การรู้จำตัวอักษรพิมพ์ภาษาลาวโดยใช้ฟัซซี่ และนิวรอลเน็ตเวิร์กแบบแพร่ย้อนกลับ

THE FUZZY AND BACK-PROPAGATION NEURAL TECHNIQUES IN RECOGNITION OF THE PRINTED LAO CHARACTERS

วิชิน ปุ่นคีน
VICHIN PUNCHEEN

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชาวิศวกรรมไฟฟ้า
บัณฑิตวิทยาลัย
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
พ.ศ. 2543

ISBN 974-622-974-5

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

การรู้จำตัวอักษรพิมพ์ภาษาลาวโดยใช้ ฟัซซี่ และนิวรอลแบบแพร่ย้อนกลับ

THE FUZZY AND BACK-PROPAGATION NEURAL TECHNIQUES IN RECOGNITION OF THE PRINTED LAO CHARACTERS



วิซิน มุ่งสิน

VIXINH MOUNGSINH

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต

สาขาวิชาวิศวกรรมไฟฟ้า

บัณฑิตวิทยาลัย

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เลขหมู่.....
เลขทะเบียน..... 38046
วัน, เดือน, ปี..... 20 พ.ย. 2543

พ.ศ. 2543

ISBN 974-622-974-5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

THE FUZZY AND BACK-PROPAGATION NEURAL TECHNIQUES IN RECOGNITION OF THE PRINTED LAO CHARACTERS

VIXINH MOUNGSINH

A THESIS SUBMITTED IN PARTIAL FULFILLMENT

OF THE REQUIREMENT FOR THE DEGREE OF

MASTER OF ENGINEERING IN ELECTRICAL ENGINEERING

SCHOOL OF GRADUATE STUDIES

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

2000

ISBN 974-622-974-5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



COPYRIGHT 2000

SCHOOL OF GRADUATE STUDIES

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บัณฑิตวิทยาลัย

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ใบรับรองวิทยานิพนธ์

หัวข้อวิทยานิพนธ์ การรู้จำตัวอักษรพิมพ์ภาษาลาวโดยใช้ฟัซซี่ และนิวรอลแบบแพร่ย้อนกลับ
THE FUZZY AND BACK-PROPAGATION NEURAL TECHNIQUES IN RECOGNITION OF THE PRINTED LAO CHARACTERS

ชื่อนักศึกษา นายวิซิน มุ่งสิน

รหัสประจำตัว 40061040

ปริญญา วิศวกรรมศาสตรมหาบัณฑิต

สาขาวิชา วิศวกรรมไฟฟ้า

อาจารย์ผู้ควบคุมวิทยานิพนธ์ รศ.ดร.ชม กิ้มปาน

คณะกรรมการสอบวิทยานิพนธ์		ลายมือชื่อ
ผศ.ดร.เอื้อน	ปิ่นเงิน	
รศ.ประทีป	บัญญัตินพรัตน์	
ผศ.ดร.บุญธีร์	เครือตราชู	
ดร.ชุติเมษฎ์	ศรีนิลทา	
รศ.ดร.ชม	กิ้มปาน	

วัน/เดือน/ปี ที่สอบ 3 ตุลาคม 2543 เวลา 12.00-13.00 น.

สถานที่สอบ ห้องสอบวิทยานิพนธ์ คณะวิศวกรรมศาสตร์ ตึก 12 ชั้น 4 ห้อง (E12-403)

บัณฑิตวิทยาลัยรับรองแล้ว

(รศ.ดร.บุญวัฒน์ อัตชู)

รักษาราชการแทนคณบดีบัณฑิตวิทยาลัย

วันที่ 20 **เดือน** ตุลาคม **พ.ศ.** 2543

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงได้อย่างดีด้วยคำแนะนำและคำปึกษาเกี่ยวกับการรู้จำตัวอักษร รวมทั้งการตรวจสอบเนื้อหาตลอดจนห้องทดลองและอุปกรณ์เครื่องใช้ในการทำงานวิจัยจาก รศ.ดร. ชม กิ้มปาน ซึ่งเป็นอาจารย์ผู้ควบคุมวิทยานิพนธ์ ผู้วิจัยรู้สึกซาบซึ่งในความอนุเคราะห์จากท่านและขอกราบขอบพระคุณเป็นอย่างสูง

ขอขอบพระคุณ อ. พิศิษฏ์ โภคารัตน์กุล นายอุกฤษณ์ มารังค์ และเพื่อนทุกคนที่ช่วยเหลือแนะนำทฤษฎีอื่นๆพร้อมทั้งช่วยตรวจเทียบและแก้ไขส่วนที่ผิดพลาด จนสำเร็จสมบูรณ์และยังให้กำลังใจผู้วิจัยอย่างใกล้ชิดตลอดมา

สุดท้ายขอขอบพระคุณผู้ที่ให้ทุนการศึกษา ขอขอบพระคุณบัณฑิตวิทยาลัยที่อำนวยความสดวกให้การศึกษาความรู้ต่างๆ ขอขอบพระคุณอาจารย์ทุกท่านที่ทำการสอนให้ความรู้ทุกอย่างแก่ผู้วิจัยครั้งนี้ คุณค่าและประโยชน์อันพึงมีจากวิทยานิพนธ์ฉบับนี้ ผู้วิจัยขอมอบแด่ผู้มีพระคุณทุกท่าน ถ้ามีความผิดพลาดใดๆผู้วิจัยขอน้อมรับไว้แต่เพียงผู้เดียว

นายวิชิน มุ่งสิน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

หน้า

บทคัดย่อภาษาไทย.....I

บทคัดย่อภาษาอังกฤษ.....II

กิตติกรรมประกาศ.....III

สารบัญ.....IV

สารบัญตาราง.....VI

สารบัญรูป.....VII

บทที่ 1. บทนำ.....1

1.1 ความเป็นมาและความสำคัญของปัญหา.....1

1.2 วัตถุประสงค์ของงานวิจัย.....3

1.3 ข้อตกลงเบื้องต้นในการทำวิทยานิพนธ์.....3

บทที่ 2. ขั้นตอนของการเตรียมข้อมูลตัวอักษร.....4

2.1 การเปลี่ยนข้อมูลบิตแมปไปเป็นไบนารี่.....4

2.2 การแยกตัวอักษร.....6

2.3 การปรับตัวอักษรให้เป็นขนาดมาตรฐาน.....6

2.4 การหาคุณลักษณะเด่นของตัวอักษร.....6

บทที่ 3. ทฤษฎีของฟัซซี่ลอจิก (Fuzzy logic).....11

3.1 คริสป์หรือฟัซซี่(Crisp or Fuzzy logic).....11

3.2 คริสป์เซตและฟัซซี่เซต(Crisp sets and Fuzzy Sets).....11

3.3 การคำนวณในฟัซซี่เซต(Fuzzy Set operation).....13

3.4 การแสดงค่าความเป็นสมาชิก.....14

3.5 การวิเคราะห์ฟัซซี่ลอจิกในระบบควบคุม.....18

3.6 การประยุกต์ใช้ฟัซซี่เซตสำหรับการรู้จำตัวอักษร.....21

บทที่ 4. ทฤษฎีนิวรอลเน็ตเวอร์ค (Neuraletwork).....27

4.1 ลักษณะโครงสร้างของนิวรอล.....28

4.2 โครงข่ายนิวรอลแบบแพร่ย้อนกลับ.....31

4.3 การใช้โครงข่ายนิวรอลในการรู้จำตัวอักษร.....41

บทที่ 5. โปรแกรมการรู้จำ.....43

5.1 การเตรียมข้อมูล.....43

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

หน้า

5.2 การฝึกสอน 45

5.3 การจดจำ.....45

บทที่ 6. การทดลองและประเมินผล.....48

6.1 กำหนดจำนวนโหนดของชั้นฮิดเดนเลเยอร์ที่เหมาะสม.....48

6.2 ทำการทดลองในกลุ่มตัวอักษาที่มีรูปแบบเดียวกันแต่มีขนาดแตกต่างกัน.....49

6.3 การทดลองในกลุ่มตัวอักษรที่มีรูปแบบและขนาดแตกต่างกัน.....49

6.4 วิเคราะห์ผลการทดลอง.....50

6.5 สรุปประเมินผล และ แนวทางการพัฒนาต่อไป.....51

เอกสารอ้างอิง.....52

ภาคผนวก.....54

ประวัติผู้เขียน.....90

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่ หน้า

3.1 แสดงค่าความเป็นสมาชิกของตัวอักษรพิมพ์ภาษาลาวในระบบฟัซซี่เซต.....23

6.1 ผลการทดสอบโครงข่ายนิวรอลโดยกำหนดฮิดเดนโหนด.....48

6.2 แสดงผลการทดสอบการรู้จำตัวอักษรที่มีรูปแบบเดียวกันแต่ขนาดแตกต่าง.....49

6.3 แสดงผลการทดสอบการรู้จำตัวอักษรที่มีรูปแบบแตกต่างกัน.....50



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูปที่ หน้า

1.1 แสดงระบบการรู้จำตัวอักษรพิมพ์ภาษาลาว.....1

1.2 ตัวอักษรพิมพ์ภาษาลาวที่ได้ลดจำนวนลงมาแล้ว.....2

1.3 ตัวเลข Arabic.....2

1.4 ตัวพยัญชนะที่ติดกับสระ.....3

2.1 แสดงภาพ Bitmap ของตัวอักษร “ ก ” 4

2.2 แสดงโครงสร้าง Bitmap ความกว้างเป็นผลคูณของ 8.....5

2.3 แสดงโครงสร้าง Bitmap ความกว้างไม่เป็นผลคูณของ 8.....5

2.4 แสดงภาพ Binary ของตัวอักษร “ ก ” 6

2.5 แสดงการทำโปรเจ็กชั่นในแนวนอนและแนวตั้งของตัวอักษร “ ก ” 9

2.6 แสดงกราฟของการทำโปรเจ็กชั่นของตัวอักษร “ ก ” 10

(a) กราฟการโปรเจ็กชั่นตามแนวนอนของตัวอักษร

(b) กราฟการโปรเจ็กชั่นตามแนวตั้งของตัวอักษร

3.1 ฟังก์ชั่นความเป็นสมาชิกของคริส์ปเซต A และ ฟัซซี่เซต $\overline{A}$ 13

3.2 แสดงกราฟของฟังก์ชั่นความเป็นสมาชิกรูปสามเหลี่ยม.....15

3.3 แสดงกราฟของฟังก์ชั่นความเป็นสมาชิกรูปสี่เหลี่ยมคางหมู.....15

3.4 แสดงกราฟฟังก์ชั่นรูปตัว S(x ; a, b).....16

3.5 แสดงกราฟฟังก์ชั่นรูประฆังคว่ำ π(x; a, b, c).....17

3.6 แสดงกราฟของฟังก์ชั่นค่าความเป็นสมาชิกของกลุ่มอายุ.....18

3.7 Basic architecture of a fuzzy logic controller (FLC) 18

3.8 diagrammatic representation of fuzzy partition 19

3.9 แสดงกราฟ ของฟังก์ชั่นความเป็นสมาชิกในแต่ละเทอร์มเซต 23

4.1(a) แสดงส่วนประกอบของเซลประสาท 28

4.1(b) แสดงจุดถ่ายสัญญาณ(Signal Transmission)จากเซลประสาทหนึ่งหาเซลประสาทตัวอื่นๆ.....29

4.2 โมดูลนิวรอลที่สร้างขึ้นร่วมกับฟังก์ชั่นการกระตุ้น.....30

4.3 แสดงกราฟของฟังก์ชั่นการกระตุ้น h(x).....30

4.4 แสดงกราฟของฟังก์ชั่นการกระตุ้น $S_{\beta}(x)$.....31

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป(ต่อ)

รูปที่ หน้า

4.5 แสดงโครงสร้างของโครงข่ายนิวรอลแบบชั้นเดียว....31

4.6 แสดงโครงสร้างของโครงข่ายนิวรอลแบบหลายชั้น....32

4.7 แสดงระบบการฝึกสอนแบบชี้นำ (Supervised Learning)....33

4.8 แสดงระบบการฝึกสอนแบบเสริม (Reinforcement Learning)....33

4.9 แสดงระบบการฝึกสอนแบบอิสระ (Unsupervised Learning)....33

4.10 แสดงกฎการฝึกสอนโดยทั่วไปของโครงข่ายนิวรอล....34

4.11 แสดงระบบการฝึกสอนของโครงข่ายนิวรอล "perceptron"....36

4.12 แสดงลักษณะโครงข่ายนิวรอลที่มีโครงสร้าง 3 ชั้น....38

4.13 แสดงขั้นตอนการฝึกสอนของโครงข่ายนิวรอล....42

5.1 แสดงการอ่านข้อมูลจาก Bitmap file....43

5.2 แสดงการแยกบรรทัดออกจากหน้ากระดาษ....44

5.3 แสดงการแยกตัวอักษรออกจากบรรทัด....44

5.4 แสดงการแบ่งส่วน UP และ DONW....44

5.5 แสดงการแบ่งส่วน LEFT และ RIGHT....45

5.6 แสดงกราฟการทำงานของการฝึกสอน....46

5.7 แสดงการรู้จำตัวอักษร "¡" ให้ค่าเอาท์พุทเท่ากับ 1....46

5.8 แสดงการรู้จำตัวอักษร พร้อม ให้ค่าเอาท์พุท....47

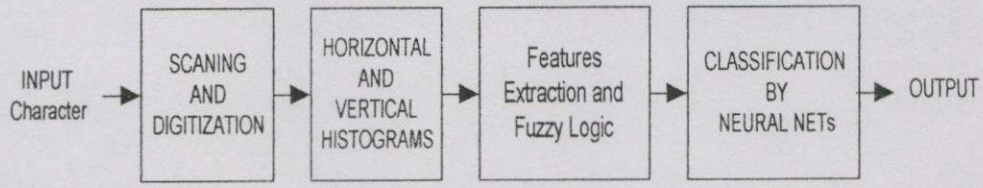
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ในปัจจุบันการรู้จำตัวอักษรได้มีการพัฒนาไปอย่างแพร่หลาย โดยเฉพาะการรู้จำตัวอักษรพิมพ์หรือเขียนแบบคัดลายมือของภาษาไทย โดยใช้กฎการวิเคราะห์โครงสร้างของตัวอักษรคือหาลักษณะเด่นของตัวอักษร (Feature extraction) และอีกวิธีก็คือเป็นการเก็บข้อมูลในเชิงสถิติของตัวอักษรแล้วนำเอาฟังก์ชั่นการตัดสินใจมาใช้ ในการพิจารณาการรู้จำตัวอักษร เช่น ผลงานวิจัยที่เสนอโดย ชม กิ้มปาน ในปี 1983 ที่กองประชุม IEEE การรู้จำตัวอักษรพิมพ์ภาษาไทยโดยใช้วิธี Matching Method คือวิเคราะห์โครงสร้างตัวอักษรและหาความเหมือนของตัวอักษรเหล่านั้น ในปี 1987 ชม กิ้มปาน ได้เสนอการรู้จำตัวอักษรพิมพ์ภาษาไทยโดยใช้กฎการแปลงข้อมูลแบบคาร์ฮูเน็นเลิฟ (Kahunen Love Transform) ในปี 1989 Airphaiboon S. ได้นำเสนอการรู้จำตัวอักษรแบบคัดลายมือภาษาไทยโดยใช้วิธีวิเคราะห์โครงสร้างส่วนหัวของตัวอักษร ในปี 1996 ทวี เปรมรัตนชัย ได้นำเสนอการรู้จำตัวอักษรพิมพ์ภาษาไทยโดยวิธีโครงข่ายนิวรอลแบบแพร่ย้อนกลับ ในปี 1999 พิศิษฎ์ โภคารัตน์กุล ได้นำเสนอ การรู้จำตัวอักษรพิมพ์ภาษาไทยโดยวิธีวัดความเหมือนแบบฟัซซี่ ส่วนงานวิจัยการรู้จำตัวอักษรในภาษาลาวนั้น ในปี 1998 นาย ขิงทอง อินทวงศ์คำ ได้นำเสนอการรู้จำลายมือเขียนตัวอักษรภาษาลาวโดยใช้วิธีการเข้ารหัสลูกโซ่ (Chain code) และ โครงข่ายประสาทเทียม (Neural Networks)

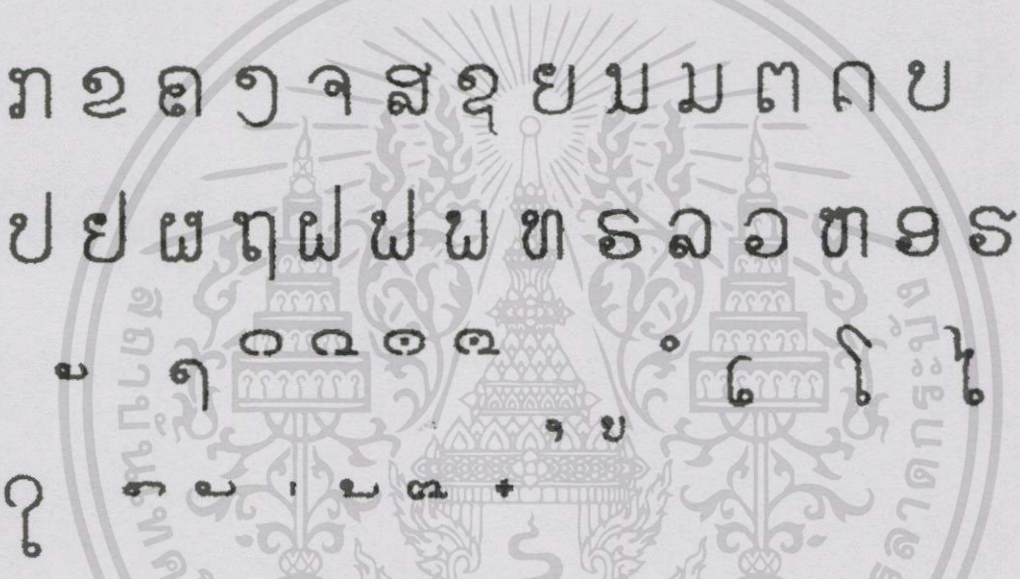
ส่วนงานวิจัยนี้เป็นการเสนออีกวิธี คือการรู้จำตัวอักษรพิมพ์ภาษาลาว โดยการใช้เทคนิคของระบบ ฟัซซี่ และ นิวรอลแบบแพร่ย้อนกลับในการรู้จำ ซึ่งระบบการรู้จำตัวอักษรได้แสดงในรูปที่ 1.1 สำหรับตัวอักษรพิมพ์ภาษาลาวมีทั้งหมด 66 ตัว คือมีพยัญชนะ 33 ตัว สระ 29 ตัว และวรรณยุกต์ 4 ตัว (ดูในภาคผนวก ข) แต่เมื่อลดรูปลงมาจะเหลืออยู่ 46 ตัว ดังที่แสดงในรูปที่ 1.2 มีพยัญชนะ 27 ตัว สระ 15 ตัวและวรรณยุกต์ 4 ตัว เพราะว่ามีตัวอักษรและสระบางตัวเป็นตัวอักษรควบ เช่น ตัวอักษร "ໜາ" ควบด้วยตัวอักษร "ໜ" และ "ງ" หรือสระ "ແX" เมื่อลดรูปจะได้ "ໂX" การลดรูปตัวอักษรลงนั่นก็เพื่อลดขนาดของข้อมูลที่จะนำมาใช้ในการรู้จำให้น้อยลงเท่าที่จำเป็น



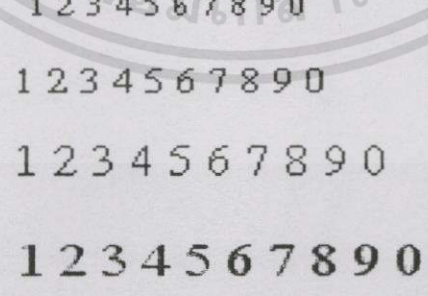
รูปที่ 1.1 แสดงระบบการรู้จำตัวอักษรพิมพ์ภาษาลาว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับตัวเลขที่ใช้ในภาษาลาวส่วนมากก็นิยมใช้ตัวเลขอารบิค (arabic) แสดงในรูปที่ 1.3 ตัวอักษรพิมพ์ภาษาลาวในปัจจุบันที่นิยมใช้จะมี 9 รูปแบบที่แตกต่างกัน แต่ในการวิจัยนี้จะเลือกตัวอักษรพิมพ์ภาษาลาวมาใช้เป็นต้นแบบในการฝึกสอนเรียนรู้มีเพียง 8 รูปแบบคือ "Alice0", "Alice1", "Alice2", "Alice3", "Alice4", "Alice5", "Chanthaboulilao" และ "Saysetthalao" นอกนั้นตัวอักษรพิมพ์ภาษาลาวยังมีบางคำที่ตัวพยัญชนะติดกันกับสระบางตัวเราไม่สามารถแยกออก ตัวอย่างแสดงในรูปที่ 1.4 ตัวอักษรพิมพ์ที่ติดกันเหล่านี้ เราได้นำมาฝึกสอนรวมเข้าไปด้วยเพื่อลดความผิดพลาดให้น้อยลง (เพราะในงานวิจัยนี้ยังไม่ได้ทำวิธีการแยกส่วนที่ติดกันของตัวอักษร)

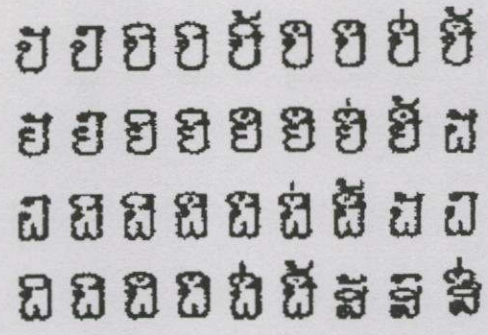


รูปที่ 1.2 ตัวอักษรพิมพ์ภาษาลาวที่ได้ลดจำนวนลงมาแล้ว



รูปที่ 1.3 ตัวเลข Arabic

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 1.4 ตัวพยัญชนะที่ติดกับตัวสระ

1.2 วัตถุประสงค์ของงานวิจัย

วิทยานิพนธ์ฉบับนี้มีวัตถุประสงค์เพื่อเป็นการพัฒนาระบบการรู้จำตัวอักษร โดยการประยุกต์ใช้ทฤษฎีการวิเคราะห์โครงสร้างตัวอักษร โดยนำเอากฎการเรียนรู้ของระบบฟัซซี่-นิวรอลเข้ามาใช้ในการรู้จำ เพื่อลดขนาดของข้อมูลและลดเวลาในการเรียนรู้ให้น้อยลง เพื่อเพิ่มความถูกต้องในการรู้จำให้มากขึ้น

1.3 ข้อตกลงเบื้องต้นในการทำวิทยานิพนธ์

งานวิจัยนี้ได้เน้นการศึกษาโครงสร้างตัวอักษรและการประยุกต์ใช้ทฤษฎีฟัซซี่-นิวรอลเข้าในการรู้จำตัวอักษรพิมพ์ภาษาลาว ตัวอักษรพิมพ์ที่นำมาใช้ในงานวิจัยนี้เน้น 4 รูปแบบที่มีคุณลักษณะและขนาดแตกต่างกันไป ส่วนอุปกรณ์ที่เป็นเครื่องมือในการวิเคราะห์จะใช้ เครื่องคอมพิวเตอร์ที่เป็นรุ่น CPU: Intel Pentium II MMX 350 MHz RAM: 64 MB Harddisk: 8 GB เครื่องอ่านภาพ (Scanner) รุ่น Hewlett Packard ScanJet 6100C และกำหนดคุณลักษณะในการอ่านภาพดังนี้

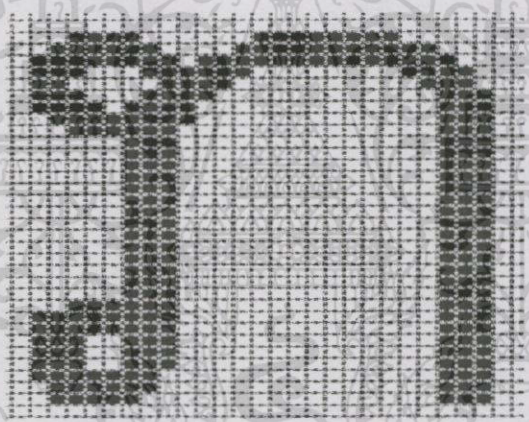
- Type: Black & White Drawing
- Brightness: 100
- Contrast: 100
- Scanling: 100 %

นอกจากนี้ยังใช้โปรแกรม MATLAB V.5.1 มาช่วยประมวลผลในการรูจำตัวอักษร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ขั้นตอนของการเตรียมข้อมูลตัวอักษร

ขั้นตอนของการการเตรียมข้อมูลภาพของตัวอักษรพิมพ์ภาษาลาว เป็นกระบวนการในการเตรียมข้อมูลเบื้องต้นก่อนเข้าสู่ระบบการรู้จำตัวอักษร เมื่อป้อนข้อมูลตัวอักษรเข้าผ่านเครื่องตรวจวาดภาพจะได้ข้อมูลภาพตัวอักษรออกมาอยู่ในลักษณะของบิตแมป(Bitmap) แสดงในรูปที่ 2.1 ก่อนอื่นเราต้องแปลงข้อมูลภาพบิตแมป(Bitmap) ของตัวอักษรที่ได้จากเครื่องตรวจวาดภาพให้เป็นข้อมูลแบบไบนารี่(binary) คือในรูปตัวเลข 0 กับ 1 เสียก่อน จากนั้นก็ทำการแยกข้อมูลตัวอักษรออกจากหน้ากระดาษแต่ละตัวแล้วก็ทำการปรับเปลี่ยนตัวอักษรให้เป็นขนาดมาตรฐานในรูปแบบเมตริก จากนั้นจึ่งทำการวิเคราะห์โครงสร้างหาคุณลักษณะเด่นของตัวอักษร เพื่อนำไปประมวลผลในโครงข่ายแบบฟัซซี่-นิวรอล เป็นขั้นตอนต่อไป



รูปที่ 2.1 แสดงภาพ Bitmap ของตัวอักษร " ກ "

2.1 การเปลี่ยนข้อมูลภาพบิตแมปไปเป็นไบนารี่

ข้อมูลภาพกราฟฟิกแบบบิตแมปมีอยู่ 2 ลักษณะ โดยพิจารณาจากความกว้างของภาพ และ นำมาใช้เป็นหลักเกณฑ์ในการเปลี่ยนข้อมูลที่อยู่ในลักษณะเลขฐาน 16 เป็นเลขฐาน 2 คือ

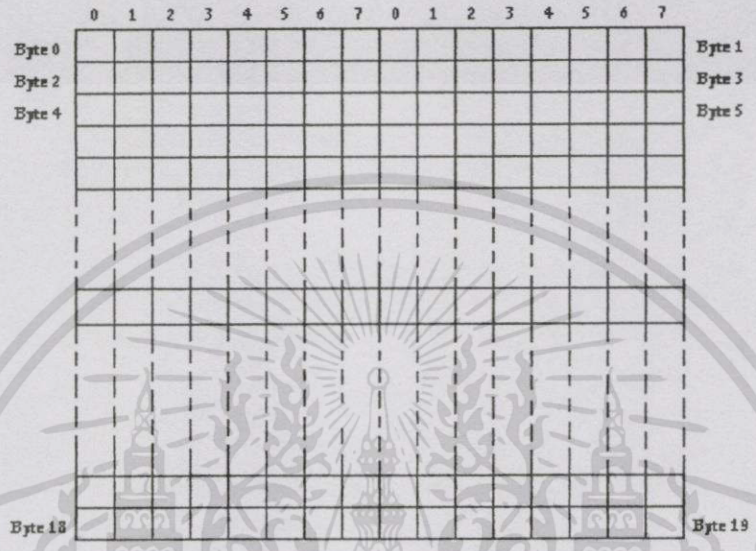
a) ถ้าความกว้างเป็นผลคูณของ 8 มีเงื่อนไขของโครงสร้างดังนี้(ดูรูปที่ 2.2)
   i) ข้อมูลในรูปของตัวเลข 0 หรือ 1 จะถูกจับเป็นกลุ่ม โดยเรียงกันไปจำนวน 8 บิท นั่นก็คือ 1 ไบท์
   ii) ลำดับของไบท์ของข้อมูลจะนับจากบนลงล่าง และ ซ้ายไปขวา
   iii) ลำดับของบิทในแต่ละไบท์จะนับจากซ้ายไปขวา และ บิททางซ้ายมือสุดนับเป็นบิทที่ 0 และบิททางขวามือสุดนับเป็นบิทที่ 7

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

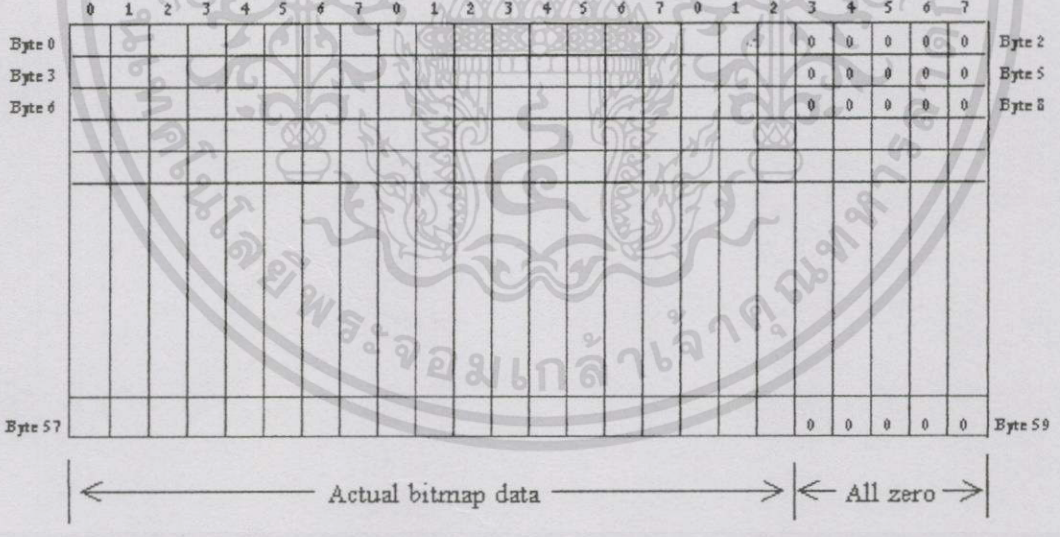
b) ถ้าความกว้างไม่เป็นผลคูณของ 8 จะมีเงื่อนไขดังนี้ (ดูรูปที่ 2.3)

i) จะนับความกว้างเป็นผลคูณของ 8 ที่น้อยที่สุดมีค่ามากกว่าความกว้างที่แท้จริงของรูป

ii) บิทที่เหลืออยู่นอกเหนือจากความกว้างที่แท้จริงจะถูกเติมด้วยเลข 0



รูปที่ 2.2 แสดงโครงสร้างบิตแมปที่ความกว้างเป็นผลคูณของ 8

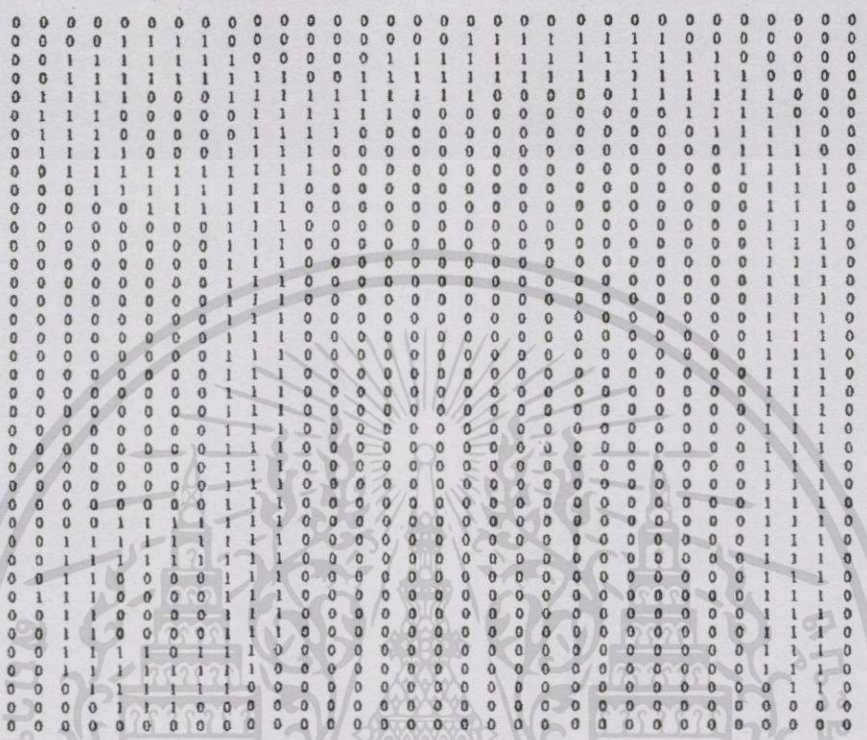


รูปที่ 2.3 แสดงโครงสร้างบิตแมปที่ความกว้างไม่เป็นผลคูณของ 8

เมื่อทราบรูปแบบข้อมูลภาพบิตแมปแล้วก็ทำการเปลี่ยนข้อมูลภาพที่เป็นเลขฐาน 16 ให้เป็นเลขฐาน 2 โดย 1 บิทของเลขฐาน 2 จะแทนจุดภาพ 1 จุด บิทที่มีค่าเป็น 0 ก็จะแทนจุดสว่างหรือจุดที่เป็นพื้นกระดาษสีขาว และ บิทที่มีค่าเป็น 1 ก็จะแทนจุดดำมืดทึบ ซึ่งก็คือจุดที่อาดเป็นเนื้อตัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อักษรหรือจุดสีดำของรูปภาพแสดงในรูปที่ 2.4 เราก็จะได้ข้อมูลภาพตัวอักษรที่ถูกต้อง และพ้อมที่จะส่งไปประมวลในขั้นต่อไป



รูปที่ 2.4 แสดงภาพไบนารี่ (Binary) ของตัวอักษร " ภ "

2.2 การแยกตัวอักษร

ก่อนอื่นเราจะแยกแต่ละบรรทัดอักษรออกจากหน้ากระดาษเสียก่อน โดยการพิจารณาหาค่ารหัสที่เป็น 0 ที่มีการต่อเนื่องกันตามแนวนอนที่เท่ากับความยาวของภาพตัวอักษร แล้วจึ่งแยกตัวอักษรแต่ละตัวออกจากบรรทัดโดยการพิจารณาหาค่ารหัสที่เป็น 0 ที่มีการต่อเนื่องกันตามแนวตั้งที่มีความยาวเท่ากับความกว้างของบรรทัดตัวอักษร เราทำการพิจารณาอย่างต่อเนื่องไปจนกว่าสิ้นสุดข้อมูลภาพหนึ่งหน้ากระดาษ

2.3 การปรับตัวอักษรให้เป็นขนาดมาตรฐาน

หลังจากแยกตัวอักษรเสร็จสิ้นเราก็ทำการปรับตัวอักษรแต่ละตัวให้เป็นรูปแบบมาตรฐานในกรอบ 90x90 จุดภาพ จากนั้นก็นำเอาภาพตัวอักษรที่ปรับเป็นมาตรฐานแล้วมาทำการประมวลเพื่อหาคุณลักษณะเด่นของตัวอักษรเป็นขั้นตอนต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4 การหาคุณลักษณะเด่นของตัวอักษร

เพื่อเป็นการลดข้อมูลภาพตัวอักษรพิมพ์ภาษาลาวให้น้อยลง เราได้กำหนดคุณลักษณะเด่นของตัวอักษรโดยเอาตามจุดภาพที่เป็นเนื้อตัวอักษรมาทำการประมวลผล

ถ้าให้ F_t เป็นรูปแบบเมตริกของตัวอักษร ที่มี $n \times m$ โดย t หมายถึงหมายเลขตัวอักษร n หมายถึงจำนวนของแถว(Row) และ m หมายถึงจำนวนของคอลัมน์(Column) และ P คือตำแหน่งจุดภาพตรงกลางของตัวอักษร ให้ $x_{i,j}$ เป็นจุดภาพของตัวอักษรในแต่ละรูปแบบที่มีค่าเท่า 1 หรือ 0 โดย $i = 1,2,3,\ldots..,\ p,\ldots..,\ n$ และ $j = 1,2,3,\ldots..,\ p,\ldots..,\ m$, เราสามารถเขียนได้ดังนี้

$$F_t = \begin{bmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,m-1} & x_{1,m} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,m-1} & x_{2,m} \\ \vdots & \vdots & \cdots & \vdots & \vdots \\ x_{p,1} & \cdots & x_{p,p} & \cdots & x_{p,j-1} \\ x_{i,1} & x_{i,2} & \cdots & x_{i,j-1} & x_{i,j} \\ \vdots & \vdots & \cdots & \vdots & \vdots \\ x_{n-1,1} & x_{n-1,2} & \cdots & x_{n-1,m-1} & x_{n-1,m} \\ x_{n,1} & x_{n,2} & \cdots & x_{n,m-1} & x_{n,m} \end{bmatrix} \qquad (2.1)$$

จากรูปแบบเมตริกของตัวอักษรที่ได้แสดงไว้ข้างบนนั้น(2.1) เราสามารถแบ่งการกำหนดออกเป็นส่วนๆได้ดังนี้

- กำหนดให้ TOT คือจำนวนรวมของจุดภาพที่เป็นเนื้อตัวอักษรทั้งหมด ดังนั้น เราสามารถเขียนสูตรการคำนวณได้คือ

$$TOT = \sum_{i=1}^{n}\sum_{j=1}^{m} x_{i,j} \qquad (2.2)$$

- กำหนด UP และ DOWN คือตำแหน่งจุดภาพที่เป็นเนื้อตัวอักษรส่วนบน และ ตำแหน่งจุดภาพที่เป็นเนื้อตัวอักษรส่วนล่าง ซึ่งให้ P คือตำแหน่งจุดภาพตรงกางตัวอักษร ดังนั้น เราสามารถเขียนสูตรการคำนวณได้คือ

$$UP = \sum_{i=1}^{p}\sum_{j=1}^{m} x_{i,j} \qquad (2.3)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$DOWN = \sum_{i=p+1}^{n}\sum_{j=1}^{m} x_{i,j} \qquad (2.4)$$

- กำหนด DW_UP คืออัตราส่วนระหว่างจำนวนจุดภาพเนื้อตัวอักษรส่วนบนและล่าง เราสามารถเขียนสูตรการคำนวณได้คือ

$$DW_UP = \frac{DOWN}{UP} \qquad (2.5)$$

- กำหยด LEFT และ RIGHT คือรวมตำแหน่งจุดภาพที่เป็นเนื้อตัวอักษรส่วนซ้าย และรวมตำแหน่งจุดภาพที่เป็นเนื้อตัวอักษรส่วนขวา ซึ่งให้ P คือตำแหน่งจุดภาพตรงกางตัวอักษร เราสามารถเขียนสูตรการคำนวณได้คือ

$$LEFT = \sum_{i=1}^{n}\sum_{j=1}^{p} x_{i,j} \qquad (2.6)$$

$$RIGHT = \sum_{i=1}^{n}\sum_{j=p+1}^{m} x_{i,j} \qquad (2.7)$$

- กำหนด RG_LF คืออัตราส่วนระหว่างจำนวนจุดภาพของเนื้อตัวอักษรขวาและซ้าย เราสามารถเขียนสูตรการคำนวณได้คือ

$$RG_LF = \frac{RIGHT}{LEFT} \qquad (2.8)$$

- ทำการโปรเจ็คชั่นตามแนวนอน (h_1-h_n) และ ตามแนวตั้ง(V_1-V_m) ตามลำดับดังแสดงในรูปที่ 2.5 เพื่อกำหนดเอาจุด MAxH คือตำแหน่งจุดภาพที่มากที่สุดในแนวนอน และหา MAxV คือตำแหน่งจุดภาพที่มากที่สุดในแนวตั้ง ซึ่งข้อมูลโปรเจ็กชั่นฟีเจอร์จะขึ้นอยู่กับลักษณะของรูปแบบตัวอักษร

สำหรับการทำโปรเจ็กชั่น ถ้าให้ H^{t}_{PROJ} เป็นเว็คเตอร์ที่เราสแกนตามแนวนอนและ n คือจำนวนแถวของเว็คเตอร์เมตริกตัวอักษร เราสามารถเขียนได้ดังนี้

$$H^{t}_{PROJ} = [h_1, h_2, h_3, ..., h_i, ..., h_n]^T \qquad (2.9)$$

$$\text{ซึ่ง } h_i = \sum_{j=1}^{m} x_{i,j} \;/\; i = 1,2,3,...., n$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากนั้นจะได้

$$MaxH = \max(H^t_{PROJ}) \qquad (2.10)$$

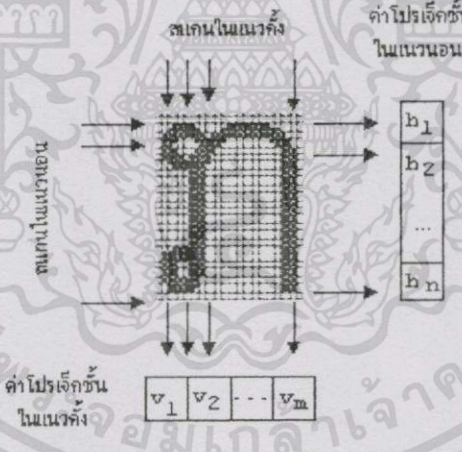
ให้ V^t_{PROJ} เป็นเวิ์คเตอร์ที่เราสแกนตามแนวตั้ง และ m จำนวนคอลัมน์ของเวิ์คเตอร์เมตริกตัวอักษร เราสามารถเขียนได้ดังนี้

$$V^t_{PROJ} = \left[v_1, v_2, \ldots, v_j, \ldots\ldots, v_m\right] \qquad (2.11)$$

$$\text{ซึ่ง } v_j = \sum_{i=1}^{n} x_{i,j} \quad / \quad j = 1,2,3,\ldots., m$$

จากนั้นจะได้

$$MaxV = \max(V^t_{PROJ}) \qquad (2.12)$$



รูปที่ 2.5 แสดงการทำโปรเจ็กชั่นในแนวนอนและแนวตั้งของตัวอักษร "ภ"

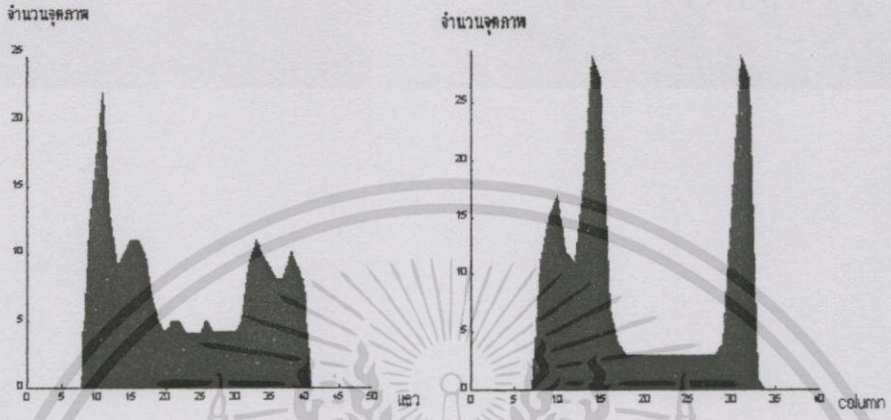
ในรูปที่ 2.6(a) และ (b) แสดงกราฟของการโปรเจ็กชั่นตามแนวนอน H^t_{PROJ} และ การโปรเจ็กชั่นตามแนวตั้ง V^t_{PROJ} ของตัวอักษร "ภ" ตามลำดับ โดยการประมวลผลจากตัวอักษรที่เป็นขนาดมาตรฐานแล้ว

ผ่านกระบวนการประมวลผลหาค่าต่างๆแล้ว เราก็จะได้คุณลักษณะเด่นของตัวอักษร โดยสามารถเขียนเป็นรูปแบบของเวิ์กเตอร์ได้คือ

$$F_t = \{TOT, UP, DOWN, LEFT, RIGHT, MaxH, MaxV, DW_UP, RG_LF\}^T \quad (2.13)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากนั้นเราก็นำเอาค่าเหล่านี้ไปประมวลผลในระบบของฟัซซี่ เป็นขั้นตอนต่อไป



(a) (b)

รูปที่ 2.6 แสดงกราฟของการทำโปรเจ็กชั่นของตัวอักษร " ป "

(a) กราฟการโปรเจ็กชั่นตามแนวนอนของตัวอักษร

(b) กราฟการโปรเจ็กชั่นตามแนวตั้งของตัวอักษร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
ทฤษฎีของฟัซซี่ลอจิก (Fuzzy logic)

ทฤษฎีฟัซซี่เซต(Fuzzy sets)ได้ถูกเสนอโดย Lotfi Zadeh ในปี 1965 [CHIN TENG LIN & C.S. GEORGE LEE. 1996:3] หลังจากนั้นได้มีผู้นำเอาทฤษฎีนี้ไปพัฒนาและประยุกต์ใช้กับงานระบบควบคุมประมวนผลสัญญาณดิจิตอล การรู้จำภาพและตัวอักษร การวิเคราะห์ทางเศรษฐศาสตร์ และ อื่นๆมากมายเช่นเครื่องซักผ้า กล้องถ่ายรูปตลอดเครื่องปรับอากาศในบ้าน

คำว่าฟัซซี่มีความหมายว่าความไม่ชัดเจน คลุมเครือ การสื่อความหมายหรือการตีความและการตัดสินในชีวิตจริงของคนเรานั้นมักจะมีลักษณะที่เป็นฟัซซี่ ในภาคนี้เราจะนำเสนอหลักการและความรู้ในการคำนวณเกี่ยวกับฟัซซี่เซต

3.1 คริส์ป หรือ ฟัซซี่ (Crisp or Fuzzy logic)

ลอจิกหมายถึง เหตุผลที่ใช้ในการแก้ปัญหาต่าง ๆ เกี่ยวกับค่าที่เป็นจริงและเท็จ(true and false) ข้อเสนอหนึ่งสามารถเป็นจริงเพียงโอกาส เดียวและเป็นเท็จในโอกาสอื่น[Valluru Rao & Hayagriva Rao. 1995:31] "ผลแอปเปิลสีแดง" เป็นข้อเสนอหนึ่ง ถ้าพบว่าหาได้ผลแอปเปิลสีเขียว ข้อเสนอดังกล่าวจะเป็นเท็จ แต่ถ้าพบว่า ผลแอปเปิลสีแดง ข้อเสนอนั้นจะเป็นจริง ดังนั้นถ้าข้อเสนอเป็นจริงค่าจริงเท่ากับ 1 ถ้าเป็นเท็จ ค่าจริงเท่ากับ 0 แต่ค่าจริงมีเพียงค่าเดียวเท่านั้นที่เป็นไปได้ มีหลายข้อเสนอสามารถรวมเข้ากันเพื่อให้เกิดเป็นข้อเสนอตัวใหม่ ด้วยความหมายของการคำนวณลอจิก

เมื่อคุณพูดว่า " ฝนจะตกวันนี้ " หรือคุณออกไปข้างนอก เพื่อรับประทานอาหารกลางแจ้ง คุณได้สร้างข้อเสนอด้วยความแน่ใจ ในกรณีนี้ข้อเสนอของคุณสามารถเป็นจริงหรือเท็จได้ทั้งสองและ ค่าจริงจะเท่ากับ 1 หรือ 0 เท่านั้น ดังนั้น ข้อเสนอของคุณสามารพูดได้ว่าเป็น "คริส์ป"

นอกจากนี้ยังมีข้อเสนอที่คุณไม่สามารถสร้างขึ้นด้วยความแน่ใจ คุณอาจพูดว่า "ผมคิดว่าฝนจะตกวันนี้" ถ้าขยายความออกไปคือคุณสามารถแบ่งระดับความแน่นอนในข้อเสนอของคุณ ซึ่งจะมีค่าประมาณ 0.8 ให้ชัดเจนลงไปกว่าค่า 1 ด้วยเหตุนี้ ฟัซซี่ลอจิกได้ถูกพัฒนาขึ้นมา ดังนั้น ฟัซซี่ลอจิกประกอบด้วยข้อเสนอที่แบ่งระดับค่าความเป็นจริงแต่ 0 หา 1

3.2 คริส์ปเซต และ ฟัซซี่เซต (Crisp sets and Fuzzy sets)

คริส์ปเซตคือกลุ่มที่รวบรวมสิ่งของแตกต่างกันโดยแยกเอาองค์ประกอบออกเป็นสองกลุ่มต่างกันคือ "สมาชิกและไม่เป็นสมาชิก" เราสามรถกำหนดฟังก์ชั่นได้ที่เรียกว่า "ฟังก์ชั่นคุณลักษณะ(Characteristic Function) หรือ ฟังก์ชั่นความเป็นสมาชิก" ให้ U เป็นเซตเอกภพ (universal

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

set) ให้ค่าความเป็นสมาชิก $\mu_A(x)$ ของคริสป์เซต A ใน U มีค่าของมันใน {0,1} และได้กำหนดค่าความเป็นสมาชิก $\mu_A(x)$=1 ถ้า x เป็นสมาชิกของ A (i.e. $x \in A$) และ 0 นอกนั้น เราเขียนสมการได้ดังนี้

$$\mu_A(x) = \begin{cases} 1 & \text{if and only } x \in A \\ 0 & \text{if and only } x \notin A \end{cases} \qquad (3.1)$$

ควรใส่ใจว่า (i) ขอบเขตของเซต A แข็งและชันและประกอบด้วยสองประเภทแตกต่างกัน (i.e., $x \in A$ or $x \notin A$), และ (ii) เซตเอกภพ U ก็เป็นคริสป์เซตเช่นกัน

ฟัซซี่เซต (fuzzy set) จะมีความแตกต่างจากคริสป์เซตคือในระบบคริสป์เซตจะบอกได้เพียงว่า x เป็นสมาชิกของเซต A (ค่าเป็นสมาชิก = 1) หรือไม่เป็นสมาชิกของเซต A (ค่าเป็นสมาชิก = 0) แต่ในฟัซซี่เซตนั้นจะสามารถยอมรับสมาชิกที่มีลักษณะถูกต้องเพียงบางส่วนหรือผิดเพียงบางส่วน กล่าวคือจะมีระดับค่าความเป็นสมาชิกในระหว่าง {0,1} ถ้าให้ U เป็นเซตเอกภพ แล้วฟัซซี่เซต $\overline{A}$ ใน U สามารถกำหนดได้ว่า

$$\overline{A} = \left\{ \left(x, \mu_{\overline{A}}(x) \right) / \, x \in U \right\} \qquad (3.2)$$

$\mu_{\overline{A}}(.)$ เรียกว่าฟังก์ชั่นความเป็นสมาชิก(membership Function) หรือ ฟังก์ชั่นคุณลักษณะ) ของเซต $\overline{A}$

$\mu_{\overline{A}}(x)$ เป็นค่าระดับความเป็นสมาชิก (Grade or degree) ของ x ในเซต $\overline{A}$

ตัวอย่าง 3.1

ให้ U เป็นเอกภพจำนวนจริง

ให้ A คริสป์เซต เป็นเซตของจำนวนจริงที่มีค่าใหญ่กว่าและเท่ากับ 5 เรามี

$$A = \{(x, \mu_A(x)) \mid x \in U\}$$

ฟังก์ชั่นความเป็นสมาชิกเราสามารถกำหนดได้คือ

$$\mu_A(x) = \begin{cases} 0, & x < 5 \\ 1, & x \geq 5 \end{cases}$$

แสดงในรูปที่ 3.1(a)

ให้ ฟัซซี่เซต $\overline{A}$ เป็นเซตของจำนวนจริงใกล้เคียง 5 เราจะมี

$$\overline{A} = \left\{ (x, \mu_{\overline{A}}(x)) \mid x \in U \right\}$$

และ ฟังก์ชั่นความเป็นสมาชิกเราสมารถกำหนดได้คือ

$$\mu_{\overline{A}}(x) = \frac{1}{1 + 10(x-5)^2}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

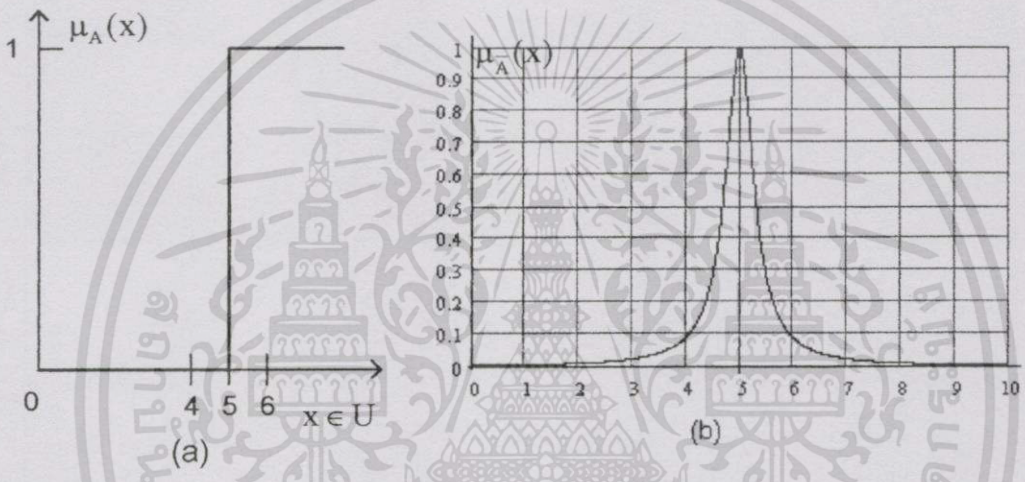
แสดงในรูปที่ 3.1(b)

ฟัซซี่เซต $\overline{A}$ ในตัวอย่างข้างบนนั้นสามารถเขียนได้ว่า

$$\overline{A} = \left\{ (x, \mu_{\overline{A}}(x)) \mid \mu_{\overline{A}}(x) = \left[1 + 10(x-5)^2\right]^{-1} \right\} \qquad (3.3)$$

ข้อควรจำว่า ในตัวอย่างที่ 3.1 นั้นเราสามารถกำหนดฟังก์ชั่นความเป็นสมาชิกเป็นอย่างอื่นได้อีก เช่นตัวอย่างว่า

$$\mu_{\overline{A}} = \frac{1}{1+(x-5)^2} \qquad (3.4)$$



รูปที่ 3.1 ฟังก์ชั่นความเป็นสมาชิกของคริสป์เซต A และ ฟัซซี่เซต $\overline{A}$

3.3 การคำนวณในฟัซซี่เซต

การคำนวณในฟัซซี่เซต ตามปกติ ถ้าเราให้ A และ B เป็นฟัซซี่เซตในเอกภพ U เราสามารถแสดงออกความสัมพันธ์ในเซตธรรมดาได้คือ

การเป็นคอมพลีเมน (Complement) ของฟัซซี่เซต เมื่อ $\mu_A(x) \in [0,1]$ คอมพลีเมนของฟัซซี่ A เราแทนด้วยฟัซซี่เซต $\overline{A}$ สามารถเขียนได้ฟังก์ชั่นความเป็นสมาชิกดังนี้

$$\mu_{\overline{A}}(x) = 1 - \mu_A(x), \quad \forall x \in U \qquad (3.5)$$

การเป็นยูเนียน (Union) ของฟัซซี่เซต A และ B ซึ่งมีฟังก์ชั่นความเป็นสมาชิก $\mu_A(x)$ และ $\mu_B(x)$ เราจะแทนด้วยฟัซซี่เซต $A \cup B$ สามารถเขียนค่าความเป็นสมาชิกได้ว่า

$$\mu_{A \cup B}(x) = \max\left(\mu_A(x), \mu_B(x)\right), \quad \forall x \in U \qquad (3.6)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การเป็นอินเตอร์เซคชั่น (Intersection) ของฟัซซี่เซต A และ Bซึ่งมีฟังก์ชั่นความเป็นสมาชิก $\mu_A(x)$และ $\mu_B(x)$ เราจะแทนด้วยฟัซซี่เซต $A \cap B$ สามารถเขียนค่าความเป็นสมาชิกได้ว่า

$$\mu_{A\cap B}(x) = \min(\mu_A(x), \mu_B(x)), \quad \forall x \in U \tag{3.7}$$

การเป็น Equality ของฟัซซี่เซต A และ B ถ้าว่า A และ B เท่ากัน ก็ต่อเมื่อว่า

$$\mu_A(x) = \mu_B(x), \quad \forall x \in U \tag{3.8}$$

การเป็น Subset: ถ้า A เป็น Subset ของ B เราแทนด้วย $A \subseteq B$ ก็ต่อเมื่อว่า

$$\mu_A(x) \leq \mu_B(x), \quad \forall x \in U \tag{3.9}$$

แต่ถ้า $A \subseteq B$ และ $A \neq B$ เราแทนด้วย $A \subset B$ เรียกว่า A เป็น proper subset ของ B

3.4 การแสดงค่าความเป็นสมาชิก

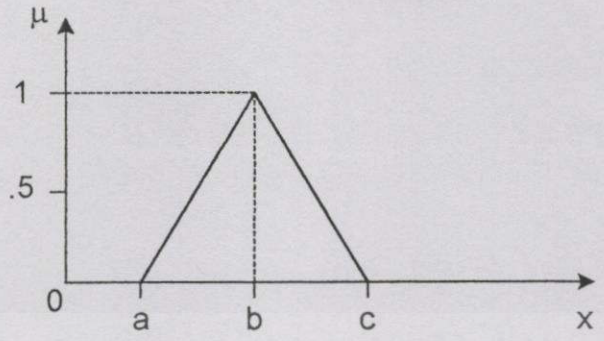
การแสดงค่าความเป็นสมาชิกสามารถทำได้หลายวิธีซึ่งเราจะได้กล่าวเพียง 4 วิธีคือ

3.4.1 รูปสามเหลี่ยม

การกำหนดฟังก์ชั่นความเป็นสมาชิกโดยใช้รูปสามเหลี่ยม นิยมใช้กันมากในระบบควบคุมที่ใช้ฟัซซี่เซต เทอมเซตที่กำหนดโดยฟังก์ชั่นนี้จะต้องมีค่าที่เหมาะสมที่สุดอยู่เพียงค่าเดียวที่ทำให้ค่าความเป็นสมาชิกลดลงเรื่อยๆเมื่อยิ่งห่างจากค่านี้มากขึ้น ดังแสดงในรูปที่ 3.2 ถ้ากำหนดให้ $a \leq b \leq c$ เมื่อ a, b และ c เป็นจำนวนจริงใดๆ สมการของฟังก์ชั่นความเป็นสมาชิกรูปสามเหลี่ยมสามารถกำหนดได้ดังนี้

$$\mu_A(x_i) = \begin{cases} \dfrac{x_i - a}{b - a}, & \text{for } a \leq x_i \leq b,\ a \neq b \\ \dfrac{c - x_i}{c - b}, & \text{for } b \leq x_i \leq c,\ b \neq c \\ 0, & \text{otherwise} \end{cases} \tag{3.10}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

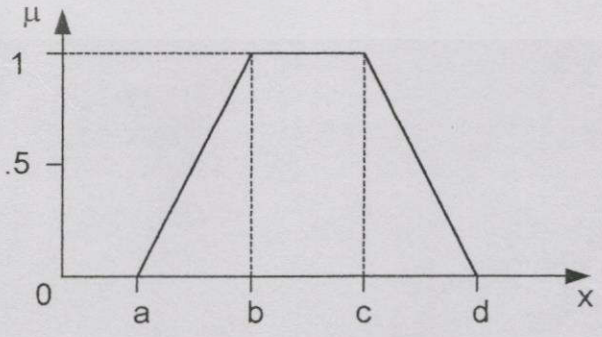


รูปที่ 3.2 แสดงกราฟของฟังก์ชั่นความเป็นสมาชิกรูปสามเหลี่ยม

3.4.2 รูปสี่เหลี่ยมคางหมู

สำหรับฟังก์ชั่นความเป็นสมาชิกที่กำหนดด้วยรูปสี่เหลี่ยมคางหมู นิยมใช้ในระบบควบคุมที่ใช้ฟัซซี่เซตเช่นกัน แต่จะพบน้อยกว่าฟังก์ชั่นที่กำหนดด้วยรูปสามเหลี่ยม ฟังก์ชั่นความเป็นสมาชิกที่กำหนดด้วยรูปสี่เหลี่ยมคางหมูจะมีช่วงค่าความเหมาะสมมากที่สุดอยู่กลุ่มหนึ่งที่ทำให้ค่าความเป็นสมาชิกของเทอมเซตนั้นๆมีค่าเท่ากับ 1 นอกนั้นจะมีค่าความเป็นสมาชิกน้อยลงเรื่อย ๆ เมื่อยิ่งห่างจากข้อมูลกลุ่มนี้ ถ้ากำหนดให้ $a \le b \le c \le d$ เมื่อ a, b, c และ d เป็นเลขจำนวนจริงใดๆ แสดงในรูปที่ 3.3 สมการของฟังก์ชั่นความเป็นสมาชิกของรูปสี่เหลี่ยมคางหมูสามารถกำหนดได้ดังนี้

$$\mu_A(x_i) = \begin{cases} \dfrac{x_i - a}{b - a}, & \text{for } a \le x_i \le b,\ a \ne b \\ 1, & \text{for } b \le x_i \le c \\ \dfrac{d - x_i}{d - c}, & \text{for } c \le x_i \le d,\ c \ne d \\ 0, & \text{otherwise} \end{cases} \qquad (3.11)$$



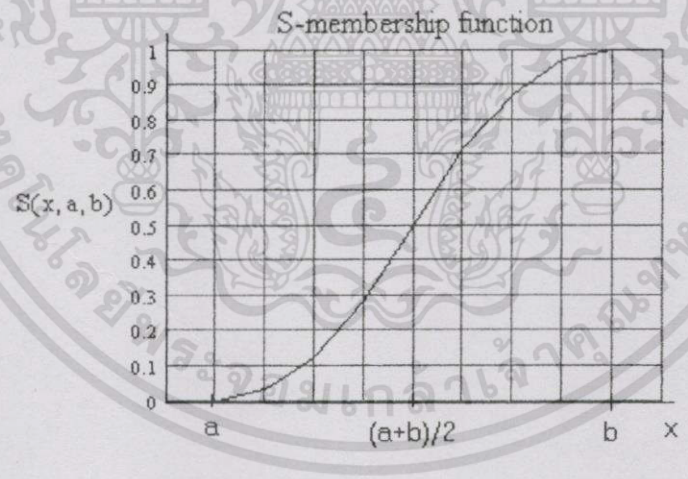
รูปที่ 3.3 แสดงกราฟของฟังก์ชั่นความเป็นสมาชิกรูปสี่เหลี่ยมคางหมู

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4.3 รูปเส้นโค้งตัว S

ฟังก์ชั่นความเป็นสมาชิกที่กำหนดด้วยเส้นโค้งรูปตัว S (S-Curve, sigmoid or logistic) เป็นการกำหนดความสัมพันธ์ระหว่างโดเมนไปยังเรนจ์แบบไม่เชิงเส้น ตัวแปรที่เหมาะสมที่จะใช้ฟังก์ชั่นนี้คือ พวกอายุการใช้งานของอุปกรณ์ต่างๆ ตัวแปรส่วนใหญ่ที่ใช้ในทางการประมวลผลภาพเป็นต้น ซึ่งตัวแปรพวกนี้มักจะมีความสัมพันธ์แบบไม่เป็นเชิงเส้นที่เป็นรูปตัว S หรือ S^{-1} ถ้ากำหนดให้ $a \le b$ เมื่อ a, b เป็นเลขจำนวนจริงใดๆ(ดูรูปที่ 3.4) สมการของฟังก์ชั่นความเป็นสมาชิกของเส้นโค้งที่เป็นรูปตัว S สามารถกำหนดได้ดังนี้

$$\mu_A(x_i) = S(x_i; a, b) = \begin{cases} 0, & \text{for } x_i < a \\ 2\left[\dfrac{x_i - a}{b - a}\right]^2, & \text{for } a \le x_i < \dfrac{a+b}{2} \\ 1 - 2\left[\dfrac{x_i - b}{b - a}\right]^2, & \text{for } \dfrac{a+b}{2} \le x_i < b \\ 1, & \text{for } x_i \ge b \end{cases} \tag{3.12}$$



รูปที่ 3.4 แสดงกราฟฟังก์ชั่นรูปตัว S(x ; a, b)

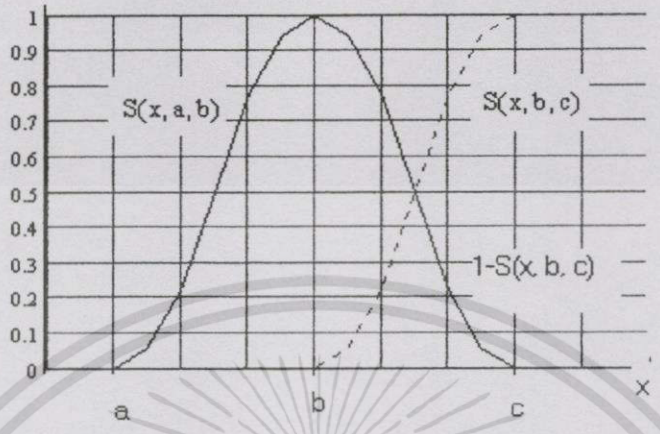
3.4.4 รูประฆังคว่ำ π-function

เทอมเซตที่ใช้ฟังก์ชั่นความเป็นสมาชิกแบบรูประฆังคว่ำ จะมีคุณสมบัติเป็นข้อมูลคล้ายกับสมการรูปสามเหลี่ยม แต่ข้อมูลของโดเมนที่แปลงไปยังเรนจ์จะมีความสัมพันธ์ในลักษณะไม่เป็นเชิงเส้น สมการของรูประฆังคว่ำมีด้วยกันหลายสมการ แต่ที่ใช้กันมากคือฟังก์ชั่น π ฟังก์ชั่น Gaussian และ ฟังก์ชั่น Beta ฟังก์ชั่น π คำนวณได้ดังนี้ (ดูรูปที่ 3.5)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

$$\mu_A(x) = \pi(x;a,b,c) = \begin{cases} S(x;a,b), & \text{for } x < b \\ 1 - S(x;b,c), & \text{for } x > b \end{cases} \qquad (3.13)$$



รูปที่ 3.5 แสดงกราฟฟังก์ชั่นรูประฆังคว่ำ $\pi(x; a, b, c)$

ตัวอย่าง 3.2

การแสดงค่าความเป็นสมาชิก สมมุติให้ฟัซซี่เซต F ของ Y ซึ่งแทนเซตวัยกลางคน เราสามารถเขียนสมการได้

$$F = \{y, \mu_F(y)) \mid y \in Y\} \qquad (3.14)$$

y เป็นตัวแปรแทนอายุของคนในวัยต่างๆ ฉะนั้นจะกำหนดเทอมเซตของวัยกลางคนอย่างไร โดยทั่วไปเราจะกำหนดให้ผู้ที่มีอายุ 45 ปี อยู่ในวัยกลางคน แต่ผู้ที่มีอายุ 38 ปี จะจัดไว้ในวัยกลางคนหรือไม่ ดังนั้นจะเห็นได้ว่า ถ้ากำหนดให้ค่าความเป็นสมาชิกสูงสุดเป็น 1 ที่อายุ 45 ปี และ มีค่าเป็น 0 ที่มีอายุน้อยกว่า 30 ปี และ มากกว่า 60 ปี เราจะแทนด้วยฟังก์ชั่นสามเหลี่ยมได้ดังต่อไปนี้

$$\mu_F(y) = \begin{cases} 0, & \text{for } 0 \le y \le 30 \\ \dfrac{y - 30}{15}, & \text{for } 30 \le y \le 45 \\ \dfrac{60 - y}{15}, & \text{for } 45 \le y \le 60 \\ 0, & \text{for } y \ge 60 \end{cases} \qquad (3.15)$$

ในรูปที่ 3.6 แสดงกราฟของฟังก์ชั่นค่าความเป็นสมาชิกของกลุ่มอายุ ซึ่งเรากำหนดให้

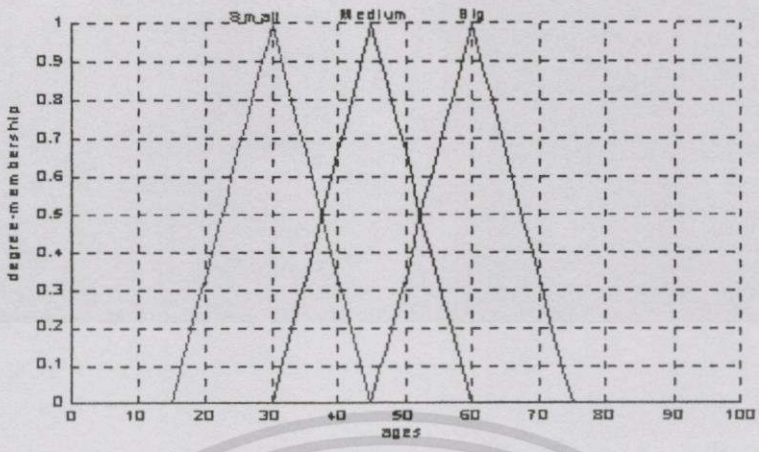
Small หมายถึงกลุ่มอายุวัยหนุ่ม

Medium หมายถึงกลุ่มอายุวัยกลาง

Big หมายถึงกลุ่มอายุวัยชรา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

38046



รูปที่ 3.6 แสดงกราฟของฟังก์ชั่นค่าความเป็นสมาชิกของกลุ่มอายุ

3.5 การวิเคราะห์ฟัซซี่ลอจิกในระบบควบคุม

ฟัซซี่ลอจิกในระบบควบคุมที่แสดงในรูปที่ 3.7 นั้นได้ประกอยด้วย 4 ส่วนคือ Fuzzifier, Fuzzy Rule Base, Inference Engine และ Defuzzifier

อินพุท x เอาท์พุท y เป็นตัวแปรของฟัซซี่เซต ตัวแปรของระบบจะมีกี่ตัวก็ได้ซึ่งจะขึ้นอยู่กับระบบ สำหรับตัวแปรเอาท์พุทจะเป็นตัวแปรที่ถูกประมาณค่าโดยแบบจำลองของฟัซซี่เพื่อนำไปออกแบบกฎฟัซซี่ใช้ในระบบควบคุมต่อไป ถ้าค่าของตัวแปรเอาท์พุท y ที่ได้จากส่วน Defuzzifier นั่นไม่ใช่การควบคุมสำหรับเครื่องจักรโรงงาน ระบบนี้จะเป็นระบบฟัซซี่ลอจิกเพื่อการตัดสินใจ ตัวแปรอินพุท X และ ตัวแปรเอาท์พุท Y สามารถกำหนดเป็นเว็คเตอร์ได้ดังนี้

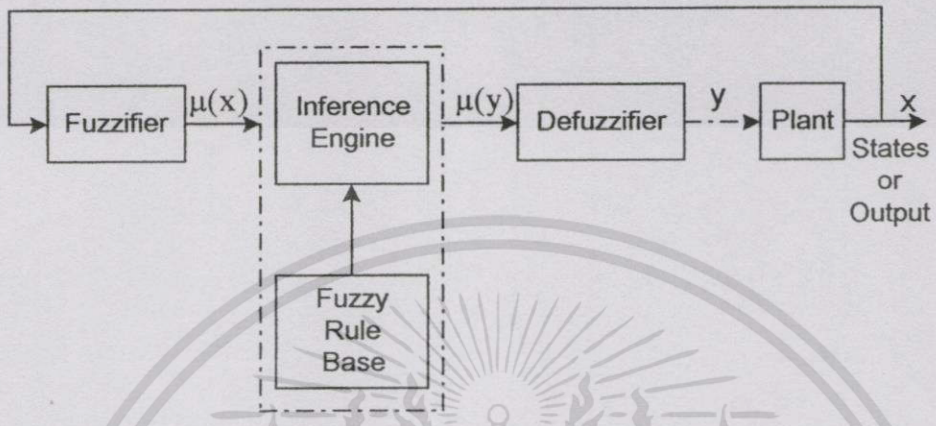
$$X = \left\{\left(x_i, U_i, \{T^1_{x_i}, T^2_{x_i}, \ldots, T^{k_i}_{x_i}\}, \{\mu^1_{x_i}, \mu^2_{x_i}, \ldots, \mu^{k_i}_{x_i}\}\right) \middle| \, i = 1,2,\ldots,n\right\} \quad (3.16)$$

$$Y = \left\{\left(y_i, V_i, \{T^1_{y_i}, T^2_{y_i}, \ldots, T^{l_i}_{y_i}\}, \{\mu^1_{y_i}, \mu^2_{y_i}, \ldots, \mu^{l_i}_{y_i}\}\right) \middle| \, i = 1,2,\ldots,m\right\} \quad (3.17)$$

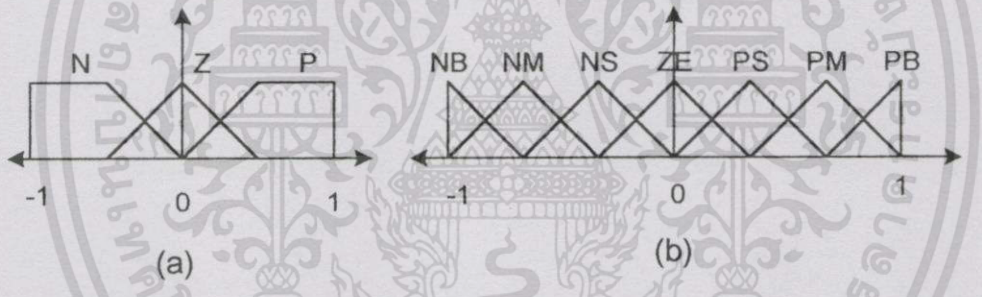
โดย x_i ได้จากเอกภพอินพุทฟัซซี่ $U = U_1 \times U_2 \times \ldots \times U_n$ และ y_i ได้จากเอกภพเอาท์พุทฟัซซี่ $V = V_1 \times V_2 \times \ldots \times V_m$ เราสังเกตได้ว่าตัวแปรอินพุท x_i ได้กำหนดคุณลักษณะด้วย $T(x_i) = \{T^1_{x_i}, T^2_{x}, \ldots, T^{k_i}_{x_i}\}$ และ $\mu(x_i) = \{\mu^1_{x_i}, \mu^2_{x_i}, \ldots, \mu^{k_i}_{x_i}\}$ ซึ่ง $T(x_i)$ เป็น เทอมเซต (term set) ของ x_i ขนาดของเทอมเซต $|T(x_i)| = k_i$ เรียกว่า Fuzzy partition ของ x_i การกำหนด Fuzzy partition ไม่จำกัดแน่นอนว่าจะมีจำนวนเทอมเซตเท่าไร มันขึ้นกับว่าเราต้องการความระเอียดมากน้อยเท่าไร เช่นในรูปที่ 3.8.(a) แสดงให้เห็นว่าในเอกภพ [-1,+1] Fuzzy partition ถูกแบ่งออกเป็น 3 เทอมเซตคือ N(negative), Z(Zero), และ P(Positive) ในรูปที่3.8 (b) Fuzzy partition ได้ถูกแบ่งออกเป็น 7 เทอมเซต NB(Negative Big), NM(Negative Medium), NS(Negative Small), ZE (Zero), PS (Positive Small), PM(Positive Medium), PB(Positive Big)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การกำหนดช่วงของแต่ละเทอมเซตสามารถกำหนดขึ้นมาเองได้ โดยคำนึงถึงค่าอินพุทที่เข้ามาในระบบว่ามีค่าเท่าไร



รูปที่ 3.7 Basic architecture of a fuzzy logic controller (FLC)



รูปที่ 3.8 diagrammatic representation of fuzzy partition

Fuzzifier เป็นขั้นตอนการแปลงค่าตัวแปรข้อมูลเข้า (input variables) ให้อยู่ในรูปค่าความเป็นสมาชิกในระบบฟัซซี่เซต โดยใช้ฟังก์ชั่นความเป็นสมาชิกที่ได้เสนอไปแล้ว แต่การเลือกใช้รูปแบบของฟังก์ชั่นนั้นต้องเลือกให้เหมาะสมกับคุณสมบัติของตัวแปร เช่นตัวแปรมีคุณสมบัติเป็นเชิงเส้นจะต้องเลือกใช้ฟังก์ชั่นความเป็นสมาชิกที่เป็นเชิงเส้นเช่นสมการรูปแบบสามเหลี่ยม สมการรูปแบบสี่เหลี่ยมคางหมู ถ้าตัวแปรมีคุณสมบัติไม่เป็นเชิงเส้นจะต้องเลือกใช้ฟังก์ชั่นความเป็นสมาชิกไม่เป็นเชิงเส้นด้วย เช่นฟังก์ชั่นรูปตัว S, π, Beta หรือ Guassian

Fuzzy Rule Base กฎการควบคุมแบบฟัซซี่ได้ถูกกำหนดอยู่ในรูปแบบของ IF-THEN Rules ซึ่งแสดงความสัมพันธ์ที่ง่ายดายระหว่างค่าอินพุทและเอาท์พุทอยู่ในระบบ รูปแบบโดยทั่วไปของกฎควบคุมฟัซซี่ ในกรณีของ multi-input-single-output system (MISO) คือ

$$R^i : \text{IF } x \text{ is } A_i \ldots., \text{ AND } y \text{ is } B_i, \text{ THEN } z = C_i, \quad i = 1, 2, 3, \ldots n \qquad (3.18)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยที่ x, ..., y,และ z เป็นตัวแปรฟัซซี่ $A_i, ..., B_i$ และ C_i เป็นเทอมเซตที่สอดคล้องกับตัวแปรฟัซซี่ x,..., y และ z ที่อยู่ในเอกภพ U,..., V, และ W ตามลำดับ ประพจน์ฟัซซี่ x is A_i...., AND y is B_i ที่ตามหลัง IF นั้นเป็นส่วนของเงื่อนไขที่เรียกว่า "Precondition" หรือส่วนของอินพุทฟัซซี่ ประพจน์ฟัซซี่ $z = C_i$ ที่ตามหลัง THEN นั้นเป็นส่วนผลเนื่องจากเงื่อนไขที่เรียกว่า "Consequent" หรือเป็นส่วนของเอาท์พุทฟัซซี่ นอกนี้แล้วกฎฟัซซี่อาจะเขียนได้อีกคือ

$$R^i : \text{IF } x \text{ is } A_i \text{ ....., AND } y \text{ is } B_i \text{ , THEN } z = f(x,...,y) \qquad (3.19)$$

โดยที่ f(x,..., y) เป็นฟังก์ชั่นของตัวแปร x,..., y

การสร้างกฎที่ใช้ควบคุมระบบจะต้องใช้ทั้งความรู้ และประสบการณ์ สำหรับจำนวนกฎในระบบของฟัซซี่ลอจิกควบคุมหนึ่งๆ จะขึ้นอยู่กับจำนวนตัวแปรอินพุท และจำนวนเทอมเซตของตัวแปร เช่น ถ้าอินพุทมี 2 ตัวแปร และแต่ละตัวแปรมี 5 เทอมเซตจะได้จำนวนกฎทั้งหมดเท่ากับ 25 กฎ

Fuzzy Inference Engine เป็นการนำค่าความเป็นสมาชิก (μ) ที่ได้รับเข้ามาไปประมวลผลตามกฎ (Rules Base) ที่ระบบได้เรียนรู้หรือออกแบบไว้เพื่อหาคำตอบ

Defuzzifier เป็นการแปลงผลจากค่าความเป็นสมาชิกที่ได้จากขั้นตอนของ Fuzzy Inference Engine ให้อยู่ในรูปของค่า output หรือเป็นกระบวนการประมาณค่าของเอาท์พุท โดยการอนุมานจากค่าความจริงในส่วนเงื่อนไขของระบบจำลองฟัซซี่ลอจิก ค่าที่ได้จะเป็นเลขจำนวนจริงที่อยู่ในโดเมนของตัวแปรเอาท์พุท ซึ่งค่านี้เป็นค่าคาดหวัง (Expected Value) ของตัวแปรฟัซซี่ เพื่อนำไปใช้งานต่อไป เทคนิคการ Defuzzifier ก็มีหลายวิธีด้วยกัน แต่ที่ใช้กันมากคือวิธีการใช้ค่าศูนย์กลาง(Center Of Area (COA)) และวิธีการใช้ค่าสูงสุดหรือค่าต่ำสุด (Mean Of Maximum (MOM))

วิธีการใช้ค่าศูนย์กลางเพื่อหาค่าคาดหวังเป็นการแสดงค่าเอาท์พุทที่สัมพันธ์กับค่าจุดศูนย์กลางมวล (Centroid) ของเอาท์พุทในระดับที่ทำงาน เพราะว่าค่าขอบของฟังก์ชั่นระดับการเป็นสมาชิกจะไม่นำมาใช้และจะไม่เกิดความชัดเจน ค่าศูนย์กลางจะสัมพันธ์กับค่าของเอาท์พุทหนึ่งค่า ผลที่ได้จะเป็นการกระทำที่หยาบๆภายในช่วงของเอาท์พุท กฎทั่วๆไปจะต้องมีการปฏิบัติเพียงจุดเดียวของแต่ละระบบ ในกรณีที่ต้องทำพ้อมๆกันหลายกฎ ในการกระทำหนึ่งครั้งจะเกิดการซ้อนทับกัน (overlap) ของฟังก์ชั่นอินพุทของระดับความเป็นสมาชิก วิธีนี้จะรวมเอาค่าเอาท์พุทของการกระทำหลายๆค่าเป็นค่าเดียวสำหรับใช้ในระบบ ค่าเอาท์พุทค่าเดียวนี้จะเป็นค่าน้ำหนักเฉลี่ยของศูนย์กลางของแต่ละฟังก์ชั่นระดับความเป็นสมาชิก

วิธีการใช้ค่าสูงสุดหรือค่าต่ำสุด เป็นการใช้ค่าสูงสุดหรือค่าต่ำสุดของระดับการเป็นสมาชิกจากการทำหลายๆครั้ง แล้วเลือกการกระทำเพียงหนึ่งรูปแบบ ถ้าหากเกิดการกระทำที่มีค่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ความเป็นสมาชิก μ สูงสุดเท่ากัน 2 ค่า จะต้องใช้รูปแบบการแก้ปัญหาอีกลักษณะหนึ่งคือใช้ค่าเฉลี่ยของค่าเอาท์พุท หรือเลือกการกระทำที่สัมพันธ์กับค่าระดับของระบบพื้นฐาน

3.6 การประยุกต์ใช้ฟัซซี่เซตสำหรับการรู้จำตัวอักษร

จากบทที่ 2 เราได้คุณลักษณะเด่นของตัวอักษร โดยการคำนวณค่าจากตัวอักษรมาเป็นเว็ดเตอร์คือ

$$F_t = \{TOT, UP, DOWN, LEFT, RIGHT, MAXH, MAXV, DW_UP, RG_LF\}^T$$

จากนั้นเราจะนำเอาค่าดังกล่าวนี้ไปปรับให้อยู่ในรูปแบบของฟัซซี่เซต โดยเรากำหนดฟังก์ชั่นความเป็นสมาชิกแบบรูปสามเหลี่ยม และให้ $U = \{u_1, u_2, u_3, ..., u_9\}$ เป็นเซตเอกภพซึ่ง $u_1, u_2, ..., u_9$ เป็นเทอร์มเซต สำหรับการแบ่งเซตในระบบฟัซซี่เซต แสดงในรูปที่ 3.9 ฟังก์ชั่นความเป็นสมาชิก เราสามารถเขียนได้ดังนี้

$$\mu_{u_1}(x_i) = \begin{cases} \dfrac{x_i}{75}, \text{ for } 0 \le x_i \le 75 \\ \dfrac{150 - x_i}{75}, \text{ for } 75 \le x_i \le 150 \\ 0 \text{ otherwise} \end{cases} \quad (3.20)$$

$$\mu_{u_2}(x_i) = \begin{cases} \dfrac{x_i - 75}{75}, \text{ for } 75 \le x_i \le 150 \\ \dfrac{225 - x_i}{75}, \text{ for } 150 \le x_i \le 225 \\ 0 \text{ otherwise} \end{cases} \quad (3.21)$$

$$\mu_{u_3}(x_i) = \begin{cases} \dfrac{x_i - 150}{75}, \text{ for } 150 \le x_i \le 225 \\ \dfrac{300 - x_i}{75}, \text{ for } 225 \le x_i \le 300 \\ 0 \text{ otherwise} \end{cases} \quad (3.22)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$\mu_{u_4}(x_i) = \begin{cases} \dfrac{x_i - 225}{75}, \text{ for } 225 \le x_i \le 300 \\ \dfrac{375 - x_i}{75}, \text{ for } 300 \le x_i \le 375 \\ 0 \text{ otherwise} \end{cases} \quad (3.23)$$

$$\mu_{u_5}(x_i) = \begin{cases} \dfrac{x_i - 300}{75}, \text{ for } 300 \le x_i \le 375 \\ \dfrac{450 - x_i}{75}, \text{ for } 375 \le x_i \le 450 \\ 0 \text{ otherwise} \end{cases} \quad (3.24)$$

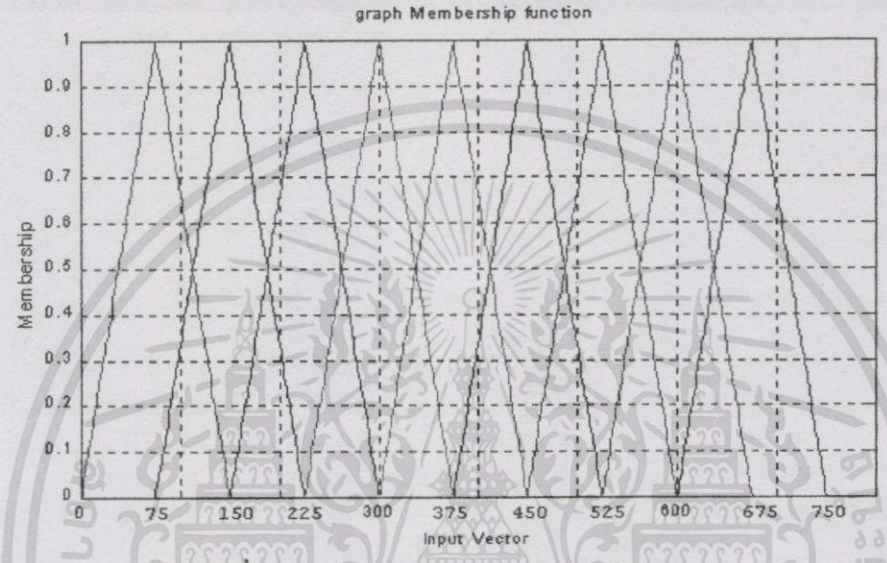
$$\mu_{u_6}(x_i) = \begin{cases} \dfrac{x_i - 375}{75}, \text{ for } 375 \le x_i \le 450 \\ \dfrac{525 - x_i}{75}, \text{ for } 450 \le x_i \le 525 \\ 0 \text{ otherwise} \end{cases} \quad (3.25)$$

$$\mu_{u_7}(x_i) = \begin{cases} \dfrac{x_i - 450}{75}, \text{ for } 450 \le x_i \le 525 \\ \dfrac{600 - x_i}{75}, \text{ for } 525 \le x_i \le 600 \\ 0 \text{ otherwise} \end{cases} \quad (3.26)$$

$$\mu_{u_8}(x_i) = \begin{cases} \dfrac{x_i - 525}{75}, \text{ for } 525 \le x_i \le 600 \\ \dfrac{675 - x_i}{75}, \text{ for } 600 \le x_i \le 675 \\ 0 \text{ otherwise} \end{cases} \quad (3.27)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$\mu_{u_9}(x_i) = \begin{cases} \dfrac{x_i - 600}{75}, \text{ for } 600 \le x_i \le 675 \\ \dfrac{750 - x_i}{75}, \text{ for } 675 \le x_i \le 750 \\ 0 \text{ otherwise} \end{cases} \quad (3.28)$$



รูปที่ 3.9 แสดงกราฟ ของฟังก์ชั่นความเป็นสมาชิกในแต่ละเทอร์มเซต

จากการหาค่าสมาชิกในแต่ละเทอร์มเซตเราสามารถกำหนดค่าความเป็นสมาชิกของเว็กเตอร์ F_t ได้คือ

$$\mu_{F_t}(x_i) = \text{Max}(\mu_{u_1}(x_i), \mu_{u_2}(x_i), ..., \mu_{u_8}(x_i)) \ , \forall x_i \in U \quad (3.29)$$

ตัวอย่างค่าความเป็นสมาชิกของตัวอักษรที่มีรูปแบบ "Alcie0" และมีขนาดเท่า 14 ที่หาได้แสดงในตาราง ที่ 3.1 ข้างล่างนี้

ตารางที่ 3.1 แสดงค่าความเป็นสมาชิกของตัวอักษรพิมพ์ภาษาลาวในระบบฟัซซี่เซต

ตัวอักษร	ก	ข	ค	ง	จ	ส	ช
เลขรหัส	1	2	3	4	5	6	7
TOT	0.6667	0.6800	0.5467	0.5600	0.5333	0.8000	0.5067
UP	0.6800	0.9333	0.8000	0.6800	0.7067	0.5067	0.6400
DOWN	0.9867	0.6133	0.7467	0.8800	0.8267	0.6933	0.8667
LEFT	0.6933	0.5733	0.7733	0.8267	0.9867	0.8400	0.5067

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.1(ต่อ)

RIGHT	0.9733	0.7467	0.7733	0.6133	0.5467	0.9600	0.9867
MaxH	0.4533	0.3600	0.3200	0.4400	0.3600	0.3733	0.3333
MaxV	0.7333	0.2533	0.6800	0.5867	0.4000	0.6267	0.3733
DW_UP	0.0108	0.0149	0.0131	0.0104	0.0066	0.0109	0.0096
RG_LF	0.0107	0.0109	0.0133	0.0142	0.0238	0.0129	0.0152
ตัวอักษร	ย	ม	ม	ต	ถ	บ	ป
เลขรหัส	8	9	10	11	12	13	14
TOT	0.8133	0.7467	0.7600	0.9200	0.5333	0.6533	0.8933
UP	0.8267	0.7467	0.7867	0.7733	0.6933	0.9600	0.8533
DOWN	0.6400	0.5067	0.5467	0.6933	0.8400	0.6933	0.7467
LEFT	0.9733	0.7467	0.8667	0.8133	0.5867	0.6933	0.8133
RIGHT	0.8400	1.0000	0.8933	0.7333	0.8800	0.9600	0.9200
MaxH	0.3600	0.4267	0.4400	0.4133	0.4400	0.4000	0.3600
MaxV	0.6000	0.5600	0.8133	0.7333	0.5600	0.8400	0.5733
DW_UP	0.0117	0.0170	0.0165	0.0117	0.0127	0.0145	0.0233
RG_LF	0.0105	0.0123	0.0132	0.0114	0.0107	0.0123	0.0146
ตัวอักษร	ฝ	ผ	ฝ	ฝ	ฟ	ฟ	ภ
เลขรหัส	15	16	17	18	19	20	21
TOT	0.9600	0.7200	0.6400	0.6400	0.6800	0.9733	0.8667
UP	0.5333	0.7067	0.6800	0.7867	0.8533	0.8133	0.8533
DOWN	0.5067	0.9867	0.6800	0.5733	0.8267	0.8400	0.7200
LEFT	0.6267	0.5200	0.7600	0.8133	0.9467	0.6133	0.7733
RIGHT	0.5867	0.8000	0.6000	0.5467	0.7333	0.5867	0.9067
MaxH	0.3600	0.4133	0.4400	0.3600	0.4133	0.3067	0.3733
MaxV	0.7733	0.6000	0.5067	0.6667	0.6400	0.7733	0.6933
DW_UP	0.0237	0.0161	0.0097	0.0267	0.0259	0.0161	0.0120
RG_LF	0.0125	0.0146	0.0174	0.0144	0.0143	0.0126	0.0145

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.1(ต่อ)

ตัวอักษร	ธ	ฌ	ว	ฑ	ฮ	ร	◌ะ
เลขรหัส	22	23	24	25	26	27	28
TOT	0.7067	0.5733	0.7067	0.9067	0.6267	0.6267	0.8800
UP	0.9200	0.9867	0.9600	0.7867	0.6533	0.9867	0.7200
DOWN	0.6267	0.5867	0.7467	0.6933	0.7200	0.6400	0.8400
LEFT	0.7067	0.8800	0.5867	0.9067	0.9333	0.7600	0.7333
RIGHT	0.5867	0.5467	0.8800	0.8133	0.6933	0.6133	0.8533
MaxH	0.4400	0.3067	0.2533	0.6000	0.4800	0.4667	0.6400
MaxV	0.5067	0.3067	0.3600	0.6667	0.4800	0.5333	0.7733
DW_UP	0.0149	0.0160	0.0169	0.0117	0.0171	0.0122	0.0115
RG_LF	0.0102	0.0148	0.0116	0.0124	0.0125	0.0106	0.0227
ตัวอักษร	◌ๆ	◌ั	◌ิ	◌ี	◌ึ	◌ื	◌ุ
เลขรหัส	29	30	31	32	33	34	35
TOT	0.6533	0.6000	0.6933	0.7600	0.9733	0.6400	0.8533
UP	0.6800	0.8533	0.9867	1.0000	0.6800	0.6933	0.9467
DOWN	0.6667	0.5467	0.7067	0.7600	0.7067	0.9467	0.8000
LEFT	0.5600	0.7733	0.9067	0.8267	0.5600	0.9067	0.7067
RIGHT	0.7867	0.6267	0.7867	0.9333	0.5867	0.5467	0.5600
MaxH	0.4533	0.3067	0.6267	0.5467	0.4133	0.4267	0.5600
MaxV	0.3733	0.5200	0.5600	0.5467	0.4000	0.6800	0.4933
DW_UP	0.0041	0.0146	0.0146	0.0125	0.0149	0.0119	0.0108
RG_LF	0.0176	0.0139	0.0139	0.0137	0.0139	0.0172	0.0219
ตัวอักษร	◌ู	เ◌	ไ◌	ใ◌	โ◌	◌็	◌่
เลขรหัส	36	37	38	39	40	41	42
TOT	0.7733	0.6133	0.5333	0.9600	0.7733	0.6533	0.5067
UP	0.8133	0.6667	0.7600	0.9733	0.8533	0.9067	0.7067
DOWN	0.9600	0.9467	0.7067	0.9333	0.6267	0.5600	0.7867
LEFT	0.9600	0.9067	0.9867	0.5733	0.8800	0.8533	0.6400

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.1(ต่อ)

RIGHT	0.8133	0.5200	0.5200	0.6133	0.8933	0.8000	0.8533
MaxH	0.4133	0.4267	0.2667	0.8800	0.6000	0.4533	0.6933
MaxV	0.4400	0.6400	0.2533	0.9733	0.6000	0.5733	0.4533
DW_UP	0.0140	0.0147	0.0061	0.0275	0.0065	0.0110	0.0137
RG_LF	0.0127	0.0109	0.0234	0.0049	0.0321	0.0131	0.0144
ตัวอักษร	◌ั	◌ั	◌ั	◌ั			
เลขรหัส	43	44	45	46			
TOT	0.9467	0.7333	0.6267	0.8667			
UP	0.5067	0.8667	0.9467	0.7333			
DOWN	0.5467	0.8667	0.5733	0.8667			
LEFT	0.7067	0.7067	0.6133	0.8667			
RIGHT	0.6533	0.5600	0.7600	1.0000			
MaxH	0.8267	0.7333	0.3333	0.3733			
MaxV	0.8267	0.3867	0.4267	0.3333			
DW_UP	0.0135	0.0196	0.0156	0.0110			
RG_LF	0.0116	0.0126	0.0128	0.0125			

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

ทฤษฎีนิวรอลเน็ตเวอร์ค(Neural Network)

โครงข่ายประสาทเทียมคือการจำลองการทำงานของระบบประสาท ของมนุษย์ได้เริ่มคิดค้นขึ้นมานับแต่ปี 1940 เช่นในปี 1943 วอร์เรน แมคคูลลอ็คช์ (Worren McCulloch) และ วอลเตอร์ พิทท์ (Walter Pitts) ซึ่งได้ออกแบบโครงข่ายเป็น 2 ชั้นคือชั้นอินพุท(input) และ ชั้นเอาท์พุท (output) โครงข่ายจะมีการเชื่อมโยงจากชั้นอินพุทไปยังชั้นเอาท์พุทโดยผ่านค่าน้ำหนัก(Weight) ซึ่งได้กำหนดไว้ตายตัว การทำงานเบื้องต้นจะใช้ฟังก์ชั่นลอจิกพื้นฐานคือ AND, NOT และ XOR โดยนำเสนอการกำหนดค่าน้ำหนักเพื่อให้โครงข่ายทำงานได้ตามฟังก์ชั่นดังกล่าว

ในปี 1949 โดแนลด์ เฮบบ์(Donald Hebb) ได้ออกแบบกฎการเรียนรู้สำหรับโครงข่ายประสาทเทียมเป็นคนแรก กฎการเรียนรู้นี้ก็คือกระบวนการปรับแต่งค่าน้ำหนักเพื่อทำให้ได้ค่าเอาท์พุท

ในปี 1957 แฟรงค์ เซนบลัทท์(Frank Rosenblatt) ได้นำเสนอรูปแบบที่สำคัญรูปแบบหนึ่งที่เรียกว่า "Perceptron" ซึ่งเป็นสถาปัตยกรรมโครงข่ายประกอบด้วย 3 ชั้นคือ Input (photo) sensory Layer, Association Layer, Output response or Classification layer และใช้กฎการเรียนรู้ของ "Perceptron" ในการปรับค่าน้ำหนักอย่างมีประสิทธิภาพดีกว่ากฎการเรียนรู้ของ เฮบบ์

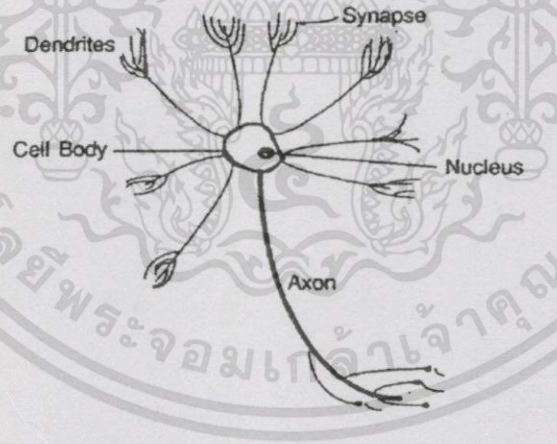
ในช่วงปี 1960 เบอร์นาร์ด วิดโร(Bernard Widrow) ได้พัฒนากฎการเรียนรู้ที่ใกล้เคียงกันกับกฎการเรียนรู้ของ "Perceptron" ซึ่งเรียกว่า "ADALINE" (ADaptive Linear NEuron) และระบบ Network ของ ADALINE ที่เรียกว่า "MADALINE" (Multiple ADALINE) และนำได้พัฒนากฎการเรียนแบบจัดตัวเองที่รู้กันดีในนามของ Least Mean Square (LMS) หรือกฎการเรียนรู้ วิดโร-ฮอฟฟ์ (Widrow-Hoff)

ในปี 1982 จอฮ์น ฮอปฟิลด์(John Hopfield) ได้นำเสนอโครงข่ายนิวรอลเน็ตเวอร์คแบบ Hopfield เวลาเดียวกัน คูนิฮิโกะ ฟูกูชิมา (Kunihiko Fukushima) นักวิจัยชาวญี่ปุ่นได้นำเสนอโครงข่ายนิวรอลเน็ตเวอร์คที่เรียกว่า "Neocognition" ที่สำคัญที่สุดในการพัฒนาผ่านมาอีกรูปแบบหนึ่ง ของระบบโครงข่ายนิวรอลที่นิยมกันมากคือการเรียนรู้ในการปรับแต่งค่าน้ำหนักในระบบหลายชั้นแบบแพร่ย้อนกลับซึ่งได้คิดค้นในเวลาแตกต่างกัน โดย เวอร์โบส(Werbos) ในปี 1974 โดย เดวิด พาร์เคอร์(David Parker) ในปี 1985 และโดย เดวิด รูเมลฮาร์ท(David Rumelhart) ฮินตัน(Hinton) และ วิลเลียมส์(Williams) ในปี 1986 ระบบการเรียนรู้แบบแพร่ย้อนกลับนี้สามารถให้ค่าน้ำหนักที่เหมาะสมเพื่อลดค่าผิดพลาดของเอาท์พุท

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

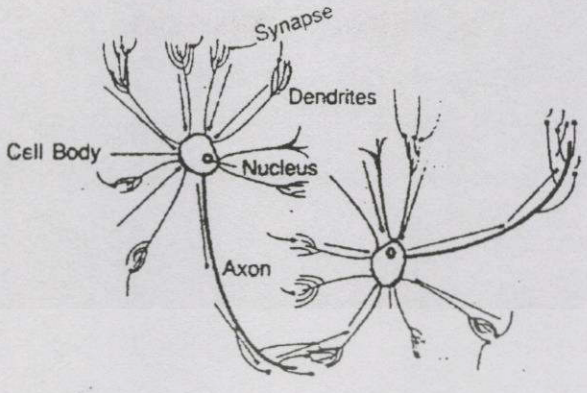
4.1 ลักษณะโครงสร้างของนิวรอล

ระบบประสาทของมนุษย์ประกอบด้วยเซลประสาทประมาณ 10^{11} เซล แต่ละเซลจะประกอบด้วย 3 ส่วนใหญ่ๆคือ เซลประสาทหรือโซมา (Cell body or Soma) เดนไดรท์ (Dendrite) และ แอกซอน(Axon) แสดงในรูปที่ 4.1(a) เซลประสาทหรือโซมาจะประกอบด้วยนิวเคลียส (Nucleus) ส่วนเดนไดรท์ (Dendrite) จะเป็นแขนงต่อออกจากเซลประสาทที่มีหลายจุด และเมื่อออกจากเซลประสาทก็จะแยกแขนงออกไปอีก สำหรับแอกซอนเป็นส่วนที่หนาและยาวต่อออกจากเศลล์ประสาทไปหาเซลประสาทอื่นๆที่ทำงานร่วมกัน เซลประสาทหนึ่งตัวจะมีแอกซอนเพียงอันเดียวเท่านั้น แต่จะมีเดนไดรท์มาก สำหรับระบบการทำงานสัญญาณจะเข้ามาหาเซลประสาททางเดนไดรท์ ด้วยจุดเชื่อมที่เรียกว่า Synapse จะถูกนำมารวมกันโดยจะมีค่า Threshold ค่าหนึ่ง หากผลรวมของสัญญาณนั้นมีค่ามากกว่า Threshold แล้วเซลประสาทก็จะส่งสัญญาณขนาดหนึ่งโดยผ่านทางแอกซอนออกไปหาเซลประสาทตัวอื่นๆ แสดงในรูปที่ 4.1(b) ลักษณะการจัดเลียงชั้นและการเชื่อมโยงระหว่างนิวรอลในสมองของมนุษย์นั้นมีการจัดเลียงที่ซับซ้อน นอกนั้นยังมีการเรียนรู้ตลอดเวลา ฉะนั้นจึงยากที่จะสร้างโมดูลขึ้นมาเพื่อเลียนแบบการทำงานของเซลประสาทในสมองมนุษย์ได้อย่างสมบูรณ์ครบถ้วน



รูปที่ 4.1(a) แสดงส่วนประกอบของเซลประสาท

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.1(b) แสดงจุดถ่ายสัญญาณ(Signal Transmission) จากเซลประสาทหนึ่งหาเซลประสาทตัวอื่นๆ

โครงข่ายนิวรอลเน็ตเวอร์คนั่นได้จำลองการทำงานเพียงบางส่วนของเซลประสาทในสมองมนุษย์โดยที่เซลนิวรอลจะรับข้อมูลอินพุทจากเซลนิวรอลอื่นๆ และแต่ละเซลจะมีจุดเชื่อมโยงระหว่างอินพุท และเอาท์พุท ในรูปที่ 4.2 แสดงลักษณะโครงสร้างของการจำลองโครงข่ายนิวรอลในหนึ่งเซลประสาทซึ่งมี m อินพุทคือ $x_1, x_2, x_3, \ldots, x_m$ เป็นจำนวนจริง และมี 1 เอาท์พุท คือ y_i ส่วนของอินพุทที่เข้ามาจะมีค่าน้ำหนัก(Weight) คือ $w_{i,1}, w_{i,2}, w_{i,3}, \ldots, w_{i,m}$ ซึ่งสามารถปรับค่าได้ ในส่วนที่ทำหน้าที่รวมเอาอินพุทที่เข้ามานั้นเราสามารถเขียนได้ดังนี้

$$net_i = w_{i,1}x_1 + w_{i,2}x_2 + \ldots + w_{i,m}x_m \qquad (4.1)$$

หรืออาจเขียนเป็นรูปของเวคเตอร์จะได้ดังนี้

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

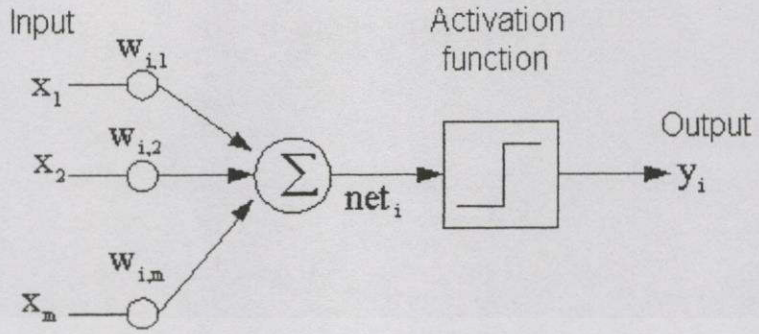
โดยที่ว่า

X = เป็นเวคเตอร์อินพุท

W = เป็นเวคเตอร์ของค่าน้ำหนัก

NET = เป็นเวคเตอร์เอาท์พุท

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.2 โมดูลนิวรอลที่สร้างขึ้นร่วมกับฟังก์ชั่นการกระตุ้น

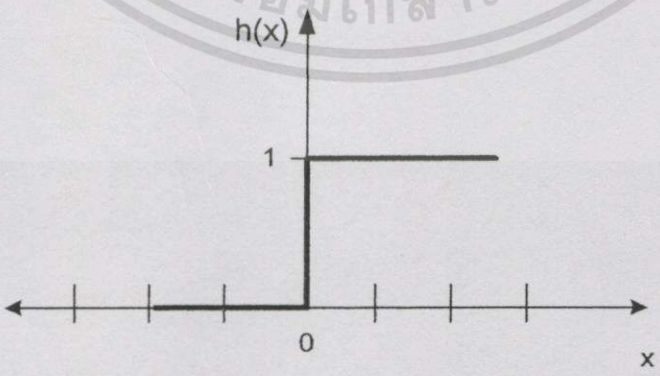
โดยทั่วไปส่วนของเอาท์พุทจะถูกกระทำต่อไปโดยส่วนที่เรียกว่า "ฟังก์ชั่นการกระตุ้น" (Activation function or Threshold function) ฟังก์ชั่นการกระตุ้นทั่วๆไปจะมีหลายรูปแบบซึ่งอาจมีลักษณะเป็นสมการเชิงเส้น หรือ ไม่เป็นเชิงเส้นเช่น Heaviside function (h(x)) แสดงในรูปที่ 4.3 sigmoid function $S_\beta(x)$ แสดงในรูปที่ 4.4 การคำนวณสามารถหาได้สำหรับทุกๆค่า $x \in R$ จากสมการกังนี้

$$h(x) = \begin{cases} 1 \text{ for } x \geq 0 \\ 0 \text{ for } x \leq 0 \end{cases} \qquad (4.2)$$

และ

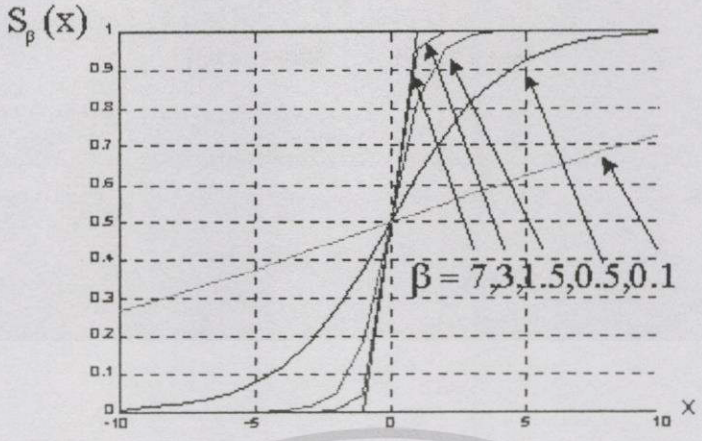
$$S_\beta(x) = \frac{1}{1 + e^{-\beta x}} \qquad (4.3)$$

สำหรับ เป็นค่าคงที่จำนวนบวกเรียกว่า Steepness parameter



รูปที่ 4.3 แสดงกราฟของฟังก์ชั่นการกระตุ้น h(x)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.4 แสดงกราฟของฟังก์ชั่นการกระตุ้น $S_\beta(x)$

ดังนั้น

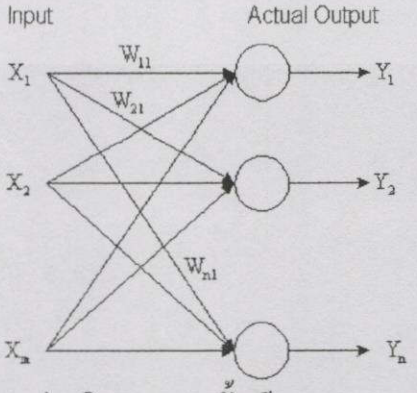
$$y_i = a(net_i) = a\left(\sum_{j=1}^{m} w_{ij} x_j\right) \qquad (4.4)$$

a(.) เป็น Activation function หรือ Threshold function

4.2 โครงข่ายนิวรอลแบบแพร่ย้อนกลับ

4.2.1 โครงข่ายนิวรอลแบบชั้นเดียว (Single Layer Neural Networks)

ในโครงข่ายนิวรอลแบบชั้นเดียวประกอบด้วยอินพุทที่มีการเชื่อมโยงไปยังเอาท์พุทด้วยค่าน้ำหนักและอาจมีการเชื่อมโยงบางส่วนถูกตัดทิ้งออกไป หรืออาจมีการป้อนกลับจากเอาท์พุทไปยังอินพุทได้เราเรียกว่า มีลักษณะการย้อนกลับ(Backward Learning) เพื่อส่งสัญญาณที่ผิดพลาดกลับไปปรับแต่งค่าของน้ำหนักให้ดีขึ้นและลดการผิดพลาดของสัญญาณที่เกิดขึ้นในการเรียนรู้กลุ่มอินพุทชุดหนึ่งๆ โครงข่ายนิวรอลที่รู้จักดีคือโครงข่ายที่เรียกว่า "เพอร์เซปตรอน" (Perceptron) ลักษณะโคสร้างได้แสดงในรูปที่ 4.5

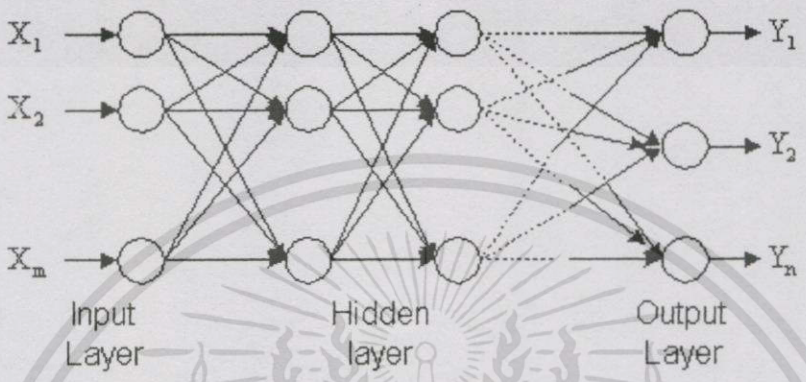


รูปที่ 4.5 แสดงโครงสร้างของโครงข่ายนิวรอลแบบชั้นเดียว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.2 โครงข่ายนิวรอลที่มีลักษณะหลายชั้น (Multilayer Neural Network)

โครงข่ายนิวรอลในลักษณะหลายชั้นเป็นโครงข่ายที่มีโครงข่ายสร้างเป็นชั้นๆที่ซับซ้อน อาจถูกสร้างจากกลุ่มของโครงข่ายนิวรอลแบบชั้นเดียวโดยเอาท์พุทของชั้นหนึ่งจะใช้เป็นอินพุทของชั้นถัดไป แสดงในรูปที่ 4.6 ความสามารถในการทำงานจะดีขึ้นกว่าแบบชั้นเดียว



รูปที่ 4.6 แสดงโครงสร้างของโครงข่ายนิวรอลแบบหลายชั้น

4.2.3 การฝึกสอนของโครงข่ายนิวรอล (Training Algorithm)

การฝึกสอนของโครงข่ายนิวรอลจะเป็นการป้อนชุดอินพุทให้แก่โครงข่ายนิวรอลและพยายามปรับแต่งน้ำหนักเพื่อให้ได้ค่าเอาท์พุทจริงออกมามีความถูกต้องตามต้องการ โดยทั่วไปเราเรียกวิธีการฝึกสอนนี้ว่า "Training Algorithm " การฝึกสอนโครงข่ายนิวรอลนี้ได้แบ่งออกเป็น 3 ประเพทคือ การฝึกสอนแบบชี้นำ การฝึกสอนแบบเสริม และ การฝึกสอนแบบอิสระ (Supervised, Reinforcement and unsupervised learning)

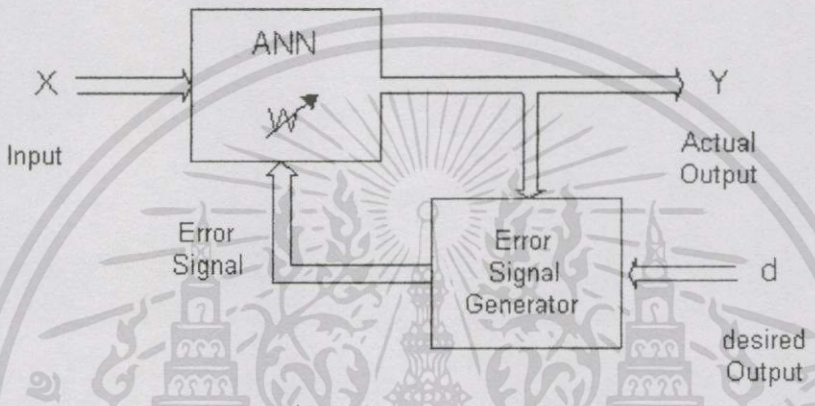
การฝึกสอนแบบชี้นำ (Supervised Learning) เป็นการฝึกสอนแบบจัดคู่กันระหว่างอินพุทและเอาท์พุทเป้าหมายที่ต้องการ เราเรียกว่า "Input-Output pairs" ระบบการฝึกสอนได้แสดงในรูปที่ 4.7 ถ้าให้ $\left(X^{(1)}, d^{(1)}\right), \left(X^{(2)}, d^{(2)}\right), \ldots, \left(X^{(k)}, d^{(k)}\right)$ เป็นจำนวนคู่ที่ทำการฝึกสอนในเมื่อแต่ละอินพุท $X^{(k)}$ ถูกส่งเข้ามาฝึกสอนในโครงข่ายนิวรอล เอาท์พุทจริง $Y^{(k)}$ ที่คำนวณออกมาจะถูกเอาไปเปรียบเทียบกับเอาท์พุทเป้าหมาย $d^{(k)}$ เพื่อหาค่าผิดพลาดที่เกิดขึ้น จากนั้นจึ่งนำเอาค่าที่ผิดพลาดป้อนกลับไปยังโครงข่ายเพื่อปรับค่าน้ำหนักทำให้ค่าเอาท์พุทจริง $Y^{(k)}$ มีแนวโน้มใกล้เคียงกับค่าเอาท์พุทเป้าหมาย $d^{(k)}$

การฝึกสอนแบบเสริม (Reinforcement Learning) ในการฝึกสอนแบบชี้นำนั้นคือแต่ละอินพุทที่เข้ามาเราได้กำหนดเอาท์พุทเป้าหมายเป็นคู่ไว้ก่อนดังได้กล่าวมาแล้วในข้างบน แต่บางครั้งเอาท์พุทจริงที่ออกมาอาจมีค่าผิดพลาดมากเกินไป ในกรณีนี้จะมีเพียงบิดหนึ่งอันเดียวเท่านั้นของข้อมูลที่จะป้อนกลับไปเพื่อชี้บอกว่าเอาท์พุทจริงถูกต้องหรือผิดพลาด การฝึกสอนแบบนี้เรียกว่า

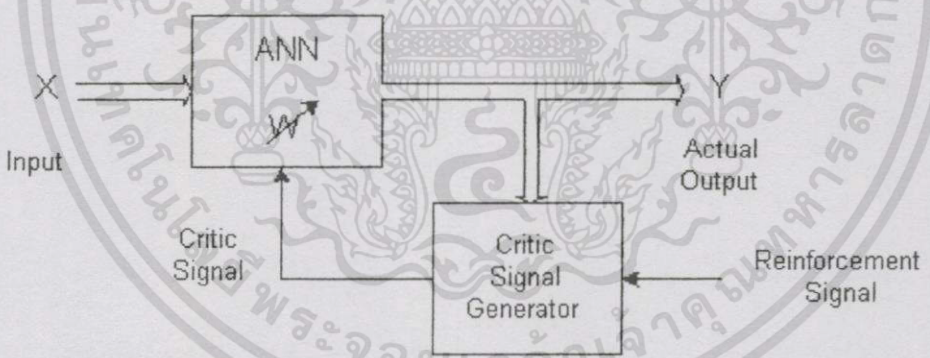
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

"Reinforcement Learning" ระบบการทำงานแสดงในรูปที่ 4.8 ข้อมูลที่ป้อนกลับไปเรียกว่า "Reinforcement Signal"

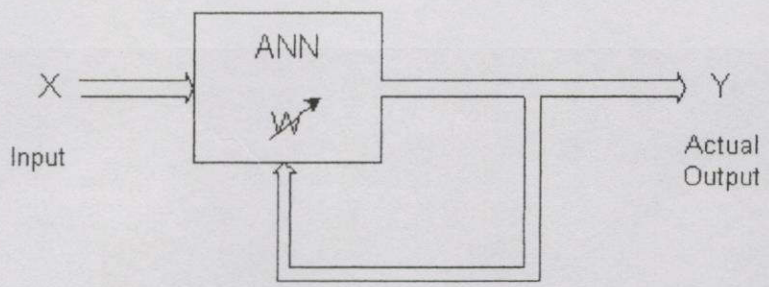
การฝึกสอนแบบอิสระ (Unsupervised Learning) เป็นการฝึกสอนโดยไม่ต้องการเอาท์พุท เป้าหมายเข้ามาช่วยในการตัดสินใจแสดงในรูปที่ 4.9 ชุดการฝึกสอนจะมีเพียงอินพุทเท่านั้น ระบบการฝึกสอนในโครงข่ายจะมีการคำนวณปรับเปลี่ยนค่าของน้ำหนักเพื่อให้ได้เอาท์พุทที่หมั่นคง การคำนวณนี้เรียกว่า "Self-organizing"



รูปที่ 4.7 แสดงระบบการฝึกสอนแบบชี้นำ (Supervised Learning)



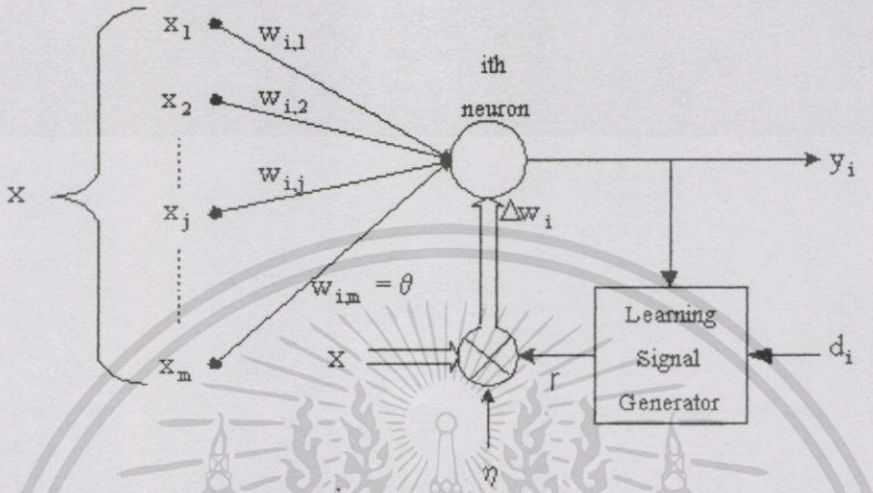
รูปที่ 4.8 แสดงระบบการฝึกสอนแบบเสริม (Reinforcement Learning)



รูปที่ 4.9 แสดงระบบการฝึกสอนแบบอิสระ (Unsupervised Learning)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ถ้าอ้างถึงวิธีการฝึกสอนที่ได้กล่าวมาข้างบนนั้น เราเห็นว่าโครงสร้างโดยทั่วไปของการฝึกสอนสำหรับโครงข่ายนิวรอลหนึ่งๆ แสดงในรูปที่ 4.10 ซึ่งค่าอินพุท $x_j, j = 1,2,...,m$ สามารถเป็นเอาท์พุทจากโครงข่ายนิวรอลชั้นอื่น ควรใส่ใจว่า ตัวแปร θ เป็น Threshold



รูปที่ 4.10 แสดงกฎการฝึกสอนโดยทั่วไปของโครงข่ายนิวรอล

จากรูปที่ 4.10 กฎการฝึกสอนทั่วไป ชี้บอกว่าการปรับเปลี่ยนค่าน้ำหนัก w_i ในโครงข่ายนิวรอลสามารถเขียนสมการได้คือ

$$\Delta w_i(t) = \eta.r.x(t) \qquad (4.8)$$

โดยที่ η เป็นจำนวนคงค่าที่เรียกว่า "Learning constant" ซึ่งกำหนดเรต (rate) ของการฝึกสอน และ r เป็นสัญญาณป้อนกลับโดยทั่วไปเป็นฟังก์ชั่นของตัวแปร w_i, x, d_i คือ

$$r = f_r(w_i, x, d_i) \qquad (4.9)$$

จากสมการ (4.8) และ (4.9) เวคเตอร์ค่าน้ำหนักในการฝึกสอนช่วงเวลา (t+1) จะเป็น

$$w_i(t+1) = w_i(t) + \eta f_r(w_i(t), x(t), d_i(t)).x(t) \qquad (4.10)$$

หรือสามารถเขียนได้อีกแบบคือ

$$w_i^{(t+1)} = w_i^{(t)} + \eta.f_r(w_i^{(t)}, x^{(t)}, d_i^{(t)}).x^{(t)} \qquad (4.11)$$

โดยตัวห้อยข้างบนหมายถึงเวลาใดเวลาหนึ่งของการฝึกสอน สมการ (4.11) ได้เกี่ยวข้องกับการปรับค่าน้ำหนักที่ยังไม่ทราบตามอันดับและสำหรับการปรับค่าน้ำหนักในเวลาต่อเนื่องสามารถแสดงได้คือ

$$\frac{dw_i(t)}{dt} = \eta.r.x(t) \qquad (4.12)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับสมการ (4.11)-(4.12) ก่อนดำเนินการฝึกสอนค่าของน้ำหนักสามารถตั้งค่าเริ่มต้นได้

สำหรับการฝึกสอนแบบชี้นำในโครงข่ายนิวรอลแบบชั้นเดียวที่แสดงลักษณะโครงสร้างของโครงข่ายนิวรอล ดังแสดงในรูปที่ 4.11 เราทำการพิจารณาโดยวิธี Gradient descent algorithm จะใช้ Training Set ของคู่อินพุทและเอาท์พุทที่ถูกต้องให้แก่โครงข่ายโดยใช้ $(X^{(k)}, d^{(k)})$ เป็นคู่อินพุท-เอาท์พุท ใน Training Set ซึ่ง $d^{(k)} = [d_1^{(k)}, d_2^{(k)}, ..., d_n^{(k)}]$ เป็นเอาท์พุทเป้าหมาย ที่ตอบสนองให้กับอินพุท $X^{(k)} = [x_1^{(k)}, x_2^{(k)}, ..., x_m^{(k)}]$ โดยที่ k =1, 2, ..., p , m เป็นจำนวนของอินพุท n เป็นจำนวนของเอาท์พุท และ p เป็นจำนวนของคู่อินพุท-เอาท์พุท ใน Training Set โดยที่เราต้องการ $Y^{(k)} = [Y_1^{(k)}, Y_2^{(k)}, ..., Y_n^{(k)}]$ เป็น เอาท์พุทจริง (Actual output) ให้มีค่าเท่ากับเอาท์พุทเป้าหมาย (Target pattern) $d^{(k)}$ หลังจากการเรียนรู้แล้วจะได้

$$Y_i^{(k)} = a(W_i^T X^{(k)}) = a(\sum_{j=1}^{m} w_{ij} x_j^{(k)}) = d_i^{(k)} \qquad (4.13)$$

ซึ่ง $i = 1,2,...,n; k = 1,2,...,p$ และ $W_i^T = [w_{i,1}, w_{i,2}, ..., w_{i,m}]^T$ เป็นค่าน้ำหนักที่เชื่อมโยงเข้ากับโหนดที่ i และ a(.) เป็น Activation function ของมัน

สำหรับโครงข่ายนิวรอล Perceptron ที่มีฟังค์การกระตุ้นเป็นลีเนียร์(Linear Threshold Unit) ให้เอาท์พุทเป้าหมาย $d_i^{(k)}$ มีค่าเป็น ± 1 เท่านั้น ดังนั้นจากสมการ (4.13) จะสามารถเขียนได้

$$Y_i^{(k)} = \text{sgn}(W_i^T X^{(k)}) = d_i^{(k)}; \; i = 1,2,...,n; k = 1,2,...,p \qquad (4.14)$$

สำหรับโครงข่ายนิวรอลที่เรียกว่า "ADALINE" การสัมพันธ์ระหว่างอินพุทและเอาท์พุทเป็น Linear ในการแก้ปัญหาโดยการฝึกสอนแบบชี้นำเช่นเดียวกันที่แสดงในสมการ (4.13) ถ้าให้ $\{(x^{(1)}, d^{(1)}), (x^{(2)}, d^{(2)}), ..., (x^{(p)}, d^{(p)})\}$ เป็นคู่อินพุท-เอาท์พุท ซึ่งมีจำนวน p คู่ เข้ามาฝึกสอนเป้าหมายคือหาค่าน้ำหนัก w_j ที่ถูกต้องเราเขียนสมการได้ดังนี้

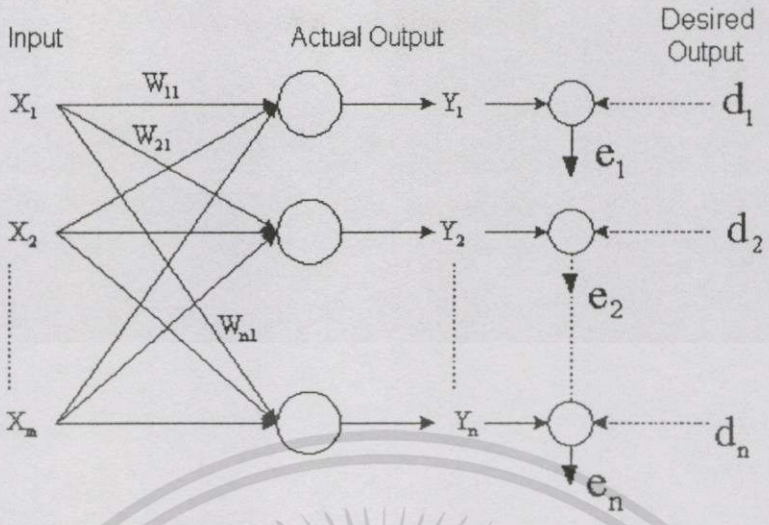
$$\sum_{j=1}^{m} w_j x_j^{(k)} = d^{(k)}, k = 1, 2, ..., p \qquad (4.15)$$

การหาค่าน้ำหนักจากสมการ (4.15) นั่น เราได้กำหนดฟังก์ชั่น E(w) เป็นค่า Mean Square Error ที่ตอบสนองให้กับอินพุท $X_j^{(k)}$ คือ

$$E(w) = \frac{1}{2}\sum_{k=1}^{p}(d^{(k)} - y^{(k)})^2$$

$$= \frac{1}{2}\sum_{k=1}^{p}(d^{(k)} - W^T X^{(k)})^2 = \frac{1}{2}\sum_{k=1}^{p}(d^{(k)} - \sum_{j=1}^{m} w_j x_j^{(k)})^2 \qquad (4.16)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.11 แสดงระบบการฝึกสอนของโครงข่ายนิวรอล "perceptron"

ถ้า $E(w)$ มีค่าน้อยที่สุดก็จะได้ w_j มีค่าดีกว่า ตามปกติ $E(w)$ จะเป็นค่าบวก แต่ถ้า $E(w)$ มีค่าเข้าใกล้กับ 0 จะทำให้ค่า $Y^{(k)}$ สามารถใกล้เคียงกับ $d^{(k)}$ เท่านั้น สำหรับ $k = 1,2,...,p$ ดังนั้นเราทดลองหาค่าน้ำหนักที่ทำให้ $E(w)$ ลดน้อยลงเท่าที่เป็นไปได้ ด้วยการพิจารณาค่าความผิดพลาดของแต่ละคู่อินพุท-เอาท์พุท $(X^{(k)}, d^{(k)})$ ซึ่งจะขึ้นอยู่กับค่าน้ำหนักที่เกี่ยวข้อง จากวิธีการคำนวณแบบ Gradient descent algorithm ตามปกติแล้วมันจะปรับเปลี่ยนแต่ละค่าน้ำหนัก w_j ด้วย Δw_j ซึ่งมีสมการคือ

$$\Delta W = \eta \nabla_w E(W) \qquad (4.17)$$

จะได้ว่า

$$\Delta w_j = \eta \frac{\partial E}{\partial w_j} = \eta \sum_{k=1}^{p} (d^{(k)} - W^T X^{(k)}) x_j^{(k)}; j = 1, 2, ..., m \qquad (4.18)$$

ถ้าการเปลี่ยนแปลงหากเกิดขึ้นโดยแต่ละอินพุทของแพทเทอร์น $X^{(k)}$ ได้รับสัญญาณป้อนกลับ เวลานั้นค่าการปรับเปลี่ยนที่ตอบสนองกับแพทเทอร์น $X^{(k)}$ จะได้

$$\Delta W_j = \eta (d^{(k)} - W^T X^{(k)}) x_j^{(k)} \qquad (4.19)$$

กฎการเรียนรู้ในสมการ (4.19) นี้เรียกว่า "Adaline Learning Rule" หรือ "Widrow-Hoff Learning Rule" และมีสัญญาณการฝึกสอน r ป้อนกลับเมื่อเปรียบเทียบกับกฎทั่วไปของการปรับเปลี่ยนค่าน้ำหนักที่มีสมการ (4.19) จะเขียนได้ $r = d - y = d - w^T x$ เมื่อ η เป็นค่าคงที่บวกเรียกว่าอัตราการเรียนรู้ (Learning rate) ค่า η มีผลต่อความถูกต้องของการประมาณค่าสุดท้ายซึ่งก็คือความเร็วในการเข้าสู่ค่าความผิดพลาดที่เข้าสู่ศูนย์ถ้าค่า η มีน้อยที่สุดการเข้าสู่ศูนย์จะช้าลงและค่าความเชื่อถือจะต่ำลง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับ linear activation function $a(net) = net$ ในกรณีนี้จากสมการ(4.16) จะได้

$$E(w) = \frac{1}{2}\sum_{k=1}^{p}\sum_{i=1}^{n}\left(d_i^{(k)} - y_i^{(k)}\right)^2$$
$$= \frac{1}{2}\sum_{k=1}^{p}\sum_{i=1}^{n}\left[d_i^{(k)} - a(w_i^T x^{(k)})\right]^2 \qquad (4.20)$$
$$= \frac{1}{2}\sum_{k=1}^{p}\sum_{i=1}^{n}\left[d_i^{(k)} - a(\sum_{j=1}^{m} w_{ij} x_j^{(k)}\right]^2$$

ซึ่ง k หมายถึงข้อมูลการฝึกสอน(Training Data) อันดับที่ k และ p หมายถึงจำนวนรวมทั้งหมดของข้อมูลการฝึกสอน โดยใช้วิธี gradient descent algorithm เราจะได้

$$\frac{\partial E}{\partial w_{ij}} = -\sum_{k=1}^{p}\left[d_i^{(k)} - a(net_i^{(k)})\right]a'(net_i^{(k)})x_j^{(k)} \qquad (4.21)$$

ซึ่ง $net_i^{(k)} = w_i^T x^{(k)}$ เป็นอินพุทที่เชื่อมโยงไปยังโครงข่ายนิวรอลอันดับที่ i ในเมื่อมีแพทเทอร์นที่ k ป้อนเข้ามา และ $a'(net_i^{(k)}) = \frac{\partial a(net_i^{(k)})}{\partial net_i^{(k)}}$ ถ้าหากเราใช้สมการ (4.21) ไปปรับเปลี่ยนค่าน้ำหนัก w_{ij} โดยเลือกค่า η น้อยที่สุดเราก็จะได้

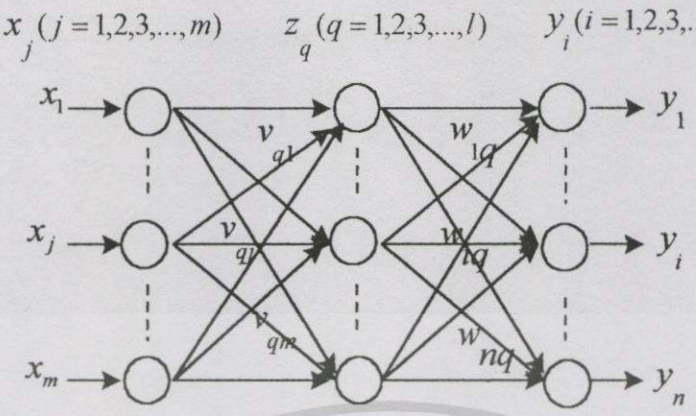
$$\Delta w_{ij} = -\eta\frac{\partial E}{\partial w_{ij}}$$
$$= -\eta[d_i^{(K)} - a(net_i^{(k)})]a'(net_i^{(k)})x_j^{(k)} \qquad (4.22)$$

สมการ (4.22) เรียกว่า "Delta learning rule" โดยทั่วไปด้วยการกำหนด learning signal r เราสามารถเขียนได้คือ

$$r = [d_i - a(w_i^T x)]a'(w_i^T x) \quad (4.23)$$

สำหรับโครงข่ายแบบหลายชั้นตัวอย่างแสดงในรูปที่ 4.12 ที่ใช้ 3 เลเยอร์คือ input layers, hidden layers, output layers ซึ่งในการคำนวณจะใช้พื้นฐานเดียวกันกับโครงข่ายนิวรอลแบบชั้นเดียวที่ประกอบด้วย อินพุท-เอาท์พุท

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.12 แสดงลักษณะโครงข่ายนิวรอลที่มีโครงสร้าง 3 ชั้น

$x_j (j = 1,2,3,...,m)$ คืออินพุทแพทเทอร์นที่มี m โหนด

$z_q (q = 1,2,3,...,l)$ คือ hidden layer ที่มี l โหนด

$y_i (i = 1,2,3,...,n)$ คืออเาท์พุทเลเยอร์ที่มี n โหนด

ใน hidden layer อยู่โหนดที่ q จะมีอินพุทเข้ามาคือ net input ที่มีสมการคือดังนี้

$$net_q = \sum_{j=1}^{m} v_{qj} x_j \qquad (4.24)$$

และจะได้ค่าเอาท์พุทจากโหนดที่ q เป็น

$$z_q = a(net_q) = a\left(\sum_{j=1}^{m} v_{qj} x_j\right) \qquad (4.25)$$

ใน output layer อยู่โหนดที่ i จะมีอินพุทเข้ามาคือ

$$\begin{aligned} net_i &= \sum_{q=1}^{l} w_{iq} z_q \\ &= \sum_{q=1}^{l} w_{iq} a\left(\sum_{j=1}^{m} v_{qj} x_j\right) \end{aligned} \qquad (4.26)$$

และจะได้ค่าเอาท์พุทที่โหนดที่ i เป็น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$y_i = a(net_i) = a\left(\sum_{q=1}^{l} w_{iq} z_q\right)$$

$$= a\left(\sum_{q=1}^{l} w_{iq} a\left(\sum_{j=1}^{m} v_{qj} x_j\right)\right) \qquad (4.27)$$

จากนั้นเราสามารถหาค่าความผิดพลาดและการแพร่ย้อนกลับของมันได้ อันดับแรกเราหาค่า Mean Square Error คือในสมการ (4.20)แล้วเราจะได้

$$E(w) = \frac{1}{2}\sum_{i=1}^{n}(d_i - y_i)^2$$

$$= \frac{1}{2}\sum_{i=1}^{n}[d_i - a(net_i)]^2 \qquad (4.28)$$

$$= \frac{1}{2}\sum_{i=1}^{n}\left[d_i - a\left(\sum_{q=1}^{n} w_{iq} z_q\right)\right]^2$$

ด้วยการใช้วิธี gradient descent algorithm ในการปรับค่าถ่วงน้ำหนัก w_{iq} และ w_{qj} ระหว่างชั้นอิดเดนเลเยอร์ กับชั้นเอาท์พุทเลเยอร์ จะหาได้จาก

$$\Delta W_{iq} = -\eta \frac{\partial E}{\partial W_{iq}} \qquad (4.29)$$

โดยใช้สมการที่ (17) และ (19) และกดสำหรับ $\frac{\partial E}{\partial W_{iq}}$ เราจะได้

$$\Delta W_{iq} = -\eta\left[\frac{\partial E}{\partial y_i}\right]\left[\frac{\partial y_i}{\partial net_i}\right]\left[\frac{\partial net_i}{\partial w_{iq}}\right]$$

$$= \eta[d_i - y_i]\left[a'(net_i)\right][z_q] \qquad (4.30)$$

$$= \eta\delta_{oi} z_q$$

δ_{oi} เป็นค่าความผิดพลาดในชั้นเอาท์พุทเลเยอร์อยู่โหนดอันดับที่ i และกำหนดด้วยสมการ

$$\delta_{oi} = -\frac{\partial E}{\partial net_i} = -\left[\frac{\partial E}{\partial y_i}\right]\left[\frac{\partial y_i}{\partial net_i}\right] \qquad (4.31)$$

$$= [d_i - y_i]\left[a'(net_i)\right]$$

ซึ่ง net_i เป็นอินพุทที่เข้ามาในชั้นเอาท์พุทเลเยอร์อันดับที i และ $a'(net_i) = \frac{\partial a(net_i)}{\partial net_i}$

สำหรับการปรับค่าน้ำหนักในชั้นอินพุทที่เชื่อมต่อกับชั้นอิดเด็นเลเยอร์ โดยทำตามวิธีของ gradient descent method ซึ่งโดยทั่วไปที่ อินพุทอยู่อันดับที่ j ที่ต่อกับอิดเด็นเลเยอร์อยู่อันดับที่ q สามารถหาจะได้จากสมการข้างล่างคือ

$$\Delta v_{qj} = -\eta\left[\frac{\partial E}{\partial v_{qj}}\right]$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$= -\eta \left[\frac{\partial E}{\partial net_q} \right] \left[\frac{\partial net_q}{\partial v_{qj}} \right]$$

$$= -\eta \left[\frac{\partial \mathrm{E}}{\partial \mathrm{z_q}} \right] \left[\frac{\partial z_q}{\partial net_q} \right] \left[\frac{\partial net_q}{\partial v_{qj}} \right] \qquad (4.32)$$

จากสมการ(4.28) แต่ละค่าความผิดพลาดของ $[\mathrm{d_i} - \mathrm{y_i}]$ ซึ่งค่า $\mathrm{i} = 1,2,....,\mathrm{n}$ จะเป็นฟังก์ชั่นของ z_q และเราจะได้

$$\Delta v_{qj} = \eta \sum_{i=1}^{n} \left[(d_i - y_i) a'(net_i) w_{iq} \right] a'(net_q) x_j \qquad (4.33)$$

โดยใช้สมการ (4.31) เราสามารถเขียนสมการ (24)ใหม่ได้คือ

$$\Delta v_{qj} = \eta \sum_{i=1}^{n} [d_{oi} - y_i] a'(net_q) x_j \qquad (4.34)$$
$$= \eta \delta_{hq} x_j$$

ซึ่ง δ_{hq} ค่าความผิดพลาดของ processing element อันดับที่ q ใน hidden layer และกำหนดเป็น

$$\delta_{\mathrm{hq}} = -\frac{\partial \mathrm{E}}{\partial \mathrm{net_q}} = -\left[\frac{\partial \mathrm{E}}{\partial \mathrm{z_q}} \right] \left[\frac{\partial \mathrm{z_q}}{\partial \mathrm{net_q}} \right]$$

$$= a'(net_q) \sum_{i=1}^{n} \delta_{oi} w_{iq} \qquad (4.35)$$

โดยที่ $\mathrm{net_q}$ เป็นอินพุทที่เข้ามาในโครงข่ายนิวรอลอันดับที่ q ในชั้น Hidden layer (มีสมการ 4.24) สัญญาณผิดพลาดในชั้นฮิดเดนเล่เยอร์ได้มีความแตกต่างกันกับสัญญาณผิดพลาดในชั้นเอาท์พุทเล่เยอร์ [คือสมการที่ (4.31) และ (4.35)] เพราะความแตกต่างนั่นโมดูลการปรับแต่งค่าน้ำหนักที่เสนออยู่ข้างบนนี้เรียกว่า "กฎการฝึกสอนโดยทั่วไปแบบ Delta"

ควรใส่ใจ ทั้งสองสมการ (4.31) และ (4.35) มีรูปแบบของการปรับค่าน้ำหนักเหมือนกับสมมการ (4.8) ซึ่งมีสัญญาณการฝึกสอนแตกต่าง เว้แต่ให้ค่า $\mathrm{r} = \delta$ การฝึกสอนแบบย้อนกลับของโครงข่ายนิวรอลหลายชั้นสามารถกำหนดรูปแบบโดยทั่วไปของสมการปรับแต่งค่าน้ำหนักได้ดังนี้

$$\Delta \mathrm{w_{ij}} = \eta \delta_{\mathrm{i}} \mathrm{x_j} = \eta \delta_{\mathrm{output-i}} \mathrm{x_{input-j}} \qquad (4.36)$$

โดยที่ output-i และ input-j กล่าวถึงจุดเชื่อมโยงจากโครงข่ายนิวรอลโหนดที่ j หาโหนดที่ i $\mathrm{x_j}$ เป็นอินพุทที่เข้ามาในโหนดที่ j ของชั้นฮิดเดนเล่เยอร์ และ δ_{i} เป็นอัตราการฝึกสอนที่มีสมการ (4.31) สำหรับชั้นเอาท์พุทเล่เยอร์สุดท้ายจะมีสมการ (4.35)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Algorithm BP: Back-Propagation Learning Rule (ดูรูปที่ 4.13)

ให้ Network ที่มี Q layer, q=1,2,...,Q ให้ ${}^{q}net_i$ เป็น net input อยู่โหนดที่ i ในชั้นที่ q ให้ ${}^{q}y_i$ เป็น output อยู่โหนดที่ i ในชั้นที่ q ให้ ${}^{q}w_{ij}$ เป็นค่าถ่วงน้ำหนักระหว่างชั้น ${}^{q-1}y_j$ หา ${}^{q}y_i$

Input: ให้คู่ Training Set $\{(X^k, d^k) / k = 1,2,3,..., p\}$

Step 0 (Initialization):เลือกค่า $\eta > 0$ และ $E_{\max} = 0.01$ (maximum tolerable error)ให้ ค่า weights to small random. Set $E = 0 \; and \; k = 1$

Step 1(Training Loop): ให้ค่าที่ โหนดที่ k ของ input pattern (layer q=1) เป็น

$${}^{q}y_i = {}^{1}y_i = x_i^{(k)} \quad \text{สำหรับทุกค่า } i$$

Step 2(Forward propagation): Propagate the signal forward through the network using

$${}^{q}y_i = a({}^{q}net_i) = a(\sum_j w_{ij} \, {}^{q-1}y_j)$$

Step 3(Output error measure) คำนวณค่าผิดพลาดและค่า signals ${}^{Q}\delta_i$ สำหรับ output layer:

$$E = \frac{1}{2}\sum_{i=1}^{n}(d_i^{(k)} - {}^{Q}y_i)^2 + E$$

$${}^{Q}\delta_i = ({}^{Q}d_i - {}^{Q}y_i)a'({}^{Q}net_i)$$

Step 4(Error back-propagation) propagate the error backward to update the weights and compute the error signals ${}^{q-1}\delta_i$ for the preceding layer $\Delta {}^{q}w_{ij} = \eta \, {}^{q}\delta_i \, {}^{q-1}y_j$ and ${}^{q}w_{ij}^{new} = {}^{q}w_{ij}^{old} + \Delta {}^{q}w_{ij}$ ${}^{q-1}\delta_i = a'({}^{q-1}net_i)\sum_j {}^{q}w_{ji} \, {}^{q}\delta_j$ for q=Q,Q-1,...,2

Step 5(Once epoch): Check whether the whole set of training data has been cycled once. If k<p, then k=k+1 and go to step 1; otherwise, go to step 6.

Step 6(Total error checking): Check whether the current total error is acceptable: if $E<E_{max}$ then terminate the training process and output the final weights; otherwise E=0,k=1 and initiate the new training epock by going to step 1.

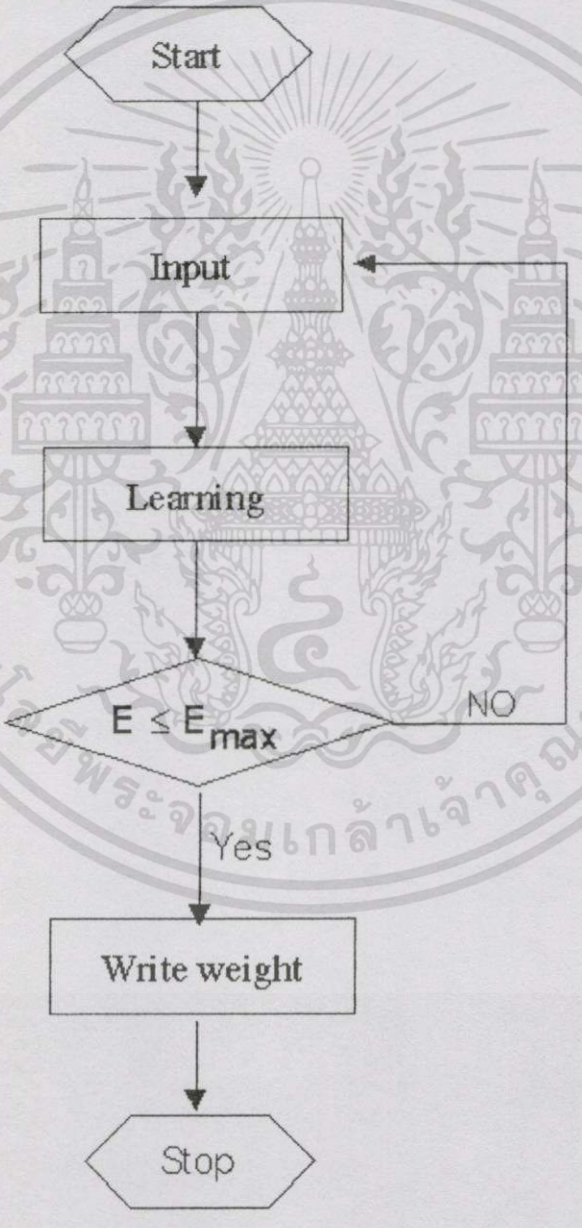
End BP

4.3 การใช้โครงข่ายนิวรอลในการรู้จำตัวอักษร

ในงานวิจัยนี้ได้ใช้วิธีของโครงข่ายนิวรอลแบบแพร่ย้อนกลับ มาฝึกสอนโดยใช้การฝึกสอนแบบชี้นำ(Supervised Learning) ซึ่งใช้โครงข่ายนิวรอลที่มี 3 ชั้นคือ อินพุทเล่เยอร์ใช้ 9 โหนด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ฮิดเดนเล่เยอร์ใช้ 120 โหนด และ เอาท์พุทเล่เยอร์ใช้ 46 โหนด ข้อมูลที่เอามาฝึกสอนมี 8 รูปแบบ แต่ละรูปแบบจะมี 4 ขนาดแตกต่างกันคือ 14, 16, 18 และ 20 (point) ได้กำหนดค่าน้ำหนักเริ่มต้นด้วยค่าแบบสุ่ม (Random) ใช้ค่าของ Learning Rate เท่ากับ 0.1 และ ค่า SSE (Sum Square Error) เท่ากับ 0.01 ค่าน้ำหนัก (weight) ที่ได้จากการฝึกสอนครั้งสุดท้ายจะเก็บไว้เพื่อใช้ทดสอบความถูกต้องของระบบการรู้จำตัวอักษร จากระบบโครงข่ายที่ได้กล่าวมาข้างบนนั่นจะได้ weight Matrix ในชั้นอิดเดนเลเยอร์ มีขนาดเท่ากับ 120×9 และ weight matrix ในชั้นเอาท์พุทเลเยอร์ มีขนาดเท่ากับ 46×120



รูปที่ 4.13 แสดงขั้นตอนการฝึกสอนของโครงข่ายนิวรอล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
โปรแกรมการรู้จำตัวอักษร

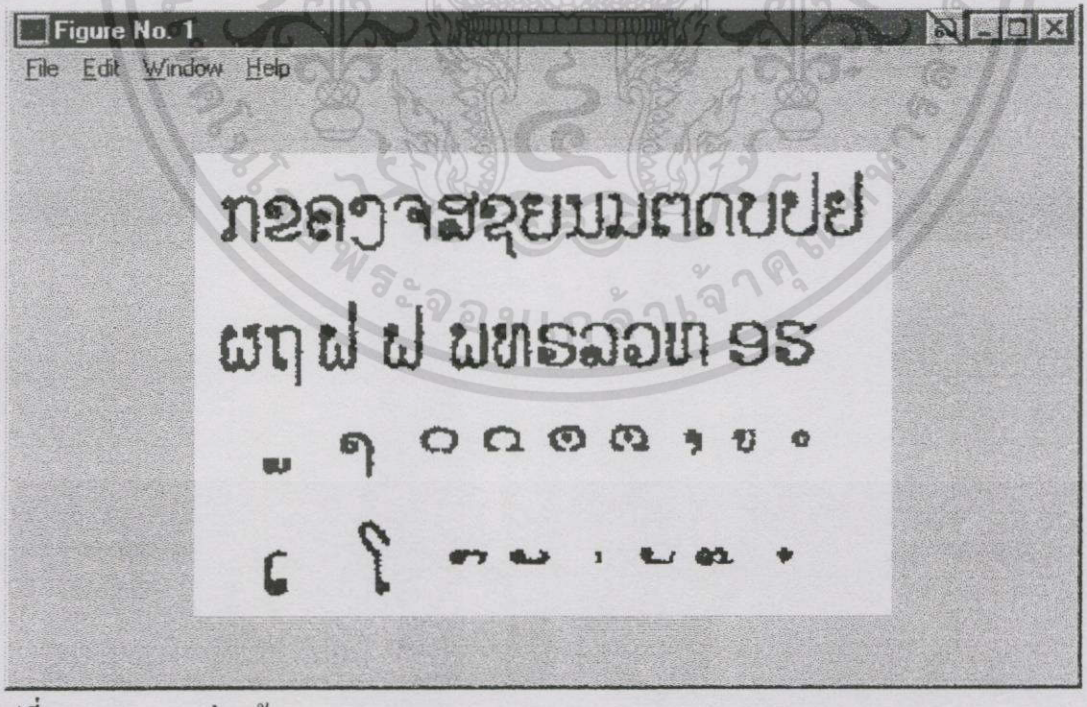
โปรแกรมการรู้จำตัวอักษรได้แบ่งออกเป็นสามส่วนคือ การเตรียมข้อมูล การฝึกสอน และ การทดสอบการรู้จำ

5.1 การเตรียมข้อมูล

จะมีโปรแกรมหลักคือ

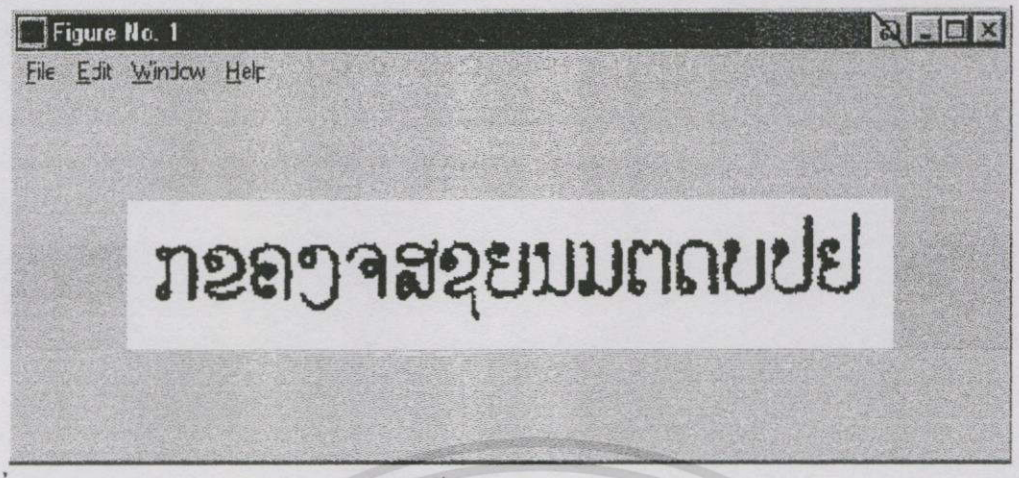
- PreAlice.m , Preneural.m , SegProjection.m , Neuro_Fuzzy.m , Memberchar.m

โดยจะทำหน้าที่ Load file Data แสดงในรูปที่ 5.1 จากนั้นจะทำการแยกบรรทัดออกจากหน้ากระดาษ ดูในรูปที่ 5.2 แล้วทำการตัดตัวอักษรแต่ละตัวออกจากบรรทัด แสดงในรูปที่ 5.3 จากนั้นจะทำการคำนวณหาค่าคุณลักษณะเด่นของตัวอักษรซึ่งแยกออกเป็นส่วนๆไป ดูรูปที่ 5.4 และรูปที่ 5.5

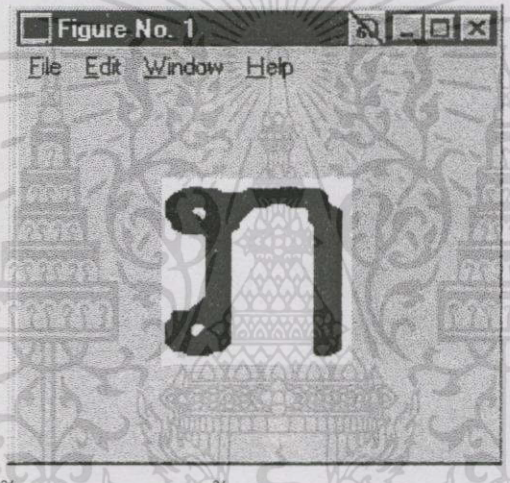


รูปที่ 5.1 แสดงการอ่านข้อมูลจาก Bitmap file

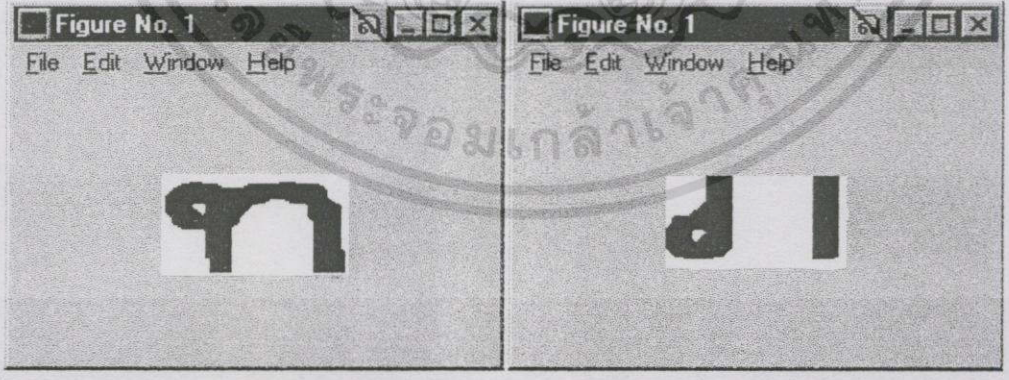
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.2 แสดงการแยกบรรทัดออกจากหน้ากระดาษ

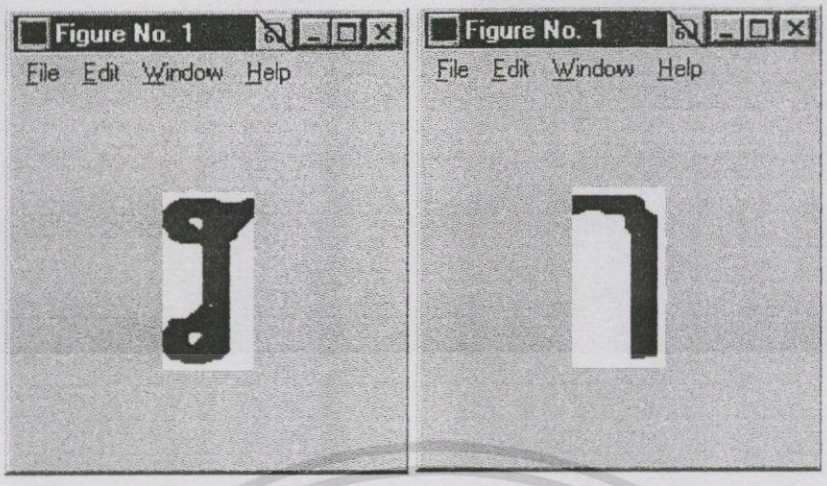


รูปที่ 5.3 แสดงการแยกตัวอักษรออกจากบรรทัด



รูปที่ 5.4 แสดงการแบ่งส่วน UP และ DONW

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.5 แสดงการแบ่งส่วน LEFT และ RIGHT

ในส่วนนี้ โปรแกรมจะทำในลักษณะเดียวกันสำหรับตัวอักษรตัวอื่นๆจนครบตัวอักษรที่มีในหน้ากระดาษ และ สุดท้ายเราจะได้ค่าของคุณลักษณะเด่นของตัวอักษรแบบฟัซซี่ ดังได้เสนอตัวอย่างไปแล้วในตารางที่ 3.1 ในบทที่ 3 เพื่อจะไปใช้ในขั้นตอนต่อไป

5.2 ขั้นตอนการฝึกสอน

ขั้นตอนการฝึกสอนคือนำเอาข้อมูลที่ได้จากข้อ 5.1 มาทำการประมวนผลเพื่อหาค่าน้ำหนัก ในส่วนนี้จะมีโปรแกรมหลักคือ

- Learning.m

ซึ่งได้แบ่งออกเป็นหลายตอนเพื่อลดเวลาของการฝึกสอนให้น้อยลง โดยใช้โปรแกรมย่อยคือ

- Learnfirst.m , Learn2.m , Learn3.m ...

ได้กำหนดรอบไว้ 100,000 กำหนด SSE =0.01 ซึ่งโปรแกรมจะทำการประมวนผลไปเรื่อยๆจนกว่าจะได้ค่าน้ำหนักที่เหมาะสม ดูรูปที่ 5.6

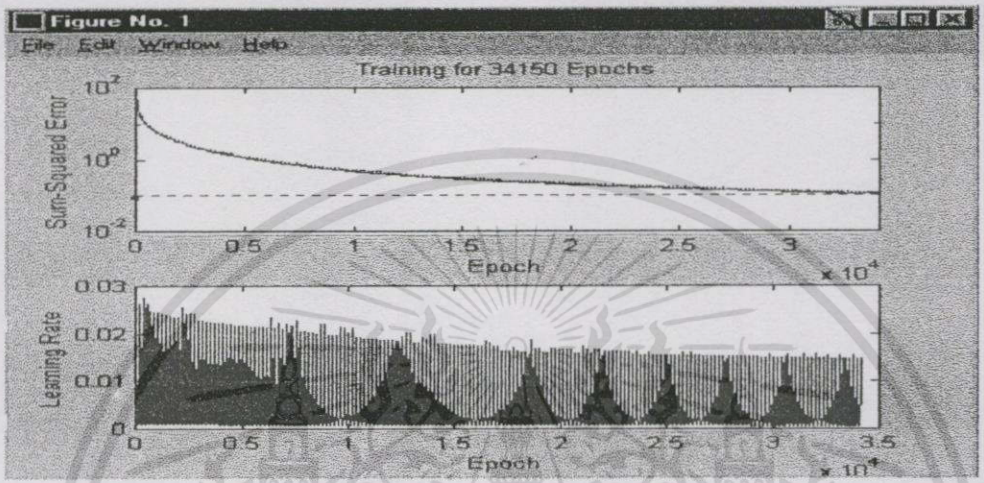
5.3 ขั้นตอนทดสอบ

ในขั้นตอนนี้เป็นขั้นตอนที่เรานำเอาตัวอักษรที่ไม่เคยฝึกสอนมาก่อนมาทดสอบ เราจะมีโปรแกรมหลักคือ

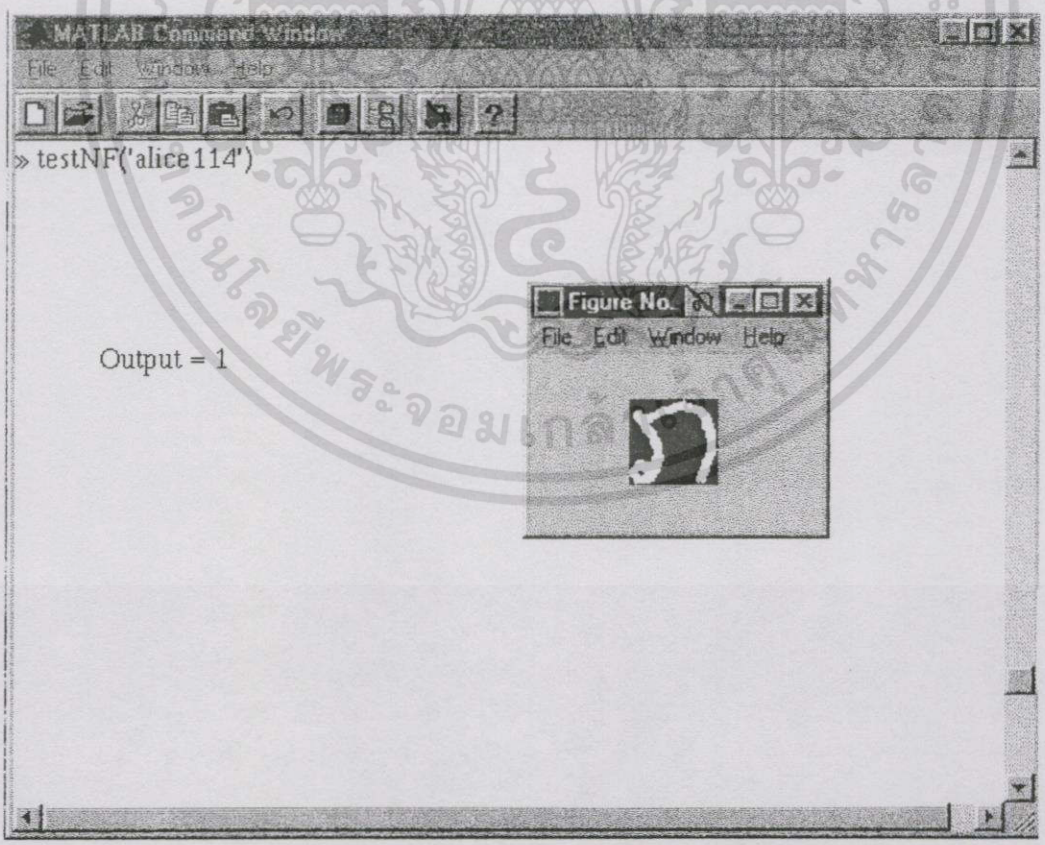
- TestNF.m , SimNet.m

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมจะทำการเตรียมข้อมูลเหมือนกับในหัวข้อ 5.1 แล้วเก็บค่าของแต่ละตัวมาคูณกับค่าน้ำหนักเพื่อหาผลรับแล้วเปรียบเทียบกับค่าเป้าหมาย จากนั้นโปรแกรมจะบอกได้ว่าเอาท์พุทเป็นตัวอักษรอะไร ซึ่งในโปรแกรมจะบอกเลขรหัสของตัวอักษรออกมา (เลขรหัสตัวอักษรที่ทำไว้ในตาราง 3.1 หัวข้อที่ 3.6 บทที่ 3) แสดงในรูปที่ 5.7



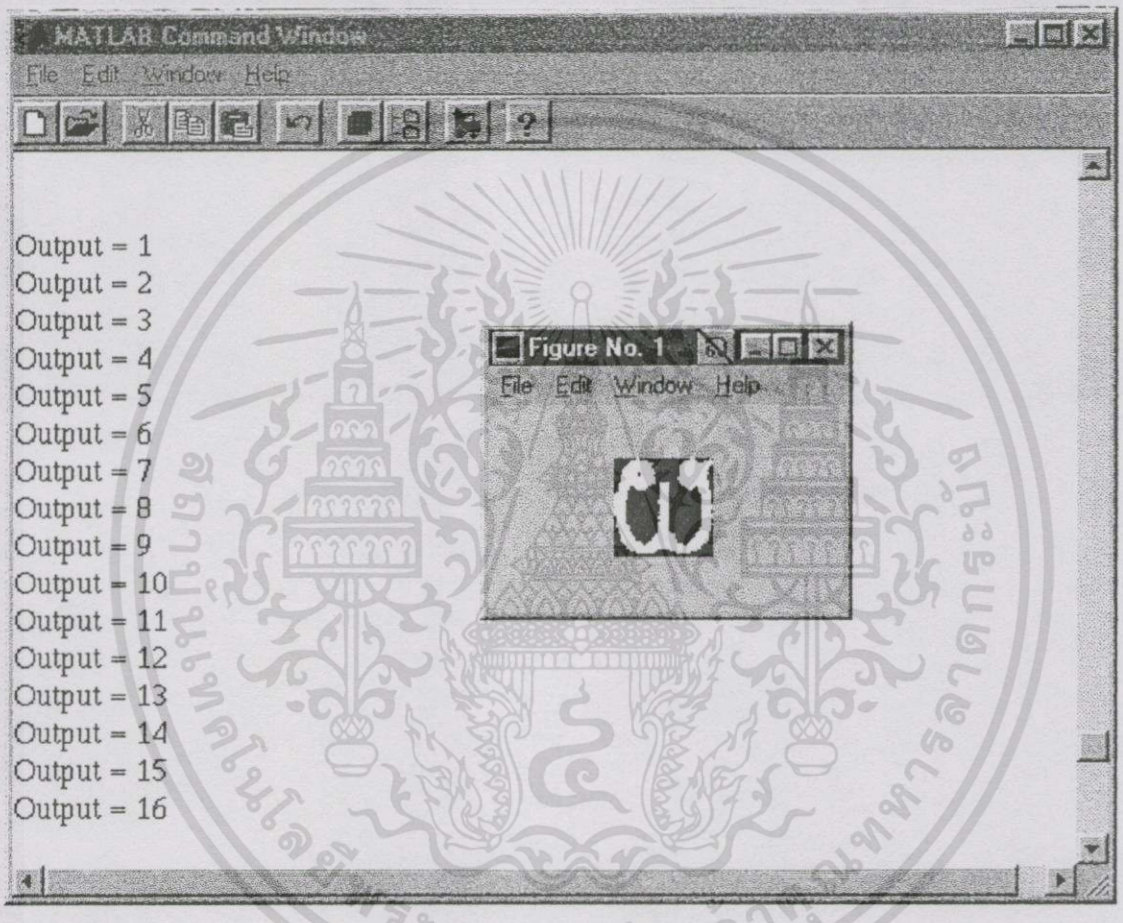
รูปที่ 5.6 แสดงกราฟการทำงานของการฝึกสอน



รูปที่ 5.7 แสดงการรู้จำตัวอักษร "ว" ให้ค่าเอาท์พุทเท่ากับ 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมทำการประมวญผลตัวอักษรต่อไปพร้อมกับแสดงค่าเอาท์พุทที่เป็นรหัสประจำตัวอักษรออกมา ดูในรูปที่ 5.8 จนกว่าจะจบหน้ากระดาษจึ่งเป็นการสิ้นสุดกระบวนการทดสอบการรู้จำตัวอักษร ซึ่งเราสามารถตรวจดูค่าความถูกต้อง และตอนท้ายโปรแกรมจะแสดงผลละเอียดค่าความถูกต้อง ค่าความผิดพลาด และ บอกรหัสตัวอักษรที่ผิดพลาดออกมาให้อีกด้วย



รูปที่ 5.8 แสดงการรู้จำตัวอักษร พร้อม ให้ค่าเอาท์พุท

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 6
การทดลองและประเมินผล

การหาความถูกต้องของการรู้จำตัวอักษรนั่น จะต้องได้ทำการทดลองหลายครั้ง หลายรูปแบบที่แตกต่างกัน เพื่อจะได้ค่าความถูกต้องที่เหมาะสม ดังนั้นในบทนี้จะได้กล่าวถึงการทดลองที่ได้ใช้ ซึ่งแบ่งออกเป็นขั้นตอนดังนี้

6.1 กำหนดจำนวนโหนดของชั้นฮิดเดนเล่เยอร์ที่เหมาะสม

ในขั้นตอนนี้เราทำการทดลองเพื่อหาจำนวนโหนดของชั้นฮิดเดนเลเยอร์ที่เหมาะสม จากข้อมูลอินพุทที่ใช้ในชั้นอินพุทเล่เยอร์ 9 โหนด ข้อมูลเอาท์พุทที่ต้องการในชั้นเอาท์พุทเล่เยอร์ 46 โหนด สำหรับชั้นฮิดเดนเล่เยอร์ได้กำหนดโหนดที่ใช้มี 6 ค่าคือ 60 80 100 120 150 200 วิธีการทดลองคือเอากลุ่มตัวอักษรมีรูปแบบเดียวกันมาทำการฝึกสอน กำหนดค่าการเรียนรู้ไว้เท่ากับ 0.1 กำหนดค่าความผิดพลาดไว้เท่ากับ 0.01 โดยเปลี่ยนแปลงจำนวนโหนดในชั้นฮิดเดนเล่เยอร์ 60 80 100 120 150 200 ตามลำดับ ผลการทดลองแสดงออกในตาราง 6.1

ตารางที่ 6.1 ผลการทดสอบโครงข่ายนิวรอลโดยกำหนดฮิดเดนโหนด

จำนวนโหนดในชั้นฮิดเดนเล่เยอร์	เวลาที่ใช้ในการฝึกสอน (ชั่วโมง)	ผลทดสอบความถูกต้องการรู้จำตัวอักษร				
		จำนวนตัวอักษรที่ทดสอบ	จำตัวอักษรถูกต้อง	จำตัวอักษรผิดพลาด	ถูกต้องร้อยละ (%)	ผิดพลาดร้อยละ (%)
60	6	368	354	14	96.20	3.80
80	8	368	361	7	98.09	1.91
100	10	368	364	4	98.91	1.09
120	12	368	368	0	100	0
150	15	368	368	0	100	0
200	19	368	368	0	100	0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในขั้นตอนนี้เราได้ค่าความถูกต้องที่ดีที่สุดในการรู้จำตัวอักษรเท่ากับ 100% โดยได้ใช้จำนวนโหนดฮิดเดนเล่เยอร์เท่ากับ 120 150 200 โหนดตามลำดับ ดังนั้นจำนวนโหนดในชั้นฮิดเดนเล่เยอร์ที่เหมาะสมสำหรับใช้ทดลองในขั้นตอนต่อไปคือ 120 โหนด

6.2 ทำการทดลองในกลุ่มตัวอักษรที่มีรูปแบบเดียวกันแต่มีขนาดแตกต่างกัน

วิธีการที่ใช้คือเอากลุ่มอักษรที่มีรูปแบบเดียวกันแต่มีขนาดแตกต่างกันมาทำการฝึกสอน โดยกำหนดอัตราค่าเรียนรู้เท่า 0.1 กำหนดค่าผิดพลาดเท่า 0.01 และ การฝึกสอนในแต่ละกลุ่มตัวอักษรได้ใช้เวลาประมาณ 72 ชั่วโมง จากนั้นเอาตัวอักษรรูปแบบเดียวกันมาทดสอบหาความถูกต้อง แสดงผลการทดสอบในตาราง 6.2

ตารางที่ 6.2 แสดงผลการทดสอบการรู้จำตัวอักษรที่มีรูปแบบเดียวกันแต่ขนาดแตกต่าง

กลุ่มตัวอักษร	ผลการทดสอบการรู้จำตัวอักษร				
	จำนวนทดสอบ	จำนวนถูกต้อง	จำนวนผิดพลาด	ถูกต้องร้อยละ (%)	ผิดพลาดร้อยละ (%)
Alice0 Lao	552	528	24	95.65	4.35
Alice1 Lao	552	531	21	96.09	3.81
Alice2 Lao	552	502	50	90.94	9.06
Alice3 Lao	552	503	49	91.12	8.88
Alice4 Lao	552	521	31	94.38	5.62
Alice5 Lao	552	522	30	94.57	5.43
Chanthabouli Lao	552	508	44	92.03	7.97
Saysettha Lao	552	514	38	93.12	6.88
ค่าเฉลี่ย	4416	4129	287	93.50	6.50

6.3 การทดลองในกลุ่มตัวอักษรที่มีรูปแบบและขนาดแตกต่างกัน

วิธีการคือเอากลุ่มตัวอักษรที่มีรูปแบบและขนาดต่างกันมาทำการฝึกสอน โดยกำหนดอัตราค่าเรียนรู้เท่า 0.1 กำหนดค่าผิดพลาดเท่า 0.01 และ ข้อมูลตัวอักษรได้แบ่งออกเป็นสองกลุ่มเพื่อลดเวลาในการฝึกสอน คือกลุ่มแรกประกอบด้วยตัวอักษร Font Alice0 Lao, Alice1 Lao, Alice2 Lao, Alice3 Lao กลุ่มที่สองประกอบด้วยตัวอักษร Font Alice4 Lao, Alice5 Lao, Chanthabouli Lao, Saysettha Lao การฝึกสอนในแต่ละกลุ่มตัวอักษรได้ใช้เวลาประมาณ 96 ชั่วโมง และผลการทดสอบได้แสดงในตารางที่ 6.3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 6.3 แสดงผลการทดสอบการรู้จำตัวอักษรที่มีรูปแบบแตกต่างกัน

กลุ่มตัวอักษร	ผลการทดสอบการรู้จำตัวอักษร				
	จำนวนทดสอบ	จำนวนถูกต้อง	จำนวนผิดพลาด	ถูกต้องร้อยละ (%)	ผิดพลาดร้อยละ (%)
กลุ่มแรก	736	685	51	93.07	6.93
กลุ่มสอง	736	680	56	92.39	7.61
ค่าเฉลี่ย	1472	1365	107	92.73	7.37

6.4 วิเคราะห์ผลการทดลอง

ในการทดลองการรู้จำตัวอักษรพิมพ์ภาษาลาวโดยใช้ตัวอักษรที่มีรูปแบบและขนาดแตกต่างกัน และใช้การประมวลผลบนเครื่อง PC รุ่น Pentium II มี CPU เท่า 350 MHz พบว่า การลดขนาดข้อมูลตัวอักษรและการประยุกต์ใช้ค่าแบบฟัซซี่เข้ามาฝึกสอนทำให้เวลาในการฝึกสอนลดลง การทดสอบในการรู้จำตัวอักษรที่มีรูปแบบเดียวกันค่าความถูกต้องได้เฉลี่ยร้อยละ 93.50 การทดสอบโดยใช้ตัวอักษรที่มีรูปแบบแตกต่างกันพบว่า ค่าความถูกต้องได้เฉลี่ยร้อยละ 92.73

สำหรับตัวอักษรพิมพ์ภาษาลาวที่มีความผิดพลาดบ่อยครั้งในการทดสอบนั่นส่วนมากจะเป็นกลุ่มตัวอักษรที่มีรูปร่างใกล้เคียงกันเช่น ตัวพะยัญชนะ “ ป ” เมื่อเอาท์พุทออกมาจะเป็นตัวพะยัญชนะ “ ป ” ตัวพะยัญชนะ “ ร ” เมื่อเอาท์พุทออกมาจะเป็นตัวพะยัญชนะ “ ร ” ตัวพะยัญชนะ “ อ ” เมื่อเอาท์พุทออกมาจะเป็นตัวพะยัญชนะ “ ร ”

6.5 สรุปและประเมินผล

ในการทดลองการรู้จำตัวอักษรพิมพ์ภาษาลาวโดยการประยุกต์ใช้ระบบฟัซซี่ และโครงข่ายนิวรอลเน็ตเวอร์กแบบแพร่ย้อนกลับได้ใช้ 3 ชั้นคือ ชั้นอินพุทเล่เยอร์ ชั้นฮิดเดนเล่เยอร์ และ ชั้นเอาท์พุทเล่เยอร์ สำหรับชั้นอินพุทได้ใช้ 9 โหนด ชั้นเอาท์พุทได้ใช้ 46 โหนด ส่วนชั้นฮิดเดนได้ใช้หลายค่าคือ 60 80 100 120 150 และ 200 โหนด ตามลำดับ เพื่อหาจำนวนโหนดที่เหมาะสม ในการทดลองนี้ได้เลือกเอาจำนวนโหนดในชั้นฮิดเดน 120 โหนด เพื่อลดเวลาที่ใช้ในการฝึกสอนให้น้อยลง ข้อมูลตัวอักษรได้ถูกแปลงและลดขนาดลงให้อยู่ในรูปแบบฟัซซี่ ก่อนส่งไปฝึกสอนในโครงข่ายนิวรอลเน็ตเวอร์ก การเพิ่มจำนวนโหนดในชั้นฮิดเดนได้มีผลทำให้ค่าความถูกต้องในการรู้จำตัวอักษร และการใช้เวลาในการฝึกสอนเพิ่มขึ้น การเพิ่มจำนวนตัวอักษรเข้าในการฝึกสอนก็ทำให้ค่าความถูกต้องในการรู้จำ และ เวลาที่ใช้ในการฝึกสอนเพิ่มขึ้นเช่นกัน ความผิดพลาดในการรู้จำส่วนมากเกิดจากการเก็บข้อมูลที่เป็นข้อมูลภาพแบบ Bitmap ที่ได้จากเครื่องสแกนเนอร์ในแต่ละ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ครั้ง ทำให้ตัวอักษรตัวเดียวกันได้รับความเข้มของแสงแตกต่างกัน เกิดจากตัวอักษรบางตัวมีรูปร่างใกล้เคียงกันมาก ตัวอักษรพิมพ์ภาษาลาวที่มีความผิดพลาดอย่างน้อยหนึ่งครั้งส่วนมากคือตัวพะยัญชนะ “ ຄ ງ ຫ ອ ຮ ” และสระ “ ◌ິ ◌ັ ◌ຼ ” การกำหนดค่าคุณลักษณะเด่นของตัวอักษรโดยที่เราลดข้อมูลมากเกินไป อาจทำให้การประมวลผลตัวอักษรผิดพลาดได้เช่นกัน

แนวทางพัฒนาในอานาคตเนื่องจากผลได้รับและความผิดพลาดที่เกิดจากการทดลองนั่น การเก็บข้อมูลภาพตัวอักษรควรจะทำในเงื่อนไขเดียวกันทุกประการ และใช้เครื่องสแกนเนอร์ที่มีความละเอียดสูง เพื่อไม่ให้รายละเอียดตัวอักษรบางตัวขาดหายไป ตัวอักษรที่มีรูปร่างใกล้เคียงกันควรจัดเป็นกลุ่มเดียวกันก่อนแล้วจึงทำการแยกภายในกลุ่มนั้นอีกครั้ง เพื่อลดเวลาในการฝึกสอน ควรทำการประมวลผลด้วยเครื่อง PC ที่มีความเร็วสูงกว่า 350 MHz ขึ้นไป การเพิ่มจำนวนตัวอักษรให้มากขึ้นในการฝึกสอนหรือเพิ่มจำนวนฮิดเดนโหนด จะทำให้ค่าความถูกต้องเพิ่มขึ้นอย่างมีประสิทธิภาพ

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารอ้างอิง

[1] C.Kimpan, A.Ttoh, and K.Kawanishi, "Recognition of printed Thai character using a matching method", IEE Proc., vol.130, Pt.E, No.6, pp. 183-188, Nov.1983.

[2] C.Kimpan, "Printed Thai character recognition using topological properties method." INT.J. Electronics, vol.60, No.3, pp. 303-329, 1986.

[3] C.Kimpan, A.Itoh,. and K.Kawanishi, "Fine classification of printed Thai character recognition using the Karhunen-Loeve expansion", Proc.IEE, vol.134,Pt.E.,No.5, pp. 257-264, Sept. 1987.

[4] Chin-Teng Lin & C.S. george Lee. 1996. Neural Fuzzy systems. The United States of America: Prentice Hall.

[5] Chom Kimpan.1986. Printed Thai Characters Recognition. Dissertation of Engineering, Faculty of Engineering, King Mongkut's Institute of Technology Chaokhurn Taharn ladkrabang.

[6] DAN W. Patterson.1996. Artificial Neural Networks Theory and Applications. Singapore: Prentice Hall.

[7] H.Ishibuchi, R.Fujioka and H.Tanaka,"Neural Networks that Learn from Fuzzy IF-THEN Rules." IEEE Trans. On Fuzzy systems. Vol.1, No.2, pp. 85-97, May 1993.

[8] Morton Nadler & ERIC P.Smith.1993. Pattern Recognition Engineering. The United States of America: NewYork.

[9] P.Hirananichakorn, T. Agui, and M.Nakajima, "A recognition method of handprinted Thai characters by local features." IECE trans., vol.E68, No. 2, pp. 83-90, Feb. 1985.

[10] P.Hirananichakorn., T. Agui,and M.Nakajima, "An online recognition method of Thai characters", IECE Trans. Vol.E68, No. 11, pp. 594-601, Nov 1985.

[11] P.Hirananichakorn., T. Agui,and M.Nakajima, "Recognition method of Thai characters by using local features." Trans. IECE, vol. E67, No. 8, pp. 425-432, Aug.1984

[12] P.Hirananichakorn., T.Agui,and M.Nakajima, "A recognition method of Thai characters." IECE Trans. Vol.E65, No. 12, pp. 737-744, Dec.1982.

[13] P.Phokharatkul and C.Kimpan, "Recognition of handprinted Thai characters Using the cavity features of character based on neural network" IEEE International Conference on 1998, pp.149-152.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

[14] Pisit Phokharatkul & Chom Kimpan.1998. Recognition of Handprinted Thai Characters using the cavity Features of character Bases on Neural Network. The 1998 IEEE Asia-Pacific Conference on Circuits and Systems, Novembre 24-27 1998. Chiangmai Thailand.

[15] S.Airphaiboon and S.Kondo, "Recognition of handprinted Thai Characters Using Loop structures." IEICE Trans. INF.& SYST., vol.E79-D,No.9, pp. 1296-1304, Sep.1996.

[16] S.Airphaiboon, M.Sangworasil, B.Attachoo, and C. Kimpan, "Recognition of handwritten Thai character considering the head of character." IEEE International Conference on Image Processing. Vol.2, Singapore, pp. 326-330, 1989.

[17] Stephen T. Welstead.1994. Neural Network and Fuzzy Logic Applications in C/C++. The United States of America: NewYork.

[18] Valluru Rao & Hyagriva Rao.1995. C++ neural Networks and Fuzzy logic. The United states of America: NewYork.

[19] พิศิษฏ์ โภคารัตน์กุล "การรู้จำอักษรตัวพิมพ์ภาษาไทยโดยใช้วิธีโปรเจ็กชั่นฟีเจอร์และฟัซซี่เทคนิค" การประชุมวิชาการวิทยาศาสตร์คอมพิวเตอร์และวิศวกรรมคอมพิวเตอร์แห่งชาติ ครั้งที่ 1 (NCSEC98) มหาวิทยาลัยเกษตรศาสตร์ 19-21 ตุลาคม 2541

[20] ขิงทอง อินทวงศ์ศา . 2541. " การรู้จำลายมือเขียนตัวอักษรภาษาลาวโดยใช้โครงข่ายประสาทเทียม " วิทยานิพนธ์วิศวกรรมศาสตรมหาบัณฑิต. สาขาวิศวกรรมไฟฟ้าบัณฑิตวิทยาลัย. สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง.

[21] สกุล คำนวนชัย. 2542. " การรู้จำตัวอักษรคัดลายมือเขียนภาษาไทยประยุกต์ใช้การทรานสฟอร์มแบบคาร์ฮูเน็นเลิฟเข้ากับโครงข่ายฟัซซี่นิวรอล " วิทยานิพนธ์วิศวกรรมศาสตรมหาบัณฑิต. สาขาวิศวกรรมไฟฟ้าบัณฑิตวิทยาลัย. สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก

ตัวอักษรในภาษาลาวทั้งหมด มี 66 ตัว คือมีพยัญชนะ 33 ตัว สระ 29 ตัว และ วรรณยุกต์ 4 ตัว รายระเอียดดังนี้

1. พยัญชนะ 33 ตัว ได้แบ่งออกเป็นสองกลุ่มคือ

1.1 พยัญชนะเดียว 27 ตัว

ກ ຂ ຄ ງ ຈ ສ ຊ ຍ ນ ມ ຕ ຖ ບ
ປ ຢ ຜ ຖ ຝ ຟ ພ ທ ຣ ລ ວ ຫ ອ ຮ

1.2 พยัญชนะควบ 6 ตัว

ໜ ໜ ໝ ໝ ໜ ໜ

2. สระ 29 ตัว ได้แบ่งออกเป็นสามกลุ่มคือ

2.1 สระเสียงสั้น 12 ตัวคือ

ะ ิ ึ ุ เ แ โ เ
เ เ เ ็

2.2 สระเสียงยาว 13 ตัวคือ

า ี ื ู เ แ โ
เ เ เ ็ ็

2.3 สระพิเศษ 4 ตัวคือ

เ า ำ ไ ใ

3. วรรณยุกต์ 4 ตัวคือ

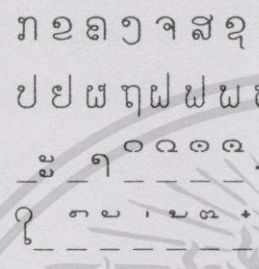
่ ้ ๊ ๋

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

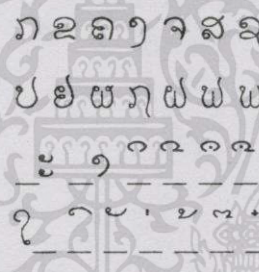
ภาคผนวก ข

รูปแบบอักษรที่นำมาฝึกสอนทั้งหมด มี 8 รูปแบบแตกต่างกัน คือ

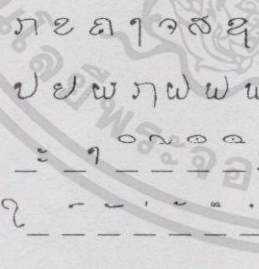
1. รูปแบบของ Font Alice0



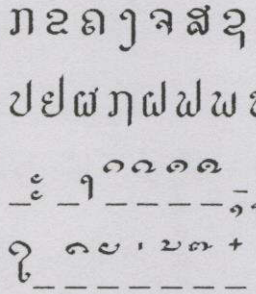
2. รูปแบบของ Font Alice1



3. รูปแบบของ Font Alice2



4. รูปแบบของ Font Alice3



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. รูปแบบของ Font Alice4

ก ข ฃ ง จ ส ช ย ม ม ต ถ บ
ป ย ผ ถ ฝ ฝ พ ห ธ ล อ ฑ ฮ ร
_ะ _า _ _ _

ภาคผนวก ค

โปรแกรมการรู้จำตัวอักษรประกอบด้วยสามส่วนคือ ส่วนแรกขั้นตอนเตรียมข้อมูลตัวอักษร ส่วนที่สองการฝึกสอนเพื่อหาค่าน้ำหนักที่เหมาะสมเก็บไว้ ส่วนที่สามการทดสอบรู้จำตัวอักษร รายระเอียดดังนี้

1. ขั้นตอนเตรียมข้อมูล

```
function PreAlice()
alc014 = PreNeural('alice014');
alc016 = PreNeural('alice016');
alc018 = PreNeural('alice018');
alc020 = PreNeural('alice020');
save d:\project1\temp\alice0 alc014 alc016 alc018 alc020;
alc014 = PreNeural('alice014');
alc016 = PreNeural('alice016');
alc018 = PreNeural('alice018');
alc020 = PreNeural('alice020');
save d:\project1\temp\alice0 alc014 alc016 alc018 alc020;
alc114 = PreNeural('alice114');
alc116 = PreNeural('alice116');
alc118 = PreNeural('alice118');
alc120 = PreNeural('alice120');
save d:\project1\temp\alice1 alc114 alc116 alc118 alc120;
alc214 = PreNeural('alice214');
alc216 = PreNeural('alice216');
alc218 = PreNeural('alice218');
alc220 = PreNeural('alice220');
save d:\project1\temp\alice2 alc214 alc216 alc218 alc220;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
alc314 = PreNeural('alice314');
alc316 = PreNeural('alice316');
alc318 = PreNeural('alice318');
alc320 = PreNeural('alice320');
save d:\project1\temp\alice3 alc314 alc316 alc318 alc320;
alc414 = PreNeural('alice414');
alc416 = PreNeural('alice416');
alc418 = PreNeural('alice418');
alc420 = PreNeural('alice420');
save d:\project1\temp\alice4 alc414 alc416 alc418 alc420;
alc514 = PreNeural('alice514');
alc516 = PreNeural('alice516');
alc518 = PreNeural('alice518');
alc520 = PreNeural('alice520');
save d:\project1\temp\alice5 alc514 alc516 alc518 alc520;
cht14 = PreNeural('chant14');
cht16 = PreNeural('chant16');
cht18 = PreNeural('chant18');
cht20 = PreNeural('chant20');
save d:\project1\temp\cht cht14 cht16 cht18 cht20;
sst14 = PreNeural('sayset14');
sst16 = PreNeural('sayset16');
sst18 = PreNeural('sayset18');
sst20 = PreNeural('sayset20');
save d:\project1\temp\sset sst14 sst16 sst18 sst20;
%==========end==========
function PreNuralMemberShip=PreNeural(InFileName)
if nargin <= 0
  InFileName = input('Type File Name that you want > ','s');
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
end;
InC=strcat(InFileName,'.bmp');
data=imread(InC);
[MaxRow,MaxColume]=size(data);
%1. Before Segment Character from palagraph we have to check something
if (data(1,1)==1)&(data(MaxRow,MaxColume)==1)
   data = ToRevert(data);
end;
imshow(data);%pause;
clc;
%2. Find Line of the Grammar
LineS = SegProjection(data,15);
%3. Segment Sentance from the Grammar
[MaxLineR,MaxLineC]=size(LineS);
[MaxDR,MaxDC]=size(data);
StartS = 1;
EndS   = 2;
PreNuralMemberShip = [];
for IdLine = 1:MaxLineR
   StartL = LineS(IdLine,StartS);
   EndL   = LineS(IdLine,EndS);
   Sentance = data(StartL:EndL,1:MaxDC);
   Sentance = imfilter(Sentance,5);%Remove noise out of sentance;
   imshow(Sentance);%pause;
   TmpMemberShip=SegFromSentance(Sentance);%Segment Char From the Sentance;
   PreNuralMemberShip = [PreNuralMemberShip,TmpMemberShip];
end; %for IdLine = 1:MaxLineR
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
%================ This Part is part of sub function ============================
function SMemberShip=SegFromSentance(Sentance)
[MaxR,MaxC]=size(Sentance);
MatrixC = SegCFSProj(Sentance);
[MaxRM,MaxCM] = size(MatrixC);
StartC = 1;
EndC   = 2;
SMemberShip = [];
for IdRM = 1:MaxRM
  OnceChar = Sentance(1:MaxR,MatrixC(IdRM,StartC):MatrixC(IdRM,EndC));
  TmpMember=SegFromOnce(OnceChar);
  SMemberShip = [SMemberShip,TmpMember];
end; %for IdCM = 1:MaxCM
%=============================================================================
function CMemberShip=SegFromOnce(Indata)
[MaxDR,MaxDC] = size(Indata);
CMemberShip = [];
for IdCol = 1:MaxDC
 for IdRow = 1:MaxDR
   if Indata(IdRow,IdCol) == 1
     [MinX MinY MaxX MaxY]=SegChar(Indata,IdRow,IdCol,1);
     if ((MaxX-MinX)>=3)&((MaxY-MinY)>=3)
       Char=Indata(MinX:MaxX,MinY:MaxY);
       %imshow(Char);pause;
       NmChar =imresize(Char,[90 90],'nearest');
       NmChar = bwmorph(NmChar,'clean');
       NmChar = bwmorph(NmChar,'fill');
       %imshow(NmChar);pause;
       NmChar = bwmorph(NmChar,'majority');
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        imshow(NmChar);%pause;
        %imshow(NmChar(1:45,:));%pause;
        %imshow(NmChar(46:90,:));%pause;
        %imshow(NmChar(:,1:45));%pause;
        %imshow(NmChar(:,46:90));%pause;
         NmChar = bwmorph(NmChar,'remove');
        %imshow(NmChar);pause;
        NmChar = bwmorph(NmChar,'thin',inf);
        %imshow(NmChar);pause;
        ReturnFuzzy = Neuro_Fuzzy(NmChar);%MemberShip Function for one char;
        CMemberShip = [CMemberShip,ReturnFuzzy];
      end; %if ((MaxX-MinX)>=3)&((MaxY-MinY)>=3)
      Indata(MinX:MaxX,MinY:MaxY) = 0;
    end;  %if Indata(IdRow,IdCol) == 1
  end; %for IdRow = 1:MaxR
end; %for IdCol = 1:MaxC
%=========================end=========================
function OutSeg=SegProjection(data,Inthreshold)
Bkground = 0;
Forground = 1;
StartS = 1; % Index indicate Start Sentance
EndS   = 2; % Index indicate End  Sentance
LineS  = 1; % Index indicate Line  Sentance
if nargin <= 1
  Inthreshold = 15;
end;
ProjectRow = sum(data,2);
[MaxRow,MaxCol]=size(ProjectRow);
%initial Program start at zero
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
for Id=1:MaxRow
  if ProjectRow(Id,1) >= Forground
    IdMean = round(Id/2);
    OutSeg(LineS,StartS) = Id-IdMean;
    break;
  end;
end;
%This start Program at For Ground
Counter = 0;
EndGrammar = 0; % 0 isn't End Sentance and 1 is End Sentance
IdRow = Id;
while IdRow~=MaxRow
    %1. Start at Forground
    for IdRow1=IdRow:MaxRow
      if ProjectRow(IdRow1,1)== Bkground
        break;
      end; %if ProjectRow(IdRow0,1)== Bkground
    end;%for IdRow1=IdRow:MaxRow
    %2. Start at Bkground after break
    for IdRow0=IdRow1:MaxRow
      if IdRow0 == MaxRow % Found Forground
        EndGrammar = 1;  break;
      elseif ProjectRow(IdRow0,1)== Bkground
        Counter = Counter+1;
      elseif ProjectRow(IdRow0,1) >= Forground
        EndGrammar = 0;  break;
      end; %if ProjectRow(IdRow0,1) == Forground
    end;%for IdRow0=IdRow1:MaxRow
    %3. Start after found Forground or EndSentance
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    if EndGrammar == 0
      if Counter >= Inthreshold %Case after that sentance
        IdMean = round(Counter/2);
        OutSeg(LineS,EndS) = IdRow0-IdMean;
        LineS = LineS+1;
        OutSeg(LineS,StartS) = IdRow0-IdMean;
        Counter = 0;
      else %Case after that It is vowel
        Counter = 0;
      end; % if Counter >= threshold
      IdRow = IdRow0; % change Index
    else % EndGrammar == 1
      IdMean  = round(Counter/2);
      OutSeg(LineS,EndS) = IdRow0-IdMean; break;
    end; % EndGrammar == 0
end; %end for IdRow = Id:MaxRow
%=================end=============================
function ReturnFuzzy = Neuro_fuzzy(normal)
  ReturnFuzzy = [];
  TOT = sum(sum(normal));
  UP  = sum(sum(normal(1:45,:)));
  DOWN = sum(sum(normal(46:90,:)));
  LEFT = sum(sum(normal(:,1:45)));
  RIGHT = sum(sum(normal(:,46:90)));
  DW_UP = DOWN/UP;
  RG_LF = RIGHT/LEFT;
  VertProj = sum(normal,1);
  [MaxV,Idcol] = max(VertProj);
  HorProj = sum(normal,2);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
[MaxH,IdRow] = max(HorProj);
%Define Feature of one character;
FeatChar =[TOT UP DOWN LEFT RIGHT MaxH MaxV DW_UP RG_LF]';
%Find Membership of one Char;
FeatChar = memberchar(FeatChar);
%Return Value of the function;
ReturnFuzzy = [ReturnFuzzy,FeatChar];
%=========================================================
function Mbchar = memberchar(x)
% x input vector;
% Y output vector;
if x == 0
  Mbchar == 1;
else
  Mbchar = [];
  y1=evalmf(x,[0 75 150],'trimf');
  y2=evalmf(x,[75 150 225],'trimf');
  y3=evalmf(x,[150 225 300],'trimf');
  y4=evalmf(x,[225 300 375],'trimf');
  y5=evalmf(x,[300 375 450],'trimf');
  y6=evalmf(x,[375 450 525],'trimf');
  y7=evalmf(x,[450 525 600],'trimf');
  y8=evalmf(x,[525 600 675],'trimf');
  y9=evalmf(x,[600 675 750],'trimf');
  Mbchar = [Mbchar,y1,y2,y3,y4,y5,y6,y7,y8,y9];
end;
[maxrow,maxcol]=size(Mbchar);
for id=1:maxrow
  [MMmb(id,1),I(id,1)]= max(Mbchar(id,:));
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
end;
%Mbchar = [x,MMmb,I];
Mbchar = MMmb;
%=========================end=========================
function OutSeg=SegCFSProj(data)
%This Program use for Segment Char From Sentance
%function OutSeg=SegCFSProj(data)
%You have to sent Data that you want  do
%define data
Bkground = 0;
Forground = 1;
StartC = 1; % Index indicate Start Character
EndC   = 2; % Index indicate End   Character
NumberC  = 1; % Index indicate Number of Character
ProjectCol = sum(data,1);
[MaxRow,MaxCol]=size(ProjectCol);
%initial Program start at zero
for Id=1:MaxCol
  if ProjectCol(1,Id) >= Forground
    IdMean = round(Id/2);
    OutSeg(NumberC,StartC) = Id-IdMean;
    break;
  end;
end;
%This start Program at For Ground
Counter = 0;
EndCharacter = 0; % 0 isn't End Character and 1 is End Character
IdCol = Id;
while IdCol~=MaxCol
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
%1. Start at Forground
for IdCol1=IdCol:MaxCol
  if ProjectCol(1,IdCol1)== Bkground
    break;
  end; %if ProjectCol(1,IdCol1)== Bkground
end;%for IdCol1=IdCol:MaxCol
%2. Start at Bkground after break
for IdCol0=IdCol1:MaxCol
  if IdCol0 == MaxCol % Found Forground
    EndCharacter = 1;  break;
  elseif ProjectCol(1,IdCol0)== Bkground
    Counter = Counter+1;
  elseif ProjectCol(1,IdCol0) >= Forground
    EndCharacter = 0;  break;
  end; %if ProjectCol(IdCol0,1) == Forground
end;%for IdCol0=IdCol1:MaxCol
%3. Start after found Forground or EndSentance
if EndCharacter == 0
  IdMean = round(Counter/2);
  OutSeg(NumberC,EndC)=IdCol0-IdMean;
  NumberC = NumberC + 1;
  OutSeg(NumberC,StartC)=IdCol0-IdMean;
  Counter = 0;
  IdCol = IdCol0; % change Index
else % EndCharacter == 1
  IdMean  = round(Counter/2);
  OutSeg(NumberC,EndC) = IdCol0-IdMean; break;
end; % EndCharacter == 0
end;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
function [MinX,MinY,MaxX,MaxY]=SegChar(data,X,Y,ChkC)
% Program use for Segment Character from the palagraph
% function [MinX,MinY,MaxX,MaxY]=SegChar(data,X,Y,ChkC)
% We have to receive data of Image and Point start X,Y and Check for convert
if nargin <= 3
  ChkC = 1; %Bkground is 0
end;
ix=X;   iy=Y;
jx=ix-1; jy=iy;
d=1;
bx=jx-ix; by=jy-iy;
round=1;
jx=ix;  jy=iy;
MinX=X; MaxX=X;
MinY=Y; MaxY=Y;
while d<=8
  dx=bx+by; dy=by-bx;
  if(dx~=0)
    dx=dx/abs(dx);
  end;
  if(dy~=0)
    dy=dy/abs(dy);
  end;
  ix=jx+dx; iy=jy+dy;
  if (data(ix,iy)==uint8(logical(ChkC)))
    if (MinX>ix) MinX=ix;   end;
    if (MinY>iy) MinY=iy;   end;
    if (MaxX<ix) MaxX=ix;   end;
    if (MaxY<iy) MaxY=iy;   end;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    if (ix==X)&(iy==Y)&(round==2)
      break;
    else
      if (ix==X)&(iy==Y)&(round==1) round=2; end;
      bx=jx-ix; by=jy-iy;
      jx=ix;   jy=iy;
      d=1;
    end;
  else
    bx=dx; by=dy; d=d+1;
  end;
end;
%=============================end=============================
function A = imfilter(B,threshold)
if nargin <= 1
  threshold = 5;
end;
A = B;
k = length(A(:,1));
t = threshold; %input('Enter threshold value : ');
dk = 0;
for i=1:k-2;
  for j=1:k-2;
    a=B(i:(i+2),j:(j+2));
    dk=sum(sum(a));
    if dk > t+2;
      A(i+1,j+1) = 1;
    elseif dk < t-2;
      A(i+1,j+1) = 0;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    end
  end
end
%=========================end=========================
function OutReverted=ToRevert(InValue)
%This Program use for Invert Data
TmpValue = double(InValue);
TmpValue = TmpValue-1;
TmpValue = TmpValue.*(-1);
OutReverted = uint8(logical(TmpValue));
%=========================end=========================
```

2. การฝึกสอน

```
function Learnfirst()
%Load data Files of font;
load d:\project1\temp\alice0;
%Define parameters;
OP = eye(46);
%Initial Neural Network
P = alc014; %Input Vectore;
T = OP;              %target vector;
S1 = 60;%Define Hidden layer;
[W1,b1,W2,b2]=initff(P,S1,'tansig',T,'tansig');
W2 = W2*0.01;
b2 = b2*0.01;
save d:\project1\weight\wbfrst W1 b1 W2 b2;
Ep = 25;                %Epochs between updating display
MnE = 100000;  %Maximum number of epochs to train
Sser = 0.02;        %Sum-squared error goal
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Lr = 0.01;          %Learning rate
LI = 1.05;          %Learning rate increase
LD = 0.7;           %Learning rate decrease
MmC = 0.9;          %Momentum constant
MerRto = 1.04;      %Maximum error ratio
P = alc014;         %Matrix of input vectors
T = OP;             %Matrix of Target vectors
TP = [Ep MnE Sser NaN NaN NaN MmC MerRto];    %Training parameters vectors
[W1,b1,W2,b2,TE,TR]=trainbpx(W1,b1,'tansig',W2,b2,'tansig',P,T,TP);
save d:\project1\weight\Wbfrst W1 b1 W2 b2;
%TE - the actual number of epochs trained.
%TR - training record: [row of errors]
Ep = 100;
TP = [Ep MnE Sser NaN NaN NaN MmC MerRto];    %Training parameters vectors
[W1,b1,W2,b2,TE,TR]=trainbpx(W1,b1,'tansig',W2,b2,'tansig',P,T,TP);
save d:\project1\weight\Wbfrst W1 b1 W2 b2;
%TE - the actual number of epochs trained.
%TR - training record: [row of errors]
%=============================end=============================
function Learning()
%Load data Files of font;
load d:\project1\temp\alice0;
load d:\project1\temp\alice1;
load d:\project1\temp\alice2;
load d:\project1\temp\alice3;
load d:\project1\temp\alice4;
load d:\project1\temp\alice5;
load d:\project1\temp\cht;
load d:\project1\temp\sset;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
load d:\project1\weight\wbfrst;
%Define parameters;
OP = eye(46);
P0 = [alc014 alc016 alc018 alc020];
P1 = [alc114 alc116 alc118 alc120];
P2 = [alc214 alc216 alc218 alc220];
P3 = [alc314 alc316 alc318 alc320];
P4 = [alc414 alc416 alc418 alc420];
P5 = [alc514 alc516 alc518 alc520];
P6 = [cht14 cht16 cht18 cht20];
P7 = [sst14 sst16 sst18 sst20];
tg = [OP OP OP OP];
Ep = 20;              %Epochs between updating display
MnE = 150000;  %Maximum number of epochs to train
Sser = 0.01;       %Sum-squared error goal
Lr = 0.1;           %Learning rate
LI = 1.05;          %Learning rate increase
LD = 0.7;          %Learning rate decrease
MmC = 0.9;        %Momentum constant
MerRto = 1.04;  %Maximum error ratio
%Learning each font;
P = P0;
T = tg;
TP = [Ep MnE Sser NaN NaN NaN MmC MerRto];
[W1,b1,W2,b2,TE,TR]=trainbpx(W1,b1,'tansig',W2,b2,'tansig',P,T,TP);
save d:\project1\weight\WbP0 W1 b1 W2 b2;
P = P1;
T = tg;
[W1,b1,W2,b2,TE,TR]=trainbpx(W1,b1,'tansig',W2,b2,'tansig',P,T,TP);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
save d:\project1\weight\WbP1 W1 b1 W2 b2;
P = P2;
T = tg;
[W1,b1,W2,b2,TE,TR]=trainbpx(W1,b1,'tansig',W2,b2,'tansig',P,T,TP);
save d:\project1\weight\WbP2 W1 b1 W2 b2;
P = P3;
T = tg;
[W1,b1,W2,b2,TE,TR]=trainbpx(W1,b1,'tansig',W2,b2,'tansig',P,T,TP);
save d:\project1\weight\WbP3 W1 b1 W2 b2;
P = P4;
T = tg;
[W1,b1,W2,b2,TE,TR]=trainbpx(W1,b1,'tansig',W2,b2,'tansig',P,T,TP);
save d:\project1\weight\WbP4 W1 b1 W2 b2;
P = P5;
T = tg;
[W1,b1,W2,b2,TE,TR]=trainbpx(W1,b1,'tansig',W2,b2,'tansig',P,T,TP);
save d:\project1\weight\WbP5 W1 b1 W2 b2;
P = P6;
T = tg;
[W1,b1,W2,b2,TE,TR]=trainbpx(W1,b1,'tansig',W2,b2,'tansig',P,T,TP);
save d:\project1\weight\WbP6 W1 b1 W2 b2;
P = P7;
T = tg;
[W1,b1,W2,b2,TE,TR]=trainbpx(W1,b1,'tansig',W2,b2,'tansig',P,T,TP);
save d:\project1\weight\WbP7 W1 b1 W2 b2;
%==========================end==========================
function Learngroup()
%Load data Files of font;
load d:\project1\temp\alice0;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
load d:\project1\temp\alice1;
load d:\project1\temp\alice2;
load d:\project1\temp\alice3;
load d:\project1\temp\alice4;
load d:\project1\temp\alice5;
load d:\project1\temp\cht;
load d:\project1\temp\sset;
load d:\project1\weight\wbfrst;
%Define parameters;
OP = eye(46);
P0 = [alc014 alc016 alc018 alc020];
P1 = [alc114 alc116 alc118 alc120];
P2 = [alc214 alc216 alc218 alc220];
P3 = [alc314 alc316 alc318 alc320];
P4 = [alc414 alc416 alc418 alc420];
P5 = [alc514 alc516 alc518 alc520];
P6 = [cht14 cht16 cht18 cht20];
P7 = [sst14 sst16 sst18 sst20];
tg = [OP OP OP OP];
Ep = 20;                %Epochs between updating display
MnE = 100000;  %Maximum number of epochs to train
Sser = 0.1;   %Sum-squared error goal
Lr = 0.01;          %Learning rate
LI = 1.05;          %Learning rate increase
LD = 0.7;           %Learning rate decrease
MmC = 0.9;          %Momentum constant
MerRto = 1.04;  %Maximum error ratio
P = [P0 P1 P2 P3];    %Matrix of input vectors
T = [tg tg tg tg]; %Matrix of Target vectors
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
TP = [Ep MnE Sser NaN NaN NaN MmC MerRto];    %Training parameters vectors
[W1,b1,W2,b2,TE,TR]=trainbpx(W1,b1,'tansig',W2,b2,'tansig',P,T,TP);
save d:\project1\weight\Wbgr1 W1 b1 W2 b2;
%TE - the actual number of epochs trained.
%TR - training record: [row of errors]
P = [P4 P5 P6 P7];
T = [tg tg tg tg];
TP = [Ep MnE Sser NaN NaN NaN MmC MerRto];    %Training parameters vectors
[W1,b1,W2,b2,TE,TR]=trainbpx(W1,b1,'tansig',W2,b2,'tansig',P,T,TP);
save d:\project1\weight\Wbgr2 W1 b1 W2 b2;
%=========================================================
function LearnAll()
%Load data Files of font;
load d:\project1\temp\alice0;
load d:\project1\temp\alice1;
load d:\project1\temp\alice2;
load d:\project1\temp\alice3;
load d:\project1\temp\alice4;
load d:\project1\temp\alice5;
load d:\project1\temp\cht;
load d:\project1\temp\sset;
load d:\project1\weight\wbfrst;
%Define parameters;
OP = eye(46);
P0 = [alc014 alc016 alc018 alc020];
P1 = [alc114 alc116 alc118 alc120];
P2 = [alc214 alc216 alc218 alc220];
P3 = [alc314 alc316 alc318 alc320];
P4 = [alc414 alc416 alc418 alc420];
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
P5 = [alc514 alc516 alc518 alc520];
P6 = [cht14 cht16 cht18 cht20];
P7 = [sst14 sst16 sst18 sst20];
tg = [OP OP OP OP];
Ep = 20;                %Epochs between updating display
MnE = 100000;  %Maximum number of epochs to train
Sser = 0.1;   %Sum-squared error goal
Lr = 0.01;          %Learning rate
LI = 1.05;          %Learning rate increase
LD = 0.7;           %Learning rate decrease
MmC = 0.9;          %Momentum constant
MerRto = 1.04;  %Maximum error ratio
P = [P0 P1 P2 P3 P4 P5 P6 P7];   %Matrix of input vectors
T = [tg tg tg tg tg tg tg tg];     %Matrix of Target vectors
TP = [Ep MnE Sser NaN NaN NaN MmC MerRto];     %Training parameters vectors
[W1,b1,W2,b2,TE,TR]=trainbpx(W1,b1,'tansig',W2,b2,'tansig',P,T,TP);
save d:\project1\weight\Wball W1 b1 W2 b2;
%TE - the actual number of epochs trained.
%TR - training record: [row of errors]
%=============================end=================================
```

3. การทดสอบ

```
function TestNF(InFileName)
load d:\project1\weight\Wbp0;%Load Weight Vector from path d:\weight;
WC1 = W1;
BC1 = b1;
WC2 = W2;
BC2 = b2;
if nargin <= 0
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  InFileName = input('Type File Name that you want > ','s');
end;
recogn = recognNF(InFileName);
A = simuff(recogn,WC1,BC1,'tansig',WC2,BC2,'tansig');
[Mrow,Mcol] = size(A);
B = compet(A);
err = 0;
real = 0;
percentError = 0;
percentreal = 0;
for j=1:Mcol
  i = j;
  if B(i,j)==1
    real = real+1;
  else
    err = err+1;
  end;
end;
ptError = (err/Mcol)*100;
ptreal = (real/Mcol)*100;
fprintf('\n1. Error char is =');disp(err);
fprintf('\tpercentage =');disp(ptError);
fprintf('\n2. Real char is =');disp(real);
fprintf('\tPercentage =');disp(ptreal);
% ========================end ========================
function ValRecogn = RecognNF(InFileName)
if nargin <= 0
  InFileName = input('Type File Name that you want > ','s');
end;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
InC=strcat(InFileName,'.bmp');
data=imread(InC);
[MaxRow,MaxColume]=size(data);
%1. Before Segment Character from palagraph we have to check something
if (data(1,1)==1)&(data(MaxRow,MaxColume)==1)
   data = ToRevert(data);
end;
imshow(data);%pause;
clc;
%2. Find Line of the Grammar
LineS = SegProjection(data,15);
%3. Segment Sentance from the Grammar
[MaxLineR,MaxLineC]=size(LineS);
[MaxDR,MaxDC]=size(data);
StartS = 1;
EndS   = 2;
ValRecogn = [];
for IdLine = 1:MaxLineR
  StartL = LineS(IdLine,StartS);
  EndL   = LineS(IdLine,EndS);
  Sentance = data(StartL:EndL,1:MaxDC);
  Sentance = imfilter(Sentance,5);%Remove noise out of image;
  %imshow(Sentance);%pause;
  TmpMemberShip=SegFromSentance(Sentance);%Segment Char From the Sentance;
  ValRecogn = [ValRecogn,TmpMemberShip];
end; %for IdLine = 1:MaxLineR
%================= This Part is part of sub function ==========================
function SMemberShip=SegFromSentance(Sentance)
[MaxR,MaxC]=size(Sentance);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
MatrixC = SegCFSProj(Sentance);
[MaxRM,MaxCM] = size(MatrixC);
StartC = 1;
EndC   = 2;
SMemberShip = [];
for IdRM = 1:MaxRM
  OnceChar = Sentance(1:MaxR,MatrixC(IdRM,StartC):MatrixC(IdRM,EndC));
  TmpMember=SegFromOnce(OnceChar);
  SMemberShip = [SMemberShip,TmpMember];
end; %for IdCM = 1:MaxCM
%=========================================================
function CMemberShip=SegFromOnce(Indata)
[MaxDR,MaxDC] = size(Indata);
CMemberShip = [];
for IdCol = 1:MaxDC
  for IdRow = 1:MaxDR
    if Indata(IdRow,IdCol) == 1
      [MinX MinY MaxX MaxY]=SegChar(Indata,IdRow,IdCol,1);
      if ((MaxX-MinX)>=3)&((MaxY-MinY)>=3)
        Char=Indata(MinX:MaxX,MinY:MaxY);
        %imshow(Char);pause;
        NmChar =imresize(Char,[90 90],'nearest');
        NmChar = bwmorph(NmChar,'clean');
        NmChar = bwmorph(NmChar,'fill');
        %imshow(NmChar);pause;
        NmChar = bwmorph(NmChar,'majority');
        imshow(NmChar);%pause;
        %imshow(NmChar(1:45,:));%pause;
        %imshow(NmChar(46:90,:));%pause;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        %imshow(NmChar(:,1:45));%pause;
        %imshow(NmChar(:,46:90));%pause;
         NmChar = bwmorph(NmChar,'remove');
        %imshow(NmChar);%pause;
        NmChar = bwmorph(NmChar,'thin',inf);
        %imshow(NmChar);pause;
        ReturnFuzzy = Neuro_Fuzzy(NmChar);%Membership Function for one char;
        %Output index of char;
        TmP = SimNet(ReturnFuzzy);
        fprintf('Output = %s \n',TmP);%pause;
        CMemberShip = [CMemberShip,ReturnFuzzy];
      end; %if ((MaxX-MinX)>=3)&((MaxY-MinY)>=3)
      Indata(MinX:MaxX,MinY:MaxY) = 0;
    end;  %if Indata(IdRow,IdCol) == 1
  end; %for IdRow = 1:MaxR
end; %for IdCol = 1:MaxC
%=================================end=================================
function OutPut=SimNet(InUnknow)
%This Program use for test system
%function OutPut=SimNet(Unknow,targets,W1,b1,W2,b2)
%OutPut is Pattern that you
load d:\project1\weight\Wbp0;%Load Weight Vector from path d:\weight;
WC1 = W1;
BC1 = b1;
WC2 = W2;
BC2 = b2;
A = simuff(InUnknow,WC1,BC1,'tansig',WC2,BC2,'tansig');
AA = compet(A);
if AA(1,1) == 1
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
 OutPut = '1';
elseif AA(2,1) == 1
 OutPut = '2';
elseif AA(3,1) == 1
 OutPut = '3';
elseif AA(4,1) == 1
 OutPut = '4';
elseif AA(5,1) == 1
 OutPut = '5';
elseif AA(6,1) == 1
 OutPut = '6';
elseif AA(7,1) == 1
 OutPut = '7';
elseif AA(8,1) == 1
 OutPut = '8';
elseif AA(9,1) == 1
 OutPut = '9';
elseif AA(10,1) == 1
 OutPut = '10';
elseif AA(11,1) == 1
 OutPut = '11';
elseif AA(12,1) == 1
 OutPut = '12';
elseif AA(13,1) == 1
 OutPut = '13';
elseif AA(14,1) == 1
 OutPut = '14';
elseif AA(15,1) == 1
 OutPut = '15';
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
elseif AA(16,1) == 1
  OutPut = '16';
elseif AA(17,1) == 1
  OutPut = '17';
elseif AA(18,1) == 1
  OutPut = '18';
elseif AA(19,1) == 1
  OutPut = '19';
elseif AA(20,1) == 1
  OutPut = '20';
elseif AA(21,1) == 1
  OutPut = '21';
elseif AA(22,1) == 1
  OutPut = '22';
elseif AA(23,1) == 1
  OutPut = '23';
elseif AA(24,1) == 1
  OutPut = '24';
elseif AA(25,1) == 1
  OutPut = '25';
elseif AA(26,1) == 1
  OutPut = '26';
elseif AA(27,1) == 1
  OutPut = '27';
elseif AA(28,1) == 1
  OutPut = '28';
elseif AA(29,1) == 1
  OutPut = '29';
elseif AA(30,1) == 1
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
 OutPut = '30';
elseif AA(31,1) == 1
 OutPut = '31';
elseif AA(32,1) == 1
 OutPut = '32';
elseif AA(33,1) == 1
 OutPut = '33';
elseif AA(34,1) == 1
 OutPut = '34';
elseif AA(35,1) == 1
 OutPut = '35';
elseif AA(36,1) == 1
 OutPut = '36';
elseif AA(37,1) == 1
 OutPut = '37';
elseif AA(38,1) == 1
 OutPut = '38';
elseif AA(39,1) == 1
 OutPut = '39';
elseif AA(40,1) == 1
 OutPut = '40';
elseif AA(41,1) == 1
 OutPut = '41';
elseif AA(42,1) == 1
 OutPut = '42';
elseif AA(43,1) == 1
 OutPut = '43';
elseif AA(44,1) == 1
 OutPut = '44';
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
elseif AA(45,1) == 1
  OutPut = '45';
elseif AA(46,1) == 1
  OutPut = '46';
end;
%==========end==========
```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

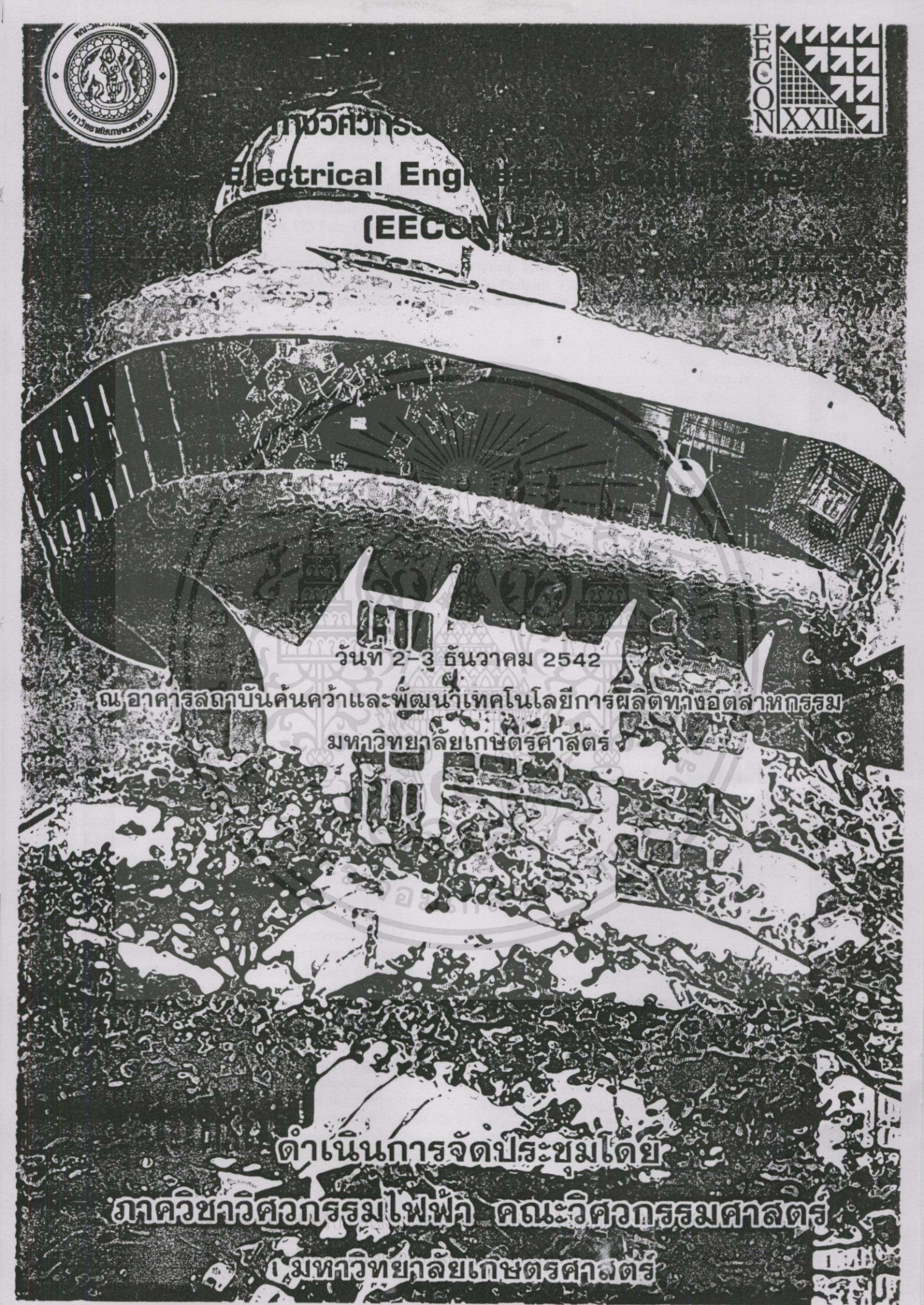
ภาคผนวก ง

ผลงานวิจัยที่ได้รับการตีพิมพ์

ผลงานวิจัยเรื่อง "RECOGNITION OF HAND PRINTED THAI CHARACTERS USING KARHUNEN – LOVE TRANSFORM AND FUZZY – NEURAL NETWORKS " การประชุมทางวิชาการวิศวกรรมไฟฟ้าครั้งที่ 22 (EECON - 22) จัดประชุมโดย ภาควิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเกษตรศาสตร

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


EECON XXII
Electrical Eng
(EECO
วันที่ 2-3 ธันวาคม 2542
ณ อาคารสถาบันค้นคว้าและพัฒนาเทคโนโลยีการผลิตทางอุตสาหกรรม
มหาวิทยาลัยเกษตรศาสตร์
ดำเนินการจัดประชุมโดย
ภาควิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์
มหาวิทยาลัยเกษตรศาสตร์

Recognition of Handprinted Thai Characters Using Karhunen-Loeve Transform and Fuzzy-Neural Networks

Vixinh Moungsinh*, Pisit Phokharatkul** and Chom Kimpan*
* Faculty of Engineering, Research Center of Communication and Information Technology,
King Mongkut's Institute of Technology Landkrabang, Bangkok, 10520, Thailand.
Tel. 3269969 ext 125 Fax. Ext 127, Email: s0060094@kmitl.ac.th, kkchom@kmitl.ac.th

** Department of Electrical Engineering, Faculty of Engineering,
Mahidol University Salaya, Nakhon Pathom, 73170, Thailand.
Tel.4419997-8 ext.6502, Fax.4419731, Email: s0060004@kmitl.ac.th

Abstract

This paper proposes the method of handprinted Thai Characters recognition using Karhunen-Loeve transform and fuzzy-neural networks. The recognition system is separated into three stages. The rough classification stage, the Thai characters are separated into some cluster domains by similarity method. The characters in each category of rough classification are tranformed by the Karhunen-Loeve transform. Eigenvector which gived maximum eigenvalues to be a standard pattern of each category. The fine classification stage, the fuzzy IF-THEN rules are used to discriminate between the characters. Finally, the fuzzy IF-THEN rules are trained in neural network for recognition of the characters. Then the success fully trained networks (in term of learning a sufficient accuracy) can be used neural weight instead of using fuzzy rules base as the traditional fuzzy system. The result of this approach indicated that the system could successfully recognized 96.2 %.

1. Introduction

On the research of the topic of Thai characters in the past seventeen years, there are many approaches have been tried to make a computer recognized the characters. The preliminary stage of this field, which concern of the use of matching method was described by Kimpan, Itoh and Kawanishi [1]. Subsequently, the topological property [2] are used to improve the recognition accuracy. In 1987, the same author proposes the recognition method of printed Thai characters using Karhuñen-Loeve expansion [3] was reported. Although, the recognition rate of this proposed method was satisfactory, but it used linear decision for discriminate the characters in a category, which have the problem of the discrimination if the score plots has several members. Hiranvanichakorn, Agui and Nakajima used local features of Thai characters, such as convexity and concavity of stoke, in their recognition system [4-7]. Airpaiboons et al. [8,9] used the head of Thai characters to recognize the handwritten Thai characters. In 1998, Phokharatkul and Kimpan [10] used the cavity features of each character base on neural networks for solving the problem in the recognition by considering the head of the Thai character.

In this research, we proposed the method of recognition of handprinted Thai characters by the Karhunen-Loeve transform and fuzzy-neural networks for solving the problem of the discrimination in the previous paper [3]. Furthermore, this method can be recognizes the handprinted characters of various scripts and to have no problem about of the head of character and slant written or ornamental written. The recognition procedure will be illustrated in the following section.

2. Characteristic of data

The Thai alphabet consists of 44 consonant Characters, 14 vowel characters, 8 Tonal symbols and 10 Thai numerals, as shown in Figure 1. The Thai language sentences composed of consonants, vowels and control voice tones on different levels.

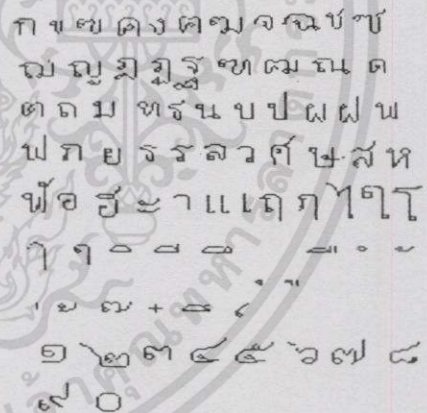


Figure 1. Thai alphabets

3. Rough Classification

In the rough classification stage, fine features are ignores by blurring. At first, to creating binary blurred pattern from each blurred pattern by computing histogram of blurred patterns, then setup threshold value to be 15, that if its histogram equal or greater than 15 to be binary 1 other to be binary 0 as shown in figure 2. The binary blurred patterns form this stage are stored in memory to be the data base, which we used comparing with the unknown blurred patterns for clustering. The blurred characters are separated into some cluster

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

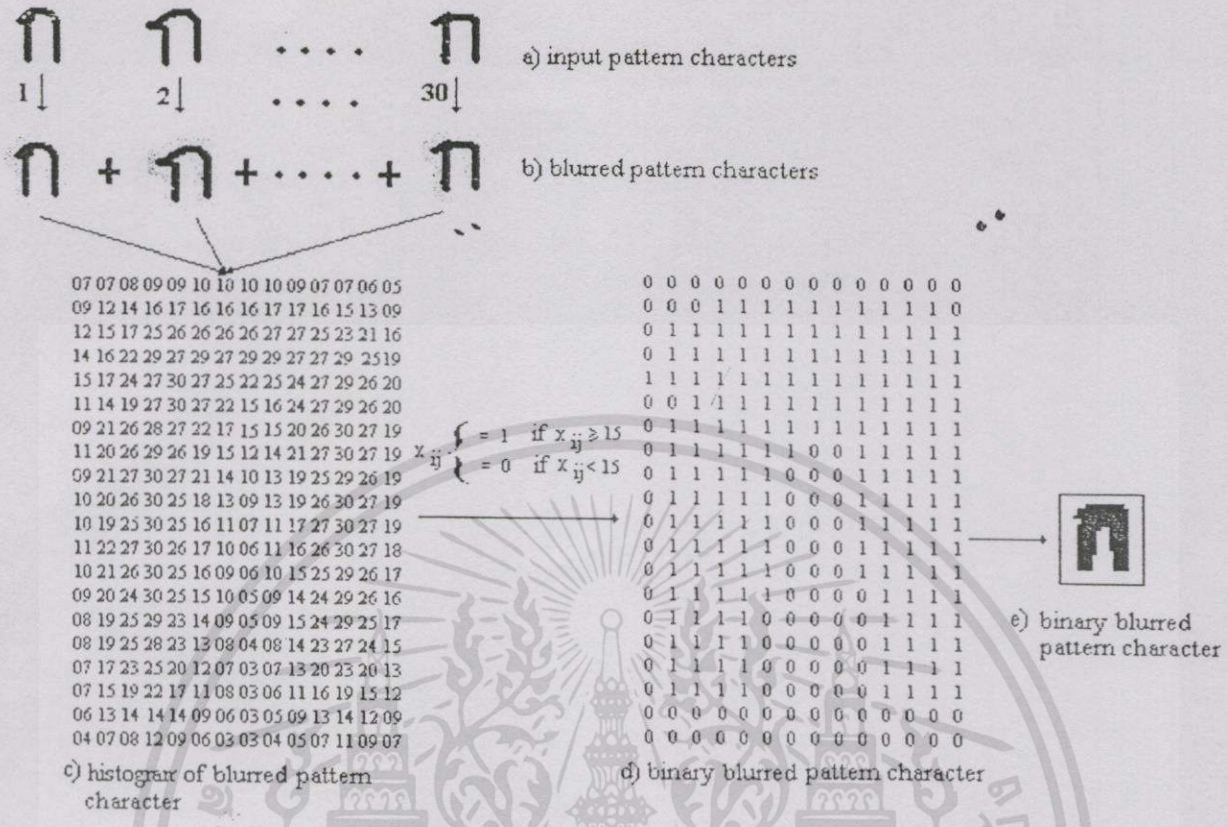


Figure 2. Construction process of standard blurred pattern characters of handprinted Thai characters "ก".

domains. The clustering criterion used in based on selection of the patterns by measuring the similarity coefficient in the reference [1]. Then the preclustering of characters in determined by setting the criterion decision value (threshold value) form the value of the similarity coefficient.

Some characters will be members of more than one cluster. Such as clusters are reassigned by investigating the width and height of characters. Characters having nearly the same width and height should be members of the same category ('category' is group derived by clustering. category consist of class of characters, and a 'class' is a set associated with the same character).

The eigenvalues and eigenvectors in the Karhunen-Loeve transform of each category are computed by the same method in the reference [3]. The eigenvector having the maximum eigenvalue, which constitute the principal component in each category, are used as a standard pattern for matching with unknown patterns. For categories, which have only one member, this stage is the final recognition.

4. Fine classification by fuzzy IF-THEN rules on eigenvector space

In the fine classification stage, in order to discriminate the characters in category, It is necessary to establish some rules upon which to base decisions concerning the class membership of the pattern characters which is confronted. One important approach to this problem is the use of fuzzy IF-THEN rules on eigenvector space to discriminate between the characters. The fuzzy classifier considered here is based on the technique of distributed fuzzy IF-THEN rules [11], where grid-type fuzzy partitions on the pattern space are used. This approach simultaneously utilizes the fuzzy rules corresponding to coarse fuzzy partitions as well as those corresponding to fine fuzzy partitions. Assume that a pattern space is the unit square $[0,1]\times[0,1]$. Suppose that m patterns $x_p=(x_{1p}, x_{2p})$, $p = 1,2,\ldots,m$, are given as training patterns from M classes $C_1, C_2, \ldots, C_M$. That is the classification of each x_p is known as one of M classes. The classification problem is to generate fuzzy IF-THEN rules that divide the pattern space into M disjoint decision areas. For this problem, the following fuzzy rule is employed:

$$R_j^k : IF\ x_{1p}\ is\ A_i^K\ AND\ x_{2p}\ is\ A_j^K,$$
$$THEN\ x_p\ belongs\ to\ G_{ij}^k\ with\ CF = CF_{ij}^k.$$

Where K is the number of fuzzy subspaces (intervals) in each axis. R_{ij}^K is the label of the fuzzy rule, G_{ij}^k is the consequent (ie., one of M classes), DF $\in [0,1]$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

is the grade of certainty of the fuzzy IF-THEN rule, and A_i^K and A_j^K are fuzzy sets with triangular membership function $\mu_i^K(x_{1p})$ and $\mu_j^K(x_{2p})$ defined by

$$\mu_i^K(x) = \mu_{A_i^K}(x) = \max\left\{1 - \frac{|x - a_i^K|}{b^K}, 0\right\}, \quad i = 1,2,\ldots K. \tag{1}$$

Where

$$a_i^K = \frac{i-1}{K-1} \quad and \quad b^K = \frac{1}{K-1}, \quad i = 1,2,\ldots K. \tag{2}$$

In the membership function of Eq. (1), a_i^K is the center where the grade of membership is equal to 1 and b^K is the spread of the membership function. The fuzzy partitions for K=2 are shown in Figure 3, where the labels of fuzzy rules are indicated in the corresponding fuzzy subspaces.

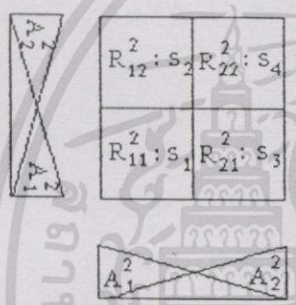


Figure 3. Show the labels of distributed fuzzy IF-THEN Rules for K=2.

Let us consider the example in the case of a category, which has four classes as shown in Fig. 4. It is seen in the figure that the four pattern populations can be conveniently separated by the boundaries of these classes, which it is generated by fuzzy IF -THEN rules on eigenvector space. Therefore, the fuzzy classification is important discriminate the characters in a category and recognized them. The curve of boundary can be used as a decision (or discriminating) function since, given a pattern x of unknown classification, we may say that x belong to Class # If x is in class # boundary, or to class * if x is in class * boundary.

5. Neural networks and fuzzy IF-THEN rules

The learning stage, we used neural networks for learning fuzzy IF-THEN rules of the recognition system. The system works compose of two distinct modes: Training and application. The difference between system is fuzzy neural system use neural weights instead of using fuzzy IF-THEN rules as in traditional fuzzy system.

6. The experimental results

We have implemented the algorithm in Matlab on a 32-bit personal computer. Karhunen-Loeve transform was used transform data, and fuzzy logic techniques were used to cluster the data and construct the features, which are not obvious. Feedforward neural networks with the backpropagation learning algorithms were use in the learning and classification of the handprinted Thai characters in each category. The experimental results are shown in table 1. The comparison of recognition rate of Karhunen-Loeve transform and fuzzy-neural networks with other methods are shown in table 2. From table 2, the recognized rate of loop (or head) structure method is very high, but the loops of the characters were defect such as broken head or solid head this particular method can not be use, which in the Karhunen-loeve fuzzy neural to have no problem.

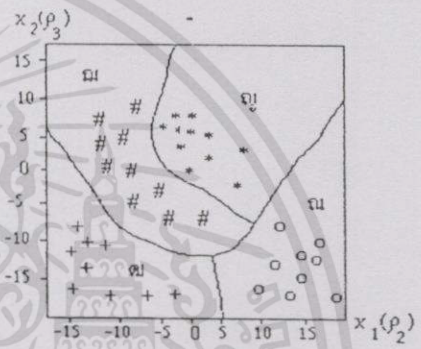


Figure 4. Score plot of four member categories.

7. Conclusion

From the score plot on eigenvector space of handprinted Thai characters, the boundary curve was drawn in the K-L domain. In Figure 4, we can see that the nonlinear boundary curve was derived from the given fuzzy vector by the fuzzy IF-THENif rules, which in the previous paper [3] can not be drawn. If the result of score plot on eigenvector space is clearly distributed, the

Table 1. Results of testing the handprinted Thai character using Karhunen-Loeve fuzzy neural method.

Sample	Recognized	Miss-Recognized
2000	96.2	3.8

Table 2. The comparison of the recognition rate of the Karhunen-Loeve fuzzy neural with other methods.

Recognition method	Percent correct (%)
Local features [6]	88.9
Cavity feature & neural net [10]	98.3
Loop structure [9]	99.0
Karhunen-loeve fuzzy neural net	96.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

classification can be constructed by using fuzzy IF-THEN rules, and then the classification condition can be given definitely.

The recognition method of Thai character using Karhunen-Loeve transform [3] depend on the performance of fuzzy classification system based on fuzzy IF-THEN rules. If it has many fuzzy IF-THEN rules, the system consumed too much computation time. Then, this work need reduced the computation time, and fuzzy neural networks can be applied for this target, because fuzzy-neural network uses neural weights to discriminate the characters in each category.

References

[1.] C.Kimpan, A.Ttoh, and K.Kawanishi, "Recognition of printed Thai character using a matching method", IEE Proc., vol.130, Pt.E, No.6, pp. 183-188, Nov.1983.

[2.] C.Kimpan, "Printed Thai character recognition using topological properties method." INT.J. Electronics, vol.60, No.3, pp. 303-329, 1986.

[3.] C.Kimpan, A.Itoh., and K.Kawanishi, "Fine classification of printed Thai character recognition using the Karhunen-Loeve expansion", Proc.IEE, vol.134,Pt.E.,No.5, pp. 257-264, Sept. 1987.

[4.] P.Hirananichakorn., T.Agui,and M.Nakajima, "A recognition method of Thai characters." IECE Trans. Vol.E65, No. 12, pp. 737-744, Dec.1982.

[5.] P.Hirananichakorn.., T. Agui,and M.Nakajima, "Recognition method of Thai characters by using local features." Trans. IECE, vol. E67, No. 8, pp. 425-432, Aug.1984

[6.] P.Hirananichakorn, T. Agui, and M.Nakajima, "A recognition method of handprinted Thai characters by local features." IECE trans., vol.E68, No. 2, pp. 83-90, Feb. 1985.

[7.] P.Hirananichakorn., T. Agui,and M.Nakajima, "An online recognition method of Thai characters", IECE Trans. Vol.E68, No. 11, pp. 594-601, Nov 1985.

[8.] S.Airphaiboon, M.Sangworasil, B.Attachoo, and C. Kimpan, "Recognition of handwritten Thai character considering the head of character." IEEE International Conference on Image Processing. Vol.2, Singapore, pp. 326-330, 1989.

[9.] S.Airphaiboon and S.Kondo, "Recognition of handprinted Thai Characters Using Loop structures." IEICE Trans. INF.& SYST., vol.E79-D,No.9, pp. 1296-1304, Sep.1996.

[10.] P.Phokharatkul and C.Kimpan, "Recognition of handprinted Thai characters Using the cavity features of character based on neural network" IEEE International Conference on 1998, pp.149-152.

[11.] H.Ishibuchi, R.Fujioka and H.Tanaka."Neural Networks that Learn from Fuzzy IF-THEN Rules." IEEE Trans. On Fuzzy systems. Vol.1, No.2, pp. 85-97, May 1993.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประวัติผู้เขียน

นายวิชิน มุ่งสิน เกิดเมื่อวันที่ 6 กรกฎาคม 2502 ที่แขวงจำปาสักดิ์ สาธารณรัฐประชาธิปไตย ประชาชนลาว สำเร็จการศึกษาจากมหาวิทยาลัยสร้างครูเวียงจันทน์ สาขาคณิตศาสตร์และฟิซิกศาสตร์ ปีการศึกษา พ.ศ. 2525

ปี พ.ศ. 2526 เข้ารับราชการในตำแหน่งอาจารย์ สังกัดมหาวิทยาลัยสร้างครูเวียงจันทน์ กระทรวงศึกษาธิการ และ ปี พ.ศ. 2536 สังกัดมหาวิทยาลัยแห่งชาติ มาถึงปัจจุบัน

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้