
        if(tall==1)
        {
/*      while(i<tempd1)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  {
        for(j=(minx-1);j<(tempw-1);j++)
        {
          putc(wpix,fp2);
        }
        putc('\n',fp2);
        i++;
  }*/
}
putc('T',fp2);
if(target<10)
{
   T1=48+target;
   T2=48;
}
else
{
   tempd=(target/10)*10;
   T1=(target-tempd)+48;
   tempd=tempd/10;
   T2=tempd+48;
}
putc(T2,fp2);
putc(T1,fp2);
putc('\n',fp2);
cenx=(maxx+minx)/2;/*x coordinate of character for rearange word */
ceny=(maxy+miny)/2;/*y coordinate of character for rearange word */
/* To rearange word we will use ceny to compare the other maxy,miny
   for finding row of each character and use cenx to compare maxx,maxy
   for finding col of each character */
fwrite(&cenx,sizeof(double),1,fp2);
fwrite(&ceny,sizeof(double),1,fp2);
fwrite(&maxx,sizeof(double),1,fp2);
fwrite(&maxy,sizeof(double),1,fp2);
fwrite(&minx,sizeof(double),1,fp2);
fwrite(&miny,sizeof(double),1,fp2);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            putc('\n',fp2);
/*          printf("\nCenx = %lf",cenx);
            printf("\nCeny = %lf",ceny);
            printf("\nMaxx = %lf",maxx);
            printf("\nMaxy = %lf",maxy);
            printf("\nMinx = %lf",minx);
            printf("\nMiny = %lf",miny);
            getchar();*/
/*end extract*/
}

double imagex(FILE *fp)
{
char    ch;
double  colx;

  colx=0;
  do
  {
   ch=getc(fp);
   colx=colx+1;
  }while(ch!='\n');
  return(colx-1);
}

double imagey(FILE *fp)
{
char    ch;
int     temp;
double  rowy;

  rowy=1;
  temp=0;
  ch=getc(fp);
  do
  {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        ch=getc(fp);
        if( ((ch==wpix)||(ch==bpix))&&(temp==0) )
        {
       temp=1;
       rowy=rowy+1;
        }
        if(ch=='\n')
        {
       temp=0;
        }
}while(!feof(fp));
return(rowy);
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.ตัวโปรแกรม Recog.c

สำหรับจดจำอักขระของภาพหน้าเอกสารซึ่งทำการแยกอักขระจากโปรแกรม Cutx.c

พร้อมจัดลำดับอักขระกลับสู่ตำแหน่งเดิมของประโยคข้อความซึ่งทำงานบนเครื่องมินิคอมพิวเตอร์

```
#include<stdio.h>
#include<ctype.h>
#include<stdlib.h>
#include<math.h>
#include<time.h>
#include<malloc.h>
#include<string.h>
#include<float.h>
#include<signal.h>

#define I    1600
#define H    230
#define OUT   87
#define LOOP  100 /*LOOP is integer for saving weight,bias,delta */

void saveb(char *argv[]);
void test(char *argv[]);
void freew(void);
void savew(char *argv[]);
void createb(void);
void initdel(void);
void sort(char *argv[],int ln);
void loaddel(char *argv[]);
void savedel(char *argv[]);
void loadw(char *argv[]);
void loadb(char *argv[]);

float  n,alpha;
double errorb;
double bh1[(H+1)],bh2[(H+1)],bo[(OUT+1)];
double deltah1[(H+1)],deltah2[(H+1)],deltao[(OUT+1)];
double dbh1[(H+1)],dbh2[(H+1)],dbo[(OUT+1)];
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
double Outputh1[(H+1)],Outputh2[(H+1)],Outputo[(OUT+1)];
double Inputh1[(H+1)],Inputh2[(H+1)],Inputo[(OUT+1)];
double *wh1[I+1],*dwh1[I+1],*wh2[H+1],*dwh2[H+1],*wo[H+1],*dwo[H+1];

int    count,times,xpos,ypos,widel[100],wideA;
int    ind,oxpos[561],oypos[561];
double Input[(I+1)];
char   str[51];/*str size must wider than the width of the widest
character */

main(int argc,char *argv[20])
{
int    load,win,fault,ln;
int    i,j,all,tempn1,tempn2;
int    col,row,Tar,T1,T2,over;
char   f[3];
char   ch;
double TAR[(OUT+1)],ms,diff,errorp,del,deltmp,twin;
double temph1,temph2,tempo,el;
double dtemph1,dtemph2,dtempo;
double cenx,ceny,maxx,maxy,minx,miny;
double numchar;
FILE   *fp,*fp13;

          if(argc!=14)
          {
            printf("\nYou enter wrong !");
            printf("\nEnter [Program name] + [file input.txt]+");
            printf("\n      [file wh1.txt] + [file  dwh1.txt]+");
            printf("\n      [file wh2.txt] + [file  dwh2.txt]+");
            printf("\n      [file wo.txt ] + [file   dwo.txt]+");
            printf("\n      [file bh1.txt] + [file   bh2.txt]+");
            printf("\n      [file bo.txt ] + [file delta.txt]");
            printf("\n           [0 or 1] /* 0 mean intial all, 1 mean load old*/");
            printf("\n           [file output of recognition]");
            printf("\n");
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    exit(1);
  }
  el=999.0;/*end of line of file argv[13] has symbol is 999*/
  n=0.1;
  alpha=0.5;
  load=atoi(argv[12]);
  for(i=0;i<=99;i++)
  {
        widel[i]=0;
  }

/* Created weight   */

  for(i=1;i<=I;i++)
  {
    if(( wh1[i]=(double * )malloc((H+1)*sizeof(double))) == NULL)
    {
      printf("\nOut of memory for allocating wh1[I][H]");
      exit(1);
    }
    for(j=1;j<=H;j++)
    {
      wh1[i][j]=(rand() % 1000)*0.0001;
      if(rand() % 2)
       wh1[i][j]=-1.0*wh1[i][j];
    }
  }
  for(i=1;i<=I;i++)
  {
    if(( dwh1[i]=(double * )malloc((H+1)*sizeof(double))) == NULL)
    {
      printf("\nOut of memory for allocating dwh1[I][H]");
      exit(1);
    }
    for(j=1;j<=H;j++)
    {
```

เอกสารนี้สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
      dwh1[i][j]=0.0;
    }
  }

  /* THIS  add */
  for(i=1;i<=H;i++)
  {
   if(( wh2[i]=(double * )malloc((H+1)*sizeof(double))) == NULL)
   {
    printf("\nOut of memory for allocating wh2[H][H]");
    exit(1);
   }
   for(j=1;j<=H;j++)
   {
    wh2[i][j]=(rand() % 1000)*0.0001;
    if(rand() % 2)
     wh2[i][j]=-1.0*wh2[i][j];
   }
  }
  for(i=1;i<=H;i++)
  {
   if(( dwh2[i]=(double * )malloc((H+1)*sizeof(double))) == NULL)
   {
    printf("\nOut of memory for allocating dwh2[I][H]");
    exit(1);
   }
   for(j=1;j<=H;j++)
   {
    dwh2[i][j]=0.0;
   }
  }
  /* THIS */

  for(i=1;i<=H;i++)
  {
   if(( wo[i]=(double * )malloc((OUT+1)*sizeof(double))) == NULL)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    {
      printf("\nOut of memory for allocating wo[H][OUT]");
      exit(1);
    }
    for(j=1;j<=OUT;j++)
    {
      wo[i][j]=(rand() % 1000)*0.0001;
      if(rand() % 2)
       wo[i][j]=-1.0*wo[i][j];
    }
  }
  for(i=1;i<=H;i++)
  {
    if(( dwo[i]=(double * )malloc((OUT+1)*sizeof(double)) == NULL)
    {
      printf("\nOut of memory for allocating dwo[H][OUT]");
      exit(1);
    }
    for(j=1;j<=OUT;j++)
    {
      dwo[i][j]=0.0;
    }
  }
if(load==0)
{
  createb();
  initdel();
  errorb=0.0;
  times=0;
  deltmp=0.0;
}
else
{
  loadb(argv);
  loaddel(argv);
  loadw(argv);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            deltmp=errorb;
          }
            errorp=0.0;
            j=0;
            tempn1=I+4+sqrt(I);
            tempn2=I+sqrt(I);
all=0;
fault=0;
if( (fp=fopen(argv[1],"rb")) == NULL)
{
        printf("\nError in open file %s",argv[1]);
        exit(1);
}
if( (fp13=fopen("temp.asc","wb")) == NULL)
{
        printf("\nError in creating file %s",argv[13]);
        exit(1);
}
j=1;
numchar=0;
ln=0;
wideA=0;
/* do
{
  fgets(str,sizeof(str),fp);
  if(str[0]=='T')
  {
                printf("\n%s",str);
                fread(&cenx,sizeof(double),1,fp);
                fread(&ceny,sizeof(double),1,fp);
                fread(&maxx,sizeof(double),1,fp);
                fread(&maxy,sizeof(double),1,fp);
                fread(&minx,sizeof(double),1,fp);
                fread(&miny,sizeof(double),1,fp);
                printf("\nCenx = %lf",cenx);
                printf("\nCeny = %lf",ceny);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                printf("\nMaxx = %lf",maxx);
                printf("\nMaxy = %lf",maxy);
                printf("\nMinx = %lf",minx);
                printf("\nMiny = %lf",miny);
                getchar();
    }
    else
    {
        if(str[0]=='L')
        {
                printf("\nWe meet L");
                printf("\n%s",str);
                getchar();
        }
        else
        {
                printf("\n%s",str);
        }
    }
 }while(!feof(fp));*/
do
{
        for(i=1;i<=I;i++)
        {
                Input[i]=0.0;

      }
        j=1;
        do
        {
                if(j==1)
                  {
                        fgets(str,sizeof(str),fp);
                }
                  fgets(str,sizeof(str),fp);
                  if(str[0]=='T')
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                {
/*                      printf("\n%s",str);*/
                        fread(&cenx,sizeof(double),1,fp);
                        fread(&ceny,sizeof(double),1,fp);
                        fread(&maxx,sizeof(double),1,fp);
                        fread(&maxy,sizeof(double),1,fp);
                        fread(&minx,sizeof(double),1,fp);
                        fread(&miny,sizeof(double),1,fp);
/*                      printf("\nCenx = %lf",cenx);
                        printf("\nCeny = %lf",ceny);
                        printf("\nMaxx = %lf",maxx);
                        printf("\nMaxy = %lf",maxy);
                        printf("\nMinx = %lf",minx);
                        printf("\nMiny = %lf",miny);
                        getchar();*/
                        break;
                }
                else
                {
                    if(str[0]=='L')
                    {
                        if(wideA!=0)
                        {
                            /*write symbol end of line*/
                            ln=ln+1;
                            widel[ln]=wideA;
                            wideA=0;
                        }
                    }
                    else
                    {
/*                      printf("\n%s",str);*/
                        for(i=0;i<sqrt(I);i++)
                        {
                            if((str[i]=='1')||(str[i]=='0'))
                            {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                                        Input[((40*j)+i)-39]=(str[i]-48)*1.0;
                                    }
                                str[i]='0';
                                }
                            }
                        }
                j=j+1;
            }while((!feof(fp))||(j<=sqrt(I)));
/*          printf("j=%d",j);*/

            /*show Input[1-->I]*/
/*          j=1;*/
/*          for(i=1;i<=sqrt(I);i++)*/
/*          {*/
/*                  printf("\n");*/
/*                  for(j=1;j<=sqrt(I);j++)*/
/*                  {*/
/*                          printf("%1.0lf",Input[(40*(i-1))+j]);*/
/*                  }*/
/*          }*/
/*          printf("\n");*/
/*          printf("\ncenx = %lf",cenx);*/
/*          printf("\nceny = %lf",ceny);*/
/*          printf("\nmaxx = %lf",maxx);*/
/*          printf("\nmaxy = %lf",maxy);*/
/*          printf("\nminx = %lf",minx);*/
/*          printf("\nminy = %lf",miny);*/
/*          getchar();*/
            /*end show*/
if(((maxy-miny)>2.0)&&((maxx-minx)>2.0)/*&&(Input[0]!=' ')*/)
{
/*          printf("\nHey we go into Forward Phase");
            getchar(); */
  /* This point can use Input[1-->I] */
            /* Forward For Recognition*/
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/* Forward phase */
 /*1Compute input for every node in hidden layer 1 */
        for(i=1;i<=H;i++) /* hidden layer 1 to H = col 1 to H hidden weight */
        {
         temph1=0.0;
         for(j=1;j<=I;j++)/* input value 1 to I of hidden layer */
         {
          temph1=temph1+(Input[j]*(wh1[j][i]));
         }
         Inputh1[i]=temph1+bh1[i];
        }
/* end 1 */
/*2Compute output for every node in hidden layer */
        for(i=1;i<=H;i++)
        {
                if( (Inputh1[i]>-88.00)&&(Inputh1[i]<88.00) )
                {
                 Outputh1[i]=1.0/(1.0+exp(-Inputh1[i]));
                }
                else
                {
                 if(Inputh1[i]>=88.00)
                    Outputh1[i]=1.000000;
                 else
                    if(Inputh1[i]<=-88.00)
                       Outputh1[i]=0.000000;
                }
        }
/* end 2 */
/* THIS ADD */
/* *Compute input for every node in hidden layer 2 */
        for(i=1;i<=H;i++) /* hidden layer 2 to H = col 1 to H hidden weight */
        {
         temph2=0.0;
         for(j=1;j<=H;j++)/* input value 1 to H of hidden layer */
         {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                temph2=temph2+(Outputh1[j]*(wh2[j][i]));
                }
                Inputh2[i]=temph2+bh2[i];
                }
        /* *Compute output for every node in hidden layer 2 */
                for(i=1;i<=H;i++)
                {
                        if( (Inputh2[i]>-88.00)&&(Inputh2[i]<88.00) )
                        {
                         Outputh2[i]=1.0/(1.0+exp(-Inputh2[i]));
                        }
                        else
                        {
                         if(Inputh2[i]>=88.00)
                            Outputh2[i]=1.000000;
                         else
                            if(Inputh2[i]<=-88.00)
                               Outputh2[i]=0.000000;
                        }
                } .
        /* end 2 */
        /* END THIS ADD */
        /*3Compute input for every node in output layer */
                for(i=1:i<=OUT:i++) /* node 1 to OUT in output layer */
                {
                 tempo=0.0;
                 for(j=1:j<=H;j++) /* input value of node i in output layer from hidden layer (node j=1
to j=H) */
                 {
                  tempo=tempo+(Outputh2[j]*(wo[j][i]));
                 }
                 Inputo[i]=tempo+bo[i];
                }
        /* end 3 */
        /*4Compute Output for every node in output layer */
                for(i=1:i<=OUT;i++)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        {
         if( (Inputo[i]>-88.00)&&(Inputo[i]<88.00) )
         {
                Outputo[i]=1.0/(1.0+exp(-Inputo[i]));
         }
         else
         {
                if(Inputo[i]>=88.00)
                   Outputo[i]=1.000000;
                else
                 if(Inputo[i]<=-88.00)
                   Outputo[i]=0.000000;
         }
        }
/* end 4 */
/*5Compute different between target and output value */
        twin=Outputo[1];
        win=1;
         for(i=1;i<=OUT;i++)
         {
                if(twin<Outputo[i])
                {
                        win=i;
                        twin=Outputo[i];
                }
         }
        printf("win number = %d",win);*/
        if((win>=1)&&(win<=10))
        {
                win=win+47;
        }
        else
        {
                if((win>=11)&&(win<=57))
                {
                        win=win+150;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                }
                else
                {
                        if(win==58)
                                win=209;
                        if(win==59)
                                win=210;
                        if(win==60)
                                win=224;
                        if((win>=61)&&(win<=65))
                        {
                                win=win+165;
                        }
                        else
                        {
                                if((win>=66)&&(win<=75))
                                {
                                        win=win+174;
                                }
                                else
                                {
                                        if((win>=76)&&(win<=81))
                                        {
                                                win=win+136;
                                        }
                                        else
                                        {
                                                if((win>=82)&&(win<=87))
                                                        win=win+149;
                                        }
                                }
                        }
                }
        }
/*      printf("\nwin = %d",win);
        printf("\nwin = %c",win);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                printf("\ncenx=%lf",cenx);
                printf("\nceny=%lf",ceny);
                printf("\nmaxx=%lf",maxx);
                printf("\nmaxy=%lf",maxy);
                printf("\nminx=%lf",minx);
                printf("\nminy=%lf",miny);
                getchar();*/
/*              fprintf(fp13,"%c",win);*/
/*              fprintf(fp13,"%lf",cenx);*/
/*              fprintf(fp13,"%lf",ceny);*/
/*              fprintf(fp13,"%lf",maxx);*/
/*              fprintf(fp13,"%lf",maxy);*/
/*              fprintf(fp13,"%lf",minx);*/
/*              fprintf(fp13,"%lf",miny);*/
/*              printf("\n%c",win);*/
/*              printf("\n%lf",cenx);*/
/*              printf("\n%lf",ceny);*/
/*              printf("\n%lf",maxx);*/
/*              printf("\n%lf",maxy);*/
/*              printf("\n%lf",minx);*/
/*              printf("\n%lf",miny);*/
                fwrite(&win,sizeof(int),1,fp13);
                fwrite(&cenx,sizeof(double),1,fp13);
                fwrite(&ceny,sizeof(double),1,fp13);
                fwrite(&maxx,sizeof(double),1,fp13);
                fwrite(&maxy,sizeof(double),1,fp13);
                fwrite(&minx,sizeof(double),1,fp13);
                fwrite(&miny,sizeof(double),1,fp13);
                wideA=wideA+1;
/*              printf("\nHey This wideA=%d",wideA);
                getchar();*/
            /* end 5 */

        /* End Forward */
    }/*end if((maxy-miny)>2.0)&&((maxx-minx)>2.0)*/
}while(!feof(fp));
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    fclose(fp);
    fclose(fp13);
    freew();
/*  printf("\nln = %d",ln);*/
    sort(argv,ln);
    return(0);
}/* end main */
void sort(char *argv[],int ln)
{
int       *ascii,*record,*recasc,real,i,j,tempin,rtail1;
int       y,k,tail1,tail2,tail3,f,incre,highest,kk;
double    *xcen,*ycen,*xmax,*ymax,*xmin,*ymin,temp;
double  *recxcen,*recycen,*recxmax,*recymax,*recxmin,*recymin;
FILE      *fp,*fp1,*fp2;

          if( (fp=fopen("temp.asc","rb")) == NULL)
          {
                    printf("\nError in opening file %s",argv[13]);
                    exit(1);
          }
          if( (fp1=fopen(argv[13],"wb")) == NULL)
          {
                    printf("\nError in opening file temp.out");
                    exit(1);
          }
          if( (fp2=fopen("temp.aft","wb")) == NULL)
          {
                    printf("\nError in opening file temp.aft");
                    exit(1);
          }
          j=1;
          do
          {
/*                  printf("\nWe go into sort() function");*/
/*                  printf("\nWidel[%d] = %d",j,widel[j]);*/
                    if( (ascii=(int *)malloc((widel[j]+1)*sizeof(int))) == NULL)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
{
        printf("\nCan't malloc ascii[%d]",widel[j]);
        exit(1);
}
if( (record=(int *)malloc((widel[j]+1)*sizeof(int))) == NULL)
{
        printf("\nCan't malloc record[%d]",widel[j]);
        exit(1);
}
if( (xcen=(double *)malloc((widel[j]+1)*sizeof(double))) == NULL)
{
        printf("\nCan't malloc xcen[%d]",widel[j]);
        exit(1);
}
if( (ycen=(double *)malloc((widel[j]+1)*sizeof(double))) == NULL)
{
        printf("\nCan't malloc ycen[%d]",widel[j]);
        exit(1);
}
if( (xmax=(double *)malloc((widel[j]+1)*sizeof(double))) == NULL)
{
        printf("\nCan't malloc xmax[%d]",widel[j]);
        exit(1);
}
if( (ymax=(double *)malloc((widel[j]+1)*sizeof(double))) == NULL)
{
        printf("\nCan't malloc ymax[%d]",widel[j]);
        exit(1);
}
if( (xmin=(double *)malloc((widel[j]+1)*sizeof(double))) == NULL)
{
        printf("\nCan't malloc xmin[%d]",widel[j]);
        exit(1);
}
if( (ymin=(double *)malloc((widel[j]+1)*sizeof(double))) == NULL)
{
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            printf("\nCan't malloc ymin[%d]",widel[j]);
            exit(1);
    }
    if( (recasc=(int *)malloc((widel[j]+1)*sizeof(int))) == NULL)
    {
            printf("\nCan't malloc recasc[%d]",widel[j]);
            exit(1);
    }
    if( (recxcen=(double *)malloc((widel[j]+1)*sizeof(double))) == NULL)
    {
            printf("\nCan't malloc recxcen[%d]",widel[j]);
            exit(1);
    }
    if( (recycen=(double *)malloc((widel[j]+1)*sizeof(double))) == NULL)
    {
            printf("\nCan't malloc recycen[%d]",widel[j]);
            exit(1);
    }
    if( (recxmax=(double *)malloc((widel[j]+1)*sizeof(double))) == NULL)
    {
            printf("\nCan't malloc recxmax[%d]",widel[j]);
            exit(1);
    }
    if( (recymax=(double *)malloc((widel[j]+1)*sizeof(double))) == NULL)
    {
            printf("\nCan't malloc recymax[%d]",widel[j]);
            exit(1);
    }
    if( (recxmin=(double *)malloc((widel[j]+1)*sizeof(double))) == NULL)
    {
            printf("\nCan't malloc recxmin[%d]",widel[j]);
            exit(1);
    }
    if( (recymin=(double *)malloc((widel[j]+1)*sizeof(double))) == NULL)
    {
            printf("\nCan't malloc recymin[%d]",widel[j]);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            exit(1);
    }
    for(i=0;i<widel[j];i++)
    {
            fread(&ascii[i],sizeof(int),1,fp);
            fread(&xcen[i],sizeof(double),1,fp);
            fread(&ycen[i],sizeof(double),1,fp);
            fread(&xmax[i],sizeof(double),1,fp);
            fread(&ymax[i],sizeof(double),1,fp);
            fread(&xmin[i],sizeof(double),1,fp);
            fread(&ymin[i],sizeof(double),1,fp);
            printf("\nRead from file temp.asc");
            printf("%c",ascii[i]);
            printf("\nxcen[%d]  = %lf",i,xcen[i]);
            printf("\nycen[%d]  = %lf",i,ycen[i]);
            printf("\nxmax[%d]  = %lf",i,xmax[i]);
            printf("\nymax[%d]  = %lf",i,ymax[i]);
            printf("\nxmin[%d]  = %lf",i,xmin[i]);
            printf("\nymin[%d]  = %lf",i,ymin[i]);
            getchar();*/
    }
    printf("\nEnb line ");*/
    /*start sort character*/
    /*sort xcen[1]=min -->xcen[widel[j]]=max*/
    do
    {
            real=0;
            for(i=0;i<widel[j];i++)
            {
                    if(xcen[i]>xcen[i+1])
                    {
                            tempin=ascii[i+1];
                            ascii[i+1]=ascii[i];
                            ascii[i]=tempin;

                            temp=xcen[i+1];
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    xcen[i+1]=xcen[i];
                    xcen[i]=temp;

                    temp=ycen[i+1];
                    ycen[i+1]=ycen[i];
                    ycen[i]=temp;

                    temp=xmax[i+1];
                    xmax[i+1]=xmax[i];
                    xmax[i]=temp;

                    temp=ymax[i+1];
                    ymax[i+1]=ymax[i];
                    ymax[i]=temp;

                    temp=xmin[i+1];
                    xmin[i+1]=xmin[i];
                    xmin[i]=temp;

                    temp=ymin[i+1];
                    ymin[i+1]=ymin[i];
                    ymin[i]=temp;
                    real=1;
                }
        }
}while(real==1);
/*end sort xcen*/
/*eliminate char209&char209 to char208*/
rtail1=0;
for(i=0;i<widel[j];i++)
{
        if((ascii[i]==209)&&(ascii[i+1]==209))
        {
                ascii[i]=208;
                for(kk=i+1;kk<widel[j];kk++)
                {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    ascii[kk]=ascii[kk+1];
                    xcen[kk]=xcen[kk+1];
                    ycen[kk]=ycen[kk+1];
                    xmax[kk]=xmax[kk+1];
                    ymax[kk]=ymax[kk+1];
                    xmin[kk]=xmin[kk+1];
                    ymin[kk]=ymin[kk+1];
                    if(kk==(widel[j]-1))
                    {
                            rtail1=rtail1+1;
                            ascii[kk+1]=0;
                    }
                }
            }
        }
        widel[j]=widel[j]-rtail1+1;
        printf("\n");*/
        /*end eliminate char209&char209 to char208*/
        /*initial variable record[0 -> widel[j]] to 0*/
        for(i=0;i<widel[j];i++)
        {
                record[i]=0;
        }
        /*end initial variable record*/
        for(i=0;i<widel[j];i++)
        {
         y=0;
         for(k=0;k<widel[j];k++)
         {
          if((xcen[i]>=xmin[k])&&(xcen[i]<=xmax[k]))
          {
           if(record[k]!=1)
           {
                y=y+1;
                record[k]=1;
                recasc[y]=ascii[k];
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                recxcen[y]=xcen[k];
                recycen[y]=ycen[k];
                recxmax[y]=xmax[k];
                recymax[y]=ymax[k];
                recxmin[y]=xmin[k];
                recymin[y]=ymin[k];
            }
          }
        }
tail1=y;
    if(y>1)
    {
      do
      {
        tail3=tail1;
        for(k=0;k<tail3;k++)
        {
          for(f=0;f<widel[j];f++)
          {
            if(recxcen[k+1]>=xmin[f])
            {
              if(recxcen[k+1]<=xmax[f])
              {
                    if(record[f]!=1)
                    {
                          tail1=tail1+1;
                          record[f]=1;
                          recasc[tail1]=ascii[f];
                          recxcen[tail1]=xcen[f];
                          recycen[tail1]=ycen[f];
                          recxmax[tail1]=xmax[f];
                          recymax[tail1]=ymax[f];
                          recxmin[tail1]=xmin[f];
                          recymin[tail1]=ymin[f];
                    }/*if(record[f]!=1)*/
              }/*if(recxcen[k+1]<=xmax[f])*/
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            }/*if(recxcen[k+1]>=xmin[f])*/
        }/*for(f=0;f<wide[j];f++)*/
    }/*for(k=0;k<tail3;k++)*/
    tail2=tail1;
    incre=0;
    if(tail2!=tail3)
    {
        incre=1;
    }
}while(incre==1);

/*3sort char in record[1->tail1] by ycen[1]=min->*/
/*ycen[tail1]=max*/
do
{
    real=0;
    for(k=1;k<tail1;k++)
    {
        if(recycen[k]>recycen[k+1])
        {
            tempin=recasc[k+1];
            recasc[k+1]=recasc[k];
            recasc[k]=tempin;

            temp=recxcen[k+1];
            recxcen[k+1]=recxcen[k];
            recxcen[k]=temp;

            temp=recycen[k+1];
            recycen[k+1]=recycen[k];
            recycen[k]=temp;

            temp=recxmax[k+1];
            recxmax[k+1]=recxmax[k];
            recxmax[k]=temp;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                temp=recymax[k+1];
                recymax[k+1]=recymax[k];
                recymax[k]=temp;

                temp=recxmin[k+1];
                recxmin[k+1]=recxmin[k];
                recxmin[k]=temp;

                temp=recymin[k+1];
                recymin[k+1]=recymin[k];
                recymin[k]=temp;
                real=1;
            }/*if(recycen[k]>recycen[k+1])*/
        }/*for(k=1;k<tail1;k++)*/
}while(real==1);

/*4 find highest char in record[1->tail1]*/
for(k=1;k<tail1;k++)
{
        if((recymax[k]-recymin[k])<(recymax[k+1]-recymin[k+1]))
        {
                highest=k+1;
        }
        else
        {
                highest=k;
        }
}
/*5 print output of record[1->tail1] to file*/

/*printf output to file fp1*/
fprintf(fp1,"%c",recasc[highest]);
for(k=(highest+1);k<=tail1;k++)
{
        fprintf(fp1,"%c",recasc[k]);
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
for(k=1;k<highest;k++)
{
        fprintf(fp1,"%c",recasc[highest-k]);
}
/*end printf output to file fp1*/

/*printf output to screen
printf("%c",recasc[highest]);
for(k=(highest+1);k<=tail1;k++)
{
        printf("%c",recasc[k]);
}
for(k=1;k<highest;k++)
{
        printf("%c",recasc[highest-k]);
}
end printf output to screen*/

for(k=1;k<=tail1;k++)
{
        recasc[k]=0;
        recxcen[k]=0.0;
        recycen[k]=0.0;
        recxmax[k]=0.0;
        recymax[k]=0.0;
        recxmin[k]=0.0;
        recymin[k]=0.0;
}

}/*end if(y>1)*/
else
{
 if(y==1)
 {
        fprintf(fp1,"%c",ascii[i]);
        record[i]=1;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    }
                }/*end else if(y>1)*/
                    }/*for(i=0;i<widel[j];i++)*/
                    /*end sort character*/
                    j=j+1;
                    fprintf(fp1,"%s","\n");
                    /*printf end line to screen
                    printf("\n");
                    end printf end line to screen*/
                    fprintf(fp2,"%s","\n");
            }while(j<=ln);
/*          printf("\nln=%d",ln);*/
/*          getchar();*/
            fclose(fp);
            fclose(fp1);
            free(record);
            free(recasc);
            free(recxcen);
            free(recycen);
            free(recxmax);
            free(recymax);
            free(recxmin);
            free(recymin);
            system("rm temp.asc");
            system("rm temp.aft");
}

void freew(void)
{
int i;
 /* start  free memory of weight */
 for(i=0;i<=H;i++)
 {
  free(wh1[i]);
  free(dwh1[i]);
 }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
 for(i=0;i<=H;i++)
 {
  free(wh2[i]);
  free(dwh2[i]);
 }
 for(i=0;i<=OUT;i++)
 {
  free(wo[i]);
  free(dwo[i]);
 }
 /* end free allocating memory for weight && delta weight */
}

void savedel(char *argv[])
{
int     i;
double  del;
FILE    *fp;
/* start save delta to file */
        if((fp=fopen(argv[11],"wb"))==NULL)
        {
                printf("\nError in creating file %s",argv[11]);
                exit(1);
        }
        /* H1 */
        for(i=1;i<=H;i++)
        {
                del=deltah1[i];
                fwrite(&del.sizeof(double),1,fp);
        }
        for(i=1;i<=H;i++)
        {
                del=Outputh1[i];
                fwrite(&del.sizeof(double),1,fp);
        }
        for(i=1;i<=H;i++)
```

เอกสารนี้สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
{
        del=dbh1[i];
        fwrite(&del,sizeof(double),1,fp);
}
/*end H1*/

/* H2 */
for(i=1;i<=H;i++)
{
        del=deltah2[i];
        fwrite(&del,sizeof(double),1,fp);
}
for(i=1;i<=H;i++)
{
        del=Outputh2[i];
        fwrite(&del,sizeof(double),1,fp);
}
for(i=1;i<=H;i++)
{
        del=dbh2[i];
        fwrite(&del,sizeof(double),1,fp);
}
/*end H2*/

/*Out */
for(i=1;i<=OUT;i++)
{
        del=deltao[i];
        fwrite(&del,sizeof(double),1,fp);
}
for(i=1;i<=OUT;i++)
{
        del=Outputo[i];
        fwrite(&del,sizeof(double),1,fp);
}
for(i=1;i<=OUT;i++)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        {
                del=dbo[i];
                fwrite(&del,sizeof(double),1,fp);
        }
        /*end Out */
        fwrite(&errorb,sizeof(double),1,fp);
        fwrite(&times,sizeof(int),1,fp);
        fclose(fp);
}
void loaddel(char *argv[])
{
int     i;
double  del;
FILE    *fp;
/* start save delta to file */
        if((fp=fopen(argv[11],"rb"))==NULL)
        {
                printf("\nError in reading file %s",argv[11]);
                exit(1);
        }
        /* H1 */
        for(i=1;i<=H;i++)
        {
                fread(&del,sizeof(double),1,fp);
                deltah1[i]=del;

        }
        for(i=1;i<=H;i++)
        {
                fread(&del,sizeof(double),1,fp);
                Outputh1[i]=del;
        }
        for(i=1;i<=H;i++)
        {
                fread(&del,sizeof(double),1,fp);
                dbh1[i]=del;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
}
/*end H1*/

/* H2 */
for(i=1;i<=H;i++)
{
        fread(&del,sizeof(double),1,fp);
        deltah2[i]=del;
}
for(i=1;i<=H;i++)
{
        fread(&del,sizeof(double),1,fp);
        Outputh2[i]=del;
}
for(i=1;i<=H;i++)
{
        fread(&del,sizeof(double),1,fp);
        dbh2[i]=del;
}
/*end H2*/

/*Out */
for(i=1;i<=OUT;i++)
{
        fread(&del,sizeof(double),1,fp);
        deltao[i]=del;
}
for(i=1;i<=OUT;i++)
{
        fread(&del,sizeof(double),1,fp);
        Outputo[i]=del;
}
for(i=1;i<=OUT;i++)
{
        fread(&del,sizeof(double),1,fp);
        dbo[i]=del;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        }
        /*end Out */
        fread(&errorb,sizeof(double),1,fp);
        fread(&times,sizeof(int),1,fp);
        fclose(fp);
}

void savew(char *argv[])
{
int    i,j,fr,fc;
double weight;
FILE   *fph1,*fpdh1,*fph2,*fpdh2,*fpo,*fpdo;

 /* start save all weight to file */
   /* save wh1[I][H] to argv[2] */
   if((fph1=fopen(argv[2],"wb"))==NULL)
   {
     printf("\nError in creating file %s",argv[2]);
     exit(1);
   }
   for(i=1;i<=I;i++)
   {
     for(j=1;j<=H;j++)
     {
         weight=wh1[i][j];
         fwrite(&weight,sizeof(double),1,fph1);
         if(ferror(fph1))
         {
          printf("\nError in writing file %s",argv[2]);
          exit(1);
         }
     }
   }
   fclose(fph1);

   /* save dwh1[I][H] to argv[3] */
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
if((fpdh1=fopen(argv[3],"wb"))==NULL)
{
  printf("\nError in creating file %s",argv[3]);
  exit(1);
}
for(i=1;i<=I;i++)
{
  for(j=1;j<=H;j++)
  {
       weight=dwh1[i][j];
       fwrite(&weight,sizeof(double),1,fpdh1);
       if(ferror(fph1))
       {
         printf("\nError in writing file %s",argv[3]);
         exit(1);
       }
  }
}
fclose(fpdh1);

/* THIS ADD */
/* save wh2[H][H] to argv[4] */
if((fph2=fopen(argv[4],"wb"))==NULL)
{
  printf("\nError in creating file %s",argv[4]);
  exit(1);
}
for(i=1;i<=H;i++)
{
  for(j=1;j<=H;j++)
  {
       weight=wh2[i][j];
       fwrite(&weight,sizeof(double),1,fph2);
       if(ferror(fph2))
       {
       printf("\nError in writing file %s",argv[4]);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        exit(1);
      }
  }
}
fclose(fph2);

/* save dwh2[H][H] to argv[5] */
if((fpdh2=fopen(argv[5],"wb"))==NULL)
{
  printf("\nError in creating file %s",argv[5]);
  exit(1);
}
for(i=1;i<=H;i++)
{
  for(j=1;j<=H;j++)
  {
      weight=dwh2[i][j];
      fwrite(&weight,sizeof(double),1,fpdh2);
      if(ferror(fph2))
      {
        printf("\nError in writing file %s",argv[5]);
        exit(1);
      }
  }
}
fclose(fpdh2);

/* THIS END ADD */

/* save wo[H][OUT] to argv[6] */
if((fpo=fopen(argv[6],"wb"))==NULL)
{
  printf("\nError in creating file %s",argv[6]);
  exit(1);
}
for(i=1;i<=H;i++)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  {
    for(j=1;j<=OUT;j++)
    {
        weight=wo[i][j];
        fwrite(&weight,sizeof(double),1,fpo);
        if(ferror(fpo))
        {
          printf("\nError in writing file %s",argv[6]);
          exit(1);
        }
    }
  }
  fclose(fpo);

  /* save dwo[H][OUT] to argv[7] */
  if((fpdo=fopen(argv[7],"wb"))==NULL)
  {
    printf("\nError in creating file %s",argv[7]);
    exit(1);
  }
  for(i=1;i<=H;i++)
  {
    for(j=1;j<=OUT;j++)
    {
        weight=dwo[i][j];
        fwrite(&weight,sizeof(double),1,fpdo);
        if(ferror(fpdo))
        {
          printf("\nError in writing file %s",argv[7]);
          exit(1);
        }
    }
  }
  fclose(fpdo);
/* end save weight to file      */
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
void loadw(char *argv[])
{
int    i,j,fr,fc;
double weight;
FILE   *fph1,*fpdh1,*fph2,*fpdh2,*fpo,*fpdo;

 /* start save all weight to file */
   /* save wh1[I][H] to argv[2] */
   if((fph1=fopen(argv[2],"rb"))==NULL)
   {
     printf("\nError in reading file %s",argv[2]);
     exit(1);
   }
   for(i=1;i<=I;i++)
   {
     for(j=1;j<=H;j++)
     {
          fread(&weight,sizeof(double),1,fph1);
          wh1[i][j]=weight;
          if(ferror(fph1))
          {
            printf("\nError in reading file %s",argv[2]);
            exit(1);
          }
     }
   }
   fclose(fph1);

   /* load dwh1[I][H] to argv[3] */
   if((fpdh1=fopen(argv[3],"rb"))==NULL)
   {
     printf("\nError in reading file %s",argv[3]);
     exit(1);
   }
for(i=1;i<=I;i++)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
{
  for(j=1;j<=H;j++)
  {
        fread(&weight,sizeof(double),1,fpdh1);
        dwh1[i][j]=weight;
        if(ferror(fpdh1))
        {
          printf("\nError in reading file %s",argv[3]);
          exit(1);
        }
  }
}
fclose(fpdh1);

/* THIS ADD */
/* load wh2[H][H] from argv[4] */
if((fph2=fopen(argv[4],"rb"))==NULL)
{
  printf("\nError in reading file %s",argv[4]);
  exit(1);
}
for(i=1;i<=H;i++)
{
  for(j=1;j<=H;j++)
  {
        fread(&weight,sizeof(double),1,fph2);
        wh2[i][j]=weight;
        if(ferror(fph2))
        {
          printf("\nError in reading file %s",argv[4]);
          exit(1);
        }
  }
}
fclose(fph2);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/* load dwh2[H][H] from argv[5] */
if((fpdh2=fopen(argv[5],"rb"))==NULL)
{
  printf("\nError in reading file %s",argv[5]);
  exit(1);
}
for(i=1;i<=H;i++)
{
  for(j=1;j<=H;j++)
  {
       fread(&weight,sizeof(double),1,fpdh2);
       dwh2[i][j]=weight;
       if(ferror(fpdh2))
       {
         printf("\nError in reading file %s",argv[5]);
         exit(1);
       }
  }
}
fclose(fpdh2);

/* THIS END ADD */

/* load wo[H][OUT] from argv[6] */
if((fpo=fopen(argv[6],"rb"))==NULL)
{
  printf("\nError in reading file %s",argv[6]);
  exit(1);
}
for(i=1;i<=H;i++)
{
  for(j=1;j<=OUT;j++)
  {
       fread(&weight,sizeof(double),1,fpo);
       wo[i][j]=weight;
       if(ferror(fpo))
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        {
          printf("\nError in reading file %s",argv[6]);
          exit(1);
        }
      }
    }
    fclose(fpo);

    /* load dwo[H][OUT] from argv[7] */
    if((fpdo=fopen(argv[7],"rb"))==NULL)
    {
      printf("\nError in reading file %s",argv[7]);
      exit(1);
    }
    for(i=1;i<=H;i++)
    {
      for(j=1;j<=OUT;j++)
      {
          fread(&weight,sizeof(double),1,fpdo);
          dwo[i][j]=weight;
          if(ferror(fpdo))
          {
            printf("\nError in reading file %s",argv[7]);
            exit(1);
          }
      }
    }
    fclose(fpdo);
  /* end load weight from file       */

}

void loadb(char *argv[])
{
int    i,j,fn;
FILE   *fbh1,*fbh2,*fbo;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
double bias;

/* start load all weight from file */
   /* load bh1[H] from argv[8] */
   if((fbh1=fopen(argv[8],"rb"))==NULL)
   {
     printf("\nError in reading file %s",argv[8]);
     exit(1);
   }
   for(i=1;i<=H;i++)
   {
     fread(&bh1[i],sizeof(double),1,fbh1);
     if(ferror(fbh1))
     {
          printf("\nError in reading file %s",argv[8]);
          exit(1);
     }
   }
   fclose(fbh1);
   /* this add */
   /* load bh2[H] from argv[9] */
   if((fbh2=fopen(argv[9],"rb"))==NULL)
   {
     printf("\nError in reading file %s",argv[9]);
     exit(1);
   }
   for(i=1;i<=H;i++)
   {
     fread(&bh2[i],sizeof(double),1,fbh2);
     if(ferror(fbh2))
     {
          printf("\nError in reading file %s",argv[9]);
          exit(1);
     }
   }
   fclose(fbh2);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
   /* end this add */

   /* load bo[OUT] to argv[10] */
   if((fbo=fopen(argv[10],"rb"))==NULL)
   {
     printf("\nError in reading file %s",argv[10]);
     exit(1);
   }
   for(i=1;i<=OUT;i++)
   {
     fread(&bo[i],sizeof(double),1,fbo);
     if(ferror(fbo))
     {
          printf("\nError in reading file %s",argv[10]);
          exit(1);
     }
   }
   fclose(fbo);
 /* end load bias from file      */

}

void saveb(char *argv[])
{
int    i,j,fn;
FILE   *fbh1,*fbh2,*fbo;

 /* start save all weight to file */
   /* save bh1[H] to argv[8] */
   if((fbh1=fopen(argv[8],"wb"))==NULL)
   {
     printf("\nError in creating file %s",argv[8]);
     exit(1);
   }
   for(i=1;i<=H;i++)
   {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  fwrite(&bh1[i],sizeof(double),1,fbh1);
  if(ferror(fbh1))
  {
      printf("\nError in writing file %s",argv[8]);
      exit(1);
  }
}
fclose(fbh1);
/* this add */
/* save bh2[H] to argv[9] */
if((fbh2=fopen(argv[9],"wb"))==NULL)
{
  printf("\nError in creating file %s",argv[9]);
  exit(1);
}
for(i=1;i<=H;i++)
{
  fwrite(&bh2[i],sizeof(double),1,fbh2);
  if(ferror(fbh2))
  {
      printf("\nError in writing file %s",argv[9]);
      exit(1);
  }
}
fclose(fbh2);
/* end this add */

/* save bo[OUT] to argv[10] */
if((fbo=fopen(argv[10],"wb"))==NULL)
{
  printf("\nError in creating file %s",argv[10]);
  exit(1);
}
for(i=1;i<=OUT;i++)
{
  fwrite(&bo[i],sizeof(double),1,fbo);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        if(ferror(fbo))
        {
            printf("\nError in writing file %s",argv[10]);
            exit(1);
        }
      }
      fclose(fbo);
  /* end save bias to file       */
}

void initdel(void)
{
int i;

        /* H1 */
        for(i=1;i<=H;i++)
        {
        deltah1[i]=0.0;
        Outputh1[i]=0.0;
        dbh1[i]=0.0;
        }
        /* H2 */
        for(i=1;i<=H;i++)
        {
        deltah2[i]=0.0;
        Outputh2[i]=0.0;
        dbh2[i]=0.0;
        }
       /* OUT */
       for(i=1;i<=OUT;i++)
       {
       deltao[i]=0.0;
       Outputo[i]=0.0;
       dbo[i]=0.0;
       }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
}/* end initdel() */

void createb(void)
{
int i;
        /* (H1)bias in hidden layer1 */
        for(i=1;i<=H;i++)
        {
         bh1[i]=(rand()%1000)/1000.00;
         if(rand()%2)
                bh1[i]=-1.00*bh1[i];
        }
        /* (H2)bias in hidden layer2 */
        for(i=1;i<=H;i++)
        {
         bh2[i]=(rand()%1000)/1000.00;
         if(rand()%2)
                bh2[i]=-1.00*bh2[i];
        }
        /* (OUT)bias in output layer */
        for(i=1;i<=OUT;i++)
        {
         bo[i]=(rand()%1000)/1000.00;
         if(rand()%2)
                bo[i]=-1.00*bo[i];
        }
}/* end createb() */
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประวัติผู้เขียน

ชื่อผู้เขียน	นายทวี เปรมรัตนชัย
วัน เดือน ปีเกิด	วันที่ 29 ตุลาคม 2513
สถานที่เกิด	จังหวัดกรุงเทพฯ
วุฒิการศึกษาระดับปริญญาตรี	วิศวกรรมศาสตร์บัณฑิต
	สาขาวิชาวิศวกรรมคอมพิวเตอร์
สถานที่สำเร็จการศึกษา	มหาวิทยาลัยเกษตรศาสตร์ บางเขน
ปีที่สำเร็จการศึกษา	2536
อาชีพปัจจุบัน	วิศวกรคอมพิวเตอร์ บริษัท ทีพีไอ จำกัด (มหาชน)

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้