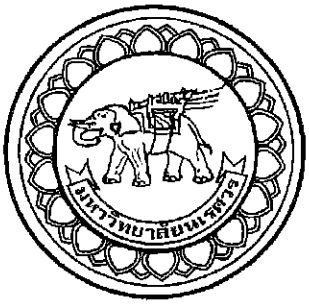


การรู้จำตัวอักษรภาษาอังกฤษโดยใช้โครงข่ายประสาทเทียม

English Characters Recognition Using An Artificial Neural Network

นางสาวพองาม	ตั้งจันทร	รหัส 45360294
นางสาวรุ่งรัตน์	เสือประดิษฐ์	รหัส 45360351
นางสาวอัปสรสินี	เกี้ยวเกิด	รหัส 45360633

I 5078696

ห้องสมุดคณะวิศวกรรมศาสตร์
วันที่รับ 28 มี.ค. 2549
เลขทะเบียน 4900044
เลขเรียกหนังสือ ว.ป. ท438ก
มหาวิทยาลัยนเรศวร

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมคอมพิวเตอร์ ภาควิศวกรรมไฟฟ้าและคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

ปีการศึกษา 2548

ใบรับรองโครงงานวิศวกรรม

หัวข้อโครงงาน	การรู้จำตัวอักษรภาษาอังกฤษโดยใช้โครงข่ายประสาทเทียม		
ผู้ดำเนินโครงงาน	นางสาวพองาม	ตั้งจันทร	รหัส 45360294
	นางสาวรุ่งรัตน์	เสือประดิษฐ์	รหัส 45360351
	นางสาวอัปสรสินี	เกี้ยวเกิด	รหัส 45360633
อาจารย์ที่ปรึกษา	ดร.พนมขวัญ	ริยะมงคล	
สาขาวิชา	วิศวกรรมคอมพิวเตอร์		
ภาควิชา	วิศวกรรมไฟฟ้าคอมพิวเตอร์		
ปีการศึกษา	2548		

.....

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร อนุมัติโครงงานฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร วิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมคอมพิวเตอร์

คณะกรรมการสอบโครงงานวิศวกรรม

.....ประธานกรรมการ
(ดร.พนมขวัญ ริยะมงคล)

..... กรรมการ
(ผู้ช่วยศาสตราจารย์ ดร.สุชาติ แย้มเม่น)

..... กรรมการ
(อาจารย์ศิริพร เดชะศิลารักษ์)

หัวข้อโครงงาน	การรู้จำตัวอักษรภาษาอังกฤษโดยใช้โครงข่ายประสาทเทียม
ผู้ดำเนินโครงงาน	นางสาวพองาม ตั้งจันทร รหัส 45360294
	นางสาวรุ่งรัตน์ เสือประดิษฐ์ รหัส 45360351
	นางสาวอัปสรสินี เกี้ยวเกิด รหัส 45360633
อาจารย์ที่ปรึกษา	ดร.พนมขวัญ ริยะมงคล
สาขาวิชา	วิศวกรรมคอมพิวเตอร์
ภาควิชา	วิศวกรรมไฟฟ้าและคอมพิวเตอร์
ปีการศึกษา	2548

บทคัดย่อ

โครงงานนี้ศึกษาและพัฒนาโปรแกรมการรู้จำอักษรภาษาอังกฤษโดยใช้โครงข่ายประสาทเทียมซึ่งเป็นเครื่องมือที่มีความยืดหยุ่นในการทำงานสูงและสามารถปรับตัวเองให้รับรู้สภาพที่เปลี่ยนแปลงได้

วิธีการรู้จำอักษรภาษาอังกฤษประกอบด้วย 3 ขั้นตอนหลักคือ ขั้นตอนการเตรียมตัวอย่างข้อมูลเพื่อเตรียมข้อมูลเข้าให้กับโครงข่ายประสาทเทียม ขั้นตอนการเรียนรู้เพื่อฝึกสอนให้โครงข่ายประสาทเทียมทำการรู้จำ และขั้นตอนการรู้จำและเพิ่มสัญญาณรบกวนเพื่อทดสอบความสามารถและทดสอบผลการรู้จำของโครงข่ายประสาทเทียม

ภาพที่ใช้ในโครงงานนี้เป็นภาพขาวดำชนิด Bitmap ขนาด 8x 6 พิกเซล จากการทดสอบพบว่าถ้าเราเพิ่มสัญญาณรบกวนเข้าไป 5% การทำงานของโครงข่ายประสาทเทียมจะสามารถรู้จำได้ทุกตัวอักษร แต่ความสามารถในการรู้จำจะลดลงเมื่อทำการเพิ่มสัญญาณรบกวน

Project title	English Characters Recognition Using An Artificial Neural Network
Name	Miss Phorngam Tungjuntorn ID. 45360294
	Miss Rungrat Seupradit ID. 45360351
	Miss Apsornsinee Keawkerd ID. 45360633
Project advisor	Dr. Panomkhawn Riyamongkol
Major	Computer Engineering
Department	Electrical and Computer Engineering
Academic year	2005

.....

Abstract

The objectives of this project are to study and develop a program for English characters recognition using an artificial neural network. An artificial neural network is a useful tool for Pattern recognition because it is flexible and can adjust itself to learn when the environment is changed.

There are 3 steps in this project : 1) Preparing dataset to be input for the artificial neural network 2) training the artificial neural network to recognize characters and 3) testing the recognition of the artificial neural network.

According to the result obtained from testing a 8x6 monochrome bitmap image, we can conclude that the artificial neural network can recognize all characters when we add 5% noise. It can recognize less characters when noise is increased.

กิตติกรรมประกาศ

โครงงานวิศวกรรมคอมพิวเตอร์ฉบับนี้ สำเร็จได้ด้วยดีก็เนื่องด้วยความอนุเคราะห์จากอาจารย์ที่ปรึกษาโครงงาน คือ ดร.พนมขวัญ ริยะมงคล ที่คอยให้ความช่วยเหลือ เอาใจใส่ และ ช่วยแนะนำในทุก ๆด้าน รวมถึงอาจารย์ทุกท่านและเพื่อน ๆ ที่มิได้กล่าวถึงที่ได้คอยแนะนำ และให้คำปรึกษา ในโอกาสนี้ทางคณะผู้จัดทำโครงงานจึงขอขอบคุณทุก ๆท่าน ที่มีส่วนช่วยให้โครงงานนี้ประสบความสำเร็จได้ด้วยดี

รวมทั้งบิดามารดาของข้าพเจ้าที่เป็นผู้สนับสนุนในทุก ๆด้านและให้โอกาสทางการศึกษาอย่างเต็มที่ ข้าพเจ้าขอระลึกในพระคุณของท่านและขอกราบขอบพระคุณมา ณ ที่นี้

นางสาวพองาม ตั้งจันทร
นางสาวรุ่งรัตน์ เสือประดิษฐ์
นางสาวอัปสรสินี เกี้ยวเกิด

สารบัญ

หน้า

บทคัดย่อภาษาไทย ..... ก
บทคัดย่ออังกฤษ ..... ข
กิตติกรรมประกาศ ..... ค
สารบัญ ..... ง
สารบัญตาราง ..... ฉ
สารบัญรูป ..... ช

บทที่1 บทนำ

1.1 หลักการและเหตุผล 1
1.2 วัตถุประสงค์ของโครงงาน 1
1.3 ขอบข่ายของโครงงาน 2
1.4 ขั้นตอนของการดำเนินงาน 2
1.5 ผลที่คาดว่าจะได้รับ 2
1.6 งบประมาณที่ใช้ 3

บทที่2 หลักการและทฤษฎี

2.1 ทฤษฎีของโครงข่ายประสาทเทียม 4

บทที่3 วิธีการดำเนินโครงงาน

3.1 การเตรียมภาพตัวอย่าง 8
3.2 กระบวนการเรียนรู้โครงข่ายประสาทเทียมแบบแพร่กลับ 12
3.3 การรู้จำตัวอักษรภาษาโดยใช้โครงข่ายประสาทเทียม 14

บทที่4 ผลการทดลองและการวิเคราะห์

4.1 ผลการทดลอง 16
4.1.1 ตัวอักษรปกติเพื่อการฝึกสอนโครงข่ายประสาทเทียม 17
4.1.2 ผลการทดลองการรู้จำโดยเพิ่มสัญญาณรบกวน ขนาด 5 % 18
4.1.3 ผลการทดลองการรู้จำโดยเพิ่มสัญญาณรบกวน ขนาด 10 % 20

สารบัญ(ต่อ)

4.1.4 ผลการทดลองการรู้จำโดยเพิ่มสัญญาณรบกวน ขนาด 15 % 21
4.1.5 ผลการทดลองการรู้จำโดยเพิ่มสัญญาณรบกวน ขนาด 20 % 23
4.2 วิเคราะห์ผลการทดลอง..... 25

บทที่5 บทสรุป

5.1 สรุปผลการทดลอง..... 27
5.2 ปัญหาในการทดลองและแนวทางแก้ไข 27
5.3 แนวทางในการพัฒนาในอนาคต 27

เอกสารอ้างอิง 28
ภาคผนวก 29
ประวัติผู้เขียนโครงงาน 36

สารบัญตาราง

ตารางที่ หน้า

4.1 แสดงการเปรียบเทียบผลการรู้จำของโครงข่ายประสาทเทียม 26

มหาวิทยาลัยนเรศวร

สารบัญรูป

รูปที่ หน้า

2.1 สถาปัตยกรรมของโครงข่ายประสาทเทียม 5

2.2 สถาปัตยกรรมของโครงข่ายประสาทเทียมแบบแพร่ย้อนกลับ 5

3.1 แสดงแผนผังขั้นตอนการทำงานของการเตรียมตัวอย่าง 8

3.2 แสดงแผนผังขั้นตอนการเรียนรู้ตัวอักษรโดยใช้โครงข่ายประสาทเทียม 9

3.3 แสดงแผนผังขั้นตอนการทำงานของการรู้จำตัวอักษรโดยใช้โครงข่ายประสาทเทียม 10

3.4 แสดงตัวอย่างภาพต้นฉบับ 11

3.5 แสดงสีแต่ละจุดของภาพ 11

3.6 แสดงภาพตัวอักษรที่ทำการปรับค่าให้อยู่ช่วง 0 ถึง 1 12

3.7 แสดงภาพโครงสร้างของโครงข่ายประสาทเทียมที่ใช้ในโปรแกรม 13

4.1 ระดับการฝึกฝนของโครงข่ายประสาทเทียม 17

4.2 ชุดอักษรที่ใช้ในการฝึกสอนให้กับโครงข่ายประสาทเทียม 18

4.3 แสดงภาพตัวอย่างของการทดสอบตัวพิมพ์เล็กที่ทำการเพิ่มสัญญาณรบกวน 5% 19

4.4 แสดงภาพตัวอย่างของการทดสอบตัวพิมพ์ใหญ่ที่ทำการเพิ่มสัญญาณรบกวน 5% 19

4.5 แสดงภาพตัวอย่างของการทดสอบสัญลักษณ์ที่ทำการเพิ่มสัญญาณรบกวน 5% 20

4.6 แสดงภาพตัวอย่างของการทดสอบตัวพิมพ์เล็กที่ทำการเพิ่มสัญญาณรบกวน 10% 20

4.7 แสดงภาพตัวอย่างของการทดสอบตัวพิมพ์ใหญ่ที่ทำการเพิ่มสัญญาณรบกวน 10% 21

4.8 แสดงภาพตัวอย่างของการทดสอบสัญลักษณ์ที่ทำการเพิ่มสัญญาณรบกวน 10% 21

4.9 แสดงภาพตัวอย่างของการทดสอบตัวพิมพ์เล็กที่ทำการเพิ่มสัญญาณรบกวน 15% 22

4.10 แสดงภาพตัวอย่างของการทดสอบตัวพิมพ์ใหญ่ที่ทำการเพิ่มสัญญาณรบกวน15% 22

4.11 แสดงภาพตัวอย่างของการทดสอบสัญลักษณ์ที่ทำการเพิ่มสัญญาณรบกวน 15% 23

4.12 แสดงภาพตัวอย่างของการทดสอบตัวพิมพ์เล็กที่ทำการเพิ่มสัญญาณรบกวน 20% 23

4.13 แสดงภาพตัวอย่างของการทดสอบตัวพิมพ์ใหญ่ที่ทำการเพิ่มสัญญาณรบกวน 20% 24

4.14 แสดงภาพตัวอย่างของการทดสอบสัญลักษณ์ที่ทำการเพิ่มสัญญาณรบกวน 20% 24

บทที่ 1
บทนำ

1.1 หลักการและเหตุผล

การรู้จำ (Recognition) คืองานวิจัยแนวหนึ่งที่มีความสำคัญ และเป็นแขนงหนึ่งของระบบปัญญาประดิษฐ์ (Artificial Intelligence) การรู้จำมีหลายชนิดเช่น การรู้จำวัตถุ การรู้จำเสียง การรู้จำภาพ และการรู้จำตัวอักษร เป็นต้น

การรู้จำตัวอักษร เป็นการนำรูปตัวอักษรที่ได้มาทำการวิเคราะห์ว่าเป็นตัวอักษรใด การวิเคราะห์สามารถใช้ได้หลายแบบ เช่น เฮาซดรอฟดิสแทนซ์ (Hausdorff Distance) ลำดับการเปลี่ยนแปลงของเส้น (Stroke Changing Sequence) การวิเคราะห์ด้วยเพนนาไลซ์ ดิสคริมิแนนท์ (Penalised Discriminant Analysis) การใช้ฟัซซี่ลอจิค (Fuzzy Logic) และการใช้โครงข่ายประสาทเทียม (Neural Networks) เป็นต้น

โครงข่ายประสาทเทียม เป็นวิธีการที่เพิ่งสร้างขึ้นมาจากการประยุกต์การทำงานของโครงข่ายประสาททางชีววิทยามาใช้สำหรับแก้ปัญหาที่แก้ได้ยากสำหรับคอมพิวเตอร์ ซึ่งจะมีผลทำให้คอมพิวเตอร์มีความสามารถใกล้เคียงกับมนุษย์ และความสามารถของโครงข่ายประสาทเทียมที่ถูกนำมาใช้งานมากที่สุด คือ Pattern Recoganize อันมีประโยชน์มากมายในการประยุกต์ เช่นการรู้จำวัตถุ การรู้จำเสียง การรู้จำภาพ และการรู้จำตัวอักษร เป็นต้น นอกจากนี้โครงข่ายประสาทเทียมยังมีความยืดหยุ่นในการทำงานสูงและสามารถปรับตัวเองให้รับรู้สภาพที่เปลี่ยนแปลงได้ ถึงแม้ว่าข้อมูลจะมีความผิดเพี้ยน (Distortion) เนื่องจากสัญญาณรบกวน (Noise) หรือ Pattern ตัวอักษรมีคุณภาพไม่ดี

จากคุณสมบัติของโครงข่ายประสาทเทียมที่มีความสามารถสูงในด้านการรู้จำ ผู้ทำโครงงานจึงสนใจที่จะนำศาสตร์ของโครงข่ายประสาทเทียม มาทำการฝึกสอนให้รู้จำตัวอักษรภาษาอังกฤษ เพื่อให้ได้ผลการรู้จำที่มีประสิทธิภาพที่สุด

1.2 วัตถุประสงค์ของโครงงาน

1. ศึกษาวิธีการรู้จำตัวอักษร โดยใช้โครงข่ายประสาทเทียมแบบแพร่ย้อนกลับเป็นหลักในการวิเคราะห์หาผลลัพธ์
2. เพื่อพัฒนาประสิทธิภาพในการรู้จำ และแนวคิดต่างๆ เพื่อเสนอแนวคิดใหม่ที่สามารถให้ประสิทธิภาพในการรู้จำที่ดีขึ้น
3. เป็นแนวทางสำหรับศึกษาและพัฒนาระบบรู้จำแบบอื่นๆที่เกี่ยวข้อง ทั้งการรู้จำวัตถุ การรู้จำเสียง และการรู้จำใบหน้าคนจากภาพ

1.3 ขอบข่ายของโครงงาน

1. ตัวอักษรที่นำมารู้จำ จำนวน 57 ตัว ประกอบด้วย

   ตัวอักษรภาษาอังกฤษพิมพ์ใหญ่ จำนวน 26 ตัว

   ตัวอักษรภาษาอังกฤษพิมพ์เล็ก จำนวน 26 ตัว

   สัญลักษณ์ภาษาอังกฤษ จำนวน 5 ตัว

2. รูปแบบตัวอักษรที่นำมารู้จำ คือ CordiaUPC ขนาด 16 หน่วยโดยการอ่านตัวอักษรแบบ Grayscale

3. การรู้จำภาพตัวอักษร เป็นการรู้จำตัวอักษรตัวตรงขนาดปกติ ไม่หนา ไม่เอียง

1.4 ขั้นตอนการดำเนินโครงงาน

กิจกรรม	ระยะเวลา											
	เม.ย.	พ.ค.	มิ.ย.	ก.ค.	ส.ค.	ก.ย.	ต.ค.	พ.ย.	ธ.ค.	ม.ค.	ก.พ.	มี.ค.
1. ศึกษาข้อมูลในการทำโครงงาน												
2. เตรียมข้อมูลตัวอย่าง												
3. ทดลองอัลกอริทึมสำหรับการรู้จำ												
4. ปรับปรุงแก้ไขอัลกอริทึม												
5. วิเคราะห์และสรุปผล												

1.5 ผลที่คาดว่าจะได้รับ

1. ได้อัลกอริทึมการรู้จำตัวอักษรภาษาอังกฤษที่มีประสิทธิภาพสูงขึ้น สามารถวิเคราะห์ตัวอักษรที่มีความคล้ายกันได้ดีขึ้น รู้จำตัวอักษรแบบอื่นที่ไม่ได้เรียนรู้โดยผลลัพธ์ถูกต้องในระดับที่ยอมรับได้ และมีความทนทานต่อสัญญาณรบกวนที่เกิดขึ้น

2. สามารถนำไปใช้ในการพัฒนาเพื่อการใช้งานจริง เมื่อการรู้จำตัวอักษรภาษาอังกฤษมีประสิทธิภาพสูงขึ้น และทนต่อการรบกวนในรูปแบบต่างๆ ย่อมเกิดการพัฒนาในเชิงพาณิชย์มากขึ้น

1.6 รายละเอียดงบประมาณตลอดโครงงาน

1. ค่าวัสดุสำนักงาน	เป็นจำนวนเงิน	1,000	บาท
2. ค่าหมึกพิมพ์	เป็นจำนวนเงิน	1,000	บาท
3. ค่าสำเนาเอกสาร	เป็นจำนวนเงิน	1,000	บาท
	รวมเป็นเงินทั้งสิ้น	3,000	บาท

หมายเหตุ ขออนุมัติถัวเฉลี่ยทุกรายการ

บทที่ 2
หลักการและทฤษฎี

การรู้จำตัวอักษรภาษาอังกฤษโดยใช้โครงข่ายประสาทเทียม มีขั้นตอนดังนี้ คือ เริ่มจากการสแกนภาพเอกสารที่เป็นตัวอักษรทั้งหมดด้วยเครื่องสแกนเนอร์ ทำการเก็บภาพที่ได้จากการสแกนเป็นสกุล BMP ภาพ Bitmap ที่ได้จะเป็นภาพ Gray scale และทำการปรับค่าให้อยู่ในระหว่าง 0 ถึง 1 เพื่อเป็นการสร้างข้อมูลนำเข้าให้กับโครงข่ายประสาทเทียม (pattern) จากนั้นจะทำการสร้างโครงข่ายประสาทเทียมแล้วจึงฝึกสอนด้วยหลักการของโครงข่ายประสาทเทียมแบบแพร่ย้อนกลับ (Back Propagation) แล้วจัดเก็บไว้เป็นฐานข้อมูลเพื่อเปรียบเทียบกับค่าที่รับเข้ามา และแสดงผลว่าเป็นตัวอักษรใด โดยขั้นตอนดังกล่าวต้องอาศัยทฤษฎีที่เกี่ยวข้องในแต่ละขั้นตอนตามลำดับดังต่อไปนี้

2.1 ทฤษฎีของโครงข่ายประสาทเทียม

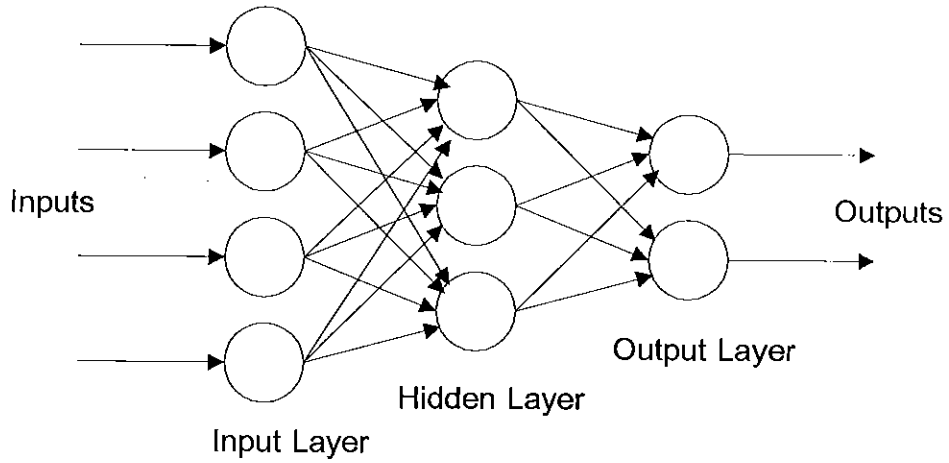
โครงข่ายประสาทเทียม (Artificial Neural Networks) ถูกสร้างขึ้นเพื่อเลียนแบบการทำงานของสมองมนุษย์ มีการทำงานแบบขนานจำนวนมาก ในที่นี้เลือกใช้โครงข่ายประสาทเทียมแบบเพอร์เซฟตรอนหลายชั้น (Multilayer Neural Network:MLP) และมีการเรียนรู้โดยใช้หลักการของโครงข่ายประสาทเทียมแบบแพร่ย้อนกลับ (Feedforward Backpropagation Neural Network) ซึ่งเป็นโครงข่ายประสาทเทียมแบบต้องมีผู้สอน (Superviced Learning)

สถาปัตยกรรมของโครงข่ายประสาทเทียมแบบเพอร์เซฟตรอนหลายชั้นมีลักษณะหลักๆ ดังนี้คือ

1. จำนวนชั้นต่างๆ โครงข่ายประสาทเทียมจะประกอบด้วยชั้นต่างๆ คือ ชั้นอินพุต (Input layer) ชั้นเอาต์พุต (Output layer) และชั้นซ่อน (Hidden layer) ซึ่งจะอยู่ระหว่างชั้นอินพุตและชั้นเอาต์พุต

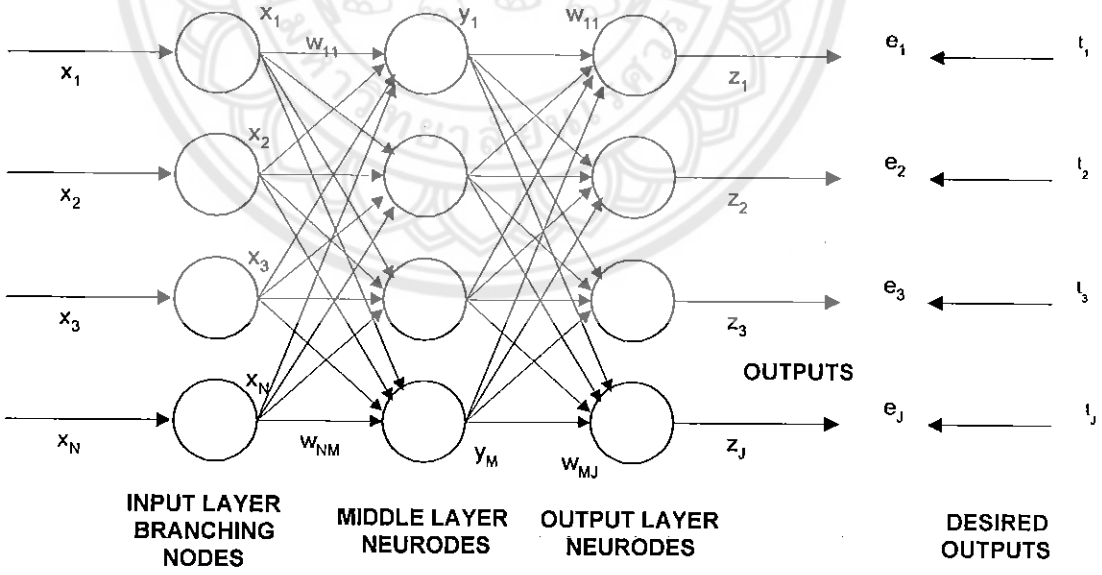
2. การเชื่อมต่อระหว่างชั้นต่างๆ การเชื่อมต่อระหว่างชั้นต่างๆนั้น ทุกๆ โหนดในชั้นอินพุตจะส่งสัญญาณไปยังทุกๆ โหนดในชั้นซ่อนชั้นแรก และทุกๆ โหนดในชั้นซ่อนชั้นแรกจะส่งสัญญาณ ไปยังทุกๆ โหนดในชั้นถัดไป จนในที่สุดทุกๆ โหนดในชั้นซ่อนชั้นสุดท้าย จะส่งสัญญาณไปยังทุกๆ โหนดในชั้นเอาต์พุต

3. การทำงานของชั้นอินพุตไม่มีการประมวลผล เพียงแค่ทำหน้าที่รับข้อมูลเข้าไปแล้วส่งค่าไปยังแต่ละโหนดในชั้นซ่อนเท่านั้น ส่วนชั้นซ่อนและชั้นเอาต์พุต นั้นมีการประมวลผล รูปที่ 2.1 แสดงสถาปัตยกรรมของโครงข่ายประสาทเทียมแบบเพอร์เซพตรอนหลายชั้น ซึ่งจะประกอบไปด้วย ชั้นของอินพุต ชั้นซ่อนจำนวน 1 ชั้น และชั้นของเอาต์พุต โดยแต่ละโหนดจะถูกเชื่อมต่อกันเป็นโครงข่าย



รูปที่ 2.1 สถาปัตยกรรมของโครงข่ายประสาทเทียม

วิธีการเรียนรู้แบบแพร่ย้อนกลับ (Backpropagation Learning) อธิบายขั้นตอนวิธีการเรียนรู้ของโครงข่ายประสาทเทียมแบบแพร่ย้อนกลับแบบพื้นฐานที่มีชั้นกลาง 1 ชั้น พร้อมตัวแปรต่างๆ แสดงในรูปที่ 2.2



รูปที่ 2.2 สถาปัตยกรรมของโครงข่ายประสาทเทียมแบบแพร่ย้อนกลับ

ขั้นตอนการเรียนรู้ของโครงข่ายประสาทเทียมแบบแพร่ย้อนกลับ แบ่งเป็นขั้นตอนดังต่อไปนี้

1. กำหนดจำนวนโหนดอินพุต (N), จำนวนโหนดเอาต์พุต (J), จำนวนโหนดของชั้นซ่อน (M), ข้อมูลอินพุต และข้อมูลเอาต์พุต ต่อจากนั้นทำการรับจำนวนรอบสูงสุดที่จะทำการเรียนรู้ (R) และค่าผิดพลาดที่ยอมรับได้

2. ตั้งค่าพารามิเตอร์ของอัตราการเรียนรู้ (η) ให้อยู่ในช่วง [0, 1]

3. สุ่มน้ำหนักเริ่มต้นให้กับทุกๆเส้นในโครงข่ายประสาทเทียมในทั้ง 2 ชั้น โดยให้มีค่าอยู่ระหว่าง [-0.5, 0.5]

4. รับค่าอินพุตของข้อมูลชุดแรก เพื่อใช้ในการคำนวณหาค่าเอาต์พุตของโครงข่ายประสาทเทียม

5. คำนวณค่าเอาต์พุตของชั้นซ่อน นำค่าอินพุตของชุดที่จะทำการคำนวณหาค่าเอาต์พุตของชั้นซ่อนออกมา แล้วจึงทำการคำนวณค่าเอาต์พุตของชั้นซ่อน ก่อนทำการปรับค่า (Activation) แล้วทำการปรับค่าเอาต์พุตของชั้นซ่อน ให้อยู่ในช่วง [0, 1] สำหรับแต่ละโหนดของชั้นซ่อน

ค่าเอาต์พุตของชั้นซ่อนก่อนทำการปรับค่า มีสมการดังนี้

$$s_m = \sum_{n=1}^{N} x_n * w_{nm} \tag{2-1}$$

กำหนดให้: s_m = เอาต์พุตของชั้นซ่อนก่อนทำการปรับค่า (activation)

x_n = อินพุตโหนดที่ n มีทั้งหมด N โหนด

w_{nm} = น้ำหนักของเส้นเชื่อมระหว่างชั้นอินพุตกับชั้นซ่อน

ค่าเอาต์พุตของชั้นซ่อนหลังทำการปรับค่า มีสมการดังนี้

$$y_m = f(s_m) \tag{2-2}$$

กำหนดให้: y_m = เอาต์พุตของชั้นซ่อน หลังทำการปรับค่าของโหนดที่ m มีทั้งหมด M โหนด

ฟังก์ชั่นที่ใช้ปรับค่า f(x) มีสมการดังนี้

$$f(x) = \frac{1}{1+e^{-x}} \tag{2-3}$$

6. คำนวณค่าเอาต์พุตของชั้นเอาต์พุต ก่อนทำการปรับค่า แล้วทำการปรับค่าเอาต์พุตของชั้นเอาต์พุตให้อยู่ในช่วง [0, 1] สำหรับแต่ละโหนดของชั้นเอาต์พุต ค่าเอาต์พุตของชั้น

$$v_j = \sum_{m=1}^{M} y_m * w_{mj} \tag{2-4}$$

กำหนดให้: v_j = เอาต์พุตของชั้นเอาต์พุตก่อนทำการปรับค่า (activation)

w_{mj} = น้ำหนักของเส้นเชื่อมระหว่างชั้นซ่อนกับชั้นเอาต์พุต

ค่าเอาต์พุตของชั้นเอาต์พุตหลังทำการปรับค่า มีสมการดังนี้

$$z_j = f(v_j) \tag{2-5}$$

กำหนดให้: z_j = ค่าเอาต์พุตที่ได้ทำการปรับค่าแล้วของชั้นเอาต์พุตโหนดที่ j มีทั้งหมด J โหนด

7. หาค่าความผิดพลาดและปรับน้ำหนัก นำเอาต์พุตที่ได้ กับ เอาต์พุตที่กำหนดไว้ มาหาค่าความผิดพลาดของข้อมูล ถ้าค่าผิดพลาดของข้อมูลน้อยกว่าค่าผิดพลาดที่ยอมรับได้ จึงทำการรับข้อมูลชุดต่อไป ถ้าไม่ใช่ปรับน้ำหนักแล้วทำการรับข้อมูลของชุดถัดไปแล้วจึงกลับไปทำ ข้อ 5 แต่ถ้าเป็นข้อมูลชุดสุดท้ายทำข้อ 8 ค่าความผิดพลาดในแต่ละชุดของข้อมูลตัวอย่าง มีสมการดังนี้

$$err = \frac{1}{J}\sum_{j=1}^{J}(t_j - z_j)^2 \tag{2-6}$$

กำหนดให้: err = ค่าผิดพลาดของข้อมูลตัวอย่าง

t_j = ค่าเอาต์พุตที่ต้องการที่ชั้นเอาต์พุต โหนดที่ j มีทั้งหมด J โหนด

z_j = ค่าเอาต์พุตที่ได้ทำการปรับค่าแล้วของชั้นเอาต์พุตโหนดที่ j มีทั้งหมด J โหนด

การปรับน้ำหนัก มีสมการดังนี้

$$w_{mj}^{(r+1)} = w_{mj}^{(r)} + \eta\left\{\left(t_j^{(q)} - z_j^{(q)}\right) * \left[z_j^{(q)}(1 - z_j^{(q)})\right] * \left[y_m^{(q)}\right]\right\} \tag{2-7}$$

$$w_{nm}^{(r+1)} = w_{nm}^{(r)} + \eta\{\sum_{j=1}^{J}(t_j^{(q)} - z_j^{(q)})[z_j^{(q)}(1 - z_j^{(q)})]w_{mj}^{(r)}\} * [y_m^{(q)}(1 - y_m^{(q)})][x_n^{(q)}] \tag{2-8}$$

กำหนดให้: η = อัตราการเรียนรู้มีค่าอยู่ระหว่าง 0 ถึง 1

r = จำนวนรอบที่จะทำการเรียนรู้ มี R เป็นจำนวนรอบที่กำหนด

q = จำนวนชุดของข้อมูลตัวอย่าง มี Q เป็นตัวกำหนด

8. หาค่าผิดพลาดรวมเฉลี่ย นำค่าผิดพลาดของชุดข้อมูลแต่ละชุดมารวมกัน แล้วทำการหาค่าเฉลี่ยเพื่อใช้ในการตรวจสอบว่าผลลัพธ์ของทุกๆ ข้อมูลในแต่ละรอบนั้นมีค่าน้อยกว่าค่าผิดพลาดที่ยอมรับได้ในทุกๆ ข้อมูลหรือไม่ ถ้าใช่แสดงว่าโครงข่ายประสาทเทียมสามารถให้ผลลัพธ์ที่ถูกต้องของทุกๆข้อมูลแล้วจบการเรียนรู้ ถ้าไม่ใช่กลับไปทำข้อ 4

ค่าผิดพลาดรวมเฉลี่ย

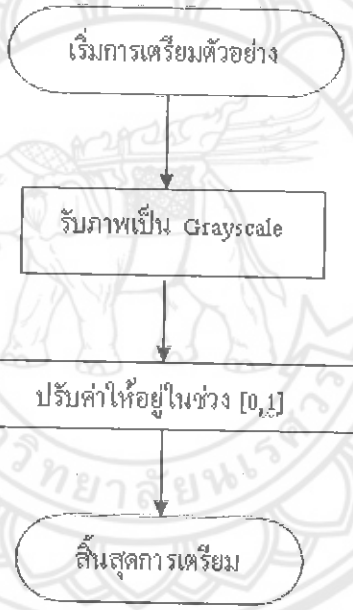
$$E = \frac{1}{Q}\sum_{q=1}^{Q} e^{(q)} \tag{2-9}$$

กำหนดให้: E = ค่าผิดพลาดรวมเฉลี่ยของข้อมูลตัวอย่าง

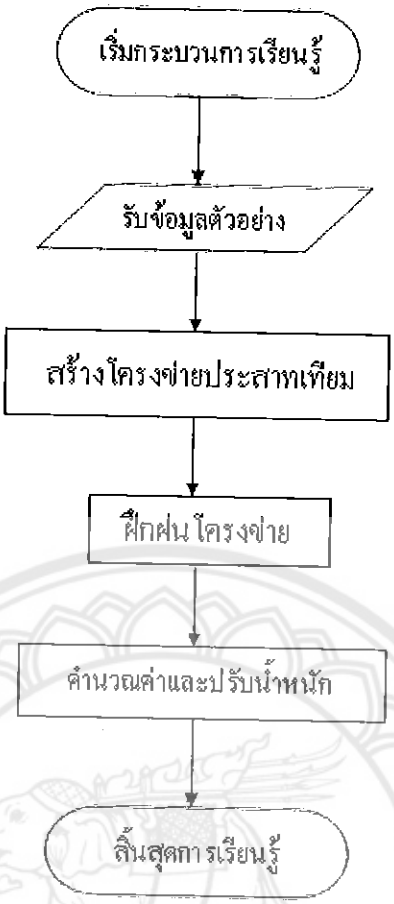
บทที่ 3
วิธีการดำเนินโครงงาน

3.1 การเตรียมภาพตัวอย่าง

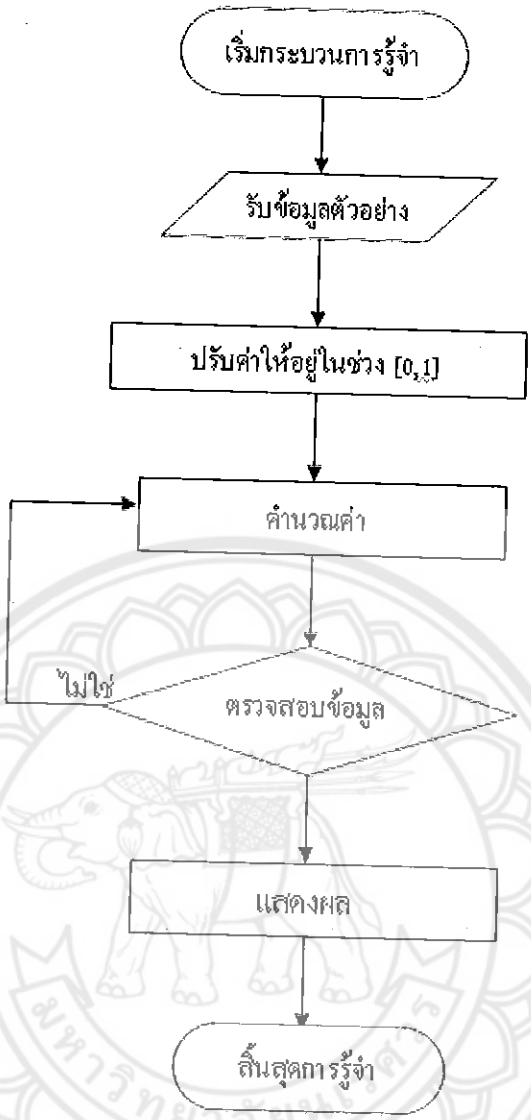
วิธีการรู้จำตัวอักษรภาษาอังกฤษโดยใช้โครงข่ายประสาทเทียม โดยรับภาพ grayscale และทำการปรับค่าเพื่อให้เหมาะสมโดยให้อยู่ระหว่าง 0 ถึง 1 และนำเข้าโครงข่ายประสาทเทียมเพื่อคำนวณหาค่าน้ำหนักที่เหมาะสม เพื่อที่จะได้ดัชนีค่าของเอาต์พุตของแต่ละตัวอักษร จะใช้หลักการและทฤษฎีดังที่กล่าวไว้ในบทที่ 2 คือการคำนวณโดยใช้โครงข่ายประสาทเทียม แบ่งขั้นตอนการประมวลผลออกเป็น 3 ส่วน คือ การเตรียมตัวอย่าง กระบวนการเรียนรู้ และกระบวนการรู้จำ แสดงในรูปที่ 3.1 3.2 และ 3.3 ตามลำดับ



รูปที่ 3.1 แสดงแผนผังขั้นตอนการทำงานของการเตรียมตัวอย่าง



รูปที่ 3.2 แสดงแผนผังขั้นตอนการเรียนรู้ตัวอักษรโดยใช้โครงข่ายประสาทเทียม



รูปที่ 3.3 แสดงแผนผังขั้นตอนการทำงานของการรู้จำตัวอักษรโดยใช้โครงข่ายประสาทเทียม

รูปที่ 3.1 แสดงแผนผังขั้นตอนการทำงานของการเตรียมภาพตัวอย่างก่อนจะนำเข้าสู่โครงข่ายประสาทเทียม รูปที่ 3.2 แสดงแผนผังขั้นตอนการทำงานของการเรียนรู้ตัวอักษรโดยใช้โครงข่ายประสาทเทียม และรูปที่ 3.3 แสดงแผนผังขั้นตอนการทำงานของการรู้จำตัวอักษรโดยใช้โครงข่ายประสาทเทียม

ต่อไปจะเป็นการอธิบายลงไปในแต่ละขั้นตอนดังที่ปรากฏตัวเลขขั้นตอนในแผนผังทั้ง 3 ส่วน หลังจากมีการรับภาพต้นฉบับซึ่งเป็นแบบ grayscale เข้ามาโดยจะนำภาพอักษรปกติเข้ามา โดยไม่มีสัญญาณรบกวน เพื่อลดมิติข้อมูลภาพที่จะใช้เป็นข้อมูลรู้จำให้แก่โครงข่าย

รูปที่ 3.4 แสดงตัวอย่างภาพต้นฉบับ

รูปที่ 3.4 แสดงตัวอย่างภาพปกติเป็นภาพต้นแบบที่จะใช้ในการฝึกฝนโครงข่ายประสาทเทียม มีขนาด 8x6 พิกเซล แบบ grayscale ซึ่งจะเป็นกลุ่มภาพคือตั้งแต่ A ถึง Z และ a ถึง z และเครื่องหมายอีก 5 แบบ

255	255	255	255	255	255
255	255	255	255	255	255
255	191	[illegible]	[illegible]	[illegible]	255
255	255	240	45	201	255
255	255	60	180	255	255
255	180	54	210	250	255
255	190	100	100	180	255
255	255	255	255	255	255

รูปที่ 3.5 แสดงสีแต่ละจุดของภาพ

ภาพที่ 3.5 แสดงสีแต่ละจุดของภาพ ซึ่งจะมีขนาดเท่ากับภาพต้นแบบเดิมคือมีขนาด 8x6 พิกเซล

0	0	0	0	0	0
0	0	0	0	0	0
0	0.25	0.50	0.50	0.25	0
0	0	0.05	0.82	0.21	0
0	0	0.76	0.29	0	0
0	0.29	0.78	0.17	0.01	0
0	0.25	0.60	0.60	0.29	0
0	0	0	0	0	0

รูปที่ 3.6 แสดงภาพตัวอักษรที่ทำการปรับค่าให้อยู่ช่วง 0 ถึง 1

รูปที่ 3.6 แสดงภาพตัวอักษรที่ทำการปรับค่าให้อยู่ช่วง 0 ถึง 1 มีขนาด 8x6 พิกเซล เพื่อป้อนค่าที่เหมาะสมให้กับ โครงข่ายประสาทเทียม โดยที่การปรับค่าเป็นไปตามสมการ ดังนี้

$$E_{(i,j)} = \frac{Max - e_{(i,j)}}{Max} \tag{3-1}$$

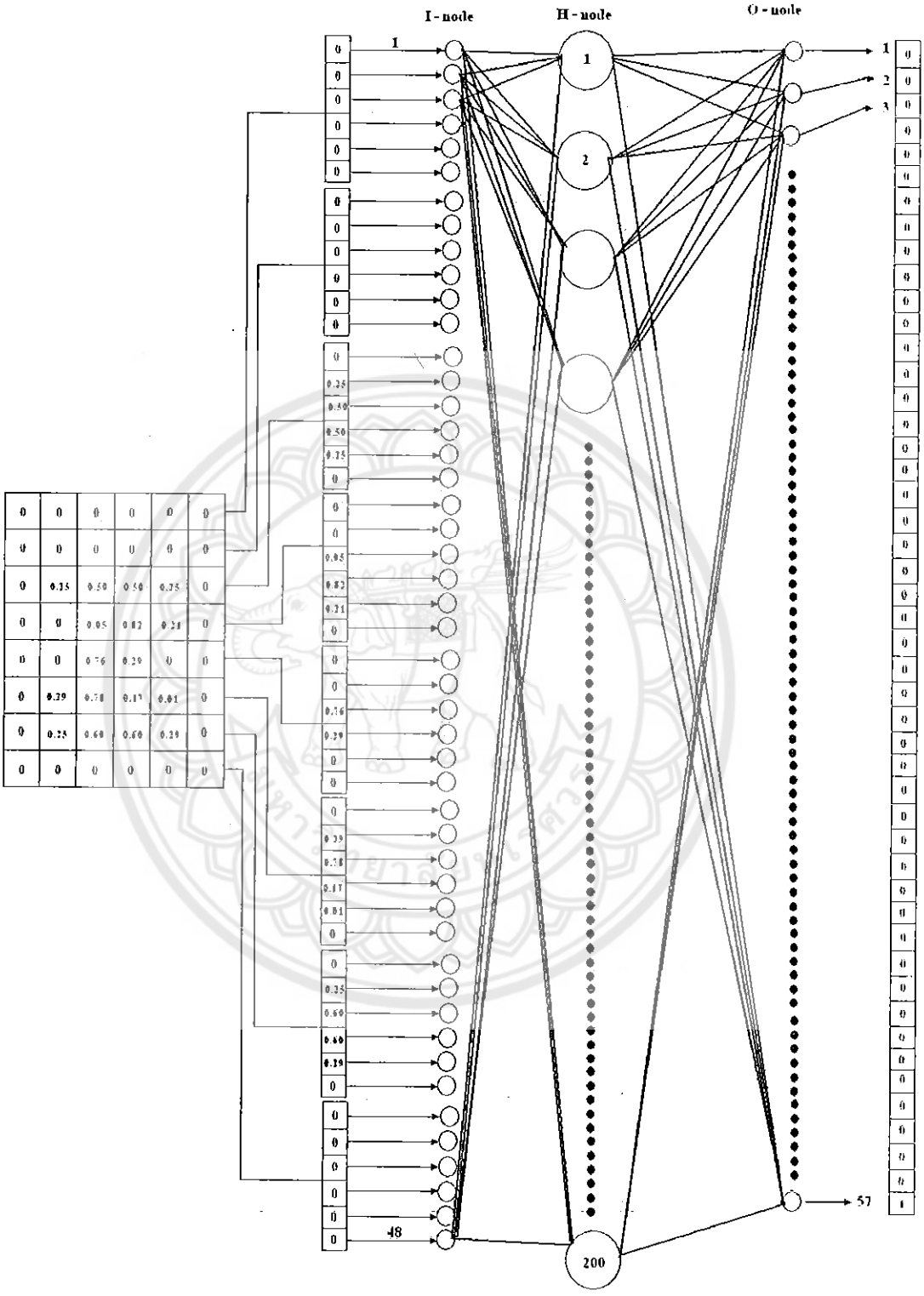
เมื่อ $e_{(i,j)}$ เป็นค่าที่จุดนั้นๆ

Max เป็นค่าสูงสุดในเมทตริกซ์

3.2 กระบวนการเรียนรู้โครงข่ายประสาทเทียมแบบแพร่กลับ

การกำหนดจำนวนหน่วยในแต่ละชั้น

โครงข่ายประสาทเทียมกำหนดให้มี 3 ชั้นคือ ชั้นรับข้อมูล ชั้นซ่อน และชั้นผลลัพธ์ ชั้นรับข้อมูลจะทำการรับค่าตัวแปรไว้เพื่อรับค่าอินพุตสำหรับจัดเก็บไว้ในอาร์เรย์มีขนาด 48 หน่วย ที่ชั้นซ่อนจะมีการปรับเปลี่ยนตามขนาดของชั้นรับข้อมูล และชั้นผลลัพธ์ จะมีเวกเตอร์ขนาดเท่ากับจำนวนตัวอักษรที่เข้าการฝึกฝนคือ 57 แต่ละตำแหน่งจะตรงกับเลขที่ชนิดของรูปแบบ



รูปที่ 3.7 แสดงภาพโครงสร้างของโครงข่ายประสาทเทียมที่ใช้ในโปรแกรม

จากรูปที่ 3.7 จะแสดงภาพของโครงข่ายประสาทเทียมโดยมีส่วนประกอบดังนี้ ชั้นรับข้อมูลมีทั้งหมด 48 โหนด ซึ่งจะรับค่ามาจากข้อมูลตัวอย่างที่ได้ทำการเตรียมไว้แล้ว ชั้นซ่อนมีจำนวน 200 โหนด ชั้นผลลัพธ์มีจำนวน 57 โหนด ซึ่งจะเก็บไว้เป็นค่าตำแหน่ง จากรูปจะมีค่าตำแหน่งอยู่ในลำดับสุดท้าย จึงจะได้ค่าเอาท์พุตเท่ากับ

z = 00000000000000000000000000000000000000000000000000000001

การหยุดการเรียนรู้ของโครงข่ายประสาทเทียม

ในการเรียนรู้ของโครงข่ายประสาทเทียมแต่ละรอบ จะเป็นการเปรียบเทียบค่าความผิดพลาดรวมเฉลี่ย (Mean Absolute Error) ของทุกๆรูปแบบกับค่าผิดพลาดสูงสุดที่ยอมรับได้ (Error Tolerance) โดยในการทดลองนี้เรากำหนดค่าผิดพลาดสูงสุดที่ยอมรับได้คือ 0.001 หากมีค่าน้อยกว่า 0.001 ก็จะหยุดการเรียนรู้

3.3 การรู้จำตัวอักษรภาษาโดยใช้โครงข่ายประสาทเทียม

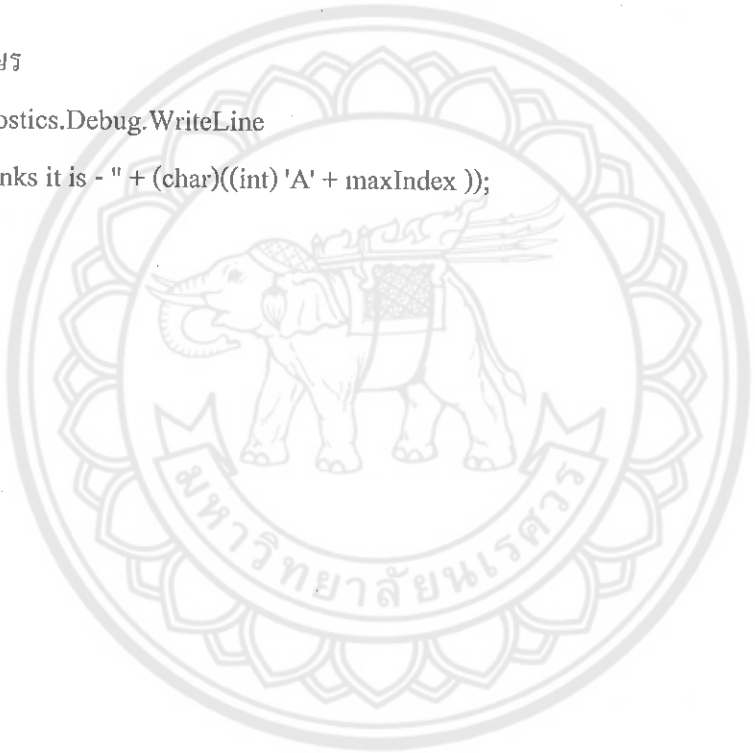
การรู้จำตัวอักษรนั้นจะทำได้โดยการรับค่าตัวแปรที่เราทำการคีย์เข้าไปจากนั้นก็นำมาปรับค่าให้อยู่ในช่วง 0 ถึง 1 และทำการนำเข้าโครงข่ายประสาทเทียมซึ่งจะทำการคำนวณค่าและปรับน้ำหนักเช่นเดียวกับรูปที่ 3.7 จากนั้นจะหาค่าดัชนีที่มากสุดของเอาต์พุตที่ได้ซึ่งจะเป็นตัวที่บอกว่าเป็นรูปแบบใด

ตัวอย่างโปรแกรมที่ใช้ในการรู้จำ

```
Double [] pattern = new double[] {
  1, 0, 0,  1,  1,
  1, 0,  1, 0, 0,
  1,  1, 0, 0, 0,
  1, 0,  1, 0, 0,
  1, 0, 0,  1, 0,
  1, 0, 0, 0, 1};
// ได้เอาต์พุตของโครงข่าย
float[] output = neuralNet.Compute(pattern);

int i, n, maxIndex = 0;
```

```
// หาค่าสูงสุดจากเอาต์พุต
float max = output[0];
for (i = 1, n = output.Length; i < n; i++)
{
   if (output[i] > max)
   {
      max = output1[i];
      maxIndex = i;
   }
}
// แสดงตัวอักษร
System.Diagnostics.Debug.WriteLine
( " network thinks it is - " + (char)((int) 'A' + maxIndex ));
```



บทที่ 4
ผลการทดลองและการวิเคราะห์

4.1 ผลการทดลอง

การรู้จำตัวอักษร เริ่มแรกจะทำการให้โครงข่ายประสาทเทียมทำการรู้จำตัวอักษรและสัญลักษณ์ทั้งหมด 57 ตัว โดยเริ่มต้น การฝึกสอนจะให้โครงข่ายประสาทเทียมรู้จำตัวอักษรปกติ โดยทำการส่งภาพขนาด 8 x 6 เข้าไปทำตามขั้นตอนการทำงานของโปรแกรม การเพิ่มสัญญาณรบกวนให้กับข้อมูลมี 2 ขั้นตอน คือ การเพิ่มสัญญาณรบกวนให้กับรูปแบบ ซึ่งการเพิ่มสัญญาณนี้จะทำการสุ่มค่ารูปแบบที่แตกต่างจากรูปแบบที่มีอยู่ทำให้เกิดรูปแบบที่มากขึ้นเพื่อที่จะนำไปฝึกสอนให้โครงข่ายประสาทเทียมมีการเรียนรู้ที่หลากหลายและแม่นยำมากขึ้น ขั้นตอนนี้จะทำก่อนการฝึกสอนให้โครงข่ายประสาทเทียม และอีกขั้นตอนหนึ่งจะทำเมื่อโครงข่ายประสาทเทียมฝึกสอนเรียบร้อยแล้วพร้อมที่จะทดสอบ นั่นคือการเพิ่มระดับสัญญาณรบกวนให้กับข้อมูล

กำหนดให้: $N = \frac{r}{100} * (w * h)$ (4-1)

N = จำนวน pixel ที่จะเพิ่มสัญญาณรบกวนในภาพ

r = ระดับสัญญาณรบกวน (5 ,10, 15, 20)

w = จำนวนคอลัมน์ (6 pixel)

h = จำนวนแถว (8 pixel)

i = random (h)

j = random (w)

c = ค่าของ gray level ในตำแหน่ง ($0 \leq c \leq 255$)

psudo code สำหรับการเพิ่มสัญญาณรบกวนในภาพ

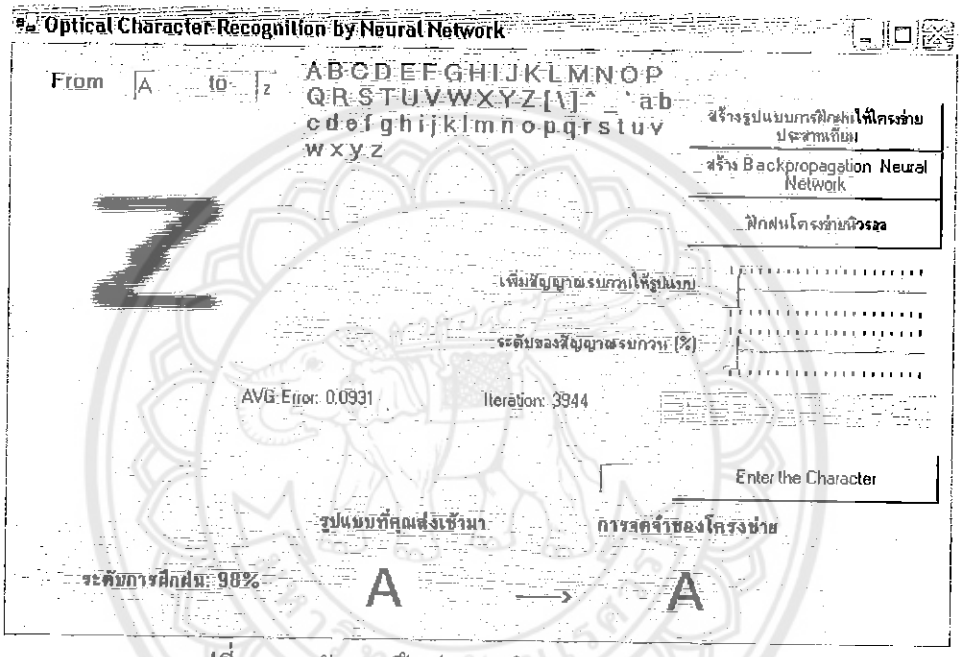
```
for k = 1 : N
{
        i = random (h)
        j = random (w)
        check i และ j จะต้องไม่ซ้ำกันในแต่ละภาพ
        X(i, j) = random (c)
}
```

สรุปออกมาเป็นสมการได้ดังนี้ $X_k(i, j) = \text{random}(c)$ โดยที่ $1 \leq k \leq N$ (4-2)

โดยจะเพิ่มสัญญาณรบกวนที่ระดับ 5% 10% 15% และ 20% เพื่อศึกษาความสามารถและผลการรู้จำของโครงข่ายประสาทเทียม

4.1.1 ตัวอักษรปกติเพื่อการฝึกสอนโครงข่ายประสาทเทียม

ตัวอักษรชุดนี้ประกอบไปด้วยพยัญชนะ สระภาษาอังกฤษและสัญลักษณ์ ทั้งหมด 57 ตัว ที่จะส่งเข้าไปในโครงข่ายประสาทเทียมเพื่อฝึกสอนให้โครงข่ายประสาทเทียมมีความรู้ สามารถรู้จำตัวอักษรได้ และจากการทดสอบให้โครงข่ายประสาทเทียมมีระดับการฝึกฝนเท่ากับ 98%

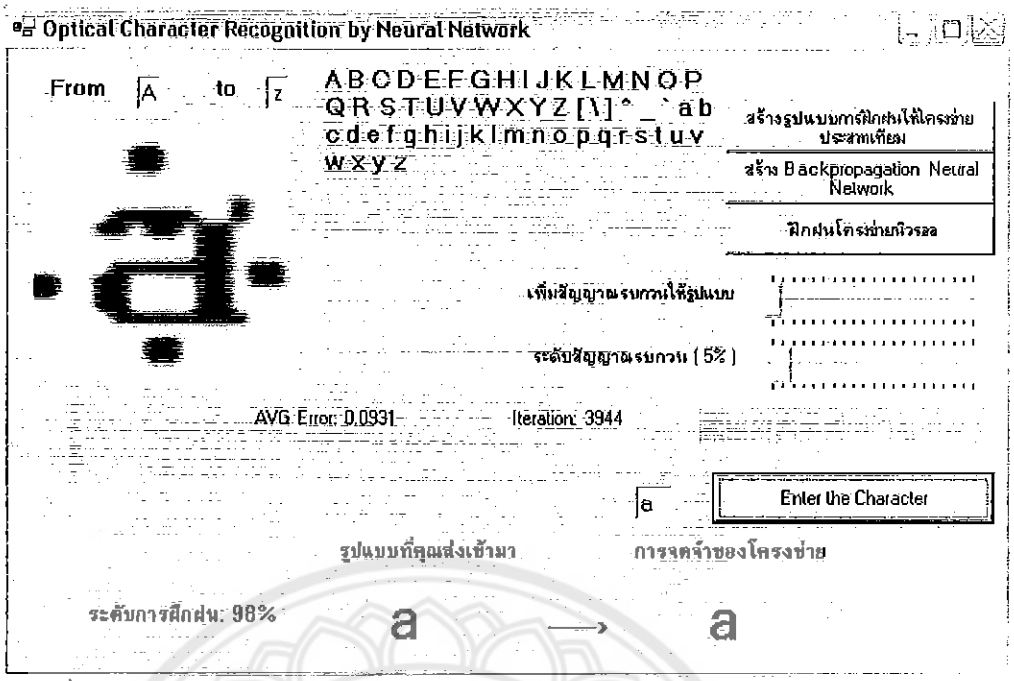


รูปที่ 4.1 ระดับการฝึกฝนของโครงข่ายประสาทเทียม

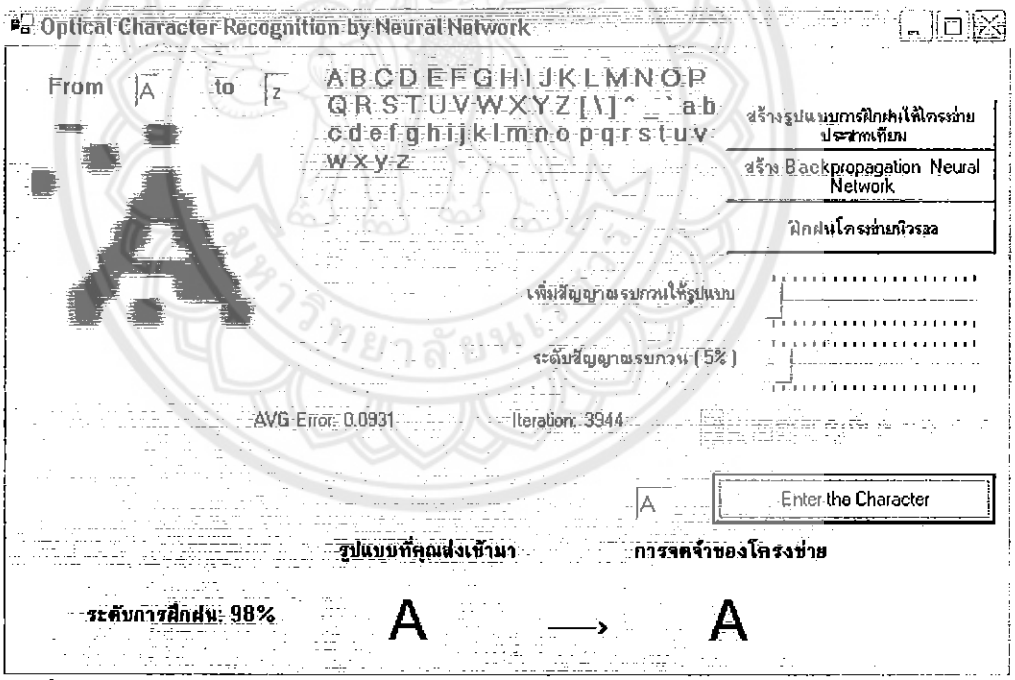
รูปที่ 4.2 ชุดอักษรที่ใช้ในการฝึกสอนให้กับโครงข่ายประสาทเทียม

4.1.2 ผลการทดลองการรู้จำโดยเพิ่มสัญญาณรบกวน ขนาด 5 %

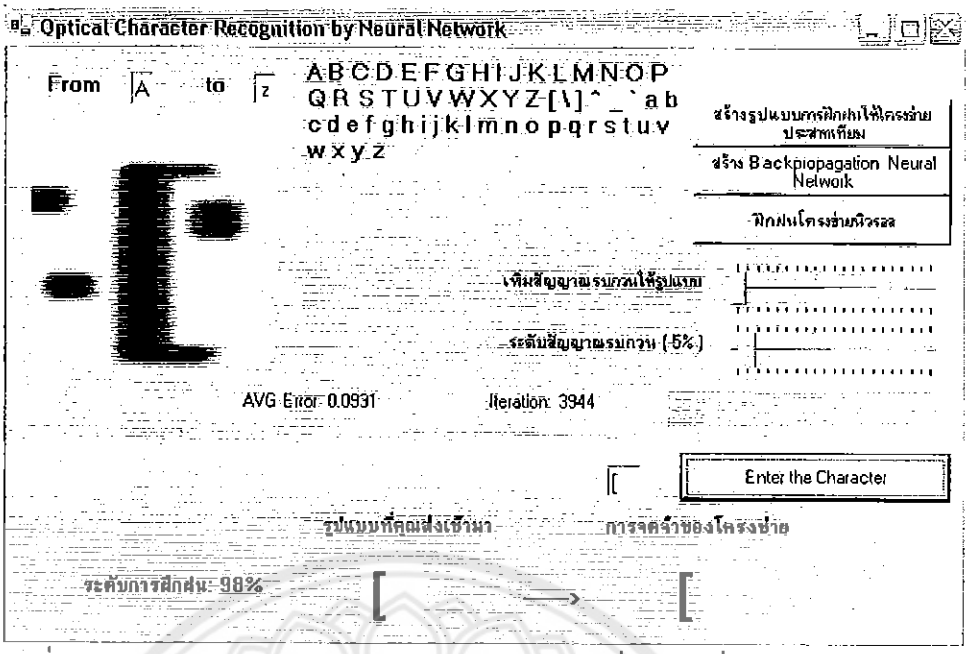
จากภาพตัวอย่างการทดสอบการรู้จำ ที่ทำการเพิ่มสัญญาณรบกวน ขนาด 5 % ให้กับภาพต้นฉบับ จะเห็นว่า ภาพยังมีส่วนสำคัญของตัวอักษรอยู่มาก ความสามารถในการรู้จำของโครงข่ายประสาทเทียมจึงมีมาก การทดสอบกับตัวอักษรทั้งหมด ให้ผลการทดสอบระหว่างข้อมูลเข้าและข้อมูลออกตรงกัน 100%



รูปที่ 4.3 แสดงภาพตัวอย่างของการทดสอบตัวพิมพ์เล็กที่ทำการเพิ่มสัญญาณรบกวน 5%



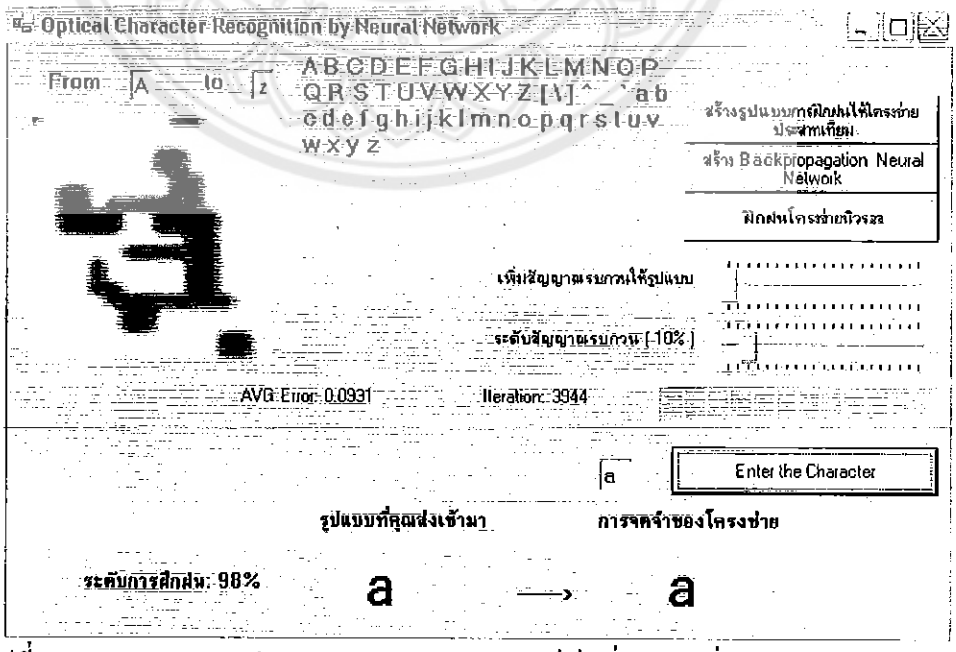
รูปที่ 4.4 แสดงภาพตัวอย่างของการทดสอบตัวพิมพ์ใหญ่ที่ทำการเพิ่มสัญญาณรบกวน 5%



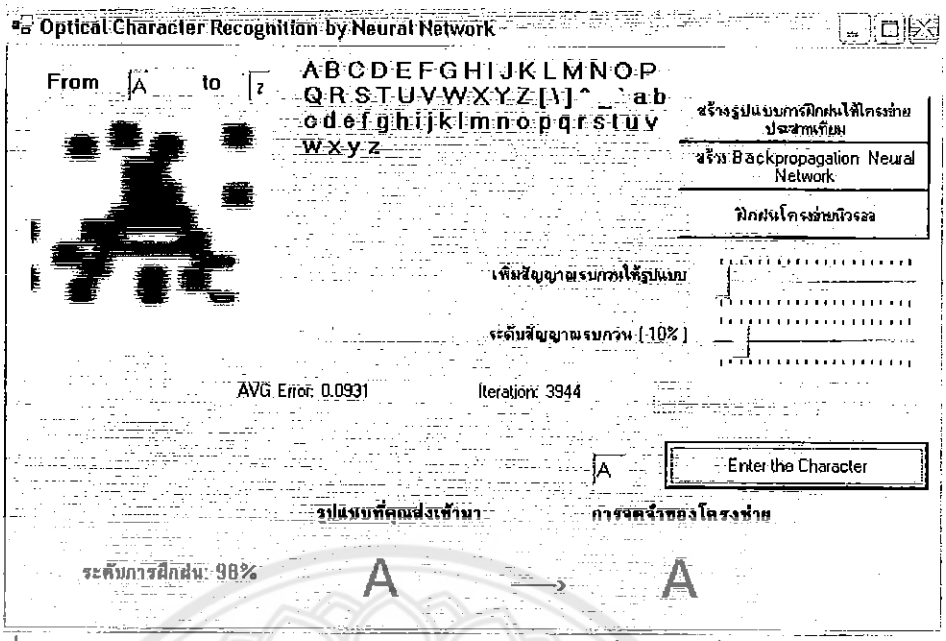
รูปที่ 4.5 แสดงภาพตัวอย่างของการทดสอบสัญลักษณ์ที่ทำการเพิ่มสัญญาณรบกวน 5%

4.1.3 ผลการทดลองการรู้จำโดยเพิ่มสัญญาณรบกวน ขนาด 10 %

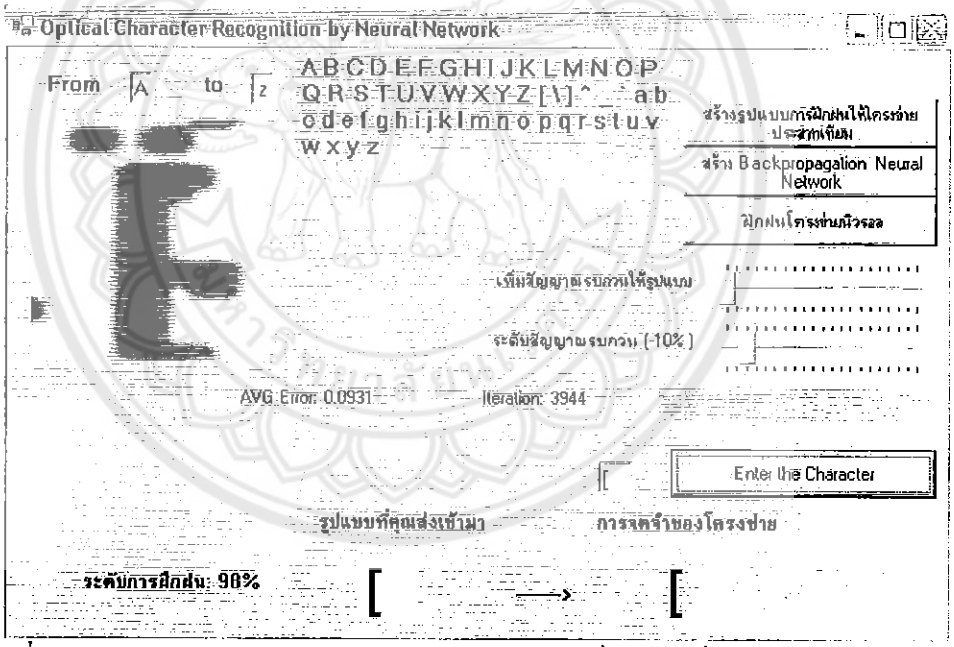
จากภาพตัวอย่างการทดสอบการรู้จำที่ทำการเพิ่มสัญญาณรบกวน ขนาด 10 % ให้กับภาพต้นฉบับ จะเห็นว่าสัญญาณรบกวนเข้าไปทำให้ส่วนสำคัญของตัวอักษรลดลง มีผลทำให้ความสามารถในการรู้จำของโครงข่ายประสาทเทียมลดลงเช่นกัน การทดสอบกับตัวอักษรทั้งหมดให้ผลการทดสอบระหว่างข้อมูลเข้าและข้อมูลออกตรงกัน 82.46%



รูปที่ 4.6 แสดงภาพตัวอย่างของการทดสอบตัวพิมพ์เล็กที่ทำการเพิ่มสัญญาณรบกวน 10%



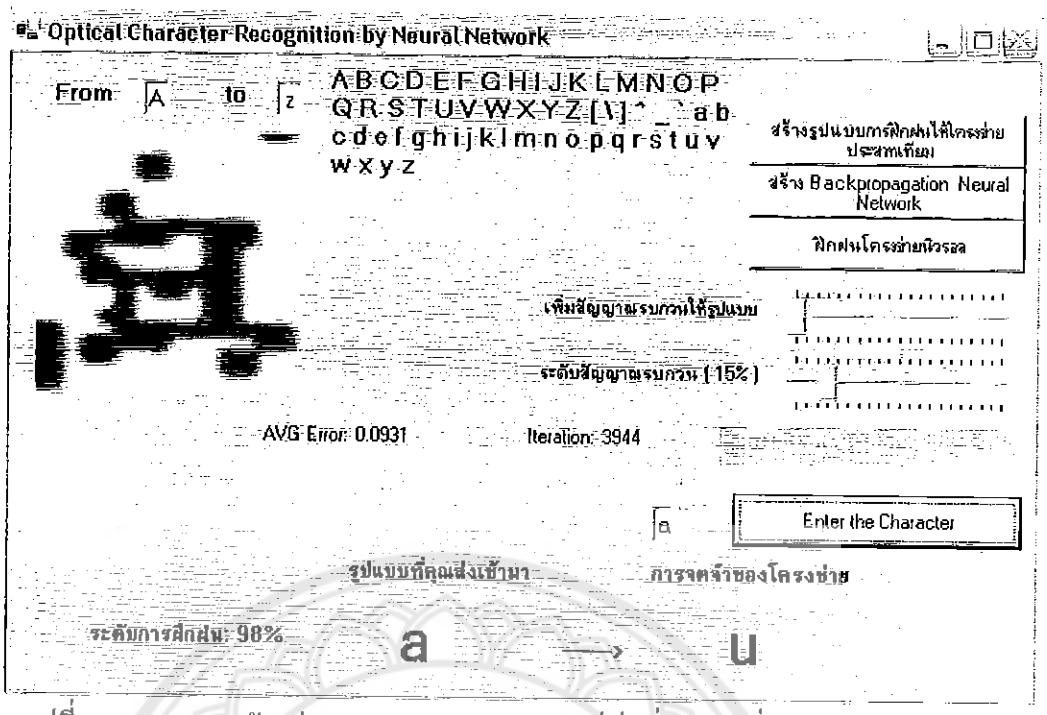
รูปที่ 4.7 แสดงภาพตัวอย่างของการทดสอบตัวพิมพ์ใหญ่ที่ทำการเพิ่มสัญญาณรบกวน 10%



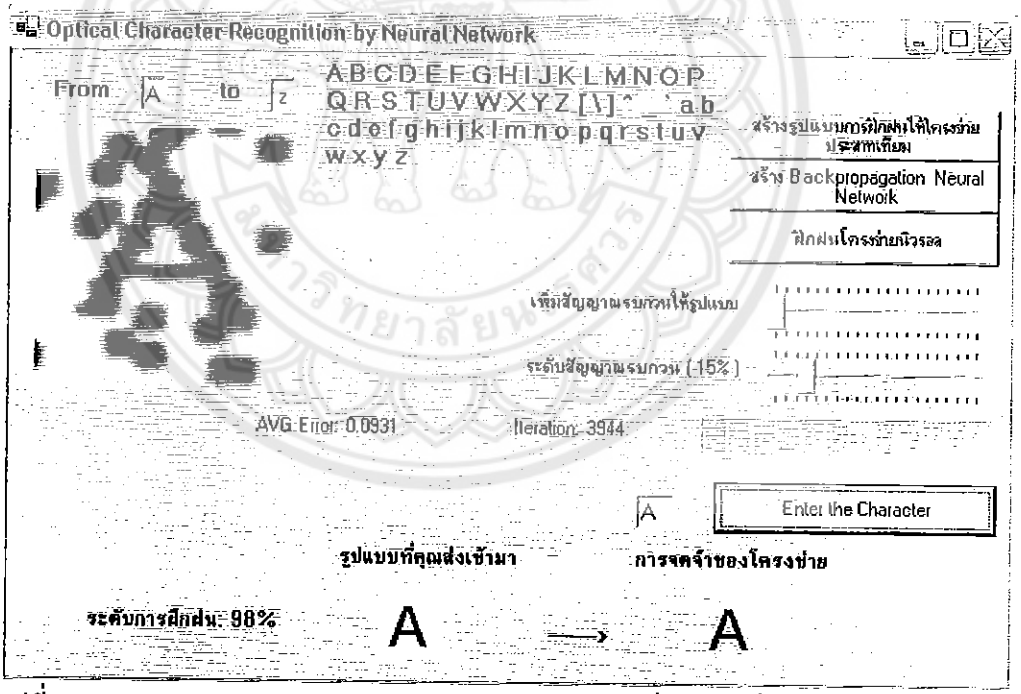
รูปที่ 4.8 แสดงภาพตัวอย่างของการทดสอบสัญลักษณ์ที่ทำการเพิ่มสัญญาณรบกวน 10%

4.1.4 ผลการทดลองการรู้จำโดยเพิ่มสัญญาณรบกวน ขนาด 15 %

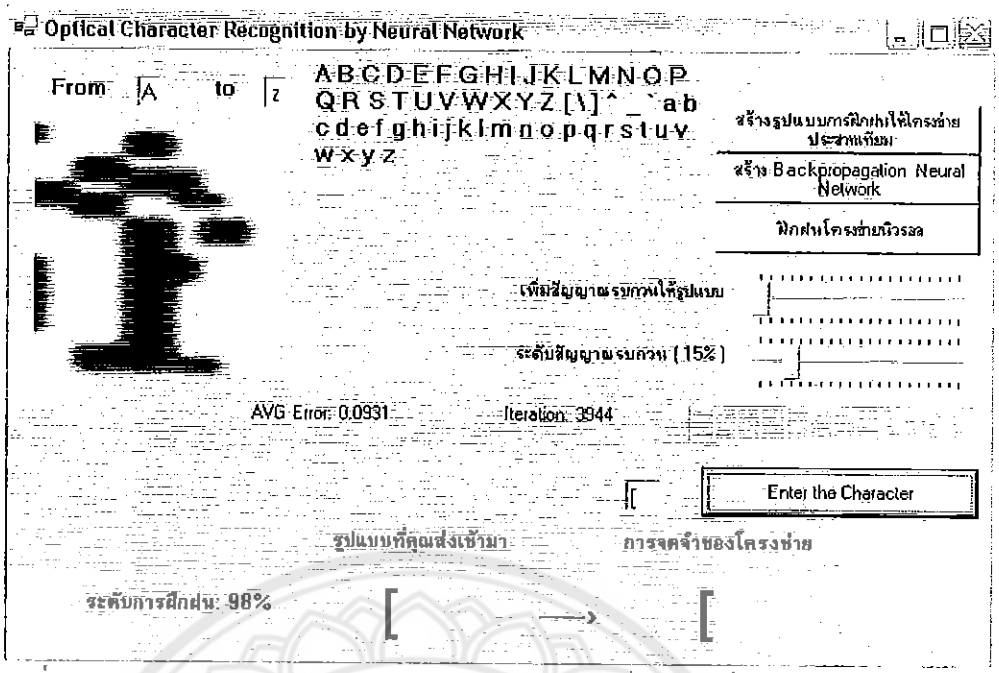
จากภาพตัวอย่างการทดสอบการรู้จำที่ทำการเพิ่มสัญญาณรบกวน ขนาด 15 % ให้กับภาพต้นฉบับ จะเห็นว่าสัญญาณรบกวนเข้าไปทำให้ส่วนสำคัญของตัวอักษรลดลงไปมาก มีผลทำให้ความสามารถในการรู้จำของโครงข่ายประสาทเทียมลดลงเช่นกัน เมื่อทำการทดสอบกับตัวอักษรทั้งหมดจะให้ผลการทดสอบระหว่างข้อมูลเข้าและข้อมูลออกตรงกัน 70.18%



รูปที่ 4.9 แสดงภาพตัวอย่างของการทดสอบตัวพิมพ์เล็กที่ทำการเพิ่มสัญญาณรบกวน 15%



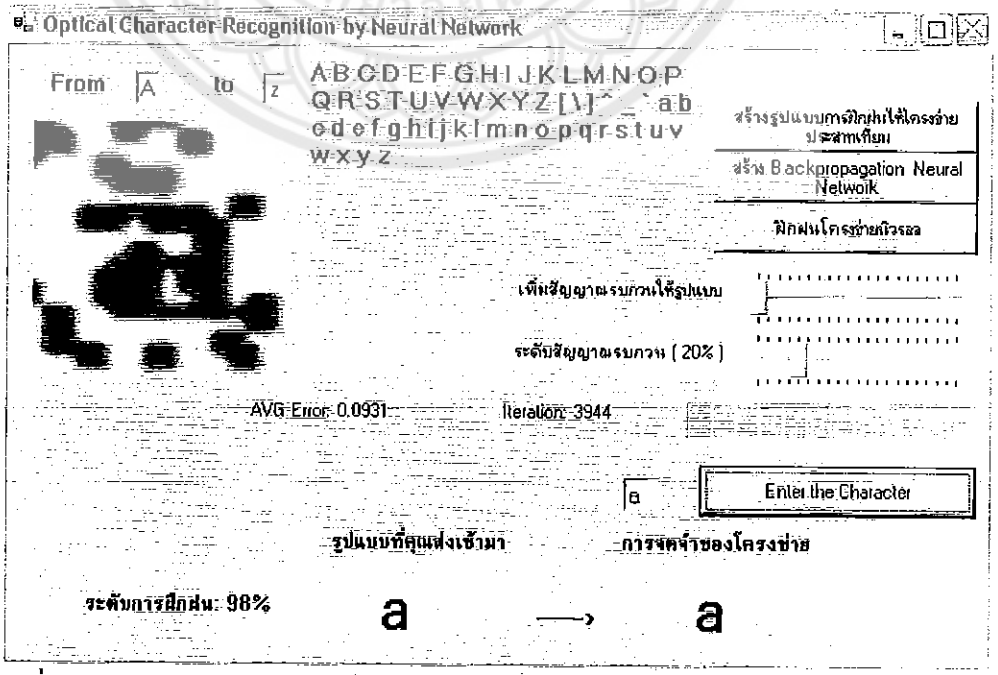
รูปที่ 4.10 แสดงภาพตัวอย่างของการทดสอบตัวพิมพ์ใหญ่ที่ทำการเพิ่มสัญญาณรบกวน 15%



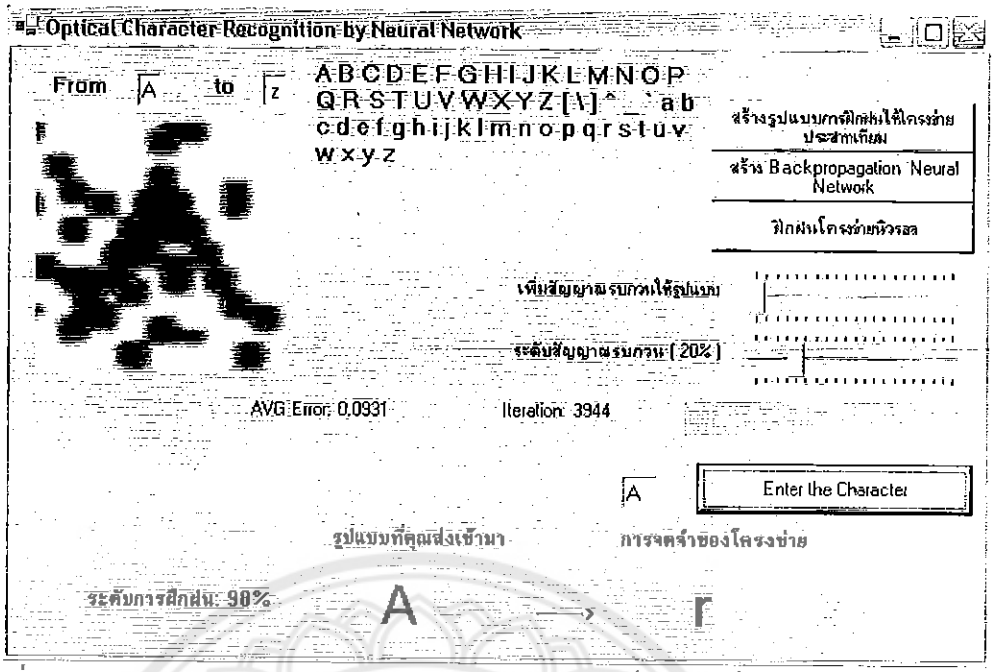
รูปที่ 4.11 แสดงภาพตัวอย่างของการทดสอบสัญลักษณ์ที่ทำการเพิ่มสัญญาณรบกวน 15%

4.1.5 ผลการทดลองการรู้จำโดยเพิ่มสัญญาณรบกวน ขนาด 20 %

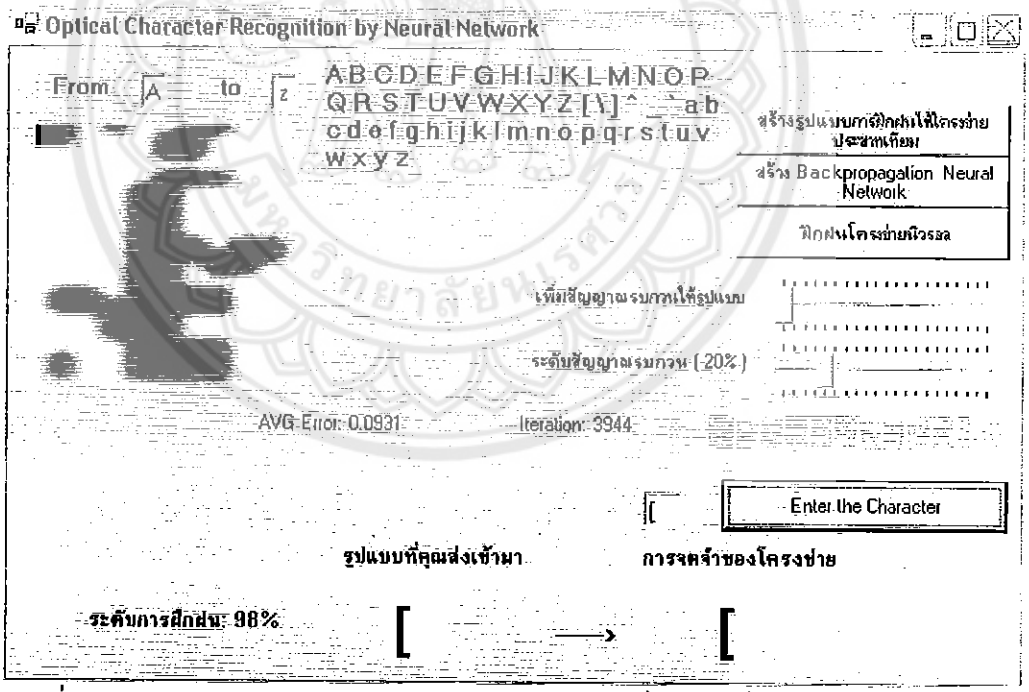
จากภาพตัวอย่างการทดสอบการรู้จำ ที่ทำการเพิ่มสัญญาณรบกวนขนาด 20 % ให้กับภาพต้นฉบับ จะเห็นว่าสัญญาณรบกวนไปทำให้ส่วนสำคัญของตัวอักษรลดลงไปอย่างมาก มีผลทำให้ความสามารถในการรู้จำของโครงข่ายประสาทเทียมลดลง เมื่อทำการทดสอบกับตัวอักษรทั้งหมดจะให้ผลการทดสอบระหว่างข้อมูลเข้าและข้อมูลออกตรงกัน 59.65%



รูปที่ 4.12 แสดงภาพตัวอย่างของการทดสอบตัวพิมพ์เล็กที่ทำการเพิ่มสัญญาณรบกวน 20%



รูปที่ 4.13 แสดงภาพตัวอย่างของการทดสอบตัวพิมพ์ใหญ่ที่ทำการเพิ่มสัญญาณรบกวน 20%



รูปที่ 4.14 แสดงภาพตัวอย่างของการทดสอบสัญลักษณ์ที่ทำการเพิ่มสัญญาณรบกวน 20%

ห้องสมุดคณะวิศวกรรมศาสตร์

I 5078696 e.2
4900044

4.2 วิเคราะห์ผลการทดลอง

จากการทดสอบการรู้จำของโครงข่ายประสาทเทียม โดยการทำการเพิ่มสัญญาณรบกวนเข้าไปในภาพตัวอักษร ตั้งแต่ขนาด 5% 10% 15% และ 20% ผลการทดสอบที่ได้ คือ เมื่อทำการเพิ่มสัญญาณรบกวนเข้าไปในโปรแกรมโครงข่ายประสาทเทียมยังสามารถรู้จำตัวอักษรเหล่านั้นได้ โดยที่ประสิทธิภาพในการรู้จำจะลดลงเมื่อทำการเพิ่มสัญญาณรบกวนให้มากขึ้น และจากการทดสอบ ตัวอักษรที่เราใช้ทดสอบที่มีผลการทดสอบผิดพลาดมากที่สุดคือ ตัวพิมพ์เล็ก รองลงมาคือ ตัวพิมพ์ใหญ่ และ สัญลักษณ์

มร.
พ483ง
2548

ตัวอักษรที่ทำการรู้จำ	รูปแบบตัวอักษรที่ได้เมื่อปริมาณสัญญาณรบกวนที่เพิ่มขึ้น			
	5%	10%	15%	20%
A	A	A	A	r
B	B	B	B	B
C	C	H	H	C
D	D	O	D	D
E	E	E	E	E
F	F	F	M	?
G	G	G	G	U
H	H	H	H	H
I	I	I	\	I
J	J	J	J	J
K	K	K	K	K
L	L	L	L	L
M	M	M	n	M
N	N	N	N	N
O	O	O	O	O
P	P	P	P	P
Q	Q	\	w	Q
R	R	R	R	R
S	S	S	d	S
T	T	T	T	[
U	U	U	U	R
V	V	W	P	V
W	W	W	W	\
X	X	X	X	K
Y	Y	Y	Y	Y
Z	Z	N	Z	Z
a	a	a	u	a
b	b	a	p	b
c	c	c	c	c
d	d	o	o	d
e	e	e	a	S
f	f	f	t	s
g	g	g	g	a
h	h	d	h	d
i	i	i	i	i
j	j	a	j	j
k	k	k	k	k
l	l	l	\	L
m	m	m	m	m
n	n	n	n	h
o	o	o	d	q
p	p	p	p	a
q	q	q	g	g
r	r	r	r	r
s	s	s	s	c
t	t	t	\	t
u	u	d	d	b
v	v	v	y	R
w	w	w	w	w
x	x	x	x	x
y	y	y	y	v
z	z	z	z	M
[	[	[	[	[
\	\	\	o	\
]	]	]	]	]
^	^	^	^	T
_	_	_	_	_
% ผิดพลาด	0	17.54%	29.82%	40.35%

ตารางที่ 4.1 แสดงการเปรียบเทียบผลการรู้จำของโครงข่ายประสาทเทียม

บทที่ 5
บทสรุป

5.1 สรุปผลการทดลอง

จากบทที่ 4 เป็นผลการทดลองและการวิเคราะห์ ในการทดสอบการรู้จำของโครงข่ายประสาทเทียม จะพบว่าหลังจากที่เราทำการฝึกฝนโครงข่ายประสาทเทียม เมื่อไม่มีสัญญาณรบกวนโครงข่ายสามารถจดจำได้ทุกรูปแบบ ต่อมาเมื่อมีการเพิ่มสัญญาณรบกวน 4 ระดับมาทดสอบการรู้จำโครงข่ายประสาทเทียมก็สามารถรู้จำตัวอักษรได้ แต่ระดับความถูกต้องจะลดลงเมื่อมีการเพิ่มสัญญาณรบกวนมากขึ้น จากตารางที่ 4.1 ซึ่งจะแสดงการเปรียบเทียบผลการรู้จำของโครงข่ายประสาทเทียม สามารถสรุปได้ว่าการรู้จำของโครงข่ายประสาทเทียมสามารถจำได้ แม้ว่าจะมีสัญญาณรบกวนในการฝึกฝน

5.2 ปัญหาในการทดลองและแนวทางแก้ไข

ปัญหา	แนวทางในการแก้ปัญหา
1. รูปแบบโครงข่ายนั้นเป็นแบบพื้นฐานซึ่งรองรับได้เฉพาะรูปแบบที่เราจัดให้	สร้างโครงข่ายให้ใหญ่ขึ้นโดยการเพิ่มโหนดในชั้นซ่อน หรือเพิ่มจำนวนชั้นซ่อน
2. รูปแบบการฝึกฝนนั้นยังไม่แม่นยำ หรือซับซ้อนเกินไปสำหรับโครงข่ายที่จะแยกรูปแบบ	อาจใช้โครงข่ายแบบอื่นหรือเปลี่ยนทฤษฎีการฝึกฝน

5.3 แนวทางในการพัฒนาในอนาคต

1. พัฒนารูปแบบของโปรแกรมให้สามารถนำไปใช้ได้จริง
2. เชื่อมต่อระหว่างโปรแกรมและเครื่องสแกนเนอร์ เพื่อความสะดวกในการใช้งานจริง
3. จัดรูปแบบการแสดงผลให้ง่าย และสะดวกต่อการใช้งานจริง

เอกสารอ้างอิง

[1] C. G. Looney, **"Pattern Recognition using Neural Networks"**, Oxford University Press, Inc., 1997.

[2] Chenwithisuk, K. 1995. **Thai Printed Characters Recognition Using A Neural Network And The Syntatic Method.** M.S. thesis, Chulalongkorn Univ., Bangkok

[3] Choruengwiwat, P., S. Jitapunkul., L. Wuttisittikulkij and P. Seehapan. 1998. **Distinctive Feature Analysis for Thai Handwritten Character Recognition Based On Modified Stroke Changing Sequence.** The 1998 IEEE Asia-Pacific Conference on Circuits and Systems, 543-546.

[4] Kijsirikul, B., S. Sinthupinyo and A. Supanwansa, 1998. **Thai Printed Character Recognition by Combining Inductive Logic Programming with Backpropagation Neural Network.** IEEE Asia-Pacific Conference on Circuits and Systems, 539-542.

[5] Looney, C. 1997. **Pattern Recognition Using Neural Networks Theory and Algorithms for Engineers and Scientists.** Oxford University Press. Inc., New York. 75-78:112-127.

[6] Rao, V. and H. Rao. 1995. **C++ Neural Networks and Fuzzy Logic.** 2th ed., MIS:Press, New York. 123-177.

[7] Srisuk, S. 1998. **New Techniques of Face Localization and Recognition.** M.S. thesis, Mahanakorn Univ., Bangkok

[8] Srisuk, S. 1999. **Thai Printed Character Recognition using the Hausdorff Distance.** The National

ภาคผนวก
โปรแกรมการรู้จำตัวอักษรภาษาอังกฤษ
โดยใช้โครงข่ายประสาทเทียม

โปรแกรมการรู้จำตัวอักษรภาษาอังกฤษโดยใช้โครงข่ายประสาทเทียม

```
public OCRNetwork (Form1 owner, int[] nodesInEachLayer):base(nodesInEachLayer)
        {
                this.owner = owner;
        }
        private int OutputPatternIndex(Pattern pattern)
        {
                for (int i = 0; i<pattern.OutputsCount;i++)
                        if (pattern.Output[i] == 1 )
                                return i;
                return -1;
        }

        public void AddNoiseToInputPattern(int levelPercent)
        {
                int i = ((NodesInLayer(0) - 1) * levelPercent)/100;
                while (i > 0)
                {
                        nodes[(int)(BackPropagationNetwork.Random(0, NodesInLayer(0) - 1))].Value
= BackPropagationNetwork.Random(0, 100);
                        i--;
                }

        }

        public int BestNodeIndex
        {
                get
                {
                        int result = -1;
                        double aMaxNodeValue = 0;
                        double aMinError = double.PositiveInfinity;
                        for (int i = 0; i< this.OutputNodesCount;i++)
                        {
```

```
                        NeuroNode node = OutputNode(i);
                        //Look for a node with maximum value or lesser error
                        if ((node.Value > aMaxNodeValue)||((node.Value >=
aMaxNodeValue)&&(node.Error < aMinError)))
                        {
                                aMaxNodeValue = node.Value;
                                aMinError = node.Error;
                                result = i;
                        }

                }
                return result;
        }
}
//Override method of the base class in order to implement our
//own training method
public override void Train(PatternsCollection patterns)
{
        //Current iteration number
        int  iteration = 0;
        if (patterns != null)
        {
                double error = 0;
                int good = 0;
                while (good < patterns.Count)
                        // Train until all patterns are correct
                {
                        if (Form1.IsTerminated) return;
                        error = 0;
                        owner.progressBar1.Value = good;
                        owner.lblShowLevel.Text = "ระดับการฝึกฝน: " + ((good *
100)/owner.progressBar1.Maximum).ToString() + "%";
                        good = 0;
                        for (int i = 0; i<patterns.Count; i++)
                        {
```

```
                    //Set the input values of the network
                    for (int k = 0; k<NodesInLayer(0); k++)
                            nodes[k].Value = patterns[i].Input[k];
            AddNoiseToInputPattern(owner.trackBarPat.Value);
                    //Run the network
                    this.Run();
                    //Set the expected result
                    for (int k = 0;k< this.OutputNodesCount;k++)
                    {
                            error += Math.Abs(this.OutputNode(k).Error),
                            this.OutputNode(k).Error = patterns[i].Output[k];
                    }
                    //Make the network to remember corresponding output
                    //values. (Teach the network)
                    this.Learn();
                    //See if network did produced correct result during
                    //this iteration
                    if (BestNodeIndex == OutputPatternIndex(patterns[i]))
                            good++;

                    iteration ++;
                    Application.DoEvents();

            }
            //Adjust weights of the links in the network to their
            //average value. (An epoch training technique)

        foreach (NeuroLink link in links)
((EpochBackPropagationLink)link).Epoch(patterns.Count);

                    if ((iteration%2) == 0)
            owner.lblShowItalation.Text = "AVG Error: " + (error / OutputNodesCount).ToString() + "
Iteration: " + iteration.ToString();
                    }
```

```
        }

}
public PatternsCollection CreateTrainingPatterns(Font font)
{
        //Create pattern collection
        // As many inputs (examples) as many elements in digitized image matrix
        // As many outputs as many characters we going to recognize.
        PatternsCollection result = new PatternsCollection(aCharsCount, aMatrixDim * aMatrixDim,
aCharsCount);
        // generate one pattern for each character
        for (int i= 0; i<aCharsCount; i++)
        {
        //CharToDoubleArray creates an image of the character and digitizes it.
        // this method can change to pass actual the image of the character
        double[] aBitMatrix = CharToDoubleArray(Convert.ToChar(aFirstChar + i), font, aMatrixDim, 0);
                //Assign matrix value as input to the pattern
                for (int j = 0; j<aMatrixDim * aMatrixDim; j++)
                        result[i].Input[j] = aBitMatrix[j];
                //Output value set to 1 for corresponding character.
                //Rest of the outputs are set to 0 by default.
                result[i].Output[i] = 1;
        }
        return result;
}

private void ShowNoise(Size sz, Graphics g,int noisePercent)
{
        int range =  sz.Height*sz.Width*noisePercent/200;
        for (int i=0; i<range; i++)
        {
                int x = (int)OCRNetwork.Random(0,sz.Width);
                int y = (int)OCRNetwork.Random(0,sz.Height);
                Rectangle r = new Rectangle(x,y,0,0);
```

```
                r.Inflate(1,1);
                Brush b;
                if ((OCRNetwork.Random(0,100))>80)
                        //80% is black noise, 20% is white noise
                        b = new SolidBrush(Color.White);
                else
                        b = new SolidBrush(Color.Black);

                g.FillRectangle(b,r);
                b.Dispose();
        }

}

public double[] CharToDoubleArray(char aChar, Font aFont, int aArrayDim, int aAddNoisePercent)
{
        double[] result = new double[aArrayDim*aArrayDim];
        Graphics gr = lblShowChar.CreateGraphics();
        Size size =Size.Round(gr.MeasureString(aChar.ToString(),aFont));
        Bitmap aSrc = new Bitmap(size.Width,size.Height);
        Graphics bmp = Graphics.FromImage(aSrc);
        bmp.SmoothingMode = System.Drawing.Drawing2D.SmoothingMode.None;
        bmp.InterpolationMode = System.Drawing.Drawing2D.InterpolationMode.NearestNeighbor;
        bmp.Clear(Color.White);
        bmp.DrawString(aChar.ToString(),aFont,new SolidBrush(Color.Black),new Point(0,0),new
StringFormat());
        ShowNoise(size,bmp,aAddNoisePercent);
        pictureBox1.Image = aSrc;
        Application.DoEvents();
        //aSrc – an image of the character
        //aArrayDim – dimension of the pattern matrix
        //calculate image quotation X step
        double xStep = (double)aSrc.Width/(double)aArrayDim;
        //calculate image quotation Y step
        double yStep = (double)aSrc.Height/(double)aArrayDim;
```

```
            for (int i=0; i<aSrc.Width; i++)
                    for (int j=0;j<aSrc.Height;j++)
                    {
                            //calculate matrix address
                            int x = (int)((i/xStep));
                            int y = (int)(j/yStep);
                            //Get the color of the pixel
                            Color c = aSrc.GetPixel(i,j);
                            //Absolute value of the color, but I guess, it is possible to
                            //use the B component of Alpha color space too...

                    }
            //Scale the matrix to fit values into a range from 0..1 (required by ANN)
            // In this method we look for a maximum value of the element
            //and then divide all elements of the matrix by this maximum value.
            return  Scale(result);
    }
    private double MaxOf(double[] src)
    {
            double res=double.NegativeInfinity;
            foreach (double d in src)
                    if (d>res) res = d;
                    return res;
    }

    private double[] Scale(double[] src)
    {
            double max = MaxOf(src);
            if (max!=0)
            {
                    for(int i=0; i<src.Length; i++)
                            src[i] = src[i]/max;
            }
            return src;
    }
```

ประวัติผู้เขียนโครงงาน

ชื่อ นางสาวพองาม ตั้งจันทร

ภูมิลำเนา 122/5 ม.1 ต.ท่าพระ อ.เมือง จ.ขอนแก่น

ประวัติการศึกษา

- จบระดับมัธยมศึกษาจากโรงเรียนขอนแก่นวิทยายน
- ปัจจุบันกำลังศึกษาในระดับปริญญาตรีชั้นปีที่ 4 สาขาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

E-mail : **por_me24@hotmail.com**

ชื่อ นางสาวรุ่งรัตน์ เสือประดิษฐ์

ภูมิลำเนา 132/2 ม.3 ต.ในเมือง อ.เมือง จ.ลำพูน

ประวัติการศึกษา

- จบระดับมัธยมศึกษาจากโรงเรียนส่วนบุญโญปถัมภ์ ลำพูน
- ปัจจุบันกำลังศึกษาในระดับปริญญาตรีชั้นปีที่ 4 สาขาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

E-mail : **rungrat135@hotmail.com**

ชื่อ นางสาวอัปสรสินี เกี้ยวเกิด

ภูมิลำเนา 66/6 ม.9 อ.ลาดยาว จ.นครสวรรค์

ประวัติการศึกษา

- จบระดับมัธยมศึกษาจากโรงเรียนสตรีนครสวรรค์
- ปัจจุบันกำลังศึกษาในระดับปริญญาตรีชั้นปีที่ 4 สาขาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

E-mail : **apsornsinee@hotmail.com**