การรู้จำตัวอักษรลายมือเขียนภาษาไทยด้วยวิธีซินแทกติก ร่วมกับแบบจำลองฟัซซี-นิวรอล

THAI ALPHABET HANDWRITING RECOGNITION USING SYNTACTIC METHOD AND FUZZY-NEURAL MODEL

อลิศรา หงษ์พร้อมญาติ
ALISSARA HONGPROMYART

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต
สาขาวิชาคณิตศาสตร์ประยุกต์
บัณฑิตวิทยาลัย
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
พ.ศ. 2546

ISBN 974-324-644-4

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

การรู้จำตัวอักษรลายมือเขียนภาษาไทยด้วยวิธีซินแทกติก ร่วมกับแบบจำลองฟัซซี – นิวรอล

THAI ALPHABET HANDWRITING RECOGNITION USING SYNTACTIC METHOD AND FUZZY – NEURAL MODEL

อลิศรา หงษ์พร้อมญาติ

ALISSARA HONGPROMYART

เลขหมู่.....
เลขทะเบียน 47704
วัน, เดือน, ปี 22 ส.ค. 2548

.b.....
.i.....

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต

สาขาวิชาคณิตศาสตร์ประยุกต์

บัณฑิตวิทยาลัย

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ.2546

ISBN 974 – 324 – 644 - 4

THAI ALPHABET HANDWRITING RECOGNITION USING SYNTACTIC METHOD AND FUZZY – NEURAL MODEL

ALISSARA HONGPROMYART

A THESIS SUBMITTED IN PARTIAL FULFILLMENT

OF THE REQUIREMENT FOR THE DEGREE OF

MASTER OF SCIENCE IN APPLIED MATHEMATICS

SCHOOL OF GRADUATE STUDIES

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

2003

ISBN 974 – 324 – 644 - 4

COPYRIGHT 2003

SCHOOL OF GRADUATE STUDIES

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

บัณฑิตวิทยาลัย

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ใบรับรองวิทยานิพนธ์

หัวข้อวิทยานิพนธ์ การรู้จำตัวอักษรลายมือเขียนภาษาไทยด้วยวิธีซินแทกติกร่วมกับแบบจำลองฟัซซี-นิวรอล

THAI ALPHABET HANDWRITING RECOGNITION USING SYNTACTIC METHOD AND FUZZY – NEURAL MODEL

ชื่อนักศึกษา นางสาวอลิศรา หงษ์พร้อมญาติ

รหัสประจำตัว 42065307

ปริญญา วิทยาศาสตรมหาบัณฑิต

สาขาวิชา คณิตศาสตร์ประยุกต์

อาจารย์ผู้ควบคุมวิทยานิพนธ์ รศ.ภัคคินี ชิตสกุล

คณะกรรมการสอบวิทยานิพนธ์		ลายมือชื่อ
ผศ.ไพรบูลย์	พันธรักษ์พงษ์	
รศ.ภัคคินี	ชิตสกุล	
ผศ.พัชรินทร์	เหมโชติ	
ผศ.ดร.สุรสิทธิ์	ราตรี	

วัน/เดือนปี ที่สอบ 9 พฤษภาคม 2546 เวลา 10.00 น. เป็นต้นไป

สถานที่สอบ ณ อาคารจุฬาภรณวลัยลักษณ์ 2 ชั้น 2 ห้อง 218

บัณฑิตวิทยาลัยรับรองแล้ว

(รศ.ดร.บุญวัฒน์ อัตชู)

คณบดีบัณฑิตวิทยาลัย

วันที่.....๒๕.....**เดือน**.....มิถุนายน.....**พ.ศ.**.....๒๕๔๖.....

หัวข้อวิทยานิพนธ์	การรู้จำตัวอักษรลายมือเขียนภาษาไทยด้วยวิธีซินแทกติกร่วมกับแบบจำลองฟัซซี – นิวรอล
นักศึกษา	นางสาวอลิศรา หงษ์พร้อมญาติ
รหัสประจำตัว	42065307
ปริญญา	วิทยาศาสตรมหาบัณฑิต
สาขาวิชา	คณิตศาสตร์ประยุกต์
อาจารย์ผู้ควบคุมวิทยานิพนธ์	รศ.ภัคคินี ชิตสกุล

บทคัดย่อ

วิทยานิพนธ์ฉบับนี้เป็นการนำเสนอวิธีการรู้จำตัวอักษรลายมือเขียนภาษาไทยด้วยวิธีซินแทกติกร่วมกับโครงข่ายประสาทเทียมแบบแพร่ย้อนกลับ (Back – Propagation Neural Network) โดยการทำงานเริ่มจากการรับข้อมูลเป็นไฟล์ภาพมาเข้าสู่กระบวนการหลักๆ 3 กระบวนการ คือ กระบวนการเตรียมข้อมูล โดยใช้เทคนิคของ Safe Point Thinning Algorithm (SPTA) ในการทำตัวอักษรให้บาง จากนั้นทำการเข้ารหัสตัวอักษรด้วยวิธีซินแทกติก ซึ่งประกอบด้วย การเปลี่ยนเส้นโครงร่างเป็นรหัสเงื่อนไข การเปลี่ยนรหัสเงื่อนไขเป็นเวกเตอร์และการเปลี่ยนเวกเตอร์เป็นแบบเปรียบพื้นฐาน ตามลำดับ ซึ่งจะได้ข้อมูลในรูปรหัสเลขฐานสองจำนวน 52 บิท หลังจากนั้นนำข้อมูลที่ได้เข้าสู่กระบวนการฝึกสอนด้วยโครงข่ายประสาทเทียมแบบแพร่ย้อนกลับ 3 ชั้น ซึ่งประกอบด้วย ชั้นอินพุท 52 โหนด ชั้นซ่อน 36 โหนด และชั้นเอาท์พุท 44 โหนด โดยในการคำนวณระหว่างทุกๆชั้นของโครงข่ายใช้ sigmoid function เป็นฟังก์ชันกระตุ้นการตัดสินใจ เพื่อหาค่าถ่วงน้ำหนัก (weight) ของแต่ละชั้นของโครงข่ายเพื่อเก็บไว้เป็นฐานข้อมูล

สุดท้ายกระบวนการทดสอบการรู้จำ นำตัวอักษรที่ต้องการทดสอบมาวิเคราะห์โดยเข้าสู่กระบวนการเตรียมข้อมูลเพื่อเปลี่ยนเป็นรหัสเลขฐานสองจำนวน 52 บิท และนำค่าถ่วงน้ำหนักจากฐานข้อมูลมาทดสอบเพื่อหาค่าผิดพลาด โดยค่าผิดพลาดที่ยอมรับได้อยู่ในช่วง $10^{-6} - 10^{-4}$ จากการทดสอบตัวอักษรลายมือเขียนภาษาไทย จำนวน 1320 ตัวอักษร ได้ความถูกต้อง 92.12%

Thesis Title	Thai Alphabet Handwriting Recognition using Syntactic Method and Fuzzy – Neural Model
Studen	Miss Alissara Hongpromyart
Studen Id.	42065307
Degree	Master of Science
Programme	Applied Mathematics
Year	2003
Thesis Adviser	Assoc.Prof.Pakkinee Chitsakul

ABSTRACT

This thesis presents the schemes of the Thai Alphabet Handwriting Recognition using Syntactic Method and Back – Propagation Neural Network. The process is started by receiving the data into picture file and then continue to main process as following : Data preparation by the safe point thinning algorithm (SPTA) and the syntactic method which is three steps, the first step is to convert the skeleton to the conditional code, the second step is to convert the conditional code to vectors and the last one is to convert this vectors to primative patterns which is binary code as 52 bits. Training by Back – Propagation Neural Network is composed with three layers , the first layer is input 52 node, the second layer is hidden 36 node and the third layer is output 44 node. They are decided by sigmoid function for weight each layers and keep to data base. Finally Recognition tester by introduce alphabet Thai handwriting input to the data preparation and weight from data base to test error. The error is accept in range 10^{-6} to 10^{-4}. The Recognition tester in 1320 Thai alphabets handwriting are corrected 92.12%.

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงไปได้ด้วยดีเพราะได้รับความกรุณาจาก รศ. ภัคคินี ชิตสกุล อาจารย์ผู้ควบคุมวิทยานิพนธ์ ซึ่งท่านได้ให้คำแนะนำและได้ให้โอกาสในการทำวิทยานิพนธ์ฉบับนี้ ขอขอบคุณอาจารย์ สงกรานต์ กันทวงศ์ ที่ท่านได้ให้คำแนะนำและช่วยแก้ไขในบางจุดที่มีปัญหา ซึ่งมีส่วนทำให้ผู้ทำวิจัยเข้าใจในปัญหานั้นมากขึ้น รวมทั้งที่ได้ให้ยืมเครื่องคอมพิวเตอร์สำหรับใช้ในการพัฒนาโปรแกรม ผู้วิจัยรู้สึกซาบซึ้งในความอนุเคราะห์เป็นอย่างสูง

ขอขอบคุณ คุณรังสรรค์ พรายแก้ว ที่ได้ให้ใช้เครื่องคอมพิวเตอร์และเครื่องสแกนเนอร์เพื่อใช้ในการเก็บรวบรวมตัวอย่างตัวอักษร ตลอดจนเพื่อนๆทุกคนที่คอยให้กำลังใจมาตลอด

สุดท้ายนี้ผู้วิจัยขอขอบพระคุณ บิดา – มารดา ซึ่งได้ให้การสนับสนุนและเป็นกำลังใจแก่ผู้วิจัยจนสำเร็จการศึกษา และขอบพระคุณผู้ให้ทุนการศึกษาในครั้งนี้

อลิศรา หงษ์พร้อมญาติ

สารบัญ

หน้า

บทคัดย่อภาษาไทย..... I

บทคัดย่อภาษาอังกฤษ..... II

กิตติกรรมประกาศ III

สารบัญ..... IV

สารบัญตาราง.....VII

สารบัญภาพ.....VIII

บทที่ 1 บทนำ..... 1

1.1 ความสำคัญและที่มีของวิทยานิพนธ์ 1

1.2 วัตถุประสงค์ของงานวิจัย 4

1.3 ข้อตกลงเบื้องต้นในการทำวิทยานิพนธ์ 4

1.4 ประโยชน์ที่คาดว่าจะได้รับจากงานวิจัย 5

บทที่ 2 เอกสารและงานวิจัยที่เกี่ยวข้อง 6

2.1 วิธีการแปลงข้อมูลภาพตัวอักษรลายมือเขียนภาษาไทย 7

2.2 การปรับปรุงคุณภาพของภาพข้อมูล 8

2.2.1 การกำจัดสัญญาณรบกวนและการปรับปรุงคุณภาพของตัวอักษร 8

2.2.2 การสำรวจพื้นที่ของภาพ 10

2.2.3 การทำตัวอักษรให้บาง 11

2.2.4 การลบกิ่งและการขลิบปลาย 12

2.3 การวิเคราะห์หาลักษณะสำคัญของตัวอักษร 12

2.3.1 การแปลงเส้นโครงร่างเป็นรหัสเงื่อนไข 13

2.3.2 การแปลงรหัสเงื่อนไขให้เป็นเวกเตอร์ 20

2.3.3 การเปลี่ยนเวกเตอร์ให้เป็นแบบเปรียบพื้นฐาน 22

2.4 การเปรียบเทียบโดยการวิเคราะห์ทางFeature.....23

2.4.1 ลักษณะเด่นที่ใช้ในการเปรียบเทียบตัวอักษร.....23

2.4.2 การแปลงแบบเปรียบพื้นฐานให้เป็นรหัสเลขฐานสอง.....27

2.5 โครงข่ายประสาทเทียม 32

2.5.1 ประวัติความเป็นมา 32

2.5.2 ลักษณะโครงข่ายประสาททางชีววิทยา 33

2.5.3 โครงข่ายประสาทเทียม 34

สารบัญ (ต่อ)

หน้า

2.6 โครงข่ายประสาทเทียมแบบต่างๆ 36
2.6.1 โครงข่ายประสาทเทียมแบบชั้นเดียว 36
2.6.2 โครงข่ายประสาทเทียมแบบหลายชั้น 37
2.7 การฝึกสอนให้กับโครงข่ายประสาทเทียม 38
2.7.1 วิธีฝึกสอนโครงข่ายประสาทเทียม 38
2.8 ฟังก์ชันกระตุ้นการตัดสินใจ 40
2.8.1 ชนิดของฟังก์ชันกระตุ้นการตัดสินใจ 40
2.8.2 การคูณเมทริกซ์สำหรับการคำนวณผลรวมของอินพุท 42
2.8.3 Bias 42
2.9 ทฤษฎีหลักการแพร่ย้อนกลับ 43
2.9.1 กฎเดลต้า 43
2.9.2 โครงข่ายประสาทเทียมแบบแพร่ย้อนกลับ 45
2.9.3 การฝึกสอนโครงข่ายประสาทเทียมแบบแพร่ย้อนกลับ 46
2.9.4 อัลกอริทึมการฝึกสอนโครงข่ายประสาทเทียมแบบแพร่ย้อนกลับ 47
2.10 การออกแบบโครงข่ายประสาทเทียม 49
2.10.1 ขนาดของโครงข่ายประสาทเทียม 49
2.10.2 จำนวนของชั้นในโครงข่าย 49
2.10.3 จำนวนของนิวรอลในแต่ละชั้น 49
2.10.4 การเลือกค่าน้ำหนักและ bias 50
2.10.5 ระยะเวลาในการฝึกสอนโครงข่าย 51
2.10.6 จำนวนคู่ของการฝึกสอน 51
2.11 งานวิจัยที่เกี่ยวข้อง 52

บทที่ 3 วิธีดำเนินการวิจัย 52
3.1 ขั้นตอนการเตรียมข้อมูล 55
3.1.1 ลักษณะของตัวอักษรที่ใช้ในการทดลอง 55
3.2 ขั้นตอนการฝึกสอน 56
3.3 ขั้นตอนการรู้จำ 57

สารบัญ (ต่อ)

หน้า

บทที่ 4 ผลการวิจัย 59

4.1 ผลการทดสอบการรู้จำ 59

4.2 วิเคราะห์ผลการทดลอง 62

4.3 เปรียบเทียบการทำงานของการรู้จำตัวอักษรคัดลายมือเขียน ภาษาไทยระบบนี้กับระบบอื่นๆ 63

บทที่ 5 สรุปผลและข้อเสนอแนะ.....64

5.1 สรุปผลการวิจัย 64

5.2 ข้อจำกัดและปัญหา 64

5.3 ข้อเสนอแนะ 65

เอกสารอ้างอิง 66

ภาคผนวก ก Safe Point Thinning Algorithm 67

ภาคผนวก ข ที่มาของกฎการเรียนรู้ 71

ภาคผนวก ค องค์ประกอบของโปรแกรมของวิทยานิพนธ์ 74

ภาคผนวก ง ตัวอักษรคัดลายมือเขียนภาษาไทย 77

ภาคผนวก จ ตัวอย่างตัวอักษร ก – ฮ ที่ผ่านกระบวนการหาลักษณะเด่น.....84

ภาคผนวก ฉ ตารางแสดงรหัสเลขฐานสองของตัวอักษรต้นแบบ 87

ประวัติผู้เขียน 96

สารบัญตาราง

ตารางที่ หน้า

2.1 แสดงความสัมพันธ์ระหว่างจุดภาพ 15

2.2 แสดงมุมของรหัสเงื่อนไข 20

2.3 แสดงมุมของรหัสเงื่อนไขหลักและรหัสเงื่อนไขรองคละกัน 21

2.4 แสดงการแปลงแบบเปรียบเส้นตรงให้เป็นเลขฐานสอง 27

2.5 แสดงการแปลงแบบเปรียบวงกลมเป็นเลขฐานสอง 27

2.6 แสดงการแปลงลักษณะเด่นของตัวอักษรเป็นรหัสเลขฐานสอง 28

2.7 แสดงตัวอย่างการแปลงข้อมูลภาพตัวอักษรให้เป็นเลขฐานสองของตัวอักษร ข 30

3.1 แสดงลักษณะหัวตัวอักษร 58

4.1 แสดงผลการทดสอบความถูกต้องของตัวอักษร จากผู้คัดลายมือ 30 คน.....60

4.2 แสดงผลการเปรียบเทียบกับงานวิจัยอื่น 61

4.3 แสดงผลการเปรียบเทียบการรู้จำของงานวิจัยนี้กับระบบการรู้จำของ
สุรพันธ์ เอื้อไพบูลย์ กับ สกุล คำนวนชัย 63

สารบัญรูป

รูปที่		หน้า
2.1	โครงสร้างการแปลงข้อมูลภาพของตัวอักษรเป็นรหัสเลขฐานสอง	7
2.2	แสดงการทำงานของการปรับปรุงคุณภาพของรูปข้อมูล	8
2.3	แสดงการกำจัดสัญญาณรบกวน	9
2.4	แสดงหน้าต่างที่ใช้กำจัดสัญญาณรบกวนและรูปที่ได้จากการกำจัดสัญญาณรบกวน	9
2.5	แสดงการปรับปรุงคุณภาพของตัวอักษร	10
2.6	แสดงภาพการสำรวจพื้นที่ที่แบ่งเป็นส่วน	10
2.7	แสดงรูปก่อนการและหลังการทำตัวอักษรให้บาง	11
2.8	แสดงรูปก่อนและหลังการ Delete Branch & Clip End	12
2.9	แสดงการหาลักษณะเด่นของข้อมูลภาพเส้นโครงร่างตัวอักษร	13
2.10	แสดงลักษณะของจุดภาพข้างเคียง	14
2.11	แสดงรหัสเงื่อนไข	18
2.12	การเข้ารหัสเงื่อนไข	20
2.13	แสดงการแปลงเวกเตอร์เส้นตรงเป็นเวกเตอร์วงกลม	22
2.14	แสดงการแปลงข้อมูลภาพตัวอักษรที่เข้ารหัสเงื่อนไขเป็นเวกเตอร์	22
2.15	แสดงรูปแบบของแบบเปรียบหลัก	23
2.16	แสดงส่วนของตัวอักษร	24
2.17	แสดงจำนวนหัวของตัวอักษร	24
2.18	แสดงจำนวนปลายของตัวอักษร	25
2.19	แสดงหน้าต่างการอ้างอิงตำแหน่งที่จุดหัวหรือจุดปลายใดๆ	25
2.20	แสดงตำแหน่งหัวแรกหรือปลายแรก และตำแหน่งหัวหรือปลายที่ 2	26
2.21	แสดงหน้าต่างขั้นตอนการแปลงข้อมูลภาพตัวอักษรให้เป็นเลขฐานสอง ของอักษร ข	29
2.22	แสดงหน้าต่างขั้นตอนการแปลงข้อมูลภาพตัวอักษรให้เป็นเลขฐานสอง ของอักษร ง	30
2.23	แสดงแผนผังลักษณะการทำงานของการแปลงตัวอักษรลายมือเขียน ภาษาไทยเป็นรหัสเลขฐานสอง	31
2.24	แสดงโครงข่ายประสาททางชีววิทยา	34
2.25	แสดงแบบจำลองของเซลล์ประสาทเทียมอย่างง่าย	34

สารบัญรูป (ต่อ)

รูปที่		หน้า
2.26	แสดงแบบจำลองของโครงข่ายประสาทเทียมอย่างง่าย	35
2.27	แสดงโครงข่ายประสาทเทียมชั้นเดียว	36
2.28	แสดงโครงข่ายประสาทเทียมหลายชั้น	37
2.29	แสดงโครงข่ายประสาทเทียมแบบแพร่ย้อนกลับกับการสอนแบบชี้นำ	39
2.30	แสดง Identity function	40
2.31	แสดง Binary step function	40
2.32	แสดง Binary sigmoid	41
2.33	แสดง Bipolar sigmoid	41
2.34	แสดงโครงข่ายประสาทเทียมที่มี bias	43
2.35	แสดงนิวรอลที่ไม่มีฟังก์ชันกระตุ้นแต่มีเอาท์พุทเป้าหมาย t และค่าผิดพลาด E	43
2.36	แสดงโครงข่ายประสาทเทียมแบบแพร่ย้อนกลับแบบมีชั้นซ่อน 1 ชั้น	46
3.1	แสดงตัวอย่างลักษณะของตัวอักษรที่ทำการวิเคราะห์(บางส่วน)	55
3.2	แสดงขั้นตอนการทำงานของการฝึกสอน	57
3.3	แสดงโครงสร้างของการทดสอบ	57

บทที่ 1
บทนำ

1.1 ความสำคัญและที่มาของวิทยานิพนธ์

ปัจจุบันระบบการประมวลผลด้วยคอมพิวเตอร์ได้ถูกพัฒนาไปอย่างรวดเร็ว เพราะประสิทธิภาพและคุณประโยชน์ที่อำนวยต่อการดำเนินธุรกิจและชีวิตประจำวัน ดังนั้นจึงมีผู้สนใจคิดค้นเครื่องมือต่างๆในการพัฒนาเทคโนโลยีเหล่านี้ เช่น ระบบผู้เชี่ยวชาญ (Expert System) ระบบปัญญาประดิษฐ์ (Artificial Intelligence:AI) ระบบการประมวลผลข้อมูล (Data Processing) ระบบจัดเตรียมเอกสาร (Document Preparation System) ฯลฯ จากความก้าวหน้าของระบบดังกล่าวทำให้การประมวลผลด้วยคอมพิวเตอร์ถูกนำไปใช้งานในเกือบจะทุกสาขาอาชีพ แต่ไม่ว่าจะในงานลักษณะใดก็ตามวิธีหนึ่งที่ผู้ใช้สามารถติดต่อกับคอมพิวเตอร์ก็คือ การใช้แป้นพิมพ์หรือคีย์บอร์ด

การป้อนข้อมูลผ่านทางแป้นพิมพ์นี้เป็นวิธีการดั้งเดิมที่ใช้กันมาตั้งแต่แรกและยังคงใช้ต่อเนื่องกันมาจนกระทั่งปัจจุบัน โดยทั่วๆไปแล้ววิธีการนี้เป็นวิธีที่ง่ายและสะดวกสำหรับผู้ใช้ที่มีความชำนาญหรือผู้ที่สามารถพิมพ์ดีดได้คล่องแคล่ว แต่สำหรับผู้ที่ขาดความชำนาญในการพิมพ์อาจเกิดปัญหาและมีโอกาสผิดพลาดได้ง่ายทั้งยังทำให้เสียเวลามากแต่ปัญหาที่สำคัญที่สุดคือ ในกรณีที่มีข้อมูลที่ต้องการจัดเก็บมากๆจำเป็นต้องหาบุคคลากรและเครื่องมือเพิ่มขึ้นซึ่งเป็นการสิ้นเปลืองงบประมาณและบุคคลากร การแก้ปัญหาที่ถูกคือการนำเทคโนโลยีใหม่ๆมาทดแทนซึ่งการรู้จำตัวอักษรด้วยคอมพิวเตอร์ก็เริ่มเข้ามามีบทบาทในด้านนี้เพิ่มขึ้นอย่างมาก เพื่อจะให้เป็นทางเลือกในการป้อนข้อมูลเข้าสู่คอมพิวเตอร์โดยการอ่านข้อความจากเอกสารหรือรับรู้จากการเขียนที่จอภาพได้โดยตรงโดยมนุษย์ไม่จำเป็นต้องป้อนข้อมูลผ่านทางแป้นพิมพ์เพียงอย่างเดียว

การรู้จำรูปแบบตัวอักษรได้มีการประยุกต์ใช้กับหลายภาษา เช่น ตัวอักษรภาษาอังกฤษซึ่งมีการพัฒนาก้าวหน้าอย่างมากจนกระทั่งสามารถนำมาประยุกต์ใช้และจำหน่ายในเชิงพาณิชย์ได้ทั้งแบบตัวพิมพ์และตัวเขียน ส่วนภาษาอื่นๆก็ได้มีงานวิจัยศึกษาพัฒนาระบบการรู้จำของภาษานั้นๆออกเผยแพร่และตีพิมพ์หลายงานวิจัย แต่สำหรับตัวอักษรภาษาไทยได้มีการศึกษาค้นคว้ากันบ้างพอสมควรแต่เนื่องจากตัวอักษรภาษาไทยมีโครงสร้างที่ค่อนข้างซับซ้อนประกอบด้วยลักษณะที่เป็นเส้นตรงผสมเส้นโค้งและวงกลมซึ่งบางครั้งมีการตัดกันของเส้นทำให้การศึกษาค้นคว้าไม่แพร่หลายและต่อเนื่องมากนัก

Schalkoff (1992) ได้จำแนกวิธีการของการรู้จำรูปแบบ (Pattern Recognition) ได้ 3 วิธี คือ

1. ระบบการรู้จำรูปแบบเชิงสถิติ (Statistical Pattern Recognition - StatPR) หรือเรียกอีกอย่างหนึ่งว่า “Decision Theoretic” เป็นวิธีที่ใช้หลักการวิเคราะห์ของคณิตศาสตร์ทางสถิติ

(Statistical basis) เป็นหลักโดยที่ข้อมูลอินพุทจะถูกแปลงให้เป็นรูปลักษณ์ (Feature) เพื่อใช้ในการเปรียบเทียบและใช้กฎการตัดสินใจ (Decision Rules) ต่างๆในการจำแนก (Classification) การรู้จำ

2. ระบบการรู้จำรูปแบบเชิงโครงร่าง (Syntactic Pattern Recognition - SyntPR) หรือเรียกอีกอย่างว่า "Structural Pattern Recognition" เป็นวิธีที่พิจารณาโครงสร้างของอินพุทให้อยู่ในรูปแบบของเซตของรูปลักษณ์ (Extracted Features) และในส่วนของการจำแนกการรู้จำจะเป็นลักษณะการวัดค่าความคล้าย (Pattern Structural Similarity) ของอินพุทโดยวิธีของ Formal Grammers Relational Description (Principally Graph)

3. ระบบการรู้จำรูปแบบโครงข่ายสมอง (Neural Pattern Recognition - NeurPR) ซึ่งมีชื่อเรียกต่างๆกัน เช่น Connectionist Modeling , Neuromorphic Modeling และ Parallel Distributed Processing (PDP) เป็นระบบโครงข่ายที่พยายามสร้างจำลองเลียนแบบโครงสร้างเซลล์สมองของมนุษย์ (Biological Neural System) ซึ่งเซลล์ (neuron) ต่างๆเหล่านี้เชื่อมต่อกันเป็นโครงข่ายโดยมีน้ำหนักของการเชื่อมต่อ (Weighted Connection) เป็นเงื่อนไขในการตัดสินใจของระบบ

การวิเคราะห์และแยกแยะตัวอักษรโดยการใช้สมการทางคณิตศาสตร์นั้นสามารถทำได้เร็วกว่าการวิเคราะห์โครงร่าง แต่ถ้ากลุ่มของตัวอักษรมีความคล้ายคลึงกันมากก็จะทำให้เกิดความผิดพลาดสูง คณิตศาสตร์ที่นำมาใช้ได้แก่ สมการการตัดสินใจแบบเชิงเส้น (Linear decision function) การวัดการกระจายโดยใช้ K - L expansion (Karhunen – Loeve expansion) การคำนวณค่าลักษณะต่อเนื่องของจุดภาพ (Connectivity) เป็นต้น ส่วนการวิเคราะห์โครงร่างของตัวอักษรนั้น ถ้าหากว่าจำนวนของตัวอักษรที่ต้องการรู้จำมีมากเวลาที่ใช้ในการรู้จำก็จะมากขึ้นด้วย ตัวอย่างของการวิเคราะห์โครงร่างได้แก่ การเข้ารหัสของลายเส้นแบบต่างๆ การพิจารณามุมของเส้นส่วนโค้ง (Curve) เป็นต้น

ปัจจุบันมีการนำเสนอวิธีการหลายๆวิธีในการรู้จำตัวอักษรภาษาไทยซึ่งได้ผลลัพธ์และประสิทธิภาพของการทำงานที่แตกต่างกัน ดังเช่น

สุรพันธ์ เอื้อไพบูลย์[1] นำเสนอวิธีการรู้จำตัวอักษรลายมือเขียนภาษาไทย โดยการนำหัวของตัวอักษรมาพิจารณาเพื่อจำแนกกลุ่มของตัวอักษรออกเป็นกลุ่มย่อย โดยให้ตัวอักษรที่มีหัวอยู่บริเวณเดียวกันอยู่กลุ่มเดียวกัน จากนั้นแต่ละกลุ่มก็จะมีวิธีการพิจารณาเปรียบเทียบตัวอักษรที่อยู่ในกลุ่มของตนเองแตกต่างกันไปขึ้นอยู่กับลักษณะเด่นของตัวอักษรที่อยู่ในกลุ่มนั้นๆ ข้อดีคือ ทำการเปรียบเทียบได้อย่างรวดเร็วแต่การนำไปประยุกต์ใช้งานกับตัวอักษรประเภทอื่นๆต้องมีการปรับปรุงโปรแกรมบางส่วน เพื่อให้สามารถรับเงื่อนไขของตัวอักษรใหม่ๆได้

สกุล คำนวณชัย[2] นำเสนอเรื่องการรู้จำตัวอักษรคัดลายมือเขียนภาษาไทย โดยทำการทดสอบกับตัวอักษรพยัญชนะทั้งหมด 44 ตัว จากนั้นทำการแปลงข้อมูลแบบคาร์ฮูเน็นเลิฟเพื่อหาชุดข้อมูลใหม่ที่เป็นตัวแทนของพยัญชนะแต่ละตัวจากนั้นนำค่าที่ได้ไปทำการลดจำนวนลักษณะเด่นของข้อมูลโดยใช้หลักการของฟัซซีเซต หลังจากนั้นนำข้อมูลที่ได้เข้ากระบวนการเรียนรู้โดยใช้

โครงข่ายประสาทเทียมแบบแพร่ย้อนกลับ ข้อดีคือ สามารถหาลักษณะร่วมของตัวอักษรตัวเดียวกันได้ทำให้การรู้จำสูง แต่ใช้จำนวนบิทในการคำนวณโครงข่ายประสาทเทียมมากทำให้ใช้เวลาในการฝึกสอนนาน

สนธยา เมรินทร์[3] นำเสนอเรื่องการศึกษาการรู้จำตัวอักษรพิมพ์ภาษาไทยโดยวิธีซินแทกติกซึ่งเป็นการพิจารณาที่โครงสร้างของตัวอักษร โดยมีการอธิบายโครงสร้างตัวอักษรในรูปของแบบเปรียบพื้นฐาน (primitive) ทำให้สามารถจำแนกตัวอักษรที่มีโครงสร้างที่แตกต่างกันออกจากกันได้อย่างรวดเร็ว รวมทั้งยังใช้วิธีการเปรียบเทียบทางลักษณะเด่น (feature) ควบคู่กันไปด้วยสำหรับตัวอักษรบางกลุ่มที่มีลักษณะคล้ายกันมาก

เดชา รัตนธาร[4] ได้ทำการปรับปรุงการรู้จำตัวอักษรไทยด้วยวิธีซินแทกติกของ สนธยา เมรินทร์[3] โดยได้นำเอาเทคนิคแบบฟัซซีลอจิกเข้ามาช่วยในการปรับปรุงให้มีความแม่นยำในการรู้จำสูงขึ้น โดยยังคงวิเคราะห์โครงสร้างของตัวอักษรด้วยวิธีซินแทกติก

วิชิน มุ่งสิน[5] นำเสนอเรื่องการรู้จำตัวอักษรพิมพ์ภาษาลาวโดยทำการปรับข้อมูลภาพตัวอักษรที่เป็นบิตแมพให้เป็นข้อมูลไบนารี จากนั้นกำหนดค่าลักษณะเด่นของตัวอักษรโดยคำนวนตามจุดเนื้อภาพตัวอักษร แล้วนำค่าที่ได้ไปหาค่าความเป็นสมาชิกของฟัซซีเซตแบบสามเหลี่ยม ค่าฟัซซีที่ได้จะนำเข้าสู่กระบวนการเรียนรู้ในโครงข่ายประสาทเทียมแบบแพร่ย้อนกลับ

สุขสถิต สุขใจ[6] นำเสนอการใช้โครงข่ายประสาทเทียมเพื่อจดจำรูปแบบลายมือเขียนตัวเลขอารบิคโดยแสดงข้อมูลเป็นมุมสัมบูรณ์ โดยข้อมูลตัวเลขอารบิคจะถูกนำมาแปลงเป็นข้อมูลมุมสัมบูรณ์ หลังจากนั้นจะนำข้อมูลที่ได้เข้าสู่กระบวนการเรียนรู้โดยใช้โครงข่ายประสาทเทียมแบบแพร่ย้อนกลับ

งานวิจัยที่ผ่านมาล้วนแต่เป็นประโยชน์และเป็นรากฐานในการพัฒนาระบบการรู้จำตัวอักษรคัดลายมือเขียนภาษาไทยต่อไปในอนาคต ปัญหาที่ประสบในงานวิจัยก่อนๆคือ การรู้จำตัวอักษรบางตัวที่มีรายละเอียดสูงหรือมีโครงสร้างที่ยากต่อการประมวลผลขั้นต้นด้วยวิธีการธรรมดาและหรือยากต่อการจำแนก เนื่องจากตัวอักษรภาษาไทยมีโครงสร้างเป็นเส้นตรงผสมเส้นโค้งและวงกลม บางครั้งมีการตัดกันของเส้นทำให้มีโครงสร้างค่อนข้างซับซ้อน การรู้จำด้วยวิธีซินแทกติกค่อนข้างเหมาะสมกับลักษณะของตัวอักษรไทย เพราะวิธีซินแทกติกเป็นการพิจารณาที่โครงสร้างของตัวอักษรเท่านั้น และอธิบายโครงสร้างของตัวอักษรในรูปของแบบเปรียบพื้นฐาน ทำให้สามารถจำแนกตัวอักษรที่มีโครงสร้างที่แตกต่างออกจากกันได้ง่ายและรวดเร็ว

ดังนั้นในวิทยานิพนธ์นี้จึงใช้วิธีการทางซินแทกติกในการรู้จำตัวอักษร แต่วิธีการทางซินแทกติกเพียงวิธีการเดียวไม่สามารถจำแนกได้อย่างชัดเจนนัก เนื่องจากตัวอักษรไทยบางตัวมีลักษณะที่คล้ายคลึงกันมาก จึงจำเป็นต้องอาศัยวิธีการเปรียบเทียบลักษณะเด่นของตัวอักษรควบคู่ไปด้วยโดยแทนค่าลักษณะเด่นเหล่านั้นด้วยรหัสเลขฐานสอง จากนั้นทำการฝึกสอนและทดสอบ

การรู้จำด้วยโครงข่ายประสาทเทียมแบบแพร่ย้อนกลับโดยใช้เทคนิคของฟัซซีช่วยในการวิเคราะห์เพื่อปรับปรุงอัตราการรู้จำให้ได้ผลลัพธ์และความถูกต้องมากที่สุด

1.2 วัตถุประสงค์ของงานวิจัย

1. ศึกษาและพัฒนาระบบการรู้จำตัวอักษรคัดลายมือเขียนภาษาไทยด้วยโครงข่ายประสาทเทียมแบบแพร่ย้อนกลับ
2. ศึกษาวิธีการทางซินแทกติกเพื่อหาลักษณะเด่นของตัวอักษรคัดลายมือเขียนภาษาไทยเพื่อใช้ในการฝึกสอนและทดสอบการรู้จำด้วยโครงข่ายประสาทเทียมแบบแพร่ย้อนกลับ

1.3 ข้อตกลงเบื้องต้นในการทำวิทยานิพนธ์

1. งานวิจัยนี้เน้นการหาลักษณะเด่นของตัวอักษรแต่ละตัวด้วยวิธีซินแทกติกและทำการแปลงลักษณะเด่นเหล่านั้นเป็นรหัสเลขฐานสอง เพื่อใช้เป็นข้อมูลอินพุทสำหรับฝึกสอนและทดสอบการรู้จำด้วยโครงข่ายประสาทเทียมแบบแพร่ย้อนกลับ
2. ตัวอักษรที่ใช้ในงานวิจัยนี้จะต้องเป็นลักษณะคัดลายมือภาษาไทยตัวบรรจง และเขียนบนกระดาษพิมพ์เขียนสีขาวในพื้นที่ที่กำหนดด้วยปากกาหมึกซึมหรือปากกาหมึกเจล
3. อักษรที่ใช้ในงานวิจัยนี้เป็นพยัญชนะไทย 44 ตัว ที่เขียนในช่องที่กำหนด ช่องละหนึ่งตัว ไม่ใช่ตัวอักษรที่เขียนขึ้นเป็นคำหรือประโยค
4. ผลการวิจัยได้จากอุปกรณ์ที่มีคุณสมบัติเฉพาะดังนี้

   4.1 เครื่องคอมพิวเตอร์

   CPU : AMD Athlon Processor ความเร็ว 133 MHz

   RAM : 768.0 MB

   Haddisk : 30 GB

   4.2 เครื่องอ่านภาพ (Scanner) Hewlett Packard รุ่น hp Scanjet 5400 C และกำหนดลักษณะในการอ่านภาพดังนี้

   Type : Black and White Bitmap

   Scanling : 100%

1.4 ประโยชน์ที่คาดว่าจะได้รับจากงานวิจัย

1. พัฒนาความรู้ในเรื่องการรู้จำตัวอักษรคัดลายมือเขียนภาษาไทยด้วยโครงข่ายประสาทเทียมแบบแพร่ย้อนกลับ
2. สามารถนำไปประยุกต์ใช้กับการรู้จำตัวอักษรลายมือเขียนภาษาอื่นๆได้

บทที่ 2
เอกสารและงานวิจัยที่เกี่ยวข้อง

เนื้อหาในบทนี้จะนำเสนอถึงวิธีการในการรู้จำตัวอักษรคัดลายมือเขียนภาษาไทย เนื้อหาในบทแยกเป็นหัวข้อดังต่อไปนี้

หัวข้อ 2.1 วิธีการแปลงข้อมูลภาพตัวอักษรลายมือเขียนภาษาไทย อธิบายถึงโครงสร้างของวิธีการแปลงข้อมูลภาพตัวอักษรให้เป็นรหัสเลขฐานสองที่ใช้ในการรู้จำ

หัวข้อ 2.2 การปรับปรุงคุณภาพของภาพข้อมูล กล่าวถึงกระบวนการเตรียมภาพข้อมูลก่อนที่จะนำไปฝึกสอนและทดสอบการรู้จำ ประกอบด้วย การกำจัดสัญญาณรบกวน การปรับความเข้มของภาพ การสำรวจพื้นที่ของภาพ การทำตัวอักษรให้บาง และการลบกิ่งและขลิบปลาย

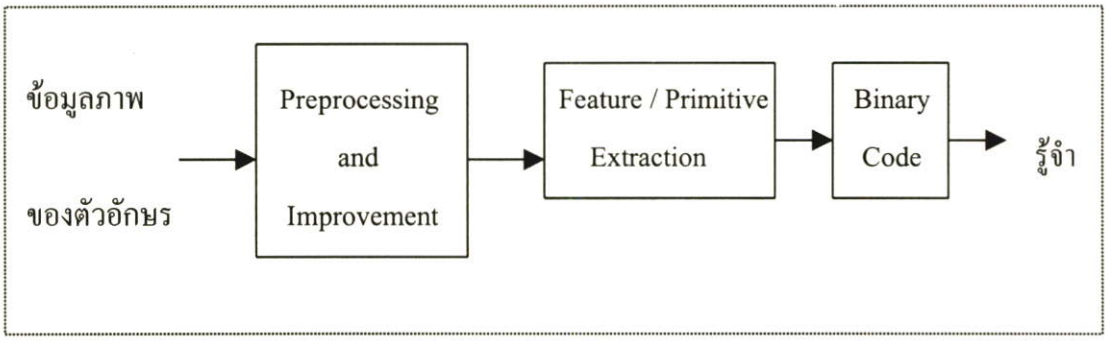
หัวข้อ 2.3 การวิเคราะห์หาลักษณะสำคัญของตัวอักษร กล่าวถึงกระบวนการหาลักษณะเด่นของภาพข้อมูลเพื่อใช้เป็นตัวแทนของตัวอักษรแต่ละตัวในการฝึกสอนและทดสอบการรู้จำ ประกอบด้วย การแปลงเส้นโครงร่างเป็นรหัสเงื่อนไข การแปลงรหัสเงื่อนไขให้เป็นเวกเตอร์ และการเปลี่ยนเวกเตอร์ให้เป็นแบบเปรียบพื้นฐาน

หัวข้อ 2.4 การเปรียบเทียบโดยการวิเคราะห์ทางลักษณะเด่น กล่าวถึงลักษณะเด่นที่ใช้เป็นตัวแทนของตัวอักษรแต่ละตัว และการแปลงลักษณะเด่นเหล่านั้นเป็นรหัสเลขฐานสองเพื่อใช้เป็นข้อมูลอินพุทเข้าสู่โครงข่ายประสาทเทียมแบบแพร่ย้อนกลับ

หัวข้อ 2.5 – 2.10 อธิบายถึงประวัติความเป็นมาและลักษณะของโครงข่ายประสาทเทียมที่นำมาประยุกต์ใช้ในงานวิจัย รวมทั้งวิธีการฝึกสอนและอัลกอริทึมการทำงานของโครงข่ายประสาทเทียมแบบแพร่ย้อนกลับ

2.1 วิธีการแปลงข้อมูลภาพตัวอักษรลายมือเขียนภาษาไทย

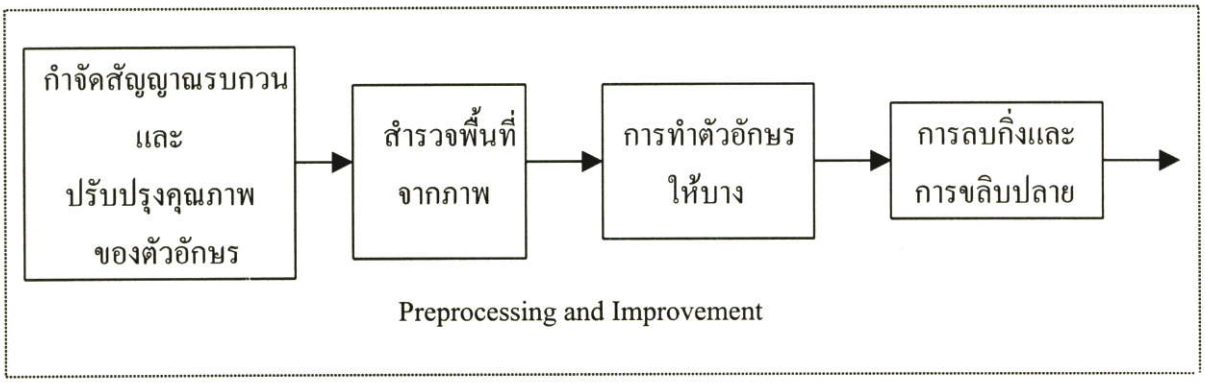
การศึกษาการรู้จำตัวอักษรคัดลายมือเขียนภาษาไทยในครั้งนี้มุ่งเน้นที่ความถูกต้องเป็นหลัก ดังนั้นวิธีการที่ใช้จึงเป็นวิธีที่ค่อนข้างซับซ้อน ดังที่กล่าวมาแล้วว่าตัวอักษรภาษาไทยมีโครงสร้างของตัวอักษรที่ค่อนข้างซับซ้อน ดังนั้นจึงจำเป็นต้องมีกระบวนการเปลี่ยนแปลงภาพข้อมูลของตัวอักษรให้มีความเหมาะสมสำหรับการรู้จำ พิจารณาโครงสร้างของระบบการรู้จำดังรูปที่ 2.1 เมื่อรับภาพตัวอักษรภาษาไทยจากอุปกรณ์รับรู้ทางแสงอันได้แก่ เครื่องสแกนเนอร์ ซึ่งจะเก็บภาพตัวอักษรภาษาไทยที่ได้ลงไปบนแฟ้มข้อมูลภาพเป็นลักษณะจุดภาพ ข้อมูลภาพของตัวอักษรจะถูกปรับปรุงคุณภาพโดยการกำจัดสัญญาณรบกวน และการปรับปรุงคุณภาพของตัวอักษร (Noise Reduction & Contrast) สำรวจพื้นที่จากภาพ (Split) เพื่อให้ข้อมูลจุดภาพที่สแกนเข้ามาสมบูรณ์ขึ้น ต่อจากนั้นภาพของตัวอักษรจะถูกทำให้บาง (Thinning) โดยให้มีความกว้างของเส้นตัวอักษรเพียง 1 จุดภาพ แล้วนำตัวอักษรที่บางมาทำการลบกิ่งและขลิบปลาย (Delete Branch & Clip End) เพื่อปรับปรุงคุณภาพของข้อมูลตัวอักษรที่ทำให้บางแล้วหลังจากนั้นจะนำมาหาจุดเด่นของภาพ เช่น จำนวนหัว มาใช้ในการรู้จำและข้อมูลของภาพตัวอักษรจะถูกเข้ารหัส (Condition Code) เพื่อหาทิศทางซึ่งจะนำรหัสต่างๆเหล่านี้มาแปลงเป็นเวกเตอร์โดยการเปรียบเทียบกับรหัสลูกโซ่ของฟรีแมน (Freeman Chin Code) 8 ทิศทาง และนำเอาเฉพาะเวกเตอร์ที่มีทิศทางเดียวกันมารวมกันเป็นแบบเปรียบพื้นฐาน (Primitive) โดยที่ภาพข้อมูลของตัวอักษรทั้งหมดจะถูกแทนด้วยแบบเปรียบพื้นฐานแล้วข้อมูลแบบเปรียบพื้นฐานเหล่านี้จะถูกเปลี่ยนเป็นรหัสเลขฐานสองเพื่อการฝึกและรู้จำต่อไป



รูปที่ 2.1 โครงสร้างการแปลงข้อมูลภาพของตัวอักษรเป็นรหัสเลขฐานสอง

2.2 การปรับปรุงคุณภาพของภาพข้อมูล (Preprocessing and Improvement)

ตัวอักษรที่ใช้ในงานวิจัยนี้มีลักษณะเป็นตัวอักษรที่เขียนอยู่ในช่องหรือพื้นที่ที่กำหนด โดยเขียนช่องละหนึ่งตัวอักษรเท่านั้น จากนั้นนำมาผ่านเครื่องสแกนเนอร์ซึ่งจะถูกจัดเก็บในรูปแบบของอิมเมจไฟล์ เช่น ไฟล์ตระกูล BMP ซึ่งในงานวิทยานิพนธ์ครั้งนี้จะเก็บภาพเป็นแบบ Binary หรือขาวดำ ซึ่งการเก็บข้อมูลเป็นแบบที่ใช้ 1 บิตข้อมูลต่อ 1 จุดภาพ กล่าวคือ 1 จุดภาพจะมีเพียง 2 ค่าเท่านั้น คือ 1 แสดงว่าเป็นจุดภาพของตัวอักษร (ดำ) และ 0 แสดงว่าไม่ใช่จุดภาพ (ขาว) โดยขบวนการรับภาพของตัวอักษรเข้ามาที่มีลักษณะเป็นจุดภาพอาจเกิดสัญญาณรบกวนปะปนเข้ามากับข้อมูลภาพได้ ดังนั้นจากรูปโครงสร้างการรู้จำในรูปที่ 2.1 จะมีกระบวนการประมวลผลและปรับสัญญาณเบื้องต้นซึ่งทำหน้าที่กำจัดสัญญาณรบกวนและปรับปรุงคุณภาพของตัวอักษร (Noise Reduction & Contrast) สำรวจพื้นที่ของภาพ (Split) การทำตัวอักษรให้บาง (Thinning) และการลบกิ่งและการขลิบปลาย (Delete Branch & Clip End) ดังแสดงการทำงานในรูปที่ 2.2



รูปที่ 2.2 แสดงการทำงานของการปรับปรุงคุณภาพของรูปข้อมูล

2.2.1 การกำจัดสัญญาณรบกวนและการปรับปรุงคุณภาพของตัวอักษร (Noise Reduction & Contrast)

2.2.1.1 การกำจัดสัญญาณรบกวน (Noise Reduction)

จะเป็นกรรมวิธีในการกำจัดจุดภาพที่ไม่ต้องการออกจากภาพข้อมูลของตัวอักษรหรือเพิ่มจุดภาพที่ขาดหายไป โดยที่จะต้องไม่ทำให้ข้อมูลหลักของภาพตัวอักษรมีการเปลี่ยนแปลงรูปร่างไปจากเดิม สัญญาณรบกวนที่พบในรูปภาพข้อมูลตัวอักษรที่สามารถแก้ไขได้คือสัญญาณรบกวนที่มีลักษณะเป็นรูเดี่ยวและจุดเดี่ยวในบริเวณตัวอักษร แสดงในรูปที่ 2.3

ก) สัญญาณรบกวนแบบรูเดี่ยว

ข) สัญญาณรบกวนแบบจุดเดี่ยว

รูปที่ 2.3 แสดงการกำจัดสัญญาณรบกวน

วิธีการกำจัดสัญญาณรบกวนทั้งสองชนิดจะใช้หน้าต่างขนาด 3x3 จุดภาพดังแสดงในรูปที่ 2.4 ตรวจสอบกับทุกๆจุดภาพของข้อมูลภาพตัวอักษร โดยที่จะมีเงื่อนไขของการตรวจสอบดังนี้

1. ถ้า P เป็นจุดภาพที่กำลังพิจารณาและจุดภาพข้างเคียงของ P ทั้ง 8 ตำแหน่ง (X0 – X7) ไม่มีจุดภาพใดๆเลย แสดงว่า P เป็นจุดภาพเดี่ยวให้ทำการลบจุดภาพนี้ออกไป

2. ถ้า P เป็นจุดภาพที่กำลังพิจารณาและจุดภาพข้างเคียงของ P ทั้ง 8 ตำแหน่ง (X0 – X7) เป็นจุดภาพทั้ง 8 ตำแหน่ง แสดงว่า P เป็นรูจุดภาพเดี่ยวให้ทำการเปลี่ยนจุดP นี้เป็นจุดภาพ

X3	X2	X1
X4	P	X0
X5	X6	X7

รูปที่ 2.4 แสดงหน้าต่างที่ใช้กำจัดสัญญาณรบกวนและรูปที่ได้จากการกำจัดสัญญาณรบกวน

2.2.1.2 การปรับปรุงคุณภาพของตัวอักษร (Contrast)

เนื่องจากข้อมูลของภาพตัวอักษรที่รับจากเครื่องสแกนเนอร์ จะมีความเข้มของสีดำบนตัวอักษรไม่เท่ากันทั้งหมดของภาพตัวอักษรแสดงดังรูปที่ 2.5 ก) จึงต้องทำการปรับปรุงคุณภาพของตัวอักษรให้มีความเข้มของจุดภาพเท่ากันทุกจุดแสดงดังรูปที่ 2.5 ข) โดยที่จะให้ความสำคัญกับจุดที่พิจารณา ขั้นตอนแรก ที่จุดใดๆนำค่าสี RGB (Red , Green , Blue) มาบวกกันแล้วหาร (div) ด้วย 3 นำค่าที่ได้ไปลบออกจาก 255 (ความสว่างสูงสุด) ดังนั้นจุดภาพสีดำจะมีค่าผลลัพธ์จากการลบมาก และจุดภาพสีขาวจะมีค่าผลลัพธ์จากการลบต่ำ ขั้นตอนที่สอง จุดที่พิจารณาจะให้มีค่าความสำคัญเป็น 10 เท่าของจุดรอบข้างที่พิจารณา จากนั้นนำจุดที่อยู่รอบๆจุดที่พิจารณาจากรูปที่ 2.4 (X0 – X7) มาบวกกันรวมทั้งจุดที่พิจารณาด้วยแล้วหารด้วยจำนวนตัวที่เราเอามาบวกกันรวมถึงค่าน้ำหนักด้วย ผลลัพธ์ที่ได้เราจะเอามาเทียบกับค่าเฉลี่ยทั้งภาพ เทียบโดยที่ต้องมีค่ามากกว่าหรือ

เท่ากับ 3 เท่า ถ้าผลลัพธ์ที่ได้มีค่ามากกว่าค่าเฉลี่ยทั้งภาพก็ให้จุดที่พิจารณานั้นเป็นจุดภาพสีดำ แต่ถ้าผลลัพธ์ที่ได้น้อยกว่าค่าเฉลี่ยทั้งภาพก็ให้จุดที่พิจารณานั้นเป็นจุดภาพสีขาว

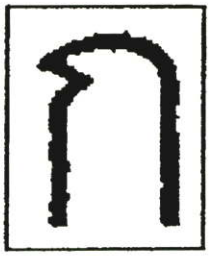
ก) แสดงรูปภาพตัวอักษรที่รับจากเครื่องสแกนเนอร์

ข) แสดงรูปการปรับปรุงคุณภาพของตัวอักษร (Contrast)

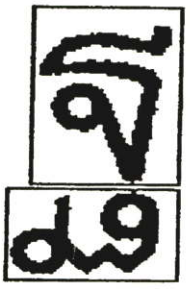
รูปที่ 2.5 แสดงการปรับปรุงคุณภาพของตัวอักษร (Contrast)

2.2.2 การสำรวจพื้นที่ของภาพ (Split)

เมื่อได้ผ่านกระบวนการปรับปรุงคุณภาพของตัวอักษรแล้วจะเข้าสู่กระบวนการสำรวจพื้นที่จากภาพ โดยจะสำรวจดูว่าถ้ามีพื้นที่สีดำของตัวอักษรอยู่ติดกัน 1 ส่วน ก็ให้นับว่าตัวอักษรนั้นมีพื้นที่เป็น 1 ส่วน เช่น ก กับ จ แต่ถ้าพบว่าในข้อมูลของภาพตัวอักษรนั้นมีพื้นที่สีดำอยู่ติดกัน 2 ส่วนก็ให้นับว่าตัวอักษรนั้นมีพื้นที่เป็น 2 ส่วน เช่น ฐ กับ ญ แสดงในรูปที่ 2.6



ก) ภาพที่สำรวจพื้นที่ได้ 1 ส่วน



ข) ภาพที่สำรวจพื้นที่ได้ 2 ส่วน

รูปที่ 2.6 แสดงภาพการสำรวจพื้นที่ที่แบ่งออกเป็นส่วน

2.2.3 การทำตัวอักษรให้บาง (Thinning)

เป็นกระบวนการเปลี่ยนแปลงภาพข้อมูลให้เหลือเฉพาะเส้นโครงสร้างของภาพอักษร (Skeleton) เท่านั้น โดยข้อมูลของภาพที่ผ่านกระบวนการทำตัวอักษรให้บางนั้นจะเหลือความกว้างของเส้นตัวอักษร 1 จุดภาพ วิธีการทำตัวอักษรให้บางนั้นมีอยู่ด้วยกันหลายวิธีขึ้นอยู่กับลักษณะของข้อมูล เช่น การทำให้บางแบบเนื่องกัน (Sequential Processing Thinning) การทำให้บางแบบขนาน (Parallel Processing Thinning) ฯลฯ เนื่องจากภาพของตัวอักษรที่ผ่านกระบวนการทำให้บางแล้วจะต้องไม่ทำให้มีการขาดตอนของลายเส้นตัวอักษร ดังนั้นจึงเลือกใช้วิธีการทำให้บางแบบเนื่องกันหรือ classical thinning ของ Pavlidis ในการทำภาพตัวอักษรให้บาง เนื่องจากวิธีดังกล่าวไม่ทำให้เกิดการขาดตอนของลายเส้นตัวอักษรและมีโครงร่างคล้ายกับตัวอักษรต้นแบบค่อนข้างมาก แต่ข้อเสียของวิธีดังกล่าวคือ จะใช้เวลาในการประมวลผลค่อนข้างมากและไม่สามารถทำให้ปรากฏรอยหยักกับหัวของตัวอักษรภาษาไทยบางตัวได้ เช่น ฃ ฅ ฆ ซ ฑ ฒ ต ซึ่งในงานวิทยานิพนธ์นี้ได้นำวิธีทำให้บางโดยใช้ SPTA (Safe Point Thinning Algorithm) (รายละเอียดดูในภาคผนวก ก) เข้ามาร่วมประยุกต์ใช้เพื่อจะปรับปรุงการทำให้บางวิธีเดิมให้ปรากฏรายละเอียดของภาพตัวอักษรชัดเจนมากขึ้น ภาพข้อมูลตัวอักษรก่อนการทำตัวอักษรให้บางและผลการทำตัวอักษรให้บางแสดงในรูปที่ 2.7

ก) แสดงรูปก่อนการทำตัวอักษรให้บาง ข) แสดงรูปหลังจากการทำตัวอักษรให้บาง

รูปที่ 2.7 แสดงรูปก่อนและหลังการทำตัวอักษรให้บาง

2.2.4 การลบกิ่งและการขลิบปลาย (Delete Branch & Clip End)

การลบกิ่งที่เกิดบนตัวอักษร (delete Branch) และการขลิบปลายของตัวอักษร (Clip End) ที่ไม่ต้องการ ได้แสดงดังรูปที่ 2.8(ก) และรูปที่ 2.8(ข)

ก. แสดงรูปก่อนการ Delete Branch & Clip End

ข. แสดงรูปหลังการ Delete Branch & Clip End

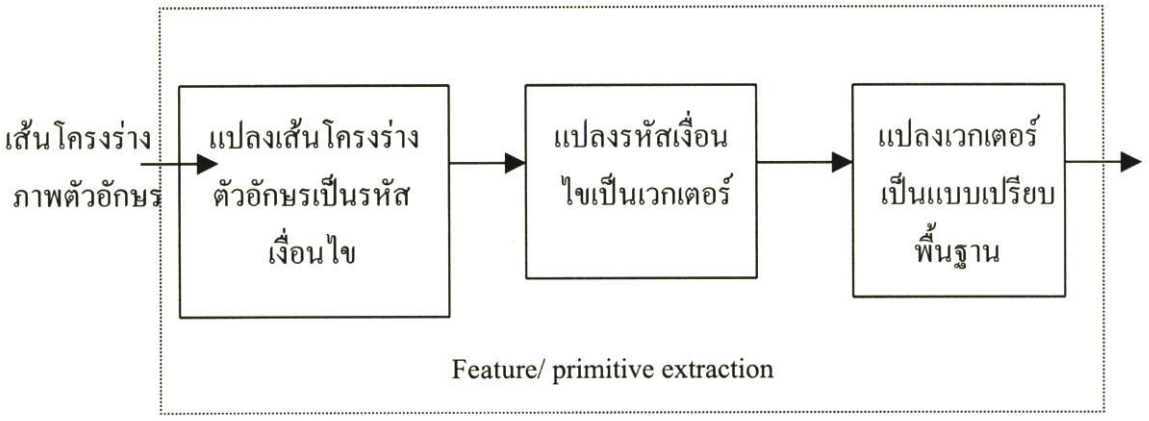
รูปที่ 2.8 แสดงรูปก่อนและหลังการ Delete Branch & Clip End

จากรูปที่ 2.8 การ Delete Branch มีแนวคิดดังนี้คือ การใช้หน้าต่าง 3x3 ในการตรวจดูว่ามีจุดไหนเป็นจุดปลาย โดยจะพิจารณาว่าไม่มีจุดอยู่รอบตัวจุดที่พิจารณาเกิน 1 จุดภาพ แล้วแกะรอยไปตามเส้นทางจากจุดหนึ่งไปยังจุดที่ติดกัน ถ้าเจอจุดที่มีจุดรอบข้างมากกว่าหรือเท่ากับ 3 จุดให้ถือว่าสิ้นสุดการแกะรอย แล้วพิจารณาว่าเส้นดังกล่าวมีขนาดยาวเกินกว่าที่กำหนดได้หรือไม่ ถ้าไม่เกินก็ให้ทำการลบเส้นนั้นทิ้งไป เพื่อตัดเส้นที่ไม่ใช่เส้นจริงของภาพที่ทำให้บางแล้วออกไปให้หมด ส่วน Clip End จะลบตรงบริเวณที่เป็นส่วนปลายของตัวอักษร จะใช้หน้าต่างขนาด 3x3 ในการตรวจดูว่ามีจุดไหนเป็นจุดปลายเช่นเดียวกับ Delete Branch โดยจากจุดปลายสุดลบไป $\frac{1}{12}$ ของความสูงจริงที่ได้จากการทำให้บาง

2.3 การวิเคราะห์หาลักษณะสำคัญของตัวอักษร (Feature / Primitive extraction)

หลังจากข้อมูลภาพของตัวอักษรผ่านกระบวนการปรับปรุงคุณภาพของภาพตัวอักษรแล้ว จะได้ข้อมูลของภาพตัวอักษรที่เหลือเฉพาะเส้นโครงร่างของภาพตัวอักษร (Skeleton) ข้อมูลของเส้นโครงร่างของภาพตัวอักษรนี้จะถูกนำมาหาลักษณะเด่นหรือลักษณะสำคัญ (Feature / Primitive extraction) เพื่อใช้เป็นข้อมูลแทนภาพของตัวอักษรแต่ละตัว และใช้เป็นข้อมูลที่ใช้ในการฝึกหัดระบบ และการรู้จำตัวอักษรต่อไป โดยวิธีของการหาลักษณะเด่นของภาพตัวอักษรนั้น จะนำภาพเส้นโครงร่างของภาพตัวอักษรที่มีความกว้างของลายเส้นตัวอักษร 1 จุดภาพ มาผ่านกระบวนการ

เข้ารหัสเงื่อนไข (Condition code) เมื่อได้จุดภาพที่เข้ารหัสเงื่อนไขแล้ว ก็จะนำรหัสเงื่อนไขเหล่านี้มาคำนวณแปลงเป็นเวกเตอร์ (Vector) โดยพิจารณาจากรหัสเงื่อนไขที่มีทิศทางเดียวกัน หลังจากที่แปลงข้อมูลของภาพตัวอักษรเป็นเวกเตอร์แล้ว จะนำเวกเตอร์เหล่านี้มาเปรียบเทียบกับรหัสแบบลูกโซ่ของฟรีแมน (Freeman chain code) เพื่อที่จะเปลี่ยนข้อมูลเวกเตอร์ให้อยู่ในรูปของแบบเปรียบหลัก และในขั้นตอนสุดท้ายของการหาลักษณะสำคัญของภาพตัวอักษรจะได้แบบเปรียบใดๆ ซึ่งใช้แทนข้อมูลภาพของตัวอักษรนั้น ดังแสดงโครงสร้างการทำงานในรูปที่ 2.9



รูปที่ 2.9 แสดงการหาลักษณะสำคัญของข้อมูลภาพเส้นโครงร่างตัวอักษร

2.3.1 การแปลงเส้นโครงร่างเป็นรหัสเงื่อนไข

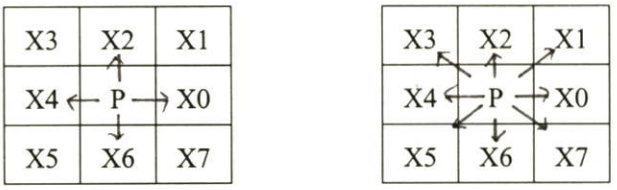
2.3.1.1 จุดภาพข้างเคียง

จุดภาพ P ที่มีคู่อันดับ (x , y) จะมีจุดข้างเคียงในแนวตั้งและแนวนอน 4 จุดภาพ ซึ่งมีคู่อันดับดังนี้ (x+1 , y) , (x-1 , y) , (x , y+1) , (x , y-1) เซตของจุดเหล่านี้เรียกว่า 4-neighbors ของจุด P เขียนแทนด้วย $N_4(p)$ ซึ่งแต่ละจุดจะมีระยะห่างจากจุด (x , y) หนึ่งหน่วย

จุดภาพข้างเคียงจุด P อีก 4 จุดภาพในแนวทแยง (Diagonal) จะมีคู่อันดับดังนี้ (x+1 , y+1) , (x+1 , y-1) , (x-1 , y+1) , (x-1 , y-1) เขียนแทนด้วย $N_D(p)$ ซึ่งจุดเหล่านี้ ถ้ารวมกับ 4- neighbors ข้างบนจะเรียกว่า 8- neighbors ของ P เขียนแทนด้วย $N_8(p)$

2.3.1.2 การคำนวณค่าตัวเลขต่อเนื่อง (Connectivity number)

ค่าตัวเลขต่อเนื่องเป็นคุณสมบัติทางโทโพโลยี (Topological Property) ของจุดภาพที่อยู่ติดกันในรูปแบบข้อมูลสองระดับ (binary step) ค่าตัวเลขต่อเนื่องนี้จะเป็นตัวเลขแสดงความสัมพันธ์ระหว่างจุดภาพข้างเคียง (Neighbors of a pixel)



รูปที่ 2.10 แสดงลักษณะของจุดภาพข้างเคียง

(ก) 4- neighbors pixel

(ข) 8- neighbors pixel

ค่าตัวเลขต่อเนื่องของจุดภาพข้างเคียงทั้ง 2 แบบ สามารถหาได้จากเงื่อนไขทาง Boolean ได้ดังนี้

$$
\begin{aligned}
NC4 = {} & (N[0] + N[2] + N[4] + N[6]) \\
& - (N[0] \times N[1] \times N[2]) \\
& - (N[2] \times N[3] \times N[4]) \\
& - (N[4] \times N[5] \times N[6]) \\
& - (N[6] \times N[7] \times N[0])
\end{aligned} \tag{2.1}
$$

กำหนดให้ จุดภาพที่มีสีดำ = 1

จุดภาพที่มีสีขาว = 0

$$
\begin{aligned}
NC8 = {} & (N[0] + N[2] + N[4] + N[6]) \\
& - (N[0] \times N[1] \times N[2]) \\
& - (N[2] \times N[3] \times N[4]) \\
& - (N[4] \times N[5] \times N[6]) \\
& - (N[6] \times N[7] \times N[0])
\end{aligned} \tag{2.2}
$$

กำหนดให้ จุดภาพที่มีสีดำ = 0

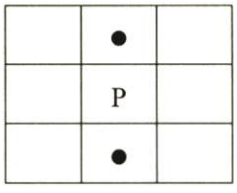
จุดภาพที่มีสีขาว = 1

ผลที่ได้จากการคำนวณโดยสมการ (2.1) หรือ (2.2) จะได้เป็นเลขจำนวนเต็มบวกที่มีค่าอยู่ระหว่าง 0 ถึง 4 ค่าตัวเลขต่อเนื่องที่คำนวณได้ทั้งสองแบบจะบอกถึงความสัมพันธ์ระหว่างจุดภาพดังตารางที่ 2.1

ตารางที่ 2.1 แสดงความสัมพันธ์ระหว่างจุดภาพ

ค่าตัวเลขต่อเนื่อง	จุดอิสระหรือจุดภายใน	จุดปลาย		จุดต่อเนื่อง					จุดแยก				จุดตัด	
NC4	0	1	0	1	2	2	2	0	3	0	1	2	4	0
NC8	0	1	1	2	2	2	1	2	1	3	3	2	0	4

ตัวอย่างการคำนวณ ค่าตัวเลขต่อเนื่อง

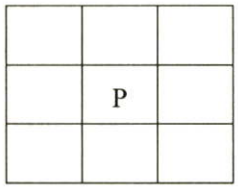


NC4 = (0+1+0+1) – (0*0*1) – (1*0*0) – (0*0*1) – (1*0*0)

= 2

NC8 = (1+0+1+0) – (1*1*0) – (0*1*1) – (1*1*0) – (0*1*1)

= 2

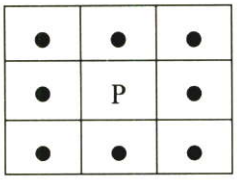


NC4 = (0+0+0+0) – (0*0*0) – (0*0*0) – (0*0*0) – (0*0*0)

= 0

NC8 = (1+1+1+1) – (1*1*1) – (1*1*1) – (1*1*1) – (1*1*1)

= 0

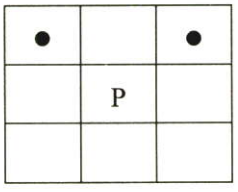


NC4 = (1+1+1+1) – (1*1*1) – (1*1*1) – (1*1*1) – (1*1*1)

= 0

NC8 = (0+0+0+0) – (0*0*0) – (0*0*0) – (0*0*0) – (0*0*0)

= 0

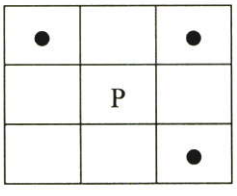


NC4 = (0+0+0+0) – (0*1*0) – (0*1*0) – (0*0*0) – (0*0*0)

= 0

NC8 = (1+1+1+1) – (1*0*1) – (1*0*1) – (1*1*1) – (1*1*1)

= 2



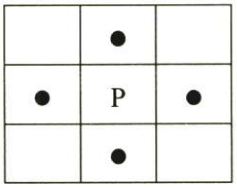
NC4 = (0+0+0+0) – (0*1*0) – (0*1*0) – (0*0*0) – (0*1*0)

= 0

NC8 = (1+1+1+1) – (1*0*1) – (1*0*1) – (1*1*1) – (1*0*1)

= 3

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

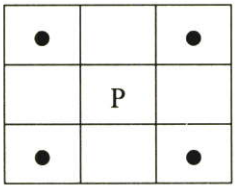


NC4 = (1+1+1+1) – (1*0*1) – (1*0*1) – (1*0*1) – (1*0*1)

= 4

NC8 = (0+0+0+0) – (0*1*0) – (0*1*0) – (0*1*0) – (0*1*0)

= 0



NC4 = (0+0+0+0) – (0*1*0) – (0*1*0) – (0*1*0) – (0*1*0)

= 0

NC8 = (1+1+1+1) – (1*0*1) – (1*0*1) – (1*0*1) – (1*0*1)

= 4

วิธีการเข้ารหัสเงื่อนไข (Condition code) โดยใช้ตารางขนาด 3x3 มีหลักการพิจารณาดังนี้

1. ถ้า X เป็นจุดภาพและมีจุดภาพรอบข้าง 1 จุดภาพ แสดงว่าเป็นจุดปลาย (end point) ของภาพตัวอักษรให้แทนจุดนั้นด้วยรหัสเงื่อนไขเลขลบ

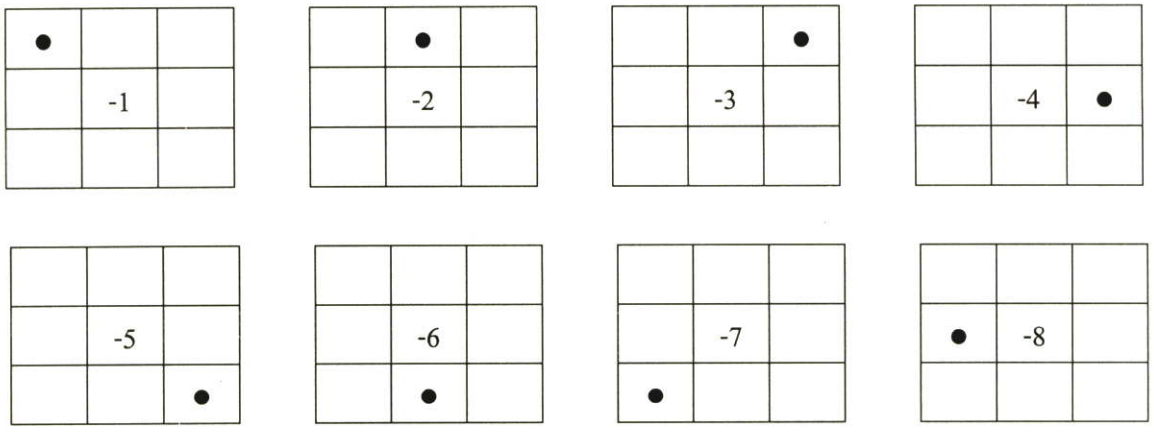
2. ถ้า X เป็นจุดภาพและมีจุดภาพรอบข้าง 2 จุดภาพ แสดงว่าเป็นจุดต่อเนื่องของภาพตัวอักษรให้แทนจุดนั้นด้วยรหัสเงื่อนไขเลขบวก

3. ถ้า X เป็นจุดภาพและมีจุดภาพรอบข้าง 3 จุดภาพ แสดงว่าเป็นจุดทางแยก (junction point) ของภาพตัวอักษรให้แทนจุดนั้นด้วยรหัสเงื่อนไขเลขลบ

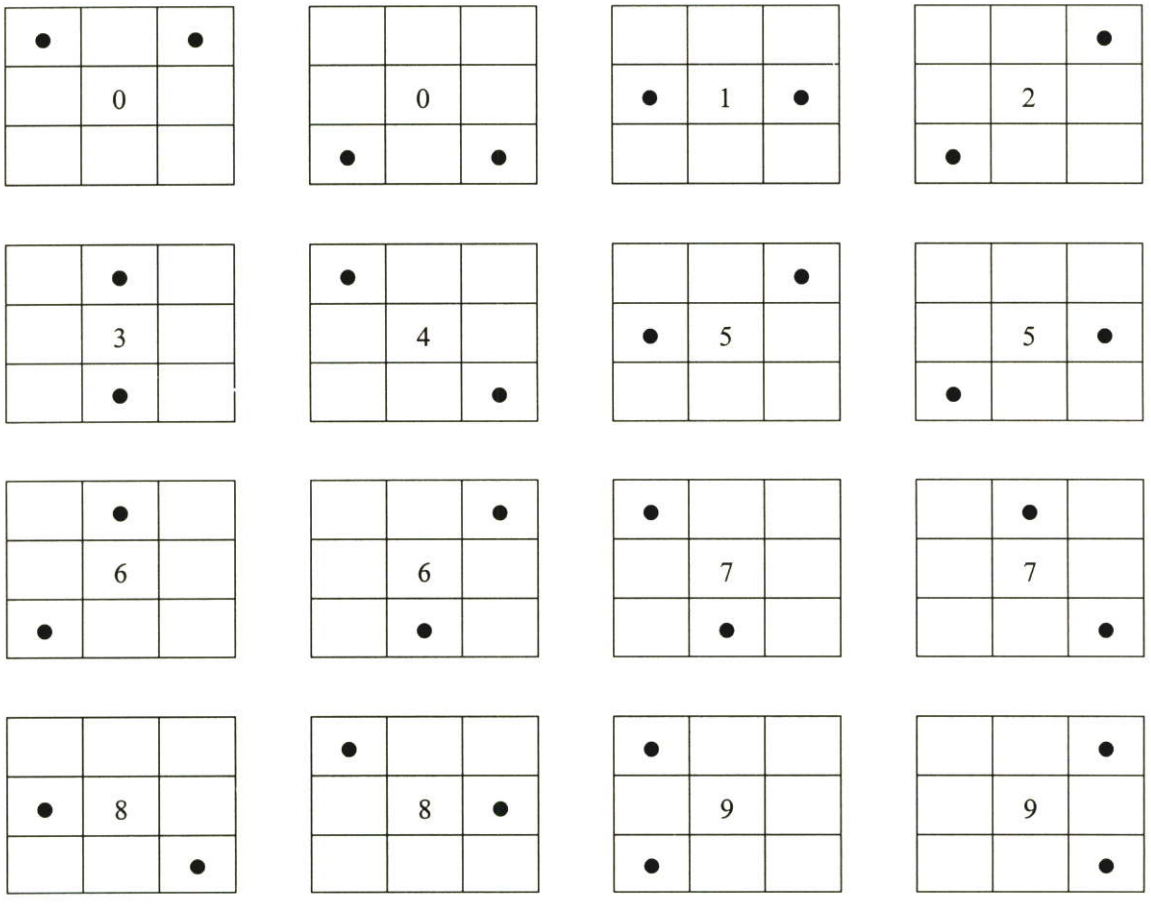
4. ถ้า X เป็นจุดภาพและมีจุดภาพรอบข้าง 4 จุดภาพ แสดงว่าเป็นจุดตัด (crossing point) ของภาพตัวอักษรให้แทนจุดนั้นด้วยรหัสเงื่อนไขเลขลบ

5. ถ้า X ไม่ใช่จุดภาพหรือเป็นจุดที่มีข้อกำหนดนอกเหนือจากที่ระบุไว้ให้แทนจุดภาพนั้นด้วยรหัสเงื่อนไข 10

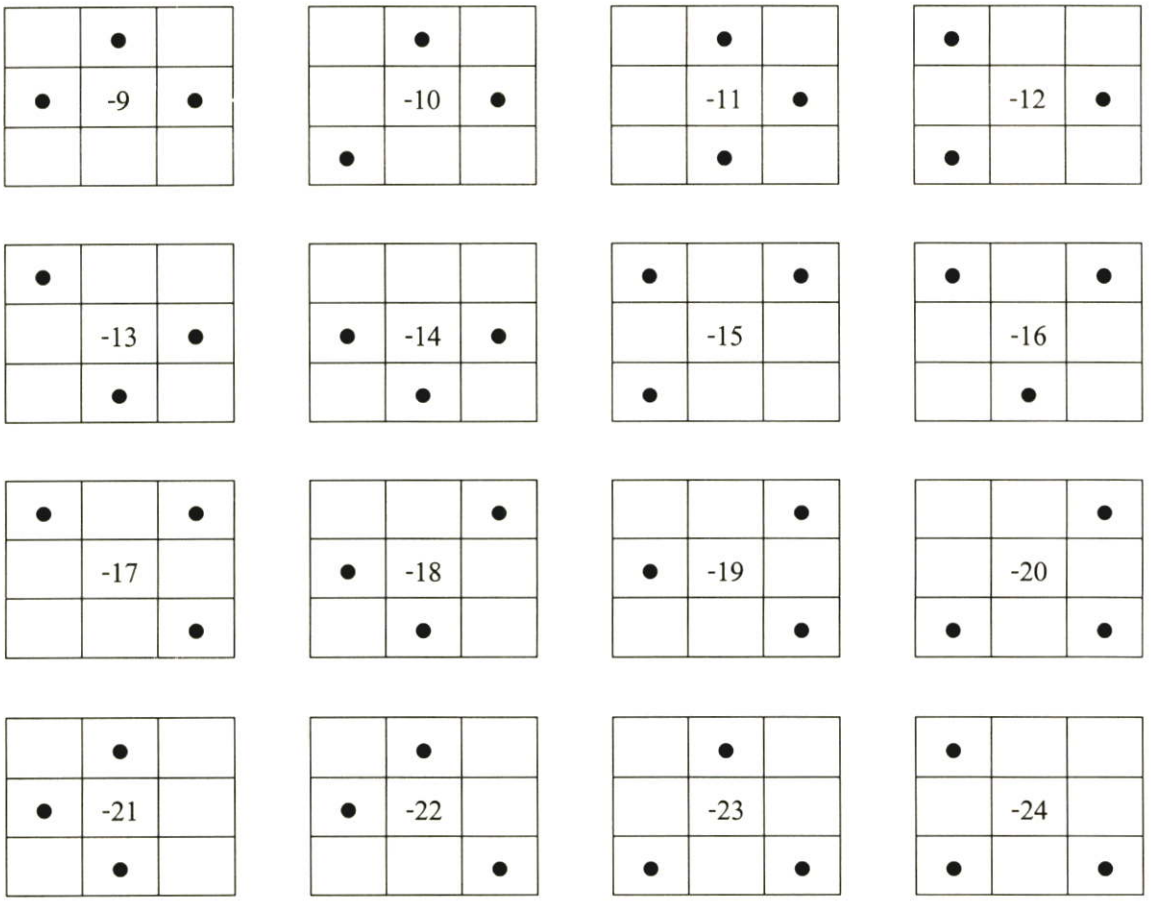
47704



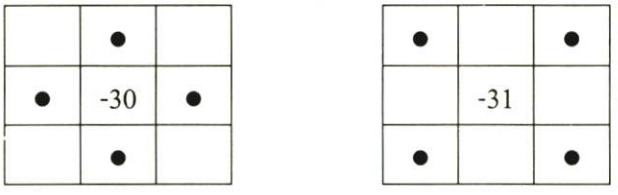
รูปที่ 2.11(ก) แสดงรหัสเงื่อนไขที่เป็นลบ (จุดปลาย)



รูปที่ 2.11(ข) แสดงรหัสเงื่อนไขที่เป็นบวก (จุดต่อเนื่อง)

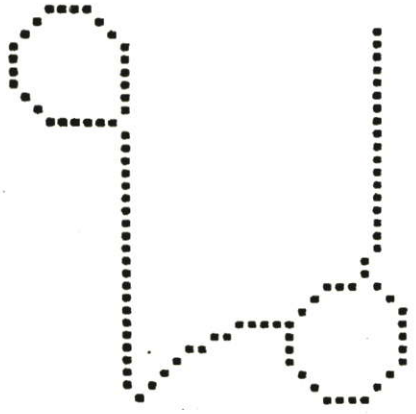


รูปที่ 2.11(ค) แสดงรหัสเงื่อนไขที่เป็นลบ (จุดทางแยก)

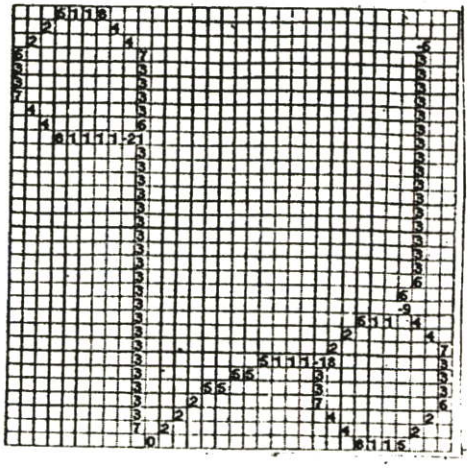


รูปที่ 2.11(ง) แสดงรหัสเงื่อนไขที่เป็นลบ (จุดตัด)

หลังจากที่ผ่านกระบวนการเข้ารหัสเงื่อนไข (Condition code) แล้วจะได้ข้อมูลของภาพตัวอักษรดังรูปที่ 2.12 ซึ่งแสดงรหัสเงื่อนไขของจุดภาพตัวอักษรในลักษณะต่างๆ เช่น ทางแยกของตัวอักษร , จุดตัดของตัวอักษร , จุดปลายของตัวอักษร และจุดต่อเนื่องของตัวอักษร



(ก) ภาพข้อมูลที่ถูกทำให้บาง



(ข) ภาพข้อมูลที่ผ่านการเข้ารหัสเงื่อนไข

รูปที่ 2.12 การเข้ารหัสเงื่อนไข

2.3.2 การแปลงรหัสเงื่อนไข (Condition code) ให้เป็นเวกเตอร์

จากรหัสเงื่อนไขชนิดต่างๆดังกล่าว สามารถที่จะจำแนกตำแหน่งต่างๆของจุดภาพนั้นว่าทำมุมในทิศทางใดตามตารางที่ 2.2 โดยรหัสเงื่อนไข 1 ถึง 4 เรียกว่ารหัสเงื่อนไขหลักซึ่งจะทำมุมเป็นทวีคูณของ 45 องศา รหัสเงื่อนไข 5 ถึง 8 เรียกว่ารหัสเงื่อนไขรองซึ่งจะทำมุมเป็นทวีคูณของ 30 องศา รหัสเงื่อนไข 0 และ 9 เป็นจุดหักเหของเส้นโครงร่างของภาพ

ตารางที่ 2.2 แสดงมุมของรหัสเงื่อนไข

รหัสเงื่อนไข	มุม (องศา)
1	0
2	45
3	90
4	135
5	30
6	60
7	120
8	150

ในกรณีของรหัสเงื่อนไขที่มีการเรียงของรหัสเงื่อนไขหลักและรหัสเงื่อนไขรองคละกันไป จะได้มุมของเส้นโครงร่างของภาพดังกล่าวในตารางที่ 2.3

ตารางที่ 2.3 แสดงมุมของรหัสเงื่อนไขหลักและรหัสเงื่อนไขรองคละกัน

รหัสเงื่อนไขเรียงต่อเนื่อง	มุม (องศา)
1 - 5 - 1	$0 < \theta < 30$
2 - 5 - 2	$30 < \theta < 45$
2 - 6 - 2	$45 < \theta < 60$
3 - 6 - 3	$60 < \theta < 90$
3 - 7 - 3	$90 < \theta < 120$
4 - 7 - 4	$120 < \theta < 135$
4 - 8 - 4	$130 < \theta < 150$
1 - 8 - 1	$150 < \theta < 180$

เมื่อสามารถคำนวณหาทิศทางของเส้นโครงร่างของตัวอักษรได้แล้ว ก็จะนำทิศทางเส้นโครงร่างตัวอักษรซึ่งคำนวณได้จากการรวมเอาจุดภาพที่มีรหัสเงื่อนไขเหมือนกัน หรือเป็นเงื่อนไขที่ต่อเนื่องกันรวมเข้าด้วยกันเป็นเวกเตอร์ โดยที่เวกเตอร์ที่ใช้งานนั้นมีอยู่ 2 ชนิด คือ

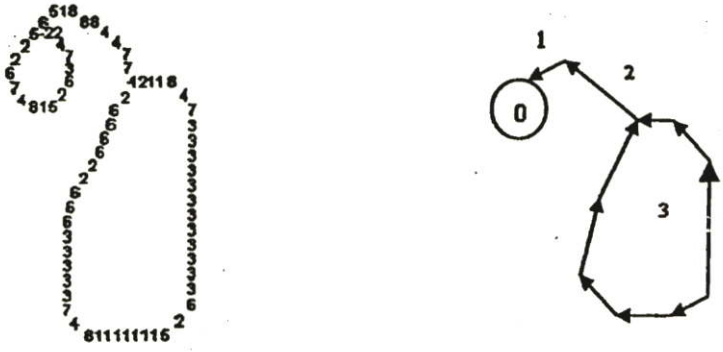
เวกเตอร์เส้นตรง ใช้ในการแทนจุดภาพตัวอักษรที่มีทิศทางเป็นเส้นตรง โดยที่จะพิจารณารวมเอาทิศทางของจุดภาพตัวอักษรที่มีทิศทางลักษณะเดียวกัน รวมกันเข้าเป็นเวกเตอร์เส้นตรงเดียวกัน

เวกเตอร์วงกลม ใช้ในการแทนจุดภาพที่มีตำแหน่งเริ่มต้นและตำแหน่งสุดท้ายที่วนมาบรรจบในตำแหน่งเดียวกัน ซึ่งอาจเป็นได้ทั้งเวกเตอร์วงกลมอิสระ หรือเวกเตอร์วงกลมที่มีเวกเตอร์อื่นมาต่อเชื่อมด้วย โดยจุดประสงค์หลักในการใช้เวกเตอร์วงกลมเพื่อที่จะใช้แทนส่วนหัวของตัวอักษร ดังนั้นเวกเตอร์วงกลมจึงถูกกำหนดให้มีจุดเชื่อมต่อเพียงจุดเดียวที่จะสามารถเชื่อมต่อกับเวกเตอร์เส้นตรง

การสร้างเวกเตอร์วงกลม จะสามารถสร้างจากเวกเตอร์เส้นตรง โดยพิจารณาจากจุดทางแยกที่ 3 ทางแยก หรือจุดตัด และมี 2 ทางแยกที่สามารถเชื่อมต่อถึงกันและกัน โดยที่ไม่มีจุดทางแยกอื่นๆอีกในระหว่างเส้นทางที่เชื่อมต่อถึงกัน แสดงว่าทางแยกทั้ง 2 นั้นมีเส้นทางการเชื่อมต่อเป็นวงกลม และทางแยกที่เหลืออีก 1 เส้นทางจะเป็นจุดเชื่อมต่อของวงกลมกับเวกเตอร์อื่น ดังแสดงในรูปที่ 2.13

(ก) ก่อนการทำเป็นเวกเตอร์วงกลม (ข) หลังจากการแปลงเป็นเวกเตอร์วงกลม

รูปที่ 2.13 แสดงการแปลงเวกเตอร์เส้นตรงเป็นเวกเตอร์วงกลม



(ก) ข้อมูลภาพที่เข้ารหัสเงื่อนไข (ข) เวกเตอร์ของภาพตัวอักษร

รูปที่ 2.14 แสดงการแปลงข้อมูลภาพตัวอักษรที่เข้ารหัสเงื่อนไขเป็นเวกเตอร์

จากรูปที่ 2.14 จะเห็นว่ารูป ช ไม่มีหางเนื่องมาจากการ Delete Branch เมื่อเข้ากระบวนการเข้ารหัสเงื่อนไขจะเห็นว่ารหัสเงื่อนไขตรงส่วนหางก็จะไม่มี และเมื่อเปลี่ยนเป็นเวกเตอร์ก็จะเห็นว่าไม่มีเวกเตอร์ที่หางของ ช กระบวนการเปลี่ยนให้เป็นแบบเปรียบ (Primitive) จะได้รูป ช จะมี 2 หัว แต่ก็ไม่ทำให้มีปัญหา เพราะว่าไม่มีอักษรตัวใดที่เหมือนนอกจาก ซ ซึ่งสามารถแยกความต่างระหว่าง ช กับ ซ ได้ตรงที่ ซ มีหัวหยักได้

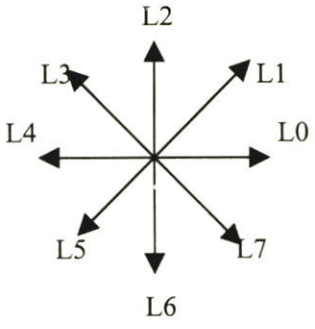
2.3.3 การเปลี่ยนเวกเตอร์ให้เป็นแบบเปรียบพื้นฐาน (Primitive)

เนื่องจากข้อมูลเวกเตอร์ที่ใช้แทนข้อมูลภาพตัวอักษรนั้น จะเป็นข้อมูลของตำแหน่งจุดเริ่มต้น และจุดปลายของเวกเตอร์ ดังนั้น เพื่อเป็นการลดขนาดข้อมูลที่ใช้แทนภาพตัวอักษร จึงนำข้อมูลเวกเตอร์ของภาพข้อมูลตัวอักษรมาเปลี่ยนให้อยู่ในรูปของแบบเปรียบพื้นฐาน โดยลักษณะของแบบเปรียบที่ใช้แทนข้อมูลเวกเตอร์ของภาพตัวอักษรมีดังนี้

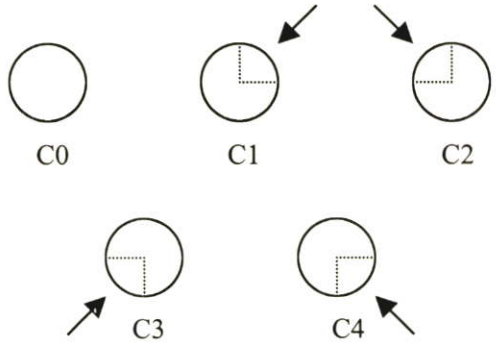
- แบบเปรียบเส้นตรง มีอยู่ด้วยกัน 8 รูปแบบ ดังรูปที่ 2.15 (ก) ซึ่งมีลักษณะเหมือนกับรหัสลูกโซ่ฟรีแมน โดยที่แบบเปรียบแต่ละรูปแบบจะใช้แทนเวกเตอร์เส้นตรงที่ทำมุมครอบคลุมพื้นที่

45 องศา เช่น แบบเปรียบ L1 จะถูกใช้แทนเวกเตอร์เส้นตรงที่ทำมุม 0 ถึง 45 องศา แบบเปรียบ L2 จะถูกใช้แทนเวกเตอร์เส้นตรงที่ทำมุม 45 ถึง 90 องศา เป็นต้น

- แบบเปรียบวงกลม มีอยู่ด้วยกัน 5 รูปแบบ ดังรูปที่ 2.15(ข) โดยแบบเปรียบแรก (C0) เป็นวงกลมที่ไม่มีจุดเชื่อมต่อกับเวกเตอร์อื่นๆ ใช้แทนเวกเตอร์วงกลมที่ไม่มีการเชื่อมต่อกับเวกเตอร์ชุดอื่นเลย ในส่วนของแบบเปรียบวงกลมที่เหลือ (C1 , C2 ,C3 , C4) จะให้เวกเตอร์วงกลมที่มีจุดเชื่อมต่อกับเวกเตอร์อื่นๆ โดยมีพื้นที่ของการเชื่อมต่ออยู่ในบริเวณพื้นที่แต่ละ 45 องศา



(ก) แบบเปรียบเส้นตรง



(ข) แบบเปรียบวงกลม

รูปที่ 2.15 แสดงรูปแบบของแบบเปรียบหลัก (primitive)

2.4. การเปรียบเทียบโดยการวิเคราะห์ทาง Feature

2.4.1 ลักษณะเด่นที่ใช้ในการเปรียบเทียบตัวอักษร

1. จำนวนส่วนของตัวอักษร (Parts)
2. อัตราความกว้างเทียบกับความสูง
3. จำนวนหัวของตัวอักษร
4. จำนวนปลายของตัวอักษร
5. ลักษณะหัวแรกที่พบ
6. ตำแหน่งของหัวแรก หรือปลายแรกที่พบ
7. ตำแหน่งของหัวที่สอง หรือปลายที่สองที่พบ
8. เวกเตอร์ทิศทาง

จำนวนส่วนของตัวอักษร เป็นการสำรวจว่าตัวอักษรแต่ละตัวนั้นๆมีพื้นที่สีดำอยู่ติดกันกี่ส่วน ถ้ามีพื้นที่สีดำอยู่ติดกัน 1 ส่วน เช่น ก , จ ก็ให้นับว่าเป็นตัวอักษรที่มี 1 ส่วน แต่ถ้าพบว่ามีพื้นที่สีดำอยู่ติดกัน 2 ส่วน เช่น ฐ , ญ ก็ให้นับว่าเป็นตัวอักษรที่มี 2 ส่วน ดังรูป

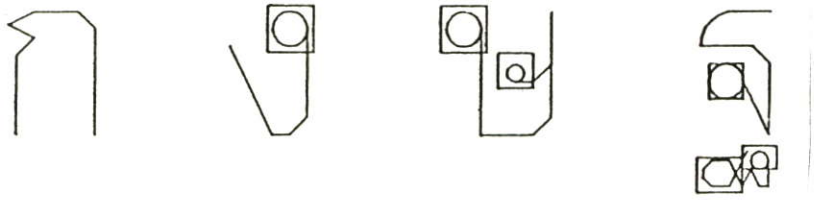
(ก) (ข)

รูปที่ 2.16 แสดงส่วนของตัวอักษร

(ก) ตัวอักษรที่มี 1 ส่วน (ข) ตัวอักษรที่มี 2 ส่วน

อัตราความกว้างเทียบกับความสูง เป็นการแยกตัวอักษรที่มีลักษณะคล้ายกันมากๆ เช่น ข กับ บ โดยเอาความกว้างจริงหารด้วย 4 แล้วนำมาหารกับความสูงจริงของตัวอักษรที่ทำให้บางแล้วและเลือกเฉพาะส่วนที่ 1 เท่านั้น ผลลัพธ์ที่ได้จะเป็นอัตราส่วนระหว่างความกว้างเทียบกับความสูงของตัวอักษรที่บางแล้ว โดยให้ความสูงมีค่าเป็น 4 ส่วน

จำนวนหัวของตัวอักษร เป็นการสำรวจหัวของตัวอักษรว่า ตัวอักษรแต่ละตัวนั้นๆมีหัวอยู่ทั้งหมดกี่หัว หัวในที่นี้หมายถึง เวกเตอร์วงกลมปิดที่อาจจะมีจุดเชื่อมต่อหรือไม่ก็ได้ ดังรูป



(ก) (ข) (ค) (ง)

รูปที่ 2.17 แสดงจำนวนหัวของตัวอักษร

(ก) ตัวอักษรที่ไม่มีหัว (ข) ตัวอักษรที่มี 1 หัว

(ค) ตัวอักษรที่มี 2 หัว (ง) ตัวอักษรที่มี 3 หัว

จำนวนปลายของตัวอักษร เป็นการสำรวจตัวอักษรแต่ละตัวนั้นๆว่ามีปลายทั้งหมดกี่ปลาย ในการตรวจว่าจุดไหนเป็นจุดปลาย ทำได้โดยพิจารณาจำนวนจุดภาพอยู่รอบจุดที่ต้องการตรวจสอบว่ามีไม่เกิน 1 จุดภาพ

(ก) (ข)

รูปที่ 2.18 แสดงจำนวนปลายของตัวอักษร
(ก) ตัวอักษรที่มี 1 ปลาย (ข) ตัวอักษรที่มี 2 ปลาย

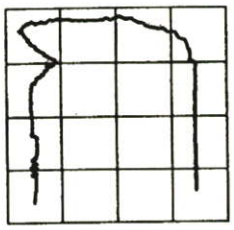
ลักษณะหัวแรกที่พบ เป็นการสำรวจดูว่าหัวแรกที่พบของตัวอักษรแต่ละตัวเป็นลักษณะใด เมื่อเทียบกับแบบเปรียบวงกลมทั้ง 5 แบบ ดังรูปที่ 2.15

ตำแหน่งของหัวแรกหรือปลายแรก และ ตำแหน่งของหัวที่สองหรือปลายที่สอง

เนื่องจากการรู้จำตัวอักษรในงานวิจัยนี้เป็นการรู้จำตัวอักษรทีละ 1 ตัวอักษร จึงไม่สามารถกำหนดเส้นบรรทัดของตัวอักษรได้ ดังนั้น วิธีการหาตำแหน่งหัวแรกหรือปลายแรก และหัวที่สองหรือปลายที่สองของตัวอักษร จึงกระทำได้โดยนำเอาความสูงจริงของส่วนที่ 1 มาหารด้วย 4 และเอาความกว้างจริงของส่วนที่ 1 มาหารด้วย 4 จะได้ตารางที่ถูกแบ่งเป็น 16 ช่อง ตำแหน่งหัวจะพิจารณาจากจุดที่มีจุดภาพมากกว่าหรือเท่ากับ 3 จุดภาพ ส่วนตำแหน่งปลายจะพิจารณาจุดที่มีจุดภาพไม่เกิน 1 จุดภาพ นำจุดที่ได้ที่แกน x หารด้วย RealWidth/4 และที่แกน y หารด้วย RealHight/4 ก็จะได้ตำแหน่งหัวหรือปลายแรก และตำแหน่งหัวหรือปลายที่สอง โดยสแกนภาพจากบนลงล่าง ตำแหน่งหัวหรือปลายอันแรก ตำแหน่งหัวหรือปลายที่สอง ดังรูปที่ 2.19

00	10	20	30
01	11	21	31
02	12	22	32
03	13	23	33

รูปที่ 2.19 แสดงหน้าต่างการอ้างอิงตำแหน่งที่จุดหัวหรือจุดปลายใดๆ



รูปที่ 2.20 แสดงตำแหน่งหัวแรกหรือปลายแรก และตำแหน่งหัวหรือปลายที่ 2

จากรูปที่ 2.19 จะเห็นว่าตำแหน่งปลายแรกเมื่อเทียบกับรูปที่ 2.20 จะอยู่ในช่องที่ 03 และตำแหน่งปลายที่ 2 เมื่อเทียบกับรูปที่ 2.20 จะอยู่ในช่องที่ 33

เวกเตอร์ทิศทาง เป็นการตรวจสอบว่าตัวอักษรแต่ละตัวมีทิศทางของเส้นโครงร่างเป็นลักษณะอย่างไร โดยเริ่มวิเคราะห์จากจุดเชื่อมรอยต่อกับหัวของตัวอักษร เวกเตอร์ทิศทางที่ได้สามารถนำมาเทียบกับแบบเปรียบเส้นตรงซึ่งมีอยู่ 8 ทิศทางได้ ดังรูปที่ 2.15 โดยเวกเตอร์ทิศทางแต่ละเส้นจะครอบคลุมพื้นที่ 45 องศา

ตัวอย่างของลักษณะเด่นที่วิเคราะห์ได้จากตัวอักษรแต่ละตัวดูได้ในภาคผนวก จ

2.4.2 การแปลงแบบเปรียบพื้นฐาน (primitive) ให้เป็นเลขฐานสอง

เนื่องจากข้อมูลของภาพตัวอักษรที่ถูกแทนด้วยแบบเปรียบพื้นฐาน (primitive) ไม่สามารถป้อนได้ในระดับข้อมูลเข้า (Input layer) ของโครงข่ายประสาทเทียมได้โดยตรง ดังนั้น จะต้องแปลงแบบเปรียบเหล่านี้ให้เป็นเลขฐานสอง (Binary) ดังตารางที่ 2.4 และตารางที่ 2.5

ตารางที่ 2.4 แสดงการแปลงแบบเปรียบเส้นตรงให้เป็นเลขฐานสอง

แบบเปรียบเส้นตรง	เลขฐานสอง
L0	0000
L1	0001
L2	0010
L3	0011
L4	0100
L5	0101
L6	0110
L7	0111

ตารางที่ 2.5 แสดงการแปลงแบบเปรียบวงกลมเป็นเลขฐานสอง

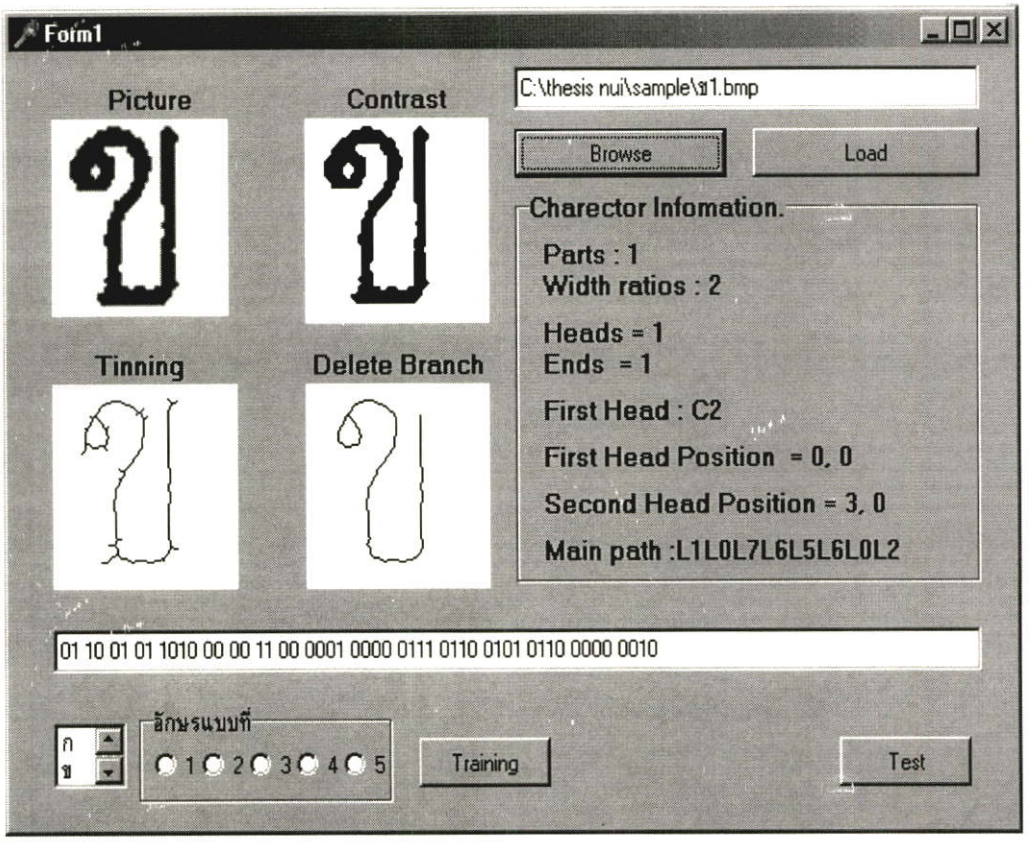
แบบเปรียบวงกลม	เลขฐานสอง
C0	1000
C1	1001
C2	1010
C3	1011
C4	1100

ตารางที่ 2.6 แสดงการแปลงลักษณะเด่นของตัวอักษรเป็นรหัสเลขฐานสอง

จำนวนส่วนของตัวอักษร Part (2 บิต)	0	00
	1	01
	2	10
	3	11
อัตรากว้าง Width ratio (2 บิต)	0	00
	1	01
	2	10
	3	11
จำนวนหัวของตัวอักษร Head (2 บิต)	0	00
	1	01
	2	10
	3	11
จำนวนปลายของตัวอักษร Ends (2 บิต)	0	00
	1	01
	2	10
	3	11
ลักษณะของหัวแรก First Head (4 บิต)	แบบเปรียบวงกลม 5 รูป แบบที่แปลงเป็นเลขฐานสองดังตารางที่ 2.5	
ตำแหน่งหัวหรือปลายแรก First Head Position (4 บิต)	ดังตัวอย่างรูปที่ 2.19	
ตำแหน่งหัวหรือปลายที่ 2 Second Head Position (4 บิต)	ดังตัวอย่างรูปที่ 2.19	
เวกเตอร์ทิศทาง Main path (32 บิต)	แบบเปรียบเส้นตรง 8 รูป แบบที่แปลงเป็นเลขฐานสองดังตารางที่ 2.4	

ตัวอย่างการแปลงข้อมูลภาพตัวอักษรเป็นรหัสเลขฐานสอง

เมื่อรับข้อมูลภาพเข้ามาแล้วก็นำมาเข้าสู่กระบวนการเตรียมข้อมูล คือ การกำจัดสัญญาณรบกวน การปรับความเข้มของภาพ การสำรวจพื้นที่ของภาพ การทำให้บาง และการลบกิ่งขลิบปลาย เมื่อได้ภาพข้อมูลตัวอักษรที่ทำให้บางแล้วนำมาหาลักษณะเด่นเพื่อใช้เป็นตัวแทนของตัวอักษร จากนั้นทำการแปลงลักษณะเด่นที่ได้ให้เป็นรหัสเลขฐานสอง ดังรูปที่ 2.21

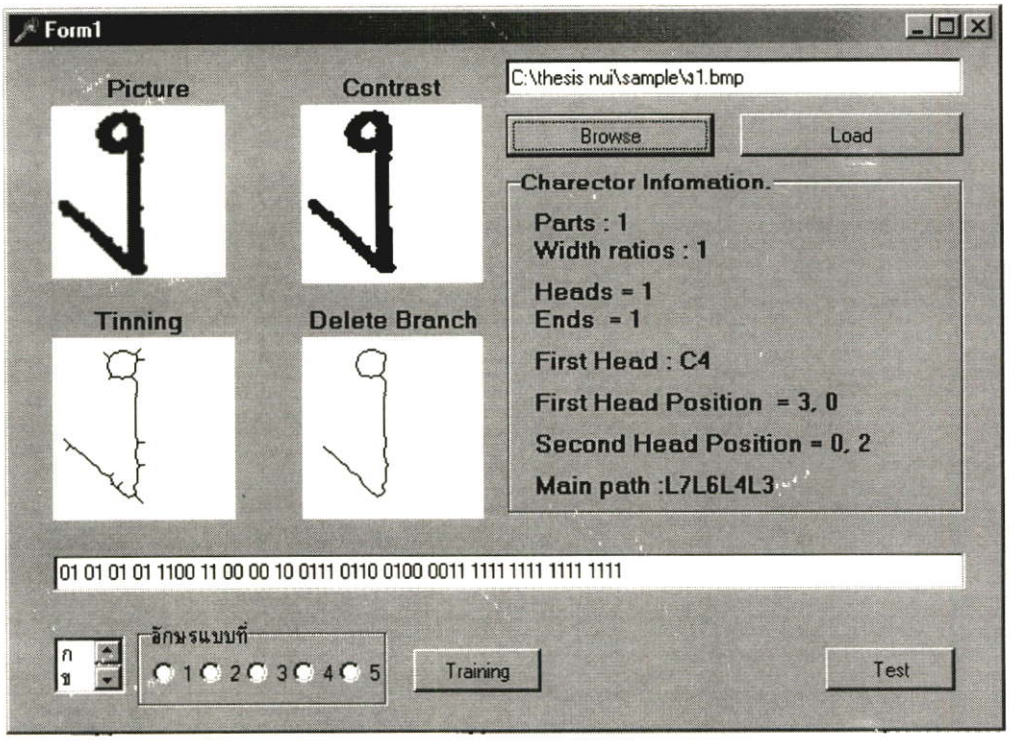


รูปที่ 2.21 แสดงหน้าต่างขั้นตอนการแปลงข้อมูลภาพตัวอักษรให้เป็นเลขฐานสอง (Binary) ของอักษร ข

จากรูปที่ 2.21 เมื่อทำการหาลักษณะเด่นของตัวอักษรและทำการแปลงเป็นรหัสเลขฐานสองแล้ว จะได้รหัสเลขฐานสองที่เป็นตัวแทนของตัวอักษรตัวนั้นๆ เพื่อเป็นข้อมูลอินพุทในการฝึกสอนและทดสอบการรู้จำต่อไป รายละเอียดการแปลงลักษณะเด่นเป็นรหัสเลขฐานสองของรูปที่ 2.21 แสดงไว้ในตารางที่ 2.7

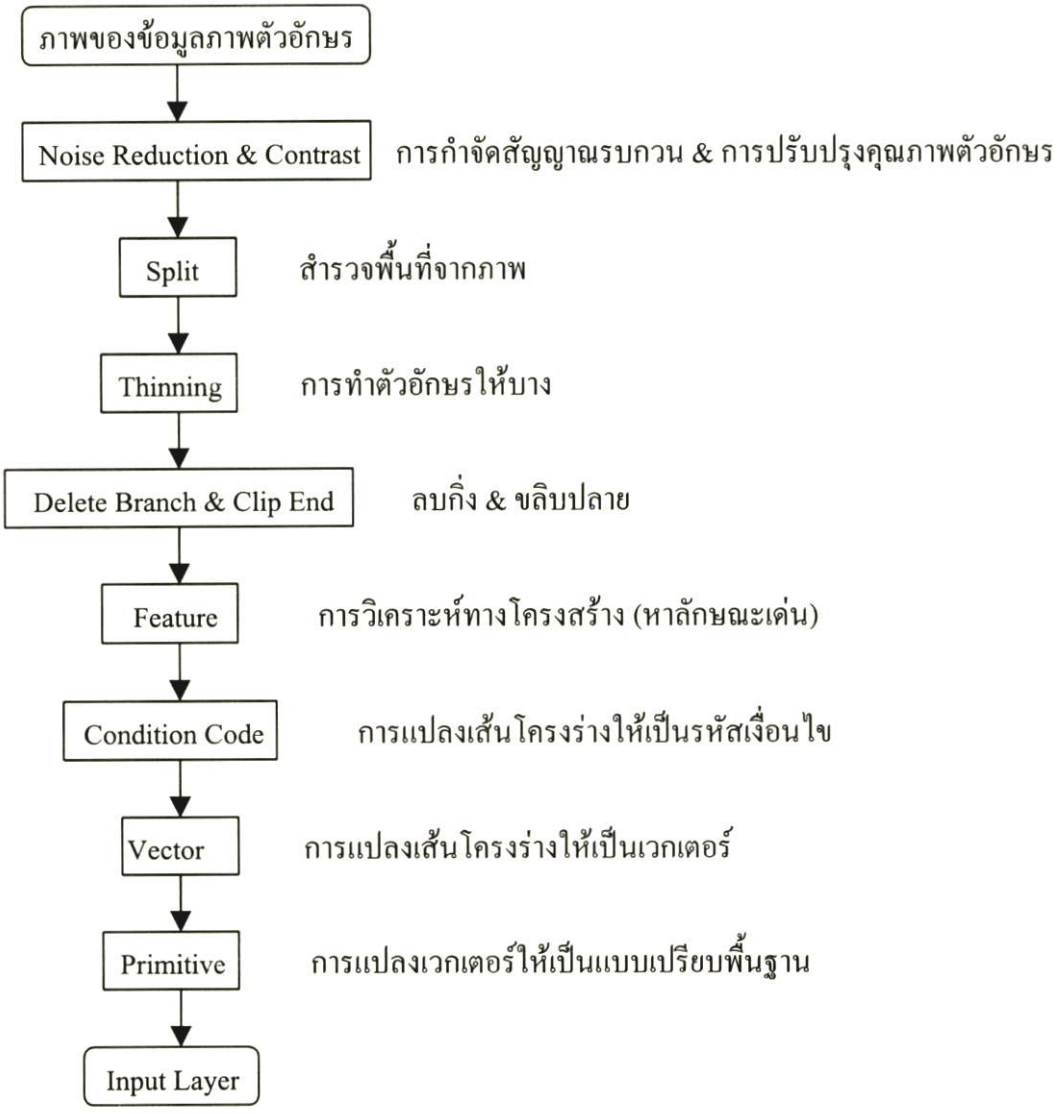
ตารางที่ 2.7 แสดงตัวอย่างการแปลงข้อมูลภาพตัวอักษรให้เป็นเลขฐานสองของตัวอักษร ข

ลักษณะเด่นที่วิเคราะห์	ลักษณะเด่นที่วิเคราะห์ได้	รหัสเลขฐานสอง
Part	1 ส่วน	01
Width ratio	2	10
Heads	1 หัว	01
Ends	1 ปลาย	01
First.H	C2	1010
First.H.P.	0 , 0	00 , 00
Second. H.P.	3 , 0	11 , 00
Main path	L1 , L0 , L7 , L6 , L5 , L6 , L0 , L2	0001 , 0000 , 0111 , 0110 , 0101 , 0110 , 0000 , 0010



รูปที่ 2.22 แสดงหน้าต่างขั้นตอนการแปลง ข้อมูลภาพตัวอักษรให้เป็นเลขฐานสองของอักษร ง

จากรูปที่ 2.22 จะเห็นว่า ส่วนที่ต่อจาก main path ตามด้วย 1111 , 1111 , 1111 , 1111 ในส่วนนี้จะเรียกว่า Dummy เป็นเลขฐานสองที่นำมาแทนจำนวนบิตที่ไม่ได้ใช้ในตัวอักษร ง นี้ ตัวอย่างการแปลงข้อมูลภาพของตัวอักษรต้นแบบเป็นรหัสเลขฐานสองแสดงไว้ในภาคผนวก จ



รูปที่ 2.23 แสดงแผนผังลักษณะการทำงานของการแปลงตัวอักษรลายมือเขียนภาษาไทยเป็นรหัสเลขฐานสอง

2.5 โครงข่ายประสาทเทียม

2.5.1 ประวัติความเป็นมา

ความเจริญของวิชาประสาทวิทยาได้ยอมรับงานวิจัยในการสร้างแบบจำลองทางคณิตศาสตร์ของเซลล์ประสาทซึ่งเลียนแบบพฤติกรรมของระบบประสาท โครงข่ายประสาทเทียมได้เริ่มมีการคิดค้นกันมาตั้งแต่ปี ค.ศ. 1940 เช่นในปี 1943 วอร์แรน แมคคูลล๊อค (Warrant McCulloch) และวอลเตอร์ พิทท์ (Walter Pitts) ได้ออกแบบโครงข่ายประสาทเทียมที่ถือเป็นต้นกำเนิดและพื้นฐานของโครงข่ายประสาทเทียมในปัจจุบัน โครงข่ายของแมคคูลล๊อคและพิทท์ประกอบด้วย ชั้นของโครงข่ายจำนวน 2 ชั้น คือ ชั้นอินพุทและชั้นเอาท์พุท โครงข่ายจะมีการเชื่อมโยงจากชั้นอินพุทไปยังชั้นเอาท์พุทโดยผ่านน้ำหนัก (weight) ซึ่งค่าน้ำหนักจะถูกกำหนดค่าไว้ตายตัว (fixed weight) การทำงานเบื้องต้นของโครงข่าย คือ นำไปใช้จำลองฟังก์ชันทางลอจิกพื้นฐาน คือ ฟังก์ชัน AND , ฟังก์ชัน OR , ฟังก์ชัน NOT และฟังก์ชัน XOR โดยนำเสนอการกำหนดค่าน้ำหนักเพื่อให้โครงข่ายทำงานได้กับการจำลองตามฟังก์ชันดังกล่าว

ในปี ค.ศ. 1949 โดแนลด์ เฮบบ์ (Donald Hebb) ได้ออกแบบกฎการเรียนรู้สำหรับโครงข่ายประสาทเทียมเป็นคนแรกกฎการเรียนรู้นี้ ก็คือ ขั้นตอนของกระบวนการปรับแต่งค่าน้ำหนักเพื่อทำให้เกิดค่าเอาท์พุท

ในปี ค.ศ. 1957 แฟรงค์ โรเซนบลัทท์ (Frank Rosenblatt) ได้นำเสนอรูปแบบของโครงข่ายประสาทเทียมที่สำคัญรูปแบบหนึ่งเรียกว่า “Perceptron” ประกอบด้วย ชั้นอินพุทต่อผ่านน้ำหนักไปชั้นเอาท์พุทและใช้กฎการเรียนรู้ของ “Perceptron” ในการปรับค่าน้ำหนักที่มีประสิทธิภาพดีกว่ากฎการเรียนรู้ของแฮบบ์

ในปี ค.ศ. 1960 เบอร์นาร์ด วิดโรว์ (Bernard Widrow) และ มาเชียน ฮอฟฟ์ (Marcian Hoff) ได้พัฒนากฎการเรียนรู้ที่ใกล้เคียงกับกฎการเรียนรู้ของ “Perceptron” ซึ่งเรียกว่า “ADALINE” (Adaptive Linear Neuron) และระบบโครงสร้างของ ADALINE ที่เรียกว่า “MADALINE” (Multiple ADALINE) และได้นำไปพัฒนากฎการเรียนรู้แบบจัดตัวเองที่รู้จักกันดีในนามของ Least Mean Square(LMS) หรือกฎการเรียนรู้วิดโรว์ – ฮอฟฟ์ (Widrow – Hoff)

ในปี ค.ศ. 1972 ทูโว โคโฮเนน (Teuvo Kohonen) ได้พัฒนาการเรียนรู้แบบจัดตัวเอง (self – organized) โครงข่ายโคโฮเนนนี้ประกอบด้วยจำนวนชั้น 2 ชั้น คือ ชั้นอินพุทและชั้นเอาท์พุท การเชื่อมโยงที่ชั้นอินพุทไปยังชั้นเอาท์พุทนั้นจะต่อผ่านน้ำหนัก ข้อแตกต่างจากโครงข่ายอื่นๆ คือ กระบวนการเรียนรู้นี้สามารถจัดกลุ่มข้อมูลอินพุทได้ด้วยตัวของมันเอง และเรียกการเรียนรู้นี้ว่าการเรียนรู้แบบแข่งขัน (competitive learning)

หลังจากปี ค.ศ. 1980 เป็นต้นมา ได้มีการพัฒนารูปแบบของโครงข่ายประสาทเทียมชนิดต่างๆ เช่น รูปแบบจัดตัวเองที่เรียกว่า “ART” คิดค้นโดย สตีเฟน กรอสเบิร์ก (Stephen Grossberg) และ เกียล คาร์เพนเตอร์ (Gail Carpenter) รูปแบบโครงข่ายฮอปฟิลด์ คิดค้นโดย จอร์น ฮอปฟิลด์

(John Hopfield) รูปแบบโครงข่าย Neocognition คิดค้นโดย คูนิฮิโกะ ฟูกูชิมา (Kunihiko Fukushima) รูปแบบโบลสแมนน์แมชชีน (Boltzmann Machine) คิดค้นโดย เดวิด ปาร์คเคอร์ (David Parker) และเดวิด รูเมลฮาร์ท (David Rumelhard) ซึ่งรูปแบบแพร่ย้อนกลับนี้สามารถให้ค่าน้ำหนักที่เหมาะสมเพื่อลดค่าผิดพลาดของเอาท์พุท

2.5.2 ลักษณะโครงข่ายประสาททางชีววิทยา (Biological Neural Network)

โครงข่ายประสาท หมายถึง โครงข่ายใยประสาทที่เชื่อมต่อกันระหว่างเซลล์ประสาทจำนวนมากมีความสามารถในการประมวลผลสูง ทำหน้าที่เป็นจุดศูนย์กลางการควบคุมกิจกรรมต่างๆ การวิจัยการสร้างโครงข่ายประสาทเทียม (Artificial Neural Networks) มีแนวความคิดเลียนแบบการทำงานของสมองชีวภาพโดยเรียนรู้และศึกษาการทำงานของสมองชีวภาพ เพื่อกำหนดแนวทางสำหรับการสร้างแบบจำลองขึ้นมาแล้วพยายามสร้างสมมติฐานลักษณะการทำงาน โดยจำลองเป็นโมเดลคณิตศาสตร์ที่มีลักษณะเดียวกันแล้วทำการคำนวณด้วยคอมพิวเตอร์

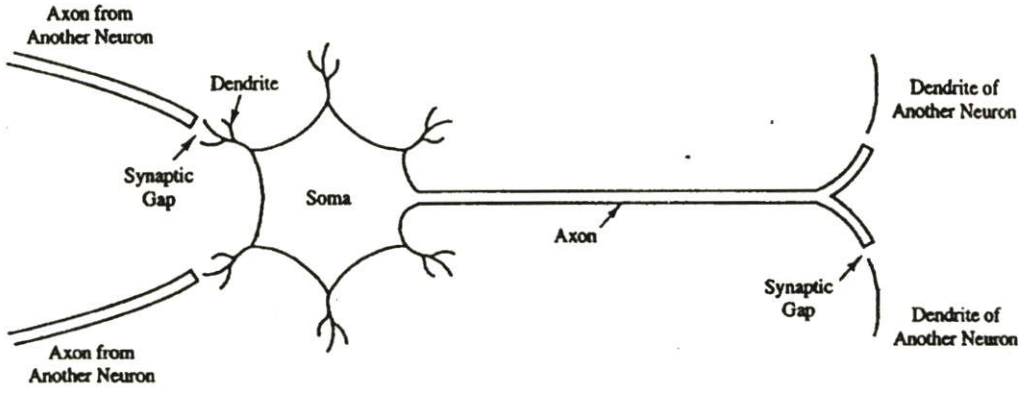
เซลล์ประสาทของสิ่งมีชีวิตมีหลายประเภทแล้วแต่หน้าที่ของมัน เซลล์ประสาทในตัวของคนเราก็เช่นกันมีอยู่หลายประเภทตามตำแหน่งและหน้าที่ เช่น เซลล์ประสาทของกล้ามเนื้อ เซลล์ประสาทในสมอง เซลล์ประสาทที่ลิ้นและจมูก เป็นต้น

ระบบเส้นประสาทของมนุษย์ประกอบด้วยเซลล์เล็กๆมากมายที่เรียกว่า นิวรอลหรือเซลล์ประสาท ซึ่งรวมตัวกันอยู่ในรูปแบบโครงสร้างที่ซับซ้อนมีอยู่ประมาณ 10^{11} นิวรอล โดยนิวรอลแต่ละตัวประกอบด้วยส่วนสำคัญ 4 ส่วน ดังแสดงในรูปที่ 2.24

1. เดนไดร์ท (Dendrites)	ตัวรับสัญญาณอินพุท (Input)
2. แอกซอน (Axon)	ตัวส่งสัญญาณเอาท์พุท (Output)
3. ซินแนปส์ (Synapses)	จุดเชื่อมต่อ (Connection point)
4. ตัวเซลล์ (Soma)	จุดศูนย์รวมสัญญาณ (Central neuron system)

แต่ละนิวรอลจะเชื่อมต่อกับนิวรอลอื่นอีกหลายนิวรอล โดยเดนไดร์ทจะทำหน้าที่รับสัญญาณที่จุดเชื่อมต่อที่เรียกว่า ซินแนปส์ และถูกนำไปเก็บสะสมในตัวเซลล์ในช่วงเวลาสั้นๆ มีผลทำให้เกิดการลดความเข้มของสัญญาณหรือเพิ่มความเข้มของสัญญาณ ซึ่งบางสัญญาณมีผลกระตุ้นตัวเซลล์ (Excitatory) บางสัญญาณมีผลยับยั้งตัวเซลล์ (Inhibiyory) เมื่อผลรวมของสัญญาณทั้งหมดเกินกว่าค่าเทรดโฮรล (Threshold) นิวรอลจะส่งสัญญาณผ่านแอกซอนไปให้นิวรอลอื่นในทันที

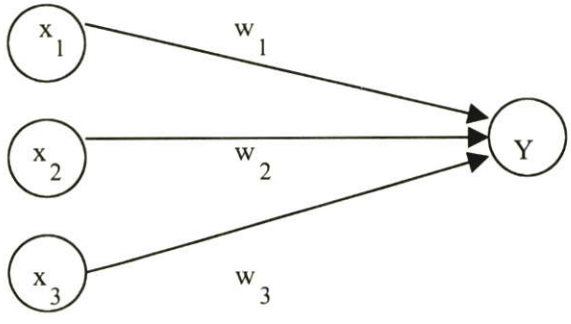
กล่าวโดยสรุปว่า นิวรอลหรือเซลล์ประสาทจะทำหน้าที่ 3 ประการ คือ รับสัญญาณที่จุดเชื่อมต่อ (Receptors) และทำการประมวลผลภายในโครงสร้างของนิวรอล (Neuron system) และส่งผลการประมวลผลผ่านปลายหนึ่งของเซลล์ไปให้เซลล์อื่นๆ (Effectors)



รูปที่ 2.24 แสดงโครงข่ายประสาททางชีววิทยา

2.5.3 โครงข่ายประสาทเทียม (Artificial Neural Networks)

การออกแบบโครงข่ายประสาทเทียมนั้นมีสมมติฐานขั้นแรกจากคุณสมบัติของระบบประสาททางชีวภาพดังที่กล่าวมา กล่าวคือ ชุดรับสัญญาณข้อมูลอินพุทของเซลล์ประสาทหนึ่งได้จากสัญญาณเอาท์พุทของเซลล์ประสาทอื่นๆผ่านทางซินแนปส์และเดนไดร์ท ข้อมูลแต่ละค่าที่รับมาจะถูกลดหรือเพิ่มขนาดด้วยซินแนปติกส์ซึ่งภายในประกอบด้วยสารเคมีประเภท K^+, Ca^{++}, Na^+, Cl^- ซึ่งจะมีลักษณะทางความนำพัลล์ (Pulse) สัญญาณไฟฟ้าเคมีที่แตกต่างกันด้วยเหตุนี้ โมดุลประสาทเทียมที่สร้างขึ้นจะต้องมีการถ่วงน้ำหนักให้กับโมดุลก่อนที่จะนำเข้าสู่โมดุลประสาทเทียม จุดนี้เรียกว่า ซินแนปติกส์เวจท์ ปริมาณของข้อมูลที่เข้าสู่นิวรอลจะถูกนำมารวมกันและตัดสินใจด้วยระดับความสนใจ (Activation level) ของนิวรอลนั้นๆแล้วส่งเป็นเอาท์พุทออกที่แอกซอนไปยังนิวรอลอื่นๆ ดังรูปที่ 2.25 และ 2.26



รูปที่ 2.25 แสดงแบบจำลองของเซลล์ประสาทเทียมอย่างง่าย

จากรูปที่ 2.25 แสดงถึงโมดุลที่สร้างขึ้นโดยแนวความคิดจากเซลล์สมองชีวภาพ สัญญาณอินพุท คือ x_1, x_2, x_3 จะถูกป้อนเข้าไปยังนิวรอลที่สร้างขึ้น Y ซึ่งเปรียบได้กับสัญญาณที่ป้อนเข้ามายังซินแนปส์ของนิวรอลชีวภาพ สัญญาณอินพุทนี้จะนำไปคูณกับค่าซินแนปติกส์เวจท์ที่มีค่าตั้งแต่ 0.00-1 คือ w_1, w_2, w_3 ผลรวมทางคณิตศาสตร์ของอินพุทและน้ำหนักจะได้เป็นเอาท์พุท ซึ่งเราจะเขียนในรูปของเวกเตอร์ได้ดังนี้

$$y_in = x_1 w_1 + x_2 w_2 + x_3 w_3 \tag{2.3}$$

หรือ

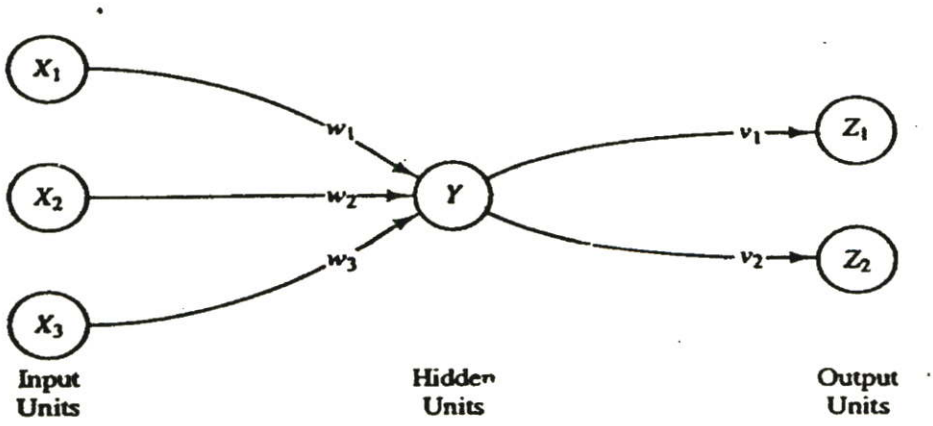
$$y_in = \sum_{i=1}^{3} x_i w_i \tag{2.4}$$

ฟังก์ชันกระตุ้นการตัดสินใจ (Activation Function)

เมื่อได้รับสัญญาณ y_in แล้วกระบวนการต่อมาที่นิวรอลต้องทำคือ ตัดสินใจ เราจึงต้องกำหนดฟังก์ชันการตัดสินใจเพื่อใช้เป็นเครื่องมือในการตัดสินใจของนิวรอล เพื่อทำให้ได้สัญญาณเอาท์พุทของนิวรอลออกมาซึ่งเชื่อมต่อไปยังนิวรอลตัวอื่นๆเป็นโครงข่าย ซึ่งเปรียบเสมือนเซลล์ประสาทจะส่งกระแสประสาทออกไปทางแอกซอนได้ก็ต่อเมื่อศักดาไฟฟ้าที่มากระตุ้นมีค่าสูงถึงระดับหนึ่ง (Threshold) จะกระตุ้นการส่งสัญญาณออกไปทางแอกซอนได้ ซึ่งเอาท์พุทที่ได้อาจเขียนเป็นสมการเชิงเส้นอย่างง่ายได้ดังนี้

$$y = f(y_in) \tag{2.5}$$

โดย f เป็นฟังก์ชันกระตุ้นการตัดสินใจ (Activation Function)



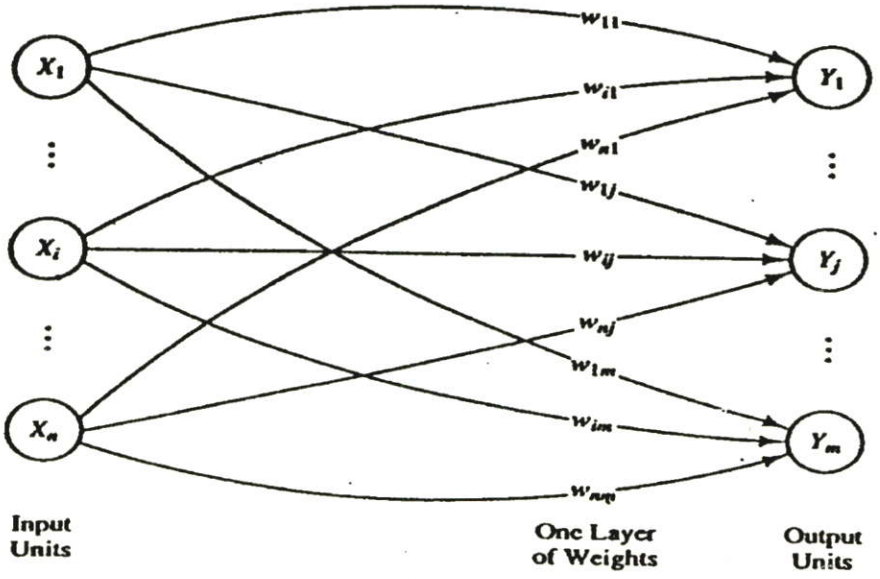
รูปที่ 2.26 แสดงแบบจำลองของโครงข่ายประสาทเทียมอย่างง่าย

จากรูปที่ 2.26 เป็นการแสดงโครงข่ายประสาทเทียมอย่างง่ายๆ คือ พิจารณานิวรอล Y ซึ่งเชื่อมต่อกับนิวรอล Z_1 และ Z_2 ด้วยน้ำหนัก v_1 และ v_2 ตามลำดับ นิวรอล Y เมื่อได้รับสัญญาณจากนิวรอล X_1, X_2 และ X_3 แล้วจะทำการส่งสัญญาณ y ไปสู่นิวรอลแต่ละอันในชั้นถัดไป โดยทั่วๆไปแล้ว ค่าที่ถูกส่งไปสู่นิวรอล Z_1 และ Z_2 จะแตกต่างกัน เพราะแต่ละสัญญาณจะถูกคูณกับน้ำหนัก v_1 หรือ v_2 ในระบบจะเห็นว่า สัญญาณกระตุ้น z_1และ z_2 ของนิวรอล Z_1 และ Z_2 จะขึ้นอยู่กับอินพุทจากนิวรอลต่างๆหลายนิวรอลไม่ใช่เพียงนิวรอลเดียว

2.6 โครงข่ายประสาทเทียมแบบต่างๆ (Typical Architectures)

2.6.1 โครงข่ายประสาทเทียมแบบชั้นเดียว (Single – Layer Neural Network Models)

โครงข่ายประสาทเทียมแบบชั้นเดียว มีค่าถ่วงน้ำหนักเชื่อมต่อเพียงชั้นเดียว หน่วยอินพุทจะแยกออกจากกันและรับสัญญาณมาจากภายนอก ส่วนหน่วยเอาท์พุทจะเป็นผลตอบสนองจากโครงข่ายประสาทเทียมในตัวอย่าง โครงข่ายประสาทเทียมแบบชั้นเดียวดังรูปที่ 2.27 หน่วยของอินพุทจะทำการเชื่อมต่อกับหน่วยของเอาท์พุททั้งหมดแต่ไม่เชื่อมต่อกับหน่วยอื่นๆของอินพุทในชั้นเดียวกันและหน่วยของเอาท์พุทจะไม่เชื่อมต่อกับหน่วยอื่นๆของเอาท์พุทในชั้นเดียวกัน สำหรับแต่ละหน่วยของเอาท์พุทจะมีลักษณะเช่นเดียวกันแต่แยกออกจากกันอาจขึ้นหรือไม่ขึ้นกับอินพุทเวกเตอร์ สังเกตโครงข่ายประสาทเทียมแบบชั้นเดียวค่าถ่วงน้ำหนักสำหรับหนึ่งเอาท์พุทไม่ได้มีอิทธิพลกับเอาท์พุทหน่วยอื่นๆหรือรับอิทธิพลจากเอาท์พุทหน่วยอื่นๆ

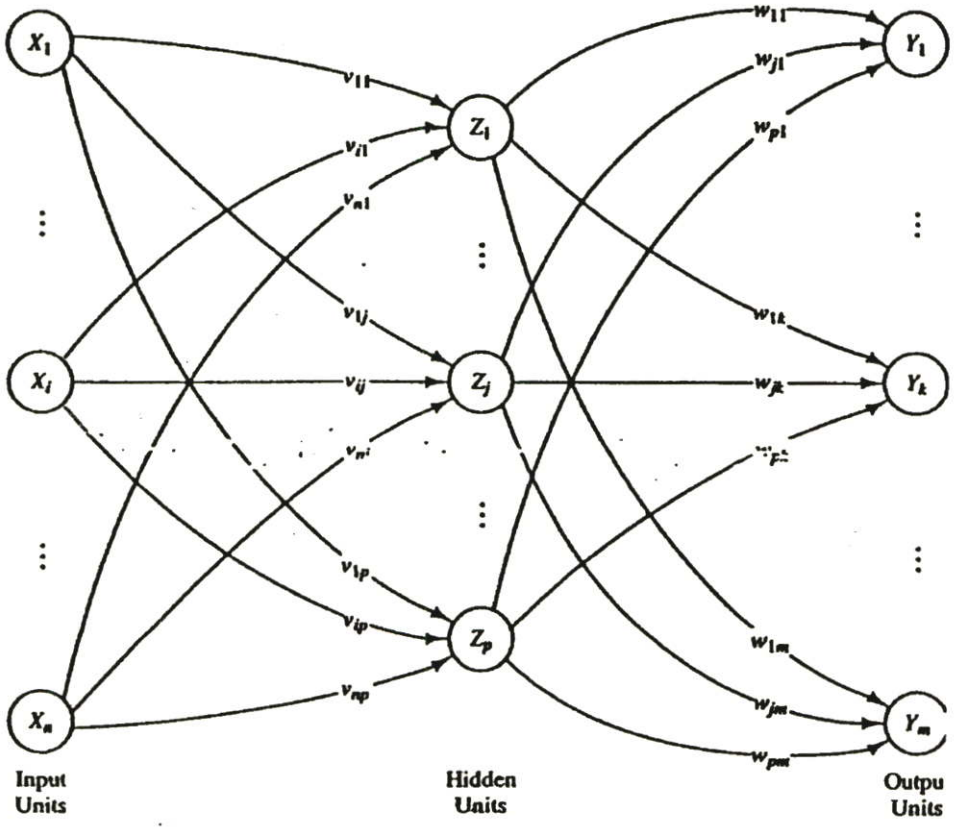


รูปที่ 2.27 แสดงโครงข่ายประสาทเทียมแบบชั้นเดียว

สำหรับค่าน้ำหนัก (weight) ในรูปที่ 2.27 มีวิธีการพิจารณาในรูปของเมตริกซ์ (weight matrix) ซึ่งหากโครงข่ายมีหลายชั้น จะช่วยให้ระบุค่าน้ำหนักได้ง่ายขึ้น และเพื่อหลีกเลี่ยงความสับสนจะกำหนดเป็นมิติ (Dimension) ของเมตริกซ์ โดยให้ n แทน จำนวนแถวหรือจำนวนของอินพุท และ m แทน จำนวนของนิวรอลที่สร้างขึ้นเขียนแทนด้วย w_{nm} ตัวอย่างเช่น น้ำหนักเชื่อมระหว่างอินพุทตัวที่ 2 กับนิวรอลตัวที่ 4 คือ w_{24}

2.6.2 โครงข่ายประสาทเทียมแบบหลายชั้น (Multi – Layer Neural Network Models)

โครงข่ายประสาทเทียมแบบหลายชั้น สามารถแก้ปัญหาที่ซับซ้อนมากๆได้ดีกว่าโครงข่ายประสาทเทียมแบบชั้นเดียวแต่การสอนให้เรียนรู้อาจยากกว่า โครงข่ายที่ซับซ้อนมากขึ้นจะมีความสามารถในการคำนวณดีขึ้น โดยโครงข่ายจะเชื่อมโยงนิวรอลที่มีโครงสร้างเป็นชั้นๆคล้ายส่วนหนึ่งของสมอง และมีการพัฒนาอัลกอริทึมเกี่ยวกับการฝึกสอนให้โครงข่ายแบบหลายชั้นทำงานได้ตามความต้องการ โครงข่ายแบบหลายชั้นอาจจะสร้างจากกลุ่มของโครงข่ายแบบชั้นเดียว เอาท์พุทของชั้นหนึ่งจะใช้เป็นอินพุทของชั้นถัดไป ชั้นที่อยู่ระหว่างชั้นอินพุทและชั้นเอาท์พุท เรียกว่า ชั้นซ่อน (hidden layer) ซึ่งชั้นซ่อนนั้นจะมีหนึ่งชั้นหรือมากกว่าก็ได้ หากชั้นซ่อนมีหลายๆชั้น ก็จะมีการตั้งชื่อเฉพาะลงไปให้กับแต่ละชั้นซ่อนนั้น ดังรูปที่ 2.28



รูปที่ 2.28 แสดงโครงข่ายประสาทเทียมแบบหลายชั้น

จากรูปที่ 2.28 เป็นโครงข่ายประสาทเทียมแบบสองชั้น ซึ่งมีการส่งข้อมูลไปทิศทางเดียว ในชั้นเดียวกันจะไม่มีการส่งข้อมูลถึงกันแต่จะมีการส่งระหว่างชั้นเชื่อมถึงกันหมด โดยในแต่ละเส้นทางการเชื่อมต่อจะมีค่าถ่วงน้ำหนักประจำอยู่

ฟังก์ชันกระตุ้นการตัดสินใจแบบไม่เป็นเชิงเส้น (The Nonlinear Activation Function)

การนำเอาท์พุทของชั้นหนึ่งมาเชื่อมกับอินพุทของชั้นถัดไปโดยผ่านฟังก์ชันกระตุ้นการตัดสินใจแบบไม่เป็นเชิงเส้น จะทำให้โครงข่ายมีความสามารถในการเรียนรู้เพิ่มมากขึ้น และสามารถกำหนดขอบเขตของเอาท์พุทให้อยู่ในช่วงที่ต้องการได้ หากไม่ผ่านฟังก์ชันดังกล่าวความสามารถของโครงข่ายจะไม่แตกต่างไปจากโครงข่ายประสาทเทียมแบบชั้นเดียว

2.7 การฝึกสอนให้กับโครงข่ายประสาทเทียม

โครงข่ายประสาทเทียมที่สร้างขึ้น เมื่อสร้างเสร็จแล้วแต่ละเซลล์ประสาทที่สร้างขึ้นนั้นจะยังไม่มีคุณลักษณะใดเลย เนื่องจากยังไม่มีการกำหนดค่าน้ำหนักให้กับซินแนปติกส์ (synaptic weight) ที่เหมาะสมกับงานที่ต้องการให้กับมัน จึงต้องมีการฝึกสอนเพื่อให้โครงข่ายประสาทเทียมที่สร้างขึ้นมีคุณลักษณะตามที่ต้องการ การฝึกสอนของโครงข่ายประสาทเทียมจะกระทำโดยการปรับเปลี่ยนค่าน้ำหนักเพื่อให้โครงข่ายจดจำรูปแบบความสัมพันธ์ระหว่างอินพุทกับเอาท์พุทได้ โดยในขั้นแรกอาจกำหนดน้ำหนักเป็นค่าสุ่มใดๆ (random weight) ก่อน แล้วจึงปรับเปลี่ยนน้ำหนักไปตามอัลกอริทึมของสมมติฐานที่กำหนดหลายๆรอบจนกว่าจะได้เอาท์พุทของโครงข่ายเหมือนกับเอาท์พุทที่ต้องการในเงื่อนไขความผิดพลาดที่ยอมรับได้

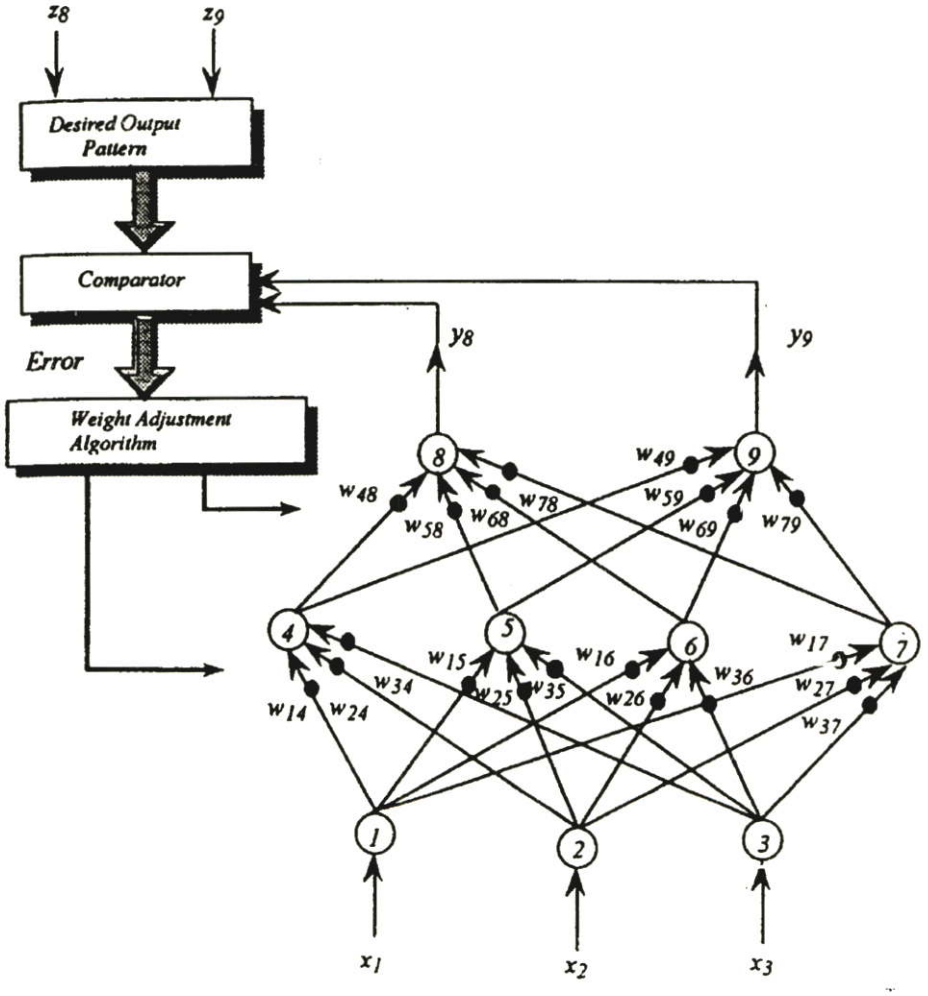
เนื่องจากค่าน้ำหนักที่ให้เป็นค่าสุ่มใดๆ โครงข่ายจึงยังไม่แสดงคุณลักษณะใดออกมา การฝึกสอนให้โครงข่ายก็คือ การปรับค่าน้ำหนักทุกๆจุดให้สอดคล้องกับอินพุทหลายๆแบบเพื่อให้ได้เอาท์พุทตามต้องการนั่นเอง หลังจากปรับน้ำหนักจนได้ค่าผิดพลาดที่เอาท์พุทเทียบกับเป้าหมายที่กำหนดน้อยลงเป็นที่พอใจแล้ว โครงข่ายประสาทเทียมนั้นก็พร้อมที่จะวิเคราะห์อินพุทและให้เอาท์พุทตามลักษณะตัวอย่างที่เคยเรียนรู้มา การเรียนรู้จะมีการปรับน้ำหนักหลายๆรอบ จนค่าน้ำหนักสอดคล้องกับตัวอย่างหลายๆตัวอย่างที่ครอบคลุมมากเพียงพอ พบว่า โครงข่ายก็จะมีความแม่นยำสูงขึ้นแต่ก็จะใช้เวลาในการฝึกสอนนานขึ้นเช่นกัน

2.7.1 วิธีฝึกสอนโครงข่ายประสาทเทียม

2.7.1.1 การฝึกสอนแบบชี้นำ (Supervised Training) เป็นการฝึกสอนแบบจับคู่กันระหว่างอินพุทและเอาท์พุทเป้าหมาย (target output) ที่ต้องการ เราเรียกว่า "Input – Output pair" ถ้าให้ $(x_1,t_1),(x_2,t_2),...,(x_k,t_k)$ เป็นจำนวนคู่ที่ทำการฝึกสอน ในเมื่อแต่ละอินพุท x_k ถูกส่งเข้ามาฝึกสอนในโครงข่ายประสาทเทียม เอาท์พุทจริง (actual output) y_k ที่คำนวณออกมาจะถูกนำไปเปรียบเทียบกับเอาท์พุทเป้าหมาย t_k เพื่อหาค่าผิดพลาดที่เกิดขึ้น จากนั้นจึงนำเอาค่าที่ผิดพลาดป้อน

กลับไปยังโครงข่ายเพื่อปรับค่าน้ำหนัก ทำให้แนวโน้มของค่าผิดพลาดที่เกิดขึ้นระหว่างเอาท์พุทจริง y_k กับเป้าหมาย t_k โดยเฉลี่ยมีค่าลดต่ำลง ตัวอย่างของการฝึกสอนแบบนี้ ได้แก่ โครงข่ายแบบแพร่ย้อนกลับ (Backpropagation)

2.7.1.2 การฝึกสอนแบบไม่มีการชี้นำ (Unsupervised Training) เป็นการฝึกสอนโดยไม่ต้องการเอาท์พุทเป้าหมายเข้ามาช่วยในการตัดสินใจ ชุดการฝึกสอนจะมีเพียงอินพุทเท่านั้น ระบบการฝึกสอนในโครงข่ายจะมีการคำนวณปรับเปลี่ยนค่าของน้ำหนักเพื่อให้เอาท์พุทที่มีค่าเสถียรภาพ การคำนวณนี้เรียกว่า "Self – Organizing" ยกตัวอย่างเช่น หากให้โครงข่ายรู้จำภาพหน้าคนๆหนึ่ง หากภาพหน้าคนๆนั้นเปลี่ยนแปลงไปเล็กน้อย ซึ่งอาจเกิดจากมีสัญญาณรบกวน โครงข่ายนั้นก็ยังบอกได้ว่าคนๆนั้นเป็นคนเดิม เป็นต้น การฝึกสอนจะไม่มีการตัดสินใจมาก่อน ไม่มีการกำหนดรูปแบบเอาท์พุทมาก่อน อาจกล่าวได้ว่า แบบเอาท์พุทจะถูกกำหนดโดยอินพุทเวกเตอร์นั่นเอง ได้แก่ โครงข่ายแบบโคโฮเนนหรือโครงข่ายการจัดตัวเองแบบ ART(Adaptive Resonance Theory)



รูปที่ 2.29 แสดงโครงข่ายประสาทเทียมแบบแพร่ย้อนกลับกับการสอนแบบชี้นำ

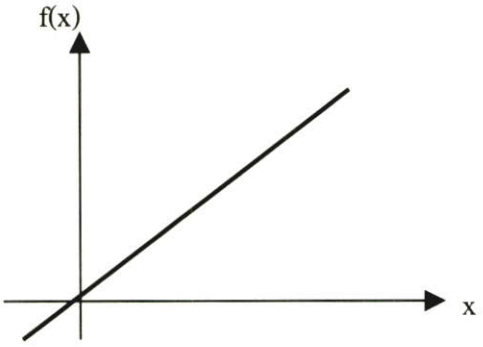
2.8 ฟังก์ชันกระตุ้นการตัดสินใจ (Activation Function)

ฟังก์ชันกระตุ้นการตัดสินใจทั่วๆไปมีหลายรูปแบบซึ่งอาจมีลักษณะเป็นสมการเชิงเส้นหรือไม่เป็นเชิงเส้นก็ได้ ขึ้นอยู่กับลักษณะโครงข่ายประสาทเทียมที่สร้างขึ้นและลักษณะของข้อมูลที่จะทำการวิเคราะห์หรือฝึกสอนให้กับโครงข่ายนั้นๆ

2.8.1 ชนิดของฟังก์ชันกระตุ้นการตัดสินใจ

1. Identity function

$$f(x) = x \quad , \forall x \tag{2.6}$$

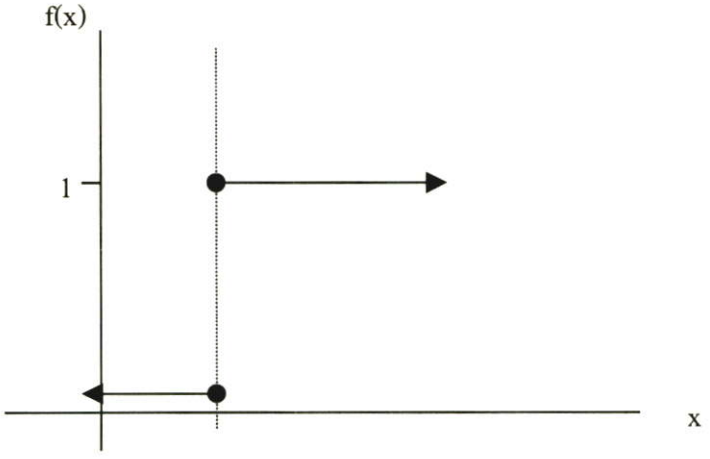


รูปที่ 2.30 แสดง Identity function

2. Binary step function

$$f(x) = \begin{cases} 1 & ; x \geq \theta \\ 0 & ; x < \theta \end{cases} \tag{2.7}$$

โดย θ เรียกว่า ค่า Threshold



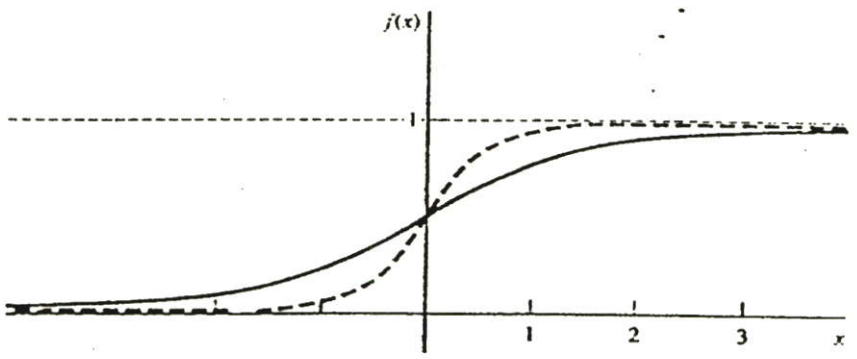
รูปที่ 2.31 แสดง Binary step function

3. Binary sigmoid

$$f(x) = \frac{1}{1+\exp(-\sigma x)} \tag{2.8}$$

$$f'(x) = \sigma f(x)[1 - f(x)] \tag{2.9}$$

โดย σ เรียกว่า Steepness parameter

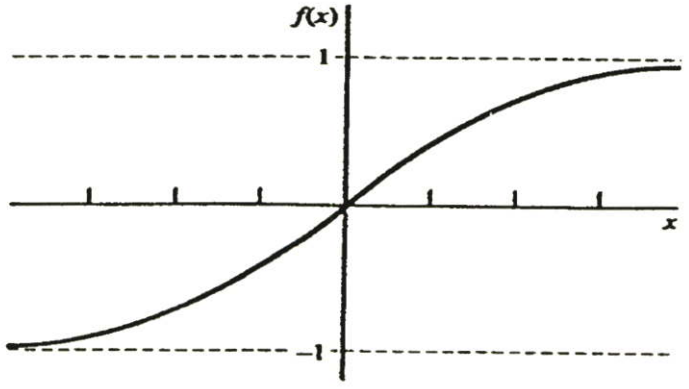


รูปที่ 2.32 แสดง Binary sigmoid

4. Bipolar sigmoid

$$\begin{aligned} g(x) &= 2f(x) - 1 \\ &= \frac{2}{1+\exp(-\sigma x)} - 1 \\ &= \frac{1-\exp(-\sigma x)}{1+\exp(-\sigma x)} \end{aligned} \tag{2.10}$$

$$g'(x) = \frac{\sigma}{2}[1 + g(x)][1 - g(x)] \tag{2.11}$$



รูปที่ 2.33 แสดง Bipolar sigmoid

2.8.2 การคูณเมตริกซ์สำหรับการคำนวณผลรวมของอินพุท

ถ้าน้ำหนักที่เชื่อมต่อของโครงข่ายประสาท คือ $W = (w_{ij})$ ผลรวมของอินพุทในชั้น Y_j (ไม่มี bias) หาได้จากผลคูณของเวกเตอร์แบบ dot product ของเวกเตอร์ $x = (x_1,...,x_i,...,x_n)$ และ $w_{\cdot j}$ (คอลัมท์ที่ j ของเมตริกซ์น้ำหนัก) โดย

$$\begin{aligned} y_in_j &= x \cdot w_{ij} \\ &= \sum_{i=1}^{n} x_i w_{ij} \end{aligned} \tag{2.12}$$

2.8.3 Bias

Bias เป็นตัวที่ใช้ในการเพิ่มเสถียรภาพให้กับระบบโดยสามารถนำเอามารวมไว้ในระบบได้ โดยการเติมส่วนประกอบ $x_0 = 1$ ลงไปในเวกเตอร์ x นั่นคือ $x = (1, x_1,...,x_i,...,x_n)$ และ bias จะถูกฝึกสอนเหมือนกับน้ำหนักตัวอื่นๆ ในที่นี้ให้ $w_{0j} = b_j$ ผลรวมอินพุทในชั้น Y_j หาได้จาก

$$\begin{aligned} y_in_j &= x \cdot w_{\cdot j} \\ &= \sum_{i=0}^{n} x_i w_{ij} \\ &= w_{0j} + \sum_{i=1}^{n} x_i w_{ij} \\ &= b_j + \sum_{i=1}^{n} x_i w_{ij} \end{aligned} \tag{2.13}$$

ถ้าในระบบมี bias รวมอยู่ด้วย ฟังก์ชันกระตุ้นการตัดสินใจมักจะเขียนได้ในรูป

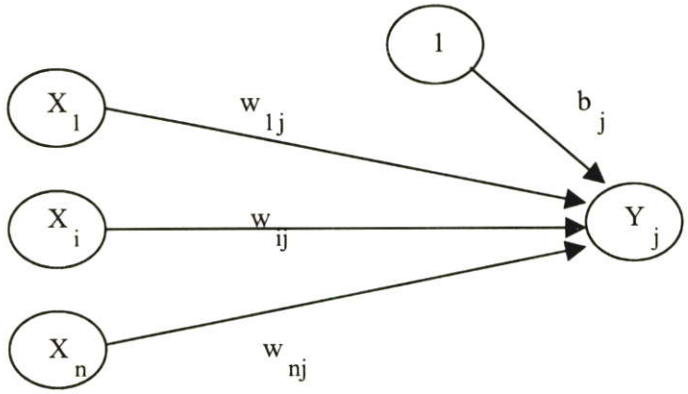
$$f(net) = \begin{cases} 1 & ;net \geq 0 \\ -1 & ;net < 0 \end{cases} \tag{2.14}$$

เมื่อ $net = b + \sum_i x_i w_i$

แต่ถ้าไม่มี bias อยู่ในระบบเราอาจแทนที่ค่าคงตัว θ (Threshold) ลงไปแทนได้ ฟังก์ชันกระตุ้นการตัดสินใจ จะเขียนได้ในรูป

$$f(net) = \begin{cases} 1 & ;net \geq \theta \\ -1 & ;net < \theta \end{cases} \tag{2.15}$$

เมื่อ $net = \sum_i x_i w_i$



รูปที่ 2.34 แสดงโครงข่ายประสาทเทียมที่มี bias

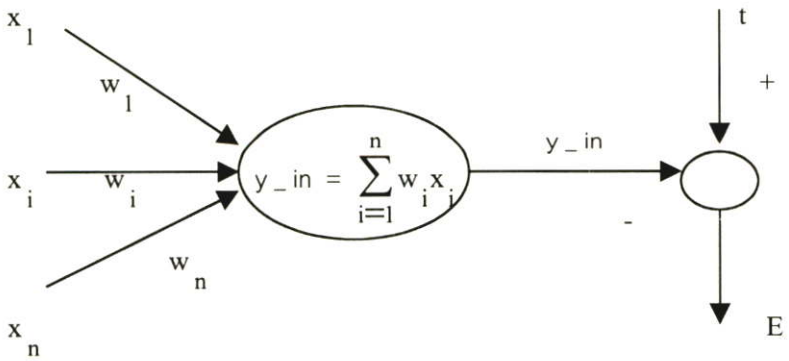
2.9 ทฤษฎีหลักการแพร่ย้อนกลับ

หลักการแพร่ย้อนกลับได้รับการพัฒนาโดยรูเมลฮาร์ทและถูกนำเสนอในช่วงปี ค.ศ. 1986 หลักการแพร่ย้อนกลับนี้สามารถนำไปใช้แก้ปัญหาในลักษณะเชิงเส้น (linear) และปัญหาที่ไม่เป็นเชิงเส้น (nonlinear) ได้ ทำให้ประยุกต์เข้ากับโครงงานต่างๆได้อย่างหลากหลายและได้รับความนิยมมากที่สุดในปัจจุบัน หลักการแพร่ย้อนกลับมีพื้นฐานมาจากกฎเดลต้า (Dealta – Rule) ที่พัฒนาขึ้นโดยวิดโรว์และฮอฟฟ์ ในช่วงปี ค.ศ. 1960 – 1962 ซึ่งกระบวนการของกฎเดลต้าจะลดค่าผิดพลาดที่เอาท์พุทของโครงข่ายเปรียบเทียบกับค่าเป้าหมายโดยค่าผิดพลาดที่ลดลงนี้เกิดจากการปรับแต่งค่าน้ำหนักที่อินพุทของแต่ละนิวรอล

2.9.1 กฎเดลต้า (Widrow – Hoff Delta Learning Rule)

2.9.1.1 กฎเดลต้าสำหรับเอาท์พุทชั้นเดียว

กฎเดลต้าจะใช้สำหรับเปลี่ยนน้ำหนักที่เชื่อมระหว่างนิวรอลซึ่งหาได้จากความแตกต่างที่น้อยที่สุดระหว่างผลรวมของอินพุทที่ส่งไปยังเอาท์พุท (y_in) และค่าเอาท์พุทเป้าหมาย t จุดประสงค์เพื่อทำให้ความผิดพลาดมีค่าน้อยที่สุดในการฝึกสอน ดังรูปที่ 2.35



รูปที่ 2.35 แสดงนิวรอลที่ไม่มีฟังก์ชันกระตุ้นแต่มีเอาท์พุทเป้าหมาย t และค่าผิดพลาด E

จากรูปที่ 2.35 ที่มาฟังก์ชันค่าความผิดพลาด E ของการฝึกสอนหาได้จาก

$$E = (t - (y_in))^2 \tag{2.16}$$

E เป็นฟังก์ชันของน้ำหนักทั้งหมด $(w_i, i = 1,...,n)$ เกรเดียนท์ของ E คือ เวกเตอร์ซึ่งประกอบด้วยอนุพันธ์ย่อยของ E เทียบกับแต่ละน้ำหนัก เกรเดียนท์จะทำให้ E เพิ่มขึ้นอย่างรวดเร็วและในทิศทางตรงกันข้ามจะทำให้ค่าผิดพลาดลดลงอย่างรวดเร็วเช่นกัน นั่นคือ ค่าผิดพลาดสามารถลดลงได้ด้วยการปรับปรุงน้ำหนัก w_I แสดงในรูปของ $-\frac{\partial E}{\partial w_I}$

จาก $y_in = \sum_{i=1}^{n} x_i w_i$

$$\begin{aligned}\frac{\partial E}{\partial w_I} &= -2(t - (y_in))\frac{\partial}{\partial w_I}(y_in) \\ &= -2(t - (y_in))x_I \end{aligned} \tag{2.17}$$

ดังนั้น ค่าผิดพลาดจะลดลงอย่างรวดเร็วด้วยการปรับปรุงน้ำหนักโดยกฎเดลต้า คือ

$$\Delta w_I = \alpha(t - (y_in))x_I \tag{2.18}$$

2.9.1.2 กฎเดลต้าสำหรับชั้นเอาท์พุทหลายชั้น

กฎเดลต้าจะใช้สำหรับเปลี่ยนน้ำหนักที่เชื่อมระหว่างนิวรอลซึ่งหาได้จากความแตกต่างระหว่างผลรวมของอินพุทที่ส่งไปยังเอาท์พุท y_in_J และค่าเอาท์พุทเป้าหมาย t_J เพื่อลดค่าผิดพลาดของแต่ละโครงสร้างจุดประสงค์เพื่อหาความผิดพลาดที่น้อยที่สุดในการฝึกสอน ซึ่งหาในทำนองเดียวกันกับกฎเดลต้าสำหรับเอาท์พุทชั้นเดียว นั่นคือ

$$E = \sum_{j=1}^{m} (t_j - (y_in_j))^2 \tag{2.19}$$

E เป็นฟังก์ชันของน้ำหนักทั้งหมด $(w_i, i = 1,...,n)$ เกรเดียนท์ของ E คือ เวกเตอร์ซึ่งประกอบด้วยอนุพันธ์ย่อยของ E เทียบกับแต่ละน้ำหนัก เกรเดียนท์จะทำให้ E เพิ่มขึ้นอย่างรวดเร็วและในทิศทางตรงกันข้ามจะทำให้ค่าผิดพลาดลดลงอย่างรวดเร็วเช่นกัน นั่นคือ ค่าผิดพลาดสามารถลดลงได้ด้วยการปรับปรุงน้ำหนัก w_{IJ} แสดงในรูปของ $-\frac{\partial E}{\partial w_{IJ}}$

$$\frac{\partial E}{\partial w_{IJ}} = \frac{\partial}{\partial w_{IJ}} \sum_{j=1}^{m} (t_J - (y_in_J))^2$$

$$= \frac{\partial}{\partial w_{IJ}} (t_J - (y_in_J))^2$$

เพราะว่า น้ำหนัก w_{IJ} มีผลกับค่าผิดพลาดของเอาท์พุทในชั้น Y_J เท่านั้น

จาก $y_in_J = \sum_{i=1}^{n} x_i w_J$ ดังนั้น

$$\frac{\partial E}{\partial w_{IJ}} = -2(t_J - (y_in_J)) \frac{\partial}{\partial w_{IJ}} (y_in_J)$$

$$= -2(t_J - (y_in_J))x_I \qquad (2.20)$$

ดังนั้น ค่าผิดพลาดจะลดลงได้อย่างรวดเร็วด้วยการปรับปรุงน้ำหนักโดยกฎเดลต้า คือ

$$\Delta w_{IJ} = \alpha(t_J - (y_in_J))x_I \qquad (2.21)$$

2.9.2 โครงข่ายประสาทเทียมแบบแพร่ย้อนกลับ (Backpropagation Neural Network , BPN)

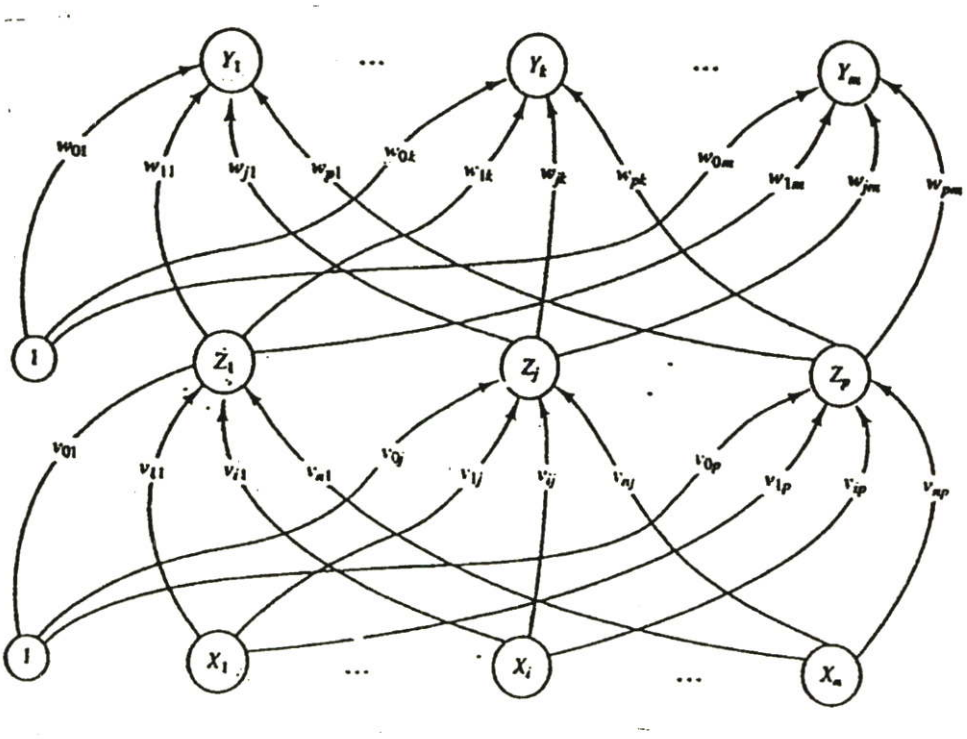
เป็นโครงข่ายที่มีจำนวนชั้นตั้งแต่ 3 ชั้นขึ้นไป คือ ชั้นอินพุท (input layer) ชั้นซ่อน (hidden layer) และชั้นเอาท์พุท (output layer) ลักษณะของโครงข่ายประสาทเทียมแบบแพร่ย้อนกลับเป็น feedforward network โดยมีการเชื่อมโยงในแต่ละชั้นแบบสมบูรณ์ นั่นคือ ทุกๆนิวรอลในชั้นอินพุทจะส่งสัญญาณไปยังทุกๆนิวรอลในชั้นซ่อนชั้นแรก และทุกๆนิวรอลในชั้นซ่อนชั้นแรกจะส่งสัญญาณไปยังทุกๆนิวรอลในชั้นถัดไป จนในที่สุดทุกๆนิวรอลในชั้นซ่อนสุดท้ายจะส่งสัญญาณไปยังทุกๆนิวรอลในชั้นเอาท์พุท

โครงข่ายประสาทเทียม เรียกได้ว่าเป็น mapping network สิ่งสำคัญที่ทำให้การ mapping ถูกต้องนั้นจำเป็นต้องมีการฝึกหัดโครงข่าย และการฝึกหัดโครงข่ายโดยอาศัยหลักการแพร่ย้อนกลับนั้นแบ่งออกได้เป็น 3 ขั้นตอน คือ

1. แพร่สัญญาณที่ชั้นอินพุทไปยังชั้นซ่อนแรก และแพร่สัญญาณไปยังชั้นซ่อนถัดไปจนถึงชั้นเอาท์พุทโดยผ่านฟังก์ชันกระตุ้น

2. คำนวณค่าผิดพลาดของค่าเอาท์พุทกับค่าเป้าหมายและแพร่ย้อนกลับค่าผิดพลาดเหล่านี้ไปยังชั้นซ่อน

3. ปรับแต่งค่าน้ำหนักที่ชั้นเอาท์พุทและชั้นซ่อนเพื่อลดค่าผิดพลาด โดยทำให้ค่าเอาท์พุทมีค่าใกล้เคียงหรือเท่ากับค่าเป้าหมาย



รูปที่ 2.36 แสดงโครงข่ายประสาทเทียมแบบแพร่ย้อนกลับแบบมีชั้นซ่อน 1 ชั้น

2.9.3 การฝึกสอนโครงข่ายประสาทเทียมแบบแพร่ย้อนกลับ

การฝึกสอนโครงข่ายประสาทเทียมแบบแพร่ย้อนกลับมี 3 ขั้นตอนดังกล่าวมาแล้ว คือ การแพร่สัญญาณอินพุทในการฝึกสอน (feedforward) การคำนวณค่าผิดพลาด (backpropagation) และการปรับแต่งค่าน้ำหนัก (adjustment of weight) จากรูปที่ 2.36 อธิบายโดยละเอียดได้ดังนี้

ในระหว่างขั้นตอน feedforward แต่ละนิวรอล X_i จะส่งสัญญาณอินพุทและกระจายสัญญาณไปแต่ละนิวรอลในชั้นซ่อน $Z_1,...,Z_p$ แต่ละนิวรอลในชั้นซ่อนจะคำนวณค่าด้วยฟังก์ชันกระตุ้นและส่งสัญญาณ z_j ไปให้แต่ละนิวรอลเอาท์พุทในชั้นเอาท์พุท แต่ละนิวรอลเอาท์พุท Y_k จะคำนวณค่าด้วยฟังก์ชันกระตุ้นและส่งสัญญาณ y_k เป็นค่าเอาท์พุทของอินพุทตัวหนึ่งๆ

ในระหว่างการฝึกสอน แต่ละนิวรอลเอาท์พุทจะเปรียบเทียบค่า y_k กับค่าเอาท์พุทเป้าหมาย t_k เพื่อหาค่าผิดพลาด $\delta_k (k = 1,...,m)$ เมื่อคำนวณ δ_k ได้แล้ว ก็ทำการกระจายค่าผิดพลาดนั้นมาที่นิวรอลในชั้นซ่อน เพื่อใช้ในการปรับปรุงค่าน้ำหนักระหว่างชั้นเอาท์พุทกับชั้นซ่อน ในทำนองเดียวกัน $\delta_j (j = 1,..., p)$ คือ ค่าผิดพลาดที่ถูกคำนวณขึ้นมาสำหรับแต่ละนิวรอลในชั้นซ่อน Z_j และ δ_j จะใช้เป็นตัวปรับปรุงค่าน้ำหนักระหว่างชั้นซ่อนกับชั้นอินพุท

หลังจากสามารถหา δ ทุกตัวได้แล้ว ค่าน้ำหนักของทุกชั้นจะถูกปรับปรุงใหม่ และเริ่มกระบวนการ feedforward ใหม่อีกครั้ง จนกระทั่งได้เงื่อนไขของค่าผิดพลาดที่ต้องการหรือยอมรับได้

2.9.4 อัลกอริทึมการฝึกสอนโครงข่ายประสาทเทียมแบบแพร่ย้อนกลับ (Training Algorithm of Backpropagation)

พิจารณาจากรูปที่ 2.36 ขั้นตอนการฝึกสอนมีดังนี้

ขั้นที่ 1 ทำการสุ่มค่าที่อยู่ใน [0,1] ให้กับค่าน้ำหนักทุกตัว (v_{ij}, w_{jk})

กำหนดเงื่อนไขค่าความผิดพลาดที่ต้องการหรือยอมรับได้

กำหนดอัตราการเรียนรู้ (α) โดยเริ่มจากค่าประมาณที่ไม่ต่ำหรือสูงจนเกินไป

feedforward

ขั้นที่ 2 แต่ละนิวรอลอินพุท $(X_i, i = 1,...,n)$ จะรับสัญญาณอินพุท x_i และแพร่กระจายสัญญาณไปทุกๆนิวรอลในชั้นถัดไป (ชั้นซ่อน)

ขั้นที่ 3 แต่ละนิวรอลในชั้นซ่อน $(Z_j, j = 1,..., p)$ ผลรวมของน้ำหนักสัญญาณอินพุท คือ

$$z_in_j = v_{0j} + \sum_{i=1}^{n} x_i v_{ij} \tag{2.22}$$

โดย v_{0j} คือ bias weight

นำมาคำนวณกับฟังก์ชันกระตุ้น เพื่อเป็นสัญญาณเอาท์พุท

$$z_j = f(z_in_j) \tag{2.23}$$

และส่งสัญญาณไปทุกๆนิวรอลในชั้นถัดไป (ชั้นเอาท์พุท)

ขั้นที่ 4 แต่ละนิวรอลในชั้นเอาท์พุท $(Y_k, k = 1,..., m)$ ผลรวมของน้ำหนักสัญญาณอินพุท คือ

$$y_in_k = w_{0k} + \sum_{j=1}^{p} z_j w_{jk} \tag{2.24}$$

โดย w_{0k} คือ bias weight

นำมาคำนวณกับฟังก์ชันกระตุ้น เพื่อเป็นสัญญาณเอาท์พุท

$$y_k = f(y_in_k) \tag{2.25}$$

backpropagation

ขั้นที่ 5 แต่ละนิวรอลเอาท์พุท $(Y_k, k = 1,..., m)$ จะนำเอาท์พุทเป้าหมายมาเปรียบเทียบกับเอาท์พุทที่ได้ เพื่อคำนวณหาค่าผิดพลาดของเอาท์พุทในรูปของฟังก์ชันค่าความผิดพลาด คือ

$$\delta_k = (t_k - y_k) f'(y_in_k) \tag{2.26}$$

คำนวณน้ำหนักที่จะใช้ปรับปรุง w_{jk} ในชั้นเอาท์พุท

$$\Delta w_{jk} = \alpha \delta_k z_j \tag{2.27}$$

คำนวณน้ำหนักที่จะใช้ปรับปรุง w_{0k} ในชั้นเอาท์พุท

$$\Delta w_{0k} = \alpha \delta_k \tag{2.28}$$

และส่ง δ_k กลับมาที่ชั้นซ่อนชั้นถัดลงมา

ขั้นที่ 6 แต่ละนิวรอลในชั้นซ่อน $(Z_j, j = 1,..., p)$ ผลรวมของ delta อินพุทจากชั้นข้างบน

$$\delta_in_j = \sum_{k=1}^{m} \delta_k w_{jk} \tag{2.29}$$

นำมาคูณกับอนุพันธ์ของฟังก์ชันกระตุ้น เพื่อคำนวณหาค่าผิดพลาดในชั้นซ่อน

$$\delta_j = \delta_in_j f'(z_in_j) \tag{2.30}$$

คำนวณน้ำหนักที่จะใช้ปรับปรุง v_{ij} ในชั้นซ่อน

$$\Delta v_{ij} = \alpha \delta_j x_i \tag{2.31}$$

คำนวณน้ำหนักที่จะใช้ปรับปรุง v_{0j}

$$\Delta v_{0j} = \alpha \delta_j \tag{2.32}$$

update weight and bias

ขั้นที่ 7 แต่ละนิวรอลเอาท์พุท $(Y_k, k = 1,..., m)$ จะถูกปรับปรุงด้วย bias และน้ำหนักค่าใหม่ $(j = 0,..., p)$

$$w_{jk}(new) = w_{jk}(old) + \Delta w_{jk} \tag{2.33}$$

แต่ละนิวรอลในชั้นซ่อน $(Z_j, j = 1,..., p)$ จะถูกปรับปรุงด้วย bias และน้ำหนักค่าใหม่ $(i = 0,..., n)$

$$v_{ij}(new) = v_{ij}(old) + \Delta v_{ij} \tag{2.34}$$

ขั้นที่ 8 ทดสอบเงื่อนไขในการจบการทำงาน

ขั้นที่ 9 จบขั้นตอนการทำงาน

จากขั้นที่ 3 และขั้นที่ 4 ฟังก์ชันกระตุ้นที่เลือกใช้สำหรับโครงข่ายแบบแพร่ย้อนกลับนั้นเป็นสิ่งที่มีความสำคัญมาก ในการเลือกฟังก์ชันกระตุ้นการตัดสินใจสิ่งที่ต้องพิจารณา คือ ฟังก์ชันนั้นต้องมีความต่อเนื่อง (continuous) สามารถหาอนุพันธ์ได้ (differentiable) และเป็นฟังก์ชันแบบไม่ลดลงทางเดียว (monotonically nondecreasing) ยิ่งกว่านั้น การคำนวณจะมีประสิทธิภาพมากขึ้น

ถ้าปฏิยานุพันธ์ของฟังก์ชันกระตุ้นการตัดสินใจนั้นสามารถคำนวณได้ง่ายๆ ฟังก์ชันกระตุ้นการตัดสินใจที่ใช้กันทั่วๆไป ค่าของปฏิยานุพันธ์จะต้องสามารถแสดงในรูปค่าของฟังก์ชันได้ โดยปกติฟังก์ชันที่คาดว่าจะคงตัว (saturate) คือ ฟังก์ชันที่ลู่เข้าสู่ค่าจำกัดที่มากที่สุดและน้อยที่สุด

ฟังก์ชันกระตุ้นการตัดสินใจที่นิยมใช้กันมาก คือ ฟังก์ชัน binary sigmoid ดังรูปที่ 2.32 และฟังก์ชัน bipolar sigmoid ดังรูปที่ 2.33 และฟังก์ชันกระตุ้นการตัดสินใจที่เลือกใช้นั้นควรจะเป็นฟังก์ชันเดียวกันตลอดการฝึกสอน จากขั้นที่ 5 และขั้นที่ 6 มีการคำนวณค่าผิดพลาดซึ่งจะถูกนำมาปรับปรุงค่าน้ำหนักในขั้นที่ 7 สาเหตุนี้ทำให้ถูกเรียกว่า "backpropagation" ส่วน f' เป็นอนุพันธ์อันดับที่หนึ่งของฟังก์ชันกระตุ้นการตัดสินใจ f

คณิตศาสตร์พื้นฐานสำหรับอัลกอริทึมของการแพร่ย้อนกลับ คือ การหาค่าที่เหมาะสมที่สุด (optimization technique) ที่รู้จักกันดีในนามของ "gradient descent" เกรเดียนท์ของฟังก์ชัน(ในที่นี้คือ ฟังก์ชันค่าความผิดพลาดและตัวแปรต่างๆ เช่น น้ำหนักของโครงข่าย) จะต้องคำนึงถึงทิศทางด้วย นั่นคือ ถ้ามีทิศทางเป็นบวก (positive) ฟังก์ชันจะเพิ่มขึ้น(increases) อย่างรวดเร็วแต่ถ้าทิศทางเป็นลบ (negative) ฟังก์ชันจะลดลง (decreases) อย่างรวดเร็ว (ภาคผนวก ข)

2.10 การออกแบบโครงข่ายประสาทเทียม (Design of the Neural Network)

2.10.1 ขนาดของโครงข่ายประสาทเทียม (Size of Neural Network)

ขนาดของโครงข่ายประสาทเทียมจะมีความสัมพันธ์กับลักษณะของปัญหาและประสบการณ์ของผู้ใช้ ผู้ที่เริ่มใช้แรกๆมักจะกำหนดให้โครงข่ายมีขนาดเล็กจึงทำให้ความสามารถในการใช้งานลดลงไปด้วย ซึ่งจริงๆแล้วความสามารถในการใช้งานควรจะเป็นตัวกำหนดขนาดของโครงข่ายมากกว่า และบางครั้งโครงข่ายที่มีขนาดใหญ่มากๆ เช่น มีจำนวนนิวรอล 1,000 หรือ 1 แสนนิวรอล ก็อาจไม่เหมาะสมกับปัญหาหรือลักษณะของข้อมูลที่ใช้

2.10.2 จำนวนของชั้นในโครงข่าย (Number of Layers)

โครงข่ายประสาทเทียมแบบแพร่ย้อนกลับมักจะมี 3 ชั้น แต่ก็ไม่จำเป็นว่าทุกๆปัญหาจะต้องมี 3 ชั้นอาจมีมากกว่าหรือน้อยกว่าก็ได้ตามความเหมาะสม ซึ่งในบางครั้งการใช้ชั้นซ่อนเล็กๆ 2 ชั้นอาจจะดีกว่าการใช้ชั้นซ่อนใหญ่ๆเพียงชั้นเดียวก็ได้ ลักษณะของชั้นซ่อนจะขึ้นอยู่กับสมมติฐานและการลดทอนลักษณะของข้อมูลอินพุท อย่างไรก็ตาม การที่มีชั้นซ่อนหลายๆชั้นก็จะเป็นการเพิ่มประสิทธิภาพของโครงข่ายแต่ในขณะเดียวกันก็ทำให้ต้องใช้เวลาและตัวอย่างในการฝึกสอนมากขึ้นเช่นกัน

2.10.3 จำนวนของนิวรอลในแต่ละชั้น (Number of Neurons in Each Layers)

จำนวนของนิวรอลในชั้นอินพุทและเอาท์พุทจะกำหนดได้จากลักษณะของปัญหา เช่น หากโครงข่ายมีข้อมูลอินพุทเป็นสเปกตัมของฟังก์ชันความหนาแน่น 128 จุด และต้องการแบ่งแยกออกเป็น 10 กลุ่ม นั่นคือ เราต้องมีนิวรอลในชั้นอินพุท 128 นิวรอล และในชั้นเอาท์พุท 10 นิวรอล ส่วน

นิวรอลในชั้นซ่อนก็ทำการทดลองไปเรื่อยๆว่าเท่าไรจึงจะเหมาะสม ถ้าใช้จำนวนนิวรอลในชั้นซ่อนน้อยเกินไปก็จะทำให้เกิดความผิดพลาดในการแยกประเภทและต้องใช้เวลาในการฝึกสอนมาก แต่ถ้าใช้จำนวนนิวรอลในชั้นซ่อนมากเกินไปก็จะทำให้ระบบจดจำรูปแบบมากเกินไป โดยไม่มีการลดทอนลักษณะเด่นออกไปจากอินพุททำให้โครงข่ายไม่สามารถหาลักษณะพื้นฐานของรูปแบบได้ ดังนั้น เมื่อเราต้องการกำหนดรูปแบบใหม่ให้กับโครงข่าย โครงข่ายก็จะไม่สามารถดำเนินการหาลักษณะที่แท้จริงของข้อมูลได้

2.10.4 การเลือกค่าน้ำหนักและ bias

2.10.4.1 การเลือกแบบสุ่ม (Random Initialization)

การเลือกค่าน้ำหนักที่เหมาะสมจะทำให้ระบบสามารถลู่เข้าสู่ค่าที่ทำให้ค่าผิดพลาดที่น้อยที่สุดได้และหาได้อย่างรวดเร็ว การปรับปรุงค่าน้ำหนักระหว่างชั้นทั้งสองชั้นขึ้นอยู่กับปฏิยานุพันธ์ทั้งสองฟังก์ชันของฟังก์ชันกระตุ้นการตัดสินใจของชั้นข้างบนและชั้นข้างล่าง ด้วยเหตุนี้ จึงมีความสำคัญมากในการเลือกค่าน้ำหนักที่จะไม่ทำให้ทั้งฟังก์ชันกระตุ้นและปฏิยานุพันธ์ของฟังก์ชันกระตุ้นนั้นเป็น 0 ค่าของน้ำหนักที่เลือกจะต้องไม่ใหญ่มากเกินไป หรือสัญญาณอินพุทของแต่ละนิวรอลในชั้นซ่อนหรือชั้นเอาท์พุทจะต้องมีค่าอยู่ในขอบเขตซึ่งปฏิยานุพันธ์ของ sigmoid function มีค่าเล็กๆ เรียกว่า "saturation region" ในอีกทางหนึ่ง ถ้าค่าน้ำหนักที่เลือกมีค่าเล็กมากเกินไป ผลรวมของอินพุทในชั้นซ่อนหรือชั้นเอาท์พุทจะเข้าใกล้ 0 ซึ่งเป็นเหตุให้การเรียนรู้เป็นไปได้ช้า

โดยทั่วๆไป การเลือกค่าน้ำหนักจะทำการสุ่มค่าที่อยู่ระหว่าง –0.5 และ 0.5 (หรือระหว่าง –1 และ 1 หรือช่วงอื่นๆที่เหมาะสม) ค่าน้ำหนักที่เลือกอาจเป็นค่าบวกหรือค่าลบก็ได้เพราะน้ำหนักสุดท้ายหลังผ่านการฝึกสอนจะเป็นได้ทั้งสองเครื่องหมาย

2.10.4.2 การเลือกแบบ Nguyen – Widrow (Nguyen – Widrow Initialization)

เป็นวิธีการเลือกค่าน้ำหนักซึ่งดัดแปลงมาจากการเลือกค่าน้ำหนักแบบสุ่ม ซึ่งจะช่วยให้การเรียนรู้เกิดได้เร็วขึ้น การลู่เข้าของการวิเคราะห์ทางคณิตศาสตร์พิจารณาจากการตอบรับของนิวรอลในชั้นซ่อนกับสัญญาณอินพุท ค่าน้ำหนักที่ส่งจากชั้นซ่อนไปยังชั้นเอาท์พุทและ bias ของชั้นเอาท์พุทจะถูกเลือกมาอย่างสุ่มจากค่าระหว่าง –0.5 และ 0.5 เหมือนกรณีทั่วๆไป

การเลือกค่าน้ำหนักจากชั้นอินพุทไปยังชั้นซ่อนจะถูกออกแบบให้สามารถปรับปรุงการเรียนรู้ของชั้นซ่อนได้ โดยการกระจายค่าน้ำหนักและ bias ลงไปในแต่ละโครงสร้างอินพุททำให้นิวรอลในชั้นซ่อนเรียนรู้ได้เร็วขึ้น

กำหนดให้ n คือ จำนวนของนิวรอลอินพุท

p คือ จำนวนของนิวรอลในชั้นซ่อน

β คือ ตัวประกอบปรับขนาด (scale factor)

$$\beta = 0.7(p)^{1/n} = 0.7\sqrt[n]{p}$$

ขั้นตอนการทำงานแบบง่ายๆคือ

สำหรับนิวรอลแต่ละนิวรอลในชั้นซ่อน (j = 1 , … ,p) ทำการเลือกเวกเตอร์น้ำหนัก(จากนิวรอลอินพุท)

$v_{ij}(old)$ = ค่าที่ได้จากการสุ่มระหว่าง –0.5 และ 0.5 (หรือระหว่าง $-\gamma$ และ γ)

คำนวณ $\|V_j(old)\|$

ทำการเลือกค่าน้ำหนักอีกครั้งโดย

$$v_{ij} = \frac{\beta v_{ij}(old)}{\|V_j(old)\|}$$

กำหนด bias

v_{0j} = ค่าที่เกิดจากการสุ่มระหว่าง $-\beta$ และ β

2.10.5 ระยะเวลาในการฝึกสอนโครงข่าย (Time to train the net)

โดยปกติแล้ว กระบวนการแพร่ย้อนกลับจะประสบความสำเร็จได้เมื่อเกิดความสมดุลระหว่างผลตอบรับที่ถูกต้องในการฝึกสอน และผลตอบรับที่ดีของอินพุทตัวใหม่ที่ใส่เข้าไป นั่นคือเกิดความสมดุลระหว่างการจดจำรูปแบบที่ฝึกสอนและกรณีตัวอย่างทั่วๆไป แต่เราไม่จำเป็นที่จะต้องนำเอาทั้งสองสิ่งนั้นมาหาค่าผิดพลาดที่น้อยที่สุดทุกครั้งไป Hecht – Nielsen(1990) ได้แนะนำว่า เราสามารถใช้เซตของรูปแบบที่ใช้ฝึกสอน และเซตของตัวอย่างที่ใช้ทดสอบการฝึกสอนมาตรวจสอบแทนก็ได้ โดยเซตทั้งสองนั้นจะต้องแยกออกจากกัน (disjoint)

การปรับค่าน้ำหนักจะใช้เป็นพื้นฐานของรูปแบบที่ใช้ฝึกสอน ในขณะที่ค่าผิดพลาดที่คำนวณได้จะถูกใช้เป็นตัวแทนของตัวอย่างที่ใช้ในการทดสอบการฝึกสอน เมื่อทำการฝึกสอนไปเรื่อยๆค่าผิดพลาดของตัวอย่างที่ใช้ทดสอบการฝึกสอนจะลดลง การฝึกสอนก็จะยังคงดำเนินต่อไปเรื่อยๆแต่ถ้าค่าผิดพลาดเริ่มที่จะเพิ่มขึ้น แปลว่า โครงข่ายต้องการรูปแบบการฝึกสอนที่เฉพาะเจาะจงมากขึ้นเมื่อถึงจุดนี้ การฝึกสอนก็จะสิ้นสุดลง

2.10.6 จำนวนคู่ของการฝึกสอน (Number of training pairs)

ความสัมพันธ์ระหว่างจำนวนรูปแบบการฝึกสอน (P) , จำนวนของน้ำหนักที่ใช้ในการฝึกสอน (W) และค่าคาดหวังของความแน่นอนในการแยกประเภท (e) Baum และ Haussler(1989) ได้เสนอไว้ว่า จากประโยค " เราจะแน่ใจได้อย่างไรว่า ภายใต้เหตุการณ์ต่างๆโครงข่ายที่นำมาฝึกสอนจะมีความถูกต้องในการแยกประเภทตัวอย่างใดๆที่เลือกมาเท่าไร และ ถ้าโครงข่ายที่ถูกฝึกสอน มีค่าความถูกต้องของการฝึกสอน เท่ากับ $1-\frac{e}{2}$ เมื่อ $0 < e \le \frac{1}{8}$ เราสามารถมั่นใจได้หรือไม่ว่า โครงข่ายจะแยกประเภทตัวอย่างที่ใช้ทดสอบการฝึกสอนได้ถูกต้องเท่ากับ $1-e$ "

เราจะสามารถมั่นใจได้ ถ้าโครงข่ายมีจำนวนรูปแบบการฝึกสอนที่มากเพียงพอ ซึ่งจำนวนรูปแบบการฝึกสอนที่เพียงพอ หาได้จากเงื่อนไขดังต่อไปนี้

$$\frac{W}{P} = e$$

หรือ

$$P = \frac{W}{e}$$

ตัวอย่างเช่น ถ้า e = 0.1 และ W = 80 แล้ว จำนวนรูปแบบการฝึกสอนที่ต้องการคือ 800 รูปแบบและจะให้ความถูกต้องในการแยกประเภทตัวอย่างที่ใช้ทดสอบ 90% โดยสมมติว่า โครงข่ายที่ใช้ฝึกสอนจะมีความถูกต้องของการฝึกสอน 95%

2.11 งานวิจัยที่เกี่ยวข้อง

สุรพันธุ์ เอื้อไพบูลย์ [1] นำเสนอการตัวอักษรลายมือเขียนภาษาไทยโดยการพิจารณาหัวของตัวอักษร โดยทำการทดสอบกับตัวอักษรลายมือเขียนจากคน 15 คน จำนวน 960 ตัว โดยมีแนวคิดที่ว่า หัวของตัวอักษรแต่ละตัวจะมีตำแหน่งที่ต่างๆกันออกไปหรือบางตัวอักษรก็ไม่มีหัวเลย จึงถือได้ว่าหัวเป็นลักษณะที่เด่นชัดประการหนึ่งของตัวอักษรภาษาไทย และสังเกตได้อีกอย่างหนึ่งว่า ในการเขียนตัวอักษรภาษาไทยตามการเขียนที่ถูกต้อง จะเริ่มโดยการเขียนหัวของตัวอักษรก่อนเสมอ ยกเว้นตัวอักษรบางตัวที่ไม่มีหัวเท่านั้น จึงได้นำเอาลักษณะเด่น คือ หัวของตัวอักษรมาใช้ในการจำแนกกลุ่มของตัวอักษร

หัวของตัวอักษรภาษาไทยในวิทยานิพนธ์นี้แบ่งได้อย่างหยาบๆมี 4 ลักษณะ คือ กลุ่มที่ 1 ไม่มีหัว กลุ่มที่ 2 มี 1 หัว กลุ่มที่ 3 มี 2 หัว และกลุ่มที่ 4 มี 3 หัว โดยในแต่ละกลุ่มจะทำการแบ่งลักษณะหรือตำแหน่งของหัวของตัวอักษรออกเป็นลักษณะย่อยๆอีก จนกระทั่งได้ลักษณะของหัวที่เป็นตัวแทนของตัวอักษรตัวนั้นๆได้

สกุล คำนวนชัย [2] นำเสนอการรู้จำตัวอักษรคัดลายมือเขียนภาษาไทยโดยประยุกต์ใช้การทรานสฟอร์มแบบคาร์ฮูเน็นเลิฟเข้ากับโครงข่ายฟัซซี่นิวรอล โดยได้พัฒนาระบบการรู้จำตัวอักษรคัดลายมือเขียนภาษาไทย โดยได้ทำการทดสอบกับตัวอักษรพยัญชนะทั้งหมด 44 ตัว ซึ่งได้ประยุกต์การแปลงข้อมูลแบบคาร์ฮูเน็นเลิฟเพื่อหาข้อมูลที่เป็นตัวแทนพยัญชนะแต่ละตัว โดยหาจากตัวอักษรแต่ละพยัญชนะที่มีรูปแบบต่างกัน 30 รูปแบบได้จากผู้คัดลายมือ 10 คนโดยเขียนคนละ 3 ครั้ง จะได้ข้อมูลที่เป็นลักษณะร่วมของพยัญชนะแต่ละตัว นอกจากนั้นได้ทำการปรับค่าของข้อมูลโดยใช้หลักการของฟัซซี่เซตให้เหลือค่าความเป็นสมาชิกของฟัซซีคือ 1 , 0.2 , 0.4 , 0.6 ,

0.8 และ 1 ตามลำดับ เป็นการลดจำนวนฟีเจอร์ (feature) ของข้อมูลตัวแทนพยัญชนะลง ข้อมูลที่ได้ก็คือ ข้อมูลที่ใช้เป็นตัวแทนพยัญชนะแต่ละตัวเข้าไปเป็นข้อมูลอินพุทให้กับขั้นตอนการเรียนรู้โดยใช้โครงข่ายประสาทเทียมแบบแพร่ย้อนกลับซึ่งมีทั้งหมด 3 ชั้น คือ ชั้นอินพุท จำนวน 512 โหนด ชั้นซ่อน จำนวน 100 โหนด และชั้นเอาท์พุท จำนวน 44 โหนด

การทดลองในงานวิทยานิพนธ์นี้ได้ทำการทดลองทั้งหมด 4 ลักษณะ โดยที่แต่ละลักษณะจะใช้วิธีที่แตกต่างกันเพื่อหาตัวแทนของพยัญชนะแต่ละตัวนำข้อมูลที่ได้ไปทำการเรียนรู้โดยใช้โครงข่ายประสาทเทียม วิธีแรกใช้วิธีการนำข้อมูลมาหาค่าเฉลี่ยแบบธรรมดาได้ผลการรู้จำถูกต้องร้อยละ 80 วิธีที่สองใช้วิธีการแปลงข้อมูลแบบคาร์ฮูเน็นเลิฟได้ผลการรู้จำถูกต้องร้อยละ 80 วิธีที่สามใช้วิธีการนำข้อมูลมาหาค่าเฉลี่ยแบบธรรมดาแล้วปรับข้อมูลให้อยู่ในระบบของฟัซซีเซตได้ผลการรู้จำถูกต้องร้อยละ 80 วิธีที่สี่ใช้วิธีการแปลงข้อมูลแบบคาร์ฮูเน็นเลิฟปรับข้อมูลให้อยู่ในระบบฟัซซีเซตได้ผลการรู้จำถูกต้องร้อยละ 85

จากการทดลองในหลายๆลักษณะทำให้ทราบว่า การหาตัวแทนของตัวอักษรแต่ละพยัญชนะโดยการแปลงข้อมูลแบบคาร์ฮูเน็นเลิฟกับค่าเฉลี่ยแบบทำธรรมดา จะทำให้ได้ผลการทดลองที่เหมือนกันแต่ผลลัพธ์ที่ได้จะให้ค่าที่มีความละเอียดมากกว่า สำหรับการนำฟัซซีเข้ามาปรับลดจำนวนฟีเจอร์ของข้อมูล จะทำให้ผลการทดลองถูกต้องเพิ่มขึ้นในอัตราร้อยละ 5 นอกจากนั้นยังทำให้เวลาในการเรียนรู้ลดลง

วิชิน มุ่มสิน นำเสนอการรู้จำตัวอักษรพิมพ์ภาษาลาวโดยใช้ฟัซซีและโครงข่ายประสาทเทียมแบบแพร่ย้อนกลับ โดยทำการปรับข้อมูลภาพตัวอักษรที่ได้จากเครื่องสแกนเนอร์ที่เป็นบิทแมพ (bitmap) ให้เป็นข้อมูลแบบไบนารี (binary) จากนั้นทำการแยกตัวอักษรออกจากหน้ากระดาษแล้วปรับตัวอักษรให้เป็นรูปแบบมาตรฐานอยู่ในกรอบ 90x90 จุดภาพ จากนั้นกำหนดค่าคุณลักษณะเด่นของตัวอักษรโดยคำนวณตามจุดภาพเนื้อตัวอักษร แล้วนำค่าที่ได้ส่งเข้าไปหาค่าความเป็นสมาชิกในระบบของฟัซซีโดยใช้ฟังก์ชันค่าความเป็นสมาชิกแบบสามเหลี่ยม ค่าฟัซซีที่ได้จะถูกส่งเข้าไปเรียนรู้ในโครงข่ายประสาทเทียมแบบแพร่ย้อนกลับที่มี 3 ชั้น คือ ชั้นอินพุท จำนวน 9 โหนด ชั้นซ่อน จำนวน 120 โหนด และชั้นเอาท์พุท จำนวน 46 โหนด

ในการทดสอบใช้ข้อมูลภาพตัวอักษรที่ผ่านการหาค่าลักษณะเด่นทางฟัซซีแล้ว 8 รูปแบบแต่ละรูปแบบตัวอักษรจะมี 4 ขนาดแตกต่างกัน คือ 14 , 16 , 18 และ 20 point ตามลำดับ พบว่า การลดขนาดของข้อมูลตัวอักษรและการประยุกต์ใช้ค่าแบบฟัซซีเข้ามาฝึกสอนทำให้เวลาในการฝึกสอนลดลง การทดสอบในการรู้จำตัวอักษรที่มีรูปแบบเดียวกันแต่ขนาดต่างกันค่าความถูกต้องได้เฉลี่ยร้อยละ 93.50 การทดสอบโดยใช้ตัวอักษรที่มีรูปแบบแตกต่างกันพบว่า ค่าความถูกต้องได้เฉลี่ยร้อยละ 92.73

บทที่ 3
วิธีดำเนินการวิจัย

เนื้อหาในบทนี้จะกล่าวถึงขั้นตอนต่างๆที่ทำในงานวิจัย โดยแบ่งออกเป็น 3 ขั้นตอนดังนี้ คือ

ขั้นตอนการเตรียมข้อมูล (Preprocessing) เป็นขั้นตอนที่รับข้อมูลบิทแมทไฟล์ภาพลายมือเขียนตัวบรรจงแต่ละตัวอักษรที่เขียนอยู่ในช่องที่กำหนดนำมาแปลงเป็นรหัสเลขฐานสองจำนวน 52 บิท โดยผ่านกระบวนการกำจัดสัญญาณรบกวน การปรับปรุงคุณภาพของตัวอักษร การสำรวจพื้นที่ของภาพ การทำให้บางและการลบกิ่งและขลิบปลาย จะได้เส้นโครงร่างของตัวอักษรที่มีความกว้างเพียง 1 จุดภาพ จากนั้นนำเส้นโครงร่างที่ได้มาแปลงเป็นรหัสเงื่อนไข และแปลงรหัสเงื่อนไขเป็นเวกเตอร์ แปลงเวกเตอร์เป็นแบบเปรียบพื้นฐาน เมื่อได้แบบเปรียบพื้นฐานที่เป็นตัวแทนของตัวอักษรแต่ละตัวแล้วทำการแปลงแบบเปรียบพื้นฐานนั้นๆให้เป็นรหัสเลขฐานสอง ซึ่งประกอบด้วยบิทข้อมูลโครงสร้างพื้นฐานและบิทข้อมูลลักษณะเด่น เพื่อใช้ในขั้นตอนการฝึกสอนและรู้จำต่อไป

ขั้นตอนการฝึกสอน (Training) เป็นขั้นตอนที่นำเอารหัสเลขฐานสองที่แทนโครงสร้างข้อมูลภาพตัวอักษรที่ได้จากขั้นตอนแรกมาฝึกสอนให้รู้จำ โดยใช้โครงข่ายประสาทเทียมแบบแพร่ย้อนกลับจำนวน 3 ชั้น ประกอบด้วยชั้นอินพุท 52 โหนด ชั้นซ่อน 36 โหนด และชั้นเอาท์พุท 44 โหนด จากนั้นทำการฝึกสอนโดยการสุ่มค่าถ่วงน้ำหนัก (weight) ที่เชื่อมระหว่างโหนดต่างๆในแต่ละชั้นของโครงข่ายประสาทเทียมแบบแพร่ย้อนกลับ ทำการคำนวณค่าอินพุทเพื่อหาค่าเอาท์พุทโดยผ่านฟังก์ชันก์กระตุ้นการตัดสินใจ นำค่าเอาท์พุทที่ได้มาเปรียบเทียบกับค่าเป้าหมายที่กำหนดไว้คำนวณหาค่าผิดพลาด ค่าผิดพลาดที่กำหนดอยู่ในช่วง $10^{-6} - 10^{-4}$ เมื่อได้ค่าถ่วงน้ำหนักที่เหมาะสมแล้วเก็บไว้ในรูปของเมทริกซ์ ใช้เป็นฐานข้อมูลในการทดสอบการรู้จำต่อไป

ขั้นตอนทดสอบการรู้จำ (Recognizing) เป็นขั้นตอนที่นำภาพตัวอักษรลายมือเขียนตัวบรรจงใดๆมาทำการทดสอบ โดยต้องนำภาพตัวอักษรคัดลายมือเขียนตัวบรรจงนั้นมาเข้าสู่กระบวนการเตรียมข้อมูลจนได้รหัสเลขฐานสองที่เป็นตัวแทนของอักษร จากนั้นนำมาเข้าสู่โครงข่ายประสาทเทียม นำค่าถ่วงน้ำหนักที่เก็บไว้ในฐานข้อมูลมาวิเคราะห์เพื่อหาว่าค่าถ่วงน้ำหนักของตัวอักษรใดให้ค่าผิดพลาดที่น้อยที่สุด เมื่อได้ค่าถ่วงน้ำหนักที่ให้ค่าผิดพลาดน้อยที่สุดแล้วโครงข่ายก็สามารถบอกได้ว่าตัวอักษรที่ต้องการทดสอบนั้นเป็นตัวอักษรใด

3.1 ขั้นตอนการเตรียมข้อมูล (Preprocessing)

3.1.1 ลักษณะของตัวอักษรที่ใช้ในการทดลอง

ในงานวิจัยนี้ เป็นการรู้จำตัวอักษรคัดลายมือเขียนภาษาไทย โดยจะทำการรู้จำเฉพาะตัวพยัญชนะ ก – ฮ ทั้งหมด 44 ตัว โดยเก็บข้อมูลจากผู้เขียนทั้งหมด 30 คน คนละ 1 ชุด ชุดละ 44 ตัว โดยเขียนลงในตารางที่กำหนดด้วยปากกาหมึกซึมหรือหมึกเจลสีดำ จากนั้นนำข้อมูลแต่ละชุดมาทำการสแกนด้วยเครื่องสแกนเนอร์เก็บเป็นไฟล์ภาพแบบขาวดำ (Black and white) ขนาดของตัวอักษร 100 x 100 จุดภาพ ดังรูปที่ 3.1 แสดงตัวอย่างของตัวอักษรที่ใช้ในงานวิจัย

รูปที่ 3.1 แสดงตัวอย่างลักษณะของตัวอักษรที่ทำการวิเคราะห์ (บางส่วน)

จากรูปที่ 3.1 ตัวอักษรที่ใช้ในงานวิจัยนี้ได้จากผู้คัด 30 คน โดยให้แต่ละคนคัดลงในช่องที่กำหนดคือคนที่ 1 คัดลงในคอลัมน์ที่ 1 คนที่ 2 คัดลงในคอลัมน์ที่ 2 จนถึงคนที่ 30 คัดลงในคอลัมน์ที่ 30 ตามลำดับ เมื่อได้ตัวอักษรที่ต้องการทดสอบครบแล้ว นำตัวอักษรนั้นมาเข้าสู่กระบวนการเตรียมข้อมูล ดังได้กล่าวไว้ในบทที่ 2 จนได้รหัสเงื่อนไขที่ใช้เป็นตัวแทนของตัวอักษรแต่ละตัว เพื่อใช้เป็นข้อมูลอินพุทสำหรับการฝึกสอนและทดสอบการรู้จำ โดยตัวอักษรของผู้คัดคนที่ 1 ถึง 5

จะใช้เป็นตัวต้นแบบในการฝึกสอน ส่วนในขั้นตอนการทดสอบการรู้จำจะใช้ตัวอักษรทั้งหมดมาทดสอบการรู้จำ (ลักษณะตัวอักษรที่ใช้ในงานวิจัยนี้ดูได้จากภาคผนวก ง)

3.2 ขั้นตอนการฝึกสอน (Training)

ในงานวิจัยนี้ได้เลือกวิธีโครงข่ายประสาทเทียมแบบแพร่ย้อนกลับ (Back – Propagation Neural Network) ที่มี 3 ชั้น คือ ชั้นอินพุท (Input Layer) ชั้นซ่อน (Hidden Layer) และ ชั้นเอาท์พุท (Output Layer) โดยที่ชั้นอินพุทก็คือ ข้อมูลที่ได้จากขั้นตอนการเตรียมข้อมูล จำนวนโหนดของแต่ละชั้นเท่ากับ 52 36 และ 44 ตามลำดับ กำหนดค่าถ่วงน้ำหนัก (weight) เริ่มต้นจากการสุ่ม อัตราการเรียนรู้ (Learning rate) อยู่ในช่วง [0 , 1] และค่าความผิดพลาด (Sum – Square Error) เท่ากับ $10^{-6} - 10^{-4}$ นำค่าถ่วงน้ำหนักในแต่ละโหนดที่ได้จากการฝึกสอน เก็บไว้ในหน่วยความจำในรูปของเมทริกซ์เพื่อนำไปใช้ในขั้นตอนการรู้จำ

ในการฝึกสอนข้อมูลแต่ละตัว จะใช้อักษรเป็นตัวต้นแบบตัวละ 5 ลักษณะ คือ ในการฝึกสอนให้โครงข่ายรู้จำตัว ก จะมีตัวอักษร ก เป็นอักษรต้นแบบ 5 ลักษณะ เมื่อได้อักษรต้นแบบแล้วก็นำมาผ่านขั้นตอนการเตรียมข้อมูลจนกระทั่งได้รหัสเลขฐานสองที่เป็นตัวแทนของอักษรต้นแบบนั้นๆ ซึ่งรหัสเลขฐานสองของอักษรต้นแบบทั้ง 5 ลักษณะ อาจเหมือนหรือต่างกันแล้วแต่ลักษณะของตัวอักษรต้นแบบเอง จากนั้นนำรหัสเลขฐานสองที่ได้มาเข้าสู่โครงข่ายประสาทเทียมแบบแพร่ย้อนกลับ เพื่อคำนวณหาค่าถ่วงน้ำหนักทั้งที่เชื่อมระหว่างชั้นอินพุทกับชั้นซ่อน และที่เชื่อมระหว่างชั้นซ่อนกับชั้นเอาท์พุท โดยดำเนินการคำนวณตามอัลกอริทึมที่กล่าวไว้ในบทที่ 2

เมื่อได้ค่าถ่วงน้ำหนักที่เชื่อมระหว่างชั้นที่ให้ผลการรู้จำถูกต้องแล้วจะเก็บค่าถ่วงน้ำหนักนั้นไว้ในรูปของเมทริกซ์ เพราะฉะนั้นตัวอักษรหนึ่งตัว จะมีค่าถ่วงน้ำหนักทั้งหมด 5 ชุด โดยเลือกใช้ฟังก์ชัน Binary sigmoid $f(x) = \dfrac{1}{1+\exp(-\sigma x)}$ เป็นฟังก์ชันกระตุ้นการตัดสินใจ (activation function)ในทุกๆชั้นของโครงข่ายประสาทเทียม

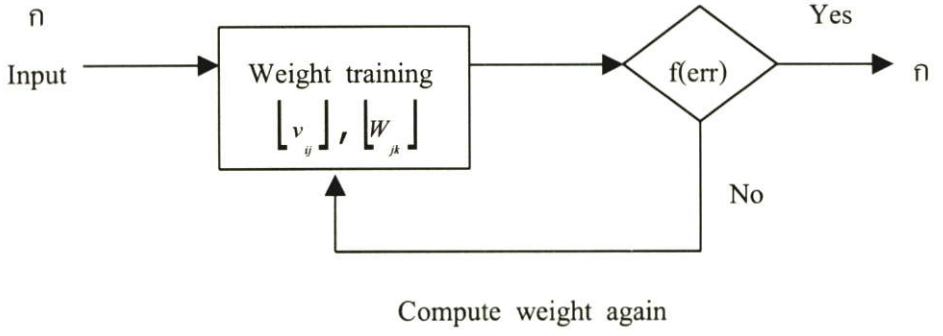
ค่าถ่วงน้ำหนักจากชั้นอินพุทไปยังชั้นซ่อนจะถูกเก็บไว้ในรูปเมทริกซ์ดังนี้

$$V_{ij}(i/p \rightarrow h) = \begin{bmatrix} v_{1,1} & \cdots & v_{1,36} \\ \vdots & & \vdots \\ v_{52,1} & \cdots & v_{52,36} \end{bmatrix}$$ เก็บไว้ในไฟล์ V.dat

ค่าถ่วงน้ำหนักจากชั้นซ่อนไปยังชั้นเอาท์พุทจะถูกเก็บไว้ในรูปเมทริกซ์ดังนี้

$$W_{jk}(h \rightarrow o/p) = \begin{bmatrix} w_{1,1} & \cdots & w_{1,44} \\ \vdots & & \vdots \\ w_{36,1} & \cdots & w_{36,44} \end{bmatrix}$$ เก็บไว้ในไฟล์ W.dat

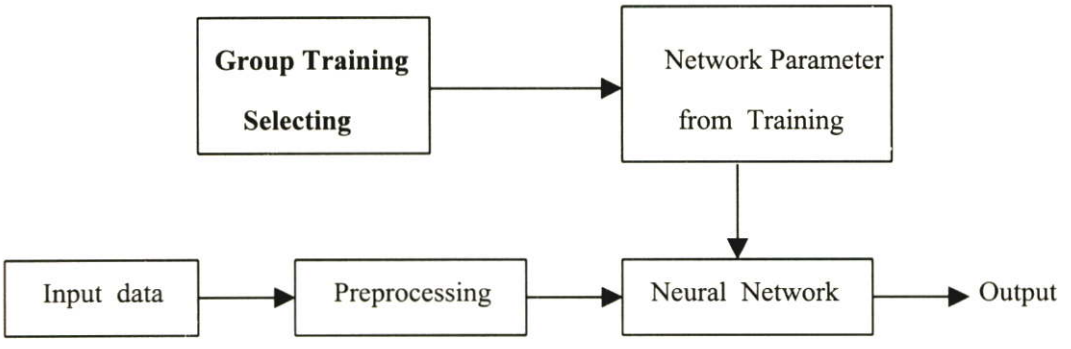
ค่าผิดพลาดที่ยอมรับได้ คือ f(err) $\approx 10^{-6} - 10^{-4}$ ถ้าเกิดปัญหาว่ามีค่าผิดพลาดใกล้เคียงกันจะเลือกตัวอักษรที่ค่าความผิดพลาดต่ำสุดหรือค่า 1- f(err) มีค่าใกล้เคียง 1 ที่สุด



รูปที่ 3.2 แสดงขั้นตอนการทำงานของการฝึกสอน

3.3 ขั้นตอนการรู้จำ (Recognizing)

โครงสร้างการทำงานในขั้นตอนนี้คือ



รูปที่ 3.3 แสดงโครงสร้างของการทดสอบ

ข้อมูลที่ต้องการทดสอบการรู้จำจะต้องผ่านขั้นตอนการเตรียมข้อมูลก่อน โดยการแยกตัวอักษรออกมาทดสอบทีละตัว จากนั้นวิเคราะห์เพื่อหารหัสเลขฐานสองที่ใช้เป็นตัวแทนของตัวอักษรตัวนั้นๆ เพื่อใช้ในการทดสอบ (test data) เมื่อได้รหัสเลขฐานสองแล้วก็นำมาเข้าสู่กระบวนการของโครงข่ายประสาทเทียมแบบแพร่ย้อนกลับ โดยนำค่าถ่วงน้ำหนักที่ได้จากการฝึกสอนมาใช้ แต่เนื่องจากค่าถ่วงน้ำหนักที่เก็บไว้มีหลายชุด ในวิทยานิพนธ์นี้จึงใช้ หลักของฟัซซี (Fuzzy) มาประยุกต์ใช้เพื่อช่วยในการพิจารณา ดังนี้คือ

ทำการแยกประเภทของตัวอักษรตามลักษณะของหัวที่ปรากฏ โดยดูจากแบบเปรียบพื้นฐานว่าตัวอักษรที่กำลังทดสอบมีหัวลักษณะใด เมื่อทราบลักษณะหัวแล้วก็นำค่าถ่วงน้ำหนัก (weight) แต่ละชุดของตัวอักษรที่เก็บไว้ในกลุ่มนั้น มาทดสอบว่าค่าถ่วงน้ำหนักชุดใดเมื่อนำมาทดสอบแล้วให้ค่าผิดพลาดน้อยที่สุดหรืออยู่ในช่วงที่ยอมรับได้ ก็จะสามารถบอกได้ว่าตัวอักษรที่กำลังทดสอบนั้นเป็นอักษรตัวใด ซึ่งวิธีการนี้จะทำประหยัดเวลาในการคำนวณและให้ความถูกต้องมากขึ้น เนื่องจากเราไม่จำเป็นต้องทำการค้นหาแบบเรียงลำดับตั้งแต่ ก – ฮ ทุกๆรอบการคำนวณ ซึ่งใช้เวลาในการประมวลผลมากและความผิดพลาดสูงกว่า

ตารางที่ 3.1 แสดงลักษณะหัวตัวอักษร

ลักษณะหัว	ตัวอักษร
C1	ข , ช , ฎ , ฏ , ภ , ร , ว
C2	ข , ฃ , ฆ , ช , ซ , ฌ , ญ , ฑ , ฒ , ถ , ล , ส , ศ
C3	ค , ฅ , ต , ผ , ฝ , ย , อ , ฮ
C4	ง , จ , ฉ , ฐ , ฒ , ด , ต , ท , น , บ , ป , พ , ฟ , ม , ษ , ห , ฬ
กลุ่มพิเศษ	ก , ธ

บทที่ 4
ผลการวิจัย

4.1 ผลการทดสอบการรู้จำ

งานวิจัยนี้ทำการแปลงข้อมูลภาพตัวอักษรคัดลายมือเขียนภาษาไทยตัวบรรจง จากผู้คัดลายมือ 30 คน คนละ 1 ชุด ชุดละ 44 ตัว โดยใช้กระบวนการดังที่กล่าวมาแล้วคือ นำมาผ่านกระบวนการเตรียมข้อมูล กระบวนการฝึกสอน และกระบวนการทดสอบรู้จำ

ในกระบวนการเตรียมข้อมูล นำพยัญชนะตัวอักษรแต่ละตัวที่ผู้คัดเขียนลงในช่องที่กำหนดมาผ่านเครื่องสแกนเนอร์เก็บเป็นไฟล์ภาพขาวดำขนาดของตัวอักษร 100x100 จุดภาพ จากนั้นทำการปรับข้อมูลไฟล์ภาพให้อยู่ในรูปรหัสเลขฐานสอง โดยใช้การวิเคราะห์ลักษณะเด่น จะได้รหัสเลขฐานสองที่เป็นตัวแทนของตัวอักษรแต่ละตัวเพื่อใช้ในการฝึกสอนและการทดสอบการรู้จำ

กระบวนการฝึกสอน นำรหัสเลขฐานสองที่ได้จากขั้นตอนการเตรียมและปรับข้อมูล มาใช้เป็นอินพุทเข้าสู่โครงข่ายประสาทเทียมแบบแพร่ย้อนกลับ เพื่อหาค่าถ่วงน้ำหนักในการจดจำรูปแบบระหว่างอินพุทกับเอาท์พุท โดยใช้วิธีการตามอัลกอริทึมการฝึกสอนโครงข่ายประสาทเทียมแบบแแพร่ย้อนกลับในหัวข้อที่ 2.8

กระบวนการทดสอบการรู้จำ นำภาพตัวอักษรที่ต้องการทดสอบมาผ่านการเตรียมข้อมูลจนได้รหัสเลขฐานสอง จากนั้นนำมาเข้าสู่โครงข่ายประสาทเทียมแบบแพร่ย้อนกลับโดยนำค่าถ่วงน้ำหนักที่ได้จากการฝึกสอนมาทดสอบว่าค่าถ่วงน้ำหนักตัวใดให้ผลที่มีค่าผิดพลาดน้อยที่สุด โครงข่ายประสาทเทียมก็จะสามารถสรุปได้ว่าตัวอักษรที่ต้องการทดสอบนั้นเป็นตัวอักษรใด

ตารางที่ 4.1 ผลการทดสอบความถูกต้องของตัวอักษร จากผู้คัดลายมือ 30 คน

ตัวอักษร	รู้จำผิดพลาด (ตัว)	ความถูกต้อง ร้อยละ (%)	ตัวอักษร	รู้จำผิดพลาด (ตัว)	ความถูกต้อง ร้อยละ (%)
ก	0	100	ท	0	100
ข	2	93.33	ธ	0	100
ฃ	3	90	น	0	100
ค	0	100	บ	4	86.67
ต	1	96.66	ป	2	93.33
ฅ	5	83.33	ผ	3	90
ง	3	90	ฝ	4	86.67
จ	0	100	พ	1	96.67
ฉ	4	86.67	ฟ	3	90
ช	3	90	ภ	2	93.33
ซ	4	86.67	ม	3	90
ฌ	0	100	ย	0	100
ญ	0	100	ร	0	100
ฎ	1	96.67	ล	0	100
ฏ	3	90	ว	0	100
ฐ	2	93.33	ศ	3	90
ฑ	3	90	ษ	6	80
ฒ	0	100	ส	6	80
ณ	0	100	ห	9	70
ด	4	86.67	ฬ	8	73.33
ต	6	80	อ	0	100
ถ	3	90	ฮ	3	90

จากตารางได้ค่าเฉลี่ยของความถูกต้องเท่ากับ 92.12%

จากผลการทดลองที่ได้ถ้านำมาเปรียบเทียบเปอร์เซ็นต์ความถูกต้องกับผลการทดลองการรู้จำตัวอักษรคัดลายมือเขียนภาษาไทยโดยประยุกต์ใช้การทรานสฟอร์มแบบคาร์ฮูเน็นเลิฟเข้ากับโครงข่ายฟัซซีนิวรอลของสกุล คำนวนชัย ได้ผลดังนี้

ตารางที่ 4.2 แสดงผลการเปรียบเทียบกับงานวิจัยอื่น

ตัวอักษรที่นำมาพิจารณา	เปอร์เซ็นต์ความถูกต้องของงานวิจัยนี้	เปอร์เซ็นต์ความถูกต้องของงานวิจัยที่นำมาเปรียบเทียบ
ก	100	91.67
ข	93.33	81.67
ฃ	90	76.67
ค	100	78.34
ฅ	96.67	76.67
ฆ	83.33	78.33
ง	90	100
จ	100	81.67
ฉ	86.67	98.34
ช	90	78.33
ซ	86.67	75.00
ฌ	100	93.34
ญ	100	96.67
ฎ	96.67	80
ฏ	90	96.67
ฐ	93.33	96.67
ฑ	90	100
ฒ	100	88.34
ณ	100	96.67
ด	86.67	83.34
ต	80	90.00
ถ	90	91.67
ท	100	98.34
ธ	100	83.33

ตารางที่ 4.2 (ต่อ)

ตัวอักษรที่นำมาพิจารณา	เปอร์เซ็นต์ความถูกต้องของงานวิจัยนี้	เปอร์เซ็นต์ความถูกต้องของงานวิจัยที่นำมาเปรียบเทียบ
น	100	93.34
บ	86.67	98.34
ป	93.33	100
ผ	90	96.67
ฝ	86.67	100
พ	96.67	98.34
ฟ	90	95.00
ภ	93.33	98.34
ม	90	98.34
ย	100	93.34
ร	100	86.67
ล	100	100
ว	100	100
ศ	90	78.33
ษ	80	86.67
ส	80	91.67
ห	70	95.00
ฬ	73.33	86.67
อ	100	86.67
ฮ	90	98.34
รวม	**92.12**	**90.53**

4.2 วิเคราะห์ผลการทดลอง

ในการทดลองการรู้จำตัวอักษรคัดลายมือเขียนภาษาไทยของวิทยานิพนธ์นี้ พบว่าเปอร์เซ็นต์ความถูกต้องของการรู้จำตัวอักษรที่มีลักษณะโครงสร้างแตกต่างกันอย่างเด่นชัดมีค่อนข้างสูง เนื่องจากสามารถแยกแยะลักษณะเด่นของตัวอักษรแต่ละตัวออกมาได้อย่างชัดเจน เช่น ก , ญ , ธ , ย แต่ตัวอักษรที่มีลักษณะโครงสร้างบางอย่างใกล้เคียงกันมากๆ เช่น ด , ต , บ , ป , ผ , ฝ ยังมีความผิดพลาดอยู่ อาจเนื่องมาจากคุณสมบัติที่กำหนดยังไม่สามารถแยกลักษณะเด่นออกมาได้ดี

พอ หรือเกิดจากการคัดลายมือที่ไม่ชัดเจนพอ ทำให้เมื่อเข้าสู่กระบวนการต่างๆแล้วข้อมูลของตัวอักษรบางอย่างขาดหายไป เช่น รอยยัก หัว หรือหางของตัวอักษร

4.3 เปรียบเทียบการทำงานของการรู้จำตัวอักษรคัดลายมือเขียนภาษาไทยระบบนี้กับระบบอื่นๆ

จากผลการทดลองเมื่อนำมาเปรียบเทียบกับงานวิจัยในระบบอื่นๆได้ผลดังนี้

ตารางที่ 4.3 แสดงผลการเปรียบเทียบการรู้จำของงานวิจัยนี้กับระบบการรู้จำของ สุรพันธ์ เอื้อไพบูลย์ กับ สกุล คำนวนชัย

ข้อเปรียบเทียบ	ระบบงานวิจัยนี้	ระบบงานวิจัย [1]	ระบบงานวิจัย [2]
1. หลักการทั่วไป	นำข้อมูลบิทแมทของตัวอักษรแต่ละตัว ที่มีรูปร่างแตกต่างกันมาหาลักษณะเด่นเพื่อเป็นตัวแทน เข้าสู่ขั้นตอนการเรียนรู้	นำเอาลักษณะเด่น คือ หัวของตัวอักษร มาใช้ในการจำแนกกลุ่มของตัวอักษร และใช้คุณสมบัติทางโทโพโลยีของตัวอักษรในการแยกตัวอักษรออกจากกลุ่ม	นำข้อมูลบิทแมทของตัวอักษรแต่ละตัว ที่มีรูปร่างแตกต่างกันมาหาตัวแทนเข้าสู่ขั้นตอนการเรียนรู้
2. ข้อมูลที่ใช้ในการทดลอง	ตัวอักษรลายมือเขียน	ตัวอักษรลายมือเขียน	ตัวอักษรลายมือเขียน
3. ขนาดตัวอักษร	100 x 100 จุดภาพ	64 x 64 จุดภาพ	32 x 16 จุดภาพ
4. โครงสร้างของโครงข่ายประสาทเทียม	มี 3 ชั้น ชั้นอินพุท 52 โหนด ชั้นซ่อน 36 โหนด ชั้นเอาท์พุท 44 โหนด	-	มี 3 ชั้น ชั้นอินพุท 512 โหนด ชั้นซ่อน 100 โหนด ชั้นเอาท์พุท 44 โหนด
5. ลักษณะของตัวอักษรที่ใช้	30 แบบ ลายมือเขียน	15 แบบลายมือเขียน	30 แบบลายมือเขียน
6. จำนวนตัวอักษรที่ใช้ในการทดสอบ	1,320 ตัว	960 ตัว	2,640 ตัว
7. ความถูกต้องในการรู้จำเฉลี่ย	92.12%	98.82%	90.53%

บทที่ 5
สรุปผลและข้อเสนอแนะ

5.1 สรุปผลการวิจัย

จากการทดลองใช้โครงข่ายประสาทเทียมแบบแพร่ย้อนกลับ ซึ่งมีทั้งหมด 3 ชั้น คือ ชั้นอินพุท ชั้นซ่อน และ ชั้นเอาท์พุท กำหนดจำนวนโหนดเป็น 52 36 และ 44 โหนด ตามลำดับ เมื่อทดสอบความถูกต้องของการรู้จำตัวอักษรคัดลายมือเขียนภาษาไทยตัวบรรจงทั้งหมด 1,320 ตัว พบว่า มีความถูกต้องในการรู้จำเฉลี่ยเท่ากับ 92.12 %

จากตาราง 4.1 จะเห็นได้ว่า ตัวอักษรบางตัวมีความถูกต้องในการรู้จำมาก เช่น ก , ฒ , ณ , ธ และก็มีตัวอักษรบางตัวที่มีความผิดพลาดสูงเช่นกัน เช่น ด , ต , ห , ฬ เป็นผลเนื่องมาจากลักษณะเด่นของตัวอักษรแต่ละตัวมีความแตกต่างกัน ตัวอักษรบางตัวมีลักษณะเด่นที่ชัดเจนก็จะมีความถูกต้องในการรู้จำมาก ส่วนตัวอักษรที่มีลักษณะเด่นใกล้เคียงกันมากก็จะมีความถูกต้องในการรู้จำน้อย

5.2 ข้อจำกัดและปัญหา

1. จากการทดลองเปรียบเทียบข้อมูลตัวอักษรที่ใช้ในการเตรียมข้อมูลและฝึกสอนระหว่างผู้คัดลายมือที่คัดตัวบรรจง กับ แบบที่คัดไม่บรรจง ทำให้ทราบว่าในระบบงานวิจัยนี้สามารถใช้ได้ดีกับตัวอักษรจากผู้ที่คัดลายมือแบบบรรจง

2. เนื่องจากตัวอักษรที่ได้จากการคัดลงบนกระดาษพิมพ์เขียนขนาด A4 ผ่านเครื่องอ่านภาพ (scanner) และเก็บข้อมูลในรูปของ Bitmap หรือ .bmp ทำให้ขนาดของข้อมูลแต่ละตัวมีขนาดใหญ่ เราจึงต้องทำการลดขนาดของตัวอักษรทุกตัวให้มีขนาด 100 x 100 จุดภาพ บางครั้งทำให้ข้อมูลบางส่วนหายไปเป็นผลทำให้ผลการวิเคราะห์ผิดพลาด ดังนั้นในงานวิจัยนี้จึงให้ผู้คัดลายมือ คัดลงในช่องที่มีขนาดจำกัดเพื่อลดความผิดพลาดที่จะเกิดขึ้น

3. อีกปัญหาหนึ่งที่เกิดขึ้นคือ สีของหมึกที่ใช้เขียนตัวอักษร ในงานวิจัยนี้ได้ทดลองให้ผู้คัดลายมือใช้ปากกาหลายๆชนิด พบว่า ปากกาลูกลื่นมีส่วนทำให้ข้อมูลที่ได้ผิดพลาดเนื่องจากสีของหมึกไม่สม่ำเสมอ เมื่อนำมาผ่านเครื่องอ่านภาพแล้วนำไปเข้ากระบวนการเตรียมข้อมูล พบว่าทำให้ลักษณะของเส้นที่ได้ไม่ถูกต้อง บางครั้งขาดตอนหรือบางครั้งเครื่องคำนวณไม่ได้ จากการทดลองพบว่า ปากกาที่เหมาะสมคือ ปากกาหมึกซึมหรือปากกาหมึกเจล เพราะสีของหมึกสม่ำเสมอ ทำให้เส้นที่ได้ไม่ขาดตอน

4. ข้อมูลภาพของตัวอักษรที่รับเข้ามาจากเครื่องสแกนเนอร์ มักจะมีจุดภาพที่ไม่ต้องการปะปนเข้ามา หรือบางจุดภาพขาดหายไป ซึ่งอาจเกิดจากมีสัญญาณรวบกวนปะปนเข้ามากับข้อมูลภาพ หรือเกิดจากประสิทธิภาพของอุปกรณ์เอง ทำให้ข้อมูลภาพไม่สมบูรณ์เหมือนต้นฉบับ ซึ่งมีผลทำให้การรู้จำผิดพลาด

5. จากตารางที่ 4.1 พบว่า ตัวอักษรบางตัวสามารถวิเคราะห์ได้ถูกต้อง 100% แต่ก็มีอักษรบางตัวที่เกิดความผิดพลาดมาก จากการวิเคราะห์ตัวอักษรที่ให้ผลผิดพลาดมาก เกิดจากตัวอักษรนั้นได้จากผู้คัดลายมือไม่ดีพอ หรือลักษณะของตัวอักษรมีโครงสร้างที่ซับซ้อนหรือใกล้เคียงกันมาก ทำให้ข้อมูลที่ได้ของแต่ละตัวอักษรไม่สามารถเป็นตัวแทนที่ดีพอที่จะใช้ในขั้นตอนการฝึกสอนหรือรู้จำได้

5.3 ข้อเสนอแนะ

1. จากการวิจัยพบว่า ตัวอักษรบางตัวที่มีลักษณะใกล้เคียงกันมาก เมื่อทำการทดสอบแล้วมีความผิดพลาดสูง ดังนั้นวิธีการแก้ปัญหาคือ ควรหาวิธีการจำแนกลักษณะเด่นของตัวอักษรให้สามารถแยกแยะความแตกต่างของตัวอักษรที่มีความคล้ายกันมากๆ ได้

2. ในการฝึกสอนการรู้จำหากเพิ่มตัวอักษรต้นแบบให้มากขึ้น โครงข่ายประสาทเทียมแบบแพร่ย้อนกลับอาจให้ค่าความถูกต้องในการรู้จำมากขึ้น

3. การกำหนดลักษณะเด่นของตัวอักษรอาจลดจำนวนบิทของข้อมูลอินพุทลงได้ เพื่อให้การประมวลผลเร็วขึ้น

เอกสารอ้างอิง

[1] สุรพันธ์ เอื้อไพบูลย์. การจดจำลายมือเขียนภาษาไทยโดยพิจารณาหัวของตัวอักษร. วิทยานิพนธ์ปริญญามหาบัณฑิต สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง, 2531.

[2] สกุล คำนวนชัย. การรู้จำตัวอักษรคัดลายมือเขียนภาษาไทยโดยประยุกต์ใช้การทรานสฟอร์แบบคาร์ฮูเน็นเลฟเข้ากับโครงข่ายฟัซซีนิวรอล. วิทยานิพนธ์ปริญญามหาบัณฑิต สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง, 2542.

[3] สนธยา เมรินทร์. การศึกษาการรู้จำตัวอักษรพิมพ์ภาษาไทยโดยใช้วิธีซินเทกติก. วิทยานิพนธ์ปริญญาวิศวกรรมศาสตรมหาบัณฑิต จุฬาลงกรณ์มหาวิทยาลัย, 2537.

[4] เดชา รัตนธาน. การรู้จำตัวอักษรพิมพ์ภาษาไทยโดยใช้เทคนิคแบบฟัซซีโลจิกและวิธีซินแทกติก. วิทยานิพนธ์ปริญญาวิศวกรรมศาสตรมหาบัณฑิต จุฬาลงกรณ์มหาวิทยาลัย, 2538.

[5] วิชิน มุ่งสิน. การรู้จำตัวอักษรพิมพ์ภาษาลาวโดยใช้ฟัซซีและนิวรอลแบบแพร่ย้อนกลับ. วิทยานิพนธ์ปริญญามหาบัณฑิต สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง, 2543.

[6] สุขสถิต สุขใจ. การใช้โครงข่ายประสาทเทียมเพื่อจดจำรูปแบบลายมือเขียนตัวเลขอารบิคแสดงข้อมูลเป็นมุมสัมบูรณ์. วิทยานิพนธ์ปริญญามหาบัณฑิต สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง, 2540.

[7] Rafael C. Gonzalez and Paul Wintz. Digital Image Processing. Addison – Wesley, 1987.

[8] Sergios Theodoridis and Konstantions Koutroumbas. Pattern Recognition. New York : Academic Press, 1999.

[9] Morton Nadler and Eric P. Smith. Pattern Recognition Engineering. New York : John Wiley & Sons, 1993.

[10] Lefteri H. Tsoukalas and Robert E. Uhrit. Fuzzy and Neural Approaches in Engineering. John Wiley & Sons, 1997.

[11] Laurene Fauserr. Foundamentals of Neural Networks Architectures, Algorithm, and Application. New Jersey : Prentice – Hall, 1994.

ภาคผนวก ก

Safe Point Thinning Algorithm

ที่มา : สุรพันธุ์ เอื้อไพบูลย์. *การจดจำตัวอักษรลายมือเขียนภาษาไทยโดยการพิจารณาหัวของตัวอักษร.* วิทยานิพนธ์สำหรับปริญญาวิศวกรรมศาสตรมหาบัณฑิต. สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง, 2531.

ในงานวิจัยนี้ได้นำวิธีการทำตัวอักษรให้บางแบบ Safe point Thinning Algorithm (SPTA) มาใช้ คือในวิธีการ SPTA ได้แบ่งจุดภาพ P ในหน้าต่าง 3 x 3 ออกเป็น 3 ชนิด

1. จุดขอบ (Edge – point) คือ จุดภาพดำที่จุดรอบข้างเป็นจุดภาพขาวใน 4 จุดรอบข้างหลักอย่างน้อย 1 จุด จากรูปที่ 2.4 จุดภาพหลักคือ X0 , X2 , X4 และ X6 ซึ่งแบ่งออกเป็น 4 ลักษณะ คือ

1.1 จุดขอบด้านซ้าย (left edge – point) คือ จุดภาพ P ที่มี X4 เป็นจุดภาพขาว

1.2 จุดขอบด้านขวา (right edge – point) คือ จุดภาพ P ที่มี X0 เป็นจุดภาพขาว

1.3 จุดขอบด้านบน (top edge – point) คือ จุดภาพ P ที่มี X2 เป็นจุดภาพขาว

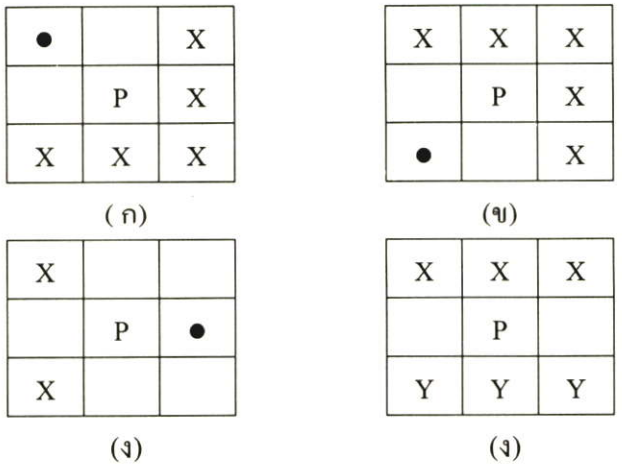
1.4 จุดขอบด้านล่าง (bottom edge – point) คือ จุดภาพ P ที่มี X6 เป็นจุดภาพขาว

2. จุดปลาย (End – point) คือ จุดภาพดำที่มีจุดรอบข้างเป็นจุดภาพดำอย่างมาก 1 จุดใน 8 จุดรอบข้าง

3. จุดหยุด (Break – point) คือ จุดภาพดำที่ไม่ควรลบเนื่องจากเป็นจุดต่อเชื่อมของภาพ

ในกระบวนการทำให้บางถือว่า ขอบของภาพไม่ใช่โครงร่างของตัวอักษร ดังนั้นเมื่อพบขอบก็จะกำหนดค่าของจุดภาพนั้นให้เป็นค่าค่าหนึ่งที่ไม่ใช่ 1 (flagging) แต่ถ้าพบจุดภาพที่มีความกว้างเพียง 1 จุดภาพก็จะกำหนดค่าของจุดภาพนั้นให้เป็น 1 (Unflagging)

การทดสอบจุดภาพที่เป็นขอบ กระทำโดยการเปรียบเทียบตารางหน้าต่างขนาด 3 x 3 ที่สแกนจากบนลงล่างบนรูปแบบของตัวอักษรกับหน้าต่างในรูป ก1



รูป ก1 แสดงหน้าต่างเงื่อนไขใน SPTA

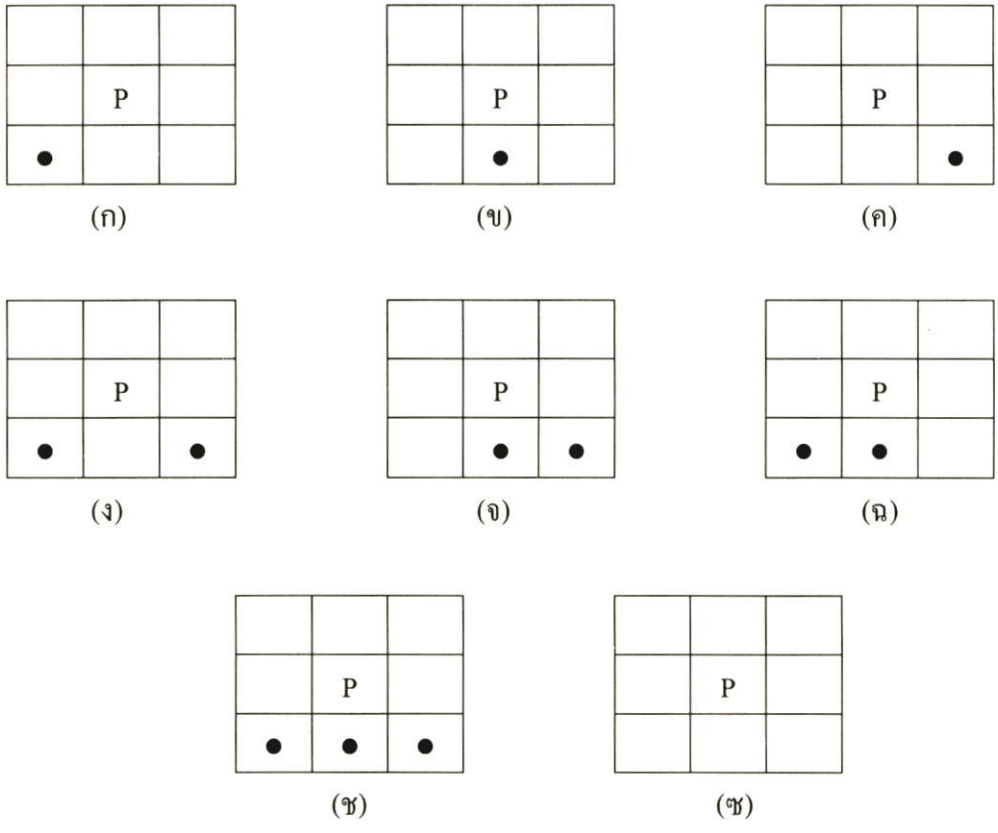
จากรูป ก1. P และ ● เป็นจุดดำ (มีค่าเป็น 1) ส่วน X และ Y เป็นจุดภาพดำหรือจุดภาพขาว เมื่อพิจารณารูปที่ 1(ก) , (ข) และ (ค) จะแบ่งการพิจารณาเป็น 2 ลักษณะคือ

1. ถ้า X ทุกตัวเป็นจุดขาว (มีค่าเป็น 0) จะได้ว่า P คือจุดภาพที่เป็นจุดปลาย (end – point) ของโครงร่าง

2. ถ้า X อย่างน้อย 1 จุดเป็นจุดดำ (มีค่าเป็น 1) จะได้ว่า P คือจุดภาพที่เป็นโครงร่างของตัวอักษร

จุด P ใดๆที่มีคุณสมบัติตาม 2 ลักษณะดังกล่าวจะถูกกำหนดค่าให้เป็น 1 (Unflagging)

หน้าต่างในรูปที่ 1(ง) จะมีการพิจารณาที่ซับซ้อนขึ้นโดยแยกการพิจารณาออกมาดังแสดงในรูป ก2.



รูป ก2. แสดงหน้าต่างของจุดภาพโครงร่าง

จากรูป ก1(ง) ถ้า X และ Y อย่างน้อย 1 จุดเป็นจุดดำจะกำหนดให้จุด P เป็นจุดโครงร่าง ในกรณีนอกเหนือจากนี้จะพิจารณาตามหน้าต่างในรูป ก2

จากรูป ก2 หน้าต่าง (ก) , (ข) และ (ค) จะกำหนดให้จุด P เป็นจุดปลายๆของโครงร่าง หน้าต่าง (ง) กำหนดให้ P เป็นจุดโครงร่าง หน้าต่าง (จ) และ (ฉ) กำหนดให้ P เป็นจุดที่ทำให้เกิด

ความกว้างขนาด 2 จุดภาพในแนวเอียง หน้าต่าง (ช) กำหนดให้จุด P เป็นกิ่งสั้นของโครงร่าง ส่วนหน้าต่าง (ซ) กำหนดให้จุด P เป็นจุดอิสระที่เกิดขึ้น

เพื่อลดความยุ่งยากในการพิจารณาเงื่อนไขของขอบและโครงร่างลงจึงได้นำเอาเงื่อนไขทั้งหมดของขอบมาเขียนเป็นเงื่อนไขทาง boolean ทำให้ง่ายต่อการเขียนโปรแกรมยิ่งขึ้น ดังแสดงใน

Left Safe – Point

S4 = n0 * (n1+n2+n6+n7) * (n2+n3) * (n6+n5)

Right Safe – Point

S0 = n4 * (n5+n6+n2+n3) * (n6+n7) * (n2+n1)

Top Safe – Point

S2 = n6 * (n7+n0+n4+n5) * (n0+n1) * (n4+n3)

Bottom Safe – Point

S6 = n2 * (n3+n4+n0+n1) * (n4+n5) * (n0+n7)

โดยถ้าค่าของ ni มีความหมายของค่าตรรกยะ

"จริง" แสดงถึงจุดนั้นเป็นจุดภาพดำและไม่ถูกแฟลก

"เท็จ" แสดงถึงจุดนั้นเป็นจุดภาพขาวและถูกแฟลก

เพื่อนำไปหาค่า Si ต่อไป

ขั้นตอนในการทำงานของ Safe – Point Thinning นั้น ได้แสดงเป็น Algorithm ได้ดังนี้ซึ่งภาพที่เข้าจะถูกสแกนทั้งในแนวแถวและในแนวคอลัมน์

Algorithm : Safe Point Thinning

Define : Type of the pattern , where HEIGHT and WIDTH are its dimensions. I represents the iteration number during the step 2 – 9. J indicates the "type" of scanning ; j = 0 detection of right and left Safe – Point , j = 2 detection of top and bottom Safe – Point

1. Set I = 0
2. Repeat steps 3 – 9
3. i = i + 1
4. For all j's do step 5 – 9

   Begin

5. For all ROWS and all COLUMNS do step 6 – 9 {scan PATTERN with type j}

Begin

6. p = PATTERN[ROW , COLUMN] {get a point p witch p is point to be examined}

7. If DARK(p) then {if p is dark and unflagged}

8. If EDGE – Point (p , n_j , n_{j+4}) then {if p is an edge – point , i.e, it has , as white point}

9. If not Safe – Point (p , n_j , n_{j+4}) then {if p does not satisfy either boolean expression or }

Set p equal to i – MAXINT else {p is a Safe – Point}

If p is not yet labeled as Safe – Point then set p equal to i

End

End

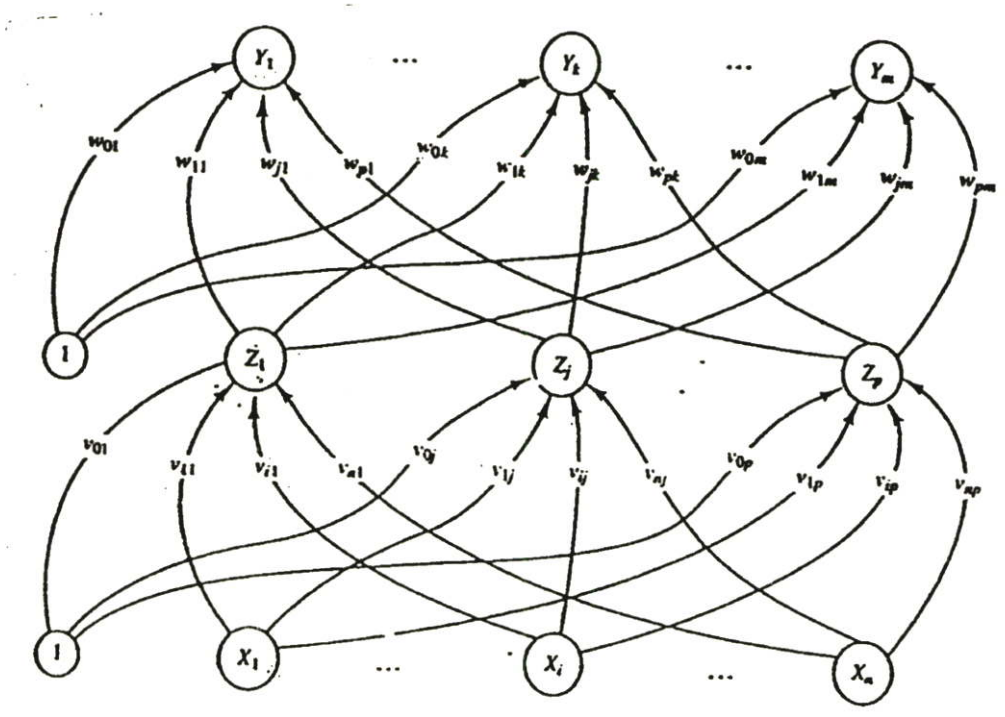
Until NO – MORE is true {NO – MORE returns true there are no more point to be flagged in the pattern}

10. End of Algorithm

ภาคผนวก ข

ที่มาของกฎการเรียนรู้

ที่มา : Laurene Fauserr. *Foundamentals of Neural Networks Architectures, Algorithm, and Application.* New Jersey : Prentice – Hall, 1994.



จากรูปข้างบน

$$y_in_k = \sum_j z_j w_{jk}$$

ค่าผิดพลาดที่น้อยที่สุด หาได้จาก

$$E = \frac{1}{2}\sum_k [t_k - y_k]^2$$

จากกฎลูกโซ่ จะได้ว่า

$$\frac{\partial E}{\partial w_{JK}} = \frac{\partial}{\partial w_{JK}} \frac{1}{2}\sum_k [t_k - y_k]^2$$

$$= -[t_K - y_K] \frac{\partial}{\partial w_{JK}} (y_k)$$

$$= -[t_K - y_K] \frac{\partial}{\partial w_{JK}} [f(y_in_K)]$$

$$= -[t_K - y_K] f'(y_in_K) \frac{\partial}{\partial w_{JK}} (y_in_K)$$

$$= -[t_K - y_K]f'(y_in_K)z_J$$

เพื่อความสะดวกกำหนด δ_K โดย

$$\delta_K = [t_K - y_K]f'(y_in_K)$$

สำหรับน้ำหนักที่เชื่อมระหว่างชั้นซ่อน Z_J หาได้จาก

$$\begin{aligned}
\frac{\partial E}{\partial v_{IJ}} &= \frac{\partial}{\partial v_{IJ}}\frac{1}{2}\sum_K [t_k - y_k]^2 \\
&= -\sum_K [t_k - y_k]\frac{\partial}{\partial v_{IJ}} y_k \\
&= -\sum_K [t_k - y_k]\frac{\partial}{\partial v_{IJ}}[f(y_in_k)] \\
&= -\sum_K [t_k - y_k]f'(y_in_k)\frac{\partial}{\partial v_{IJ}}(y_in_k) \\
&= -\sum_k \delta_k \frac{\partial}{\partial v_{IJ}}(y_in_k) \\
&= -\sum_k \delta_k \frac{\partial}{\partial v_{IJ}}(z_J w_{Jk}) \\
&= -\sum_k \delta_k w_{Jk}\frac{\partial}{\partial v_{IJ}}(z_J) \\
&= -\sum_k \delta_k w_{Jk}\frac{\partial}{\partial v_{IJ}}[f(z_in_J)] \\
&= -\sum_k \delta_k w_{Jk} f'(z_in_j)\frac{\partial}{\partial v_{IJ}}(z_in_j) \\
&= -\sum_k \delta_k w_{Jk} f'(z_in_j)\frac{\partial}{\partial v_{IJ}}(x_J w_{IJ}) \\
&= -\sum_k \delta_k w_{Jk} f'(z_in_j)x_I
\end{aligned}$$

เพื่อความสะดวก กำหนด δ_J โดย

$$\delta_J = \sum_k \delta_k w_{jk} f'(z_in_j)$$

ดังนั้นการปรับปรุงน้ำหนักของนิวรอลเอาท์พุท (output unit) หาได้จาก

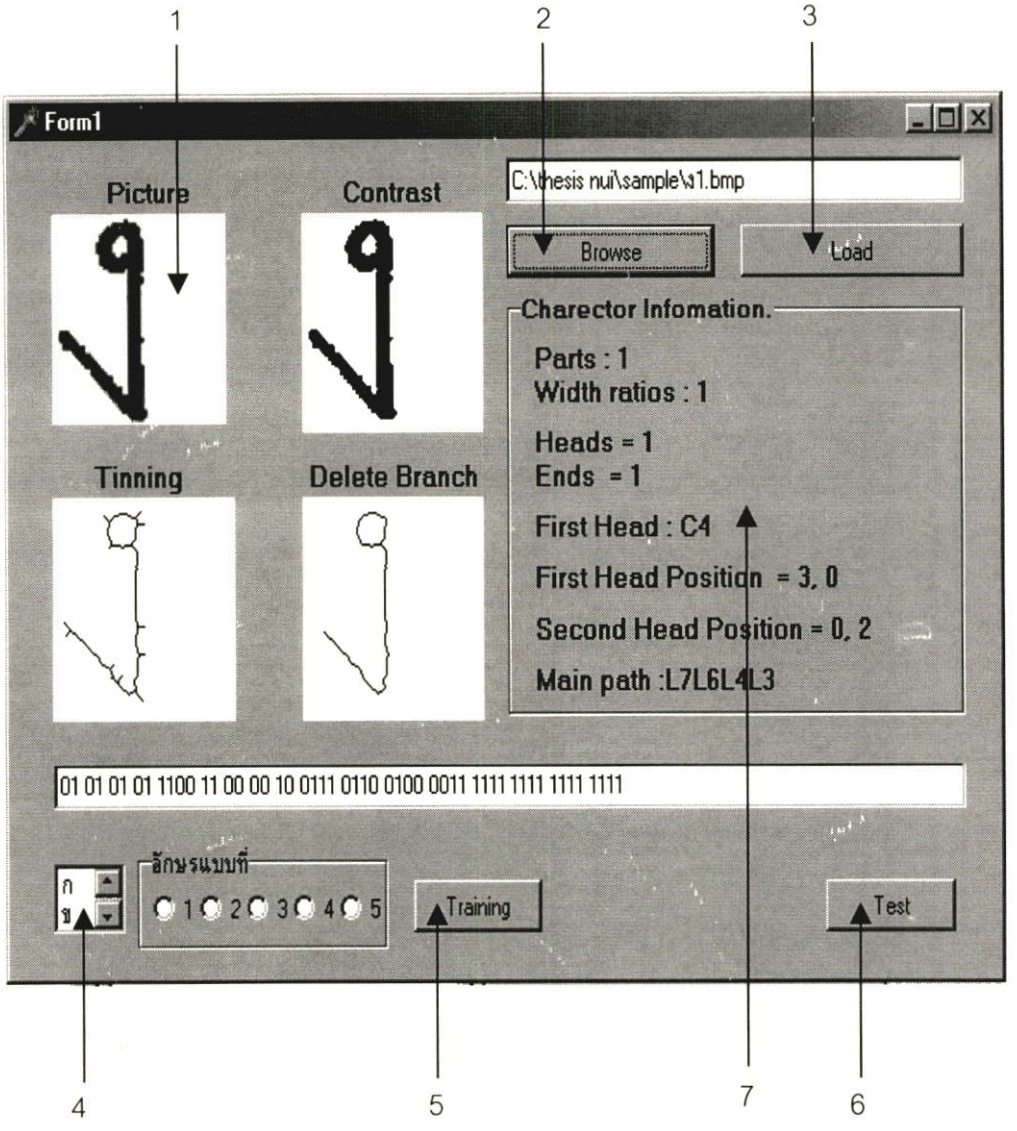
$$\begin{aligned}
\Delta w_{jk} &= -\alpha\frac{\partial E}{\partial w_{jk}} \\
&= \alpha[t_k - y_k]f'(y_in_k)z_j \\
&= \alpha\,\delta_k z_j
\end{aligned}$$

และการปรับปรุงน้ำหนักของนิวรอลในชั้นซ่อน (hidden unit) หาได้จาก

$$\begin{aligned}\Delta v_{ij} &= -\alpha \frac{\partial E}{\partial v_{ij}} \\ &= \alpha f'(z_in_j) x_i \sum_k \delta_k w_{jk} \\ &= \alpha \delta_j x_i\end{aligned}$$

ภาคผนวก ค

องค์ประกอบของโปรแกรมของวิทยานิพนธ์



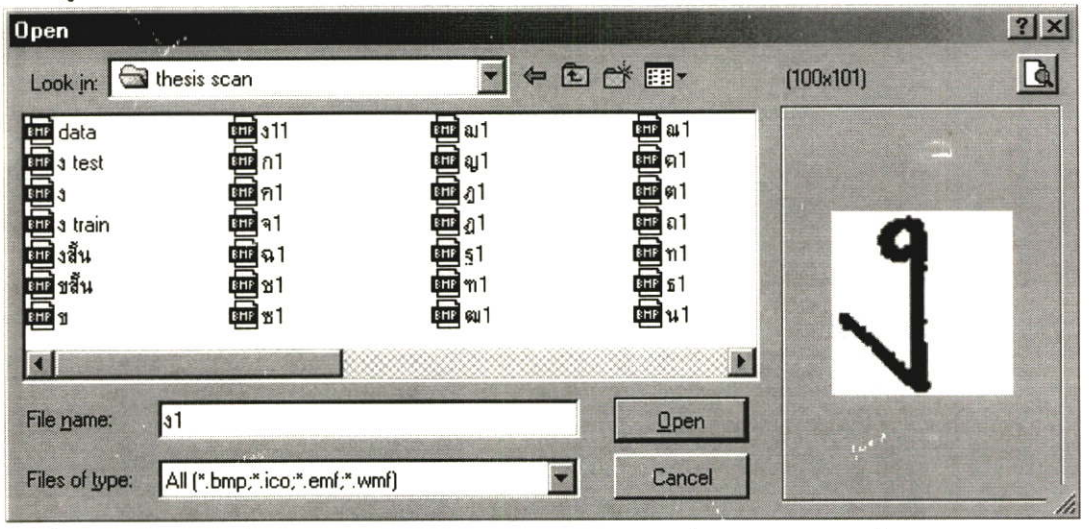
1. ช่องแสดงรูปภาพตัวอักษร
2. ปุ่มเลือกไฟล์ภาพตัวอักษร (ปุ่มนี้จะเรียก Dialog Box สำหรับเปิดไฟล์ภาพตัวอักษรขึ้นมา)
3. ปุ่มโหลดไฟล์ภาพตัวอักษร
4. ช่องเลือกตัวอักษรสำหรับการ Training (จะเลือกต่อเมื่อ ต้องการให้โปรแกรมเรียนรู้ภาพตัวอักษรต้นแบบสำหรับการรู้จำ และจะต้องทำการเลือกตัวอักษรก่อนที่จะกดปุ่ม Training ทุกครั้ง)
5. ปุ่ม Training ให้กดเมื่อต้องการให้โปรแกรมทำการเรียนรู้และจำภาพตัวอักษร
6. ปุ่มทดสอบการรู้จำของโปรแกรม
7. ส่วนแสดงลักษณะพิเศษของตัวอักษร

โปรแกรมนี้ เป็นโปรแกรมสำหรับการเรียนรู้จำตัวอักษรคัดลายมือเขียนภาษาไทย โดยโปรแกรมจะทำการรับเอาไฟล์ภาพตัวอักษรที่มีนามสกุลเป็นไฟล์ .BMP และโปรแกรมจะทำการประมวลผลภาพตัวอักษรเดียว ครั้งละหนึ่งภาพตัวอักษรเท่านั้น

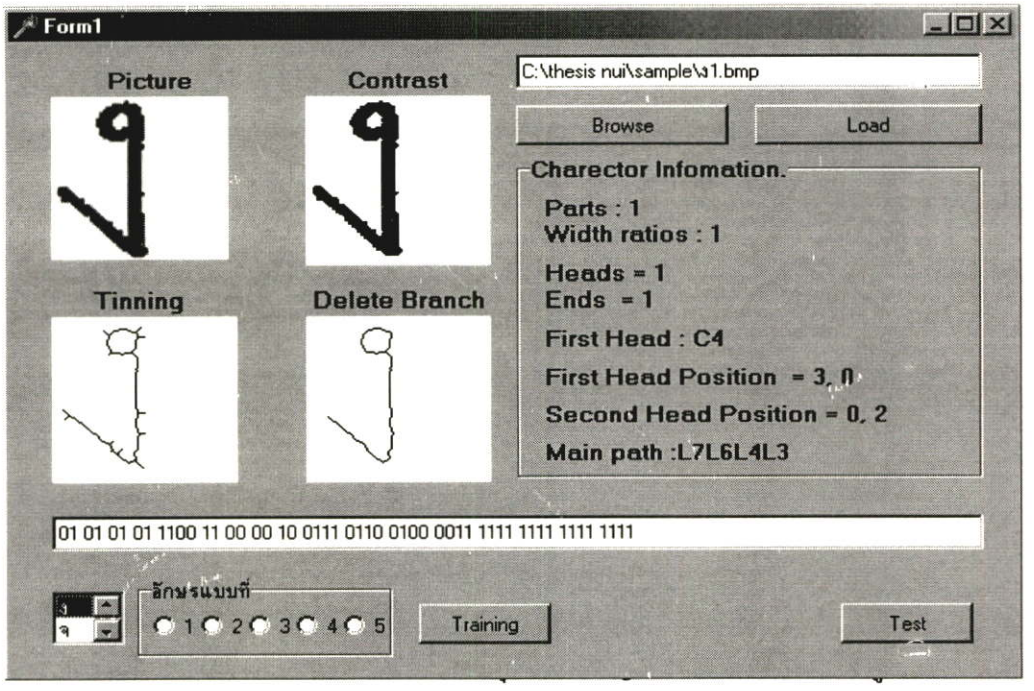
ขั้นตอนการ Training ให้โปรแกรมรู้จำตัวอักษร

1. ทำการสแกนภาพตัวอักษรต้นแบบที่ต้องการให้โปรแกรมรู้จำ โดยต้องทำการเก็บไฟล์ภาพอักษรที่มีนามสกุลเป็น .BMP เท่านั้น (หนึ่งภาพตัวอักษรต่อหนึ่งไฟล์)

2. กดปุ่ม 2 เพื่อทำการเลือกโหลดไฟล์ภาพตัวอักษรที่ต้องการเมื่อกดปุ่มแล้วจะปรากฏ Dialog box ดังรูปข้างล่าง ทำการเลือกไฟล์ที่ต้องการ



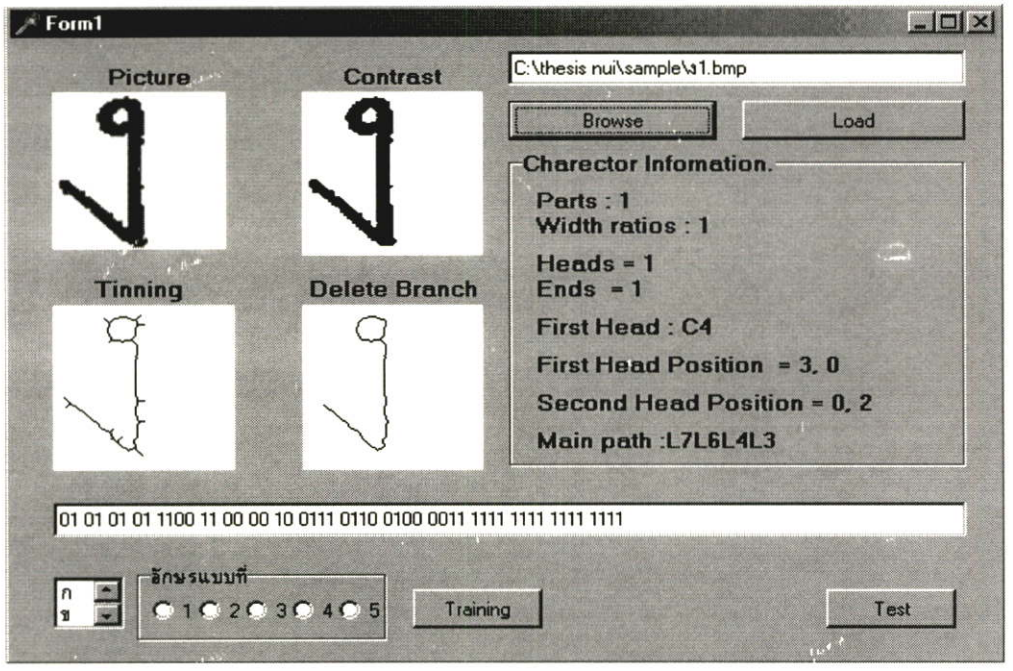
3. เมื่อเลือกไฟล์ภาพที่ต้องการแล้ว ให้ทำการเลือกตัวอักษรที่ต้องการให้จำ ที่ช่องเลือกตัวอักษร (ช่องที่ 4) ดังรูป



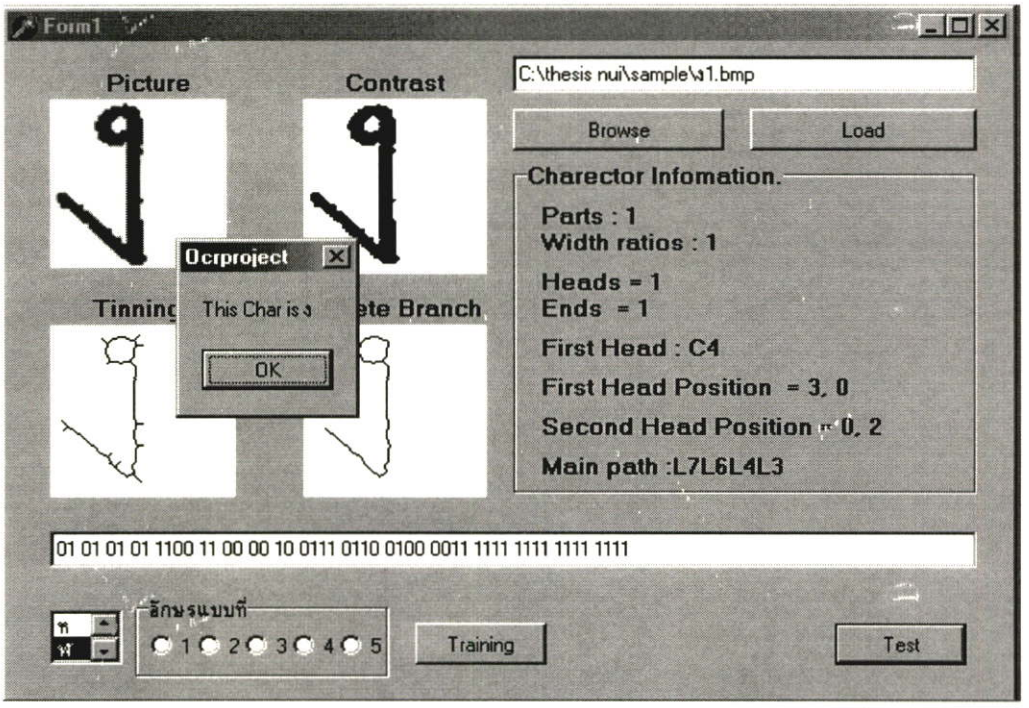
ขั้นตอนการทดสอบการรู้จำตัวอักษร

1. ทำการสแกนรูปภาพตัวอักษรที่ต้องการทดสอบการรู้จำ โดยทำการเก็บไฟล์ภาพ เป็นนามสกุล .BMP

2. กดปุ่ม Browse (ปุ่มที่ 2) แล้วเลือกไฟล์ภาพที่ทำการเก็บไว้ตามขั้นตอนที่ 1 ดังรูป



3. เมื่อเลือกไฟล์ภาพเรียบร้อยแล้วให้กดปุ่ม Test เพื่อเริ่มการรู้จำตัวอักษร ผลที่ได้แสดงดังภาพ ซึ่งจะมี Massage Dialog box แจ้งการรู้จำว่าเป็นตัวอักษรใดดังรูป



ภาคผนวก ง

ตัวอักษรคัดลายมือเขียนภาษาไทย

ตัวอักษรที่ใช้ในงานวิจัยนี้เป็นตัวอักษรที่ให้ผู้คัดเขียนตัวอักษรลงในช่องที่กำหนดคนละหนึ่งช่อง คือคนที่ 1 เขียนลงในช่องที่ 1 คนที่ 2 เขียนลงในช่องที่ 2 จนถึงคนที่ 30 เขียนลงในช่องที่ 30 ตามลำดับ โดยตัวอักษรของผู้คัดคนที่ 1 – 5 ใช้เป็นตัวอักษรต้นแบบในการฝึกสอน ส่วนการทดสอบการรู้จำจะใช้ตัวอักษรทั้งหมดในการทดสอบ

คนที่	1	2	3	4	5	6	7	8	9	10
ก	ก	ก	ก	ก	ก	ก	ก	ก	ก	ก
ข	ข	ข	ข	ข	ข	ข	ข	ข	ข	ข
ฃ	ฃ	ฃ	ฃ	ฃ	ฃ	ฃ	ฃ	ฃ	ฃ	ฃ
ค	ค	ค	ค	ค	ค	ค	ค	ค	ค	ค
ฅ	ฅ	ฅ	ฅ	ฅ	ฅ	ฅ	ฅ	ฅ	ฅ	ฅ
ฆ	ฆ	ฆ	ฆ	ฆ	ฆ	ฆ	ฆ	ฆ	ฆ	ฆ
ง	ง	ง	ง	ง	ง	ง	ง	ง	ง	ง
จ	จ	จ	จ	จ	จ	จ	จ	จ	จ	จ
ฉ	ฉ	ฉ	ฉ	ฉ	ฉ	ฉ	ฉ	ฉ	ฉ	ฉ
ช	ช	ช	ช	ช	ช	ช	ช	ช	ช	ช
ซ	ซ	ซ	ซ	ซ	ซ	ซ	ซ	ซ	ซ	ซ
ฌ	ฌ	ฌ	ฌ	ฌ	ฌ	ฌ	ฌ	ฌ	ฌ	ฌ
ญ	ญ	ญ	ญ	ญ	ญ	ญ	ญ	ญ	ญ	ญ
ฎ	ฎ	ฎ	ฎ	ฎ	ฎ	ฎ	ฎ	ฎ	ฎ	ฎ
ฏ	ฏ	ฏ	ฏ	ฏ	ฏ	ฏ	ฏ	ฏ	ฏ	ฏ
ฐ	ฐ	ฐ	ฐ	ฐ	ฐ	ฐ	ฐ	ฐ	ฐ	ฐ
ฑ	ฑ	ฑ	ฑ	ฑ	ฑ	ฑ	ฑ	ฑ	ฑ	ฑ
ฒ	ฒ	ฒ	ฒ	ฒ	ฒ	ฒ	ฒ	ฒ	ฒ	ฒ
ณ	ณ	ณ	ณ	ณ	ณ	ณ	ณ	ณ	ณ	ณ
ด	ด	ด	ด	ด	ด	ด	ด	ด	ด	ด
ต	ต	ต	ต	ต	ต	ต	ต	ต	ต	ต
ถ	ถ	ถ	ถ	ถ	ถ	ถ	ถ	ถ	ถ	ถ
ท	ท	ท	ท	ท	ท	ท	ท	ท	ท	ท

คนที่	1	2	3	4	5	6	7	8	9	10
ธ	ธ	ธ	ธ	ธ	ธ	ธ	ธ	ธ	ธ	ธ
น	น	น	น	น	น	น	น	น	น	น
บ	บ	บ	บ	บ	บ	บ	บ	บ	บ	บ
ป	ป	ป	ป	ป	ป	ป	ป	ป	ป	ป
ผ	ผ	ผ	ผ	ผ	ผ	ผ	ผ	ผ	ผ	ผ
ฝ	ฝ	ฝ	ฝ	ฝ	ฝ	ฝ	ฝ	ฝ	ฝ	ฝ
พ	พ	พ	พ	พ	พ	พ	พ	พ	พ	พ
ฟ	ฟ	ฟ	ฟ	ฟ	ฟ	ฟ	ฟ	ฟ	ฟ	ฟ
ภ	ภ	ภ	ภ	ภ	ภ	ภ	ภ	ภ	ภ	ภ
ม	ม	ม	ม	ม	ม	ม	ม	ม	ม	ม
ย	ย	ย	ย	ย	ย	ย	ย	ย	ย	ย
ร	ร	ร	ร	ร	ร	ร	ร	ร	ร	ร
ล	ล	ล	ล	ล	ล	ล	ล	ล	ล	ล
ว	ว	ว	ว	ว	ว	ว	ว	ว	ว	ว
ศ	ศ	ศ	ศ	ศ	ศ	ศ	ศ	ศ	ศ	ศ
ษ	ษ	ษ	ษ	ษ	ษ	ษ	ษ	ษ	ษ	ษ
ส	ส	ส	ส	ส	ส	ส	ส	ส	ส	ส
ห	ห	ห	ห	ห	ห	ห	ห	ห	ห	ห
ฬ	ฬ	ฬ	ฬ	ฬ	ฬ	ฬ	ฬ	ฬ	ฬ	ฬ
อ	อ	อ	อ	อ	อ	อ	อ	อ	อ	อ
ฮ	ฮ	ฮ	ฮ	ฮ	ฮ	ฮ	ฮ	ฮ	ฮ	ฮ

คนที่	11	12	13	14	15	16	17	18	19	20
ก	ก	ก	ก	ก	ก	ก	ก	ก	ก	ก
ข	ข	ข	ข	ข	ข	ข	ข	ข	ข	ข
ฃ	ฃ	ฃ	ฃ	ฃ	ฃ	ฃ	ฃ	ฃ	ฃ	ฃ
ค	ค	ค	ค	ค	ค	ค	ค	ค	ค	ค
ฅ	ฅ	ฅ	ฅ	ฅ	ฅ	ฅ	ฅ	ฅ	ฅ	ฅ
ฆ	ฆ	ฆ	ฆ	ฆ	ฆ	ฆ	ฆ	ฆ	ฆ	ฆ
ง	ง	ง	ง	ง	ง	ง	ง	ง	ง	ง
จ	จ	จ	จ	จ	จ	จ	จ	จ	จ	จ
ฉ	ฉ	ฉ	ฉ	ฉ	ฉ	ฉ	ฉ	ฉ	ฉ	ฉ
ช	ช	ช	ช	ช	ช	ช	ช	ช	ช	ช
ซ	ซ	ซ	ซ	ซ	ซ	ซ	ซ	ซ	ซ	ซ
ฌ	ฌ	ฌ	ฌ	ฌ	ฌ	ฌ	ฌ	ฌ	ฌ	ฌ
ญ	ญ	ญ	ญ	ญ	ญ	ญ	ญ	ญ	ญ	ญ
ฎ	ฎ	ฎ	ฎ	ฎ	ฎ	ฎ	ฎ	ฎ	ฎ	ฎ
ฏ	ฏ	ฏ	ฏ	ฏ	ฏ	ฏ	ฏ	ฏ	ฏ	ฏ
ฐ	ฐ	ฐ	ฐ	ฐ	ฐ	ฐ	ฐ	ฐ	ฐ	ฐ
ฑ	ฑ	ฑ	ฑ	ฑ	ฑ	ฑ	ฑ	ฑ	ฑ	ฑ
ฒ	ฒ	ฒ	ฒ	ฒ	ฒ	ฒ	ฒ	ฒ	ฒ	ฒ
ณ	ณ	ณ	ณ	ณ	ณ	ณ	ณ	ณ	ณ	ณ
ด	ด	ด	ด	ด	ด	ด	ด	ด	ด	ด
ต	ต	ต	ต	ต	ต	ต	ต	ต	ต	ต
ถ	ถ	ถ	ถ	ถ	ถ	ถ	ถ	ถ	ถ	ถ
ท	ท	ท	ท	ท	ท	ท	ท	ท	ท	ท

คนที่	11	12	13	14	15	16	17	18	19	20
ธ	ธ	ธ	ธ	ธ	ธ	ธ	ธ	ธ	ธ	ธ
น	น	น	น	น	น	น	น	น	น	น
บ	บ	บ	บ	บ	บ	บ	บ	บ	บ	บ
ป	ป	ป	ป	ป	ป	ป	ป	ป	ป	ป
ผ	ผ	ผ	ผ	ผ	ผ	ผ	ผ	ผ	ผ	ผ
ฝ	ฝ	ฝ	ฝ	ฝ	ฝ	ฝ	ฝ	ฝ	ฝ	ฝ
พ	พ	พ	พ	พ	พ	พ	พ	พ	พ	พ
ฟ	ฟ	ฟ	ฟ	ฟ	ฟ	ฟ	ฟ	ฟ	ฟ	ฟ
ภ	ภ	ภ	ภ	ภ	ภ	ภ	ภ	ภ	ภ	ภ
ม	ม	ม	ม	ม	ม	ม	ม	ม	ม	ม
ย	ย	ย	ย	ย	ย	ย	ย	ย	ย	ย
ร	ร	ร	ร	ร	ร	ร	ร	ร	ร	ร
ล	ล	ล	ล	ล	ล	ล	ล	ล	ล	ล
ว	ว	ว	ว	ว	ว	ว	ว	ว	ว	ว
ศ	ศ	ศ	ศ	ศ	ศ	ศ	ศ	ศ	ศ	ศ
ษ	ษ	ษ	ษ	ษ	ษ	ษ	ษ	ษ	ษ	ษ
ส	ส	ส	ส	ส	ส	ส	ส	ส	ส	ส
ห	ห	ห	ห	ห	ห	ห	ห	ห	ห	ห
ฬ	ฬ	ฬ	ฬ	ฬ	ฬ	ฬ	ฬ	ฬ	ฬ	ฬ
อ	อ	อ	อ	อ	อ	อ	อ	อ	อ	อ
ฮ	ฮ	ฮ	ฮ	ฮ	ฮ	ฮ	ฮ	ฮ	ฮ	ฮ

คนที่	21	22	23	24	25	26	27	28	29	30
ก	ก	ก	ก	ก	ก	ก	ก	ก	ก	ก
ข	ข	ข	ข	ข	ข	ข	ข	ข	ข	ข
ฃ	ฃ	ฃ	ฃ	ฃ	ฃ	ฃ	ฃ	ฃ	ฃ	ฃ
ค	ค	ค	ค	ค	ค	ค	ค	ค	ค	ค
ฅ	ฅ	ฅ	ฅ	ฅ	ฅ	ฅ	ฅ	ฅ	ฅ	ฅ
ฆ	ฆ	ฆ	ฆ	ฆ	ฆ	ฆ	ฆ	ฆ	ฆ	ฆ
ง	ง	ง	ง	ง	ง	ง	ง	ง	ง	ง
จ	จ	จ	จ	จ	จ	จ	จ	จ	จ	จ
ฉ	ฉ	ฉ	ฉ	ฉ	ฉ	ฉ	ฉ	ฉ	ฉ	ฉ
ช	ช	ช	ช	ช	ช	ช	ช	ช	ช	ช
ซ	ซ	ซ	ซ	ซ	ซ	ซ	ซ	ซ	ซ	ซ
ฌ	ฌ	ฌ	ฌ	ฌ	ฌ	ฌ	ฌ	ฌ	ฌ	ฌ
ญ	ญ	ญ	ญ	ญ	ญ	ญ	ญ	ญ	ญ	ญ
ฎ	ฎ	ฎ	ฎ	ฎ	ฎ	ฎ	ฎ	ฎ	ฎ	ฎ
ฏ	ฏ	ฏ	ฏ	ฏ	ฏ	ฏ	ฏ	ฏ	ฏ	ฏ
ฐ	ฐ	ฐ	ฐ	ฐ	ฐ	ฐ	ฐ	ฐ	ฐ	ฐ
ฑ	ฑ	ฑ	ฑ	ฑ	ฑ	ฑ	ฑ	ฑ	ฑ	ฑ
ฒ	ฒ	ฒ	ฒ	ฒ	ฒ	ฒ	ฒ	ฒ	ฒ	ฒ
ณ	ณ	ณ	ณ	ณ	ณ	ณ	ณ	ณ	ณ	ณ
ด	ด	ด	ด	ด	ด	ด	ด	ด	ด	ด
ต	ต	ต	ต	ต	ต	ต	ต	ต	ต	ต
ถ	ถ	ถ	ถ	ถ	ถ	ถ	ถ	ถ	ถ	ถ
ท	ท	ท	ท	ท	ท	ท	ท	ท	ท	ท

คนที่	21	22	23	24	25	26	27	28	29	30
ธ	ธ	ธ	ธ	ธ	ธ	ธ	ธ	ธ	ธ	ธ
น	น	น	น	น	น	น	น	น	น	น
บ	บ	บ	บ	บ	บ	บ	บ	บ	บ	บ
ป	ป	ป	ป	ป	ป	ป	ป	ป	ป	ป
ผ	ผ	ผ	ผ	ผ	ผ	ผ	ผ	ผ	ผ	ผ
ฝ	ฝ	ฝ	ฝ	ฝ	ฝ	ฝ	ฝ	ฝ	ฝ	ฝ
พ	พ	พ	พ	พ	พ	พ	พ	พ	พ	พ
ฟ	ฟ	ฟ	ฟ	ฟ	ฟ	ฟ	ฟ	ฟ	ฟ	ฟ
ภ	ภ	ภ	ภ	ภ	ภ	ภ	ภ	ภ	ภ	ภ
ม	ม	ม	ม	ม	ม	ม	ม	ม	ม	ม
ย	ย	ย	ย	ย	ย	ย	ย	ย	ย	ย
ร	ร	ร	ร	ร	ร	ร	ร	ร	ร	ร
ล	ล	ล	ล	ล	ล	ล	ล	ล	ล	ล
ว	ว	ว	ว	ว	ว	ว	ว	ว	ว	ว
ศ	ศ	ศ	ศ	ศ	ศ	ศ	ศ	ศ	ศ	ศ
ษ	ษ	ษ	ษ	ษ	ษ	ษ	ษ	ษ	ษ	ษ
ส	ส	ส	ส	ส	ส	ส	ส	ส	ส	ส
ห	ห	ห	ห	ห	ห	ห	ห	ห	ห	ห
ฬ	ฬ	ฬ	ฬ	ฬ	ฬ	ฬ	ฬ	ฬ	ฬ	ฬ
อ	อ	อ	อ	อ	อ	อ	อ	อ	อ	อ
ฮ	ฮ	ฮ	ฮ	ฮ	ฮ	ฮ	ฮ	ฮ	ฮ	ฮ

ภาคผนวก จ

ตัวอย่างตัวอักษร ก – ฮ ที่ผ่านกระบวนการหาลักษณะเด่น

ตาราง จ1 แสดงลักษณะเด่นของตัวอักษรบางตัวที่ทำการทดสอบ

ตัวอักษร	ส่วน	อัตราความกว้าง	จำนวนหัว	จำนวนปลาย	ลักษณะหัวแรกที่พบ	ตำแหน่งหัวแรกหรือปลายแรก	ตำแหน่งหัวที่สองหรือปลายที่สอง	เวกเตอร์ทิศทาง
ก	1	2	0	2	-	03	33	L2L3L2L1L0L7L6
ข	1	2	1	1	C2	00	30	L1L0L7L6L5L6L0L2
ฃ	1	1	1	1	C2	00	30	L2L0L6L5L6L7L0L1
ค	1	2	1	1	C3	12	33	L5L3L2L1L0L7L6
ฅ	1	2	1	1	C3	12	33	L5L3L2L1L0L7L1L7
ฆ	1	2	2	1	C2	00	12	L2L1L7L7L6L5L6
ง	1	1	1	1	C4	30	02	L7L6L4L3
จ	1	2	1	1	C4	22	00	L7L1L2L3L4L5
ฉ	1	3	2	1	C4	12	32	L6L1L0
ช	1	2	2	0	C1	00	20	L2L0L7
ซ	1	2	1	1	C2	02	30	L2L1L7L1L0L6L5L6
ฌ	1	4	2	1	C2	03	22	L2L3L2L1L0L7L6
ญ	2	3	1	1	C2	03	30	L2L3L1L0L7L6L0L1
ฎ	1	2	2	0	C1	12	23	L2L1L0L7L6L5L3
ฏ	1	1	2	0	C1	11	13	L2L3L1L0L7L6L4L3
ฐ	2	3	1	1	C4	23	30	L7L1L2L4L3L1L0

ตัวอักษร	ส่วน	อัตราความกว้าง	จำนวนหัว	จำนวนปลาย	หัวแรกที่พบ	ตำแหน่งหัวแรกหรือปลายแรก	ตำแหน่งหัวที่สองหรือปลายที่สอง	เวกเตอร์ทิศทาง
ฑ	1	4	1	1	C2	00	33	L1L0L1L7L6L5L6L1
ฒ	1	4	2	1	C4	12	22	L6L5L3L2L1L0L7L1
ณ	1	4	2	1	C2	03	32	L2L3L1L0L7L6L1L0
ค	1	2	1	1	C4	22	33	L6L5L3L2L1L7L6
ต	1	2	1	1	C4	12	33	L6L5L3L2L1L0L7L1
ถ	1	2	1	1	C2	03	33	L2L3L1L0L7L6
ท	1	3	1	1	C4	01	33	L6L1L0L7L6
ธ	1	2	0	2	-	01	30	L6L0L2L4L1L0L1
น	1	3	2	1	C4	01	22	L6L1L0
บ	1	3	1	1	C4	11	30	L6L0L2
ป	1	2	1	1	C4	11	30	L6L0L2
ผ	1	2	1	1	C3	01	30	L5L6L0L1L7L1L2
ฝ	1	1	1	1	C3	02	30	L6L0L7L6L1L2
พ	1	4	1	1	C4	01	30	L6L7L1L2L0L6L7L1
ฟ	1	2	1	1	C4	02	30	L6L1L2L1L7L6L7L1
ภ	1	3	1	1	C1	13	33	L1L2L3L1L0L7L6
ม	1	2	2	1	C4	11	12	L7L5L6

ตัวอักษร	ส่วน	อัตราความกว้าง	จำนวนหัว	จำนวนปลาย	หัวแรกที่พบ	ตำแหน่งหัวแรกหรือปลายแรก	ตำแหน่งหัวที่สองหรือปลายที่สอง	เวกเตอร์ทิศทาง
ย	1	3	1	1	C3	00	30	L5L6L7L6L0L2
ร	1	2	1	1	C1	33	30	L2L3L4L2L1L0
ล	1	2	1	1	C2	03	00	L2L1L7L1L2L3L4L5
ว	1	2	1	1	C1	32	00	L2L3L4L5L6
ศ	1	2	2	1	C2	12	21	L7L1L2L3
ษ	1	3	2	0	C4	11	32	L6L0L2
ส	1	2	3	1	C2	03	21	L2L1L0
ห	1	3	2	1	C4	11	31	L6L1L2
ฟ	1	2	2	0	C4	12	31	L6L1L2L7L6L7L6L7
อ	1	2	1	1	C3	02	00	L5L6L0L2L3L4L5
ฮ	1	1	2	0	C3	02	30	L6L0L1L2

ภาคผนวก ฉ

ตารางแสดงรหัสเลขฐานสองของตัวอักษรต้นแบบ

ตัวอักษร	ตัวที่	รหัสเลขฐานสอง
ก	1	01 10 00 10 1111 00 11 11 11 0010 0011 0010 0001 0000 0111 0110 1111
	2	01 10 00 10 1111 00 11 11 11 0010 0011 0001 0000 0111 0110 1111 1111
	3	01 10 00 10 1111 00 11 11 11 0010 0001 0000 0111 0110 0101 1111 1111
	4	01 10 00 10 1111 00 11 11 11 0010 0011 0001 0000 0111 0110 1111 1111
	5	01 10 00 10 1111 00 11 11 11 0010 0011 0010 0001 0000 0111 0110 1111
ข	1	01 10 01 01 1010 00 00 11 00 0001 0000 0111 0110 0101 0110 0000 0010
	2	01 10 01 01 1010 00 00 11 00 0001 0000 0111 0110 0101 0110 0111 0000
	3	01 10 01 01 1001 01 00 11 00 0000 0111 0110 0101 0110 0000 0001 0010
	4	01 10 01 01 1010 00 00 11 00 0001 0000 0111 0110 0101 0110 0000 0001
	5	01 01 01 01 1010 00 00 11 00 0001 0000 0111 0110 0101 0110 0000 0001
ฃ	1	01 01 01 01 1010 00 00 11 00 0010 0000 0110 0101 0110 0111 0000 0001
	2	01 01 01 01 1010 00 00 11 00 0010 0000 0110 0101 0110 0111 0000 0001
	3	01 01 01 01 1010 00 00 11 00 0010 0000 0110 0101 0110 0111 0000 0001
	4	01 01 01 01 1010 00 00 11 00 0010 0000 0110 0101 0110 0111 0000 0001
	5	01 01 01 01 1010 00 00 11 00 0010 0000 0110 0101 0110 0111 0000 0001
ค	1	01 10 01 01 1011 01 10 11 11 0101 0011 0010 0001 0000 0111 0110 1111
	2	01 10 01 01 1011 01 10 11 11 0101 0011 0010 0001 0000 0111 0110 1111
	3	01 10 01 01 1011 01 10 11 11 0101 0011 0010 0001 0000 0111 0110 1111
	4	01 10 01 01 1011 01 10 11 11 0101 0011 0010 0001 0000 0111 0110 1111
	5	01 10 01 01 1011 01 10 11 11 0101 0011 0010 0001 0000 0111 0110 1111
ฅ	1	01 10 01 01 1011 01 10 11 11 0101 0011 0010 0001 0000 0111 0001 0111
	2	01 10 01 01 1011 01 10 11 11 0101 0110 0011 0010 0001 0000 0111 0001
	3	01 11 01 01 1011 01 10 11 11 0101 0011 0010 0001 0000 0111 0000 0111
	4	01 10 01 01 1011 01 10 11 11 0101 0011 0010 0001 0000 0111 0001 0000
	5	01 10 01 01 1011 01 10 11 11 0101 0011 0010 0001 0111 0001 0111 0110

ตัวอักษร	ตัวที่	รหัสเลขฐานสอง
ฆ	1	01 10 10 01 1010 00 00 01 10 0010 0001 0111 0001 0111 0110 0101 0110
	2	01 10 10 01 1010 00 00 01 10 0010 0001 0000 0111 0001 0111 0110 0101
	3	01 10 10 01 1010 00 00 01 10 0010 0001 0000 0111 0001 0111 0110 0101
	4	01 10 10 01 1010 00 00 01 10 0010 0001 0111 0001 0000 0110 0101 0110
	5	01 10 10 01 1010 00 00 01 10 0001 0111 0001 0111 0110 0101 0110 1111
ง	1	01 01 01 01 1100 11 00 00 10 0111 0110 0100 0011 1111 1111 1111 1111
	2	01 10 01 01 1100 11 01 00 10 0110 0100 0011 1111 1111 1111 1111 1111
	3	01 10 01 01 1100 11 00 00 10 0110 0100 0011 1111 1111 1111 1111 1111
	4	01 01 01 01 1100 11 01 00 11 0110 0101 0011 1111 1111 1111 1111 1111
	5	01 01 01 01 1100 11 01 00 10 0110 0101 0011 1111 1111 1111 1111 1111
จ	1	01 10 01 01 1100 10 10 00 00 0111 0001 0010 0011 0100 0101 1111 1111
	2	01 10 01 01 1100 10 10 00 00 0111 0001 0010 0011 0100 0101 1111 1111
	3	01 10 01 01 1100 10 10 00 00 0111 0001 0010 0011 0100 0101 1111 1111
	4	01 10 01 01 1100 10 10 00 00 0111 0001 0010 0011 0100 0101 1111 1111
	5	01 10 01 01 1100 10 10 00 00 0111 0001 0010 0011 0100 0101 1111 1111
ฉ	1	01 11 10 01 1100 01 10 11 10 0110 0001 0000 1111 1111 1111 1111 1111
	2	01 11 10 01 1100 01 10 11 10 0110 0001 0000 1111 1111 1111 1111 1111
	3	01 11 10 01 1100 01 10 11 10 0111 0110 0001 0000 0001 0000 1111 1111
	4	01 11 10 01 1100 01 10 11 10 0110 0111 0001 0000 1111 1111 1111 1111
	5	01 11 10 01 1100 01 10 10 10 0110 0111 0001 0000 1111 1111 1111 1111
ช	1	01 10 10 00 1001 00 00 10 00 0010 0000 0111 1111 1111 1111 1111 1111
	2	01 10 10 00 1010 00 00 11 00 0001 0000 0111 1111 1111 1111 1111 1111
	3	01 10 10 00 1001 00 00 10 00 0001 0000 0111 1111 1111 1111 1111 1111
	4	01 01 01 00 1001 00 00 10 00 0010 0001 0111 1111 1111 1111 1111 1111
	5	01 10 10 00 1010 00 00 11 00 0001 0000 0111 1111 1111 1111 1111 1111

ตัวอักษร	ตัวที่	รหัสเลขฐานสอง
ซ	1	01 10 01 01 1010 00 10 11 00 0010 0001 0111 0001 0000 0110 0101 0110
	2	01 10 01 01 1010 00 01 11 00 0010 0001 0111 0000 0110 0000 0010 0001
	3	01 10 01 01 1010 00 01 11 00 0010 0001 0000 0110 0101 0110 0000 0010
	4	01 10 01 01 1010 00 01 11 00 0010 0001 0111 0000 0111 0101 0110 0000
	5	01 10 01 01 1010 00 01 11 00 0010 0001 0111 0000 0111 0101 0110 0111
ฌ	1	01 11 10 01 1010 00 11 10 10 0010 0011 0010 0001 0000 0111 0110 1111
	2	01 11 10 01 1010 00 11 10 10 0010 0011 0001 0000 0111 0110 1111 1111
	3	01 11 10 01 1010 00 10 10 10 0010 0011 0010 0001 0000 0111 0110 1111
	4	01 11 10 01 1010 00 11 10 10 0010 0011 0001 0000 0111 0110 1111 1111
	5	01 11 10 01 1010 00 11 10 10 0010 0011 0010 0001 0000 0111 0110 1111
ญ	1	10 11 01 01 1010 00 11 11 00 0010 0011 0001 0000 0111 0110 0000 0001
	2	10 11 01 01 1010 00 11 11 00 0010 0011 0010 0001 0000 0111 0110 0000
	3	10 11 01 01 1010 00 11 11 00 0010 0011 0010 0001 0000 0111 0110 0000
	4	10 11 01 01 1010 00 11 11 00 0010 0011 0001 0000 0111 0110 0000 0001
	5	10 11 01 01 1010 00 11 11 00 0010 0011 0001 0000 0111 0110 0111 0000
ฎ	1	01 10 10 00 1001 01 10 10 11 0010 0001 0000 0111 0110 0101 0011 1111
	2	01 10 10 00 1001 01 10 10 11 0010 0001 0000 0111 0110 0101 0011 1111
	3	01 10 10 00 1001 01 10 10 11 0010 0011 0001 0000 0111 0110 0101 0011
	4	01 10 10 00 1001 01 10 10 11 0010 0011 0001 0000 0111 0110 0101 0011
	5	01 01 10 00 1001 01 10 10 11 0010 0001 0000 0111 0110 0101 0011 1111
ฏ	1	01 01 10 00 1001 01 01 01 11 0010 0011 0001 0000 0111 0110 0100 0011
	2	01 01 10 00 1001 01 01 01 11 0010 0011 0001 0000 0111 0110 0100 0011
	3	01 01 10 00 1001 01 10 01 11 0010 0011 0001 0000 0111 0110 0101 0011
	4	01 01 10 00 1001 01 01 01 11 0001 0010 0001 0000 0111 0110 0101 0011
	5	01 01 10 00 1001 01 01 01 11 0010 0011 0001 0000 0111 0110 0100 0011

ตัวอักษร	ตัวที่	รหัสเลขฐานสอง
ฐ	1	10 11 01 01 1100 10 11 11 00 0111 0001 0010 0100 0011 0001 0000 1111
	2	10 10 01 01 1100 10 11 11 00 0000 0001 0010 0011 0100 0011 0001 0000
	3	10 10 01 01 1100 10 11 11 00 0111 0001 0010 0011 0001 0000 0111 0000
	4	10 11 01 01 1100 01 11 11 00 0111 0001 0010 0011 0100 0001 0000 1111
	5	10 11 01 01 1100 10 11 11 00 0000 0001 0010 0011 0100 0011 0001 0000
ฑ	1	01 11 01 01 1010 00 00 11 11 0001 0000 0001 0111 0110 0101 0110 0001
	2	01 11 01 01 1010 00 00 11 11 0001 0111 0001 0111 0110 0101 0110 0001
	3	01 11 01 01 1010 00 01 11 11 0010 0001 0111 0001 0111 0110 0101 0110
	4	01 11 01 01 1010 00 01 11 11 0010 0001 0111 0001 0110 0101 0110 0001
	5	01 11 01 01 1010 00 01 11 11 0010 0001 0111 0000 0111 0110 0101 0110
ฒ	1	01 11 10 01 1100 01 10 10 10 0110 0101 0011 0010 0001 0000 0111 0001
	2	01 11 10 01 1100 01 10 10 10 0110 0101 0100 0010 0001 0000 0111 0001
	3	01 11 10 01 1100 01 10 10 10 0110 0101 0010 0001 0111 0001 0111 0110
	4	01 11 10 01 1100 01 10 10 10 0110 0101 0100 0011 0010 0001 0111 0000
	5	01 11 10 01 1100 01 10 10 10 0101 0011 0010 0001 0111 0001 0111 0110
ณ	1	01 11 10 01 1010 00 11 11 10 0010 0011 0001 0000 0111 0110 0001 0000
	2	01 11 10 01 1010 00 10 11 10 0010 0001 0000 0110 0001 0000 1111 1111
	3	01 11 10 01 1010 00 11 11 10 0010 0011 0001 0000 0111 0110 0000 0001
	4	01 11 10 01 1010 00 11 11 10 0010 0011 0001 0000 0111 0110 0001 1111
	5	01 11 10 01 1010 00 11 11 10 0010 0001 0000 0111 0110 0001 0000 1111
ด	1	01 10 01 01 1100 10 10 11 11 0110 0101 0011 0010 0001 0111 0110 1111
	2	01 10 01 01 1100 01 01 11 11 0110 0101 0011 0010 0001 0000 0111 0110
	3	01 10 01 01 1100 01 01 11 11 0110 0101 0010 0001 0000 0111 0110 1111
	4	01 10 01 01 1100 01 01 11 11 0110 0101 0011 0010 0001 0000 0111 0110
	5	01 10 01 01 1100 01 10 11 11 0110 0101 0011 0010 0001 0000 0111 0110

ตัวอักษร	ตัวที่	รหัสเลขฐานสอง
ต	1	01 10 01 01 1100 01 10 11 11 0110 0101 0011 0010 0001 0000 0111 0001
	2	01 11 01 01 1100 01 10 11 11 0110 0101 0010 0001 0000 0111 0001 0000
	3	01 10 01 01 1100 01 10 11 11 0110 0101 0100 0010 0001 0000 0111 0001
	4	01 10 01 01 1011 01 10 11 11 0110 0101 0011 0010 0001 0000 0111 0110
	5	01 10 01 01 1100 01 01 11 11 0110 0101 0011 0010 0001 0000 0111 0001
ถ	1	01 10 01 01 1010 00 11 11 11 0010 0011 0001 0000 0111 0110 1111 1111
	2	01 10 01 01 1010 00 11 11 11 0010 0011 0001 0000 0111 0110 0111 1111
	3	01 10 01 01 1010 00 11 11 11 0010 0011 0001 0000 0111 0110 1111 1111
	4	01 10 01 01 1010 00 11 11 11 0010 0011 0001 0000 0111 0110 1111 1111
	5	01 10 01 01 1010 00 11 11 11 0010 0011 0001 0000 0111 0110 1111 1111
ท	1	01 11 01 01 1100 00 01 11 11 0110 0001 0000 0111 0110 1111 1111 1111
	2	01 11 01 01 1100 01 01 11 11 0110 0001 0000 0111 0110 1111 1111 1111
	3	01 11 01 01 1100 00 01 11 11 0110 0001 0000 0111 0110 1111 1111 1111
	4	01 11 01 01 1100 01 01 11 11 0110 0000 0001 0000 0111 0110 1111 1111
	5	01 11 01 01 1100 01 01 11 11 0110 0001 0010 0001 0000 0111 0110 1111
ธ	1	01 10 00 10 1111 00 01 11 00 0110 0000 0010 0100 0001 0000 0001 1111
	2	01 10 00 10 1111 00 10 11 00 0110 0000 0010 0011 0100 0011 0001 0000
	3	01 10 00 10 1111 01 01 11 00 0110 0000 0001 0010 0100 0010 0000 1111
	4	01 10 00 10 1111 00 01 11 00 0110 0000 0001 0010 0100 0010 0001 0000
	5	01 10 00 10 1111 00 01 11 00 0110 0000 0010 0100 0001 0000 1111 1111
น	1	01 11 10 01 1100 00 01 10 10 0110 0001 0000 1111 1111 1111 1111 1111
	2	01 11 10 01 1100 00 01 11 10 0110 0001 0000 1111 1111 1111 1111 1111
	3	01 11 10 01 1100 00 01 10 10 0110 0001 0000 1111 1111 1111 1111 1111
	4	01 11 10 01 1100 00 01 10 10 0110 0000 0001 1111 1111 1111 1111 1111
	5	01 11 10 01 1100 00 01 11 10 0110 0001 0000 1111 1111 1111 1111 1111

ตัวอักษร	ตัวที่	รหัสเลขฐานสอง
บ	1	01 11 01 01 1100 01 01 11 00 0110 0000 0010 1111 1111 1111 1111 1111
	2	01 10 01 01 1100 01 01 11 00 0110 0000 0010 0011 1111 1111 1111 1111
	3	01 11 01 01 1100 01 01 11 00 0110 0000 0001 0010 1111 1111 1111 1111
	4	01 11 01 01 1100 01 01 11 00 0110 0000 0001 0010 1111 1111 1111 1111
	5	01 11 01 01 1100 01 01 11 00 0110 0000 0010 1111 1111 1111 1111 1111
ป	1	01 10 01 01 1100 01 01 11 00 0110 0000 0010 1111 1111 1111 1111 1111
	2	01 10 01 01 1100 01 10 11 00 0110 0000 0010 1111 1111 1111 1111 1111
	3	01 10 01 01 1100 01 01 11 00 0110 0000 0001 0010 1111 1111 1111 1111
	4	01 10 01 01 1100 01 01 11 00 0110 0000 0010 1111 1111 1111 1111 1111
	5	01 10 01 01 1100 01 01 11 00 0111 0110 0000 0001 0010 1111 1111 1111
ผ	1	01 10 01 01 1011 00 01 11 00 0101 0110 0000 0001 0111 0001 0010 1111
	2	01 10 01 01 1011 00 01 11 00 0110 0001 0000 0111 0001 0010 1111 1111
	3	01 10 01 01 1011 00 01 11 00 0110 0001 0010 0001 0111 0001 0010 1111
	4	01 10 01 01 1011 00 01 11 00 0110 0001 0010 0001 0111 0001 0010 1111
	5	01 10 01 01 1011 00 01 11 00 0110 0000 0001 0111 0001 0010 1111 1111
ฝ	1	01 01 01 01 1011 00 10 11 00 0110 0001 0111 0110 0001 0010 1111 1111
	2	01 01 01 01 1011 00 10 11 00 0110 0001 0111 0110 0111 0010 1111 1111
	3	01 01 01 01 1011 00 10 11 00 0110 0001 0111 0001 0010 1111 1111 1111
	4	01 01 01 01 1011 00 10 11 00 0110 0000 0001 0111 0110 0111 0001 0010
	5	01 10 01 01 1011 00 01 11 00 0110 0001 0111 0001 0010 1111 1111 1111
พ	1	01 11 01 01 1100 00 01 11 00 0110 0111 0001 0010 0000 0110 0111 0001
	2	01 11 01 01 1100 00 01 11 00 0110 0001 0010 0001 0010 0000 0111 0110
	3	01 11 01 01 1100 01 01 11 00 0110 0001 0010 0001 0111 0110 0001 0010
	4	01 11 01 01 1100 00 01 11 00 0110 0111 0001 0010 0001 0111 0110 0001
	5	01 11 01 01 1100 00 01 11 00 0110 0001 0010 0001 0111 0110 0111 0000

ตัวอักษร	ตัวที่	รหัสเลขฐานสอง
ฟ	1	01 10 01 01 1100 00 10 11 00 0110 0001 0010 0001 0111 0110 0111 0001
	2	01 10 01 01 1100 01 10 11 00 0110 0001 0010 0000 0111 0110 0111 0001
	3	01 10 01 01 1100 00 10 11 00 0110 0000 0001 0010 0000 0111 0110 0001
	4	01 11 01 01 1100 00 10 11 00 0110 0001 0010 0001 0111 0110 0001 0010
	5	01 10 01 01 1100 00 10 11 00 0110 0001 0010 0001 0111 0010 1111 1111
ภ	1	01 11 01 01 1001 01 11 11 11 0001 0010 0011 0001 0000 0111 0110 1111
	2	01 11 01 01 1001 00 10 11 11 0010 0011 0001 0000 0111 0110 1111 1111
	3	01 11 01 01 1001 01 11 11 11 0010 0011 0001 0000 0111 0110 1111 1111
	4	01 11 01 01 1001 01 11 11 11 0010 0011 0001 0000 0111 0110 1111 1111
	5	01 11 01 01 1001 00 10 11 11 0010 0011 0010 0001 0000 0111 0110 1111
ม	1	01 10 10 01 1100 01 01 01 10 0110 1111 1111 1111 1111 1111 1111 1111
	2	01 10 10 01 1100 01 01 01 10 0111 0101 0110 1111 1111 1111 1111 1111
	3	01 10 10 01 1100 01 01 01 10 0110 0111 0110 1111 1111 1111 1111 1111
	4	01 10 10 01 1100 01 01 01 10 0111 0110 1111 1111 1111 1111 1111 1111
	5	01 11 11 01 1111 01 01 01 00 0001 1111 1111 1111 1111 1111 1111 1111
ย	1	01 11 01 01 1011 00 00 11 00 0101 0110 0111 0110 0000 0010 1111 1111
	2	01 10 01 01 1011 00 00 11 00 0101 0110 0111 0110 0000 0001 0000 0001
	3	01 10 01 01 1011 00 00 11 00 0101 0110 0111 0101 0110 0000 0010 1111
	4	01 10 01 01 1011 00 00 11 00 0101 0110 0111 0110 0000 0010 1111 1111
	5	01 10 01 01 1011 00 00 11 00 0101 0110 0111 0101 0110 0111 0000 0001
ร	1	01 10 01 01 1001 11 11 11 00 0010 0011 0100 0010 0001 0000 1111 1111
	2	01 10 01 01 1001 10 11 11 00 0010 0011 0100 0011 0001 0000 0001 1111
	3	01 10 01 01 1001 11 11 11 00 0010 0011 0100 0011 0001 0000 1111 1111
	4	01 10 01 01 1001 11 11 11 00 0010 0011 0100 0001 0000 1111 1111 1111
	5	01 10 01 01 1001 11 11 11 00 0010 0011 0100 0011 0100 0011 0001 0000

ตัวอักษร	ตัวที่	รหัสเลขฐานสอง
ล	1	01 10 01 01 1010 00 11 00 00 0010 0001 0111 0001 0010 0011 0100 0101
	2	01 10 01 01 1010 00 11 00 00 0010 0000 0111 0000 0010 0011 0100 0101
	3	01 10 01 01 1010 01 11 00 00 0010 0001 0111 0000 0010 0011 0100 0101
	4	01 10 01 01 1010 00 11 00 00 0010 0001 0111 0001 0010 0011 0100 0101
	5	01 10 01 01 1010 00 11 00 00 0010 0000 0111 0010 0011 0100 0101 1111
ว	1	01 10 01 01 1001 11 10 00 00 0010 0011 0100 0101 0110 1111 1111 1111
	2	01 10 01 01 1001 11 11 00 00 0001 0010 0011 0100 0101 1111 1111 1111
	3	01 01 01 01 1001 11 11 00 00 0010 0011 0100 0101 1111 1111 1111 1111
	4	01 01 01 01 1001 11 11 00 00 0001 0010 0011 0100 0101 0110 1111 1111
	5	01 01 01 01 1001 11 11 00 00 0001 0010 0011 0100 0101 0110 1111 1111
ศ	1	01 10 10 01 1111 01 10 10 01 0111 0001 0010 0011 1111 1111 1111 1111
	2	01 10 10 01 1111 01 01 10 01 0111 0001 0010 0011 1111 1111 1111 1111
	3	01 10 10 01 1111 01 01 10 01 0010 0001 1111 1111 1111 1111 1111 1111
	4	01 10 10 01 1111 01 10 10 01 0000 0001 0010 0011 1111 1111 1111 1111
	5	01 10 10 01 1111 01 10 10 01 0010 0001 0000 1111 1111 1111 1111 1111
ษ	1	01 11 10 00 1100 01 01 11 10 0110 0000 0010 1111 1111 1111 1111 1111
	2	01 10 10 01 1100 00 01 11 10 0110 0111 0000 0010 1111 1111 1111 1111
	3	01 10 10 00 1100 00 01 11 10 0110 0000 0001 0010 0100 0011 1111 1111
	4	01 11 10 01 1100 01 01 11 10 0111 0110 0000 0010 1111 1111 1111 1111
	5	01 10 10 01 1100 01 00 10 10 0110 0000 0010 0011 1111 1111 1111 1111
ส	1	01 10 10 01 1010 00 11 10 01 0010 0001 1111 1111 1111 1111 1111 1111
	2	01 10 10 01 1010 00 11 10 01 0010 0001 0000 1111 1111 1111 1111 1111
	3	01 10 11 01 1010 00 11 01 01 0010 0001 1111 1111 1111 1111 1111 1111
	4	01 10 11 01 1010 00 11 01 01 0010 0001 1111 1111 1111 1111 1111 1111
	5	01 10 11 01 1010 00 11 01 01 0010 0001 1111 1111 1111 1111 1111 1111

ตัวอักษร	ตัวที่	รหัสเลขฐานสอง
ห	1	01 11 10 01 1100 01 01 11 01 0110 0001 0010 1111 1111 1111 1111 1111
	2	01 11 10 01 1100 01 01 11 01 0110 0001 1111 1111 1111 1111 1111 1111
	3	01 10 10 01 1100 01 01 11 01 0110 0001 1111 1111 1111 1111 1111 1111
	4	01 10 10 01 1100 01 01 11 01 0111 0110 0001 1111 1111 1111 1111 1111
	5	01 10 10 01 1100 01 01 11 01 0110 0001 0010 0001 0010 0001 0010 1111
ฬ	1	01 10 10 00 1100 01 10 11 01 0110 0001 0010 0111 0110 0111 0110 0111
	2	01 10 10 00 1100 00 10 11 00 0110 0001 0010 0111 0110 0001 0010 1111
	3	01 11 10 00 1100 00 10 11 01 0110 0001 0111 0001 0010 1111 1111 1111
	4	01 11 10 00 1100 00 10 11 01 0110 0001 0010 0001 0010 0111 0110 0001
	5	01 10 10 00 1100 00 10 11 01 0111 0110 0001 0010 0001 0111 0110 0001
อ	1	01 10 01 01 1011 00 10 00 00 0101 0110 0000 0010 0011 0100 0101 1111
	2	01 10 01 01 1011 00 10 00 00 0110 0000 0010 0011 0100 0101 1111 1111
	3	01 10 01 01 1011 00 10 00 00 0110 0000 0010 0011 0100 0101 1111 1111
	4	01 10 01 01 1011 00 10 00 00 0110 0000 0001 0010 0011 0100 0101 1111
	5	01 10 01 01 1011 00 01 00 00 0110 0000 0010 0011 0100 1111 1111 1111
ฮ	1	01 01 10 00 1011 00 10 11 00 0110 0000 0001 0010 1111 1111 1111 1111
	2	01 01 10 00 1011 00 10 11 00 0110 0000 0001 0010 0011 1111 1111 1111
	3	01 01 10 00 1011 00 10 11 00 0110 0000 0010 1111 1111 1111 1111 1111
	4	01 10 10 00 1011 00 10 11 00 0110 0000 0001 0010 0011 0101 1111 1111
	5	01 10 10 00 1011 00 10 10 00 0110 0000 0010 0011 0101 1111 1111 1111

ประวัติผู้เขียน

นางสาวอลิศรา หงษ์พร้อมญาติ เกิดเมื่อวันที่ 23 เมษายน 2519 ที่จังหวัดกรุงเทพมหานคร สำเร็จการศึกษาระดับปริญญาตรีจากมหาวิทยาลัยเกษตรศาสตร์ คณะวิทยาศาสตร์ ภาควิชาคณิตศาสตร์ เมื่อปีการศึกษา 2539

ปี พ.ศ. 2540 เข้าทำงานในมหาวิทยาลัยหัวเฉียวเฉลิมพระเกียรติ ตำแหน่งอาจารย์ประจำสังกัดคณะวิทยาศาสตร์และเทคโนโลยี สาขาวิชาคณิตศาสตร์และสถิติ