

การเลือกลักษณะเด่นสำหรับการจำตัวอักษรตัวพิมพ์บนโครงข่ายประสาทเทียม

PRINTED CHARACTER RECOGNITION USING FEATURE
SELECTION ON NEURAL NETWORK

มีชัย โลหะการ

MEECHAI LOHAKAN

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต

สาขาวิชาวิศวกรรมไฟฟ้า

บัณฑิตวิทยาลัย

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ. 2546

ISBN 974-324-495-6

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

การเลือกลักษณะเด่นสำหรับการจำตัวอักษรตัวพิมพ์บนโครงข่ายประสาทเทียม

PRINTED CHARACTER RECOGNITION USING FEATURE
SELECTION ON NEURAL NETWORK



มีชัย โลหะการ

MEECHAI LOHAKAN

เลขหมู่.....
เลขทะเบียน..... 47647
วัน, เดือน, ปี..... 21 ส.ค. 2546

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

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต

สาขาวิชาวิศวกรรมไฟฟ้า

บัณฑิตวิทยาลัย

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ. 2546

ISBN 974-324-495-6

**PRINTED CHARACTER RECOGNITION USING FEATURE
SELECTION ON NEURAL NETWORK**

MEECHAI LOHAKAN

**A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF ENGINEERING IN ELECTRICAL ENGINEERING
SCHOOL OF GRADUATE STUDIES
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG**

2003

ISBN 974-324-495-6

บัณฑิตวิทยาลัย
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ใบรับรองวิทยานิพนธ์

หัวข้อวิทยานิพนธ์ การเลือกลักษณะเด่นสำหรับการจำตัวอักษรตัวพิมพ์บนโครงข่ายประสาทเทียม
PRINTED CHARACTER RECOGNITION USING FEATURE SELECTION
ON NEURAL NETWORK

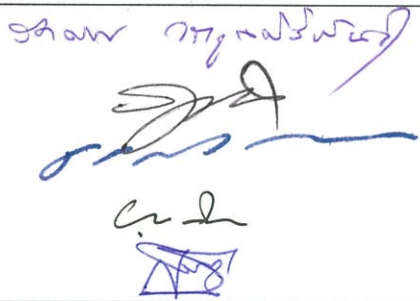
ชื่อนักศึกษา นายมีชัย โลหะการ

รหัสประจำตัว 41061143

ปริญญา วิศวกรรมศาสตรมหาบัณฑิต

สาขาวิชา วิศวกรรมไฟฟ้า

อาจารย์ผู้ควบคุมวิทยานิพนธ์ รศ.ดร.สุรพันธุ์ เอื้อไพบูลย์

คณะกรรมการสอบวิทยานิพนธ์		ลายมือชื่อ
รศ.ดร.รัตติกง	วราภูลศิริพันธุ์	
ผศ.สุชาติ	คุณทวีเทพ	
รศ.ดร.มนัส	สังวรศิลป์	
ดร.ยุทธนา	คิศจิเดียว	
รศ.ดร.สุรพันธุ์	เอื้อไพบูลย์	

วัน/เดือน/ปี ที่สอบ 16 พฤษภาคม 2546 เวลา 10.30-12.30 น.

สถานที่สอบ ณ อาคาร 12 ชั้น ชั้น 4 (ห้อง E12-403)

บัณฑิตวิทยาลัยรับรองแล้ว

(รศ.ดร.บุญวัฒน์ อัครฐ์)
คณบดีบัณฑิตวิทยาลัย

วันที่.....๒๖.....เดือน.....๕.....พ.ศ.....๒๕๔๖

หัวข้อวิทยานิพนธ์	การเลือกลักษณะเด่นสำหรับการจำตัวอักษรตัวพิมพ์บน โครงข่ายประสาทเทียม
นักศึกษา	นายมีชัย โลหะการ
รหัสประจำตัว	41061143
ปริญญา	วิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชา	วิศวกรรมไฟฟ้า
พ.ศ.	2546
อาจารย์ควบคุมวิทยานิพนธ์	รองศาสตราจารย์ ดร.สุรพันธุ์ เอื้อไพฑูริย์

บทคัดย่อ

ในปัจจุบันกระบวนการรู้จำตัวอักษร ได้นำทฤษฎีโครงข่ายประสาทเทียมมาประยุกต์ใช้กันมากขึ้น เพื่อเพิ่มประสิทธิภาพในการรู้จำ โครงสร้างที่นิยมใช้กันคือ Feed Forward Backpropagation เนื่องจากมีเปอร์เซ็นต์ผิดพลาดต่ำ แต่จุดด้อยของโครงสร้างนี้อยู่ที่ใช้เวลาในการเรียนรู้เวลานาน และมีโครงสร้างที่ซับซ้อน

วิธีการที่นำเสนอนี้เป็นวิธีการใหม่ในการรู้จำตัวอักษรตัวพิมพ์ภาษาอังกฤษ โดยใช้โครงข่ายประสาทเทียมแบบGNU ซึ่งมีโครงสร้างที่ไม่ซับซ้อนใช้เวลาในการเรียนรู้สั้น(Learning) และประยุกต์วิธีการ ออปติไมซ์ ในการเลือกจุดภาพ(Pixel) ที่มีข้อมูลบรรจุอยู่สูงในการนำไปสร้าง GRAM ซึ่งเป็นส่วนของหน่วยความจำที่เก็บค่าที่ได้เรียนรู้ไว้ และนำระยะทางแฮมมิง ไปใช้ในการเปรียบเทียบผลลัพธ์เพื่อใช้ในการเรียกคืน(Recall) ผลลัพธ์ในการทดลองนอกจากจะแสดงให้เห็นว่าสามารถเรียกคืนได้แล้วยังสามารถทนทานต่อความผิดพลาดบางส่วนได้

Thesis Title	Printed Character Recognition using Feature Selection on Neural Network
Student	Mr. Meechai Lohakan
Student ID.	41061143
Degree	Master of Engineering
Programme	Electrical Engineering
Year	2003
Thesis Advisor	Assoc.Prof. Dr. Surapan Airphaiboon

ABSTRACT

At present, the Artificial Neural Network is applied in the process of Character Recognition to improve a recognition rate. The Feed Forward Backpropagation model is popular because it has a few of error rate. However, the disadvantages of this model are the complexity of the model and the long learning time.

In this method, printed English character Recognition using 1 Neural Network is proposed. The network has a simple model and short training time. In optimization part, the pixels that have maximum information is selected for building GRAM. GRAM, store the training values, is the memory used for recall by Hamming Distance. Finally, the output of experiment shows that the network can recall printed English character efficiently. In addition, it can tolerate the distorted pattern.

กิตติกรรมประกาศ

วิทยานิพนธ์นี้สำเร็จลุล่วงไปได้ด้วยดี ก็เนื่องมาจากการให้ความอนุเคราะห์ให้คำปรึกษาที่ดีในด้านวิชาการ และการสนับสนุนจาก รองศาสตราจารย์ ดร.สุรพันธุ์ เอื้อไพบูรณ์ ซึ่งเป็นอาจารย์ผู้ควบคุมวิทยานิพนธ์ รองศาสตราจารย์ ดร. มนัส สังวรศิลป์ และ ดร. ยุทธนา กิดใจเดียว ตลอดจนเพื่อนๆ ที่ห้องทดลองที่คอยให้กำลังใจ และช่วยเหลือในด้านต่างๆ เป็นอย่างดี จึงใคร่ขอขอบพระคุณทุกท่านไว้ ณ ที่นี้

มีชัย โลหะการ

สารบัญ

	หน้า
บทคัดย่อภาษาไทย.....	I
บทคัดย่อภาษาอังกฤษ.....	II
กิตติกรรมประกาศ.....	III
สารบัญ.....	IV
สารบัญตาราง.....	VI
สารบัญภาพ.....	VII
บทที่ 1 บทนำ.....	1
1.1 ความเป็นมาและความสำคัญของปัญหา.....	1
1.2 วัตถุประสงค์ของการวิจัย.....	1
1.3 ทฤษฎี หรือแนวคิดที่ใช้ในการวิจัย.....	1
1.4 ขอบเขตการวิจัย.....	2
1.5 วิธีที่ใช้ในการวิจัย.....	2
บทที่ 2 การเตรียมข้อมูลภาพ.....	3
2.1 รูปภาพต้นแบบ.....	3
2.2 ตำแหน่งมาตรฐาน.....	5
2.3 ข้อมูล 1 มิติ.....	9
บทที่ 3 โครงข่ายประสาทเทียม.....	3
3.1 โครงข่ายประสาททางชีววิทยา.....	14
3.2 โครงข่ายประสาทเทียม.....	15
3.3 แอส โซซิเอทีฟเมมโมรี.....	48
3.4 ลักษณะเด่น (Feature).....	53
บทที่ 4 การทดลอง.....	54
4.1 รูปภาพต้นแบบ.....	54
4.2 การเตรียมข้อมูลภาพ.....	56
4.3 การออกแบบโครงข่ายประสาทเทียม.....	59

สารบัญ(ต่อ)

	หน้า
4.4 การทดสอบ.....	64
4.5 ผลการทดสอบ.....	66
4.6 สรุปผลการทดสอบ.....	70
บทที่ 5 สรุปผลการวิจัย และข้อเสนอแนะ.....	73
5.1 สรุปผลการวิจัย.....	73
5.2 ข้อเสนอแนะ.....	73
บรรณานุกรม.....	75
ภาคผนวก.....	77
ภาคผนวก ก	78

สารบัญตาราง

ตารางที่	หน้า
3.1 ทรานส์เฟอร์ฟังก์ชันแบบต่างๆ.....	20
3.2 ฟังก์ชัน AND.....	39
3.3 ฟังก์ชัน XOR.....	40
4.1 รูปภาพต้นแบบ(บางส่วน).....	55
4.2 รูปแบบของเอาต์พุตข้อมูล (บางส่วน).....	60
4.3 ข้อมูลของ GRAM ที่ได้สร้างขึ้นตามตำแหน่งที่ได้จากการออปติไมซ์.....	62
4.4 การเปรียบเทียบเวลาในการเรียนรู้.....	71

สารบัญภาพ

รูปที่	หน้า
2.1 ระบบของการจำตัวอักษร.....	3
2.2 รูปภาพสองระดับ (Binary Image).....	4
2.3 วิธีการจัดรูปภาพให้อยู่ในตำแหน่งมาตรฐาน.....	5
2.4 การหากรอบภาพ.....	6
2.5 ตำแหน่งมาตรฐาน.....	7
2.6 ตำแหน่งมาตรฐาน.....	7
2.7 การย้ายภาพ.....	8
2.8 ตัวแปรที่สำคัญในกระบวนการย้ายภาพ.....	9
2.9 การแปลงภาพ 2 มิติ ไปเป็นข้อมูล 1 มิติ.....	10
3.1 โครงสร้างและส่วนประกอบของเซลล์ประสาท (Neuron).....	11
3.2 เซลล์ประสาทเทียม.....	16
3.3 เซลล์ประสาทเทียมแบบเซลล์เดียวอินพุตเดียว.....	17
3.4 ทรานส์เฟอร์ฟังก์ชันแบบฮาร์ดลิมิต.....	18
3.5 เซลล์ประสาทเทียมแบบเซลล์เดียวหลายอินพุต.....	19
3.6 ทรานส์เฟอร์ฟังก์ชันที่ถูกเลื่อนในแนวนอนตามค่าไบอัส.....	21
3.7 โครงข่ายประสาทเทียมแบบชั้นเดียว.....	24
3.8 โครงข่ายประสาทเทียมแบบหลายชั้น.....	28
3.9 โครงข่ายประสาทเทียมที่มี 2 อินพุต และ 1 เอาต์พุต ที่ใช้แก้ปัญหา AND.....	38
3.10 ฟังก์ชันฮาร์ดลิมิต.....	39
3.11 จุดต่างๆ บนระนาบและเส้นแบ่งพื้นที่เอาต์พุตของการแก้ปัญหา AND.....	40
3.12 การใช้เพอเซปตรอนปรกติแบ่งพื้นที่เอาต์พุตซึ่งไม่สามารถแก้ปัญหา XOR ได้.....	41
3.13 โครงข่ายประสาทเทียม 3 อินพุต และ 1 เอาต์พุต ที่ใช้แก้ปัญหา XOR.....	41
3.14 โครงข่ายประสาทเทียมแบบหลายชั้นเน้นไปที่ตัวแปรที่ต้องใช้ในการคำนวณ.....	43
3.15 โครงข่ายประสาทเทียมแบบหลายชั้นที่ใช้ประมาณค่าของฟังก์ชัน $g(p)$	44
3.16 ระยะทางแฮมมิง.....	48
3.17 โครงสร้างทั่วไปของแอสโซซิเอทีฟเมมโมรี.....	49
3.18 โครงสร้างของ GNU.....	50
3.19 การเรียนรู้ของ GNU.....	51

สารบัญญภาพ(ต่อ)

รูปที่	หน้า
3.20 การออปติไมซ์.....	52
3.21 การสร้าง GRAM.....	52
3.22 การเรียกคืน (Recall).....	53
4.1 ระบบของการจำตัวอักษร.....	54
4.2 การสร้างรูปภาพต้นแบบ 2 ระดับ.....	55
4.3 การสร้างรูปภาพต้นแบบ 2 ระดับ.....	56
4.4 โปรแกรมในสถานะที่รอรับคำสั่ง.....	57
4.5 การเตรียมข้อมูล.....	58
4.6 การเรียนรู้ของ GNU.....	59
4.7 โครงสร้างของ GNU.....	60
4.8 การต่อ Input Connection (N).....	61
4.9 การสร้าง GRAM.....	62
4.10 การเรียกคืน (Recall).....	63
4.11 กลุ่มรูปภาพที่เหมือนกับการเรียนรู้ทุกประการ.....	64
4.12 กลุ่มรูปภาพที่มีตำแหน่งของภาพไม่อยู่ในตำแหน่งมาตรฐาน.....	64
4.13 กลุ่มรูปภาพที่มีสัญญาณรบกวน.....	65
4.14 กลุ่มรูปภาพที่มีตัวอักษรไม่สมบูรณ์.....	65
4.15 ประสิทธิภาพการจำของกลุ่มรูปภาพที่เหมือนกับการเรียนรู้ทุกประการ.....	66
4.16 ผลลัพธ์ของ โปรแกรมกลุ่มรูปภาพที่เหมือนกับการเรียนรู้ทุกประการ.....	66
4.17 ประสิทธิภาพการจำของกลุ่มรูปภาพที่มีตำแหน่งของภาพไม่อยู่ในตำแหน่งมาตรฐาน.....	67
4.18 ผลลัพธ์ของ โปรแกรมกลุ่มรูปภาพที่มีตำแหน่งของภาพไม่อยู่ในตำแหน่งมาตรฐาน.....	67
4.19 ประสิทธิภาพการจำของกลุ่มรูปภาพที่มีสัญญาณรบกวน.....	68
4.20 ผลลัพธ์ของ โปรแกรมกลุ่มรูปภาพที่มีสัญญาณรบกวน.....	68
4.21 ประสิทธิภาพการจำของกลุ่มรูปภาพที่มีตัวอักษรไม่สมบูรณ์.....	69
4.22 ผลลัพธ์ของ โปรแกรมกลุ่มรูปภาพที่มีตัวอักษรไม่สมบูรณ์.....	69
4.23 ประสิทธิภาพเมื่อทำการทดลองกับภาพต้นแบบจาก Scanner.....	70
4.24 กลุ่มรูปภาพเมื่อทำการทดลองกับภาพต้นแบบจาก Scanner.....	70
4.25 ประสิทธิภาพการจำเฉลี่ยจำแนกตามตัวอักษร.....	71

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ในการรู้จำตัวอักษรนั้นได้มีการพัฒนามาเป็นเวลานาน และมีหลายวิธีในการรู้จำเช่น การพิจารณาหัวของตัวอักษร, การใช้ส่วนโค้ง-เว้า, การใช้มุมสมมาตร เป็นต้น ในปัจจุบันกระบวนการรู้จำพัฒนาไปใช้ทฤษฎีโครงข่ายประสาทเทียม(Artificial Neural Network) โดยโครงสร้างที่นิยมใช้กันอย่างแพร่หลายคือ Feed Forward Backpropagation ซึ่งมี เปอร์เซ็ปชันการผิดพลาดต่ำ และใช้กันมากในแถบอเมริกา แต่การออกแบบโครงสร้างนี้ค่อนข้างซับซ้อน และจุดด้อยอีกประการคือ การเรียนรู้ (Learning) ที่ต้องใช้เวลา

ในงานวิจัยนี้ได้นำเสนอแนวความคิดใหม่ของโครงข่ายประสาทเทียม ใช้ในการรู้จำตัวอักษรตัวพิมพ์ ภาษาอังกฤษที่สามารถใช้เวลาในการเรียนรู้สั้น โครงสร้างที่เลือกใช้เป็นแบบ การรู้จำภาพนิ่ง (Static Recognition) เป็นโครงสร้างแบบ Feedforward ซึ่งสามารถรู้จำตัวอักษรได้ในการวนรอบเพียงครั้งเดียว (1 Iteration) การเรียนรู้ใช้แบบมีผู้สอน (Supervised Learning) และการเรียกคืนเป็นแบบ Auto-Associative Memory

1.2 วัตถุประสงค์ของการวิจัย

เพื่อสร้างระบบการรู้จำตัวอักษรภาษาอังกฤษตัวพิมพ์ที่ใช้เวลาในการเรียนรู้สั้น

1.3 ทฤษฎี หรือแนวความคิดที่ใช้ในงานวิจัย

1. การเตรียมข้อมูล
2. แนวคิดจากระบบประสาทของมนุษย์
3. ทฤษฎีโครงข่ายประสาทเทียม
4. ทฤษฎีการอปติไมซ์ของโครงข่ายประสาทเทียมแบบ GNU
5. ระยะเวลาแฮมมิง

1.4 ขอบเขตการวิจัย

1. เป็นตัวอักษรตัวพิมพ์ ภาษาอังกฤษ A-Z
2. เป็นภาพบิตแมป (Bitmap) แบบ 2 ระดับ ขนาด 64×64 จุดภาพ
3. ขนาดของตัวอักษรต้องมีความใกล้เคียงกัน
4. รูปภาพที่นำมาทดสอบต้องเป็นตัวอักษรที่สมบูรณ์

1.5 วิธีที่ใช้ในการดำเนินงานวิจัย

1. ศึกษาทฤษฎีโครงข่ายประสาทเทียมในแบบต่างๆ
2. ศึกษาทฤษฎีการอปติไมซ์ของโครงข่ายประสาทเทียมแบบ GNU
3. ศึกษาการเตรียมข้อมูล
4. ทำการสร้างระบบทั้งส่วนของการเตรียมข้อมูล และ โครงข่ายประสาทเทียมเพื่อใช้ในการจำตัวอักษร
5. ทำการทดลองกับกลุ่มข้อมูลแบบต่างๆที่ได้เตรียมไว้
6. สรุปผลการวิจัย และเสนอแนะแนวทางในการพัฒนาต่อไป
7. เสนองานวิจัยในรูปวิทยานิพนธ์

ในวิทยานิพนธ์ฉบับนี้แบ่งออกได้เป็น 5 บทดังนี้

บทที่ 1 บทนำ

บทที่ 2 อธิบายถึงการเตรียมข้อมูลภาพ ประกอบด้วย การเตรียมรูปภาพต้นแบบ การหาตำแหน่งมาตรฐานโดยใช้จุดศูนย์กลาง การย้ายภาพ และการแปลงข้อมูล 2 มิติ ไปเป็นข้อมูล 1 มิติ

บทที่ 3 กล่าวถึงทฤษฎีของโครงข่ายประสาทเทียม เซลล์ประสาททางชีววิทยา เซลล์ประสาทเทียม โครงข่ายประสาทเทียมแบบชั้นเดียว โครงข่ายประสาทเทียมแบบหลายชั้น ระยะทางแฮมมิง GNU การอปติไมซ์ การเรียนรู้ และการเรียกคืน

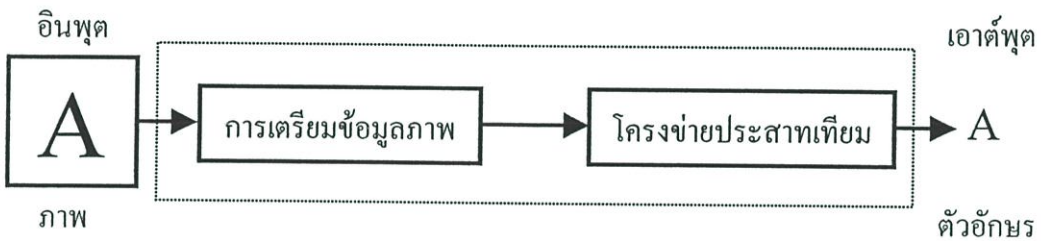
บทที่ 4 อธิบายถึงการใช้ทฤษฎีในบทที่ 2 และ 3 มาสร้างเป็นระบบรู้จำ การออกแบบโครงข่ายประสาทเทียม การสร้างรูปภาพต้นแบบ การเตรียมข้อมูลภาพ การออกแบบ GRAM และการทดสอบโดยใช้กลุ่มของข้อมูลแบบต่างๆ

บทที่ 5 สรุปผลการวิจัย และข้อเสนอแนะ

บทที่ 2

การเตรียมข้อมูลภาพ

การเตรียมข้อมูลภาพ หรือเรียกอีกอย่างว่า การจัดการล่วงหน้า (Pre-Processing) เป็นกระบวนการที่สำคัญอีกกระบวนการหนึ่ง โดยขั้นตอนของการเตรียมข้อมูลภาพนั้นขึ้นอยู่กับลักษณะของโครงข่ายประสาทเทียม ที่เลือกใช้เพื่อเป็นการเตรียมข้อมูลของภาพให้อยู่ในรูปแบบมาตรฐาน (Normalized Image) ก่อนจะนำไปเข้าโครงข่ายประสาทเทียมต่อไป โครงข่ายประสาทเทียมที่พร้อมใช้งานนั้นต้องผ่านการเรียนรู้จนกระทั่งได้ค่าถ่วงน้ำหนักที่เสถียรแล้ว ระบบของการรู้จำตัวอักษรโดยทั่วไปเป็นดังรูปที่ 2.1



รูปที่ 2.1 ระบบของการรู้จำตัวอักษร

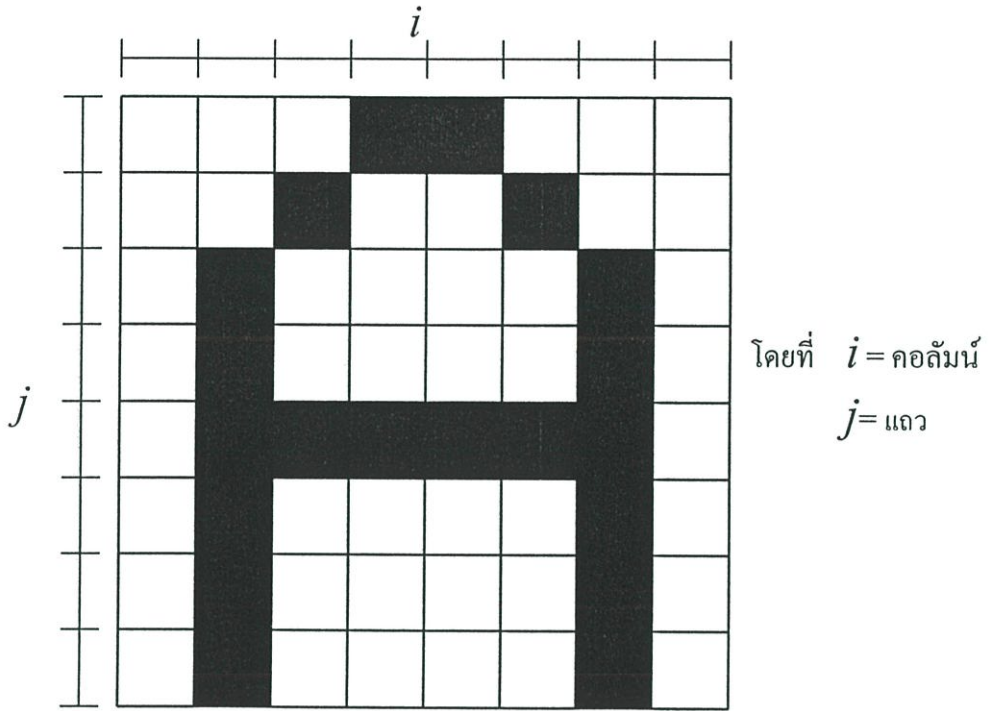
2.1 รูปภาพต้นแบบ

ภาพอินพุต (Input Pattern) มีลักษณะเป็นรูปภาพแบบบิตแมป (Bitmap) ที่เป็นภาพสองระดับ (Binary Image) เพื่อให้ง่ายในการนำไปใช้งานภาพสองระดับเป็นรูปภาพที่ประกอบไปด้วยส่วนที่เป็นเนื้อภาพ และส่วนที่ว่างเปล่าดังรูปที่ 2.2 ส่วนที่เป็นสีดำเป็นส่วนที่เป็นเนื้อภาพและส่วนที่เป็นสีขาวคือส่วนที่เป็นส่วนว่างเปล่า

ในการนำภาพสองระดับมาใช้งานนั้นต้องทำการแปลงความภาพนั้นไปเป็นสมการ หรือเป็นกระบวนการทางคณิตศาสตร์ก่อนจึงจะสามารถนำภาพสองระดับนั้นไปคำนวณได้ และกระบวนการที่เหมาะสมในการแทนค่าทางคณิตศาสตร์ของภาพสองระดับคือ กระบวนการทางเมทริกซ์ การแทนค่าในภาพสองระดับนั้นทำได้โดยสามารถได้ 2 แบบ

1. ให้สีดำแทนค่าด้วย 1 และสีขาวแทนค่าด้วย 0
2. ให้สีดำแทนค่าด้วย 0 และสีขาวแทนค่าด้วย 1

สามารถแทนค่าภาพสองระดับในรูปที่ 2.2 ได้ดังสมการที่ 2.1 และ 2.2 ในทางปฏิบัติต้องเลือกใช้เพียงอย่างใดอย่างหนึ่ง



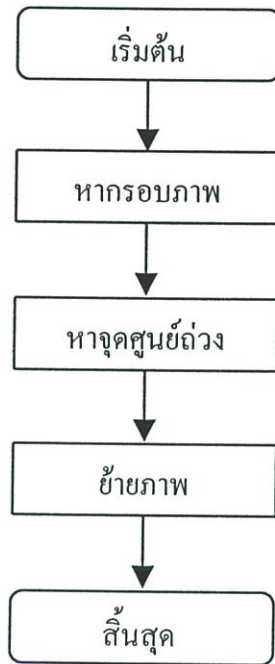
รูปที่ 2.2 รูปภาพสองระดับ (Binary Image)

$$f(i, j) = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \quad (2.1)$$

$$f(i, j) = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 & 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 & 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 & 1 & 1 & 0 & 1 \end{bmatrix} \quad (2.2)$$

2.2 ตำแหน่งมาตรฐาน

การปรับตัวอักษรให้อยู่ในตำแหน่งมาตรฐานนั้น เป็นสิ่งที่จำเป็นในระบบการรู้จำตัวอักษร ตำแหน่งมาตรฐานเป็นตำแหน่งที่ถูกกำหนดขึ้นตามการออกแบบนั้นๆ เพื่อทำการปรับรูปภาพที่เข้ามาให้ตรงตามตำแหน่งมาตรฐานที่กำหนดไว้ โดยทั่วไปแล้วตำแหน่งมาตรฐานจะเป็นตำแหน่งกึ่งกลางของรูปภาพวิธีการจัดรูปภาพให้อยู่ในตำแหน่งมาตรฐานแสดงได้ดังรูปที่ 2.5



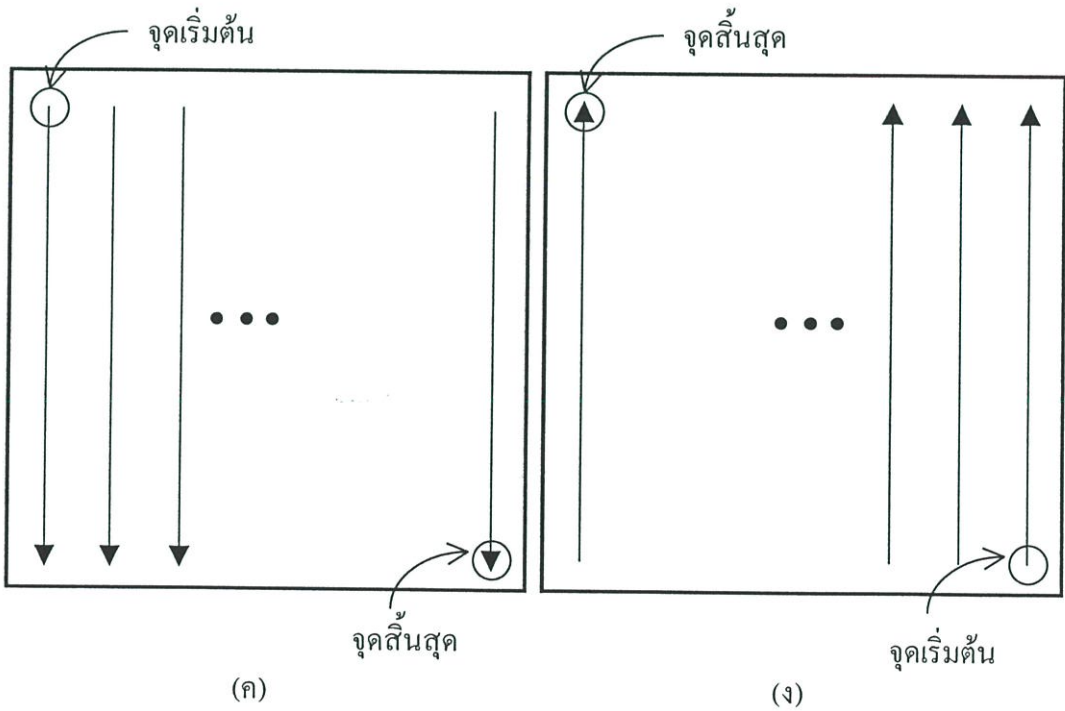
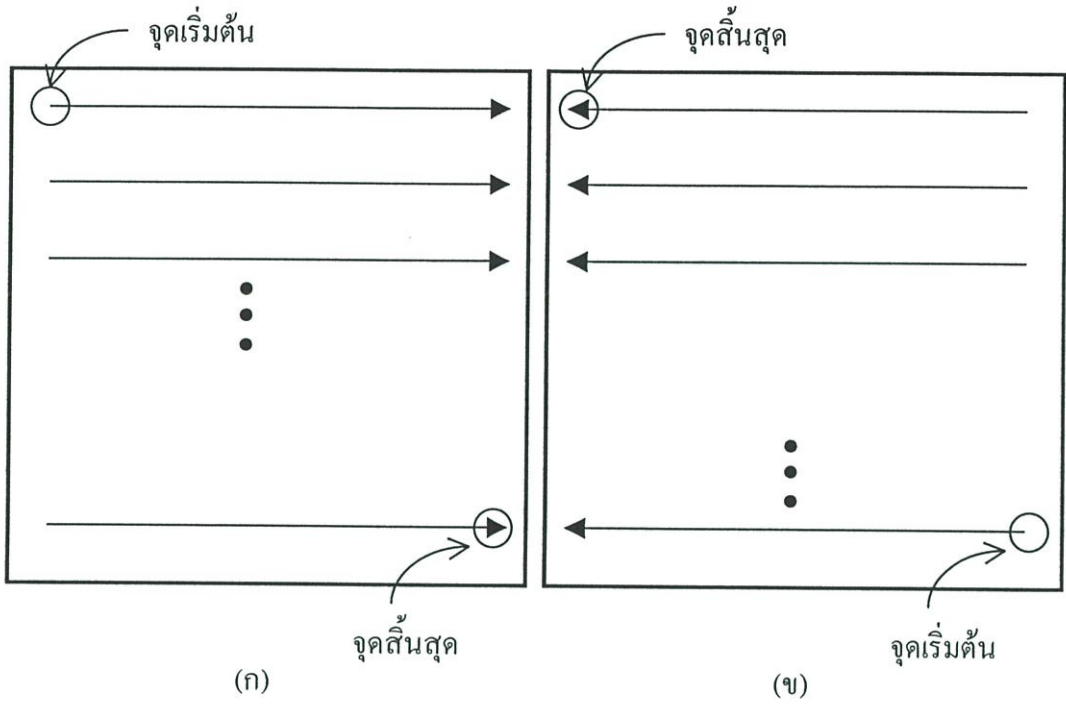
รูปที่ 2.3 วิธีการจัดรูปภาพให้อยู่ในตำแหน่งมาตรฐาน

2.2.1 การหาขอบภาพ (Border)

การหาขอบภาพต้องทำการสแกนหา 4 ครั้งคือ

- กรอบบน (UB) สแกนจากซ้ายไปขวา และบนลงล่างจนกระทั่งพบจุดที่เป็นสีดำ จึงหยุดการสแกน และเก็บตำแหน่ง i และ j นั้นไว้เป็นกรอบบน
- กรอบล่าง (DB) สแกนจากขวาไปซ้าย และล่างขึ้นบนจนกระทั่งพบจุดที่เป็นสีดำจึงหยุดการสแกน และเก็บตำแหน่ง i และ j นั้นไว้เป็นกรอบล่าง
- กรอบซ้าย (LB) สแกนจากบนลงล่าง และซ้ายไปขวาจนกระทั่งพบจุดที่เป็นสีดำจึงหยุดการสแกน และเก็บตำแหน่ง i และ j นั้นไว้เป็นกรอบซ้าย
- กรอบขวา (RB) สแกนจากล่างขึ้นบน และซ้ายไปขวาจนกระทั่งพบจุดที่เป็นสีดำจึงหยุดการสแกน และเก็บตำแหน่ง i และ j นั้นไว้เป็นกรอบขวา

กรอบภาพที่ได้จะนำมาใช้ในการหาจุดศูนย์กลางต่อไป วิธีการหาขอบภาพดังแสดงให้เห็นในรูปที่ 2.4



รูปที่ 2.4 การหากรอบภาพ

(ก) การหากรอบบน (UB)

(ข) การหากรอบล่าง (DB)

(ค) การหากรอบซ้าย (LB)

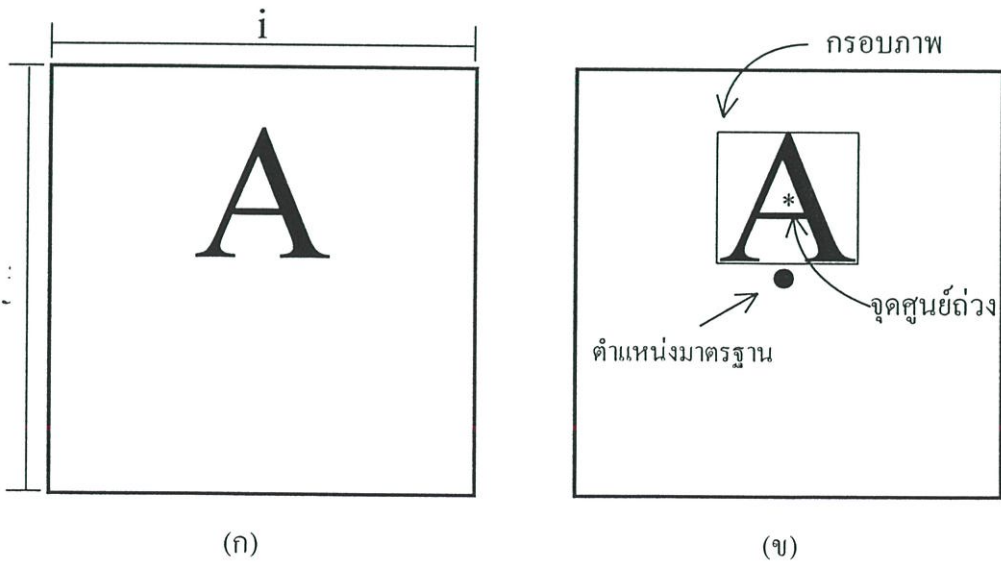
(ง) การหากรอบขวา (RB)



รูปที่ 2.5 ตำแหน่งมาตรฐาน

2.2.1 จุดศูนย์กลางถ่วง (Center of Gravity)

จุดศูนย์กลางถ่วงจะแตกต่างกับจุดกึ่งกลางภาพ โดยจุดศูนย์กลางถ่วงเป็นการนำเนื้อหาภาพซึ่งในที่นี้คือจุดดำมาเป็นตัวแปรสำคัญในการพิจารณาด้วย ซึ่งแตกต่างกับจุดกึ่งกลางภาพดังรูปที่ 2.6



รูปที่ 2.6 ตำแหน่งมาตรฐาน

(ก) ภาพอินพุต

(ข) กรอบ ตำแหน่งมาตรฐานและจุดศูนย์กลางถ่วง

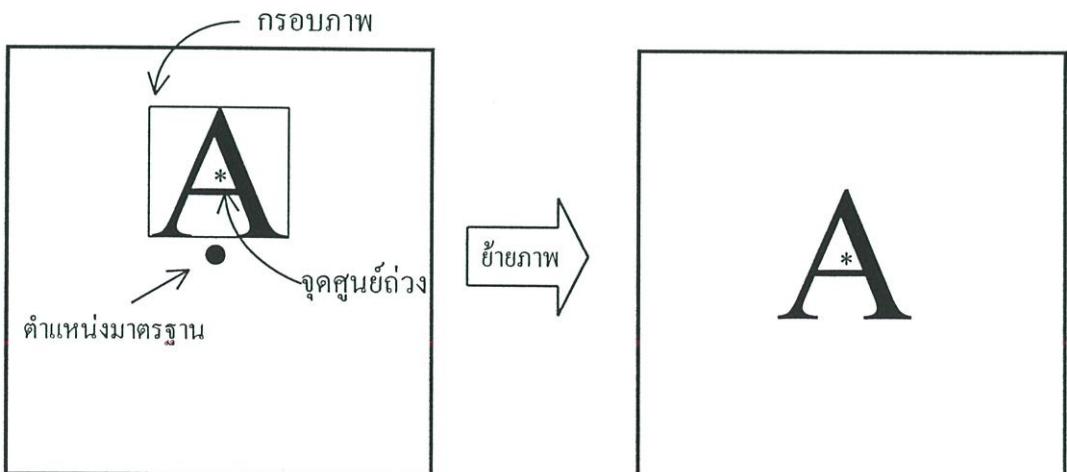
$$I_x = \frac{\sum_{i=LB}^{RB} \sum_{j=UB}^{DB} i f(i,j)}{\sum_{i=LB}^{RB} \sum_{j=UB}^{DB} f(i,j)} \quad (2.3)$$

$$I_y = \frac{\sum_{i=LB}^{RB} \sum_{j=UB}^{DB} j f(i,j)}{\sum_{i=LB}^{RB} \sum_{j=UB}^{DB} f(i,j)} \quad (2.4)$$

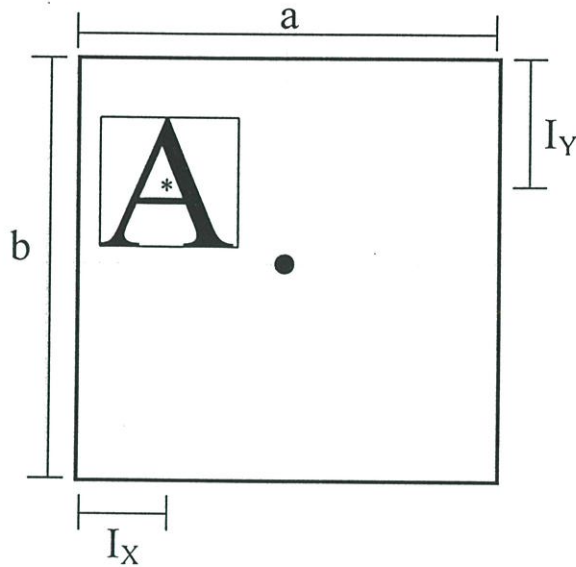
เมื่อ LB=กรอบซ้าย RB=กรอบขวา
 UB=กรอบบน DB=กรอบล่าง
 I_x =จุดศูนย์กลางถ่วงตามแนวแกนนอน
 I_y =จุดศูนย์กลางถ่วงตามแนวแกนตั้ง
 $f(i,j)$ =ฟังก์ชันจุดภาพ

2.2.1 การย้ายภาพ

เมื่อได้ตำแหน่งของจุดศูนย์กลางถ่วงแล้วขั้นตอนต่อไปต้องทำการย้ายภาพให้จุดศูนย์กลางถ่วงทับกับตำแหน่งมาตรฐานที่ได้กำหนดไว้โดยการย้ายภาพเป็นไปตามสมการ(2.5)



รูปที่ 2.7 การย้ายภาพ



รูปที่ 2.8 ตัวแปรที่สำคัญในกระบวนการย้ายภาพ

$$f'(i,j) = f\left[i + \left(\frac{a}{2} - I_X\right), j + \left(\frac{b}{2} - I_Y\right)\right] \quad (2.5)$$

เมื่อ $f'(i,j)$ = ภาพใหม่

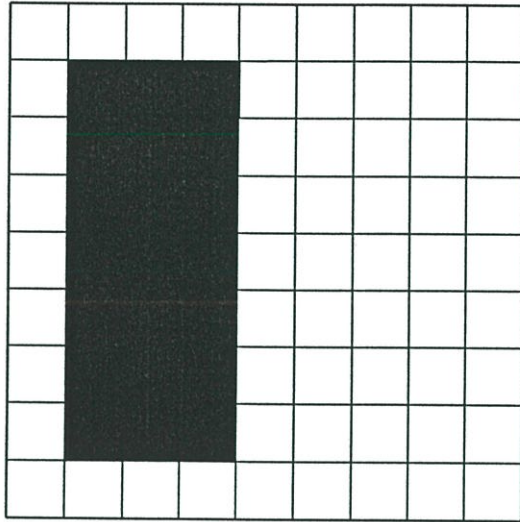
i = ตำแหน่งใดๆในแกนอนมีค่า $LB \leq i \leq RB$

j = ตำแหน่งใดๆในแกนตั้งมีค่า $UB \leq j \leq DB$

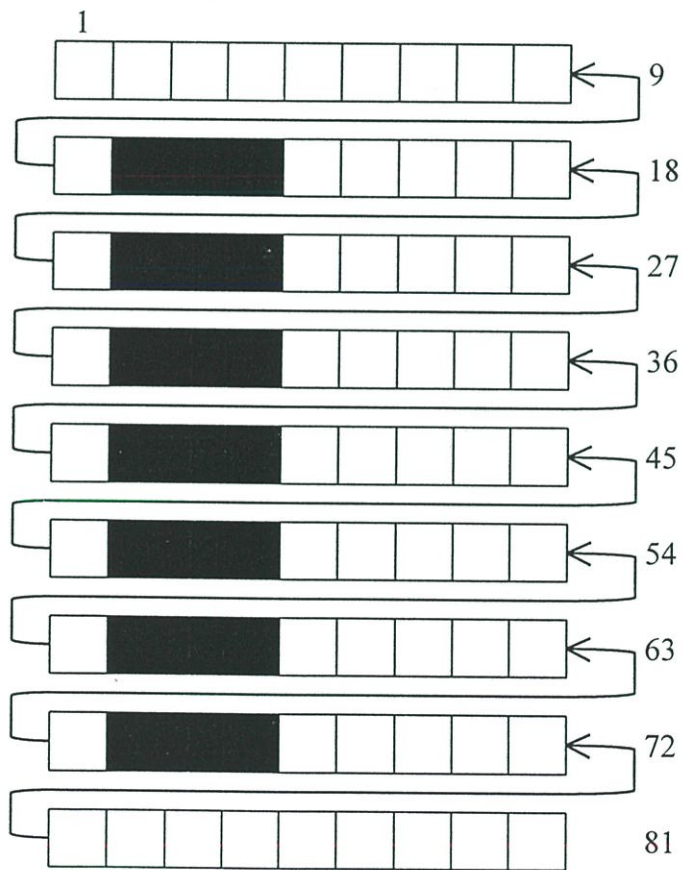
กระบวนการย้ายภาพเป็นวิธีการของคอมพิวเตอร์กราฟฟิก ที่ใช้ในการย้ายภาพใดๆแต่ในที่นี้ใช้ในการย้ายภาพสองระดับเพื่อปรับตำแหน่งของภาพ ให้อยู่ในตำแหน่งมาตรฐานที่ได้กำหนดไว้คือตรงกลางของภาพซึ่งเป็นจุดค่าในรูปที่ 2.8 หัวข้อถัดไปจะกล่าวถึงการนำภาพที่ปรับแต่งแล้วไปทำการแปลงข้อมูลเป็นแบบมิตติเดียวเพื่อนำเข้าสู่โครงข่ายประสาทเทียมต่อไป

2.3 ข้อมูล 1 มิติ

ในกระบวนการสุดท้ายของการเตรียมข้อมูลต้องทำการเปลี่ยนข้อมูลภาพแบบ 2 มิติไปเป็นข้อมูลแบบมิตติเดียวก่อนนำเข้าสู่โครงข่ายประสาทเทียม อย่างไรก็ตามการเปลี่ยนข้อมูลเป็นมิตติเดียวอาจไม่จำเป็นเสมอไปในการออกแบบ โดยจะขึ้นอยู่กับลักษณะของโครงข่ายประสาทเทียมที่ได้ออกแบบไว้ และข้อกำหนดของโครงข่ายประสาทเทียมแบบที่เลือกใช้ วิธีการเปลี่ยนข้อมูลภาพแบบ 2 มิติ แสดงดังรูปที่ 2.9 วิธีการแปลงให้ทำการอ่านข้อมูลจากภาพ 2 มิติ ณ ตำแหน่งที่ 1 จากนั้นตรวจสอบว่าข้อมูลที่อ่านมาได้เป็นข้อมูลภาพสีใด แล้วจึงนำไปแปลงเป็น เมทริกซ์ขนาด 1 มิติตามข้อมูลภาพที่อ่านได้โดยสีดำให้มีค่า 1 แต่สีขาวให้มีค่า 0



(ก)



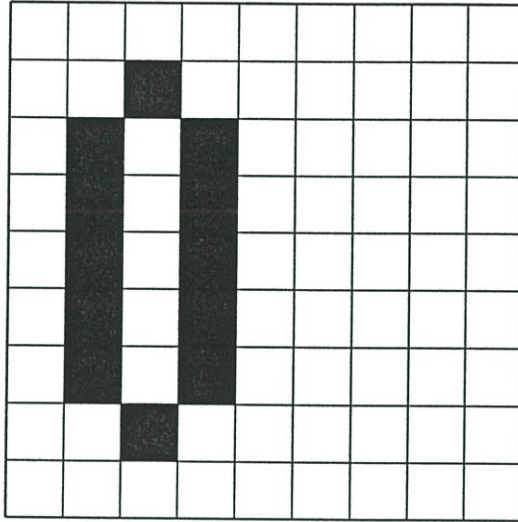
(ข)

รูปที่ 2.9 การแปลงภาพ 2 มิติ ไปเป็นข้อมูล 1 มิติ

(ก) ภาพ 2 มิติ

(ข) ข้อมูล 1 มิติ

ตัวอย่างที่ 2.1 จากภาพ 2 ระดับที่กำหนดให้จงปรับให้อยู่ตำแหน่งมาตรฐาน



จากการหาดำแหน่งมาตรฐานในรูปที่ 2.3 ต้องทำการหา

กรอบของภาพ $UB=2$

$DB=8$

$LB=2$

$RB=4$

จุดศูนย์กลาง แทนค่าในสมการที่(2.1) ได้

$$\begin{aligned}
 I_x &= \frac{\sum_{i=2}^4 \sum_{j=2}^8 i f(i,j)}{\sum_{i=2}^4 \sum_{j=2}^8 f(i,j)} \\
 &= \frac{3+2+4+2+4+2+4+2+4+2+4+2+4+3}{12} \\
 &= \frac{36}{12} \\
 &= 3
 \end{aligned}$$

$$I_Y = \frac{\sum_{i=2}^4 \sum_{j=2}^8 j f(i,j)}{\sum_{i=2}^4 \sum_{j=2}^8 f(i,j)}$$

$$= \frac{2+3+3+4+4+5+5+6+6+7+7+8}{12}$$

$$= \frac{60}{12}$$

$$= 5$$

จุดศูนย์กลางของภาพนี้คือ (3,5)

ทำการย้ายภาพโดยแทนค่าลงในสมการที่ 2.3 ได้

$$f'(i,j) = f\left[i + \left(\frac{9}{2} - 3\right), j + \left(\frac{9}{2} - 5\right)\right]$$

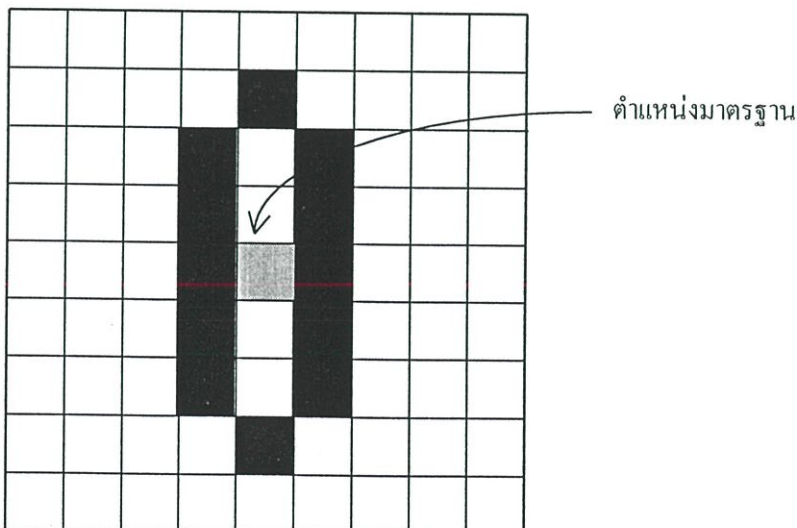
$$= f[i + (4.5 - 3), j + (4.5 - 5)]$$

$$= f[i + (5 - 3), j + (5 - 5)]$$

$$= f[i + (-2), j + 0]$$

$$= f[i - 2, j]$$

จากสมการที่ได้นำไปแทนค่า $LB \leq i \leq RB$ และ $UB \leq j \leq DB$ ได้ภาพที่ผ่านการปรับตำแหน่งเข้าสู่ตำแหน่งมาตรฐานดังรูป

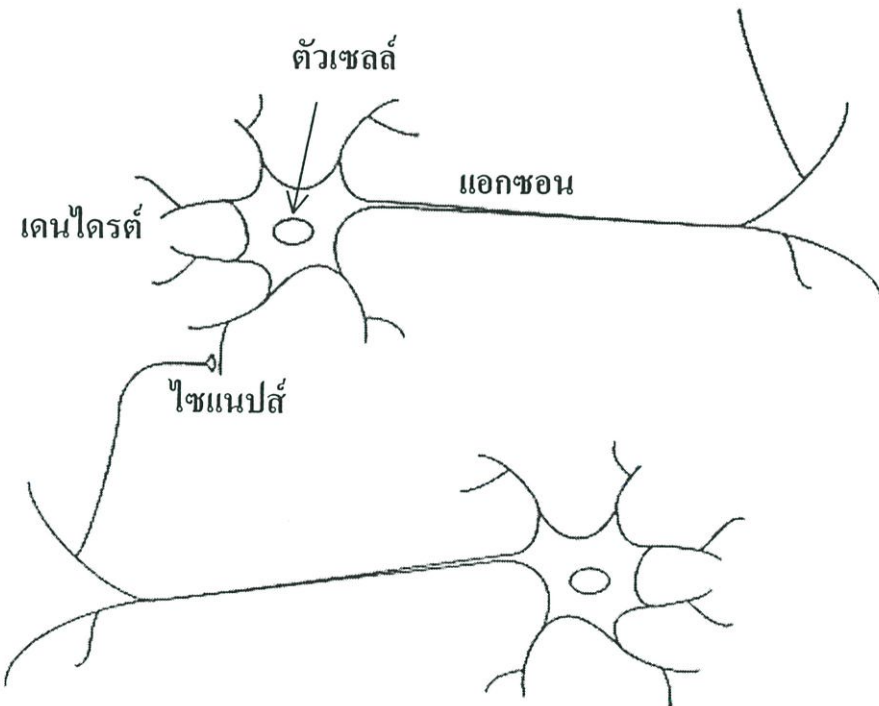


บทที่ 3

โครงข่ายประสาทเทียม

3.1 โครงข่ายประสาททางชีววิทยา

โครงข่ายประสาทเทียมชนิดต่างๆที่ใช้ในปัจจุบัน มีต้นแบบในการพัฒนามาจากโครงข่ายประสาททางชีววิทยาทั้งตัวโครงสร้างและหลักการทำงาน โครงข่ายประสาททางชีววิทยามีเซลล์ประสาท (Neuron) หลายร้อยล้านเซลล์เป็นส่วนประกอบ และมีกิ่งก้านมากมายที่ต่อกันเป็นระบบประสาท ดังในรูปที่ 3.1



รูปที่ 3.1 โครงสร้างและส่วนประกอบของเซลล์ประสาท (Neuron)

เซลล์ประสาทจัดเป็นหน่วยที่เล็กที่สุดของระบบประสาทที่สามารถทำงานได้ โดยเซลล์ประสาทมีส่วนประกอบสำคัญดังนี้

1. ตัวเซลล์ (Cell Body) มีรูปร่างหลายแบบ เช่น กลม หรือ รี ประกอบด้วยนิวเคลียสอยู่กลางเซลล์ ตัวเซลล์มีเดนไดรต์ (Dendrites) เป็นส่วนที่รับสัญญาณกระแสประสาทเข้ามาประมวลผล และมีแอกซอน (Axon) เป็นตัวส่งกระแสประสาทออก

2. เดนไดรต์ (Dendrites) เป็นกิ่งก้านที่ยื่นออกมาของตัวเซลล์มีลักษณะเป็นเส้น หรือใย เดนไดรต์อาจมีหนึ่งเส้น หรือหลายเส้น ทำหน้าที่นำกระแสประสาทไฟฟ้าเข้าสู่ตัวเซลล์

3. แอกซอน (Axon) เป็นส่วนที่ยื่นออกมาจากตัวเซลล์เพียงเส้นเดียวเท่านั้น แตกต่างกับเดนไดรต์ที่มีมากกว่าหนึ่งเส้นได้ แอกซอนทำหน้าที่นำกระแสประสาทออกจากตัวเซลล์ไปยังเซลล์ประสาทอื่น

4. ไซแนปส์ (Synapse) เป็นจุดที่ต่อระหว่าง เดนไดรต์กับตัวเซลล์ ถ้าเดนไดรต์มีหลายเส้น ไซแนปส์ก็จะมีหลายจุด อาจกล่าวได้ว่าจำนวนไซแนปส์ขึ้นอยู่กับจำนวนเดนไดรต์

3.1.1 การทำงานของเซลล์ประสาท

การทำงานของเซลล์ประสาทเป็นการนำกระแสประสาท (Nerve Impulse) หรือเป็นการส่งสัญญาณประสาทที่มีการเปลี่ยนแปลงทางไฟฟ้าเคมี (Electrochemical Reaction) ทุกส่วนของเซลล์ประสาทมีสมบัติเป็นแหล่งกำเนิดของกระแสประสาทเมื่อได้รับการกระตุ้น สิ่งกระตุ้นส่วนใหญ่อยู่ในรูปของพลังงาน เช่น อุณหภูมิ สารเคมีต่างๆ หรือ แม้กระทั่งกระแสประสาท สิ่งกระตุ้นจะไปกระตุ้นเซลล์ประสาท และมีการถ่ายทอดกระแสประสาทไปตามใยประสาทต่างๆ

กระแสประสาทจะเกิดขึ้นเมื่อเซลล์ประสาทได้รับสิ่งกระตุ้น และหากแรงกระตุ้นมากกว่าระดับเรชชโฮลด์ (Threshold Level) เซลล์ประสาทก็จะส่งกระแสประสาทออกไปยังเซลล์ประสาทอื่นที่อยู่ผ่านทางแอกซอน การกระตุ้นด้วยสิ่งเร้ามากๆ ไม่สามารถทำให้กระแสประสาทเคลื่อนที่ได้เร็วขึ้น ทั้งนี้เพราะการเคลื่อนที่ของเซลล์ประสาทต้องอาศัยพลังงานจากกระบวนการเผาผลาญของร่างกาย (Metabolism) ของเซลล์ประสาทเองโดยตรง ไม่ได้อาศัยพลังงานจากสิ่งเร้า

3.1.2 ความเร็วในการประมวลผลของเซลล์ประสาท

โดยปรกติแล้ว การส่งสัญญาณกระแสประสาทระหว่างเซลล์ประสาทนั้นมีความช้ามากเมื่อเทียบกับสัญญาณทางไฟฟ้าในปัจจุบัน แต่หากพิจารณาตามโครงสร้างของโครงข่ายประสาทจะพบว่า เซลล์ประสาทจะต่อถึงกันแบบขนาน (Parallel Structure) และสามารถประมวลผลแบบขนาน (Parallel Processing) ด้วยการประมวลผลแบบขนานนี้เองโครงข่ายประสาทที่มีอยู่ในสิ่งมีชีวิตจึงทำงานได้เร็ว และมากกว่าการทำงานของคอมพิวเตอร์มาก ในด้านการจดจำ (Recognition) การเรียนรู้ (Learning)

3.1.3 การเรียนรู้ของโครงข่ายประสาท

ตามธรรมชาติของโครงข่ายประสาทที่มีอยู่ในสิ่งมีชีวิตต่างๆ ต้องมีการเปลี่ยนแปลงตลอดเวลา ทั้งการสร้างใหม่ และการเสื่อมสภาพ หรือแม้กระทั่งเกิดการเชื่อมต่อใหม่ๆระหว่างเซลล์ประสาท โดยตัวที่ต่อเชื่อมก็คือเดนไดรต์นั่นเอง การพัฒนาความแข็งแรงของไซแนปส์ซึ่งเป็นจุดต่อระหว่างตัวเซลล์กับเดนไดรต์ก็จะทำให้เกิดการเปลี่ยนแปลงภายในตัวเซลล์ด้วย นั่นหมายถึงการได้เรียนรู้ประสบการณ์ และความรู้ใหม่ๆ ของโครงข่ายประสาท เช่น การที่เด็กแรกเกิดจำหน้าผู้เป็นมารดาได้ กระบวนการพัฒนาความสามารถของสิ่งมีชีวิตนั้นอยู่ที่สมองเป็นหลัก ดังนั้นหากโครงข่ายประสาทที่ถูกใช้งานบ่อย ก็จะมีความแข็งแรงและมีจุดเชื่อมต่อถึงกันระหว่างเซลล์ประสาทซับซ้อนมากขึ้น ในทางกลับกันโครงข่ายที่ไม่ได้ใช้งานจะมีเซลล์ประสาทย่อยและไม่ซับซ้อน

3.2 โครงข่ายประสาทเทียม

3.2.1 ความเป็นมาของโครงข่ายประสาทเทียม

ความเจริญก้าวหน้าของวิชาประสาทวิทยาได้ยอมรับงานวิจัย ในการสร้างแบบจำลองทางคณิตศาสตร์ของเซลล์ประสาทซึ่งเลียนแบบพฤติกรรมของระบบประสาท แนวความคิดนี้ย้อนหลังไปเริ่มที่ปี ค.ศ. 1970 เมื่อแบบจำลองแบบแรกของระบบประสาทเป็นการเริ่มต้นโดย McCulloch และ Pitts (1943) ต่อมา Hebb (1949) ได้ปรับปรุงกฎของการเรียนรู้ให้มีประสิทธิภาพมากขึ้น และทำให้งานวิจัยอื่นๆปฏิบัติตามแนวความคิดนี้มาตลอด 20 ปี เช่น Minsky (1954) และ Rosenblatt (1958) Rosenblatt มีชื่อเสียงกับอัลกอริทึมการเรียนรู้ของเพอร์เซปตรอน (Perceptron) ในขณะเดียวกัน Widrow และ Hoff ได้พัฒนาการเรียนรู้ที่สำคัญของ เพอร์เซปตรอน เป็น Widrow-Hoff

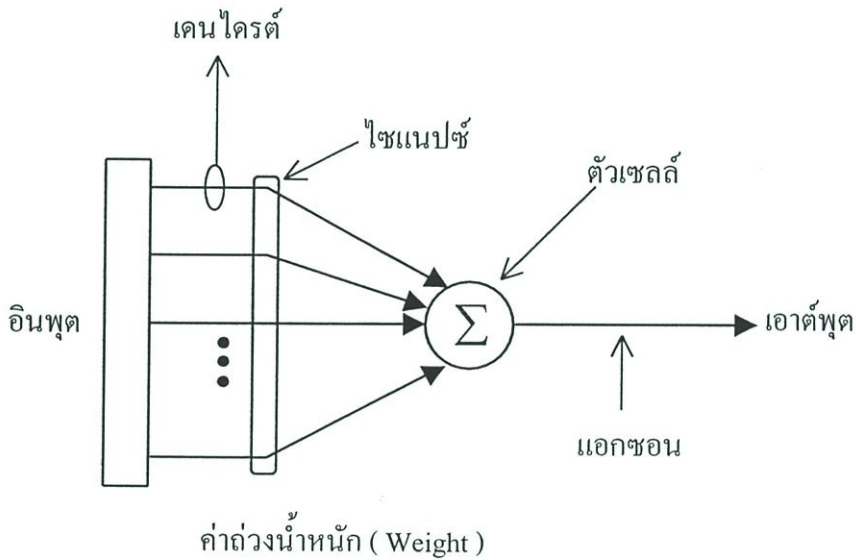
ต่อมา Minsky และ Papert (1969) ได้ชี้ข้อจำกัดของทฤษฎีของแบบจำลองเครือข่ายประสาทเทียมแบบชั้นเดียวในหนังสือ Perceptron ของพวกเขา ทำให้งานวิจัยทางด้านโครงข่ายประสาทเทียมถูกทิ้งไปประมาณ 20 ปี โดยไม่มีใครสนใจ แต่นักวิจัยบางคนก็ยังคงทำการวิจัยกันอย่างต่อเนื่อง สำหรับตัวอย่างนักวิจัยที่ยังคงทำการวิจัยอยู่ เช่น Anderson (1970) และ Grossberg (1977) ทำงานสำคัญบนแบบจำลองในทางจิตศาสตร์Kohonen(1972)พัฒนาแบบจำลอง Associative Memory

ตอนต้นปี 1980 โครงข่ายประสาทเทียมใกล้จะกลับมาได้รับความสนใจขึ้นใหม่ทั้งนี้เพราะ Hopfield (1982) ได้เสนอแนวความคิดของพลังงานที่น้อยที่สุดในทางฟิสิกส์ เข้าในโครงข่ายประสาทเทียม บทความของเขามีอิทธิพลให้เทคโนโลยีโครงข่ายประสาทเทียมกลับมาได้รับความสนใจใหม่ด้วยการเพิ่มโมเมนตัม (Momentum) ต่อมา Feldman และ Ballard (1982) ได้ทำในส่วนของการเชื่อมต่อที่สัมพันธ์กันซึ่งเป็นที่นิยม และเป็นพื้นฐานในงานวิจัยโครงข่ายประสาทเทียม

ในกลางปี 1980 หนังสือ Parallel Distributed Processing ที่เขียนโดย Rumelhart และ McClelland (1986) ทำให้เกิดผลกระทบที่ยิ่งใหญ่บนคอมพิวเตอร์ ความรู้ด้านวิทยาศาสตร์ชีวภาพ อัลกอริทึมการเรียนรู้ของ แบคพรอพพาเกชัน (Backpropagation) ที่พัฒนาโดย Rumelhart, Hinton และ Willuams (1986) เสนอการแก้ปัญหาที่มีสมรรถภาพสูงในการฝึกสอนโครงข่ายประสาทเทียมแบบหลายชั้น และทำให้สิ่งเลวร้ายที่มีกับเพอร์เซปตรอนหายไป ความสำเร็จที่น่าตื่นเต้นของการเริ่มต้นนี้เป็นการทดลองให้เห็นจริงโดยระบบ NETalk ที่พัฒนาร่วมกันโดยนักวิทยาศาสตร์ 2 คน คือ Sejnowski และ Rosenberg (1987) เป็นระบบเปลี่ยนตัวอักษรภาษาอังกฤษไปเป็นเสียงพูดที่ฉลาดอย่างน่าชื่นชม อย่างไรก็ตามแนวความคิดนี้ของแบคพรอพพาเกชัน ยังถูกพัฒนาโดย Werbos (1974) และ Parker (1982) อย่างอิสระอีกด้วย ในปัจจุบันนักวิจัยยังคงวิจัยคิดค้นแบบจำลองทางโครงข่ายประสาทใหม่ๆขึ้นมาอีก เพื่อลดเวลาการเรียนรู้ และเพิ่มประสิทธิภาพให้ดีขึ้น เพื่อใช้เป็นเครื่องมือในการแก้ปัญหาทางคณิตศาสตร์ที่ซับซ้อนมากขึ้น (สุขสถิต สุขใจ . 2540 : 5)

3.2.2 เซลล์ประสาทเทียม

โครงข่ายประสาทเทียม เป็นแบบจำลอง (Model) ทางคณิตศาสตร์ที่เลียนแบบโครงข่ายประสาททางชีววิทยา เพื่อแก้ปัญหาทางคณิตศาสตร์ และวิศวกรรม ดังนั้นโครงข่ายประสาทเทียมจึงประกอบไปด้วยเซลล์ประสาทเทียม ซึ่งเป็นหน่วยที่เล็กที่สุดของโครงข่ายประสาทที่ยังสามารถทำงานได้ เซลล์ประสาทเทียมจึงถูกจำลองให้มีลักษณะเฉพาะที่สำคัญเช่นเดียวกับเซลล์ประสาทดังรูปที่ 3.2

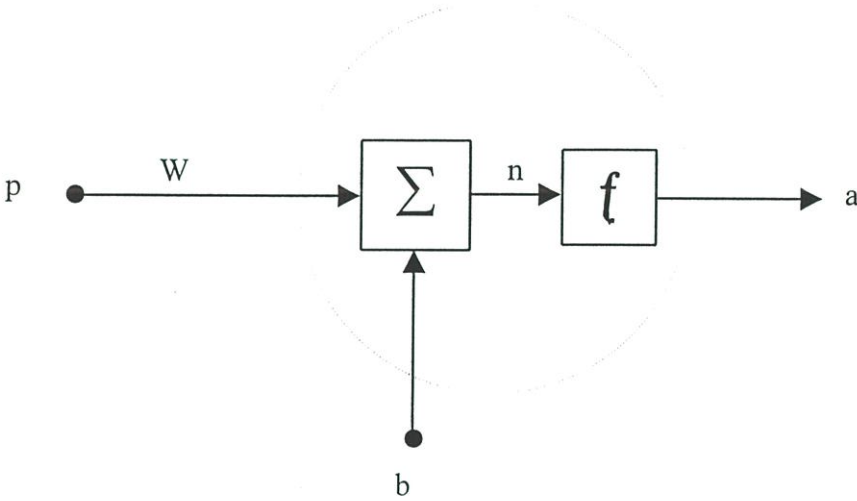


รูปที่ 3.2 เซลล์ประสาทเทียม

หากลองพิจารณารูปที่ 3.1 ประกอบกับรูปที่ 3.2 จะเห็นว่าเซลล์ประสาทเทียมมีโครงสร้างพื้นฐานคล้ายกับเซลล์ประสาททางชีววิทยา แบบจำลองในรูปที่ 3.2 เป็นแบบจำลองของโครงข่ายประสาทเทียมเกือบทั้งหมดใช้วิธีการนี้เป็นพื้นฐาน โดยอินพุตที่เป็นทางเข้าของสัญญาณ หรือข้อมูลที่นำมาประมวลผลถูกต่อด้วยกิ่งก้านของเดนไดรต์ ผ่านค่าถ่วงน้ำหนักที่สัมพันธ์กันซึ่งเปรียบเสมือนความแข็งแรงของไซแนปส์ ค่าที่ได้จากการถ่วงน้ำหนักถูกส่งต่อเข้าตัวเซลล์เพื่อรวบรวมและกำหนดระดับของการกระตุ้นของเซลล์ประสาท ผลของกระบวนการถูกส่งออกไปยังเอาต์พุตที่เปรียบได้กับแอกซอน

เพื่อให้เห็นการทำงานของโครงข่ายประสาทเทียมได้ง่าย แบบจำลองเซลล์ประสาทเทียมแบบเซลล์เดียว จัดเป็นแบบจำลองที่ง่ายที่สุดในการอธิบายการทำงานของโครงข่ายประสาทเทียม ซึ่งแบบจำลองนี้มีการใช้ฟังก์ชันทางคณิตศาสตร์มาช่วยในการคำนวณ และสามารถเขียนแทนได้ด้วยเมทริกซ์ การตัดสินใจใช้การแทนค่าในทรานส์เฟอร์ฟังก์ชันซึ่งมีอยู่ด้วยกันหลายแบบสามารถเลือกใช้ให้เหมาะสมกับงานแต่ละงานได้ ดังจะกล่าวในหัวข้อถัดไป

3.2.3 เซลล์ประสาทเทียมแบบเซลล์เดียวอินพุตเดียว (Single-Input Neuron)



รูปที่ 3.3 เซลล์ประสาทเทียมแบบเซลล์เดียว อินพุตเดียว

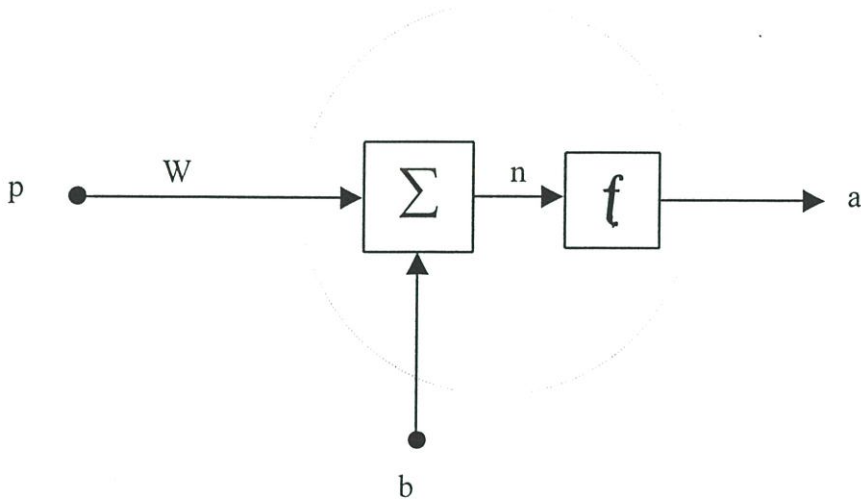
$$a = f(wp + b) \tag{3.1}$$

- เมื่อ p = สการ์ลาอินพุต (Scalar Input) b = ไบอัส (Bias or Offset)
- W = ค่าถ่วงน้ำหนัก (Weight) a = เอาต์พุต (Output)
- f = แอคติเวชันฟังก์ชัน (Activation Function) หรือ
ทรานส์เฟอร์ฟังก์ชัน (Transfer Function)
- $n = wp + a$

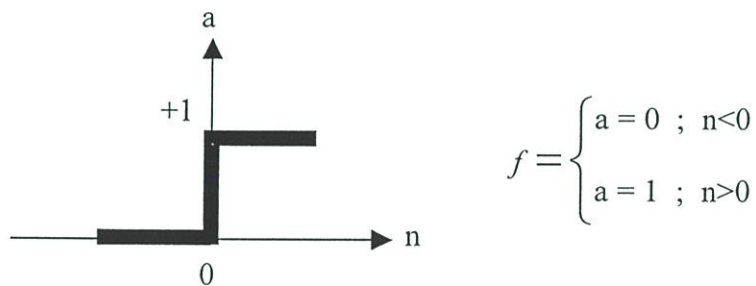
เซลล์ประสาทเทียมแบบเซลล์เดียวอินพุตเดียว ในรูปที่ 3.3 มีค่า p เป็นอินพุตซึ่งเป็นค่าสการ์ลา (เนื่องจากมีอินพุตเดียว) ค่า p จะถูกคูณด้วยค่าน้ำหนัก (w) ก่อนไปรวมกับค่า b หรือค่าออฟเซตใน Σ ผลที่ได้จากการรวมกันเป็นค่า n ซึ่งต้องนำเข้าไปใน f ซึ่งเป็นกระบวนการการตัดสินใจสุดท้ายก่อนที่จะให้ผลลัพธ์ a ออกมา

เซลล์ประสาทเทียมแบบเซลล์เดียวอินพุตเดียวเป็นเซลล์ประสาทขั้นพื้นฐานที่สุด และง่ายที่สุดในการทำความเข้าใจการบวนการทำงานของโครงข่ายประสาทเทียม ตัวอย่างที่ 3.1 เป็นการยกตัวอย่างการคำนวณแบบง่ายๆ โดยเลือกใช้ทรานส์เฟอร์ฟังก์ชันแบบฮาร์ดลิมิต ซึ่งเป็นทรานส์เฟอร์ฟังก์ชันที่ไม่ซับซ้อนในการคำนวณ

ตัวอย่างที่ 3.1 จงหาค่า a ของเซลล์ประสาทเทียมแบบเซลล์เดียวอินพุตเดียว
เมื่อ $w = 3, p = 2, b = -1.5$



เมื่อกำหนดทรานส์เฟอร์แบบฮาร์ดลิมิต (Hard Limit) หรือสเตปฟังก์ชัน (Step Function)
ดังรูปที่ 3.4



รูปที่ 3.4 ทรานส์เฟอร์ฟังก์ชันแบบฮาร์ดลิมิต

จากสมการที่ 3.1 แทนค่าลงในสมการได้

$$a = f(2 \cdot 3 - 1.5)$$

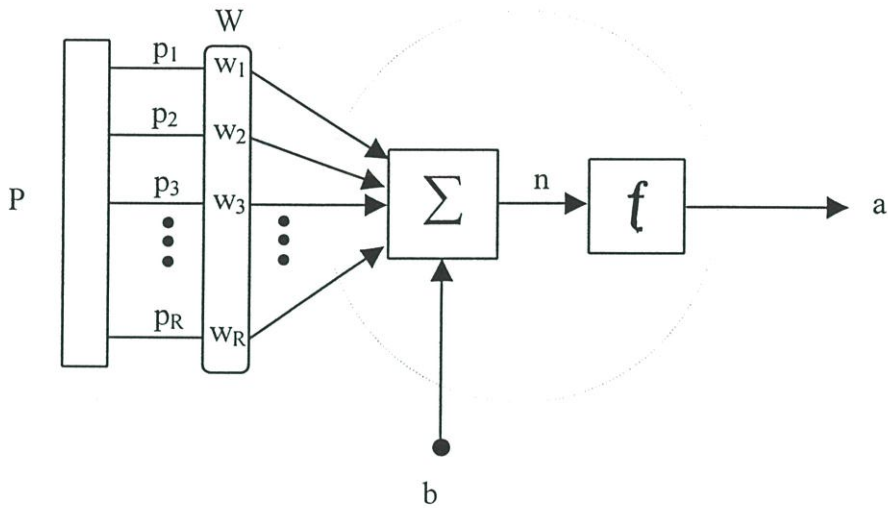
$$a = f(4.5)$$

จากทรานส์เฟอร์ฟังก์ชัน ค่า $n > 0$

คำตอบที่ได้คือ $a = 1$

ในการใช้งานส่วนใหญ่ของโครงข่ายประสาทเทียมต้องมีอินพุตมากมายหลายอินพุต โดยสามารถพัฒนาเซลล์ประสาทเทียมแบบเซลล์เดียวอินพุตเดียวไปเป็น โครงสร้างที่ซับซ้อนขึ้นได้ เป็นเซลล์ประสาทเทียมแบบเซลล์เดียวหลายอินพุตดังในหัวข้อถัดไป

3.2.4 เซลล์ประสาทเทียมแบบเซลล์เดียวหลายอินพุต (Multiple-Input Neuron)



รูปที่ 3.5 เซลล์ประสาทเทียมแบบเซลล์เดียวหลายอินพุต

$$\text{เมื่อ } \mathbf{p} = [p_1 \ p_2 \ p_3 \ \dots \ p_R] \quad (3.2)$$

$$\mathbf{W} = [w_1 \ w_2 \ w_3 \ \dots \ w_R] \quad (3.3)$$

$$n = p_1 w_1 + p_2 w_2 + p_3 w_3 + \dots + p_R w_R + b \quad (3.4)$$

หรือเขียนได้ในรูปเวกเตอร์

$$n = \mathbf{Wp} + b$$


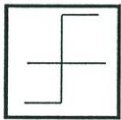


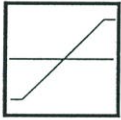




ดังนั้น

$$a = f(\mathbf{Wp} + b)$$

เซลล์ประสาทเทียมแบบนี้มีความซับซ้อนมากขึ้นกว่าแบบก่อนหน้า โดยโครงสร้างนี้มีอินพุตหลายแขนง อินพุตแต่ละเส้นจะต่อกับค่าถ่วงน้ำหนัก โดยที่ค่าถ่วงน้ำหนักแต่ละตำแหน่งอาจมีค่าเท่ากันหรือไม่ก็ได้ ภายหลังจากที่อินพุตคูณกับค่าน้ำหนักแล้วจึงผ่านเข้าไปใน Σ โดยค่าที่มีผลอีกค่าหนึ่งในการรวมค่าคือค่าไบอัส ภายหลังจากการรวมกันทั้งหมดเป็นค่า n ค่านี้ถูกป้อนเข้าไปใน f เพื่อเป็นการตัดสินใจครั้งสุดท้ายด้วยทรานส์เฟอร์ฟังก์ชัน เซลล์ประสาทเทียมแบบนี้ใช้ในการแก้ปัญหาแบบง่าย ๆ ไม่ซับซ้อน

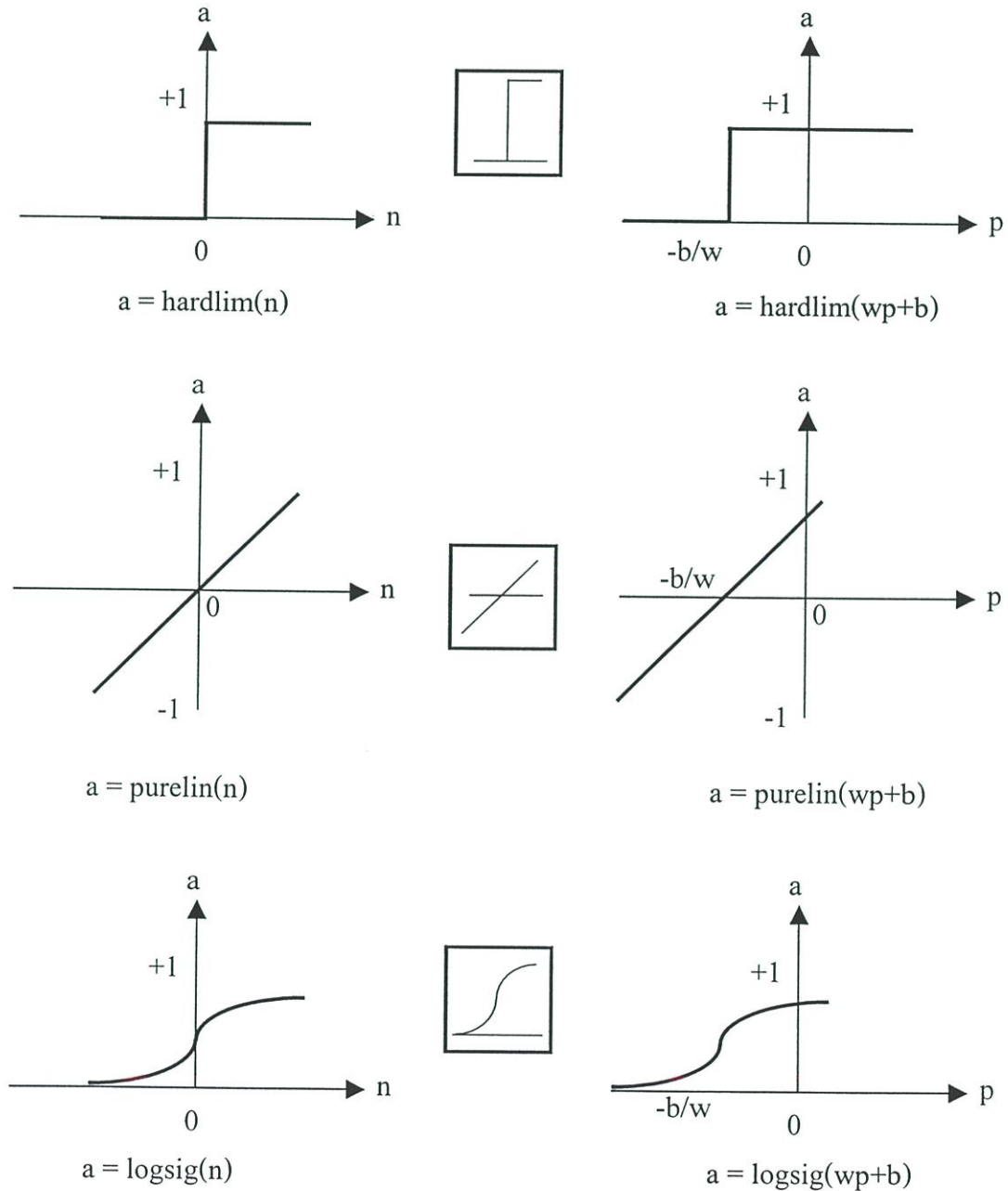
การเลือกใช้ทรานส์เฟอร์ฟังก์ชันในการตัดสินใจนั้น ต้องพิจารณาลักษณะสมบัติของข้อมูลอินพุต และเอาต์พุตที่ต้องการเพื่อให้เหมาะสมกับการใช้งาน ทรานส์เฟอร์ฟังก์ชันนั้นมีอยู่ด้วยกันหลายแบบดังที่แสดงไว้ในตารางที่ 3.1

ตารางที่ 3.1 ทรานส์เฟอร์ฟังก์ชันแบบต่างๆ

ชื่อฟังก์ชัน	อินพุต/เอาต์พุต	สัญลักษณ์	MATLAB Function
Hard Limit	$a = 0 \quad n < 0$ $a = 1 \quad n \geq 0$		hardlim
Symmetrical Hard Limit	$a = -1 \quad n < 0$ $a = +1 \quad n \geq 0$		hardlims
Linear	$a = n$		purelin
Saturating Linear	$a = 0 \quad n < 0$ $a = n \quad 0 \leq n < 1$ $a = 1 \quad n > 1$		satlin
Symmetric Saturating Linear	$a = -1 \quad n < -1$ $a = n \quad -1 \leq n < 1$ $a = 1 \quad n > 1$		satlins
Log-Sigmoid	$a = \frac{1}{1 + e^{-n}}$		logsig
Hyperbolic Tangent Sigmoid	$a = \frac{e^n - e^{-n}}{e^n + e^{-n}}$		tansig
Positive Linear	$a = 0 \quad n < 0$ $a = n \quad 0 \leq n$		poslin
Competitive	$a = 1$ neuron with max n $a = 0$ all other neurons		compet

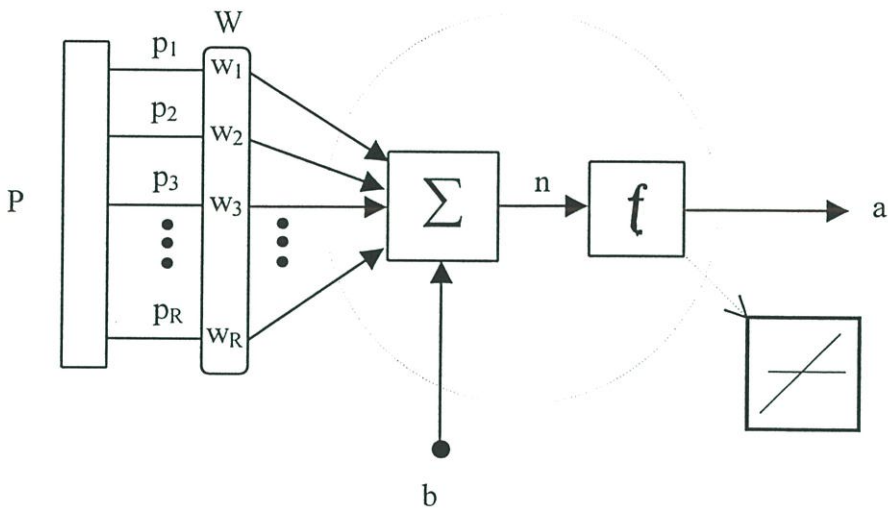
3.2.5 ทรานส์เฟอร์ฟังก์ชัน (Transfer Function)

ทรานส์เฟอร์ฟังก์ชันที่ใช้ในโครงข่ายประสาทเทียมมีอยู่ด้วยกันหลายแบบ ทั้งฟังก์ชันที่เป็นเชิงเส้น (Linear Function) และไม่เป็นเชิงเส้น (Nonlinear Function) การเลือกใช้ก็ขึ้นอยู่กับลักษณะของงาน ค่าที่มีผลต่อทรานส์เฟอร์ฟังก์ชันที่สำคัญอีกค่าหนึ่งคือค่าไบอัส(b)หรือค่าออฟเซต (Offset) ค่านี้จะทำให้ทรานส์เฟอร์ฟังก์ชันเลื่อนตำแหน่งได้ทางแกนนอน (Horizontal Shift) ดังแสดงไว้ในรูปที่ 3.6



รูปที่ 3.6 ทรานส์เฟอร์ฟังก์ชันที่ถูกเลื่อนในแนวนอนตามค่าไบอัส

ตัวอย่างที่ 3.2 จงหาค่า a ของเซลล์ประสาทเทียมแบบเซลล์เดียวหลายอินพุต



กำหนดให้

$$p = [2 \ 4 \ 0 \ 1 \ 5 \ 3]$$

$$w = [1 \ 0.5 \ 0 \ 1 \ 0.1 \ 1]$$

$$b = 2$$

ทรานส์เฟอร์ฟังก์ชันแบบเชิงเส้น $a = f(n)$ เมื่อ $a = n$

แทนค่าลงในสมการ (3.4)

$$n = 2 + 2 + 0 + 1 + 0.5 + 3$$

$$n = 8.5$$

จาก $a = f(Wp + b)$

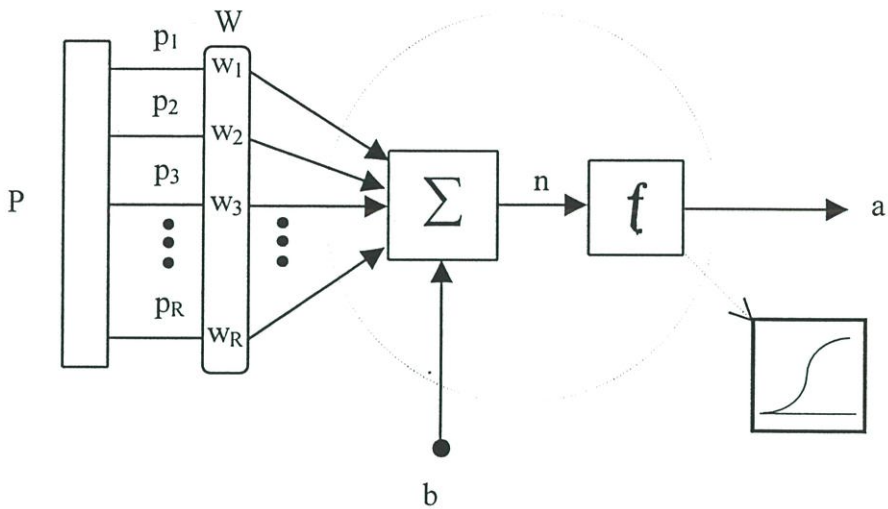
และ $n = Wp + b$

ดังนั้น $a = f(8.5)$

คำตอบที่ได้คือ $a = 8.5$

ทรานส์เฟอร์ฟังก์ชันแบบเชิงเส้นดังในตัวอย่างที่ 3.2 มีใช้ใน โครงข่ายประสาทเทียมแบบ ADALINE และ Adaptive Filter ADALINE ส่วนตัวอย่างที่ 3.3 เป็นตัวอย่างที่ใช้ทรานส์เฟอร์ฟังก์ชันแบบลอก-ซิกมอยด์ในการตัดสินใจ ซึ่งทรานส์เฟอร์ฟังก์ชันแบบนี้เป็นทรานส์เฟอร์ฟังก์ชันอีกแบบที่นิยมใช้ในปัจจุบัน

ตัวอย่างที่ 3.3 จงหาค่า a ของเซลล์ประสาทเทียมแบบเซลล์เดียวหลายอินพุต



กำหนดให้

$$p = [1 \ 0 \ 2 \ 3]$$

$$w = [2 \ 0.5 \ 0.5 \ 1]$$

$$b = 2$$

ทรานส์เฟอร์ฟังก์ชันแบบลอก-ซิกมอยด์ $a = \frac{1}{1 + e^{-n}}$

แทนค่าลงในสมการ (3.4)

$$n = 2 + 0 + 1 + 3$$

$$n = 6$$

จาก $a = f(Wp + b)$

และ $n = Wp + b$

ดังนั้น $a = f(6)$

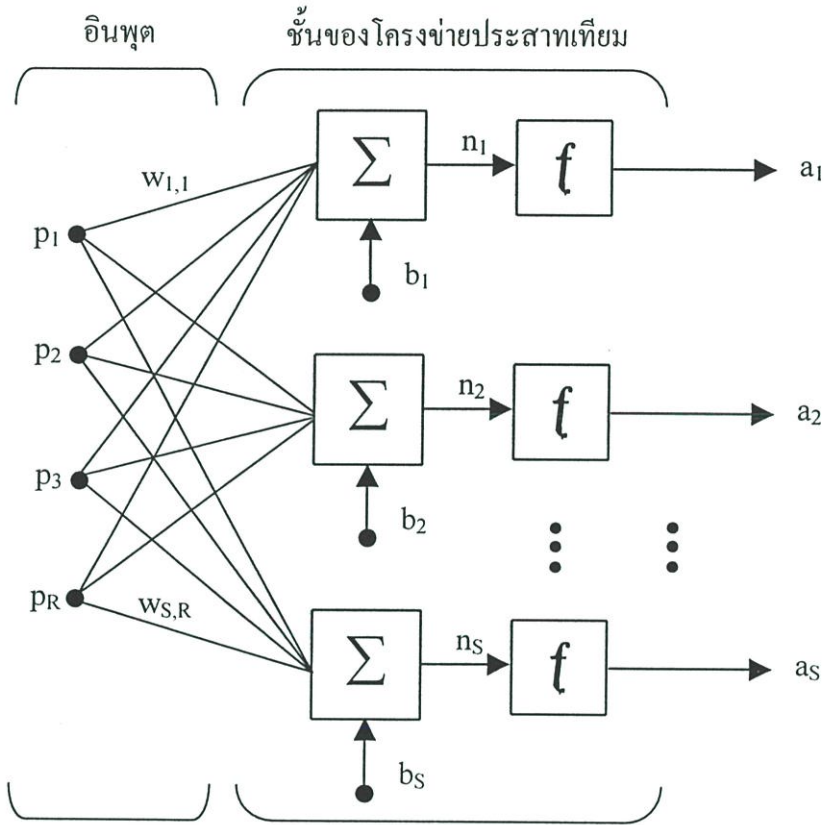
$$a = \frac{1}{1 + e^{-6}}$$

คำตอบที่ได้คือ $a = 0.99752$

ทรานส์เฟอร์ฟังก์ชันแบบลอก-ซิกมอยด์ (Log-Sigmoid) ในตัวอย่างที่ 3.3 มีความซับซ้อนกว่าฟังก์ชันเชิงเส้นในตัวอย่างที่ 3.2 ทรานส์เฟอร์ฟังก์ชันแบบนี้นิยมใช้กันอย่างแพร่หลายในโครงข่ายประสาทเทียมแบบหลายชั้นโดยใช้การเรียนรู้แบบแบคพรอพพาเกชัน ซึ่งจะกล่าวรายละเอียดอีกครั้งในหัวข้อ 3.2.7.2

3.2.6 โครงข่ายประสาทเทียมแบบชั้นเดียว (Single Layer of Neuron Network)

โดยปกติแล้วเซลล์ประสาทเทียมแบบเซลล์เดียว ไม่สามารถใช้แก้ปัญหาที่ซับซ้อนได้ ส่วนใหญ่ต้องใช้เซลล์ประสาทเทียมที่มากกว่าหนึ่งเซลล์และมีอินพุตหลายอินพุต โดยลักษณะการต่อเป็น โครงสร้างแบบขนาน หรือเรียกลักษณะการต่อแบบนี้ว่า “ชั้น (Layer)”



รูปที่ 3.7 โครงข่ายประสาทเทียมแบบชั้นเดียว

$$\mathbf{p} = [p_1 \ p_2 \ p_3 \ \dots \ p_R] \quad (3.5)$$

$$\mathbf{b} = [b_1 \ b_2 \ b_3 \ \dots \ b_R] \quad (3.6)$$

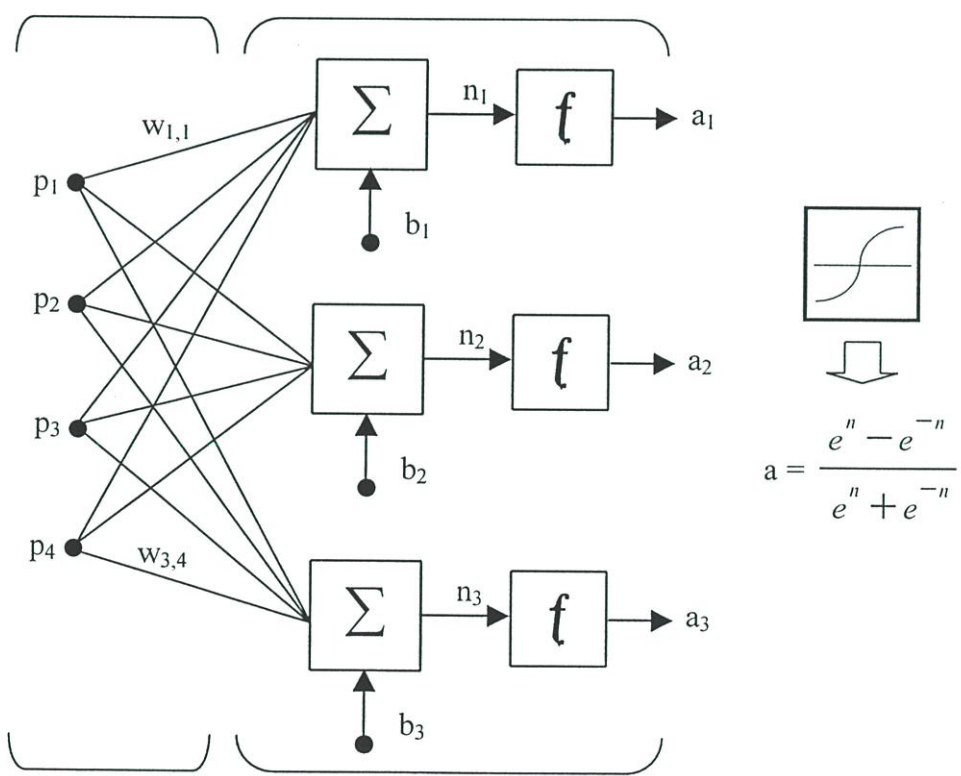
$$\mathbf{W} = \begin{bmatrix} w_{1,1} & w_{1,2} & w_{1,3} & \dots & w_{1,R} \\ w_{2,1} & w_{2,2} & w_{2,3} & \dots & w_{2,R} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ w_{S,1} & w_{S,2} & w_{S,3} & \dots & w_{S,R} \end{bmatrix} \quad (3.7)$$

ดังนั้นเอาต์พุตเวกเตอร์เป็น

$$\mathbf{a} = \mathbf{f}(\mathbf{W}\mathbf{p} + \mathbf{b}) \quad (3.8)$$

เนื่องจากว่าโครงข่ายประสาทเทียมแบบชั้นเดียวมีเซลล์ประสาทเทียมหลายตัวดังนั้นค่าถ่วงน้ำหนักที่ต้องใช้จะมีจำนวนที่แปรตามค่าอินพุตเวกเตอร์ \mathbf{p} และจำนวนเซลล์ประสาทเทียม (S) ดังสมการที่ 3.7 เซลล์แต่ละเซลล์ในโครงข่ายประสาทเทียมมีค่าไบอัสของตัวเองซึ่งค่าไบอัสในแต่ละเซลล์อาจมีค่าเท่ากันหรือไม่ก็ได้ จำนวนของค่าไบอัสเท่ากับจำนวนเซลล์ประสาทเทียม(S) การรวมกันของค่าต่างๆถูกรวมใน Σ จากนั้นจะได้ค่า n ที่ถูกป้อนเข้าสู่ทรานส์เฟอร์ฟังก์ชัน เซลล์ประสาทเทียมแต่ละเซลล์มีทรานส์เฟอร์ฟังก์ชันของตัวเอง ซึ่งอาจใช้ทรานส์เฟอร์ฟังก์ชันเหมือนกันหมดทุกเซลล์หรือไม่ก็ได้ขึ้นอยู่กับารออกแบบโครงข่ายประสาทเทียม เอาต์พุตเวกเตอร์ที่ได้คือค่า \mathbf{a} ซึ่งเป็นผลลัพธ์ของโครงข่ายประสาทเทียม

ตัวอย่างที่ 3.4 จงหาค่า \mathbf{a} ของเซลล์ประสาทเทียมแบบชั้นเดียวโดยมีเซลล์ประสาทเทียม 3 ตัว รับข้อมูลอินพุต 4 จุด และมีทรานส์เฟอร์ฟังก์ชันเหมือนกันทั้งหมด โดยใช้ทรานส์เฟอร์ฟังก์ชันแบบไฮเปอร์โบลิกแทนเจนต์ซิกมอยด์



กำหนดให้ $\mathbf{p} = [1 \ 0 \ 3 \ 7]$ $\mathbf{b} = [0.1 \ 0.1 \ 0.1]$ และ $W = \begin{bmatrix} 1 & 0.5 & 2 & 0.1 \\ 1 & 3 & 1 & 0.2 \\ 3 & 0.2 & 1 & 1 \end{bmatrix}$

จาก $\mathbf{n} = \mathbf{Wp} + \mathbf{b}$

แต่ค่าอินพุต และค่าถ่วงน้ำหนักอยู่ในรูปเวกเตอร์ ดังนั้นเมทริกซ์จึงสามารถเขียนได้อีกแบบคือ

$$\mathbf{n} = \mathbf{Wp}^T + \mathbf{b} \quad (3.9)$$

แทนค่าในสมการ(3.9) ได้

$$\begin{aligned} \mathbf{n} &= \begin{bmatrix} 1 & 0.5 & 2 & 0.1 \\ 1 & 3 & 1 & 0.2 \\ 3 & 0.2 & 1 & 1 \end{bmatrix} [1 \ 0 \ 3 \ 7]^T + [0.1 \ 0.1 \ 0.1] \\ &= \begin{bmatrix} 1 & 0.5 & 2 & 0.1 \\ 1 & 3 & 1 & 0.2 \\ 3 & 0.2 & 1 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 3 \\ 7 \end{bmatrix} + [0.1 \ 0.1 \ 0.1] \\ &= [1+0+6+0.7 \ 1+0+3+1.4 \ 3+0+3+7] + [0.1 \ 0.1 \ 0.1] \\ &= [7.8 \ 5.5 \ 13.1] \end{aligned}$$

$$n_1 = 7.8$$

$$n_2 = 5.5$$

$$n_3 = 13.1$$

แทนค่า n ใน ไฮเปอร์โบลิกแทนเจนต์ซิกมอยด์ ทรานส์เฟอร์ฟังก์ชันได้


$$\begin{aligned} \mathbf{a} &= \begin{bmatrix} \frac{e^{n_1} - e^{-n_1}}{e^{n_1} + e^{-n_1}} & \frac{e^{n_2} - e^{-n_2}}{e^{n_2} + e^{-n_2}} & \frac{e^{n_3} - e^{-n_3}}{e^{n_3} + e^{-n_3}} \end{bmatrix} \\ &= \begin{bmatrix} \frac{e^{7.8} - e^{-7.8}}{e^{7.8} + e^{-7.8}} & \frac{e^{5.5} - e^{-5.5}}{e^{5.5} + e^{-5.5}} & \frac{e^{13.1} - e^{-13.1}}{e^{13.1} + e^{-13.1}} \end{bmatrix} \\ &= [0.9999 \ 0.9999 \ 0.9999] \end{aligned}$$

ดังนั้น $a_1 = 0.9999$

$$a_2 = 0.9999$$

$$a_3 = 0.9999$$

ตัวอย่างที่ 3.4 จงหาค่า a ของเซลล์ประสาทเทียมแบบชั้นเดียวใช้ทรานส์เฟอร์ฟังก์ชัน เหมือนกันหมดโดยกำหนดให้

$$W = \begin{bmatrix} 0.2 & 0.5 & 0.2 & 0.1 \\ 0.1 & 0.3 & 0.1 & 0.4 \\ 0.6 & 0.9 & 0.8 & 0.7 \\ 0.4 & 0.2 & 0.6 & 0.2 \end{bmatrix}$$


$$a = -1 \quad n < -1$$

$$a = n \quad -1 \leq n < 1$$

$$a = 1 \quad n > 1$$

$$p = [4 \ 3 \ 2 \ 1]$$

$$b = [0.1 \ 0.1 \ 0.1 \ 0.1]$$

แทนค่าในสมการ(3.9)ได้

$$n = \begin{bmatrix} 0.2 & 0.5 & 0.2 & 0.1 \\ 0.1 & 0.3 & 0.1 & 0.4 \\ 0.6 & 0.9 & 0.8 & 0.7 \\ 0.4 & 0.2 & 0.6 & 0.2 \end{bmatrix} [4 \ 3 \ 2 \ 1]^T + [-2.3 \ -3.9 \ 0 \ -4]$$

$$= \begin{bmatrix} 0.2 & 0.5 & 0.2 & 0.1 \\ 0.1 & 0.3 & 0.1 & 0.4 \\ 0.6 & 0.9 & 0.8 & 0.7 \\ 0.4 & 0.2 & 0.6 & 0.2 \end{bmatrix} \begin{bmatrix} 4 \\ 3 \\ 2 \\ 1 \end{bmatrix} + [-2.3 \ -3.9 \ 0 \ -4]$$

$$= [3 \ 1.9 \ 7.4 \ 3.6] + [-2.3 \ -3.9 \ 0 \ -4]$$

$$= [0.7 \ -2 \ 7.4 \ -0.4]$$

แทนค่า n ใน Symmetric Saturating Linear ทรานส์เฟอร์ฟังก์ชันได้

$$a = [0.7 \ -1 \ 1 \ -0.4]$$

ดังนั้น

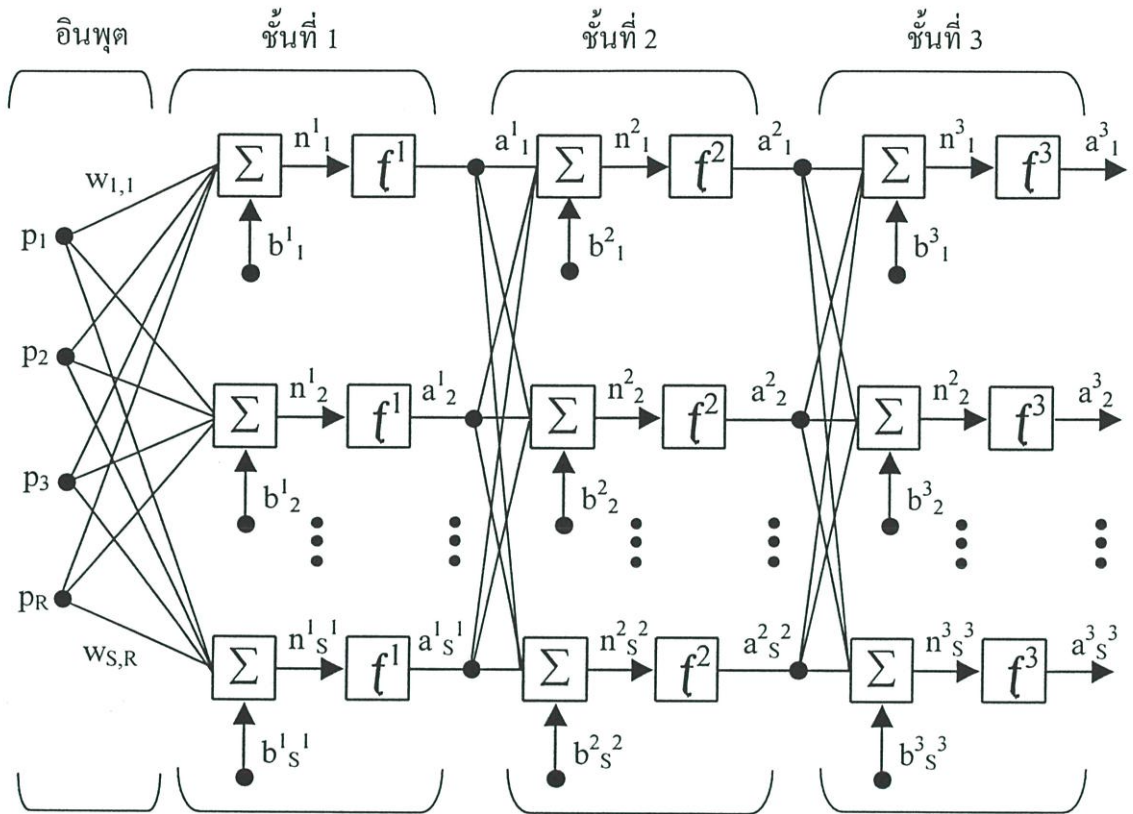
$$a_1 = 0.7$$

$$a_2 = -1$$

$$a_3 = 7.4$$

$$a_4 = -0.4$$

3.2.7 โครงข่ายประสาทเทียมแบบหลายชั้น (Multiple Layers of Neuron Network)



รูปที่ 3.8 โครงข่ายประสาทเทียมแบบหลายชั้น

สมการเอาต์พุตคือ

$$a^3 = f^3(W^3 f^2(W^2 f^1(n^1) + b^2 + b^3)) \tag{3.10}$$

และสามารถกล่าวได้ว่า

$$n^1 = W^1 p + b^1 \tag{3.11}$$

$$n^2 = W^2 f^1(n^1) + b^2 \tag{3.12}$$

$$n^3 = W^3 f^2(n^2) + b^3 \tag{3.13}$$

หรือสามารถเขียนเป็นสมการเอาต์พุตใหม่ได้

$$a^3 = f^3(W^3 f^2(W^2 f^1(W^1 p + b^1) + b^2) + b^3) \tag{3.14}$$

จากโครงสร้างของโครงข่ายประสาทเทียมแบบหลายชั้นในรูปที่ 3.8 จะเห็นว่ามีความซับซ้อนมากกว่าแบบชั้นเดียวมาก โดยโครงสร้างประกอบไปด้วยส่วนของอินพุตเวกเตอร์ p และชั้นต่างๆของโครงข่ายอีก สามชั้น โดยเรียกชั้นที่ 1 และชั้นที่ 2 ว่าชั้นซ่อน (Hidden Layer) ชั้นที่ 3 คือชั้นเอาต์พุต (Output Layer) ชั้นแต่ละชั้นประกอบไปด้วยโครงสร้างที่คล้ายกัน ซึ่งเหมือนกับโครงข่ายประสาทเทียมแบบชั้นเดียวที่นำมาเรียงกันเป็นโครงข่ายใหม่ที่ซับซ้อนมากขึ้น ชั้นแต่ละชั้นประกอบไปด้วย ส่วนที่รับค่าอินพุตเข้ามา และนำไปคูณกับค่าถ่วงน้ำหนักที่ต้องมีในแต่ละชั้น จากนั้นจะถูกลำนำไปรวมกับค่าไบอัส หลังจากนั้นจึงนำไปเข้าทรานส์เฟอร์ฟังก์ชันในการตัดสินใจ การมีชั้นซ่อนหลายๆชั้นมีส่วน ทำให้โครงข่ายประสาทเทียมนั้นสามารถแก้ปัญหาที่ซับซ้อนขึ้นได้ แต่ก็มีคามยุ่งยากในการออกแบบมากขึ้น

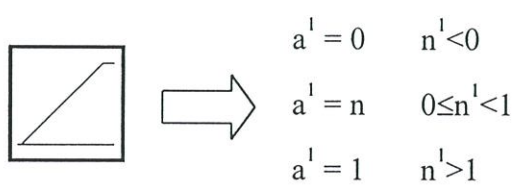
ในแต่ละชั้นของโครงข่ายประสาทเทียมมีโครงสร้างในลักษณะที่คล้ายกัน ส่วนที่ต่างกันคือค่าถ่วงน้ำหนักที่อาจจะไม่จำเป็นต้องเท่ากันในทุกๆชั้นของโครงข่ายประสาทเทียม จำนวนของค่าถ่วงน้ำหนักขึ้นอยู่กับจำนวนอินพุต และจำนวนเซลล์ประสาทในชั้นนั้นๆ ส่วนที่อาจจะต่างกันอีกส่วนคือ ทรานส์เฟอร์ฟังก์ชันที่สามารถออกแบบให้แต่ละชั้นมีความแตกต่างกันได้ ผลลัพธ์ของชั้นเอาต์พุตแสดงให้เห็นดังสมการที่ 3.10 และ 3.14

ตัวแปรต่างๆอาจมีความสับสนเล็กน้อย แต่ในที่นี้ตัวแปรที่ใช้เป็นเลขยกกำลังนั้นหมายถึงชั้นของโครงข่ายประสาทเทียมมิได้หมายถึงค่ายกกำลังเหมือนในคณิตศาสตร์ปกติ ส่วนตัวห้อยในตัวแปรหมายถึงตัวเซลล์ในแต่ละโครงข่ายนั้นๆ

ตัวอย่างที่ 3.5 จงหาค่า a^3 ของเซลล์ประสาทเทียมแบบ 3 ชั้นโดยใช้ ค่าถ่วงน้ำหนักดังที่กำหนดให้ ทรานส์เฟอร์ฟังก์ชันเหมือนกันหมดทั้งสามชั้น และค่าไบอัสเป็นศูนย์

กำหนดให้ $p = [3 \ 2 \ 1 \ 7 \ 0 \ 6 \ 4]$

$$W^1 = \begin{bmatrix} 0.2 & 0.5 & 0.2 & 0.1 & 0.1 & 0.4 & 0.3 \\ 0.1 & 0.3 & 0.1 & 0.4 & 0.1 & 0.2 & 0.5 \\ 0.6 & 0.9 & 0.8 & 0.7 & 0.2 & 0.4 & 0.6 \\ 0.4 & 0.2 & 0.6 & 0.2 & 0.3 & 0.8 & 0.7 \\ 0.2 & 0.2 & 0.2 & 0.2 & 0.4 & 0.1 & 0.2 \end{bmatrix} \quad W^{2,3} = \begin{bmatrix} 0.2 & 0.1 & 0.1 & 0.1 & 0.1 \\ 0.1 & 0.3 & 0.1 & 0.4 & 0.1 \\ 0.6 & 0.9 & 0.8 & 0.7 & 0.2 \\ 0.1 & 0.1 & 0.2 & 0.2 & 0.3 \\ 0.2 & 0.2 & 0.2 & 0.2 & 0.4 \end{bmatrix}$$



จากโจทย์สามารถบอกได้ว่าอินพุตมี 7 อินพุต แต่ละอินพุตต่อเข้ากับโครงข่ายชั้นแรกที่มี เซลล์ประสาทเทียม 5 เซลล์ โดยสามารถพิจารณาได้จากค่าถ่วงน้ำหนักที่กำหนดให้มีขนาด $S \times R$ ค่า S คือจำนวนเซลล์ประสาทเทียม ส่วนค่า $R=7$ มีค่าเท่ากับอินพุตเวกเตอร์ p นั่นเอง

ค่าอินพุตเวกเตอร์ p ที่มี 7 ค่า เมื่อผ่านโครงข่ายประสาทเทียมชั้นแรก จะถูกลดลงเหลือ 5 ตามจำนวนเซลล์ประสาทเทียมในโครงข่าย และถูกส่งต่อไปยังโครงข่ายประสาทเทียมชั้นที่ 2 และชั้นที่ 3 ตามลำดับ ดังนั้นสามารถสรุป หรือคาดคะเนได้ว่าเอาต์พุตของโครงข่ายประสาทเทียมตามตัวอย่างนั้นจะมีเอาต์พุต 5 เอาต์พุตตามเซลล์ประสาทชั้นสุดท้ายนั่นเอง

จากสมการ(3.14) ต้องหาค่า n^1 ได้

$$n^1 = W^1 p + b^1$$

และ

$$a^1 = f(n^1)$$

แทนค่าลงในสมการได้

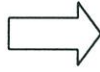
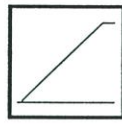
$$n^1 = \begin{bmatrix} 0.2 & 0.5 & 0.2 & 0.1 & 0.1 & 0.4 & 0.3 \\ 0.1 & 0.3 & 0.1 & 0.4 & 0.1 & 0.2 & 0.5 \\ 0.6 & 0.9 & 0.8 & 0.7 & 0.2 & 0.4 & 0.6 \\ 0.4 & 0.2 & 0.6 & 0.2 & 0.3 & 0.8 & 0.7 \\ 0.2 & 0.2 & 0.2 & 0.2 & 0.4 & 0.1 & 0.2 \end{bmatrix} [3 \ 2 \ 1 \ 7 \ 0 \ 6 \ 4]^T + [0]$$

$$= \begin{bmatrix} 0.2 & 0.5 & 0.2 & 0.1 & 0.1 & 0.4 & 0.3 \\ 0.1 & 0.3 & 0.1 & 0.4 & 0.1 & 0.2 & 0.5 \\ 0.6 & 0.9 & 0.8 & 0.7 & 0.2 & 0.4 & 0.6 \\ 0.4 & 0.2 & 0.6 & 0.2 & 0.3 & 0.8 & 0.7 \\ 0.2 & 0.2 & 0.2 & 0.2 & 0.4 & 0.1 & 0.2 \end{bmatrix} \begin{bmatrix} 3 \\ 2 \\ 1 \\ 7 \\ 0 \\ 6 \\ 4 \end{bmatrix} + [0]$$

$$= [6.1 \ 7 \ 14.1 \ 10.8 \ 4] + [0]$$

$$= [6.1 \ 7 \ 14.1 \ 10.8 \ 4]$$

จากทรานส์เฟอร์ฟังก์ชัน



$$\begin{aligned} a^2 = 0 & \quad n^2 < 0 \\ a^2 = n & \quad 0 \leq n^2 < 1 \\ a^2 = 1 & \quad n^2 > 1 \end{aligned}$$

ได้เอาต์พุตของโครงข่ายประสาทเทียมชั้นแรกคือ

$$\mathbf{a}^1 = [1 \ 1 \ 1 \ 1 \ 1]$$

พิจารณาชั้นที่ 2 ได้

$$\mathbf{n}^2 = \mathbf{W}^2 \mathbf{a}^1 + \mathbf{b}^2$$

และ

$$\mathbf{a}^2 = \mathbf{f}(\mathbf{n}^2)$$

แทนค่าลงในสมการได้

$$\begin{aligned} \mathbf{n}^2 &= \begin{bmatrix} 0.2 & 0.1 & 0.1 & 0.1 & 0.1 \\ 0.1 & 0.3 & 0.1 & 0.4 & 0.1 \\ 0.6 & 0.9 & 0.8 & 0.7 & 0.2 \\ 0.1 & 0.1 & 0.2 & 0.2 & 0.3 \\ 0.2 & 0.2 & 0.2 & 0.2 & 0.4 \end{bmatrix} [1 \ 1 \ 1 \ 1 \ 1]^T + [0] \\ &= \begin{bmatrix} 0.2 & 0.1 & 0.1 & 0.1 & 0.1 \\ 0.1 & 0.3 & 0.1 & 0.4 & 0.1 \\ 0.6 & 0.9 & 0.8 & 0.7 & 0.2 \\ 0.1 & 0.1 & 0.2 & 0.2 & 0.3 \\ 0.2 & 0.2 & 0.2 & 0.2 & 0.4 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} + [0] \\ &= [0.6 \ 1 \ 3.2 \ 0.9 \ 1.2] + [0] \\ &= [0.6 \ 1 \ 3.2 \ 0.9 \ 1.2] \end{aligned}$$

ใช้ทรานส์เฟอร์ฟังก์ชันเหมือนกับโครงข่ายประสาทเทียมชั้นแรกได้

$$\mathbf{a}^2 = [0.6 \ 1 \ 1 \ 0.9 \ 1]$$

พิจารณาชั้นที่ 3 ได้

$$\mathbf{n}^3 = \mathbf{W}^3 \mathbf{a}^2 + \mathbf{b}^3$$


และ

$$\mathbf{a}^3 = \mathbf{f}(\mathbf{n}^3)$$

แทนค่าลงในสมการได้

$$\begin{aligned} \mathbf{n}^3 &= \begin{bmatrix} 0.2 & 0.1 & 0.1 & 0.1 & 0.1 \\ 0.1 & 0.3 & 0.1 & 0.4 & 0.1 \\ 0.6 & 0.9 & 0.8 & 0.7 & 0.2 \\ 0.1 & 0.1 & 0.2 & 0.2 & 0.3 \\ 0.2 & 0.2 & 0.2 & 0.2 & 0.4 \end{bmatrix} [0.6 \ 1 \ 1 \ 0.9 \ 1]^T + [0] \\ &= \begin{bmatrix} 0.2 & 0.1 & 0.1 & 0.1 & 0.1 \\ 0.1 & 0.3 & 0.1 & 0.4 & 0.1 \\ 0.6 & 0.9 & 0.8 & 0.7 & 0.2 \\ 0.1 & 0.1 & 0.2 & 0.2 & 0.3 \\ 0.2 & 0.2 & 0.2 & 0.2 & 0.4 \end{bmatrix} \begin{bmatrix} 0.6 \\ 1 \\ 1 \\ 0.9 \\ 1 \end{bmatrix} + [0] \\ &= [2.4 \ 4.7 \ 11.8 \ 3 \ 3.8] + [0] \\ &= [2.4 \ 4.7 \ 11.8 \ 3 \ 3.8] \end{aligned}$$

ใช้ทรานส์เฟอร์ฟังก์ชันเหมือนกับโครงข่ายประสาทเทียมชั้นแรก และชั้นที่ 2



$$\begin{aligned} a^3 &= 0 & n^3 < 0 \\ a^3 &= n^2 & 0 \leq n^3 < 1 \\ a^3 &= 1 & n^3 > 1 \end{aligned}$$

ดังนั้นเอาต์พุตเวกเตอร์ที่ได้คือ

$$\mathbf{a}^3 = [1 \ 1 \ 1 \ 1 \ 1]$$

หรือ

$$\begin{aligned} a_1^3 &= 1 & a_2^3 &= 1 & a_3^3 &= 1 \\ a_4^3 &= 1 & a_5^3 &= 1 & & \end{aligned}$$

ตัวอย่างที่ 3.6 จงหาค่า a^3 ของเซลล์ประสาทเทียมแบบ 3 ชั้น โดยใช้ ค่าถ่วงน้ำหนักทรานส์เฟอร์ฟังก์ชัน และค่าไบอัสเป็นดังที่กำหนดให้

กำหนดให้ $\mathbf{p} = [-1 \ 2 \ 5 \ 4 \ -2 \ 0 \ -5]$

$$W^1 = \begin{bmatrix} 0.2 & 0.2 & 0.2 & 0.1 & 0.1 & 0.4 & 0.3 \\ 0.1 & 0.3 & 0.1 & 0.4 & 0.1 & 0.2 & 0.2 \\ 0.2 & 0.7 & 0.4 & 0.7 & 0.2 & 0.1 & 0.6 \\ 0.4 & 0.2 & 0.3 & 0.2 & 0.3 & 0.8 & 0.7 \\ 0.2 & 0.2 & 0.1 & 0.2 & 0.4 & 0.1 & 0.2 \end{bmatrix}$$

$$W^2 = \begin{bmatrix} -0.2 & 0.01 & 0.02 & 1 & 0.1 \\ 0.01 & 0.03 & 0.01 & 0.04 & 0.01 \\ 0.6 & 0.9 & 0.8 & 0.7 & 0.2 \\ 0.1 & 0.1 & 0.2 & 0.2 & 0.3 \\ 0.2 & 0.2 & 0.2 & 0.2 & 0.4 \end{bmatrix}$$

$$W^3 = \begin{bmatrix} -0.2 & 0.01 & 0.02 & -1 & 0.1 \\ 0.01 & 0.03 & 0.01 & 0.04 & 0.01 \\ 0.6 & 0.9 & 0.8 & 0.7 & 0.2 \\ 0.1 & 0.1 & 0.2 & 0.2 & 0.3 \\ 0.2 & 0.2 & 0.2 & 0.2 & 0.4 \end{bmatrix}$$

$$b^1 = [0 \ 0.1 \ -5 \ 1 \ 0]$$

$$b^2 = [0 \ 0 \ 0 \ -0.1 \ 0]$$

$$b^3 = [0 \ 0 \ 0 \ -0.1 \ 0]$$

$$f^1 \Rightarrow \begin{array}{|c|} \hline \text{Graph of } f^1: \text{A straight line with a positive slope.} \\ \hline \end{array} \Rightarrow \begin{array}{l} a^1 = -1 \quad n^1 < -1 \\ a^1 = n \quad -1 \leq n^1 < -1 \\ a^1 = 1 \quad n^1 > 1 \end{array}$$

$$f^2 \Rightarrow \begin{array}{|c|} \hline \text{Graph of } f^2: \text{A straight line with a positive slope, starting from the origin.} \\ \hline \end{array} \Rightarrow \begin{array}{l} a^2 = 0 \quad n^2 < 0 \\ a^2 = n^2 \quad n^2 \geq 0 \end{array}$$

$$f^3 \Rightarrow \begin{array}{|c|} \hline \text{Graph of } f^3: \text{An S-shaped curve (sigmoid function).} \\ \hline \end{array} \Rightarrow a^3 = \frac{1}{1 + e^{-n^3}}$$

จากสมการ

$$\mathbf{a}^3 = \mathbf{f}^3(\mathbf{W}^3 \mathbf{f}^2(\mathbf{W}^2 \mathbf{f}^1(\mathbf{W}^1 \mathbf{p} + \mathbf{b}^1) + \mathbf{b}^2 + \mathbf{b}^3))$$

$$\mathbf{n}^1 = \mathbf{W}^1 \mathbf{p} + \mathbf{b}^1$$

แทนค่าลงในสมการหาได้

$$\mathbf{n}^1 = \begin{bmatrix} 0.2 & 0.2 & 0.2 & 0.1 & 0.1 & 0.4 & 0.3 \\ 0.1 & 0.3 & 0.1 & 0.4 & 0.1 & 0.2 & 0.2 \\ 0.2 & 0.7 & 0.4 & 0.7 & 0.2 & 0.1 & 0.6 \\ 0.4 & 0.2 & 0.3 & 0.2 & 0.3 & 0.8 & 0.7 \\ 0.2 & 0.2 & 0.1 & 0.2 & 0.4 & 0.1 & 0.2 \end{bmatrix} [-1 \ 2 \ 5 \ 4 \ -2 \ 0 \ -5]^T + \mathbf{b}^1$$

$$= \begin{bmatrix} 0.2 & 0.2 & 0.2 & 0.1 & 0.1 & 0.4 & 0.3 \\ 0.1 & 0.3 & 0.1 & 0.4 & 0.1 & 0.2 & 0.2 \\ 0.2 & 0.7 & 0.4 & 0.7 & 0.2 & 0.1 & 0.6 \\ 0.4 & 0.2 & 0.3 & 0.2 & 0.3 & 0.8 & 0.7 \\ 0.2 & 0.2 & 0.1 & 0.2 & 0.4 & 0.1 & 0.2 \end{bmatrix} \begin{bmatrix} -1 \\ 2 \\ 5 \\ 4 \\ -2 \\ 0 \\ -5 \end{bmatrix} + \mathbf{b}^1$$

$$= [-0.1 \ 1.4 \ 2.6 \ -1.8 \ -0.3] + [0 \ 0.1 \ -5 \ 1 \ 0]$$

$$= [-0.1 \ 1.5 \ -2.4 \ 0.8 \ -0.3]$$

จากทรานส์เฟอร์ฟังก์ชันของโครงข่ายประสาทเทียมชั้นแรก

$$\mathbf{f}^1 \quad \longrightarrow \quad \begin{array}{|c|} \hline \text{Graph of a sigmoid function} \\ \hline \end{array} \quad \longrightarrow \quad \begin{array}{ll} \mathbf{a}^1 = -1 & \mathbf{n}^1 < -1 \\ \mathbf{a}^1 = \mathbf{n} & -1 \leq \mathbf{n}^1 < 1 \\ \mathbf{a}^1 = 1 & \mathbf{n}^1 > 1 \end{array}$$

ได้เอาต์พุตของโครงข่ายประสาทเทียมชั้นแรกคือ

$$\mathbf{a}^1 = [-0.1 \ 1 \ -1 \ 0.8 \ -0.3]$$

พิจารณาชั้นที่ 2 ได้

$$\mathbf{n}^2 = \mathbf{W}^2 \mathbf{a}^1 + \mathbf{b}^2$$

และ

$$\mathbf{a}^2 = \mathbf{f}(\mathbf{n}^2)$$

แทนค่าลงในสมการได้

$$\begin{aligned} \mathbf{n}^2 &= \begin{bmatrix} -0.2 & 0.01 & 0.02 & 1 & 0.1 \\ 0.01 & 0.03 & 0.01 & 0.04 & 0.01 \\ 0.6 & 0.9 & 0.8 & 0.7 & 0.2 \\ 0.1 & 0.1 & 0.2 & 0.2 & 0.3 \\ 0.2 & 0.2 & 0.2 & 0.2 & 0.4 \end{bmatrix} [-0.1 \ 1 \ -1 \ 0.8 \ -0.3]^T + \mathbf{b}^2 \\ &= \begin{bmatrix} -0.2 & 0.01 & 0.02 & 1 & 0.1 \\ 0.01 & 0.03 & 0.01 & 0.04 & 0.01 \\ 0.6 & 0.9 & 0.8 & 0.7 & 0.2 \\ 0.1 & 0.1 & 0.2 & 0.2 & 0.3 \\ 0.2 & 0.2 & 0.2 & 0.2 & 0.4 \end{bmatrix} \begin{bmatrix} -0.1 \\ 1 \\ -1 \\ 0.8 \\ -0.3 \end{bmatrix} + \mathbf{b}^2 \\ &= [0.78 \ 0.048 \ 0.54 \ -0.04 \ 0.02] + [0 \ 0 \ 0 \ -0.1 \ 0] \\ &= [0.78 \ 0.048 \ 0.54 \ -0.14 \ 0.02] \end{aligned}$$

จากทรานส์เฟอร์ฟังก์ชันของโครงข่ายประสาทเทียมชั้นที่ 2

$$\mathbf{f}^2 \quad \Rightarrow \quad \boxed{\text{เส้นตรง}} \quad \Rightarrow \quad \begin{array}{ll} \mathbf{a}^2 = 0 & \mathbf{n}^2 < 0 \\ \mathbf{a}^2 = \mathbf{n}^2 & \mathbf{n}^2 \geq 0 \end{array}$$

ได้เอาต์พุตของโครงข่ายประสาทเทียมชั้นที่ 2 คือ

$$\mathbf{a}^2 = [0.78 \ 0.048 \ 0.54 \ 0 \ 0.02]$$

พิจารณาชั้นที่ 3 ได้

$$\mathbf{n}^3 = \mathbf{W}^3 \mathbf{a}^2 + \mathbf{b}^3$$

และ

$$\mathbf{a}^3 = \mathbf{f}(\mathbf{n}^3)$$

แทนค่าลงในสมการได้

$$\begin{aligned} \mathbf{n}^3 &= \begin{bmatrix} -0.2 & 0.01 & 0.02 & -1 & 0.1 \\ 0.01 & 0.03 & 0.01 & 0.04 & 0.01 \\ 0.6 & 0.9 & 0.8 & 0.7 & 0.2 \\ 0.1 & 0.1 & 0.2 & 0.2 & 0.3 \\ 0.2 & 0.2 & 0.2 & 0.2 & 0.4 \end{bmatrix} [0.78 \ 0.048 \ 0.54 \ 0 \ 0.02]^T + \mathbf{b}^3 \\ &= \begin{bmatrix} -0.2 & 0.01 & 0.02 & -1 & 0.1 \\ 0.01 & 0.03 & 0.01 & 0.04 & 0.01 \\ 0.6 & 0.9 & 0.8 & 0.7 & 0.2 \\ 0.1 & 0.1 & 0.2 & 0.2 & 0.3 \\ 0.2 & 0.2 & 0.2 & 0.2 & 0.4 \end{bmatrix} \begin{bmatrix} 0.78 \\ 0.048 \\ 0.54 \\ 0 \\ 0.02 \end{bmatrix} + \mathbf{b}^3 \\ &= [-0.14272 \ 0.55944 \ 0.9472 \ 0.1968 \ 0.2816] + [0 \ 0 \ 0 \ -0.1 \ 0] \\ &= [-0.14272 \ 0.55944 \ 0.9472 \ 0.0968 \ 0.2816] \end{aligned}$$

จากทรานส์เฟอร์ฟังก์ชันของโครงข่ายประสาทเทียมชั้นที่ 3

$$\mathbf{f}^3 \Rightarrow \boxed{\text{Sigmoid}} \Rightarrow \mathbf{a}^3 = \frac{1}{1 + e^{-\mathbf{n}^3}}$$

ได้เอาต์พุตของโครงข่ายประสาทเทียมชั้นที่ 3 คือ

$$\mathbf{a}^3 = [0.464380 \ 0.63632 \ 0.72055 \ 0.52418 \ 0.56993]$$

3.2.7 การเรียนรู้ (Learning)

โดยปกติสิ่งมีชีวิตที่มีโครงข่ายประสาทเมื่อแรกเกิดจะต้องมีการเรียนรู้สิ่งต่างๆรอบตัวทั้งมีผู้สอนและเรียนรู้ด้วยตัวเองเพื่อให้มีชีวิตรอดได้ในธรรมชาติ ในกรณีเดียวกับโครงข่ายประสาทเทียมการที่จะมีความสามารถในการจดจำ (Recognition) หรือแยกแยะ (Classification) สิ่งต่างๆได้นั้นจำเป็นที่จะต้องมีการเรียนรู้ โดยสามารถแบ่งการเรียนรู้ได้เป็น 2 แบบใหญ่ๆคือแบบมีผู้สอน (Supervised Learning) และแบบไม่มีผู้สอน (Unsupervised Learning) ผลลัพธ์ที่ได้จากการเรียนรู้คือค่าถ่วงน้ำหนักที่เสถียร และสามารถใช้แก้ปัญหาในขอบเขตที่กำหนดไว้อย่างถูกต้องในที่นี้จะกล่าวถึงการเรียนรู้แบบมีผู้สอนเท่านั้น

3.2.7.1 กฎการเรียนรู้ของเพอเซปตรอน (Perceptron Learning Rule)

โครงข่ายประสาทเทียมแบบชั้นเดียว มีค่าถ่วงน้ำหนักเชื่อมต่อเพียงชั้นเดียว อินพุตแยกออกจากกันและรับสัญญาณต่างๆมาจากภายนอก ส่วนหน่วยเอาต์พุตเป็นผลตอบสนองจากโครงข่ายประสาทเทียม ในตัวอย่างโครงข่ายประสาทเทียมแบบชั้นเดียวในรูปที่ 3.7 หน่วยของอินพุตทำการเชื่อมต่อกับหน่วยเอาต์พุตทั้งหมดแต่ไม่เชื่อมต่อกับหน่วยอื่นๆของอินพุตด้วยกัน และหน่วยของเอาต์พุตไม่เชื่อมต่อกับหน่วยอื่นๆ ของชั้นเอาต์พุตด้วยกัน

กฎการเรียนรู้แบบเพอเซปตรอน (Perceptron) ใช้วิธีการสอนแบบต้องมีผู้สอน กล่าวคือ ต้องมีเอาต์พุตที่ต้องการเป็นตัวเปรียบเทียบกับอินพุตที่ป้อนให้กับโครงข่ายประสาทเทียม หากค่าที่เปรียบเทียบมีค่าผิดพลาดมาก ก็ต้องทำการปรับค่าน้ำหนักในโครงข่ายใหม่ตามวิธีของการของเพอเซปตรอน ในทางกลับกันถ้าการเปรียบเทียบนั้นอยู่ในเกณฑ์ที่ยอมรับได้ก็ถือว่าค่าถ่วงน้ำหนักนั้นสามารถใช้ได้ แต่ต้องไม่ลืมว่าต้องกระทำครบทุกคู่อินพุตกับเอาต์พุตแล้ว โดยกฎการเรียนรู้แบบเพอเซปตรอนสามารถเขียนเป็นกระบวนการทำงานได้ดังนี้

1. ทำการสุ่มค่าน้อยๆ ให้กับค่าถ่วงน้ำหนักทุกตัว (W) กำหนดค่าผิดพลาดรวมสูงสุด (E_r) และกำหนดค่าอัตราการเรียนรู้ (α)

2. ใส่อินพุตเวกเตอร์ \mathbf{p} ในชั้นอินพุตซึ่งต้องทำการนอมอลไลซ์ก่อน และกำหนดค่าให้เวกเตอร์ \mathbf{a} ซึ่งเป็นเอาต์พุตที่ต้องการ

$$\mathbf{p} = [p_1 \ p_2 \ p_3 \ \dots \ p_r] \quad \mathbf{d} = [d_1 \ d_2 \ d_3 \ \dots \ d_s] \quad (3.15)$$

3. คำนวณค่า \mathbf{n}

$$\mathbf{n} = \mathbf{W}\mathbf{p}^T + \mathbf{b} \quad (3.16)$$

4. คำนวณค่าเอาต์พุตจากชั้นเอาต์พุตผ่านทรานส์เฟอร์ฟังก์ชัน

$$\mathbf{a} = \mathbf{f}(\mathbf{W}\mathbf{p} + \mathbf{b}) \quad (3.17)$$

5. คำนวณค่าผิดพลาดจาก

$$E = d - a \quad (3.18)$$

6. ปรับปรุงค่าถ่วงน้ำหนักระหว่างชั้นอินพุตกับชั้นเอาต์พุต

$$W = W + \alpha \delta p \quad (3.19)$$

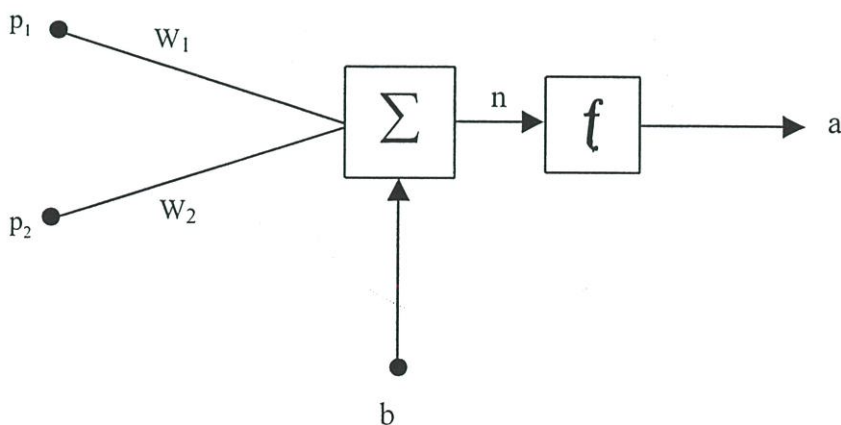
7. ทำจนครบทุกชุดของข้อมูลที่ต้องการสอนให้กับโครงข่ายประสาทเทียม ถ้ายังไม่ครบให้กลับไปเริ่มทำที่ข้อ 2

8. ตรวจสอบค่าผิดพลาดรวมที่คำนวณได้จากข้อ 5 (E) กับค่าผิดพลาดที่กำหนด (E_k) ว่าน้อยกว่าหรือไม่ $E < E_k$ ถ้ามากกว่ากลับไปเริ่มทำที่ข้อ 2 และเริ่มที่ข้อมูลชุดแรก ถ้าน้อยกว่าแสดงว่าการสอนเสร็จสมบูรณ์ให้เก็บค่าถ่วงน้ำหนัก (W) ไว้เพื่อใช้งานต่อไปและจบขั้นตอนการสอน

จากข้อที่ 4 ทราנסเฟอร์ฟังก์ชันซึ่งมีอยู่ด้วยกันหลายแบบขึ้นอยู่กับทางเลือกใช้อย่างที่ได้กล่าวไปแล้ว เช่น ฟังก์ชันลอจิสติก ฟังก์ชันไฮเปอร์โบลิกแทนเจนต์ ฟังก์ชันฮาร์ดลิมิต ดังที่แสดงไว้ในตารางที่ 1

การแก้ปัญหาโดยใช้กฎการเรียนรู้แบบเพอเซปตรอนไม่ประสบความสำเร็จมากนัก เนื่องจากเป็นการเรียนรู้แบบแรกๆที่นักวิทยาศาสตร์ได้คิดขึ้น แต่ต่อมาก็ถูกแก้ไขและพัฒนาแนวความคิดของการเรียนรู้ให้สามารถแก้ปัญหาที่ซับซ้อนขึ้นได้

เนื่องจากความสามารถในการแก้ปัญหาของการเรียนรู้แบบเพอเซปตรอน ใช้ได้กับปัญหาต่างๆ เท่านั้น เช่น ปัญหา AND , ปัญหา OR เป็นต้น ซึ่งปัญหาที่ถุณนำมาแก้ด้วยเพอเซปตรอนเป็นปัญหาที่ง่ายๆ โดยใช้เส้นตรงเพียงเส้นเดียว แบ่งแยกเอาต์พุตที่ต้องการออกจากกันได้ดังตัวอย่างปัญหา AND ต่อไปนี้



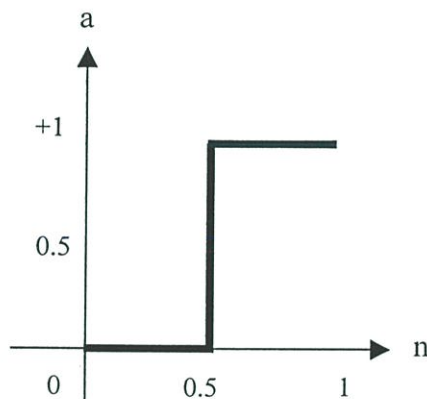
รูปที่ 3.9 โครงข่ายประสาทเทียมที่มี 2 อินพุต และ 1 เอาต์พุต ที่ใช้แก้ปัญหา AND

ตารางที่ 3.2 ฟังก์ชัน AND

P_1	P_2	a
0	0	0
0	1	0
1	0	0
1	1	1

ตารางของฟังก์ชัน AND มีอินพุตเป็นเลขฐานสองและต้องการค่าถ่วงน้ำหนักที่เหมาะสมกับอินพุตเพื่อที่โครงข่ายนี้สามารถคำนวณค่าเอาต์พุตจากอินพุตได้อย่างถูกต้อง

จากภาพที่ 3.9 สามารถหาค่าถ่วงน้ำหนักได้หลายชุดที่เหมาะสมกับโครงข่ายประสาทเทียมนี้ขึ้นอยู่กับข้อกำหนดเทรคโบลด์ ซึ่งในที่นี้ได้เลือกใช้ทรานส์เฟอร์ฟังก์ชันฮาร์ดลิมิต ที่มีการปรับแต่งการเกิดของทรานส์เฟอร์ฟังก์ชันเล็กน้อยมีลักษณะดังรูปข้างล่างนี้

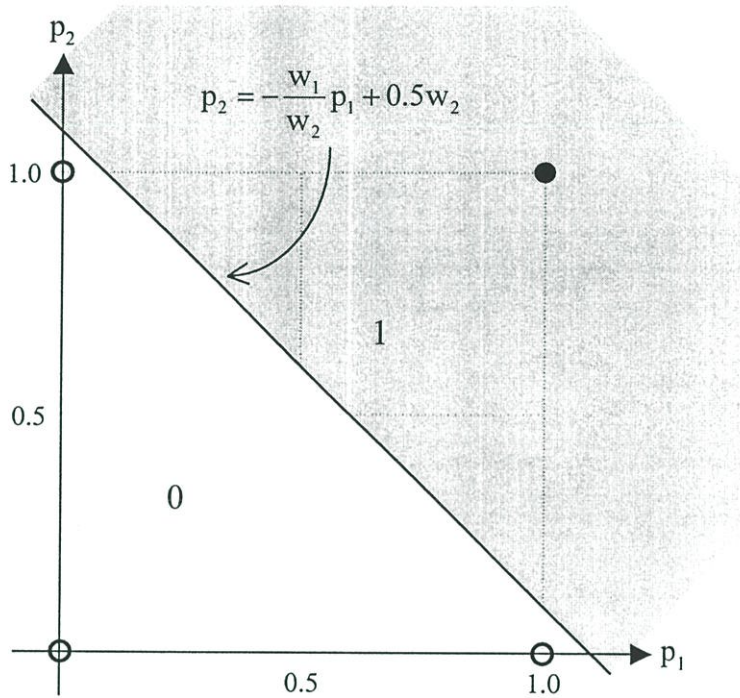


รูปที่ 3.10 ฟังก์ชันฮาร์ดลิมิต

เมื่อทำการเขียนสมการทรานส์เฟอร์ฟังก์ชันจากรูปที่ 3.10 จะได้ดังนี้

$$f = \begin{cases} a = 0 & ; n < 0 \\ a = 1 & ; n > 0.5 \end{cases}$$

เมื่อทำการสอนตามขั้นตอนจะได้ค่า $w_1 = 0.49$ และ $w_2 = 0.49$ และสามารถนำไปใช้กับโครงข่ายประสาทเทียมที่ได้ทำการออกแบบไว้ ซึ่งให้ค่าเอาต์พุตออกมาสัมพันธ์กับอินพุตที่ป้อนเข้าให้กับโครงข่ายประสาทเทียมอย่างถูกต้องตามที่ทำการสอนไว้

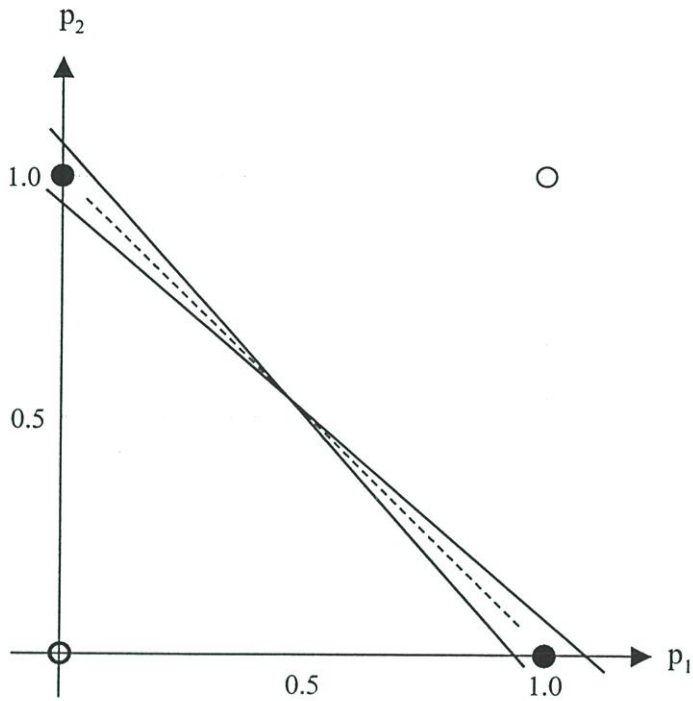


รูปที่ 3.11 จุดต่างๆ บนระนาบและเส้นแบ่งพื้นที่เอาต์พุตของการแก้ปัญหา AND

จากภาพที่ 3.11 สมการเส้นตรงที่ได้สามารถแบ่งจุดที่มีอยู่ออกเป็น 2 ส่วนด้วยกันคือส่วนที่ให้เอาต์พุตเป็น 1 และ 0 ส่วนที่ให้เอาต์พุตเป็น 1 ใช้เป็นจุดทึบและส่วนที่แรเงาส่วนที่ให้เอาต์พุตเป็น 0 ใช้เป็นจุดโปร่ง หากพิจารณาแล้วเห็นได้ว่าเป็นเพอเซปตรอนสามารถแก้ปัญหาข้างต้นได้เป็นอย่างดี แต่ถ้าปัญหามีความสลับซับซ้อนมากขึ้น การใช้เพอเซปตรอนในการแก้ปัญหาก็จะมีความยุ่งยากขึ้นตามไปด้วย ยกตัวอย่างเช่นปัญหา XOR ในตารางที่ 3.3 ซึ่งเป็นปัญหาอย่างง่าย แต่ไม่สามารถใช้สมการเส้นตรงมาแบ่งแยกพื้นที่เอาต์พุตได้อย่างถูกต้องบนระนาบเดียวกันรูปที่ 3.12

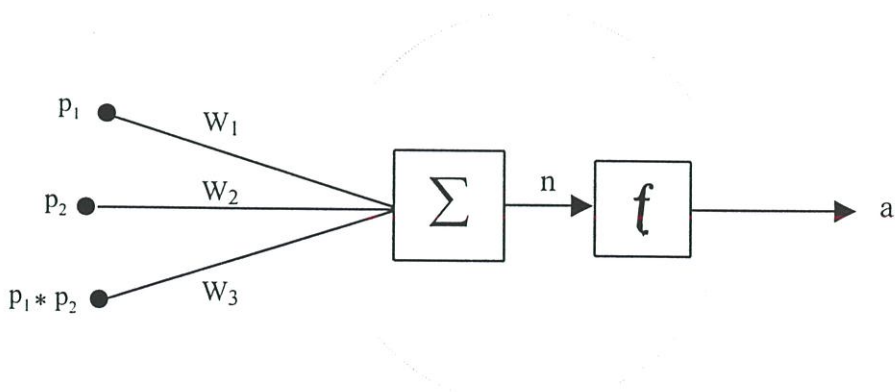
ตารางที่ 3.3 ฟังก์ชัน XOR

p_1	p_2	a
0	0	0
0	1	1
1	0	1
1	1	0



รูปที่ 3.12 การใช้เพอเซปตรอนปรกติแบ่งพื้นที่เอาต์พุตซึ่งไม่สามารถแก้ปัญหา XOR ได้

การใช้เพอเซปตรอนแบบธรรมดาไม่สามารถแก้ปัญหา XOR ได้เพราะว่าไม่สามารถลากเส้นแบ่งเขตที่แยกเอาต์พุตที่เป็น 0 และ 1 ออกจากกันเป็นสองส่วนได้ดังรูปที่ 3.21 อย่างไรก็ตาม การจะแก้ปัญหา XOR โดยใช้เพอเซปตรอนยังคงทำได้อยู่เพียงแต่ กระทำได้โดยการเลียนแบบการออกแบบของโครงข่ายประสาทเทียมที่ใช้ในการแก้ปัญหา AND จากที่กล่าวมาแล้วโดยการเพิ่มอินพุตอีกหนึ่งอินพุต รวมทั้งหมดเป็นสามอินพุต และมีหนึ่งเอาต์พุตเช่นกัน เหตุที่ต้องเพิ่มอินพุตเข้ามา เพราะว่าการที่จะเพิ่มข้อมูลที่เป็นความแตกต่างของข้อมูลอินพุตสองอินพุตแรก ซึ่งผู้ออกแบบต้องการจะใส่ข้อมูลอะไรลงไปอินพุตที่สามเพื่อให้สามารถแยกอินพุตทั้งหมดได้ ดังนั้นลักษณะ โครงสร้างของโครงข่ายประสาทเทียมจะเปลี่ยนไปดังรูปที่ 3.13



รูปที่ 3.13 โครงข่ายประสาทเทียม 3 อินพุต และ 1 เอาต์พุต ที่ใช้แก้ปัญหา XOR

จากรูปที่ 3.14 ได้เลือกอินพุตหน่วยที่ 3 โดยได้จากค่าผลคูณของอินพุตแรก กับอินพุตที่สองเพื่อเป็นข้อมูลเพิ่มเติมเพื่อใช้ในการจัดแยกข้อมูล ทรานส์เฟอร์ฟังก์ชันที่ใช้จะเป็นทรานส์เฟอร์ฟังก์ชันแบบเดียวกับการแก้ปัญหา AND จะได้ดังนี้

$$f = \begin{cases} a = 0 & ; n < 0 \\ a = 1 & ; n > 0.5 \end{cases}$$

เมื่อทำการสอนตามขั้นตอนที่กล่าวไปแล้วจะได้ค่าถ่วงน้ำหนัก

$$W_1 = 0.63$$

$$W_2 = 0.63$$

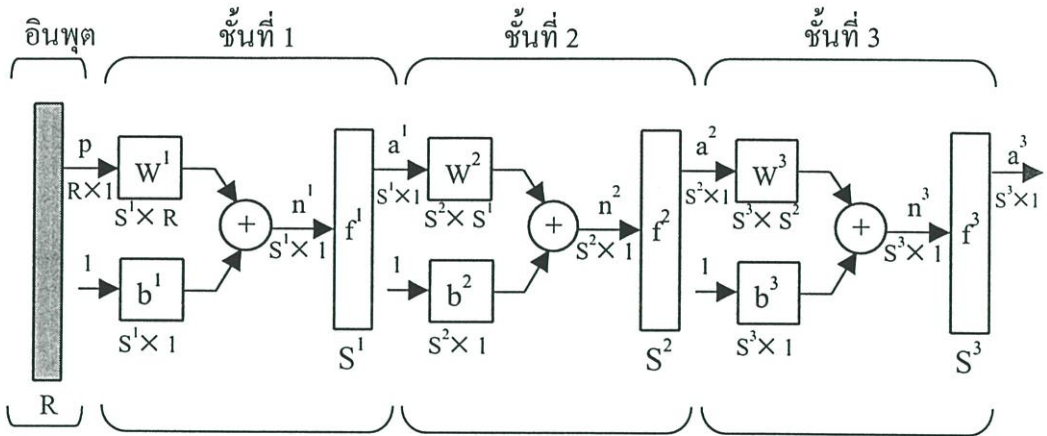
$$W_3 = -0.77$$

ค่าถ่วงน้ำหนักที่คำนวณได้ สามารถนำไปใช้กับโครงข่ายประสาทเทียมที่ได้ทำการออกแบบไว้ ซึ่งให้ค่าเอาต์พุตออกมาสัมพันธ์กับอินพุตที่ป้อนเข้าให้กับโครงข่ายประสาทเทียมอย่างถูกต้องตามที่ทำการสอนไว้

เมื่อปัญหาที่มีความสลับซับซ้อนมากขึ้น การแก้ปัญหาด้วยเพอเซปตรอนจะยุ่งยากตามขึ้นไป ด้วย จากตัวอย่างในการแก้ปัญหา XOR จะเห็นได้ว่าอินพุตหน่วยที่ 3 เป็นส่วนที่เพิ่มขึ้นมาซึ่งเป็นส่วนที่ทำให้ยุ่งยากเนื่องจากต้องการหาข้อมูลที่เหมาะสมมาใส่ให้ถูกต้อง ถ้าปัญหาที่มีความยุ่งยากกว่านี้อาจจะต้องมีส่วนนี้มากขึ้น ทำให้เพิ่มความยุ่งยากมากขึ้น ในการแก้ปัญหาส่วนนี้ควรใช้โครงข่ายประสาทเทียมแบบหลายชั้น จะมีความเหมาะสมกว่าดังจะกล่าวในหัวข้อต่อไป

3.2.7.2 แบทพรอพพาเกชัน (Backpropagation Neural Network)

โครงข่ายประสาทเทียมแบบหลายชั้นดังรูปที่ 3.8 สามารถแก้ปัญหาซับซ้อนมากๆ ได้ดีกว่าโครงข่ายประสาทเทียมแบบชั้นเดียว แต่แน่นอนว่าการสอนอาจยากกว่า โครงข่ายปรอทเทียมแบบหลายชั้นเป็นโครงข่ายที่มีชั้นของชั้นซ่อนหนึ่งหรือมากกว่านั้น ระหว่างส่วนอินพุตและส่วนเอาต์พุต การส่งข้อมูลถูกส่งไปทิศทางเดียวซึ่งจะเห็นได้ว่าข้อมูลส่งจากซ้ายไปขวา ในชั้นเดียวกันไม่มีการส่งข้อมูลถึงกัน แต่จะมีการส่งระหว่างชั้นเชื่อมกันหมดในแต่ละเส้นทางการเชื่อมต่อ มีค่าถ่วงน้ำหนักประจำอยู่ และใช้วิธีการสอนแบบต้องมีเอาต์พุตที่ต้องการเป็นตัวเปรียบเทียบหรือต้องมีผู้สอน ซึ่งเมื่อเปรียบเทียบเอาต์พุตกับเอาต์พุตที่ต้องการ ก็ได้ค่าผิดพลาดของชั้นเอาต์พุตโดยในรูปที่ 3.8 คือชั้นที่ 3 นั่นเอง และนำค่าผิดพลาดของชั้นเอาต์พุตที่ได้นี้ส่งย้อนกลับไปเพื่อใช้ในการคำนวณค่าผิดพลาดของชั้นซ่อน และนำค่าผิดพลาดที่ได้ทั้งหมดไปปรับปรุงค่าถ่วงน้ำหนักในแต่ละชั้นต่อไป ในการส่งค่าผิดพลาดย้อนกลับ การส่งค่าผิดพลาดย้อนกลับจากชั้นเอาต์พุตไปยังชั้นซ่อนนี้เองเป็นสาเหตุที่เรียกว่าแบคพรอพพาเกชัน ในรูปที่ 3.8 เป็นการอธิบายที่เน้นถึงโครงสร้างของโครงข่ายประสาทเทียม แต่เพื่อให้ง่ายต่อการอธิบายต่อไปจึงต้องทำการปรับรูปที่ 3.8 เป็นรูปที่ 3.14 ให้มีลักษณะโครงสร้างที่ซับซ้อนน้อยลงโดยเน้นไปที่ตัวแปรที่ต้องใช้ในการคำนวณ



รูปที่ 3.14 โครงข่ายประสาทเทียมแบบหลายชั้นเน้นไปที่ตัวแปรที่ต้องใช้ในการคำนวณ

กระบวนการเรียนรู้ของแบคพรอปพาทะชั้นมีดังนี้

1. สุ่มค่าให้กับค่าถ่วงน้ำหนัก (W) และค่าไบอัส (b) ทุกตัว ทุกชั้นกำหนดค่าผิดพลาดรวมสูงสุด (E_k)
2. ใส่อินพุตเวกเตอร์ p ในชั้นอินพุตซึ่งต้องทำการนอมอลไลซ์ก่อน และกำหนดค่าให้เวกเตอร์ d ซึ่งเป็นเอาต์พุตที่ต้องการ

$$p = [p_1 \ p_2 \ p_3 \ \dots \ p_R] \quad d = [d_1 \ d_2 \ d_3 \ \dots \ d_S]$$

3. คำนวณค่า a^3 ของชั้นเอาต์พุตผ่านทรานส์เฟอร์ฟังก์ชัน

$$a^3 = f(W^3 a^2 + b^3) \quad (3.20)$$

4. คำนวณค่าผิดพลาดจาก

$$E = d - a \quad (3.21)$$

5. หาค่าอนุพันธ์ของฟังก์ชัน (f') ในทุกชั้น

6. คำนวณค่าความไว (Sensitivity) ของทุกชั้นย้อนจากเอาต์พุตไปอินพุตจาก

$$S^M = -2\dot{F}^M(n^M)(d-a) \quad M = \text{จำนวนชั้น} \quad (3.22)$$

$$S^m = \dot{F}^m(n^m)(W^{m+1})^T S^{m+1} \quad m = M-1, \dots, 2, 1 \quad (3.23)$$

หรืออาจกล่าวได้ว่า

$$S^M \rightarrow S^{M-1} \rightarrow \dots \rightarrow S^2 \rightarrow S^1$$

7. ปรับปรุงค่าถ่วงน้ำหนัก

$$W^m(k+1) = W^m(k) - \alpha S^m (a^{m-1})^T \quad (3.24)$$

8. ปรับปรุงค่าไบอัส

$$b^m(k+1) = b^m(k) - \alpha S^M \tag{3.25}$$

9. ทำจนครบทุกชุดของข้อมูลที่ต้องการสอนให้กับโครงข่ายประสาทเทียม ถ้ายังไม่ครบให้กลับไปเริ่มทำที่ข้อ 2

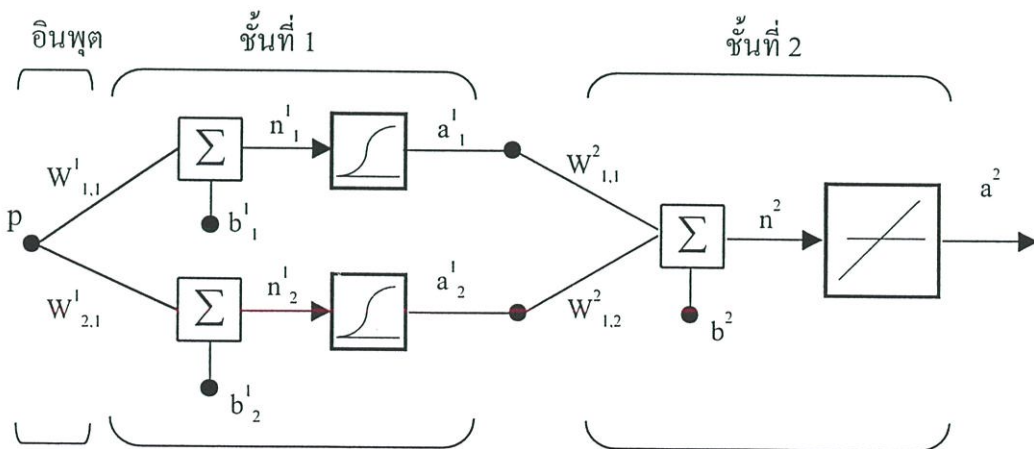
10. ตรวจสอบค่าผิดพลาดรวมที่คำนวณได้จากข้อ 4 (E) กับค่าผิดพลาดที่กำหนด (E_k) ว่าน้อยกว่าหรือไม่ ($E < E_k$) ถ้ามากกว่ากลับไปเริ่มทำที่ข้อ 2 และเริ่มที่ข้อมูลชุดแรก ถ้าน้อยกว่าแสดงว่าการสอนเสร็จสมบูรณ์ให้เก็บค่าถ่วงน้ำหนัก (W) ไว้ใช้งานต่อไปและจบขั้นตอนการสอน

จากข้อ 3 ทราเน็ตเฟอร์ฟังก์ชันของชั้นเอาต์พุต และทราเน็ตเฟอร์ฟังก์ชันของชั้นซ่อนขึ้นอยู่กับทางเลือกใช้งาน เช่น ฟังก์ชันลอจิสติก ฟังก์ชันไฮเปอร์โบลิกแทนเจนต์ ฟังก์ชันฮาร์ดลิมิต เป็นต้น ซึ่งฟังก์ชันที่เลือกใช้ควรเหมือนกัน จากขั้นตอนการสอนจะเห็นได้ว่าในข้อ 7 และ ข้อ 8 มีการปรับปรุงค่าถ่วงน้ำหนัก และค่าไบอัส ตามลำดับ สาเหตุนี้ทำให้ถูกเรียกว่าแบคพรอพพาเกชัน ส่วน f' เป็นอนุพันธ์อันดับหนึ่งของฟังก์ชัน

การแก้ปัญหาโดยใช้วิธีแบคพรอพพาเกชันประสบผลสำเร็จพอสมควร เนื่องจากมีความสามารถใช้ในการแก้ปัญหาที่สลับซับซ้อนได้โดยง่าย แต่การออกแบบโครงข่ายประสาทเทียมยังไม่สามารถหาวิธีการออกแบบได้รวดเร็ว และเหมาะสมกับข้อมูลการเรียนรู้ของโครงข่ายประสาทเทียมยังต้องใช้เวลาอย่างมากด้วย ตัวอย่างที่ 3.6 เป็นการใช้งานโครงข่ายประสาทเทียมโดยใช้การเรียนรู้แบบแบคพรอพพาเกชันที่ใช้สูตรการคำนวณจากที่กล่าวไปแล้ว

ตัวอย่างที่ 3.6 จงใช้โครงข่ายประสาทเทียมแบบหลายชั้นในการประมาณค่าฟังก์ชันที่กำหนดให้โดยใช้การเรียนรู้แบบแบคพรอพพาเกชัน

$$g(p) = 1 + \sin\left(\frac{\pi}{4}p\right) ; -2 \leq p \leq 2$$



รูปที่ 3.15 โครงข่ายประสาทเทียมแบบหลายชั้นที่ใช้ประมาณค่าของฟังก์ชัน $g(p)$

1. ทำการส่งค่าให้กับค่าถ่วงน้ำหนัก และค่าไบอัสทุกตัว

$$W^1(0) = \begin{bmatrix} -0.27 \\ -0.41 \end{bmatrix} \quad b^1(0) = \begin{bmatrix} -0.48 \\ -0.13 \end{bmatrix}$$

$$W^2(0) = [0.48 \quad -0.17] \quad b^2(0) = [0.48]$$

$$E_k = 0.01$$

2. ใส่ค่า อินพุต p และคำนวณหา d

$$p = a^0 = 1$$

$$d = g(1) = 1 + \sin\left(\frac{\pi}{4} \cdot 1\right) = 1.0707$$

3. คำนวณค่า a^2 ของชั้นเอาต์พุตผ่านทรานส์เฟอร์ฟังก์ชัน

$$\begin{aligned} a^1 &= f'(W^1 a^0 + b^1) \\ &= \text{logsig}\left(\begin{bmatrix} -0.27 \\ -0.41 \end{bmatrix} [1] + \begin{bmatrix} -0.48 \\ -0.13 \end{bmatrix}\right) \\ &= \text{logsig}\left(\begin{bmatrix} -0.75 \\ -0.54 \end{bmatrix}\right) \\ &= \begin{bmatrix} \frac{1}{1+e^{0.75}} \\ \frac{1}{1+e^{0.54}} \end{bmatrix} \\ &= \begin{bmatrix} 0.321 \\ 0.368 \end{bmatrix} \end{aligned}$$

$$\begin{aligned} a^2 &= f^2(W^2 a^1 + b^2) \\ &= \text{purelin}\left(\begin{bmatrix} 0.09 & -0.17 \end{bmatrix} \begin{bmatrix} 0.321 \\ 0.368 \end{bmatrix} + [0.48]\right) \\ &= \text{purelin}(0.466) \\ &= 0.466 \end{aligned}$$

4. คำนวณค่าผิดพลาดจาก

$$\begin{aligned} E &= d - a \\ &= 1.0707 - 0.466 \\ &= 1.261 \end{aligned}$$

5. หาค่าอนุพันธ์ของฟังก์ชัน (f') ในทุกชั้น

$$\begin{aligned}\dot{f}^1(n) &= \frac{d}{dn} \left(\frac{1}{1+e^{-n}} \right) \\ &= \frac{e^{-n}}{(1+e^{-n})^2} \\ &= \left(1 - \frac{1}{(1+e^{-n})} \right) \left(\frac{1}{(1+e^{-n})} \right) \\ &= (1-a^1)(a^1)\end{aligned}$$

$$\begin{aligned}\dot{f}^2(n) &= \frac{d}{dn}(n) \\ &= 1\end{aligned}$$

6. คำนวณค่าความไว (Sensitivity) ของทุกชั้นย้อนกลับจากชั้นเอาต์พุตไปหาอินพุตจากกระบวนการ

$$S^M \rightarrow S^{M-1} \rightarrow \dots \rightarrow S^2 \rightarrow S^1$$

ชั้นเอาต์พุต หรือชั้นที่ 2

$$\begin{aligned}S^M &= -2\dot{F}^M(n^M)(d-a) \\ &= -2[\dot{f}^2(n^2)](1.261) \\ &= -2[1](1.261) \\ &= -2.522\end{aligned}$$

ชั้นที่ 1

$$\begin{aligned}S^m &= \dot{F}^m(n^m)(W^{m+1})^T S^{m+1} \\ S^1 &= \dot{F}^1(n^1)(W^2+1)^T S^2 \\ &= \begin{bmatrix} (1-a_1^1)(a_1^1) & 0 \\ 0 & (1-a_2^1)(a_2^1) \end{bmatrix} \begin{bmatrix} 0.09 \\ -0.17 \end{bmatrix} [-2.522] \\ &= \begin{bmatrix} (1-0.321)(0.321) & 0 \\ 0 & (1-0.368)(0.368) \end{bmatrix} \begin{bmatrix} 0.09 \\ -0.17 \end{bmatrix} [-2.522] \\ &= \begin{bmatrix} 0.218 & 0 \\ 0 & 0.233 \end{bmatrix} \begin{bmatrix} -0.227 \\ 0.429 \end{bmatrix} \\ &= \begin{bmatrix} -0.0495 \\ 0.0997 \end{bmatrix}\end{aligned}$$

7. ปรับปรุงค่าถ่วงน้ำหนัก

ให้อัตราการเรียนรู้ (α) = 0.1

ชั้นที่ 2

$$W^m(k+1) = W^m(k) - \alpha S^M (a^{m-1})^T$$

$$\begin{aligned} W^2(1) &= W^2(0) - \alpha S^2 (a^1)^T \\ &= [0.09 \quad -0.17] - 0.1[-2.522][0.321 \quad 0.368] \\ &= [0.171 \quad -0.0772] \end{aligned}$$

ชั้นที่ 1

$$\begin{aligned} W^1(1) &= W^1(0) - \alpha S^1 (a^0)^T \\ &= \begin{bmatrix} -0.27 \\ -0.41 \end{bmatrix} - 0.1 \begin{bmatrix} -0.0495 \\ 0.0997 \end{bmatrix} [1] \\ &= \begin{bmatrix} -0.265 \\ -0.420 \end{bmatrix} \end{aligned}$$

8. ปรับปรุงค่าถ่วงค่าไบอัส

ให้อัตราการเรียนรู้ (α) = 0.1

ชั้นที่ 2

$$b^m(k+1) = b^m(k) - \alpha S^M$$

$$\begin{aligned} b^2(1) &= b^2(0) - \alpha S^2 \\ &= [0.48] - 0.1[-2.522] \\ &= [0.732] \end{aligned}$$

ชั้นที่ 1

$$\begin{aligned} b^1(1) &= b^1(0) - \alpha S^1 \\ &= \begin{bmatrix} -0.48 \\ -0.13 \end{bmatrix} - 0.1 \begin{bmatrix} -0.0495 \\ 0.0997 \end{bmatrix} \\ &= \begin{bmatrix} -0.475 \\ -0.140 \end{bmatrix} \end{aligned}$$

ในที่นี้ได้แสดงวิธีทำให้แ่รอบเดียว จะต้องกระทำซ้ำไปเรื่อยๆจนกว่าค่าผิดพลาดอยู่ในชั้นยอมรับได้คือ $E < E_k$ จากนั้นเก็บค่าถ่วงน้ำหนักกับค่าไบอัสที่เป็นคำตอบสุดท้ายไว้ใช้งานในการประมาณค่า $g(p)$ เพื่อใช้กับโครงข่ายที่ออกแบบไว้

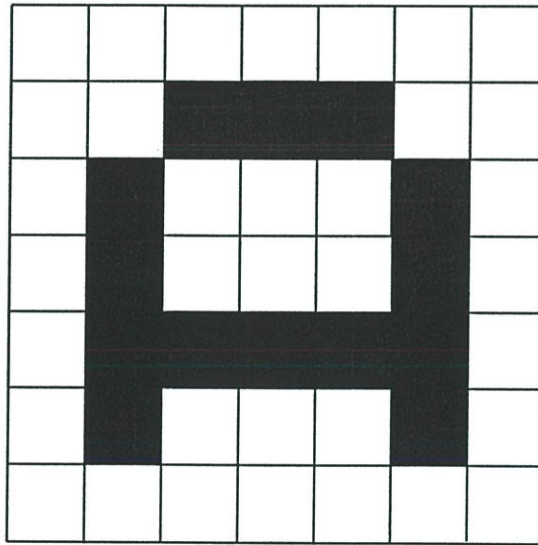
ในหัวข้อต่อไปเป็นการกล่าวถึงทฤษฎีที่ใช้ในการทำวิทยานิพนธ์นี้ซึ่งเป็นแบบจำลองโครงข่ายประสาทเทียมอีกแบบที่ใช้ในการจดจำตัวอักษรตัวพิมพ์คือ แอส โซซิเอทีฟแมม โมรี

3.3 แอสโซซิเอทีฟเมมโมรี

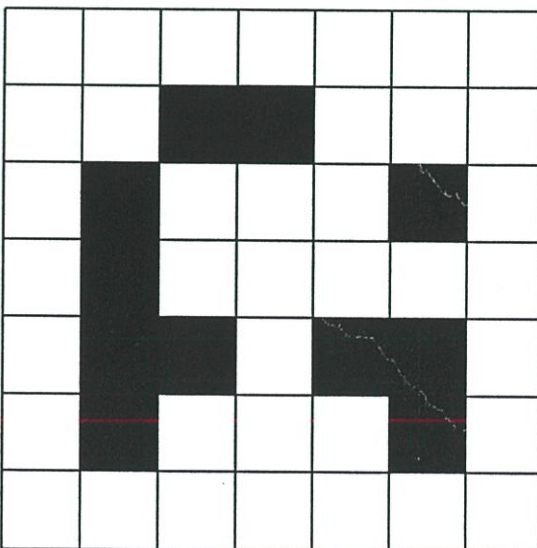
ในกระบวนการรู้จำของโครงข่ายประสาทเทียม ที่ต้องการแยกแยะข้อมูลต้องมีการวัดระยะทางเข้ามาเกี่ยวข้องเพื่อหากลุ่มของข้อมูลที่ใกล้ที่สุด ในกรณีของแอสโซซิเอทีฟเมมโมรีก็เช่นกัน แต่เนื่องจากว่าข้อมูลอินพุตเป็นรูปภาพสองระดับ ระยะทางที่นิยมใช้วัดของโครงสร้างนี้คือระยะทางแฮมมิง

3.3.1 ระยะทางแฮมมิง (Hamming Distance)

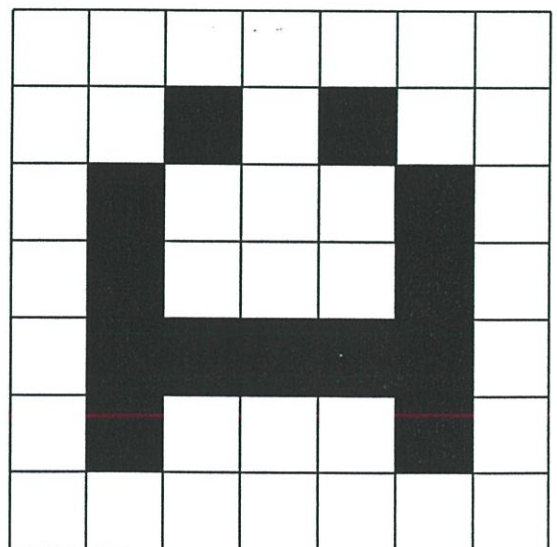
ระยะทางแฮมมิงเป็นการวัดจำนวนความแตกต่างของบิต ถ้ามีค่าน้อยแสดงว่าความแตกต่างน้อยแสดงว่ารูปภาพมีความเหมือนกัน ในทางกลับกันถ้าค่าระยะทางแฮมมิงมีค่ามากนั้นคือรูปภาพไม่เหมือนกัน ดังรูปที่ 3.16



(ก)



(ข)



(ค)

รูปที่ 3.16 ระยะทางแฮมมิง

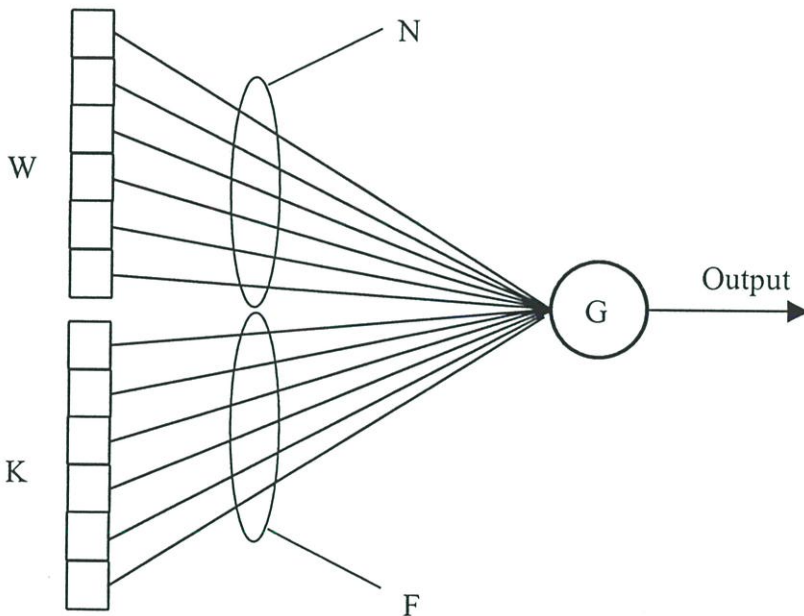
จากรูปที่ 3.16 (ก) เป็นรูปภาพต้นแบบ ส่วน (ข) และ (ค) เป็นรูปภาพที่ต้องการหาระยะทางแฮมมิงโดยใช้ (ก) เป็นตัวเปรียบเทียบได้

- ระยะทางแฮมมิงระหว่าง (ก) และ (ข) คือ 3

- ระยะทางแฮมมิงระหว่าง (ก) และ (ค) คือ 1

3.3.2 แอสโซซิเอทีฟเมมโมรี

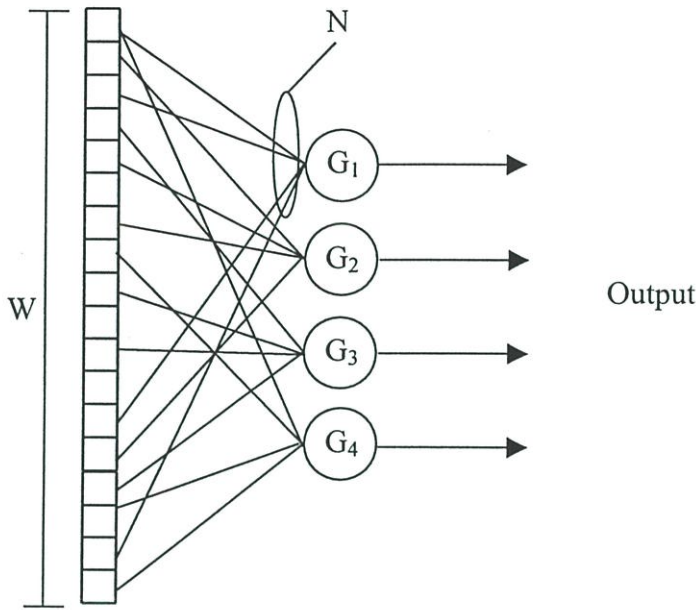
รูปที่ 3.17 เป็นโครงสร้างทั่วไปของแอสโซซิเอทีฟเมมโมรี โดยมีส่วนที่เป็นอินพุตที่ต่อกับข้อมูลของรูปภาพที่เป็นแบบมิติเดียว(N) จำนวนของบิตที่ต่อจากรูปภาพนั้นอาจจะต่อทั้งหมดทุกบิตของรูปภาพก็ได้แต่จะทำให้การทำงานช้าลง ดังนั้นในบางแบบจำลองจะกระทำการหาทิศทางที่มีความสำคัญเท่านั้นเท่านั้น เพื่อเป็นการลดจุดต่อของอินพุตลงจากนั้นค่าที่ได้จะนำเข้าไปในส่วนที่ใช้ในการตัดสินใจ และให้ผลลัพธ์ออกมาทางเอาต์พุต ผลลัพธ์บางส่วนจะถูกป้อนกลับ(F) ไปใช้ในการทำงานต่อไป



รูปที่ 3.17 โครงสร้างทั่วไปของแอสโซซิเอทีฟเมมโมรี

- เมื่อ
- W = จำนวนบิตอินพุตของรูปภาพ (External Input Field)
 - K = จำนวนบิตป้อนกลับ (Feedback Filed)
 - N = จำนวนจุดต่อจากรูปภาพอินพุตที่ป้อนเข้าสู่เซลล์ประสาท(Input Connection)
 - F = จำนวนจุดต่อจากการป้อนกลับที่ป้อนเข้าสู่เซลล์ประสาท (Feedback Connection)

3.3.3 GNU (General Neuron Unit)



รูปที่ 3.18 โครงสร้างของ GNU

เมื่อ W = จำนวนบิตอินพุตของรูปภาพ

N = จำนวนจุดต่อจากรูปภาพอินพุตที่ป้อนเข้าสู่เซลล์ประสาท

GNU เป็นอีกรูปแบบหนึ่งของแอสโซซิเอทีฟเมโมรีโดยโครงสร้างเป็นแบบ Feed Forward Single Layer ซึ่งมีโครงสร้างที่ไม่ซับซ้อน โดยในแต่ละเซลล์ประสาทจะมี Input Connection (N) ต่อกับ External Input Field จำนวนของ N นั้นสามารถกำหนดได้ ขึ้นอยู่กับการออกแบบ ในทางปฏิบัติ External Input Field ก็คือ รูปภาพนั่นเองจำนวนเซลล์ประสาทที่ต้องใช้ขึ้นอยู่กับจำนวนของเอาต์พุตที่ต้องการเป็นไปตามสมการที่ 3.25

$$A = 2^G \quad (3.25)$$

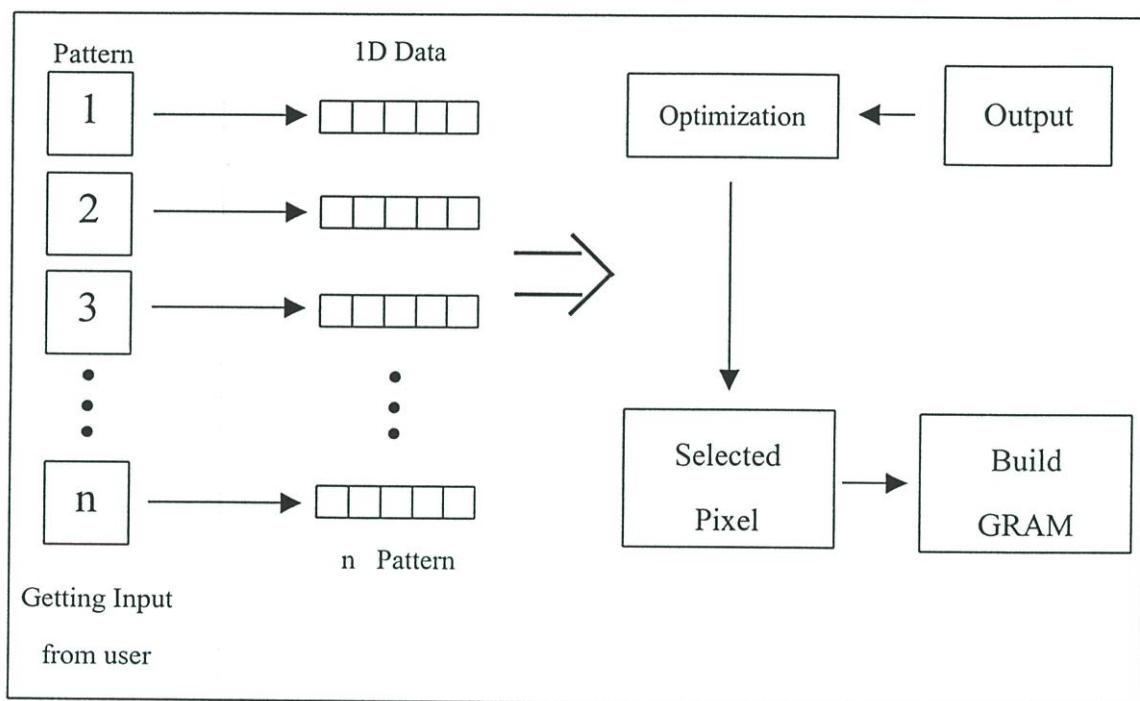
เมื่อ A = จำนวนของเอาต์พุตที่สามารถจดจำได้

G = จำนวนเซลล์ประสาทเทียม

3.3.3.1 การทำงานของ GNU (General Neuron Unit)

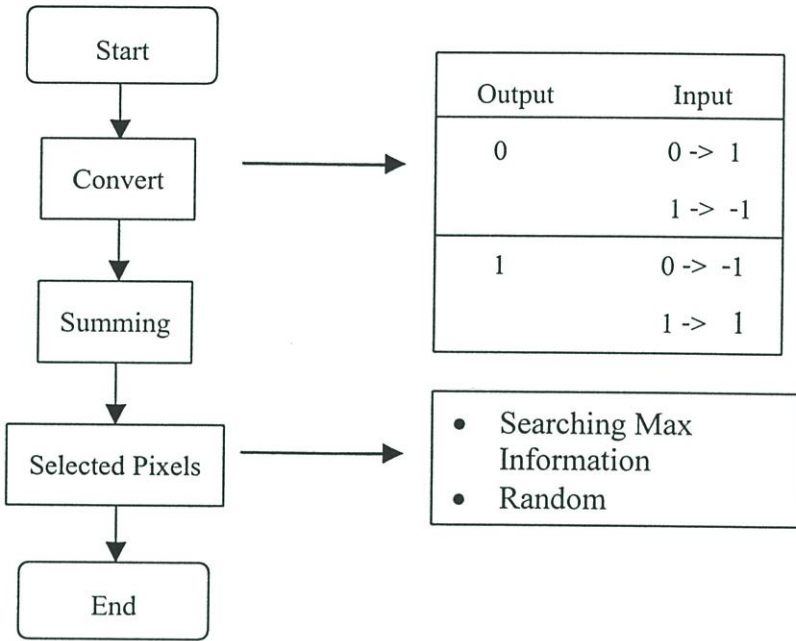
การทำงานของ GNU มีหลักการเหมือนกับโครงข่ายประสาทเทียมโดยทั่วไป กล่าวคือมีการเรียนรู้ และการเรียกคืน (Recall)

การเรียนรู้ของโครงสร้างนี้สามารถเขียนได้ดังรูปที่ 3.19 เนื่องจากโครงสร้างเป็นดังรูปที่ 3.18 เป็นโครงสร้างแบบไม่มีค่าน้ำหนัก(Weightless) จึงทำให้การเรียนรู้ทำได้อย่างรวดเร็ว

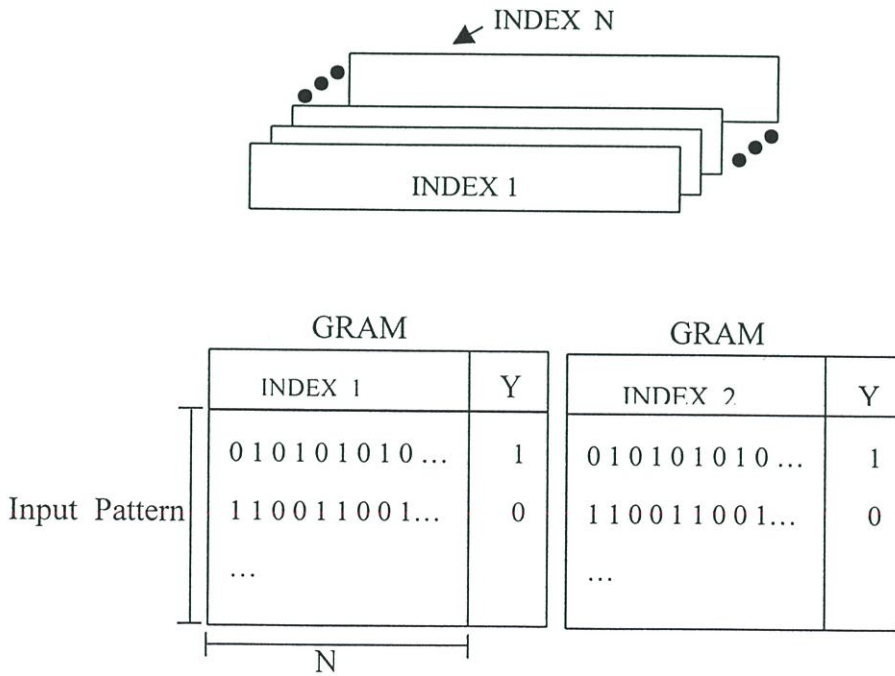


รูปที่ 3.19 การเรียนรู้ของ GNU

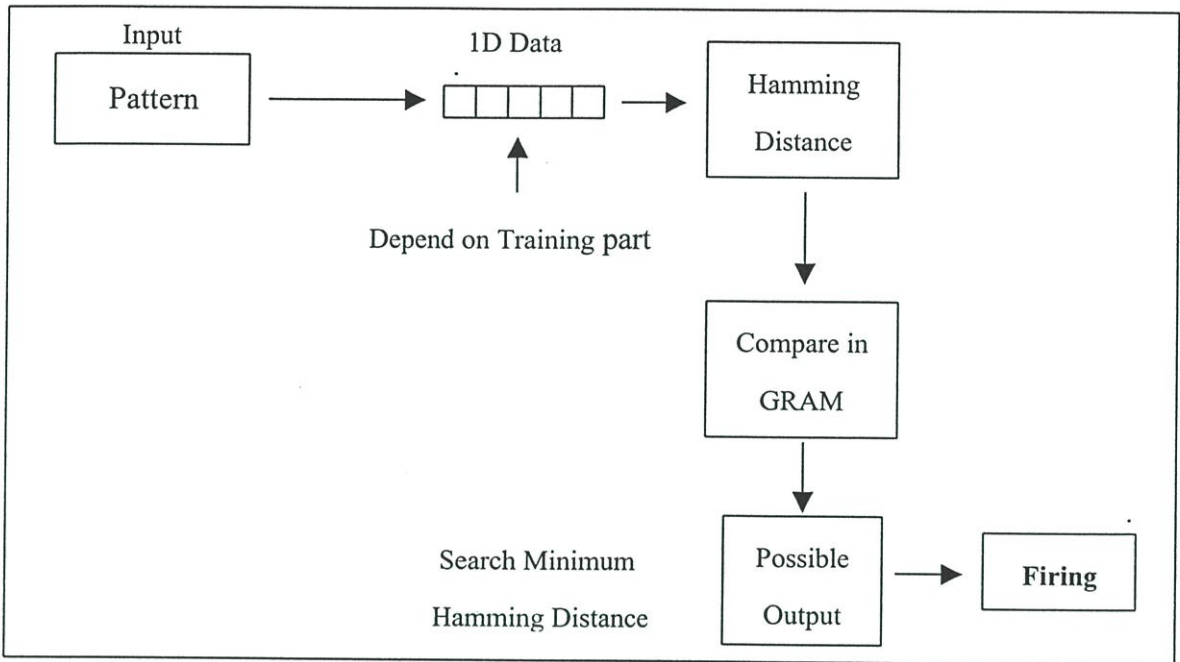
การเรียนรู้ของ GNU เริ่มต้นจากการนำรูปภาพมาตรฐานมาแปลงให้อยู่ในรูปแบบของข้อมูลแบบมิติเดียวกระบวนการนี้กล่าวไว้ในบทที่ 3 หลังจากการแปลงเป็นข้อมูลมิติเดียวต้องนำรูปภาพมาตรฐานทั้งหมดมาทำการ ออปติไมซ์ (Optimize) ซึ่งเป็นวิธีการในการเลือกตำแหน่งของภาพที่มีความสำคัญก่อนนำไปสร้าง GRAM ดังรูปที่ 3.20 หลังจากการออปติไมซ์ ค่าที่ได้เป็นค่าตำแหน่งของข้อมูล ตำแหน่งที่ได้ต้องนำไปสร้างเป็น GRAM ดังรูปที่ 3.21 เพื่อเก็บไว้สำหรับการเรียกคืนดังรูปที่ 3.22



รูปที่ 3.20 การออปติไมซ์



รูปที่ 3.21 การสร้าง GRAM



รูปที่ 3.22 การเรียกคืน (Recall)

3.4 ลักษณะเด่น (Feature)

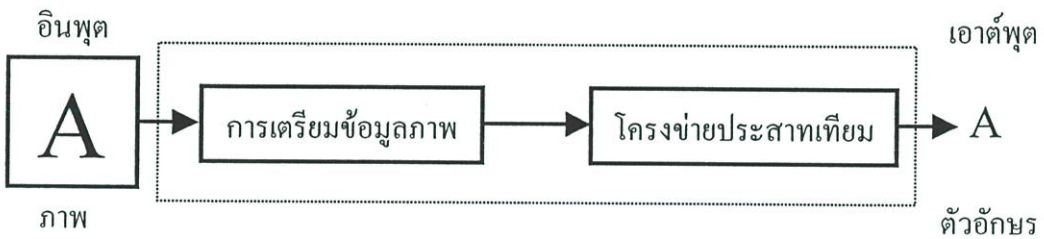
ในการหาลักษณะเด่นของตัวอักษรที่ใช้ในงานวิจัยนี้ จะหมายถึงการหา Entropy ของบิตของรูปภาพในรูปภาพต้นแบบ เพื่อจะนำบิตที่มีข้อมูลบรรจุอยู่สูงสุดมาสร้างเป็น GRAM มิได้หมายถึงลักษณะเด่นของตัวอักษร เช่น หัวของตัวอักษร, ส่วนเว้า, ส่วนโค้ง ของตัวอักษรแต่อย่างใดซึ่งการหา Entropy คือการ การออปติไมซ์ ที่กล่าวไปแล้วนั่นเอง

บทที่ 4

การทดลอง

จากรูปที่ 4.1 เป็นระบบที่ถูกสร้างขึ้นเพื่อใช้ในการทดสอบแนวความคิดของ GNU โดยประกอบไปด้วยส่วนที่เป็นอินพุตที่ป้อนเข้าเป็นรูปภาพ ผ่านกระบวนการเตรียมข้อมูลภาพ เพื่อปรับเข้าสู่ตำแหน่งมาตรฐาน แล้วส่งเข้าโครงข่ายประสาทเทียมเพื่อแยกแยะตัวอักษร โดยข้อกำหนดเบื้องต้นที่ได้ตั้งไว้มีดังนี้

1. เป็นตัวอักษรตัวพิมพ์ ภาษาอังกฤษ A-Z
2. เป็นภาพบิตแมป (Bitmap) แบบ 2 ระดับ ขนาด 64×64 จุดภาพ
3. ขนาดของตัวอักษรต้องมีความใกล้เคียงกัน
4. รูปภาพที่นำมาทดสอบต้องเป็นตัวอักษรที่สมบูรณ์ แต่อาจไม่จำเป็นต้องอยู่ที่ตำแหน่งมาตรฐาน

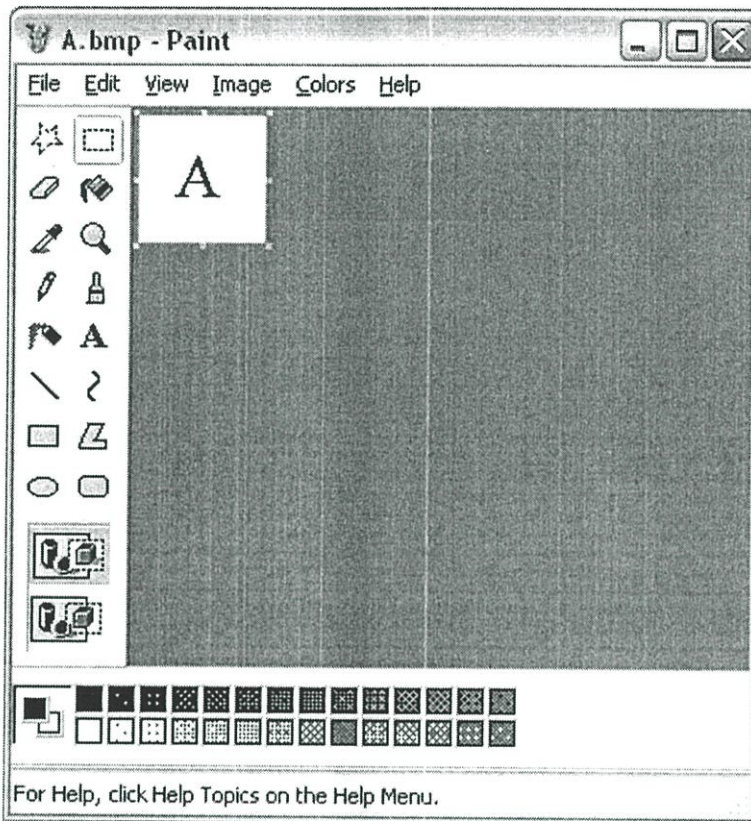


รูปที่ 4.1 ระบบของการจำตัวอักษร

ข้อกำหนดที่สร้างขึ้น เป็นข้อมูลเบื้องต้นที่จำเป็นซึ่งต้องนำไปออกแบบการทำงานในส่วนต่างๆ ตั้งแต่การเตรียมรูปภาพต้นแบบ การเตรียมข้อมูลภาพ และการออกแบบโครงข่ายประสาทเทียม

4.1 รูปภาพต้นแบบ

การสร้างรูปภาพต้นแบบใช้โปรแกรมวาดรูปทั่วไปในการสร้างในที่นี้ใช้โปรแกรม Paint ในการสร้างรูปภาพต้นแบบดังรูปที่ 4.1 โดยกำหนดรูปแบบของไฟล์เป็นแบบบิตแมปและใช้สีขาวเป็นพื้นตัวอักษรเป็นสีดำนั้นหมายถึง สีขาวให้แทนข้อมูลด้วย 0 และสีดำแทนข้อมูลด้วย 1 ในตารางที่ 4.1 เป็นรูปภาพต้นแบบบางส่วนที่ได้สร้างไว้เพื่อนำไปใช้ต่อไป



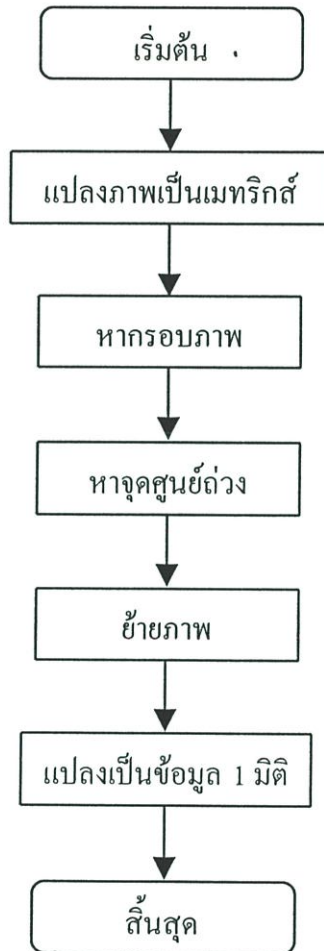
รูปที่ 4.2 การสร้างรูปภาพต้นแบบ 2 ระดับ

ตารางที่ 4.1 รูปภาพต้นแบบ(บางส่วน)

	ตัวอักษร				
แบบที่ 1	A	B	C	D	E
แบบที่ 2	A	B	C	D	E
แบบที่ 3	A	B	C	D	E

4.2 การเตรียมข้อมูลภาพ

การเตรียมข้อมูลภาพ เป็นการหาตำแหน่งมาตรฐานของตัวอักษร โดยการเรียนรู้ของโครงข่ายประสาทเทียมต้องผ่านการหาตำแหน่งมาตรฐานด้วยเช่นกัน เพื่อให้ได้ความถูกต้องสูงสามารถเขียนเป็นขั้นตอนการออกแบบได้ดังรูปที่ 4.3



รูปที่ 4.3 การสร้างรูปภาพต้นแบบ 2 ระดับ

การแปลงภาพบิตแมปเป็นเมทริกสนั้นกล่าวไว้แล้วในบทที่ 3 โดยใช้สมการที่ 3.1 การหากรอบภาพมีตัวแปรที่ต้องทราบค่าก่อนนำไปเขียนโปรแกรมคือขนาดของรูปภาพ ซึ่งได้กำหนดไว้ในข้อกำหนดเบื้องต้นคือ 64×64 จุดภาพจากนั้นกระทำการ หากรอบโดยใช้ตัวแปรดังนี้

- กรอบบน (UB)
- กรอบล่าง (DB)
- กรอบซ้าย (LB)
- กรอบขวา (RB)

เมื่อได้กรอบภาพนำค่าที่ได้หาจุดศูนย์กลางถ่วงของรูปภาพโดยใช้สมการที่ 3.3 และ 3.4 ค่าที่ได้จากการหาจุดศูนย์กลางถ่วงคือ I_x และ I_y กระบวนการต่อไปทำการย้ายรูปภาพสู่ตำแหน่งมาตรฐานโดย

$$a = 64$$

$$b = 64$$

ดังนั้นตำแหน่งมาตรฐานที่ได้ออกแบบไว้คือ

$$\frac{a}{2} = 32$$

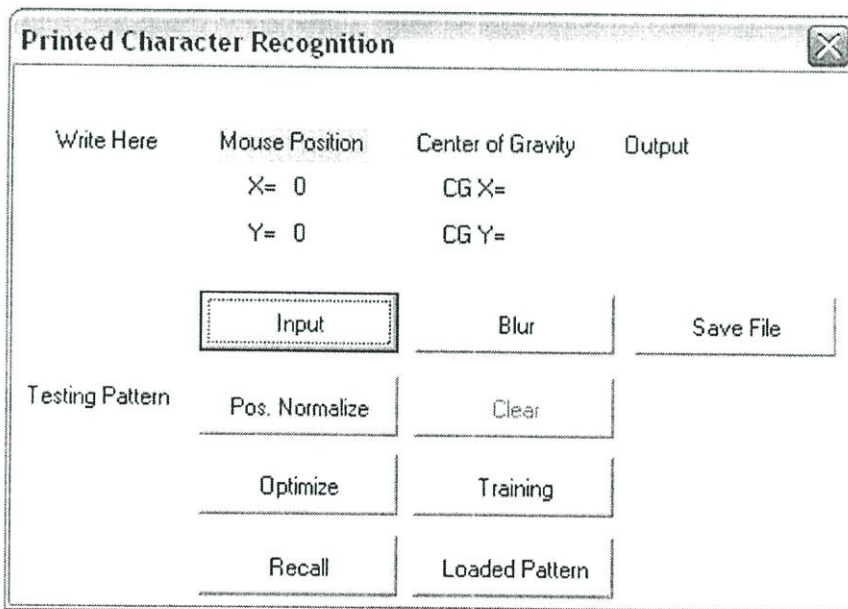
$$\frac{b}{2} = 32$$

การย้ายภาพไปยังตำแหน่งมาตรฐานใช้สมการที่ 3.5 ทำการย้ายดังรูปที่ 4.5 จากนั้นให้แปลงข้อมูลรูปภาพไปเป็นข้อมูลแบบ 1 มิติ โดยใช้กระบวนการดังรูปที่ 3.9 หลังจากการผ่านกระบวนการนี้ข้อมูลก็พร้อมที่จะป้อนเข้าโครงข่ายประสาทเทียมเพื่อทำการต่อไป

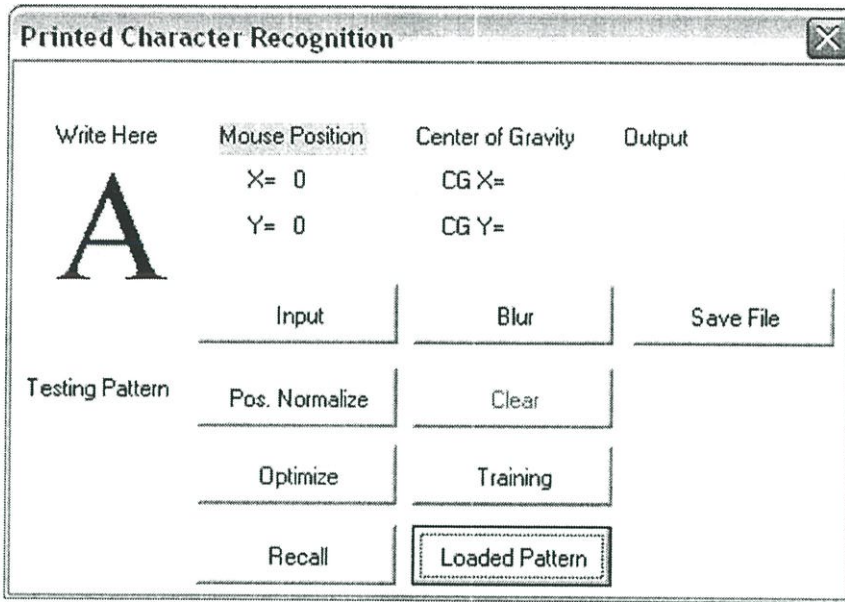
ในรูปที่ 4.5 (ก) เป็นการนำรูปภาพบิตแมป แบบ 2 ระดับเข้ามาทำการทดสอบโดยรูปภาพที่นำมาทดสอบเป็นรูปภาพตัวอักษร A ที่ยังไม่ได้ปรับตำแหน่งมาตรฐานอินพุตถูกนำเข้ามาแสดงในตำแหน่ง Write Here จากนั้นกดปุ่ม Pos. Normalize ปุ่มนี้กระทำการตามรูปที่ 4.3 ผลลัพธ์ที่ได้มีด้วยกัน 2 ส่วน คือ ส่วนที่เป็นการหาจุดศูนย์กลางถ่วงของรูปภาพ ณ ตำแหน่ง Testing Pattern ณ ตำแหน่งนี้ยังไม่ทำการย้ายภาพ และแสดงเส้นกรอบของรูปภาพที่หาได้ ค่าที่คำนวณได้เป็นค่าจุดศูนย์กลางถ่วงดังนี้

$$I_x = CG X$$

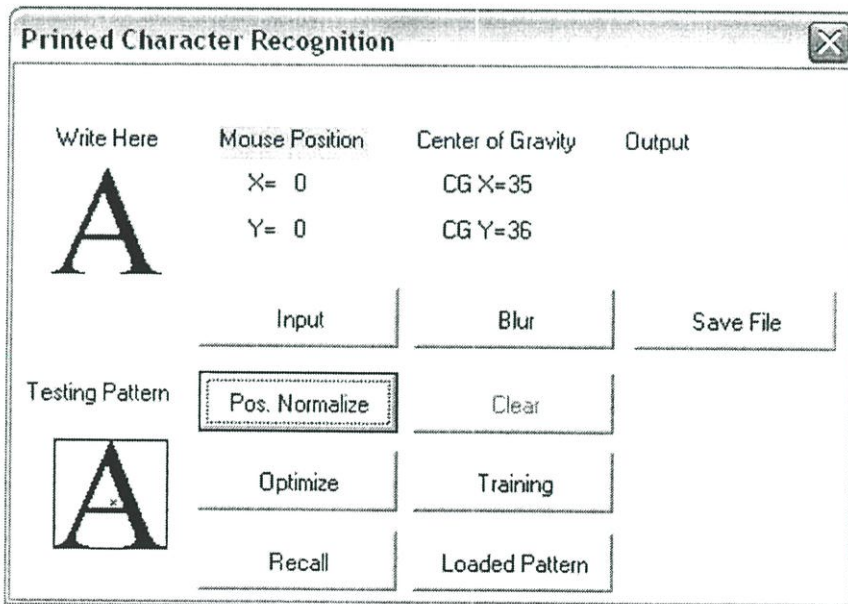
$$I_y = CG Y$$



รูปที่ 4.4 โปรแกรมในสถานะที่รอรับคำสั่ง



(ก)



(ข)

รูปที่ 4.5 การเตรียมข้อมูล

(ก) รูปภาพที่ยังไม่ปรับตำแหน่งมาตรฐาน

(ข) รูปภาพที่ปรับเข้าสู่ตำแหน่งมาตรฐานแล้ว

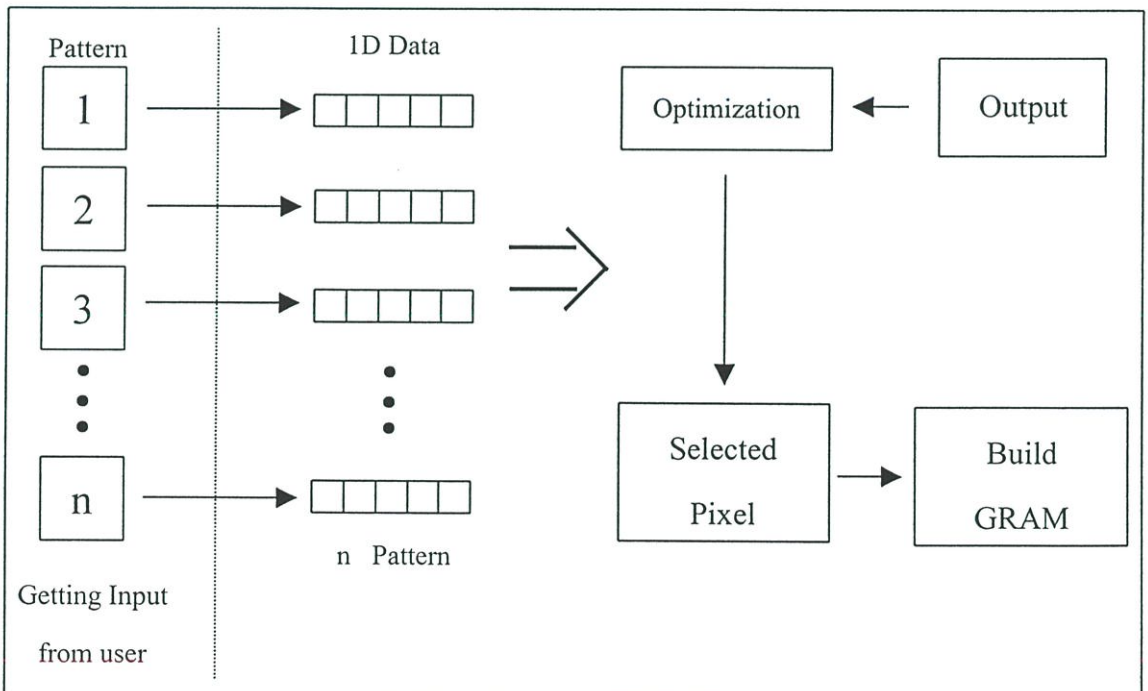
4.3 การออกแบบโครงข่ายประสาทเทียม

ส่วนที่ต่อจากการเตรียมข้อมูลภาพเป็นกระบวนการของโครงข่ายประสาทเทียม ซึ่งมีอยู่ด้วยกันหลายแบบอย่างที่กล่าวไว้ในบทที่ 2 ในที่นี้เลือกใช้แบบ GNU มีเหตุผลดังนี้

1. เป็น โครงสร้างที่ไม่ซับซ้อน ออกแบบง่าย
2. ใช้เวลาในการเรียนรู้สั้น
3. ใช้ระยะทางแฮมมิงซึ่งเหมาะสมกับรูปภาพสองระดับ

กระบวนการเรียนรู้แสดงได้ดังรูปที่ 4.5 หัวใจสำคัญคือการออปติไมซ์เพื่อหาดำแหน่งของข้อมูลที่เหมาะสมที่สุดกับรูปภาพต้นแบบที่กำหนดให้ ซึ่งรูปภาพต้นแบบทั้งหมดที่ใช้สามารถหาได้ดังนี้

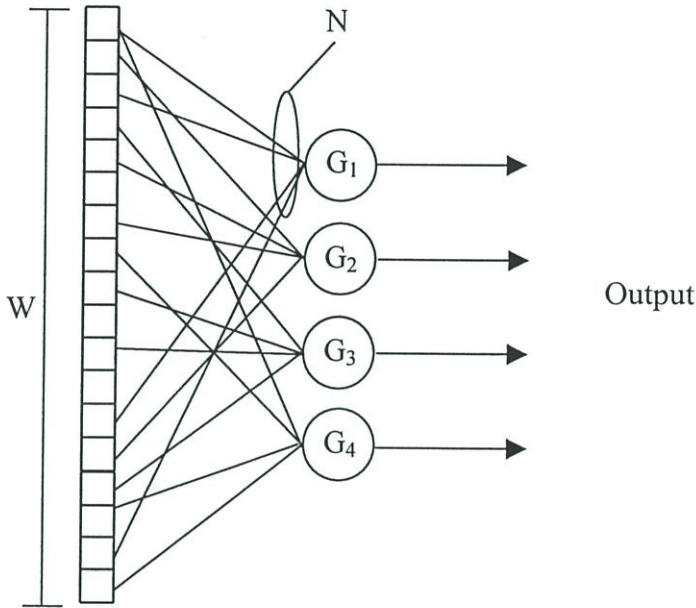
$$\begin{aligned}
 \text{จำนวนรูปภาพต้นแบบ (n)} &= 26 \times \text{จำนวนแบบ} \\
 &= 26 \times 3 \\
 &= 78 \text{ รูปแบบ} \\
 \text{จำนวนเอาต์พุตที่ต้องการ} &= 26 \text{ รูปแบบ}
 \end{aligned}$$



รูปที่ 4.6 การเรียนรู้ของ GNU

4.3.1 การออกแบบการออปติไมซ์

สามารถทำได้โดยต้องกำหนด เอาต์พุตของข้อมูลของและจำนวนเซลล์ประสาทเทียม พิจารณารูปที่ 4.7 จำนวนบิตของเอาต์พุตคือจำนวนเซลล์ประสาทดังนั้นสามารถกำหนดเอาต์พุตที่ สัมพันธ์กับรูปแบบมาตรฐานได้ดังตารางที่ 4.2



รูปที่ 4.7 โครงสร้างของ GNU

ตารางที่ 4.2 รูปแบบของเอาต์พุตข้อมูล (บางส่วน)

	ข้อมูลประจำตัวอักษร													
A	1111	1111	1111	1111	1111	1111	1111	1111	1111	1111	1111	1111	1111	1111
	0000	0000	0000	0000	0000	1111	1111	1111	1111	1111	0000	0000	0000	0000
	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
B	0000	1111	1111	1111	1111	1111	1111	1111	1111	1111	1111	1111	1111	1111
	1111	1111	1111	1111	1111	0000	0000	0000	0000	0000	1111	1111	1111	1111
	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
C	0000	1111	1111	1111	1111	1111	1111	1111	1111	1111	1111	1111	1111	1111
	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
	1111	1111	1111	1111	1111	1111	1111	1111	1111	1111	1111	1111	1111	1111

ชุดจำแนก Pattern

จากตารางที่ 4.1 การออกแบบส่วนของรหัสนั้นประกอบด้วยสองส่วนคือชุดจำแนก Pattern และส่วนของชุดประจำตัวอักษร โดยชุดจำแนก Pattern ได้ออกแบบให้ ตั้งฉากกันเพื่อให้สามารถแยกแยะกลุ่มของข้อมูลได้อย่างมีประสิทธิภาพเพิ่มขึ้น ส่วนชุดประจำตัวอักษรจะเหมือนกันหมดในแต่ละรูปแบบของตัวอักษร

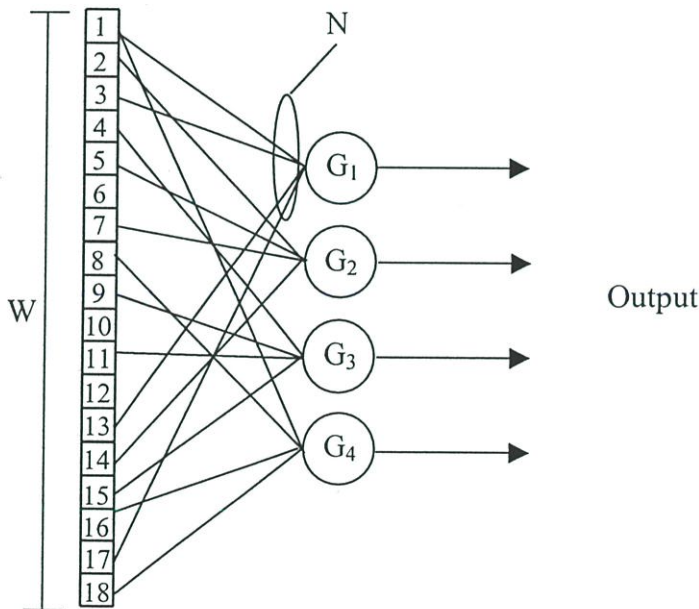
หลังจากได้ข้อมูลอินพุตที่เป็นรูปภาพ และเอาต์พุตที่กำหนดขึ้นมา กระบวนการต่อไปนำสองส่วนนี้ไปทำการการถอดโมไซค์ตามวิธีการดังรูปที่ 2.20 เมื่อผ่านกระบวนการนี้จะได้ตำแหน่งที่ต้องต่อกับข้อมูลอินพุตที่เป็นแบบมิตติเดียวดังนี้

G_1 ต่อกับ บิตที่ 1, 3, 13, 17

G_2 ต่อกับ บิตที่ 2, 5, 7, 14

G_3 ต่อกับ บิตที่ 4, 9, 11, 15

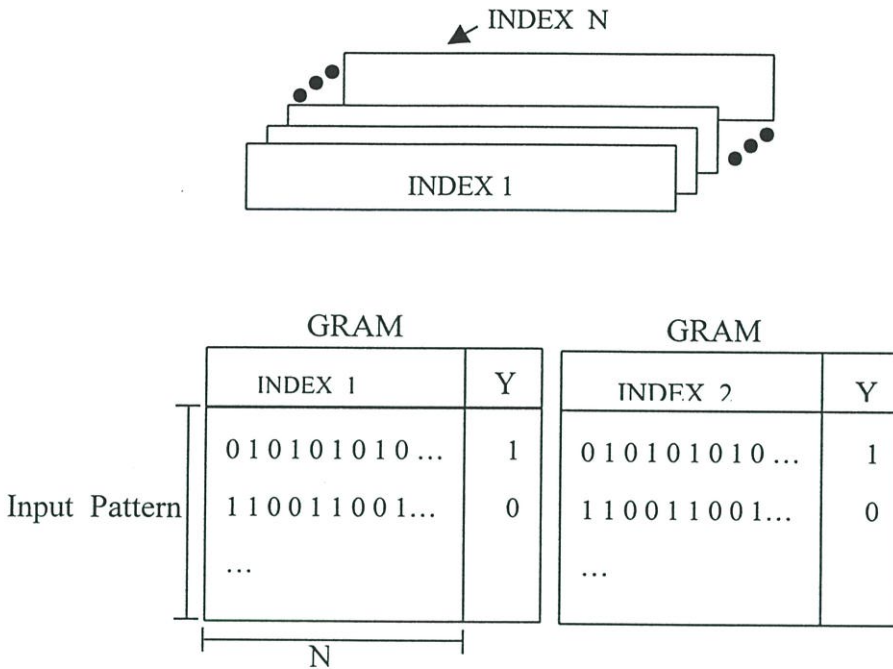
G_4 ต่อกับ บิตที่ 1, 8, 16, 18



รูปที่ 4.8 การต่อ Input Connection (N)

จำนวน N ในแต่ละเซลล์อาจออกแบบให้เท่ากันหรือไม่ก็ได้ แต่เพื่อให้ง่ายต่อการออกแบบได้ออกแบบให้จำนวน N ในทุกๆ เซลล์ เท่ากันหมด

4.3.2 การออกแบบGRAM



รูปที่ 4.9 การสร้าง GRAM

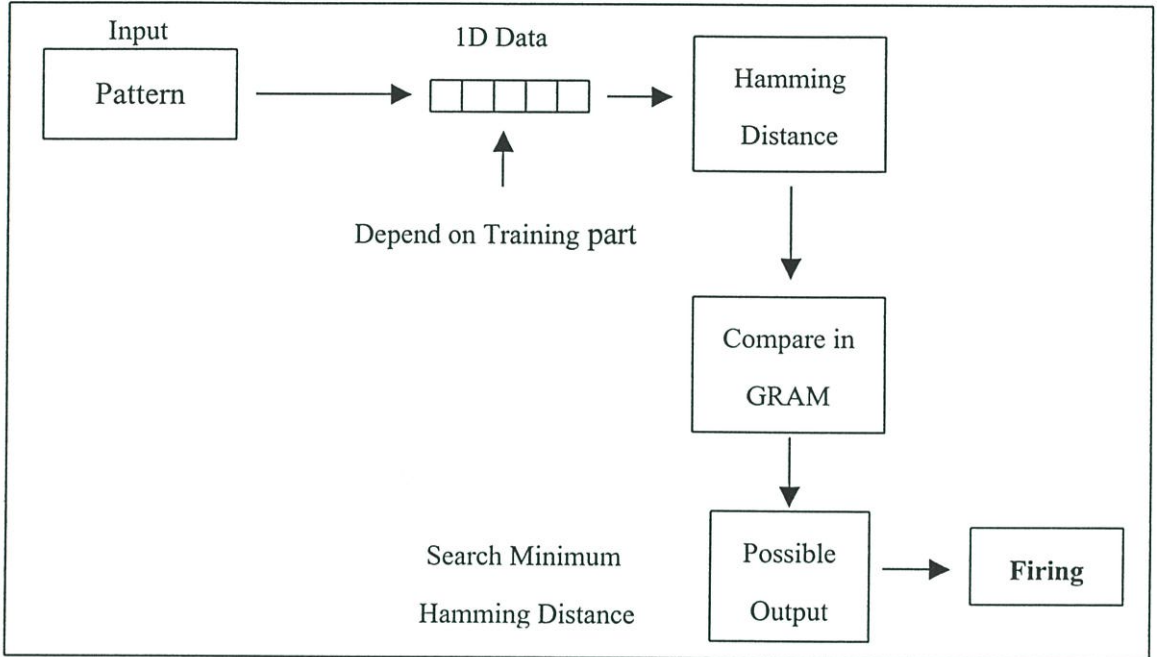
การออกแบบ GRAM ต้องกำหนดส่วนประกอบ 2 ส่วนที่สำคัญคือ INDEX และส่วนของเอาต์พุต โดยส่วนของ INDEX คือตำแหน่งของข้อมูลที่ได้กระทำการอุปติไมซ์ จำนวนของ INDEX คือค่าของ N อีกส่วนหนึ่งคือส่วนของเอาต์พุตที่เป็นไปตามค่าที่ได้ออกแบบไว้ดังตารางที่ 4.1 จำนวนของ GRAM คือจำนวนของเซลล์ประสาท

ตารางที่ 4.3 ข้อมูลของ GRAM ที่ได้สร้างขึ้นตามตำแหน่งที่ได้จากการอุปติไมซ์

INDEX 1									Output
1	3	4	6	8	20	57	65	89	
1	1	1	0	1	0	1	1	0	1
1	0	0	1	0	1	1	0	1	1
0	1	1	1	1	0	0	0	0	0
0	0	0	1	0	1	1	1	0	0
1	1	0	1	1	0	0	0	1	1
0	1	0	1	0	1	0	1	1	1
0	1	1	1	1	0	0	1	0	1
0	0	1	1	0	1	0	1	0	1

4.3.3 การเรียกคืน (Recall)

หลังจากการเรียนรู้ตามวิธีการดังกล่าวแล้วจะได้ส่วนที่เป็น GRAM เพื่อใช้ในการเรียกคืน โดยมีวิธีการดังรูปที่ 4.9



รูปที่ 4.10 การเรียกคืน (Recall)

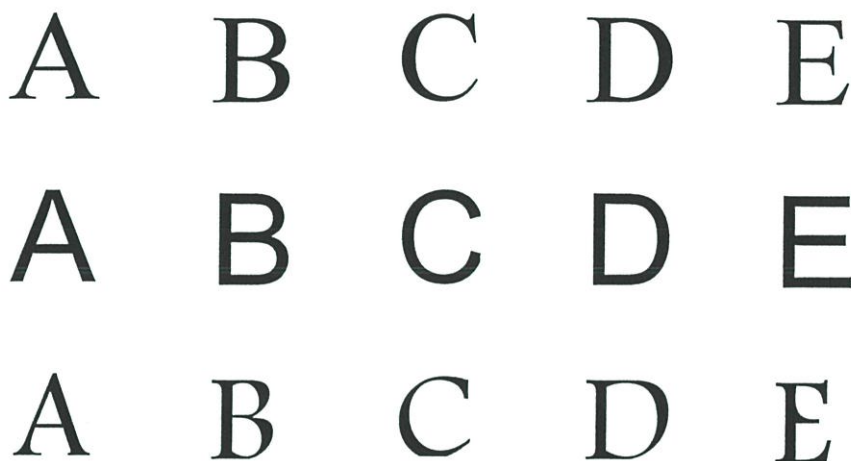
ในกระบวนการเรียกคืนนั้นต้องทำการแปลงข้อมูลเป็นแบบมิติเดียวก่อน ข้อมูลแบบมิติเดียวที่ป้อนเข้าโครงข่ายประสาทเทียม จะถูกต่อผ่าน Input Connection ตำแหน่งเดียวกันกับที่ได้ปฏิบัติมาซึ่งไว้แล้วในช่วงของการเรียนรู้

จากนั้นทำการหาระยะทางแฮมมิงว่าข้อมูลที่รับเข้ามาตรงกับข้อมูลที่อยู่ใน GRAM ชุดใดก็ทำการ Firing ออกมา หากว่าข้อมูลนั้นไม่ตรงกับข้อมูลที่อยู่ GRAM ค่าระยะทางแฮมมิงที่น้อยที่สุด อยู่ ณ ตำแหน่งใดก็จะให้ค่านั้นออกมา

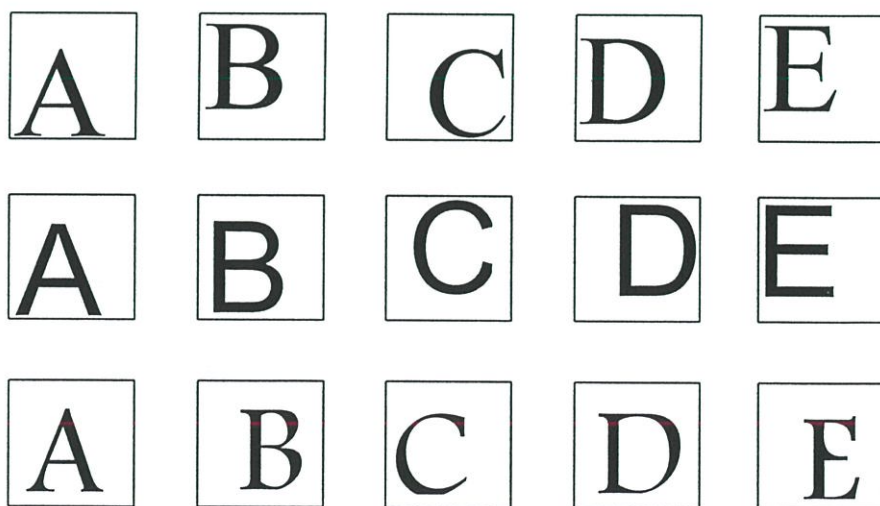
4.4 การทดสอบ

การทดสอบแบ่งออกได้ดังนี้

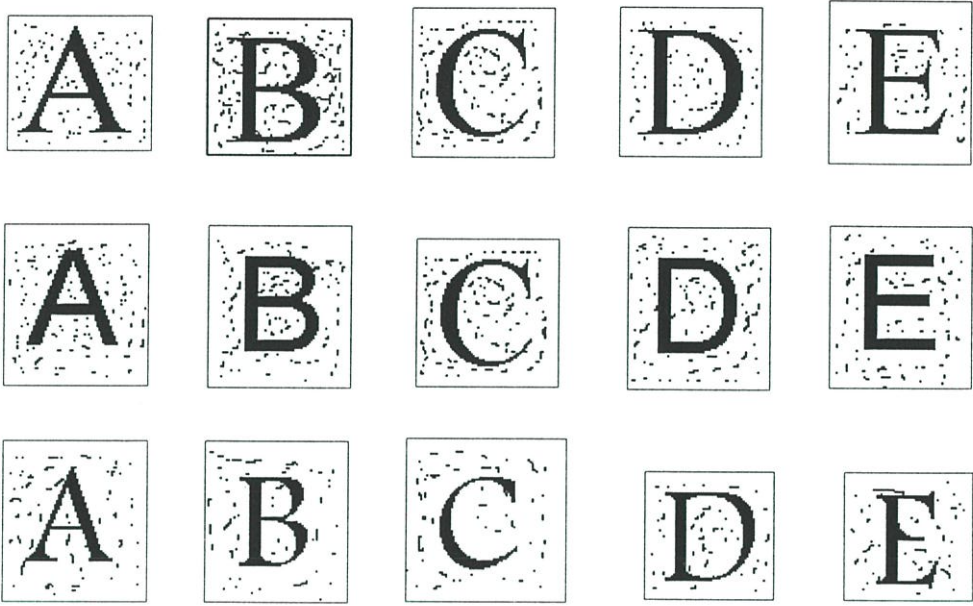
1. กลุ่มรูปภาพที่เหมือนกับการเรียนรู้ทุกประการ
2. กลุ่มรูปภาพที่มีตำแหน่งของภาพไม่อยู่ในตำแหน่งมาตรฐาน
3. กลุ่มรูปภาพที่มีสัญญาณรบกวน
4. กลุ่มรูปภาพที่มีตัวอักษรไม่สมบรูณ์



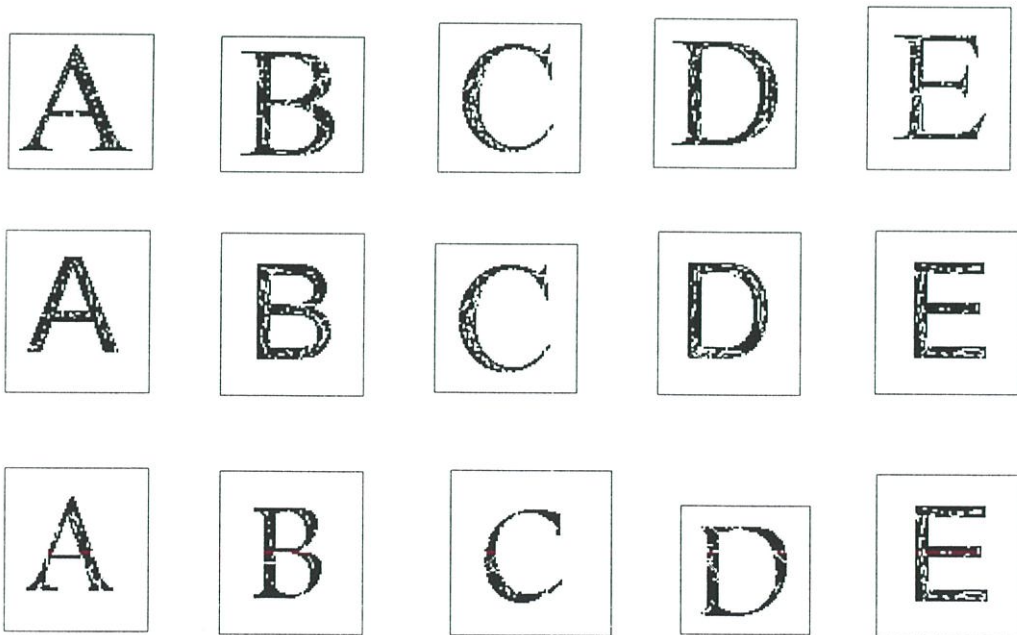
รูปที่ 4.11 กลุ่มรูปภาพที่เหมือนกับการเรียนรู้ทุกประการ



รูปที่ 4.12 กลุ่มรูปภาพที่มีตำแหน่งของภาพไม่อยู่ในตำแหน่งมาตรฐาน



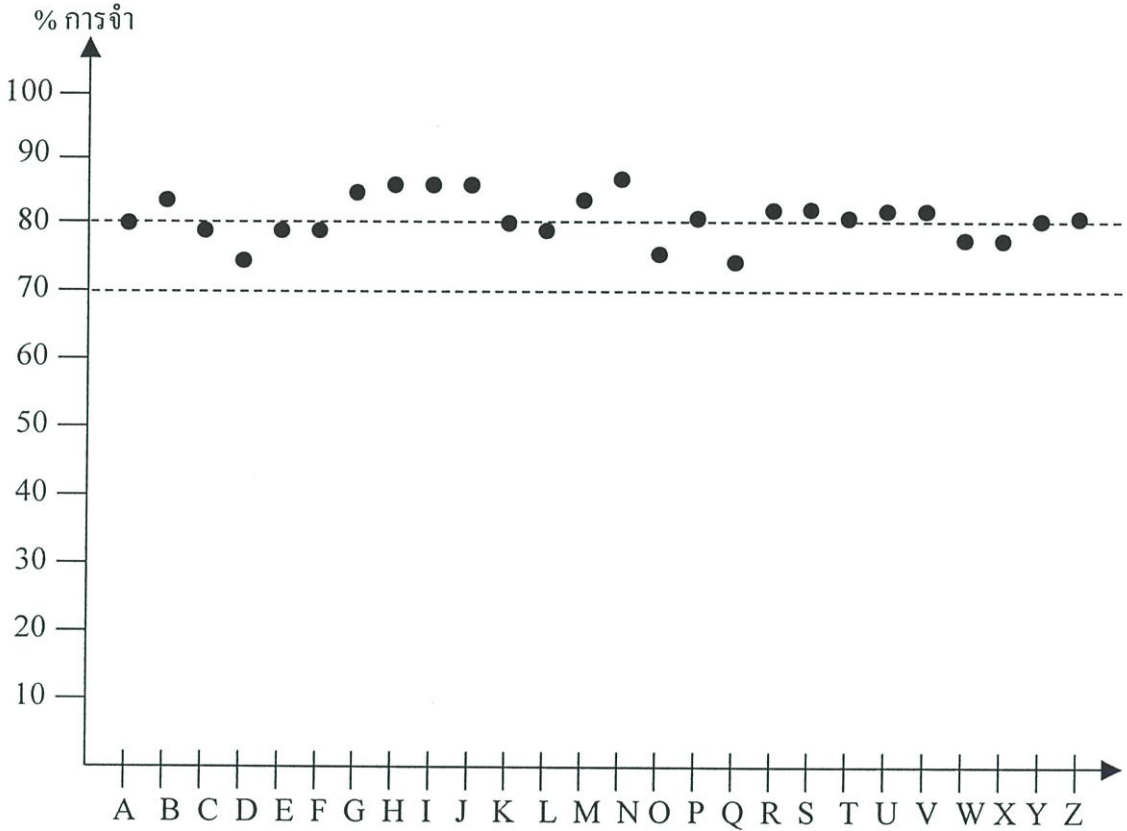
รูปที่ 4.13 กลุ่มรูปภาพที่มีสัญญาณรบกวน



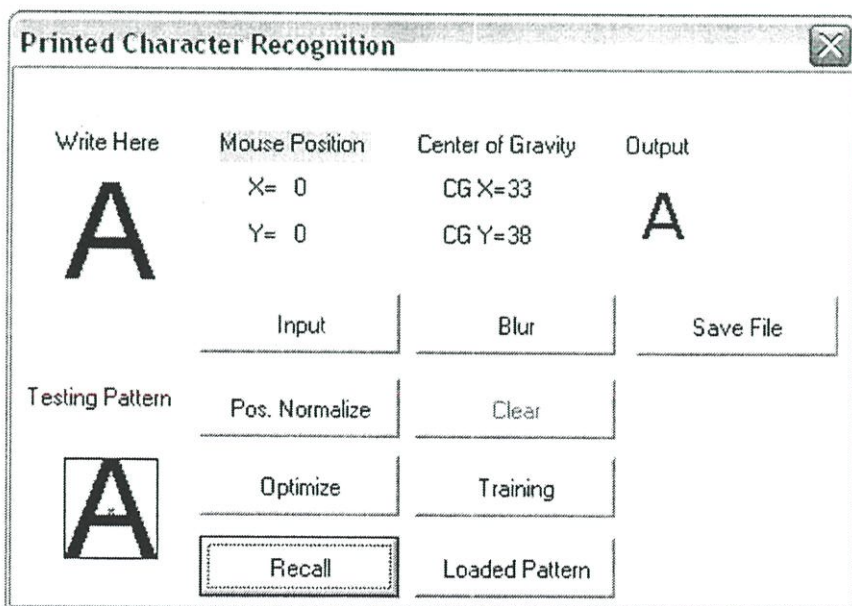
รูปที่ 4.14 กลุ่มรูปภาพที่มีตัวอักษร ไม่สมบรูณ์

4.5 ผลการทดสอบ

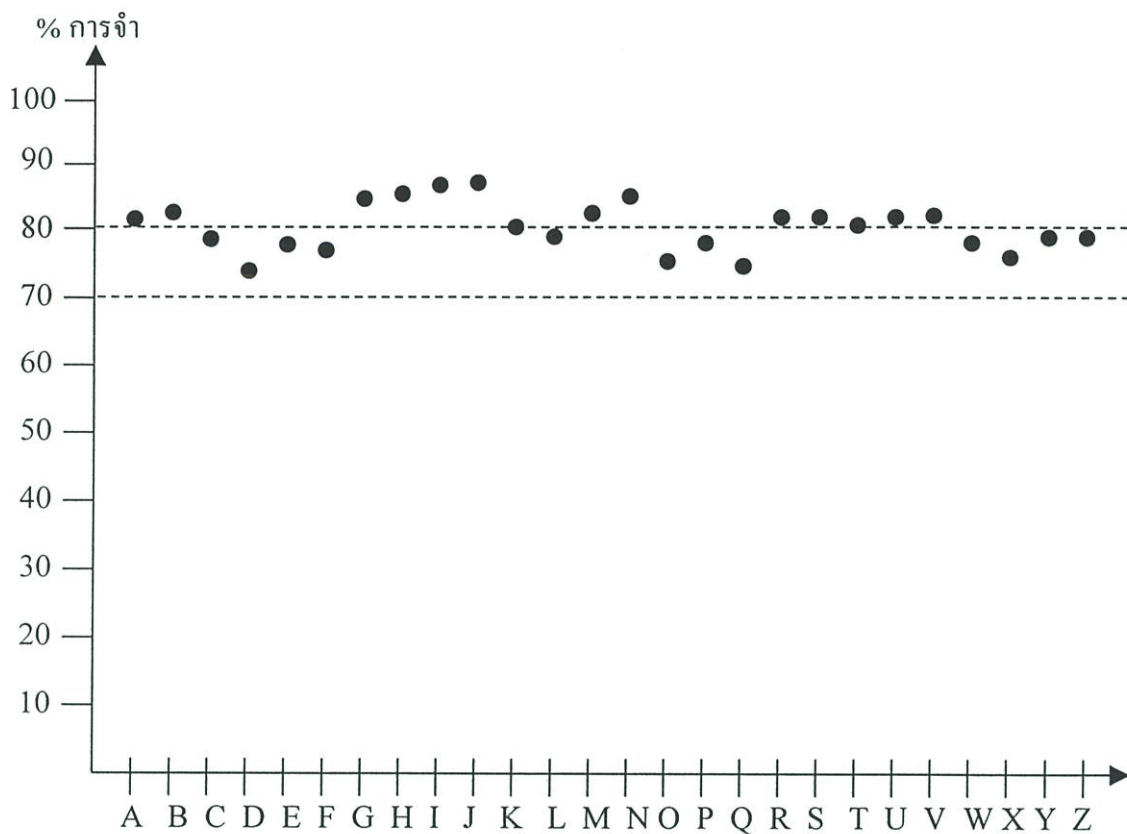
ผลการทดสอบตัวอักษรทั้ง 4 กลุ่ม แสดงดังกราฟที่ 4.15 โดยเป็นการแสดงความสัมพันธ์ระหว่างเปอร์เซ็นต์การจำกับตัวอักษรภาษาอังกฤษตั้งแต่ A-Z โดยรูปภาพอินพุตในแต่ละการฟังจะถูกป้อนเข้าระบบที่ได้ออกแบบไว้แล้ว



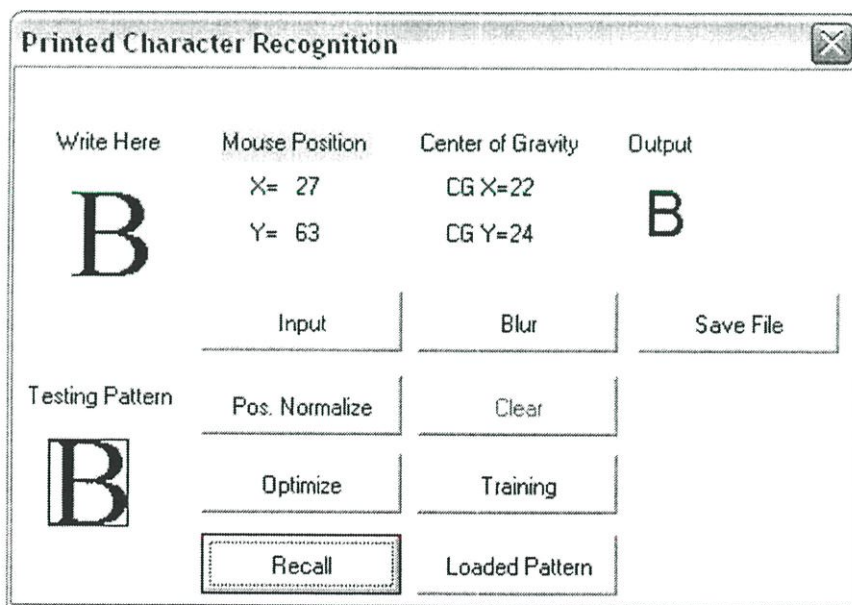
รูปที่ 4.15 ประสิทธิภาพการจำของกลุ่มรูปภาพที่เหมือนกับการเรียนรู้ทุกประการ



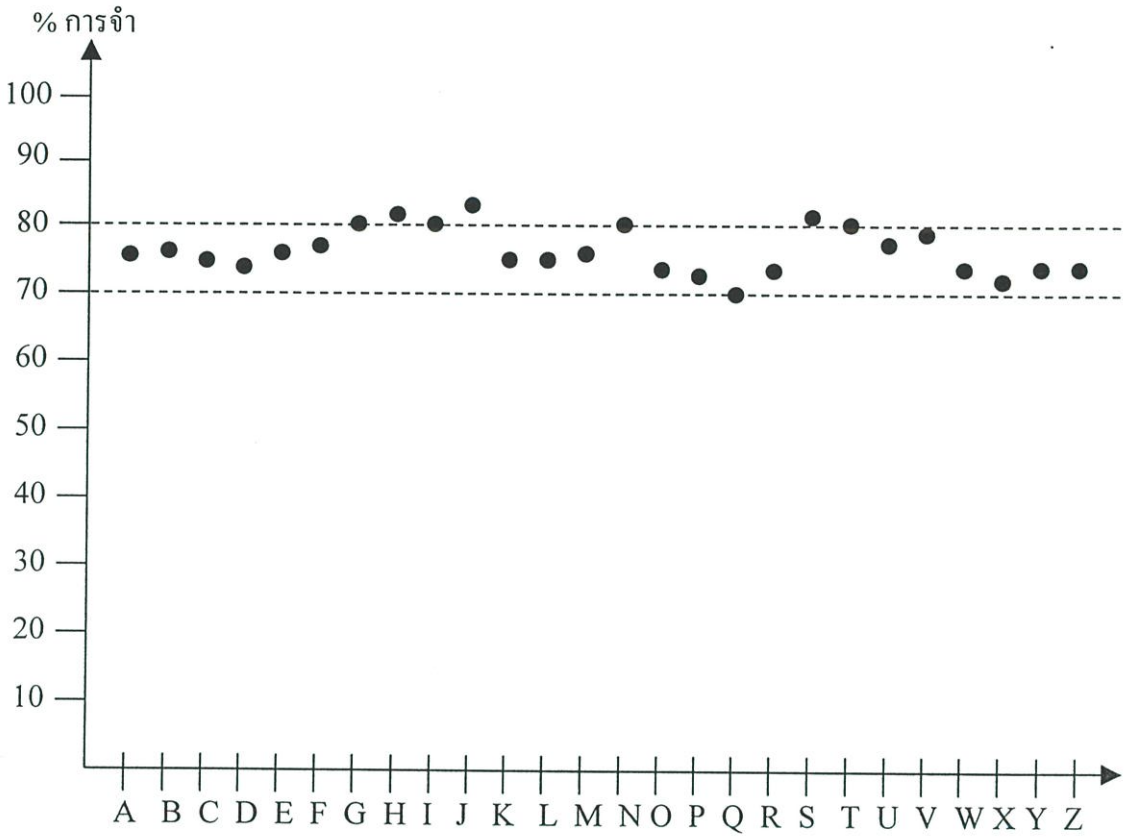
รูปที่ 4.16 ผลลัพธ์ของโปรแกรมกลุ่มรูปภาพที่เหมือนกับการเรียนรู้ทุกประการ



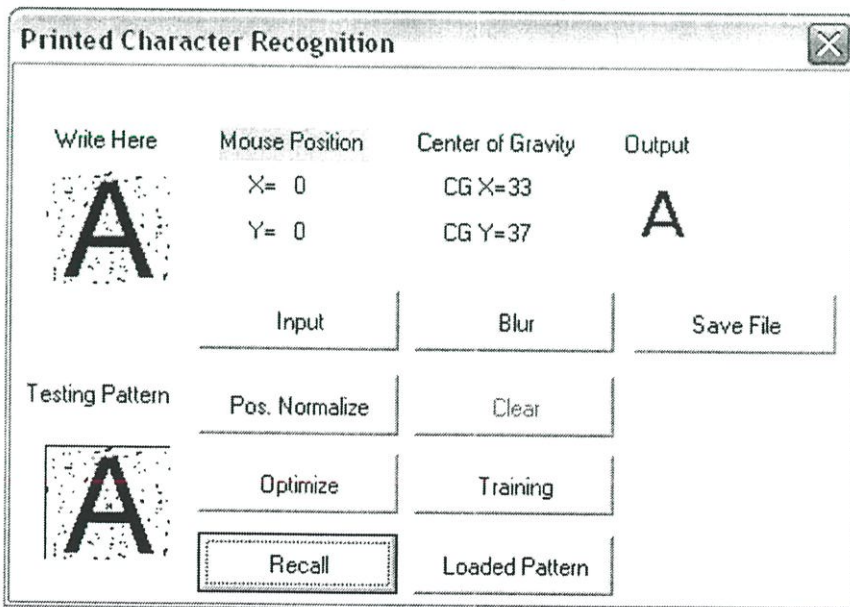
รูปที่ 4.17 ประสิทธิภาพการจำของกลุ่มรูปภาพที่มีตำแหน่งของภาพไม่อยู่ในตำแหน่งมาตรฐาน



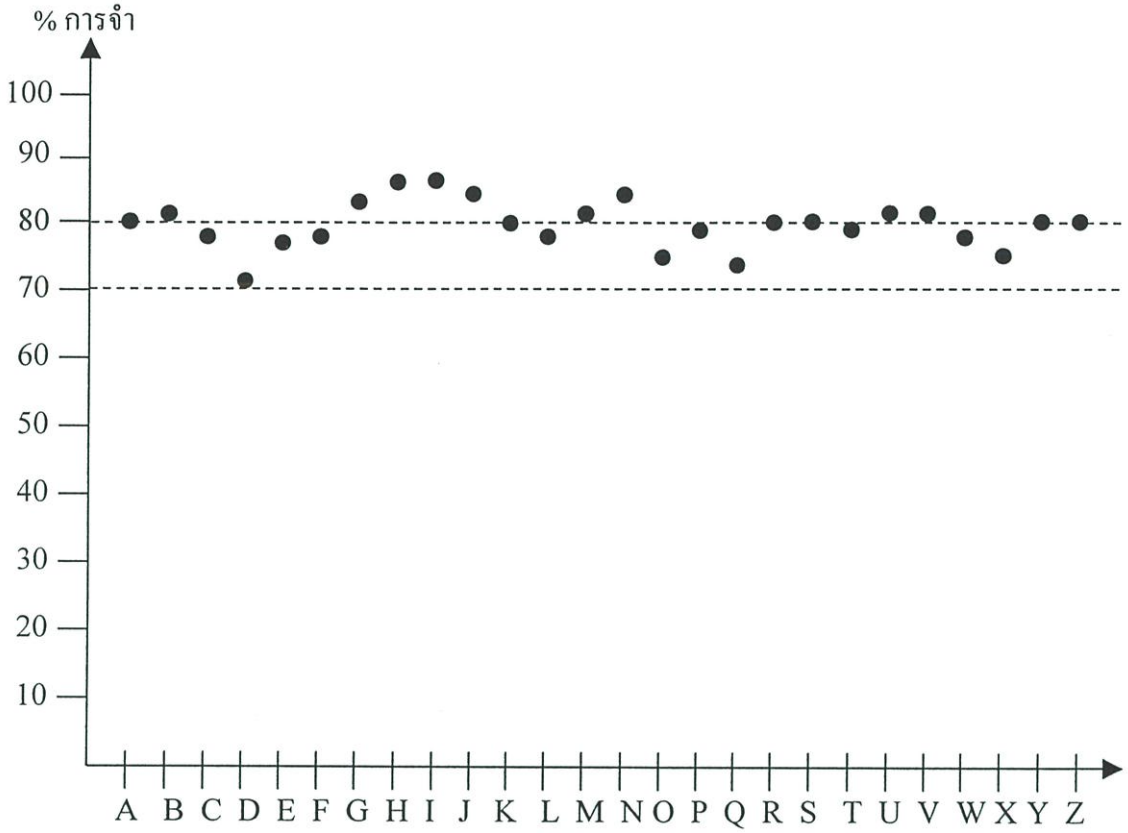
รูปที่ 4.18 ผลลัพธ์ของโปรแกรมกลุ่มรูปภาพที่มีตำแหน่งของภาพไม่อยู่ในตำแหน่งมาตรฐาน



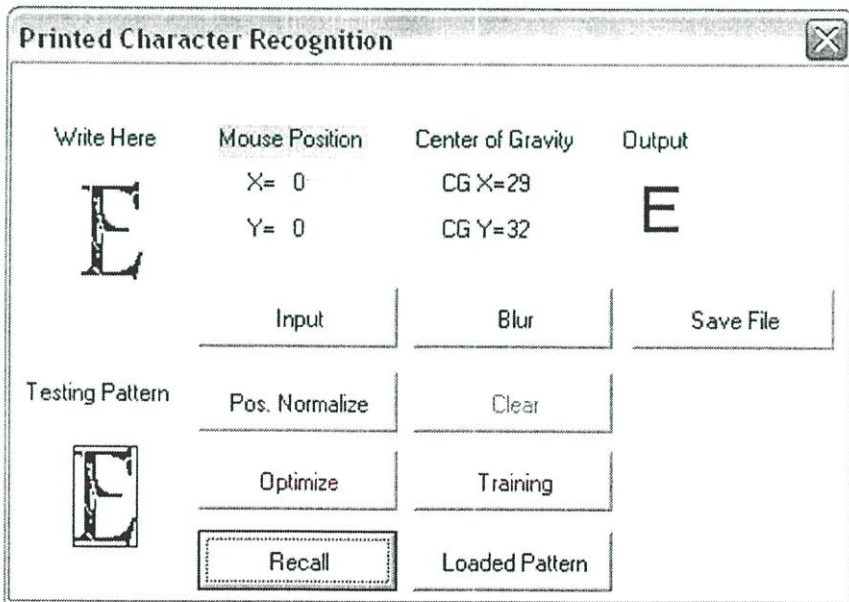
รูปที่ 4.19 ประสิทธิภาพการจำของกลุ่มรูปภาพที่มีสัญญาณรบกวน



รูปที่ 4.20 ผลลัพธ์ของโปรแกรมกลุ่มรูปภาพที่มีสัญญาณรบกวน



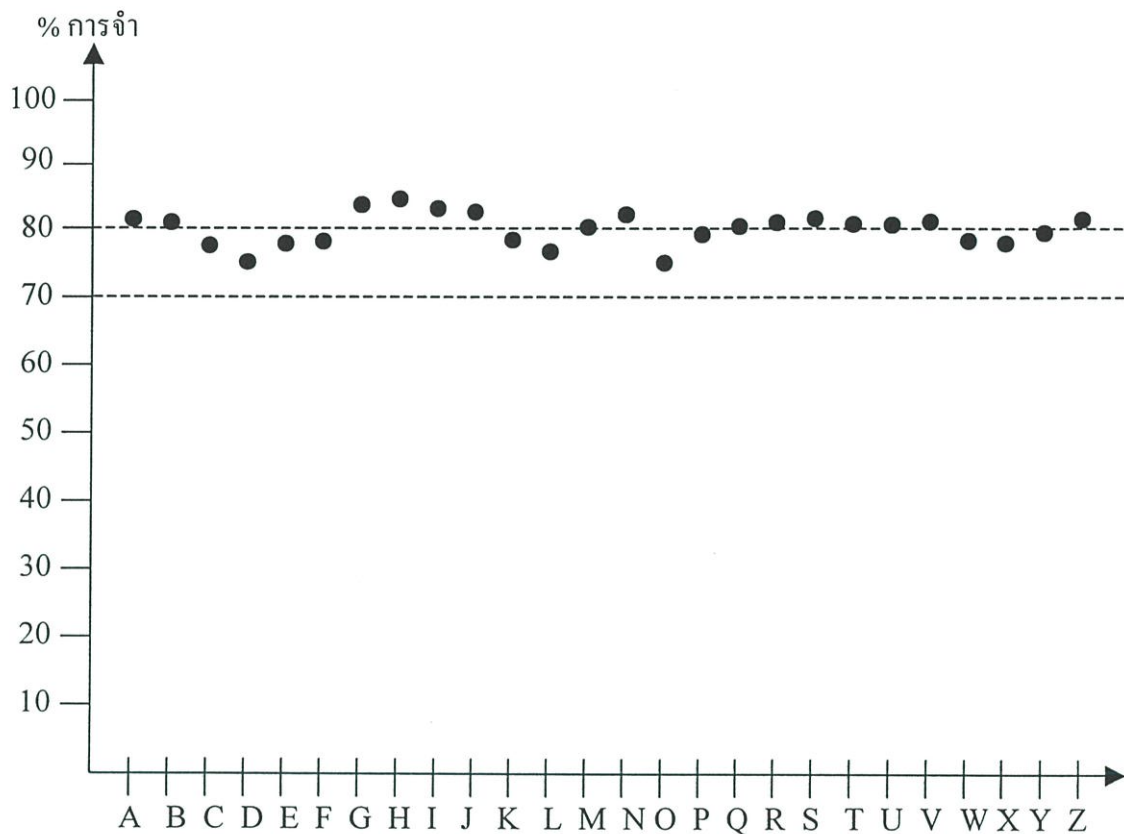
รูปที่ 4.21 ประสิทธิภาพการจำของกลุ่มรูปภาพที่มีตัวอักษรไม่สมมาตร



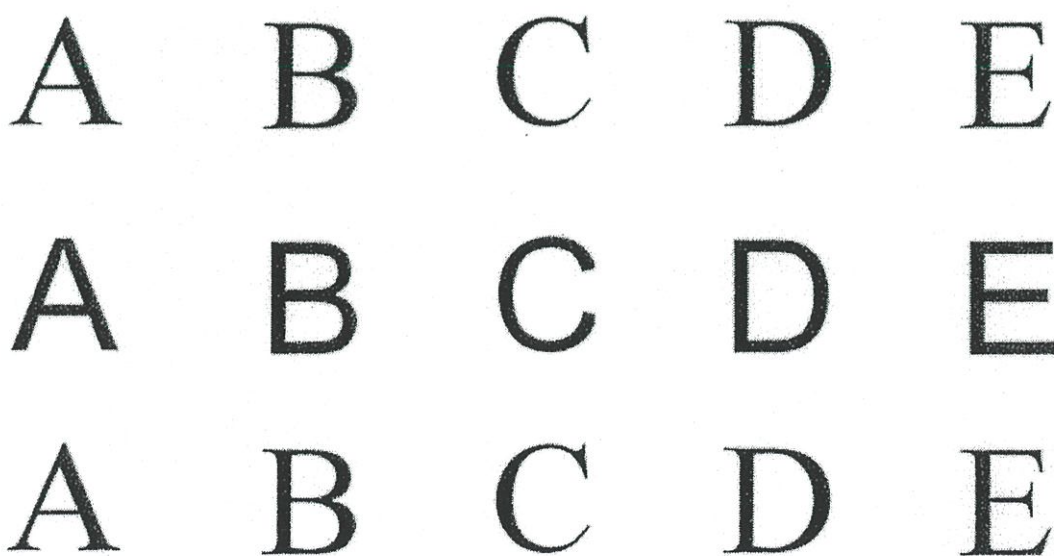
รูปที่ 4.22 ผลลัพธ์ของโปรแกรมกลุ่มรูปภาพที่มีตัวอักษรไม่สมมาตร

4.6 สรุปผลการทดสอบ

การทดสอบที่ใช้กับตัวอักษรต้นแบบจาก Scanner แสดงได้ดังรูปที่ 4.23

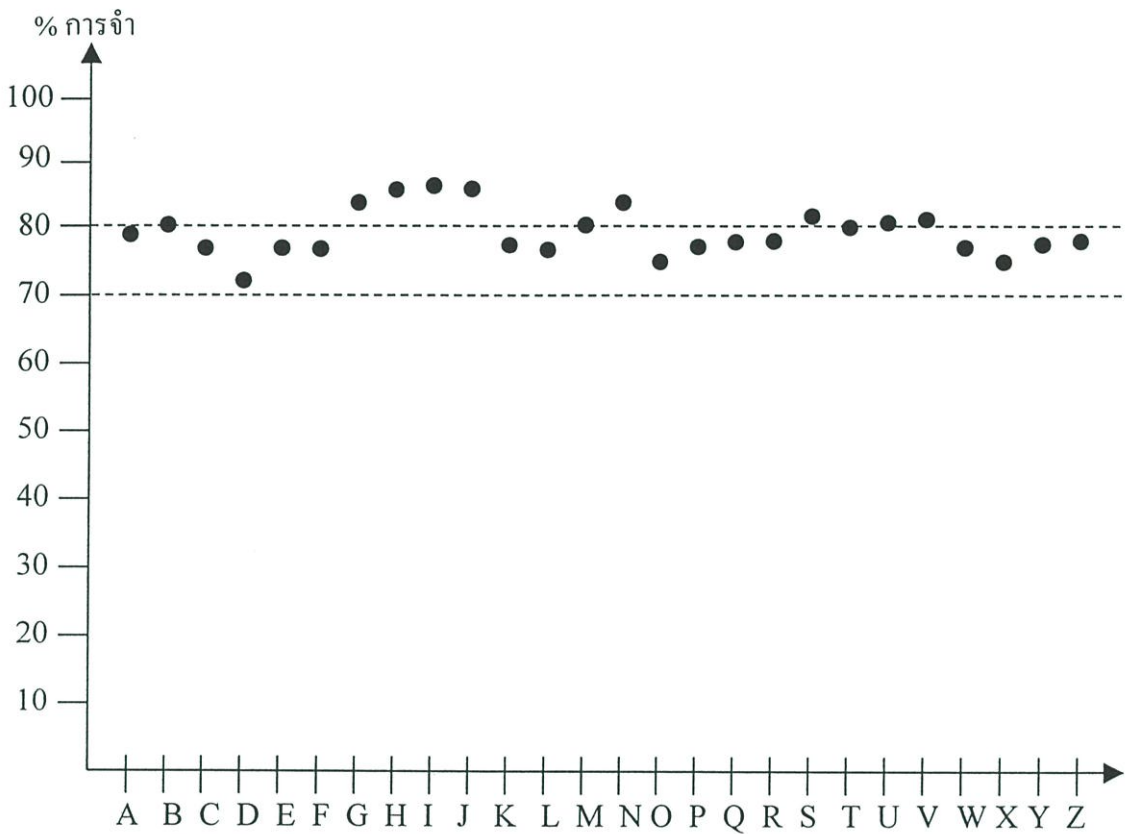


รูปที่ 4.23 ประสิทธิภาพเมื่อทำการทดลองกับภาพต้นแบบจาก Scanner



รูปที่ 4.24 กลุ่มรูปภาพเมื่อทำการทดลองกับภาพต้นแบบจาก Scanner

ระบบที่ได้ออกแบบขึ้นมาี้มีความสามารถในการรู้จำตัวอักษร ได้ทั้ง 4 แบบ โดยมีเปอร์เซนต์เฉลี่ยต่อตัวอักษรดังรูปที่ 4.25



รูปที่ 4.25 ประสิทธิภาพการจำเฉลี่ยจำแนกตามตัวอักษร

$$\text{Normalized Hamming Distacne} = \frac{HD_{ave}}{\text{Bit}} \times 100\% \quad (4.1)$$

โดย HD_{ave} = ระยะทางแฮมมิงเฉลี่ย

Bit = จำนวนบิต

ซึ่งค่า NormailizedHamming Distance ที่ได้จากการทดลอง คือ 3.25%

ตารางที่ 4.4 การเปรียบเทียบเวลาในการเรียนรู้

แบบจำลอง	จำนวน Pattern	
	26 Pattern	78 Pattern
GNU	10 นาที	30 นาที
Backpropagation	30 นาที	มากกว่า 2 ชั่วโมง

ถึงแม้ว่าระบบโดยรวมที่ได้ออกแบบไว้สามารถจำตัวอักษรได้ แต่พบว่าตัวอักษรที่คล้ายกันมากๆ ก็อาจทำให้การ Recall ให้เอาต์พุตที่ผิดพลาดได้เช่น

- O คล้าย Q
- D คล้าย O

เปอร์เซ็นต์การจำอาจไม่เที่ยงตรงจากการมีสัญญาณรบกวนแทรกอยู่ในตัวอักษร แต่พบว่ากลุ่มรูปภาพที่มีตัวอักษรไม่สมบูรณ์มีเปอร์เซ็นต์การจำดีกว่า กลุ่มที่มีสัญญาณรบกวน

บทที่ 5

สรุปผลการวิจัย และข้อเสนอแนะ

5.1 สรุปผลการวิจัย

งานวิจัยนี้สร้างระบบรู้จำตัวอักษรตัวพิมพ์ ภาษาอังกฤษขึ้นเพื่อใช้ในการทดสอบแนวความคิดของ GNU โดยประกอบไปด้วยส่วนที่เป็นอินพุตที่ป้อนเข้าเป็นรูปภาพ ผ่านกระบวนการเตรียมข้อมูลภาพ เพื่อปรับเข้าสู่ตำแหน่งมาตรฐาน แล้วส่งเข้าโครงข่ายประสาทเทียม จุดเด่นของงานวิจัยนี้ คือ โครงข่ายประสาทเทียมที่ใช้มีลักษณะไม่ซับซ้อน ใช้เวลาในการเรียนรู้สั้น และใช้ระยะทางแฮมมิงในการแยกแยะตัวอักษร

ข้อมูลรูปภาพต้นแบบที่ใช้ในการเรียนรู้มีทั้งสิ้น 78 รูปแบบ โดยมีการแยกแยะทางเอาต์พุต 26 รูปแบบ การทดสอบได้มีการกลุ่มรูปภาพดังนี้

1. กลุ่มรูปภาพที่เหมือนกับการเรียนรู้ทุกประการ
2. กลุ่มรูปภาพที่มีตำแหน่งของภาพไม่อยู่ในตำแหน่งมาตรฐาน
3. กลุ่มรูปภาพที่มีสัญญาณรบกวน
4. กลุ่มรูปภาพที่มีตัวอักษรไม่สมบูรณ์

โดยทั่วไปเปอร์เซ็นต์การจำเฉลี่ยอยู่ในเกณฑ์ที่ไม่สูงมากหากเปรียบเทียบกับ โครงข่ายแบบแบคพรอพากชัน แต่โครงสร้างแบบ GNU ใช้เวลาในการเรียนรู้สั้นกว่ามาก จากการเรียนรู้ 78 รูปแบบ ใช้เวลาประมาณ 30 นาที ความผิดพลาดที่เกิดขึ้นในการทดลองนั้น เพราะตัวอักษรบางตัวมีลักษณะคล้ายกันมากเกินไป เช่น O กับ Q และการมีสัญญาณรบกวนในรูปภาพส่วนรูปภาพตัวอักษรที่ไม่สมบูรณ์นั้น ไม่ส่งผลกระทบต่อกระบวนการจำ

5.2 ข้อเสนอแนะ

การปรับปรุงระบบโครงข่ายประสาทเทียมแบบ GNU ให้มีประสิทธิภาพมากขึ้นสามารถทำได้โดย

5.2.1 การเตรียมข้อมูล

1. ปรับปรุงส่วนของการเตรียมข้อมูลให้สามารถกำจัดสัญญาณรบกวนออกได้
2. เพิ่มการปรับขนาดมาตรฐาน โดยสามารถปรับตัวอักษรที่เล็กให้มีขนาดใหญ่ขึ้น หรือตัวอักษรที่มีขนาดใหญ่ให้มีขนาดเล็กลง

5.2.2 โครงข่ายประสาทเทียม

1. ควกรออกแบบเพิ่มเติมในส่วนของ Feedback Connection เพื่อช่วยให้มีประสิทธิภาพมากขึ้น
2. พัฒนาให้ใช้กับตัวอักษรลายมือเขียน
3. ปรับปรุงให้โครงข่ายมีการเรียนรู้โดยอัตโนมัติจะช่วยย่นระยะเวลาในการเรียนรู้ลงได้อีก

บรรณานุกรม

1. สุขสถิต สุขใจ. การใช้โครงข่ายประสาทเทียมเพื่อการจดจำลายมือเขียนตัวเลขอาราบิกแสดงข้อมูลเป็นมมัมสัมพันธ์. วิทยานิพนธ์ปริญญามหาบัณฑิต . กรุงเทพฯ : บัณฑิตวิทยาลัย สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง, 2540.
2. สุรพันธุ์ เอื้อไพบุลย์. การจดจำลายมือเขียนภาษาไทยโดยการพิจารณาหัวของตัวอักษร วิทยานิพนธ์ปริญญามหาบัณฑิต . กรุงเทพฯ : บัณฑิตวิทยาลัย สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง, 2531
3. Aleksander , Igor and Helen Morton. An Introduction to Neural Computing. 2nd ed. London: Chapman & Hall, 1992.
4. Boullart, L., A Krijgsman and R. A. Vigerhoeds. Application of Artificial Intelligence in Process Control. Oxford : Pergamon Press, 1992.
5. Chester, Michael. Neural Networks A Tutorial. Englewood Cliffs : Prentice Hall, 1993.
6. Day hoff, Judith E. Neural Network Architectures An Introduction. New York : Van Nostrand Reinhold, 1990.
7. Eberhart, Russell C. and Roy W. Dobbins. Neural Network PC Tools A Practical Guide. San Diego : Academic Press, 1990.
8. Embree, Paul M. and Bruce Kimble. C Language Algorithms for Digital Signal Processing. Englewood Cliffs : Prentice Hall, 1991.
9. Fausett, Laurene. Fundamentals of Neural Networks Architectures, Algorithms, and Applications. Englewood Cliffs : Prentice Hall, 1994.
10. Fairhurst, Michael C. Computer Vision for Robotic Systems An Introduction. London : Pretice Hall, 1988.
11. Freeman, James A. Simulating Neural Networks with Mathematica. New York : Addison Wesley, 1994 .
12. Freeman, James A. and David M. Skapura. Neural Networks Algorithms, Applications, and Programming Techniques. New York : Addison – Wesley, 1991.
13. Fu, Limin. Neural Networks in Computer Intelligence. Singapore : McGraw – Hill, 1994.

14. Gandhi, Meeta, Tilak Shetty and Rajiv Shah. Vijay Mukhi's The "C" Odyssey Windows – The Brave New World. Singapore : Tech Publications, 1992.
15. Hecht-Nielsen, Robert. Neurocomputing. New York : Addison-Wesley, 1991.
16. Johnson, Nelson. Advanced Graphics in C : Programming and Techniques. Berleley : McGraw-Hill, 1987.
17. Karayiammis, Nicolaos B. and Anastasions N. Venetsanopoulos. Artificial Neural Networks Learning Algorithms, Performance Evaluation, and Applications. Norwell : Kluwer Academic Publishers, 1993.
18. Khanna, Tarun. Foundations of Neural Networks. New York : Addison-Wesley, 1990.
19. KITJAIDURE YUTTANA, The General Neural Unit Aspect of sequence recognition, Ph.D. Thesis, Department of Electrical and Engineering, Imperial college of science, Technology and Medicine, University of London, 1998.
20. Kosko, Bart. Neural Networks and Fuzzy Systems. Englewood Cliffs : Prentice-hall, 1992.
21. Linggard, R, D J Myers and C Nightingale. Neural Networks for Vision, Speech and Natural Language. London : Chapman & Hall, 1992.
22. Masters, Timothy. Practical Neural Network Recipes in C++. London : Academic Press, 1993.
23. Maitree, Kanchit. Structure Analysis of Handwriting to Recognition System. Doctoral Dissertation. Tokai : Tokai University, 1994.
24. Pao, Yoh-Han. Adaptive Pattern Recognition and Neural Networks. New York : Addison – Wesley, 1989.
25. Ritter, Helge, Thomas Martinetz and Klaus Schulten. Neural Computation and Self-Organizing Maps An Introduction. New York : Addison-Wesley, 1991.
26. Wasserman, Philip D . Neural Computing Theory and Practice. New York : Van Nostrand Reinhold, 1989.
27. Zurada, Jacek M. Introduction to Artificial Neural Systems. Singapore: West Publishing, 1992.

ภาคผนวก

ภาคผนวก ก

Source Code

```

//-----
#include <vcl.h>
#include "stdio.h"
#pragma hdrstop
#include "Main.h"
//-----

#pragma package(smart_init)
#pragma resource "*.dfm"
#pragma resource "extrares.res"

const crDraw = 1 ;           //Define value about changing mouse pointer
bool chMouse,first=true ;   //Enable Check Mouse Position

int p[64][64], blur[64][64];
int x2, x1, y2, y1, ix, iy; //Border and Center of Gravity
static pn[64][64];         //Pattern Normalized
TForm1 *Form1;
//-----
__fastcall TForm1::TForm1(TComponent* Owner)
: TForm(Owner)
{
}
//-----
void __fastcall TForm1::Image1MouseMove(TObject *Sender, TShiftState Shift,
int X, int Y)
{
Image1->Cursor = TCursor(crDraw);
}

```

```

TextX->Caption=X;
TextY->Caption=Y;
if(chMouse)
    Image1->Canvas->LineTo(X,Y);
}
//-----
void __fastcall TForm1::Image1MouseDown(TObject *Sender,
    TMouseButton Button, TShiftState Shift, int X, int Y)
{
    chMouse = true;
    Image1->Canvas->MoveTo(X,Y);
    Image1->Canvas->Pixels[X][Y]=clBlack;
}
//-----
void __fastcall TForm1::Image1MouseUp(TObject *Sender, TMouseButton Button,
    TShiftState Shift, int X, int Y)
{
    chMouse=false;
}
//-----
void __fastcall TForm1::FormCreate(TObject *Sender)
{
    HINSTANCE HInst;
    HInst = reinterpret_cast<HINSTANCE>(HInstance);
    Screen->Cursors[crDraw] = LoadCursor(HInst, "DRAW");
}
//-----
void __fastcall TForm1::TestClick(TObject *Sender)
{ int i,j;

```

```

for(i=0;i<64;i++)
  for(j=0;j<64;j++){
    if(Image1->Canvas->Pixels[i][j]==clBlack){
      p[i][j]=1;
      Image2->Canvas->Pixels[i][j]=clBlack;
    }
    else{
      Image2->Canvas->Pixels[i][j]=clWhite;
      p[i][j]=0;
    }
  }
}

//-----

void __fastcall TForm1::BlurClick(TObject *Sender)
{
  int i,j;
  for(i=0;i<64;i++)
    for(j=0;j<64;j++){
      if(p[i][j]==1){
        blur[i+1][j]=1;
        blur[i+1][j-1]=1;
        blur[i][j-1]=1;
        blur[i-1][j-1]=1;
        blur[i-1][j]=1;
        blur[i-1][j+1]=1;
        blur[i][j+1]=1;
        blur[i+1][j+1]=1;
      }
    }
}

```

```
for(i=0;i<64;i++)
for(j=0;j<64;j++){
    if(blur[i][j]==1){
        Image2->Canvas->Pixels[i][j]=clBlack;
        p[i][j]=1;
    }
    else{
        Image2->Canvas->Pixels[i][j]=clWhite;
        p[i][j]=0;
    }
}
}
//-----
void __fastcall TForm1::ClearClick(TObject *Sender)
{ int i,j;
  Image1->Stretch = true;
  Image1->Picture->LoadFromFile("blank.BMP");
  Image2->Stretch = true;
  Image2->Picture->LoadFromFile("blank.BMP");
  for(i=0;i<64;i++)
    for(j=0;j<64;j++){
      p[i][j]=0;
      blur[i][j]=0;
      pn[i][j]=0;
    }
}
//-----
```

```
void __fastcall TForm1::PosNormClick(TObject *Sender)
{
    register int i,j ; //Border
    i=0; j=0;
    while(p[i][j]==0){
        j++;
        if(j==63){
            j=0;
            i++;
        }
    }
    x1=i-1; //Left bound

    i=63;
    j=63;
    while(p[i][j]==0){
        j--;
        if(j==0){
            j=63;
            i--;
        }
    }
    x2=i+1; //Right bound

    i=0;j=0;
    while(p[i][j]==0){
        i++;
        if(i==63){
```

```

    i=0;
    j++;
}
}
y1=j-1;//Uper bound

```

```

i=63;j=63;
while(p[i][j]==0){
    i--;
    if(i==0){
        i=63;
        j--;
    }
}
y2=j+1;//Lower bound

```

```

Image2->Canvas->MoveTo(x1,y1);
Image2->Canvas->LineTo(x2,y1);
Image2->Canvas->LineTo(x2,y2);
Image2->Canvas->LineTo(x1,y2);
Image2->Canvas->LineTo(x1,y1);

```

```

int A=0,B=0;// Center of Gravity
for(i=x1;i<x2;i++){
    for(j=y1;j<y2;j++){
        A=A+(p[i][j]*j);
        B=B+(p[i][j]);
    }
}

```

```

iy=A/B;
A=0;B=0;
for(j=y1;j<y2;j++){
  for(i=x1;i<x2;i++){
    A=A+(p[i][j]*i);
    B=B+(p[i][j]);
  }
}

```

```

ix=A/B;

```

```

Image2->Canvas->Pixels[ix-1][iy-1]=clBlue;
Image2->Canvas->Pixels[ix][iy]=clBlue;
Image2->Canvas->Pixels[ix+1][iy+1]=clBlue;
Image2->Canvas->Pixels[ix+1][iy-1]=clBlue;
Image2->Canvas->Pixels[ix-1][iy+1]=clBlue;
Textix->Caption=ix;
Textiy->Caption=iy;

```

```

//This is Moving Pattern to Center(position Normalize)

```

```

for(i=x1;i<x2;i++){
  for(j=y1;j<y2;j++){
    pn[i+(32-ix)][j+(32-iy)]=p[i][j];
  }
}
for(i=0;i<64;i++){
  for(j=0;j<64;j++){
    if(pn[i][j]==1){
      Image1->Canvas->Pixels[i][j]=clBlack;
    }
  }
}

```

```
        else{
            Image1->Canvas->Pixels[i][j]=clWhite;
        }
    }
}
//-----

void __fastcall TForm1::OptimClick(TObject *Sender)
{
    OpenFileDialog->DefaultExt = ".des";
    OpenFileDialog->FileName = "*.des";
    if (OpenDialog1->Execute())
    {
        Form1->Caption = OpenFileDialog->FileName;
        FILE *stream;
        int handle;

        /* create a file */
        stream = fopen("DUMMY.FIL", "w");

        /* obtain the file handle associated with the stream */
        handle = fileno(stream);

        /* display the handle number */
        printf("handle number: %d\n", handle);

        /* close the file */
        fclose(stream);
    }
}
```

```

}

//-----
#include <sys\stat.h>
#include <string.h>
#include <stdio.h>
#include <fcntl.h>
#include <io.h>

void __fastcall TForm1::BitBtn2Click(TObject *Sender)
{
    if(OpenPictureDialog1->Execute()){
        Image1->Picture->LoadFromFile(OpenPictureDialog1->FileName);
    }
}

//-----
void __fastcall TForm1::HammingClick(TObject *Sender)
{
    int pa[64][64], hamming[64][64];
    int ham[78][168][168];
    int a=0,b=0,c=0;
    static ph[64][64];
    int i=0, j=0, k=0;
    for(i=0;i<78;i++)
        for(j=0;j<64;j++)
            for(k=0;k<64;k++)
                if(p[j][k]==0)
                    { ham[i][j][k]=0; a++;
                    }
                else {ham[i][j][k] = 1;
                    b++;}}
}

```