การเรียนรู้และจดจำลักษณะตัวอักษร

Character Recognition

โดย

นายสมชาย อิทธิพลกังวาล

รหัส 41067099

H001676

อาจารย์ที่ปรึกษา

ดร.อาริต ธรรมโน

วัน เดือน ปี 25 ธ.ค. 2543

เลขทะเบียน 01676

เลขเรียกหนังสือ วพ. ส141 ก 2543

"ห้องสมุดคณะเทคโนโลยีสารสนเทศ สจล."

รายงานนี้เป็นส่วนหนึ่งของวิชาโครงการพัฒนาระบบงาน

หลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ

ภาคเรียนที่ 1 ปีการศึกษา 2543

คณะเทคโนโลยีสารสนเทศ

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชื่อหัวข้อ	การเรียนรู้และจดจำลักษณะตัวอักษร
นักศึกษา	นายสมชาย อิทธิพลกังวาล
อาจารย์ที่ปรึกษา	ดร.อาริต ธรรมโน
ระดับการศึกษา	วิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ
แขนงวิชา	วิทยาการสารสนเทศ
ปีการศึกษา	2543

บทคัดย่อ

ปัจจุบันเครื่องคอมพิวเตอร์เข้ามามีบทบาทกับชีวิตประจำวันมากขึ้น มีการนำคอมพิวเตอร์เข้ามาประมวลผลหรือทำงานบางอย่างที่มีความสามารถที่คล้ายหรือใกล้เคียงกับมนุษย์ เช่น การนำคอมพิวเตอร์มาใช้ในการเรียนรู้และจดจำตัวอักษร โดยการใช้โครงข่ายปราสาทเทียม (Neural Network) ดังนั้น ในการศึกษาครั้งนี้ผู้ศึกษาจึงสนใจศึกษาโครงข่ายประสาทเทียมแบบแพร่ย้อนกลับ (Backpropagation Neural Network) และสร้างแบบจำลองเพื่อทดสอบความสามารถของแบบจำลองดังกล่าว

ผลการทดลองพบว่าแบบจำลองโครงข่ายประสาทเทียมแบบแพร่ย้อนกลับที่ดีที่สุด คือ มีจำนวนนิวรอนในชั้นข้อมูลเข้า (Input Layer) เท่ากับ 20 x 20 = 400 นิวรอน จำนวนนิวรอนในชั้นซ่อน (Hidden Layer) เท่ากับ 50 นิวรอน จะทำให้โครงข่ายมีความสามารถในการระบุตัวอักษรที่ใช้ในการเรียนรู้ได้ทั้งหมด และเมื่อทดสอบโดยใช้ตัวอักษร CordiaUPC ขนาด 26 ตัวอักษรต่อนิ้ว ตัวหนา และ ตัวอักษร AngsanaUPC ขนาด 26 ตัวอักษรต่อนิ้ว ในการเรียนรู้จำนวน 52 ตัวอักษร หลังจากการเรียนรู้จึงนำแบบจำลองไปทดสอบกับตัวอักษรที่ใช้ในการเรียนรู้แม้จะมีความผิดพลาด 10% ก็สามารถระบุได้ทั้งหมด และสามารถระบุตัวอักษรอื่น ๆ ที่ใช้ในการทดลองได้ 228 ตัวอักษร จาก 312 ตัวอักษร คิดเป็น 73.08%

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Title	Character Recognition
Student	Mr. Somchai Ittipronkungwan
Advisor	Dr. Arit Thammano
Level of Study	Master of Science in Information Technology
Major	Information Science
Academic Year	2000

ABSTRACT

Nowsaday, people use computer in many activities. Even though some task that working likes human process such as learning and recognize character by using Neural Network. To test the process of learning and recognize character, I used 'Backpropagation Neural Network' as a testing model.

After experiment of learning 52 characters of 26 points/inch CordiaUPC and 26 points/inch AngsanaUPC, I found out that the best model which have capable to learning most characters should have 20 x 20 = 400 neural of input layer and 50 neural hidden layer. After testing this model, the result was shown that this model can identify 218 of 312 characters which is 78.08 %.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ข้าพเจ้าขอกราบขอบพระคุณ ดร. อาริต ธรรมโน อาจารย์ที่ปรึกษา ที่ได้กรุณาสละเวลาอันมีค่า ดูแลเอาใจใส่ ชี้แนะ สอน และให้คำแนะนำต่าง ๆ ทั้งในด้านแนวความคิดและวิชาการ จนกระทั่งโครงการศึกษาพัฒนาระบบงานฉบับนี้สำเร็จลุล่วงด้วยดี และขอขอบคุณ คุณกิตติชัย ชีวาสุขถาวร ที่กรุณาช่วยพัฒนาโปรแกรมที่เป็นประโยชน์ต่อการศึกษาและพัฒนาแบบจำลอง ตลอดจน เพื่อน ๆ และ พี่ ๆ IS6 สมทบทุกท่านที่ให้คำแนะนำทางด้านเทคนิค กำลังใจตลอดมา

สุดท้ายนี้ โครงการศึกษาพัฒนาระบบงานนี้จะไม่อาจสำเร็จลุล่วงได้ หากขาดบุคคลที่มีพระคุณสูงสุดสองท่านคือ คุณพ่อ และคุณแม่ ที่เป็นผู้เฝ้ารอด้วยความอดทน และสนับสนุนให้กำลังใจในทุก ๆ ด้าน จนกระทั่งถึงวันนี้

สมชาย อิทธิพลกังวาล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

หน้า

บทคัดย่อภาษาไทย I

บทคัดย่อภาษาอังกฤษ II

กิตติกรรมประกาศ III

สารบัญ IV

สารบัญตาราง VI

บทที่ 1 บทนำ 1

1.1 ความเป็นมา 1

1.2 วัตถุประสงค์ในการศึกษา 1

1.3 แนวทางการศึกษา 2

1.4 ผลที่คาดว่าจะได้รับ 2

บทที่ 2 ทฤษฎีและผลงานวิจัยที่เกี่ยวข้อง 3

2.1 บทนำ 3

2.2 โครงข่ายประสาทเทียม (Neural Network) 4

2.3 แบบจำลองของโครงข่ายประสาทเทียม 5

2.4 อัลกอริธึมการเรียนรู้ของโครงข่าย 7

2.5 โครงข่ายประสาทเทียมแบบแพร่ย้อนกลับ (Backpropagation Network) 8

2.6 ผลงานวิจัยที่เกี่ยวข้อง 9

บทที่ 3 วิธีการศึกษา 12

3.1 สถาปัตยกรรมที่ใช้ในการทดลอง 12

3.2 วิธีการสร้างแบบจำลอง 12

3.3 ขั้นตอนการเตรียมข้อมูล 13

3.4 ขั้นตอนการทำงานของแบบจำลอง 14

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้า

บทที่ 4 ผลการทดลอง 17
4.1 ตัวอักษรที่ใช้ในการทดลอง 17
4.2 ผลการทดลอง 18

บทที่ 5 สรุปผลการทดลอง 55
5.1 สรุปผลการทดลอง 55
5.2 ปัญหาและข้อจำกัดของการทำงานของแบบจำลอง 59
5.2 ข้อเสนอแนะ 60

บรรณานุกรม 61
ภาคผนวก 63
แบบตัวอักษรที่ใช้ในการทดลอง 63
โปรแกรมแบบจำลองโครงข่ายสำหรับการเรียนรู้ (Learning) 67
โปรแกรมสำหรับการทดสอบ (Testing) แบบจำลองโครงข่าย 76
ประวัติผู้เขียน 82

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

หน้า

ตารางที่ 4.1 แสดงผลการทดลองกับตัวอักษร CordiaUPC ขนาด 16 จุดต่อนิ้ว ตัวหนา 5 ตัวอักษร ด้วยจำนวนนิวรอนในชั้นซ่อน 200 นิวรอน 18
ตารางที่ 4.2 แสดงผลการทดลองกับตัวอักษร CordiaUPC ขนาด 16 จุดต่อนิ้ว ตัวหนา 26 ตัวอักษร ด้วยจำนวนนิวรอนในชั้นซ่อน 200 นิวรอน 19
ตารางที่ 4.3 แสดงผลการทดลองกับตัวอักษร CordiaUPC ขนาด 16 จุดต่อนิ้ว ตัวหนา ด้วยจำนวนนิวรอนในชั้นซ่อน 150 นิวรอน 20
ตารางที่ 4.4 แสดงผลการทดลองกับตัวอักษร CordiaUPC ขนาด 16 จุดต่อนิ้ว ตัวหนา ขวยจำนวนนิวรอนในชั้นซ่อน 100 นิวรอน 21
ตารางที่ 4.5 แสดงผลการทดลองกับตัวอักษร CordiaUPC ขนาด 16 จุดต่อนิ้ว ตัวหนา ด้วยจำนวนนิวรอนในชั้นซ่อน 50 นิวรอน 22
ตารางที่ 4.6 แสดงผลการทดลองกับตัวอักษร CordiaUPC ขนาด 16 จุดต่อนิ้ว ตัวหนา ด้วยจำนวนนิวรอนในชั้นซ่อน 250 นิวรอน 23
ตารางที่ 4.7 แสดงผลการทดลองกับตัวอักษร CordiaUPC ขนาด 16 จุดต่อนิ้ว ตัวหนา ด้วยจำนวนนิวรอนในชั้นซ่อน 50 นิวรอน โดยมีการเปลี่ยนค่าอัตราการเรียนรู้ 24
ตารางที่ 4.8 แสดงผลการทดลองกับตัวอักษร CordiaUPC ขนาด 16 จุดต่อนิ้ว ตัวหนา ด้วยจำนวนนิวรอนในชั้นซ่อน 100 นิวรอน โดยมีการเปลี่ยนค่าอัตราการเรียนรู้ 25
ตารางที่ 4.9 แสดงผลการทดลองแบบจำลองโครงข่ายซึ่งมีจำนวนนิวรอนในชั้นข้อมูลเข้าจำนวน 144 นิวรอน และจำนวนนิวรอนในชั้นซ่อน จำนวน 50 นิวรอน ซึ่งมีผลการทดลองดังนี้ 27
ตารางที่ 4.10 แสดงผลการทดลองแบบจำลองโครงข่ายซึ่งมีจำนวนนิวรอนในชั้นข้อมูลเข้าจำนวน 144 นิวรอน และจำนวนนิวรอนในชั้นซ่อน จำนวน 100 นิวรอน ซึ่งมีผลการทดลองดังนี้ 29

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง (ต่อ)

หน้า

ตารางที่ 4.11 แสดงผลการทดลองกับตัวอักษร CordiaUPC ขนาด 26 จุดต่อนิ้ว ตัวหนา ด้วยจำนวนนิวรอนในชั้นซ่อน 50 นิวรอน โดยมีการเปลี่ยนค่าอัตราการเรียนรู้ 31
ตารางที่ 4.12 แสดงผลการทดลองกับตัวอักษร CordiaUPC ขนาด 26 จุดต่อนิ้ว ตัวหนา ด้วยจำนวนนิวรอนในชั้นซ่อน 100 นิวรอน โดยมีการเปลี่ยนค่าอัตราการเรียนรู้ 32
ตารางที่ 4.13 แสดงผลการทดลองกับตัวอักษร CordiaUPC ขนาด 26 จุดต่อนิ้ว ตัวหนา ด้วยจำนวนนิวรอนในชั้นซ่อน 150 นิวรอน โดยมีการเปลี่ยนค่าอัตราการเรียนรู้ 33
ตารางที่ 4.14 แสดงผลการทดลองกับตัวอักษร CordiaUPC ขนาด 26 จุดต่อนิ้ว ตัวหนา ด้วยจำนวนนิวรอนในชั้นซ่อน 250 นิวรอน โดยมีการเปลี่ยนค่าอัตราการเรียนรู้ 34
ตารางที่ 4.15 แสดงผลการทดลองแบบจำลองโครงข่ายซึ่งมีจำนวนนิวรอนในชั้นข้อมูลเข้าจำนวน 400 นิวรอน และจำนวนนิวรอนในชั้นซ่อน จำนวน 50 นิวรอน ซึ่งมีผลการทดลองดังนี้ 35
ตารางที่ 4.16 แสดงผลการทดลองแบบจำลองโครงข่ายซึ่งมีจำนวนนิวรอนในชั้นข้อมูลเข้าจำนวน 400 นิวรอน และจำนวนนิวรอนในชั้นซ่อน จำนวน 100 นิวรอน ซึ่งมีผลการทดลองดังนี้ 37
ตารางที่ 4.17 แสดงผลการทดลองแบบจำลองโครงข่ายซึ่งมีจำนวนนิวรอนในชั้นข้อมูลเข้าจำนวน 400 นิวรอน และจำนวนนิวรอนในชั้นซ่อน จำนวน 150 นิวรอน ซึ่งมีผลการทดลองดังนี้ 39
ตารางที่ 4.18 แสดงผลการทดลองแบบจำลองโครงข่ายซึ่งมีจำนวนนิวรอนในชั้นข้อมูลเข้าจำนวน 400 นิวรอน และจำนวนนิวรอนในชั้นซ่อน จำนวน 200 นิวรอน ซึ่งมีผลการทดลองดังนี้ 41

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง (ต่อ)

หน้า

ตารางที่ 4.19 แสดงผลการทดลองแบบจำลองโครงข่ายซึ่งมีจำนวนนิวรอนในชั้นข้อมูลเข้าจำนวน 400 นิวรอน และจำนวนนิวรอนในชั้นซ่อน จำนวน 250 นิวรอน ซึ่งมีผลการทดลองดังนี้ 43
ตารางที่ 4.20 แสดงผลการทดลองกับตัวอักษร CordiaUPC ขนาด 26 จุดต่อนิ้ว ตัวหนา และ AngsanaUPC ขนาด 26 จุดต่อนิ้ว ด้วยจำนวนนิวรอนในชั้นซ่อน 50 นิวรอน โดยมีการเปลี่ยนค่าอัตราการเรียนรู้ 46
ตารางที่ 4.21 แสดงผลการทดลองกับตัวอักษร CordiaUPC ขนาด 26 จุดต่อนิ้ว ตัวหนา และ AngsanaUPC ขนาด 26 จุดต่อนิ้ว ด้วยจำนวนนิวรอนในชั้นซ่อน 100 นิวรอน โดยมีการเปลี่ยนค่าอัตราการเรียนรู้ 47
ตารางที่ 4.22 แสดงผลการทดลองกับตัวอักษร CordiaUPC ขนาด 26 จุดต่อนิ้ว ตัวหนา และ AngsanaUPC ขนาด 26 จุดต่อนิ้ว ด้วยจำนวนนิวรอนในชั้นซ่อน 150 นิวรอน โดยมีการเปลี่ยนค่าอัตราการเรียนรู้ 48
ตารางที่ 4.23 แสดงผลการทดลองแบบจำลองโครงข่ายซึ่งมีจำนวนนิวรอนในชั้นข้อมูลเข้าจำนวน 400 นิวรอน และจำนวนนิวรอนในชั้นซ่อน จำนวน 50 นิวรอน กับตัวอักษร Cordia และ AngsanaUPC ซึ่งมีผลการทดลองดังนี้ 49
ตารางที่ 4.24 แสดงผลการทดลองแบบจำลองโครงข่ายซึ่งมีจำนวนนิวรอนในชั้นข้อมูลเข้าจำนวน 400 นิวรอน และจำนวนนิวรอนในชั้นซ่อน จำนวน 100 นิวรอน กับตัวอักษร Cordia และ AngsanaUPC ซึ่งมีผลการทดลองดังนี้ 51
ตารางที่ 4.25 แสดงผลการทดลองแบบจำลองโครงข่ายซึ่งมีจำนวนนิวรอนในชั้นข้อมูลเข้าจำนวน 400 นิวรอน และจำนวนนิวรอนในชั้นซ่อน จำนวน 150 นิวรอน กับตัวอักษร Cordia และ AngsanaUPC ซึ่งมีผลการทดลองดังนี้ 53
ตารางที่ 5.1 แสดงผลการทดลองการเปลี่ยนแปลงอัตราการเรียนรู้ 55
ตารางที่ 5.2 ตารางสรุปผลการทดสอบแบบจำลองโครงข่ายประสาทเทียมที่ใช้ข้อมูลในชั้นข้อมูลเข้าจำนวน 144 นิวรอน 56

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง (ต่อ)

หน้า

ตารางที่ 5.3 ตารางสรุปผลการทดสอบแบบจำลองโครงข่ายประสาทเทียมที่ใช้ข้อมูลใน
ชั้นข้อมูลเข้าจำนวน 400 นิวรอน โดยใช้ตัวอักษร CordiaUPC ขนาด 26
จุดต่อนิ้ว ตัวหนาในการเรียนรู้ 57
ตารางที่ 5.4 ตารางสรุปผลการทดสอบแบบจำลองโครงข่ายประสาทเทียมที่ใช้ข้อมูลใน
ชั้นข้อมูลเข้าจำนวน 400 นิวรอน โดยใช้ตัวอักษร CordiaUPC ขนาด 26
จุดต่อนิ้ว ตัวหนา และ AngsanaUPC ขนาด 26 จุดต่อนิ้ว ในการเรียนรู้ 58

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ความเป็นมา

ปัจจุบันคอมพิวเตอร์เข้ามามีบทบาทที่สำคัญในชีวิตประจำวันมากขึ้นเรื่อย ๆ มีการพัฒนาเทคโนโลยีนี้มาโดยตลอดไม่ว่าจะเป็นทางด้านฮาร์ดแวร์ และซอฟต์แวร์ ตลอดระยะเวลาที่ผ่านมามีการนำคอมพิวเตอร์มาใช้ในการเก็บบันทึกข้อมูล จัดพิมพ์เอกสาร ช่วยการทำงานทางด้านการออกแบบ ควบคุมการทำงานของอุปกรณ์ต่าง ๆ เป็นต้น เพื่อช่วยอำนวยความสะดวกในการทำงาน

หลังจากที่คอมพิวเตอร์สามารถอำนวยความสะดวกในการทำงานได้แล้ว ได้มีนักวิจัยหลายท่านได้ทำการศึกษาและวิจัยมีความเห็นว่าคอมพิวเตอร์น่าจะมีความสามารถทางด้านอื่นๆ ได้อีกนอกจากที่กล่าวมาข้างต้น โดยให้คอมพิวเตอร์มีความสามารถทางด้านปัญญาประดิษฐ์ (Artificial Intelligent) เช่น การอ่านหนังสือ การจัดหมวดหมู่หนังสือ การเรียนรู้และจดจำลักษณะหรือรูปแบบต่าง ๆ ไม่ว่าจะเป็นการรู้จำใบหน้า ลายนิ้วมือ ตัวอักษรพิมพ์ และตัวอักษรลายมือเขียน โดยการใช้โครงข่ายประสาทเทียม (Neural Network) ในการแยกแยะลักษณะตัวอักษร ได้มีการเรียนรู้และสร้างทฤษฎีการรู้จำตัวอักษรต่าง ๆ มากมาย ซึ่งในแต่ละวิธีก็จะมีความเหมาะสมกับปัญหาที่เกิดขึ้นเฉพาะด้าน

การศึกษาครั้งนี้เป็นการศึกษาถึงสถาปัตยกรรมโครงข่ายประสาทเทียม (Neural Network) ต่าง ๆ ตลอดจนผลงานวิจัยที่เกี่ยวข้อง เพื่อนำมาใช้ในการพัฒนาสร้างแบบจำลองในการเรียนรู้และจดจำลักษณะตัวอักษร และทำการทดลอง เพื่อทราบถึงผลและปัญหาที่เกิดขึ้นจากการทดลอง

1.2 วัตถุประสงค์ในการศึกษา

1. เพื่อศึกษาวิธีการและรูปแบบต่าง ๆ ที่โครงข่ายประสาทเทียม (Neural Network) ใช้ในการเรียนรู้
2. เพื่อทราบถึงโครงข่ายที่เหมาะสมในการรู้จำ และระบุตัวอักษร (Character Recognition)
3. เพื่อเป็นแนวทางในการศึกษาต่อไปในอนาคต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.3 แนวทางการศึกษา

ผู้ศึกษาจะทำการศึกษารูปแบบการทำงานของโครงข่ายประสาทเทียม (Neural Network) ต่าง ๆ ที่สามารถนำมาใช้ในการเรียนรู้และจดจำลักษณะตัวอักษร หลังจากนั้นจะทำการจำลองโครงข่ายประสาทเทียมขึ้น แล้วนำโครงข่ายที่ได้มาทดลองให้เรียนรู้ลักษณะของตัวอักษร (Learning) ทำการบันทึกผลที่ได้จากการทดลอง ค่าความผิดพลาด เพื่อนำไปใช้ในการปรับปรุง เปลี่ยนแปลงโครงข่ายให้สามารถเรียนรู้และจดจำลักษณะตัวอักษรได้ดีขึ้น หลังจากนั้นจึงทำการสรุปผลที่ได้ว่าเป็นอย่างไร ปัญหาที่เกิดมีอะไรบ้าง และมีข้อเสนอแนะประการใดต่อไป

1.4 ผลที่คาดว่าจะได้รับ

1. ทำให้ทราบถึงวิธีการทำงานของโครงข่ายประสาทเทียม (Neural Network) ในการเรียนรู้ และจดจำลักษณะตัวอักษร (Character Recognition)

2. ทำให้ทราบถึงปัญหาต่าง ๆ และวิธีการแก้ไข ที่เกิดขึ้นในการทำรูปแบบจำลองโครงข่ายประสาทเทียม (Neural Network) ที่ใช้ในการทดลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎีและผลงานวิจัยที่เกี่ยวข้อง

2.1 บทนำ

สมองของมนุษย์นั้นประกอบด้วยเซลประสาท หรือนิวรอน ประมาณ 10^{11} ตัว โดยแต่ละเซลประสาทจะมีเส้นเชื่อมโยงไปยังประสาทอื่น ๆ ประมาณ 10^{4} เส้น และเวลาในการสับเปลี่ยนข้อมูลระหว่างนิวรอนอยู่ในระดับของ 10^{-3} วินาที ซึ่งนับว่าช้ามากเมื่อเทียบกับเวลาในการ Switch ของคอมพิวเตอร์เพียง 10^{-10} วินาทีเท่านั้น อย่างไรก็ตามสมองมนุษย์สามารถประมวลผลได้รวดเร็วมาก เช่น คนเราสามารถจำหน้าแม่ของตนได้ภายในเวลาเพียง 10^{-1} วินาที ซึ่งน่าจะต้องผ่านการประมวลผลเป็นร้อย ๆ ขั้นตอน ด้วยความสามารถของสมองมนุษย์ที่มีขั้นตอนการประมวลผลที่สลับซับซ้อน และการที่มนุษย์สามารถจดจำสิ่งต่าง ๆ ได้ เกิดจากการเรียนรู้ในสิ่งนั้น ๆ ก่อน ไม่เช่นนั้นสมองก็ไม่สามารถจดจำอะไรได้เลย แม้ปัจจุบันจะยังไม่สามารถสรุปขั้นตอนการเรียนรู้ของสมองได้ชัดเจนนัก ก็อาจสรุปได้คร่าว ๆ ว่า เซลประสาทในสมองต่างประมวลผลเพื่อวิเคราะห์และจดจำลักษณะเฉพาะต่าง ๆ เช่น การจำว่าเป็นเสียงเด็ก เสียงผู้ใหญ่ เสียงผู้หญิง เสียงผู้ชาย และเป็นเสียงของใคร และก็อาจลืมสิ่งต่าง ๆ ที่เคยจดจำไว้ด้วย เนื่องจากสมองมีภาระกิจมากมายในการประมวลผลสิ่งต่าง ๆ ในแต่ละวัน แต่ก็ยังสามารถที่จะเรียนรู้สิ่งใหม่ ๆ ได้ ดังนั้นจึงมีนักวิชาการทางด้านคอมพิวเตอร์ได้เสนอรูปแบบและวิธีการจำลองการทำงานเพื่อให้คอมพิวเตอร์สามารถแก้ปัญหาที่ซับซ้อนได้เช่นเดียวกับสมองมนุษย์มากที่สุด และเรียกแบบจำลองนี้ว่าโครงข่ายประสาทเทียม (Neural Network) ซึ่งในอดีตที่ผ่านมาการศึกษาทางด้านโครงข่ายประสาทเทียม (Neural Network) สามารถแบ่งกลุ่มของนักศึกษาวิจัยได้ 2 กลุ่ม คือ

- ศึกษาและจำลองกระบวนการประมวลผลของสมองสิ่งมีชีวิต
- พัฒนากระบวนการ Machine Learning โดยไม่สนใจว่ากระบวนการเหล่านั้นจะเลียนแบบหรือคล้ายกับการทำงานของสิ่งมีชีวิตหรือไม่

ในการที่จะนำโครงข่ายประสาทเทียม (Neural Network) มาประยุกต์ใช้งานนั้น จะต้องมีการเรียนรู้จากตัวอย่างเพื่อจะได้มีความรู้ที่จะนำไปแก้ปัญหาต่อไป ซึ่งสามารถแบ่งได้เป็น 2 ขั้นF ตอน ดังนี้

1. ขั้นตอนการฝึกหัด (Training) หรือการเรียนรู้ (Learning) สามารถแบ่งได้ 2 วิธี คือ (อัญชลี วานิชทวีวัฒน์: 2540)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.1 การเรียนรู้แบบมีครูสอน (Supervised Learning Algorithm) การสอนวิธีนี้จะเป็นการกำหนดเซตการสอนให้กับโครงข่าย ซึ่งประกอบด้วยข้อมูลเข้า (Input) และแบบข้อมูลที่ต้องการ (Desired Output) ที่ต้องการ เมื่อป้อน Input เข้าสู่โครงข่าย ๆ ก็จะทำการประมวลผลจะได้คำตอบและค่าถ่วงน้ำหนัก แล้วนำคำตอบที่ได้มาคำนวณหาความผิดพลาด (Error) เพื่อนำมาปรับค่าถ่วงน้ำหนักให้ได้ค่าข้อมูลของออกตรงตามค่าข้อมูลที่ต้องการ และจะทำการสอนจนกว่าค่าความผิดพลาดที่ได้จากการคำนวณน้อยกว่าค่าความผิดพลาดที่ยอมรับได้ก็จะหยุดการสอน และค่าถ่วงน้ำหนักที่ได้จากการสอนจะเปรียบเสมือนฟังก์ชั่นที่ใช้ในการแปลงข้อมูล ตัวอย่างแบบจำลองนี้เช่น Backpropagation

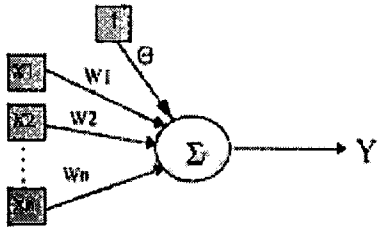
1.2 การเรียนรู้แบบไม่มีครูสอน (Unsupervised Learning Algorithm) เป็นการป้อน Input เข้าสู่โครงข่ายเพียงอย่างเดียวไม่มีการส่งค่าใด ๆ ให้คำนวณย้อนกลับ การเรียนรู้ในลักษณะนี้เป็นลักษณะของการแบ่งกลุ่ม (Categorization) ตามชุดตัวอย่างแบบข้อมูลเข้าทั้งหมดที่ใช้ในการเรียนรู้ โดยโครงข่ายจะมี Output อยู่ด้วยกันหลายนิวรอน แต่ละนิวรอนจะแทนกลุ่มข้อมูลที่มีคุณสมบัติที่เหมือนกัน เมื่อป้อน Input เข้าสู่โครงข่าย ๆ จะทำการคำนวณค่าความสัมพันธ์ที่อยู่ภายในเซตของ Input โดยอาศัยค่าถ่วงน้ำหนักเป็นตัวแยกความแตกต่างของ Input ไปยังนิวรอน Output ของโครงข่าย การสอนลักษณะนี้ไม่สามารถระบุได้ว่า Output นิวรอนใดเป็นข้อมูลของกลุ่มไหน ผู้ใช้ต้องกำหนดเอง ตัวอย่างแบบจำลองนี้เช่น Counterpropropagation (CPN) และ Self-Organizing Map (SOM)

2. ขั้นตอนการทดสอบ (Testing) หรือใช้งานจริง เป็นการนำโครงข่ายที่ได้หลังจากการฝึกหัดหรือเรียนรู้ไปใช้งานจริง

2.2 โครงข่ายประสาทเทียม (Neural Network)

โครงข่ายประสาทเทียม (Neural Network) คือการจำลองรูปแบบทางชีววิทยาทางโครงสร้างสมองของมนุษย์ ซึ่งเป็นการจำลองเอาบางส่วนที่จำเป็นมาพร้อมผสมกับรูปแบบการจัดการโครงสร้างที่ถูกคิดค้นขึ้น เพื่อทำให้เกิดฟังก์ชั่นที่ต้องการ โดยจะแทนเซลประสาทด้วย ส่วนรับข้อมูลเข้า (Input) ส่วนประมวลผลนิวรอน ในการรับค่า Input ต่าง ๆ ซึ่งก็คือค่าน้ำหนัก (Weight) หลังจากนั้นจึงนำไปประมวลผล เพื่อส่ง Output ออกจากนิวรอน โดยแบบจำลองระบบแรกถูกนำเสนอโดย McCulloch และ Pitt ในปี ค.ศ. 1943 (อ้างใน วาภิรมย์ มนัสรังสี: 2540) ซึ่งเป็นแบบจำลองของเซลล์ประสาทดังรูปที่ 2.1 อินพุต x_I (สำหรับ $i = 1,2,\ldots,n$) จะมีค่าเป็น {0,1} ซึ่งจะขึ้นอยู่กับสัญญาณอินพุตจากเซลล์อื่นในขณะนั้นว่าจะมีหรือไม่มีสัญญาณ ส่วนสัญญาณที่ส่งต่อไปยังเซลล์ถัดไปซึ่งเซลล์ผลลัพธ์ (จะแทนด้วย o) และ Firing Level ของแบบจำลองนี้ถูกกำหนดโดย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.1 แสดงนิวรอนของโครงข่ายประสาทเทียม

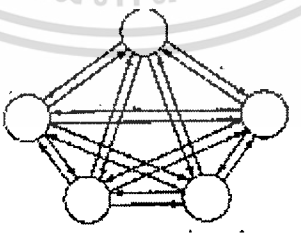
$$O^{k+1} = \begin{cases} 1 & \text{if } \sum W_i x_i^k \geq T \\ 0 & \text{otherwise} \end{cases}$$

โดยที่ k = 0,1,2,... เป็นช่วงเวลาแบบไม่ต่อเนื่อง w_i เป็นค่าถ่วงน้ำหนัก ที่เชื่อมต่อกับอินพุตที่ i ซึ่งถ้า $w_i = 1$ แสดงถึงการกระตุ้น และถ้า $w_i = -1$ จะมีการยับยั้งการส่งผ่านสัญญาณ และ T เป็นค่าความต่างศักดิ์เทรสโฮลด์หรือขีดเริ่มเปลี่ยนซึ่งถ้าค่าผลรวมของผลคูณระหว่าค่าถ่วงน้ำหนักกับสัญญาณอินพุตจะต้องมากกว่า T จึงจะมีสัญญาณส่งผ่านไปยังเซลล์อื่นได้

2.3 แบบจำลองของโครงข่ายประสาทเทียม

แบบจำลองที่ใช้ในการเชื่อมต่อระหว่างนิวรอนในโครงข่าย สามารถแบ่งได้ 3 ลักษณะ คือ (David M.Skapura., 1996)

1. โครงข่ายแบบเชื่อมต่อกันหมด (Totally Connected Network) เป็นโครงข่ายที่ทุกนิวรอนเชื่อมต่อถึงกันหมด ตามรูปที่ 2.2



รูปที่ 2.2 แสดงโครงข่ายแบบเชื่อมต่อกันหมด

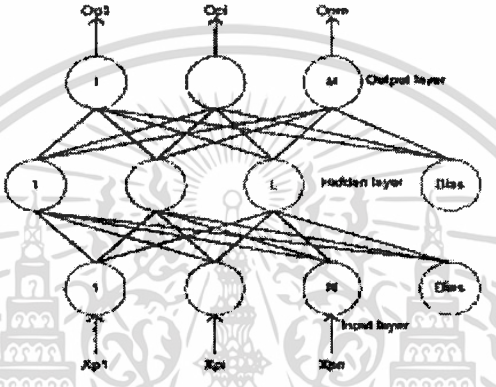
2. โครงข่ายแบบลำดับชั้น (Layer Network) โดยโครงข่ายนี้สามารถแบ่งออกเป็นระดับชั้นต่าง ๆ ออกเป็น 3 ระดับด้วยกันคือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ระดับชั้นข้อมูลเข้า (Input Layer) จะเป็นส่วนที่รับข้อมูลเข้าสู่โครงข่าย เพื่อส่งให้กับชั้นซ่อนหรือชั้นข้อมูลออกต่อไป

- ระดับชั้นซ่อน (Hidden Layer) เป็นส่วนที่ใช้ในการประมวลผล โดยรับข้อมูล (Weight) มาจากชั้นข้อมูลเข้าหรือชั้นซ่อนด้วยกันมาประมวลผล แล้วส่งข้อมูลที่ได้จากการประมวลผลสู่ชั้นอื่น ซึ่งอาจเป็นชั้นซ่อนหรือเป็นชั้นข้อมูลออกก็ได้

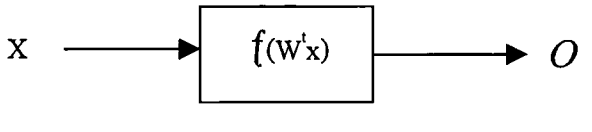
- ระดับชั้นข้อมูลออก (Output Layer) เป็นชั้นที่ส่งผลลัพธ์ที่ได้จากโครงข่าย ดังรูปที่ 2.3



รูปที่ 2.3 แสดงระดับชั้นของโครงข่ายประสาทเทียม

ลักษณะการส่งผ่านข้อมูลภายในลำดับชั้น สามารถแบ่งได้เป็น 2 วิธี คือ (วาภิรมย์ มนัสรังสี . 2540:49)

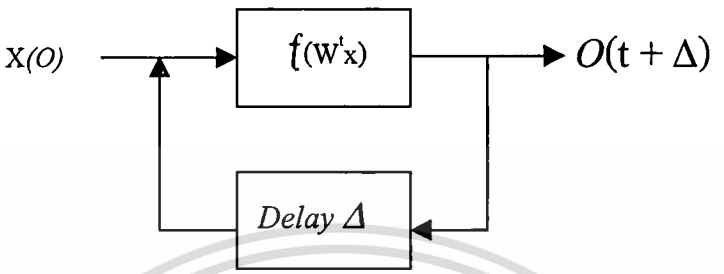
- โครงข่ายที่ส่งสัญญาณไปข้างหน้า (Feedforword Networks) จะประกอบด้วยชั้นต่าง ๆ ของโครงข่าย โดยชั้นแรกจะเป็นชั้น Input และ ชั้นสุดท้ายจะเป็นชั้น Output ส่วนระหว่างชั้น Input และ Output จะมีชั้นซ่อนหรือไม่ก็ได้ ขึ้นอยู่กับอัลกอริทึมที่ใช้สอนโครงข่าย เช่น โครงข่ายแบบ Multilayer Perceptron จะมีชั้นซ่อนมากกว่า 1 ชั้นก็ได้ หรือ โครงข่าย Self-Organizing Map ของ Kohonen จะมีเพียงชั้น Input และ Output เท่านั้น การเชื่อมต่อระหว่างชั้นของโครงข่ายจะมีค่าน้ำหนักเป็นตัวเชื่อม และสัญญาณที่ป้อนเข้ามาจะถูกส่งไปตามทิศทางของลูกศรจนถึงชั้นของ Output โดยไม่มีการป้อนกลับ ดังรูปที่ 2.4



รูปที่ 2.4 แสดงบล็อคไดอะแกรมของโครงข่าย Feedforward

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- โครงข่ายที่มีการป้อนกลับ (Feedback Network) จะมีลักษณะคล้ายกับโครงข่ายแบบ Feedforword Networks แต่จะมีส่วนที่เพิ่มเข้ามาคือส่วนของการป้อนกลับดังรูปที่ 2.5 และการป้อนกลับจะมีการหน่วงเวลาไปจากเดิม



รูปที่ 2.5 แสดงบล็อคไดอะแกรมของโครงข่าย Feedback

3. โครงข่ายแบบผสม (Network of Mixed Connectivity) เป็นแบบผสมระหว่างแบบที่ 1 และ 2 แสดงให้เห็นดังรูปที่ 2.6

รูปที่ 2.6 แสดงโครงข่ายแบบผสม

2.4 อัลกอริธึมการเรียนรู้ของโครงข่าย

ในการเรียนรู้ของโครงข่ายจะมีขั้นตอนของการเรียนรู้หรือฝึกฝนโครงข่าย โดยในการเรียนรู้นั้นจะใช้ฟังก์ชั่นทางคณิตศาสตร์ โดยมีรูปแบบโครงข่ายประสาทเทียมตามรูปที่ 2.1 ซึ่งมีขั้นตอนดังนี้

1. ตั้งค่าเริ่มต้นของน้ำหนัก (Weight) สำหรับแต่ละชุดของตัวอย่าง โดยใช้วิธีการสุ่มตัวเลขให้มีค่าน้อย ๆ มักอยู่ระหว่าง –1 ถึง 1

2. สำหรับแต่ละชุดตัวอย่าง [X1,X2,X3,….Xn]: [t] โดย [X1,X2,X3,….Xn] เป็น Input Vector และ [t] คือเป้าหมาย (Target หรือ Desired Output) ให้ทำ

2.1 ส่งตัวอย่างเข้าทาง Input Layer

2.2 คำนวณค่า Output โดยคำนวณค่าของ $\sum_{t=1}^{n} W_t X_t - \theta$ แล้วเปรียบเทียบกับ t (Target)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3 ถ้า Output ที่ได้ไม่เท่ากับ t ให้ทำการปรับค่าน้ำหนัก (Weight) ตามแต่กฎการเรียนรู้จะกำหนด

3. ถ้าโครงข่ายให้ค่า Output ตรงกับเป้าหมาย (Target) สำหรับทุกชุดของตัวอย่าง จะเป็นการสิ้นสุดการเรียนรู้ มิฉะนั้นให้เริ่มกลับไปทำข้อ 2 อีกครั้ง

2.5 โครงข่ายประสาทเทียมแบบแพร่ย้อนกลับ (Backpropagation Network)

โครงข่ายแบบแพร่ย้อนกลับ เป็นแบบจำลองโครงข่ายที่มีการเชื่อมโยงกันเป็นโครงข่ายแบบเป็นลำดับชั้น โดยประกอบด้วย Input Layer, Hidden Layer ซึ่งอาจมากกว่า 1 Layer ก็ได้ และ Output Layer เป็นวิธีการเรียนรู้แบบมีครูสอน (Supervised Learning Algorithm) และใช้หลักโครงข่ายที่มีการป้อนกลับ (Feedback Network) ในการเรียนรู้เพื่อปรับข้อผิดพลาดที่ได้จากการคำนวณหาค่า Output

อัลกอริธึมการเรียนรู้ของโครงข่ายแบบแพร่ย้อนกลับ (Backpropagation Network) มีดังนี้ (เฉพาะในกรณีที่มีชั้นซ่อนชั้นเดียว)

ตัวแปรต่าง ๆ ที่ใช้ในการอ้างอิง

X_i = Input บิต ที่ i สมมติว่ามี Input ทั้งหมด n บิต

Y_j = ค่ากระตุ้น (Activation) ของบิตที่ j ของชั้นซ่อน สมมติว่ามีทั้งหมด m บิต

O_k = Output บิตที่ k สมมติว่ามีทั้งหมด p บิต

α = อัตราการเรียนรู้ มีค่าในช่วง (0,1]

T_k = Target หรือ Desired Output บิตที่ k สมมติว่ามีทั้งหมด p บิต

ขั้นตอนการเรียนรู้มีดังนี้

1. สร้างโครงข่ายประสาทเทียมที่มีจำนวน Input และ Output เท่ากับจำนวน Input บิต และ Output บิต ส่วนจำนวนของนิวรอนในชั้นซ่อนนั้นแล้วแต่กำหนด จะเป็นเท่าไรก็ได้ ซึ่งควรจะมากพอสมควร

2. กำหนดค่าเริ่มต้นแก่ค่าถ่วงน้ำหนัก (Weight) ทุกเส้นของการเชื่อมต่อในโครงข่ายประสาทเทียม ซึ่งจะเป็นค่าตัวเลขสุ่มที่อยู่ในช่วง –1 ถึง 1

3. คำนวณค่า $\sum_{i=1}^{i=n} X_i W_{ij} = U_j$ สำหรับแต่ละนิวรอน j ของชั้นซ่อน

4. คำนวณค่ากระตุ้น (Activation) สำหรับแต่ละนิวรอน j ของชั้นซ่อน: $Y_j = f(U_j)$ โดยฟังก์ชันที่นิยมใช้คือ $f(x) = \frac{1}{1+e^{-x}}$ (Logistic Function)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. คำนวณค่า $\sum_{i=1}^{i=m} Y_j W_{jk} = V_k$ สำหรับแต่ละนิวรอน k ของชั้น Output

6. คำนวณค่ากระตุ้น (Activation) สำหรับแต่ละนิวรอน j ของชั้น Output : $O_k = f(V_t)$

7. นำค่า O_k มาเปรียบเทียบกับ Target (หรือ Desired Output) และคำนวณหาค่าผิดพลาด (Error) ที่เกิดขึ้น ถ้าค่าความผิดพลาดที่ได้น้อยกว่าระดับที่ตั้งไว้ ก็จะสิ้นสุดการเรียนรู้ มิฉะนั้นทำข้อ 8 ต่อ

8. คำนวณค่าของการปรับค่าถ่วงน้ำหนัก (Weight) สำหรับแต่ละค่าดังนี้

8.1 สำหรับเส้นเชื่อมระหว่างชั้นซ่อนกับชั้น Output

$$\Delta W_{ik} = \alpha \delta_k Y_j \quad \text{โดยที่ } \delta_k = (T_k - O_k) f'(V_k)$$

$$\text{ซึ่ง } f'(V_k) = O_k(1 - O_k)$$

8.2 สำหรับเส้นเชื่อมระหว่างชั้น Input กับชั้นซ่อน

$$\Delta W_{ij} = \alpha \delta_j X_i \quad \text{โดยที่ } \delta_j = \sum \delta_k W_{jk} f'(U_j)$$

9. ปรับค่าถ่วงน้ำหนักของเส้นเชื่อมจากนิวรอน r ไปยังนิวรอน s ด้วยสมการต่อไปนี้

$$W_{rs}\,(new) = W_{rs}\,(old) + \Delta W_{rs}$$

10. กลับไปทำข้อ 3 อีกครั้ง

2.6 ผลงานวิจัยที่เกี่ยวข้อง

ผลงานวิจัยด้านการวิเคราะห์ตัวอักษรภาษาไทยได้มีนักวิจัยผู้ทรงคุณวุฒิหลายท่านทำการศึกษาค้นคว้าไว้มากมาย ซึ่งพอสรุปวิธีการทำงานได้ดังนี้

1. ผลงานวิจัย รศ.ดร.ชม กิ้มปาน เรื่องการรู้จำรูปแบบอักษรพิมพ์ภาษาไทยของคอมพิวเตอร์ ซึ่งได้นำเสนอไว้ 2 วิธีคือ (อ้างในอภิรักษ์ จิรายุสกุล:2539)

การรู้จำแบบซ้อนทับ (Matching Method) โดยใช้หลักการที่ว่ารูปร่างลักษณะของตัวอักษรภาษาไทยมีรูปร่างที่คล้ายคลึงกัน ซึ่งสามารถจัดเป็นกลุ่มอย่างคร่าว ๆ ได้ ขั้นตอนการวิเคราะห์แบ่งออกเป็น 2 ขั้นตอนใหญ่ ๆ คือ ขั้นตอนการวิเคราะห์แบบหยาบ ขั้นตอนนี้ใช้เทคนิคการทำเบลอเพื่อกำจัดรายละเอียดปลีกย่อย และนำทฤษฎีการกระจายแบบคาร์ฮูแนนโลบ (Karhunen Loeve expansion) หารูปแบบมาตราฐานสำกรับเก็บไว้เป็นตัวแทนของกลุ่ม ส่วนขั้นตอนการวิเคราะห์อย่างละเอียดมี 2 วิธี วิธีแรกใช้การซ้อนทับเป็นส่วน ๆ (subpattern matching) ในการแยกตัวอักษรออกจากกัน วิธีที่สองใช้ฟังชั่นก์การตัดสินใจแบบเชิงเส้น (linear decision functions) บนระนาบของไอเกนเวคเตอร์ที่ได้จากการกระจายแบบคาร์ฮูแนนโลบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การวิเคราะห์โครงสร้าง (Structure Analysis Method) วิธีการนี้นำคุณสมบัติทางโทโพโลยี (Topology) ช่วยในการพิจารณา การแบ่งกลุ่มอย่างหยาบของภาพตัวอักษรที่ต้องการวิเคราะห์ ทำโดยการพิจารณาความสูงและความกว้างของตัวอักษร ซึ่งก่อนการวิเคราะห์จะทำภาพตัวอักษรให้บางก่อน จากนั้นจึงนำคุณสมบัติทางโทโพโลยีมาวิเคราะห์ความแตกต่างของตัวอักษร

2. ผลงานวิจัยของนายทวี เปรมรัตนชัย (2538), รศ.ดร.ชม กิ้มปาน และ ผศ.ดร.บุญธีร์ เครือตราชู เรื่องการรู้จำอักขระตัวพิมพ์ภาษาไทยด้วยวิธีโครงข่ายนิวรอนแบบแบคพรอบพาเกชั่น เสนอการฝึกสอนโครงข่ายนิวรอนด้วยอัลกอริทึมแบบแบคพรอบพาเกชั่น โดยรับข้อมูลจากตัวอักษรที่สแกนจากเครื่องสแกนภาพและนำข้อมูลนั้นมาผ่านโปรแกรมแปลงภาพไฟล์กราฟฟิกไปเป็นไบนารีไฟล์ จากนั้นจะผ่านโปรแกรมแยกอักขระเพื่อนำมาใช้เป็นข้อมูลอินพุตของโปรแกรมนิวรอนที่ใช้ฝึก และทำการทดสอบประสิทธิภาพการรู้จำ โดยโครงข่ายที่จำลองมีขนาด 4 เลเยอร์ คือ 1 อินพุตเลเยอร์ขนาด 1,600 โหนด, 2 ฮิดเดนเลเยอร์ ขนาด 230 โหนด, 1 เอาท์พุตเลเยอร์ขนาด 87 โหนด โดยทำการจำลองโครงข่ายบนเครื่องมินิคอมพิวเตอร์ รุ่น HP 9000/827 ซึ่งมีระบบปฏิบัติการ คือ ยูนิกซ์รุ่น HP-UX 9.04 ผ่านระบบเน็ตเวิร์กเพื่อใช้ในการฝึกสอนโครงข่ายและการรู้จำ ซึ่งผลการวิจัยพกว่าการฝึกสอนใช้เวลา 13 วัน และผลการรู้จำอักษรถูกต้องเฉลี่ย 98% จากอักขระทั้งหมดที่ใช้ในการทดสอบทั้งหมด 60,000 ตัว และเปอร์เซนต์ในการรู้จำหน้ากระดาษในการทดสอบเมื่อทำการกำจัดสิ่งรบกวนจะมีความถูกต้องเฉลี่ย 98% โดยความเร็วของการรู้จำ 1 ตัวอักษรเฉลี่ยประมาณ 1.2 วินาที

3. ผลงานวิจัยของนายอภิรักษ์ จิรายุสกุล (2539) และผศ.ดร.บุญธีร์ เครือตราชู เรื่องการวิเคาะห์ตัวพิมพ์อักษรภาษาไทยโดยใช้ CPN โดยนำเสนอการนำโครงข่ายประสาทเทียม Counterpropragation Network (CPN) มาประยุกต์ใช้ในการจดจำตัวอักษรภาษาไทย และได้มีการเปลี่ยนแปลงโครงสร้างหลักของ CPN 2 ส่วน คือ ส่วนวิธีการเรียนรู้โดยใช้วิธีการแยกตัวสอนตัวอย่างข้อมูลทีละกลุ่ม ส่วนที่ 2 คือให้โครงสร้างชั้นในของคอมแพททิทิพมีความสามารถในการเพิ่มจำนวนโหนด และทำการฝึกสอนโครงข่ายประสาทเทียมและทดสอบประสิทธิภาพการรู้จำ โดยผลการทดสอบความสามารถในการวิเคราะห์ตัวพิมพ์อักษรภาษาไทย 75 ตัว ซึ่งใช้ข้อมูลที่พิมพ์ด้วยเครื่องพิมพ์เลเซอร์จำนวน 15,000 ตัว และข้อมูลที่ผ่านการถ่ายเอกสารจำนวน 15,000 ตัว มีความสามารถในการวิเคราะห์ได้ 98.1% โดยจำนวนตัวอย่างที่นำมาสอนโครงข่ายจำนวน 1,500 ตัว

4. ผลงานวิจัยของนางสาวอัญชลี วานิชทวีวัฒน์ (2540) และ ผศ.ดร.บุญธีร์ เครือตราชู เรื่องการจดจำอักขระภาษาไทยตัวพิมพ์ด้วยโครงข่ายประสาทเทียมแบบเซลล์ออร์แกนไนซิงแมปซ์ โดยการนำเสนอโครงข่ายประสาทเทียมแบบ Self-Organizing Maps (SOM) และ Adaptive Self-Organizing Map มาประยุกต์ใช้ในการจดจำตัวอักษรภาษาไทย และทำการฝึกสอนโครงข่าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประสาทเทียมและทดสอบประสิทธิภาพการรู้จำ โดยการฝึกด้วยฟอนต์อังศนายูพีซีขนาด 12 จุดต่อนิ้ว 75 แบบ จำนวน 1,500 ตัว ด้วยแบบจำลอง SOM จะต้องเสียเวลาในการหาโครงข่ายที่เหมาะสม ส่วน ASOM จะใช้เวลาในการฝึกน้อยกว่าประมาณ 6 ชั่วโมง (บนเครื่อง hp 9000/827) 318 รอบ ใช้ 218 โหนด ประสิทธิภาพในการจดจำจากการทดสอบด้วยตัวอักษร15,000 ตัว ที่เป็นตัวอักษรที่คมชัดและตัวอักษรที่มีสัญญาณรบกวนมีค่าเฉลี่ยเท่ากับ 99.77% , 96.87% ตามลำดับ

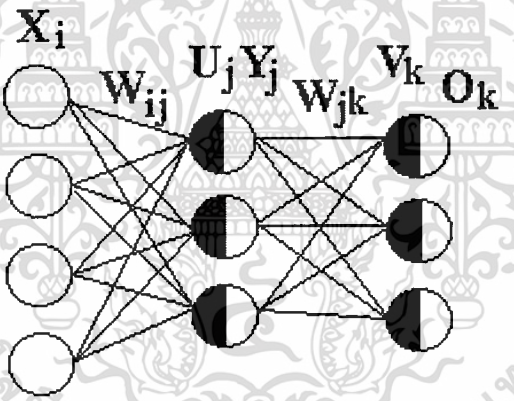
5. ผลงานวิจัยของนางสาวเพ็ญพรรณ ใช้ฮวดเจริญ (2541) และ ผศ.ดร.บุญธีร์ เครือตราชู เรื่องการจดอักษรพิมพ์ภาษาไทยโดยโครงข่ายประสาทเทียมแบบจำลองอะแดปทีฟรีโซแนนซ์เทียรี่ โดยการนำเสนอโครงข่ายประสาทเทียมแบบอะแดปทีฟรีโซแนนซ์เทียรี่ มาประยุกต์ใช้ในการจดจำตัวอักษรภาษาไทย และทำการฝึกสอนโครงข่ายประสาทเทียมและทดสอบประสิทธิภาพการรู้จำ และได้ทำการปรับปรุงแบบจำลองในส่วนของการเรียนรู้ และการเปลี่ยน Threshold เพื่อลดเวลาในขั้นตอนของการเรียนรู้ และลดความผิดพลาดในการจดจำตัวอักษร โดยการฝึกด้วยตัวอักษรภาษาไทย 75 แบบ แบบละ 20 ตัว และทำการทดสอบด้วยตัวอักษรภาษาไทย 75 แบบ แบบละ 200 ตัว จำนวน 15,000 ตัว ซึ่งใช้ข้อมูลที่พิมพ์ด้วยเครื่องพิมพ์เลเซอร์ โดยใช้ฟอนต์อังศนายูพีซีขนาด 12 จุดต่อนิ้ว จะได้ผลความถูกต้องประมาณ 99.13% ใช้เวลาในการฝึก 20 นาที และจำนวนรอบในการฝึก 6 รอบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
วิธีการศึกษา

3.1 สถาปัตยกรรมที่ใช้ในการทดลอง

โครงข่ายประสาทเทียม (Neural Network) ที่ใช้ในการทดลองนี้ เป็นโครงข่ายประสาทเทียมแบบแพร่ย้อนกลับ (Backpropagation Network) ซึ่งเป็นโครงข่ายที่มีการเชื่อมโยงกันเป็นโครงข่ายแบบเป็นลำดับชั้น ตามรูปที่ 3.1 โดยประกอบด้วย Input Layer, Hidden Layer จำนวน 1 Layer และ Output Layer เป็นวิธีการเรียนรู้แบบมีครูสอน (Supervised Learning Algorithm) ที่นำมาใช้ในการเรียนรู้และจดจำลักษณะตัวอักษรในการศึกษาครั้งนี้



รูปที่ 3.1 แสดงโครงข่ายประสาทเทียมแบบแพร่ย้อนกลับที่ใช้ในการทดลอง

3.2 วิธีการสร้างแบบจำลอง

ขั้นตอนในการสร้างแบบจำลองที่ใช้ในการศึกษาครั้งนี้มีดังนี้

1. ศึกษารูปแบบ องค์ประกอบ และอัลกอริทึมของโครงข่ายประสาทเทียม ซึ่งในการศึกษาครั้งนี้ศึกษาโครงข่ายประสาทเทียมแบบแพร่ย้อนกลับ (Backpropagation Network)

2. ทดสอบการคำนวณของโครงข่ายประสาทเทียม ในขั้นต้นจะใช้โปรแกรมประยุกต์ Microsoft Excel ช่วยในการคำนวณขั้นต้นโดยการสมมติโครงข่ายประสาทเทียม (Neural Network) ซึ่งมีจำนวนนิวรอนในชั้นข้อมูลเข้าจำนวน 4 โหนด มีจำนวนนิวรอนในชั้นซ่อนจำนวน 4 โหนด และมี จำนวนนิวรอนในชั้นข้อมูลออกจำนวน 3 โหนด

3. ทำการสุ่มค่าถ่วงน้ำหนัก (Weight) ให้กับแบบจำลองโครงข่ายประสาทเทียม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. ทำการสมมติข้อมูลนำเข้าสู่ชั้นข้อมูลเข้า (Input Layer) ซึ่งมีค่าอยู่ในช่วง 0 ถึง 1

5. ทำการคำนวณทุกขั้นตอน ตั้งแต่การคำนวณหาค่าของนิวรอนในชั้นซ่อน (Hidden Layer) จึงถึงการปรับค่าความผิดพลาด เพื่อดูว่าการคำนวณถูกต้องหรือไม่

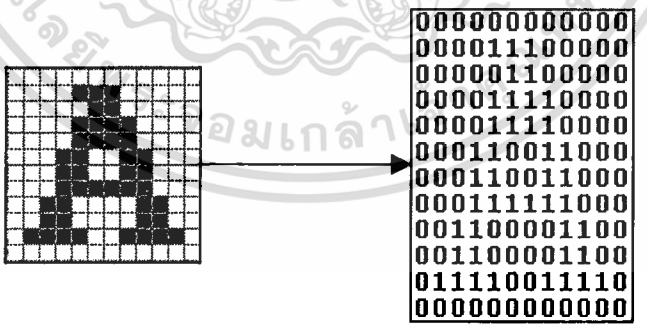
6. ทำการสร้างแบบจำลองโดยใช้โปรแกรมภาษา C++ บนเครื่องไมโครคอมพิวเตอร์

7. ทำการทดสอบแบบจำลองที่พัฒนาขึ้นมาจากโปรแกรมภาษา C++ แล้วนำค่าถ่วงนำหนักซึ่งได้มาจากการสุ่ม (Random) ในแบบจำลองที่พัฒนามาจากโปรแกรมภาษา C++ กับแบบจำลองที่บันทึกในโปรแกรมประยุกต์ Microsoft Excel แล้วทำการคำนวณข้อมูล และสังเกตว่าผลที่ได้จากการคำนวณว่าเหมือนกันไม่ ทั้งนี้เพื่อเป็นการตรวจสอบความถูกต้องของแบบจำลองที่พัฒนาขึ้นมาจากโปรแกรมภาษา C++

8. เมื่อได้ผลลัพธ์ที่เท่ากันแล้ว ก็จะนำแบบจำลองที่พัฒนาขึ้นมาจากโปรแกรมภาษา C++ ไปใช้ในการทดสอบการเรียนรู้และจดจำลักษณะตัวอักษรต่อไป

3.3 ขั้นตอนการเตรียมข้อมูล

การเตรียมข้อมูลเพื่อที่จะนำไปใช้ในการเรียนรู้และทดสอบแบบจำลองโครงข่ายประสาทเทียม โดยทำการแปลงตัวอักษรซึ่งเป็นฟอนต์ Courier ตัวหนา ขนาด 12 จุดต่อนิ้ว จำนวน 26 ตัวอักษร ให้อยู่ในลักษณะแฟ้มข้อมูลตัวอักษร (Text File) ขนาด 12 x 12 ตำแหน่ง ซึ่งประกอบด้วยข้อมูล 2 ตัวลักษณะด้วยกัน คือ 0 ซึ่งหมายถึงพื้นที่ว่างเปล่า และ 1 หมายถึงพื้นที่ใช้ในการแสดงให้เห็นเป็นตัวอักษร เช่น

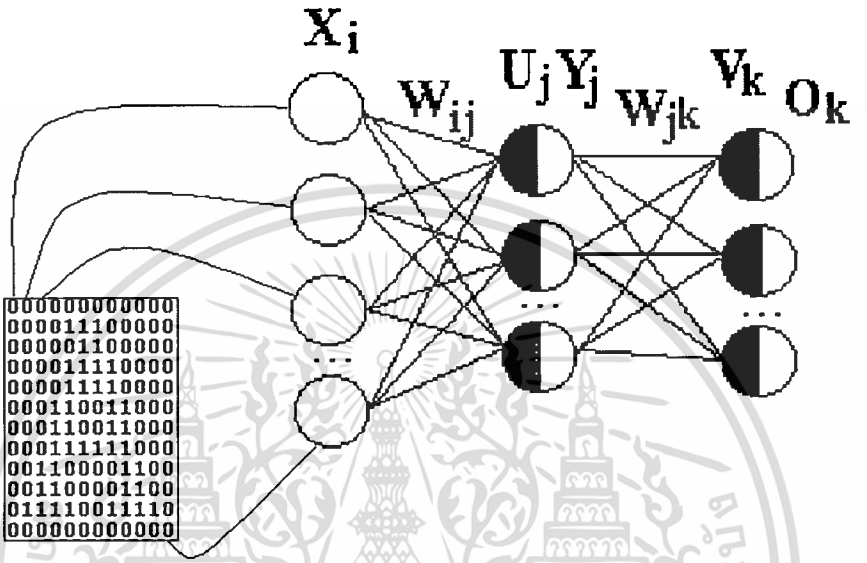


รูปที่ 3.2 แสดงตัวอย่างการแปลงตัวอักษรให้อยู่ในรูปของ Text File

นำอักษร A ซึ่งอยู่ในรูปของ Text File ที่ได้เข้าสู่โครงข่ายโดยแทนข้อมูลแต่ละตัว (ซึ่งมีค่าเป็น 0 หรือ 1) และเพื่อป้องกันความผิดพลาดของข้อมูลบางส่วน จึงแทนค่า 0 ด้วยค่า 0.2 และ แทนค่า 1 ด้วยค่า 0.8 เป็นข้อมูลของแต่ละนิวรอน ในส่วนของการรับข้อมูลเข้าสู่ระบบ (Input Layer) ดังรูปที่ 3.3 เพื่อให้แบบจำลองโครงข่ายเรียนรู้ และปรับการเรียนรู้จนกว่าค่าความผิดพลาดที่โครงข่าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คำนวณได้มีค่าผิดพลาดที่ยอมรับได้ ก็จะหยุดการเรียนรู้ แล้วนำโครงข่ายนั้นมาทดสอบกับข้อมูลที่มีความผิดพลาด (Noise) บ้างในบางส่วน เพื่อเป็นข้อพิสูจน์ว่าโครงข่ายสามารถเรียนรู้และจดจำลักษณะตัวอักษรได้หรือไม่



รูปที่ 3.3 แสดงการนำข้อมูลเข้าสู่แบบจำลองโครงข่ายประสาทเทียม

3.4 ขั้นตอนการทำงานของแบบจำลอง

แบบจำลองโครงข่ายประสาทเทียมที่พัฒนาขึ้นมาขั้นตอนการทำงานดังต่อไปนี้

1. อ่านค่าถ่วงน้ำหนักจากแฟ้มข้อมูลที่เก็บไว้ หากไม่พบทำการสุ่มข้อมูลถ่วงน้ำหนัก (Weight) ใหม่ทุกเส้นของการเชื่อมต่อในโครงข่ายประสาทเทียม ซึ่งจะเป็นค่าตัวเลขสุ่มที่อยู่ในช่วง –1 ถึง 1

2. อ่านข้อมูลจากแฟ้มข้อมูล Text File ซึ่งเป็นข้อมูลนำเข้าของโครงข่าย พร้อมทั้งแปลข้อมูลให้อยู่ในรูปของตัวเลข 0.2 และ 0.8

3. คำนวณค่า $\sum_{i=1}^{i=n} X_i W_{ij} = U_j$ สำหรับแต่ละนิวรอน j ของชั้นซ่อน

4. คำนวณค่ากระตุ้น (Activation) สำหรับแต่ละนิวรอน j ของชั้นซ่อน:$Y_j = f(U_j)$ โดยฟังก์ชันที่นิยมใช้คือ $f(x) = \frac{1}{1+e^{-x}}$ (Logistic Function)

5. คำนวณค่า $\sum_{j=1}^{j=m} Y_j W_{jk} = V_k$ สำหรับแต่ละนิวรอน k ของชั้น Output

6. คำนวณค่ากระตุ้น (Activation) สำหรับแต่ละนิวรอน j ของชั้น Output : $O_k = f(V_t)$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7. นำค่า O_k มาเปรียบเทียบกับ Target (หรือ Desired Output) และคำนวณหาค่าผิดพลาดทั้งหมด (Sum Error) ที่เกิดขึ้น (เป็นผลรวมของค่าความผิดพลาดกำลังสอง ซึ่งได้มาจากการคำนวณค่าผิดพลาดในแต่ละรอบของการเรียนรู้ เช่น ใน 1 รอบของการเรียนรู้มีตัวอักษรทั้งหมด 5 ตัวอักษร คือ A ถึง E ในครั้งแรกที่มีการคำนวณค่าความผิดพลาดที่เกิดขึ้นจากตัวอักษร A ถึง E) ถ้าค่าความผิดพลาดรวมที่ได้น้อยกว่าระดับที่ตั้งไว้ ก็จะสิ้นสุดการเรียนรู้ มิฉะนั้นทำข้อ 8 ต่อ

8. คำนวณค่าของการปรับค่าถ่วงน้ำหนัก (Weight) สำหรับแต่ละค่าถ่วงน้ำหนัก

8.1 สำหรับเส้นเชื่อมระหว่างชั้นซ่อนกับชั้น Output

$$\Delta w_{ik} = \alpha \delta_k Y_j \text{ โดยที่ } \delta_k = (T_k - O_k) f'(V_k)$$

$$\text{ซึ่ง } f'(V_k) = O_k(1 - O_k)$$

8.2 สำหรับเส้นเชื่อมระหว่างชั้น Input กับชั้นซ่อน

$$\Delta w_{ij} = \alpha \delta_j X_i \text{ โดยที่ } \delta_j = \sum \delta_k W_{jk} f'(U_j)$$

9. ปรับค่าถ่วงน้ำหนักของเส้นเชื่อมจากนิวรอน r ไปยังนิวรอน s ด้วยสมการต่อไปนี้

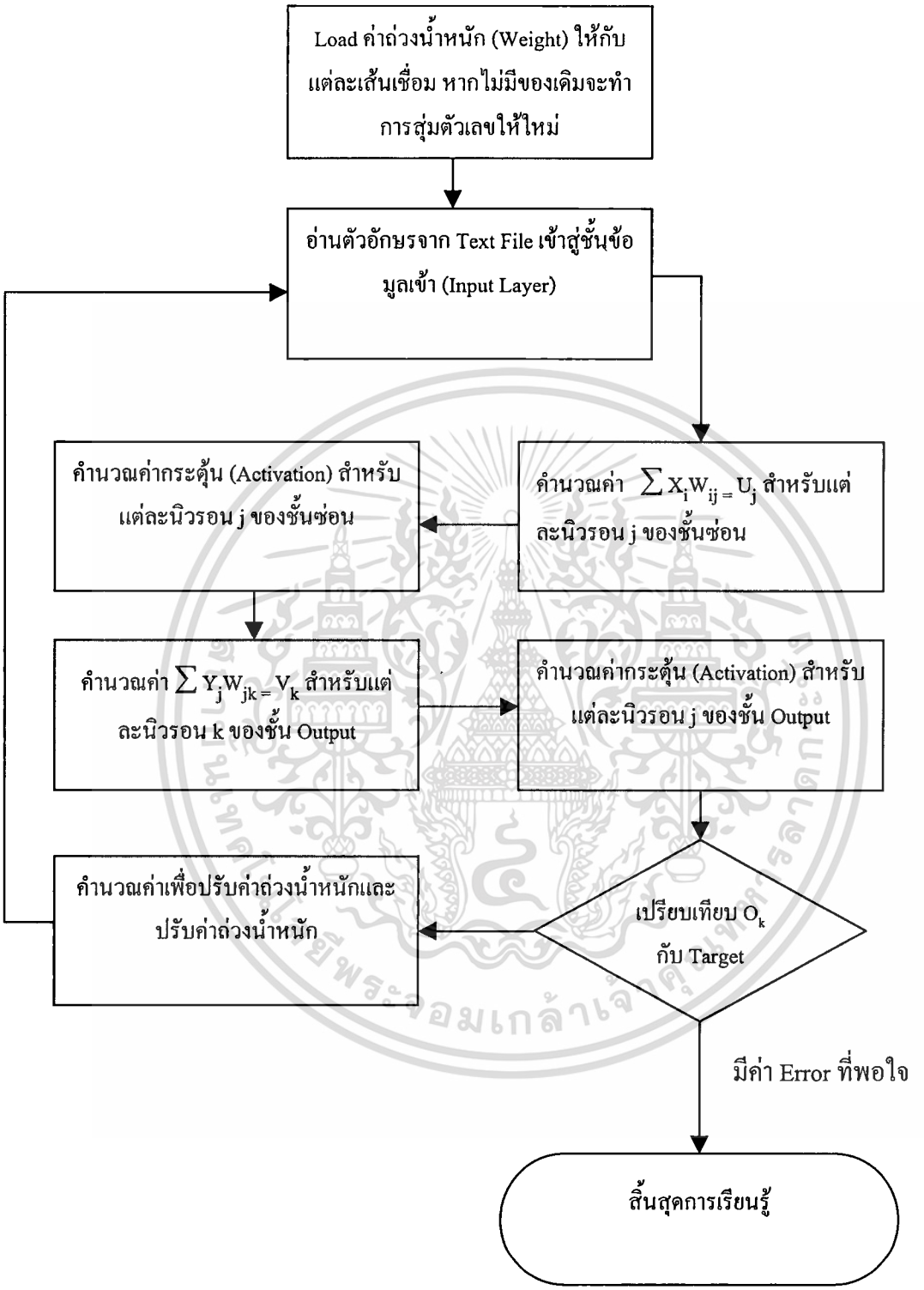
$$W_{rs}\,(\text{new}) = W_{rs}\,(\text{old}) + \Delta W_{rs}$$

10. ปรับค่าถ่วงน้ำหนักของเส้นเชื่อมภายในโครงข่าย

11. กลับไปทำข้อ 3 อีกครั้ง

เมื่อเสร็จสิ้นการเรียนรู้ จากนั้นจึงนำโครงข่ายประสาทเทียม (Neural Network) ที่ได้ไปทำการทดสอบ (Testing) ว่าสามารถระบุตัวอักษรได้หรือไม่ หากพบว่าโครงข่ายยังไม่สามารถเรียนรู้ตัวอักษรในลักษณะเดียวกันกับที่ใช้ในการเรียนรู้โครงข่ายได้ ก็จะหาวิธีการปรับปรุง แก้ไขโครงข่าย หรือเปลี่ยนแปลงข้อมูลที่ใช้ในการเรียนรู้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.4 แสดงขั้นตอนการทำงานของแบบจำลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

ผลการทดลอง

4.1 ตัวอักษรที่ใช้ในการทดลอง

ตัวอักษรที่ใช้ในการทดลองจะทำการแปลงจากแฟ้มข้อมูลภาพ (Graphic File) เป็นแฟ้มข้อมูลตัวอักษร (Test File) โดยตัวอักษรที่ใช้มี 2 ประเภทด้วยกันคือ

1. ตัวอักษรที่ใช้ในการเรียนรู้ (Learn) ซึ่งตัวอักษรที่ใช้มี 2 ขนาดคือ

1.1 ขนาด 12 x 12 พิกเซล (Pixel) ตัวอักษร CordiaUPC ขนาด 16 จุดต่อนิ้ว ตัวหนา

1.2 ขนาด 20 x 20 พิกเซล (Pixel) ตัวอักษร CordiaUPC ขนาด 26 จุดต่อนิ้ว ตัวหนา

2. ตัวอักษรที่ใช้ในการทดสอบ (Test) ความถูกต้องของโครงข่ายที่ใช้ในการทดลอง ซึ่งตัวอักษรที่ใช้มี 2 ขนาดคือ

2.1 ขนาด 12 x 12 พิกเซล (Pixel) คือ ตัวอักษร CordiaUPC ขนาด 16 จุดต่อนิ้ว ตัวหนาที่มีความผิดพลาด (Noise) ตัวอักษร Noise 10% ตัวอักษร AngsanaUPC ขนาด 16 จุดต่อนิ้ว ตัวอักษร AngsanaUPC ขนาด 16จุดต่อนิ้ว ตัวหนา ตัวอักษร Arial ขนาด 10 จุดต่อนิ้ว ตัวอักษร BrowalliaUPC ขนาด 16 จุดต่อนิ้ว ตัวอักษร DelleniaUPC ขนาด 16 จุดต่อนิ้ว ตัวอักษร EucrosiaUPC ขนาด 16 จุดต่อนิ้ว ตัวอักษร FreesiaUPC ขนาด 16 จุดต่อนิ้ว ตัวอักษร Garamond ขนาด 11 จุดต่อนิ้ว ตัวอักษร JasmineUPC ขนาด 16 จุดต่อนิ้ว และ ตัวอักษร MSSansSerif ขนาด 10 จุดต่อนิ้ว

2.2 ขนาด 20 x 20 พิกเซล (Pixel) คือ ตัวอักษรที่ใช้ในการเรียนรู้ ตัวอักษร CordiaUPC ขนาด 26 จุดต่อนิ้ว ตัวหนาที่มีความผิดพลาด (Noise) 5% และ 10% ตัวอักษร AngsanaUPC ขนาด 26 จุดต่อนิ้ว ตัวอักษร Arial ขนาด 17 จุดต่อนิ้ว ตัวอักษร BookAntiqua ขนาด 18 จุดต่อนิ้ว ตัวอักษร BrowalliaUPC ขนาด 26 จุดต่อนิ้ว ตัวอักษร DelleniaUPC ขนาด 26 จุดต่อนิ้ว ตัวอักษร EucrosiaUPC ขนาด 26 จุดต่อนิ้ว ตัวอักษร FreesiaUPC ขนาด 26 จุดต่อนิ้ว ตัวอักษร Garamond ขนาด 20 จุดต่อนิ้ว JasmineUPC ขนาด 26 จุดต่อนิ้ว และ ตัวอักษร MSSansSerif ขนาด 17 จุดต่อนิ้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 ผลการทดลอง

จากการทดลองแบบจำลองโครงข่ายประสาทเทียม ที่ใช้ในการรู้จำและระบุตัวอักษร ผู้ศึกษาสรุปได้สรุปเป็นการทดลองในขั้นตอนต่างๆ ดังต่อไปนี้

ข้อมูล	รายละเอียด
ตัวอักษรที่ใช้	CordiaUPC ขนาด 16 จุดต่อนิ้ว ตัวหนา
จำนวนตัวอักษรที่ใช้ในการเรียนรู้	5
ขนาดข้อมูลของแต่ละตัวอักษรใน Text File	12 x 12
จำนวนนิวรอนในชั้นข้อมูลเข้า	144
จำนวนนิวรอนในชั้นซ่อน	200
จำนวนนิวรอนในชั้นข้อมูลออก	26
อัตราการเรียนรู้ (Learning Rate)	0. 1
ค่าความผิดพลาดรวม (ค่าความผิดพลาด^2 x 10^5)	0.000011
ทดสอบด้วยข้อมูลที่ใช้ในการเรียนรู้	5 ตัวอักษร (100%)

ตารางที่ 4.1 แสดงผลการทดลองกับตัวอักษร CordiaUPC ขนาด 16 จุดต่อนิ้ว ตัวหนา 5 ตัวอักษร ด้วยจำนวนนิวรอนในชั้นซ่อน 200 นิวรอน

แบบจำลองโครงข่ายที่ใช้ในการเรียนรู้ในการทดลองครั้งนี้ มีจำนวนนิวรอนในชั้นข้อมูลเข้าจำนวน 144 นิวรอน จำนวนนิวรอนในชั้นซ่อนจำนวน 200 นิวรอน จำนวนนิวรอนในข้อมูลออกจำนวน 26 นิวรอน ทำการเรียนรู้ตัวอักษรโดยใช้ฟอนต์ CordiaUPC ขนาด 16 จุดต่อนิ้ว ตัวหนา จำนวน 5 ตัวอักษร ได้แก่ A, B, C, E และ F มีอัตราการเรียนรู้ (Learning Rate) เท่ากับ 0.1 ผลที่ได้มีค่าความผิดพลาดรวม 0.000011 จึงหยุดการเรียนรู้แล้วนำโครงข่าย (Neural Network) ที่ได้ไปใช้ทดสอบ (Testing) ผลการทดลองปรากฏว่าสามารถรู้จำตัวอักษรทั้ง 5 ตัวอักษรที่ใช้ในการเรียนรู้ได้ 100%

สรุปการทดลองในขั้นตอนนี้ทำให้ทราบว่าแบบจำลองโครงข่ายสามารถที่จะเรียนรู้และระบุตัวอักษรได้ จึงทำการทดลองในขั้นต่อไปโดยให้แบบจำลองโครงข่ายทำการเรียนรู้ตัวอักษรเพิ่มขึ้นเป็นจำนวน 26 ตัวอักษร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อมูล	รายละเอียด
ตัวอักษรที่ใช้	CordiaUPC ขนาด 16 จุดต่อนิ้ว ตัวหนา
จำนวนตัวอักษรที่ใช้ในการเรียนรู้	26
ขนาดข้อมูลของแต่ละตัวอักษรใน Text File	12 x 12
จำนวนนิวรอนในชั้นข้อมูลเข้า	144
จำนวนนิวรอนในชั้นซ่อน	200
จำนวนนิวรอนในชั้นข้อมูลออก	26
อัตราการเรียนรู้ (Learning Rate)	0.1
ค่าความผิดพลาดรวม (ค่าความผิดพลาด^2 x 10^8)	0.000476
ทดสอบด้วยข้อมูลที่ใช้ในการเรียนรู้	14 ตัวอักษร (53.85%)

ตารางที่ 4.2 แสดงผลการทดลองกับตัวอักษร CordiaUPC ขนาด 16 จุดต่อนิ้ว ตัวหนา 26 ตัวอักษร ด้วยจำนวนนิวรอนในชั้นซ่อน 200 นิวรอน

แบบจำลองโครงข่ายที่ใช้ในการเรียนรู้ในการทดลองครั้งนี้ มีจำนวนนิวรอนในชั้นข้อมูลเข้าจำนวน 144 นิวรอน จำนวนนิวรอนในชั้นซ่อนจำนวน 200 นิวรอน จำนวนนิวรอนในข้อมูลออกจำนวน 26 นิวรอน ทำการเรียนรู้ตัวอักษรโดยใช้ฟอนต์ CordiaUPC ขนาด 16 จุดต่อนิ้ว ตัวหนา จำนวน 26 ตัวอักษร มีอัตราการเรียนรู้ (Learning Rate) เท่ากับ 0.1 ผลที่ได้มีค่าความผิดพลาดรวม 0.000476 จึงหยุดการเรียนรู้แล้วนำโครงข่าย (Neural Network) ที่ได้ไปใช้ทดสอบ (Testing) ผลการทดลองปรากฏว่าสามารถรู้จำตัวอักษรได้ 14 ตัวอักษร คิดเป็น 53.85% ไม่สามารถระบุตัวอักษร A, O, J, M, N, O, P, T, U, W และ Y ได้เท่านั้น

สรุปการทดลองในขั้นตอนนี้ทำให้ทราบว่าแบบจำลองโครงข่ายยังไม่สามารถระบุตัวอักษรได้ทั้งหมด ผู้ศึกษามีข้อสันนิษฐานว่าจำนวนนิวรอนในชั้นซ่อนอาจจะไม่เหมาะสม ดังนั้นในการทดลองขั้นต่อไปจึงทำการปรับโครงข่ายให้มีจำนวนนิวรอนในชั้นซ่อนลดลงเหลือเพียง 150 นิวรอน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อมูล	รายละเอียด
ตัวอักษรที่ใช้	CordiaUPC ขนาด 16 จุดต่อนิ้ว ตัวหนา
จำนวนตัวอักษรที่ใช้ในการเรียนรู้	26
ขนาดข้อมูลของแต่ละตัวอักษรใน Text File	12 x 12
จำนวนนิวรอนในชั้นข้อมูลเข้า	144
จำนวนนิวรอนในชั้นซ่อน	150
จำนวนนิวรอนในชั้นข้อมูลออก	26
อัตราการเรียนรู้ (Learning Rate)	0.1
ค่าความผิดพลาดรวม (ค่าความผิดพลาด^2 x 10^8)	0.000460
ทดสอบด้วยข้อมูลที่ใช้ในการเรียนรู้	24 ตัวอักษร (92.31%)

ตารางที่ 4.3 แสดงผลการทดลองกับตัวอักษร CordiaUPC ขนาด 16 จุดต่อนิ้ว ตัวหนา ด้วยจำนวนนิวรอนในชั้นซ่อน 150 นิวรอน

แบบจำลองโครงข่ายที่ใช้ในการเรียนรู้ในการทดลองครั้งนี้ มีจำนวนนิวรอนในชั้นข้อมูลเข้าจำนวน 144 นิวรอน จำนวนนิวรอนในชั้นซ่อนจำนวน 150 นิวรอน จำนวนนิวรอนในข้อมูลออกจำนวน 26 นิวรอน ทำการเรียนรู้ตัวอักษรโดยใช้ฟอนต์ CordiaUPC ขนาด 16 จุดต่อนิ้ว ตัวหนา จำนวน 26 ตัวอักษร มีอัตราการเรียนรู้ (Learning Rate) เท่ากับ 0.1 ผลที่ได้มีค่าความผิดพลาดรวม 0.000460 จึงหยุดการเรียนรู้แล้วนำโครงข่าย (Neural Network) ที่ได้ไปใช้ทดสอบ (Testing) ผลการทดลองปรากฏว่าสามารถรู้จำตัวอักษรได้ 24 ตัวอักษร คิดเป็น 92.31% ไม่สามารถระบุตัวอักษร J และ L ได้

สรุปการทดลองในขั้นตอนนี้ทำให้ทราบว่าแบบจำลองโครงข่ายยังไม่สามารถระบุตัวอักษรได้ทั้งหมด ผู้ศึกษามีข้อสันนิษฐานว่าจำนวนนิวรอนในชั้นซ่อนอาจจะไม่เหมาะสม ดังนั้นในการทดลองขั้นต่อไปจึงทำการปรับโครงข่ายให้มีจำนวนนิวรอนในชั้นซ่อนลดลงเหลือเพียง 100 นิวรอน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อมูล	รายละเอียด
ตัวอักษรที่ใช้	CordiaUPC ขนาด 16 จุดต่อนิ้ว ตัวหนา
จำนวนตัวอักษรที่ใช้ในการเรียนรู้	26
ขนาดข้อมูลของแต่ละตัวอักษรใน Text File	12 x 12
จำนวนนิวรอนในชั้นข้อมูลเข้า	144
จำนวนนิวรอนในชั้นซ่อน	100
จำนวนนิวรอนในชั้นข้อมูลออก	26
อัตราการเรียนรู้ (Learning Rate)	0.1
ค่าความผิดพลาดรวม (ค่าความผิดพลาด^2 x 10^8)	0.000846
ทดสอบด้วยข้อมูลที่ใช้ในการเรียนรู้	26 ตัวอักษร (100.00%)
ทดสอบด้วยข้อมูลที่ใช้ในการเรียนรู้ที่มีความผิดพลาด 10%	17 ตัวอักษร (65.38%)

ตารางที่ 4.4 แสดงผลการทดลองกับตัวอักษร CordiaUPC ขนาด 16 จุดต่อนิ้ว ตัวหนา ด้วยจำนวนนิวรอนในชั้นซ่อน 100 นิวรอน

แบบจำลองโครงข่ายที่ใช้ในการเรียนรู้ในการทดลองครั้งนี้ มีจำนวนนิวรอนในชั้นข้อมูลเข้าจำนวน 144 นิวรอน จำนวนนิวรอนในชั้นซ่อนจำนวน 100 นิวรอน จำนวนนิวรอนในข้อมูลออกจำนวน 26 นิวรอน ทำการเรียนรู้ตัวอักษรโดยใช้ฟอนต์ CordiaUPC ขนาด 16 จุดต่อนิ้ว ตัวหนา จำนวน 26 ตัวอักษร มีอัตราการเรียนรู้ (Learning Rate) เท่ากับ 0.1 ผลที่ได้มีค่าความผิดพลาดรวม 0.000846 จึงหยุดการเรียนรู้แล้วนำโครงข่าย (Neural Network) ที่ได้ไปใช้ทดสอบ (Testing) ผลการทดลองปรากฏว่าสามารถรู้จำตัวอักษรได้ทั้งหมด 26 ตัวอักษร คิดเป็น 100.00% เมื่อโครงข่ายสามารถระบุตัวอักษรได้ทั้งหมด จึงได้ลองนำโครงข่ายไปทดสอบกับตัวอักษรที่ใช้ในการเรียนรู้เช่นกันแต่มีความผิดพลาด 10% ผลปรากฏว่าสามารถรู้จำตัวอักษรได้เพียง 17 ตัวอักษร คิดเป็น 65.38% ซึ่งโครงข่ายไม่สามารถระบุตัวอักษร A, F, K, L, N, O, R, V และ Y ได้

สรุปการทดลองในขั้นตอนนี้ได้ทำให้ทราบว่าแบบจำลองโครงข่ายแม้ว่าจะสามารถระบุระบุตัวอักษรได้ทั้งหมด แต่ก็ยังไม่สามารถระบุตัวอักษรที่มีความผิดพลาด 10% ได้ทั้งหมด ดังนั้นผู้ศึกษาจึงได้ทำการทดลองในขั้นต่อไป โดยทำการปรับโครงข่ายให้มีจำนวนนิวรอนในชั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซ่อนลดลง เพื่อศึกษาผลที่ได้จากการทดลองว่าแบบจำลองโครงข่ายที่มีจำนวนนิวรอนเท่าใดหมาะสมกับการใช้งานมากที่สุด

ข้อมูล	รายละเอียด
ตัวอักษรที่ใช้	CordiaUPC ขนาด 16 จุดต่อนิ้ว ตัวหนา
จำนวนตัวอักษรที่ใช้ในการเรียนรู้	26
ขนาดข้อมูลของแต่ละตัวอักษรใน Text File	12 x 12
จำนวนนิวรอนในชั้นข้อมูลเข้า	144
จำนวนนิวรอนในชั้นซ่อน	50
จำนวนนิวรอนในชั้นข้อมูลออก	26
อัตราการเรียนรู้ (Learning Rate)	0.1
ค่าความผิดพลาดรวม (ค่าความผิดพลาด^2 x 10^8)	0.001715
ทดสอบด้วยข้อมูลที่ใช้ในการเรียนรู้	26 ตัวอักษร (100.00%)
ทดสอบด้วยข้อมูลที่ใช้ในการเรียนรู้ที่มีความผิดพลาด 10%	23 ตัวอักษร (88.46%)

ตารางที่ 4.5 แสดงผลการทดลองกับตัวอักษร CordiaUPC ขนาด 16 จุดต่อนิ้ว ตัวหนา ด้วยจำนวนนิวรอนในชั้นซ่อน 50 นิวรอน

แบบจำลองโครงข่ายที่ใช้ในการเรียนรู้ในการทดลองครั้งนี้ มีจำนวนนิวรอนในชั้นข้อมูลเข้าจำนวน 144 นิวรอน จำนวนนิวรอนในชั้นซ่อนจำนวน 50 นิวรอน จำนวนนิวรอนในข้อมูลออกจำนวน 26 นิวรอน ทำการเรียนรู้ตัวอักษรโดยใช้ฟอนต์ CordiaUPC ขนาด 16 จุดต่อนิ้ว ตัวหนา จำนวน 26 ตัวอักษร มีอัตราการเรียนรู้ (Learning Rate) เท่ากับ 0.1 ผลที่ได้มีค่าความผิดพลาดรวม 0.001715 จึงหยุดการเรียนรู้แล้วนำโครงข่าย (Neural Network) ที่ได้ไปใช้ทดสอบ (Testing) ผลการทดลองปรากฏว่าสามารถรู้จำตัวอักษรได้ทั้งหมด 26 ตัวอักษร คิดเป็น 100.00% เมื่อโครงข่ายสามารถระบุตัวอักษรได้ทั้งหมด จึงได้ลองนำโครงข่ายไปทดสอบกับตัวอักษรที่ใช้ในการเรียนรู้เช่นกันแต่มีความผิดพลาด 10% ผลปรากฏว่าสามารถรู้จำตัวอักษรได้เพียง 23 ตัวอักษร คิดเป็น 88.46% ซึ่งโครงข่ายไม่สามารถระบุตัวอักษร L, M และ V ได้

สรุปการทดลองในขั้นตอนนี้ได้ทำให้ทราบว่าแบบจำลองโครงข่ายแม้ว่าจะสามารถระบุระบุตัวอักษรได้ทั้งหมด แต่ก็ยังไม่สามารถระบุตัวอักษรที่มีความผิดพลาด 10% ได้ทั้งหมด ดัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

นั้นผู้ศึกษาจึงได้ทำการทดลองในขั้นต่อไป โดยทำการปรับโครงข่ายให้มีจำนวนนิวรอนในชั้นซ่อนเพิ่มขึ้นบ้าง เพื่อศึกษาผลที่ได้จากการทดลองว่าแบบจำลองโครงข่ายที่มีจำนวนนิวรอนเท่าใดหมาะสมกับการใช้งานมากที่สุด

ข้อมูล	รายละเอียด
ตัวอักษรที่ใช้	CordiaUPC ขนาด 16 จุดต่อนิ้ว ตัวหนา
จำนวนตัวอักษรที่ใช้ในการเรียนรู้	26
ขนาดข้อมูลของแต่ละตัวอักษรใน Text File	12 x 12
จำนวนนิวรอนในชั้นข้อมูลเข้า	144
จำนวนนิวรอนในชั้นซ่อน	250
จำนวนนิวรอนในชั้นข้อมูลออก	26
อัตราการเรียนรู้ (Learning Rate)	0.1
ค่าความผิดพลาดรวม (ค่าความผิดพลาด^2 x 10^5)	0.000274
ทดสอบด้วยข้อมูลที่ใช้ในการเรียนรู้	14 ตัวอักษร (53.85%)

ตารางที่ 4.6 แสดงผลการทดลองกับตัวอักษร CordiaUPC ขนาด 16 จุดต่อนิ้ว ตัวหนา ด้วยจำนวนนิวรอนในชั้นซ่อน 250 นิวรอน

แบบจำลองโครงข่ายที่ใช้ในการเรียนรู้ในการทดลองครั้งนี้ มีจำนวนนิวรอนในชั้นข้อมูลเข้าจำนวน 144 นิวรอน จำนวนนิวรอนในชั้นซ่อนจำนวน 250 นิวรอน จำนวนนิวรอนในข้อมูลออกจำนวน 26 นิวรอน ทำการเรียนรู้ตัวอักษรโดยใช้ฟอนต์ CordiaUPC ขนาด 16 จุดต่อนิ้ว ตัวหนา จำนวน 26 ตัวอักษร มีอัตราการเรียนรู้ (Learning Rate) เท่ากับ 0.1 ผลที่ได้มีค่าความผิดพลาดรวม 0.000274 จึงหยุดการเรียนรู้แล้วนำโครงข่าย (Neural Network) ที่ได้ไปใช้ทดสอบ (Testing) ผลการทดลองปรากฏว่าสามารถรู้จำตัวอักษรได้เพียง 14 ตัวอักษร คิดเป็น 53.85% ไม่สามารถระบุตัวอักษร A, G, H, I, J, L, O, T, V, X, Y และ Z ได้

สรุปการทดลองในขั้นตอนนี้ทำให้ทราบว่าแบบจำลองโครงข่ายที่มีจำนวนนิวรอนในชั้นซ่อนขนาด 50 และ 100 เป็นขนาดที่เหมาะสมกว่าจำนวนนิวรอนในชั้นซ่อนขนาด 150, 200, 250

สำหรับแบบจำลองโครงข่ายที่มีจำนวนนิวรอนในชั้นซ่อน 50 และ 100 นิวรอน หลังจากได้มีการทดสอบโดยให้โครงข่ายเรียนรู้แล้ว ผู้ศึกษามีข้อสังเกตุว่า ควรมีการปรับปรุงให้โครงข่ายมีการเรียนรู้ได้เร็วขึ้นจึงมีการให้แบบจำลองโครงข่ายมีการเปลี่ยนแปลงอัตราการเรียนรู้ (Learning

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Rate) ในระหว่างที่เรียนรู้ (Training) โดยในการปรับแบบจำลองโครงข่ายจะเก็บค่า Total Error ที่ต่ำสุดไว้ เมื่อโครงข่ายไม่มีการเปลี่ยน Total Error เป็นจำนวน 50 รอบของการเรียนรู้ แบบจำลองโครงข่ายจะทำการลดอัตราการเรียนรู้ (Learning Rate) ลง แล้วเริ่มเก็บบันทึก Total Error ใหม่ ทำเช่นนี้ไปเรื่อย ๆ จนกว่าค่าอัตราการเรียนรู้ (Learning Rate) ถึงจุดต่ำสุดที่กำหนด แบบโครงข่ายจะไม่มีการเปลี่ยนแปลงอัตราการเรียน (Learning Rate) รู้อีก

ในขั้นตอนต่อไปจะเป็นการทดลองกับแบบจำลองโครงข่ายที่มีการเปลี่ยนแปลงอัตราการเรียนรู้ (Learning Rate) ในระหว่างที่มีการเรียนรู้ โดยจะทำการทดสอบกับแบบจำลองโครงข่ายที่มีจำนวนนิวรอนในชั้นซ่อน 50 และ 100 นิวรอน

ข้อมูล	รายละเอียด (เดิม)	รายละเอียด
ตัวอักษรที่ใช้	CordiaUPC ขนาด 16 จุดต่อนิ้ว ตัวหนา	CordiaUPC ขนาด 16 จุดต่อนิ้ว ตัวหนา
จำนวนตัวอักษรที่ใช้ในการเรียนรู้	26	26
ขนาดข้อมูลของแต่ละตัวอักษรใน Text File	12 x 12	12 x 12
จำนวนนิวรอนในชั้นข้อมูลเข้า	144	144
จำนวนนิวรอนในชั้นซ่อน	50	50
จำนวนนิวรอนในชั้นข้อมูลออก	26	26
อัตราการเรียนรู้ (Learning Rate)	0.1	0.7 - 0.1
ค่าความผิดพลาดรวม (ค่าความผิดพลาด^2 x 10^8)	0.001715	0.000086
ทดสอบด้วยข้อมูลที่ใช้ในการเรียนรู้	26 ตัวอักษร (100.00%)	26 ตัวอักษร (100.00%)
ทดสอบด้วยข้อมูลที่ใช้ในการเรียนรู้ที่มีความผิดพลาด 10%	23 ตัวอักษร (88.46%)	24 ตัวอักษร (92.31%)

ตารางที่ 4.7 แสดงผลการทดลองกับตัวอักษร CordiaUPC ขนาด 16 จุดต่อนิ้ว ตัวหนา ด้วยจำนวนนิวรอนในชั้นซ่อน 50 นิวรอน โดยมีการเปลี่ยนค่าอัตราการเรียนรู้

แบบจำลองโครงข่ายที่ใช้ในการเรียนรู้ในการทดลองครั้งนี้ มีจำนวนนิวรอนในชั้นข้อมูลเข้าจำนวน 144 นิวรอน จำนวนนิวรอนในชั้นซ่อนจำนวน 50 นิวรอน จำนวนนิวรอนในข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ออกจำนวน 26 นิวรอน ทำการเรียนรู้ตัวอักษรโดยใช้ฟอนต์ CordiaUPC ขนาด 16 จุดต่อนิ้ว ตัวหนา จำนวน 26 ตัวอักษร มีอัตราการเรียนรู้ (Learning Rate) ที่ปรับเปลี่ยนระหว่างการเรียนรู้อยู่ในช่วง0.1 – 0.7 ผลที่ได้มีค่าความผิดพลาดรวม 0.000086 จึงหยุดการเรียนรู้แล้วนำโครงข่าย (Neural Network) ที่ได้ไปใช้ทดสอบ (Testing) ผลการทดลองปรากฏว่าสามารถรู้จำตัวอักษรได้ทั้งหมด 26 ตัวอักษร คิดเป็น 100.00% เมื่อโครงข่ายสามารถระบุตัวอักษรได้ทั้งหมด จึงได้ลองนำโครงข่ายไปทดสอบกับตัวอักษรที่ใช้ในการเรียนรู้เช่นกันแต่มีความผิดพลาด 10% ผลปรากฏว่าสามารถรู้จำตัวอักษรได้เพียง 24 ตัวอักษร คิดเป็น 92.31% ซึ่งโครงข่ายไม่สามารถระบุตัวอักษร O และ Y ได้

ข้อมูล	รายละเอียด (เดิม)	รายละเอียด
ตัวอักษรที่ใช้	CordiaUPC ขนาด 16 จุดต่อนิ้ว ตัวหนา	CordiaUPC ขนาด 16 จุดต่อนิ้ว ตัวหนา
จำนวนตัวอักษรที่ใช้ในการเรียนรู้	26	26
ขนาดข้อมูลของแต่ละตัวอักษรใน Text File	12 x 12	12 x 12
จำนวนนิวรอนในชั้นข้อมูลเข้า	144	144
จำนวนนิวรอนในชั้นซ่อน	100	100
จำนวนนิวรอนในชั้นข้อมูลออก	26	26
อัตราการเรียนรู้ (Learning Rate)	0.1	0.7 – 0.1
ค่าความผิดพลาดรวม (ค่าความผิดพลาด^2 x 10^8)	0.000846	0.000060
ทดสอบด้วยข้อมูลที่ใช้ในการเรียนรู้	26 ตัวอักษร (100.00%)	26 ตัวอักษร (100.00%)
ทดสอบด้วยข้อมูลที่ใช้ในการเรียนรู้ที่มีความผิดพลาด 10%	17 ตัวอักษร (65.38%)	23 ตัวอักษร (88.46%)

ตารางที่ 4.8 แสดงผลการทดลองกับตัวอักษร CordiaUPC ขนาด 16 จุดต่อนิ้ว ตัวหนา ด้วยจำนวนนิวรอนในชั้นซ่อน 100 นิวรอน โดยมีการเปลี่ยนค่าอัตราการเรียนรู้

แบบจำลองโครงข่ายที่ใช้ในการเรียนรู้ในการทดลองครั้งนี้ มีจำนวนนิวรอนในชั้นข้อมูลเข้าจำนวน 144 นิวรอน จำนวนนิวรอนในชั้นซ่อนจำนวน 100 นิวรอน จำนวนนิวรอนในข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ออกจำนวน 26 นิวรอน ทำการเรียนรู้ตัวอักษรโดยใช้ฟอนต์ CordiaUPC ขนาด 16 จุดต่อนิ้ว ตัวหนา จำนวน 26 ตัวอักษร มีอัตราการเรียนรู้ (Learning Rate) ที่ปรับเปลี่ยนระหว่างการเรียนรู้อยู่ในช่วง0.1 – 0.7 ผลที่ได้มีค่าความผิดพลาดรวม 0.000060 จึงหยุดการเรียนรู้แล้วนำโครงข่าย (Neural Network) ที่ได้ไปใช้ทดสอบ (Testing) ผลการทดลองปรากฏว่าสามารถรู้จำตัวอักษรได้ทั้งหมด 26 ตัวอักษร คิดเป็น 100.00% เมื่อโครงข่ายสามารถระบุตัวอักษรได้ทั้งหมด จึงได้ลองนำโครงข่ายไปทดสอบกับตัวอักษรที่ใช้ในการเรียนรู้เช่นกันแต่มีความผิดพลาด 10% ผลปรากฏว่าสามารถรู้จำตัวอักษรได้เพียง 23 ตัวอักษร คิดเป็น 88.46% ซึ่งโครงข่ายไม่สามารถระบุตัวอักษร A, O และ V ได้

สำหรับการทดลองที่ผ่านมาทำให้ทราบว่า การปรับอัตราการเรียนรู้นอกจากจะทำให้แบบจำลองโครงข่ายสามารถเรียนรู้ได้รวดเร็วขึ้นแล้ว ยังให้ความถูกต้องมากกว่าเดิมอีกด้วย การทดลองในขั้นต่อไป เป็นการนำแบบจำลองใช้ทดสอบกับตัวอักษรชนิดอื่น ๆ ซึ่งได้แก่ ตัวอักษร CordiaUPC ขนาด 16 จุดต่อนิ้ว ตัวหนาที่มีความผิดพลาด (Noise) ตัวอักษร Noise 10% ตัวอักษร AngsanaUPC ขนาด 16 จุดต่อนิ้ว ตัวอักษร AngsanaUPC ขนาด 16จุดต่อนิ้ว ตัวหนา ตัวอักษร Arial ขนาด 10 จุดต่อนิ้ว ตัวอักษร BrowalliaUPC ขนาด 16 จุดต่อนิ้ว ตัวอักษร DelleniaUPC ขนาด 16 จุดต่อนิ้ว ตัวอักษร EucrosiaUPC ขนาด 16 จุดต่อนิ้ว ตัวอักษร FreesiaUPC ขนาด 16 จุดต่อนิ้ว ตัวอักษร Garamond ขนาด 11 จุดต่อนิ้ว ตัวอักษร JasmineUPC ขนาด 16 จุดต่อนิ้ว และ ตัวอักษร MSSansSerif ขนาด 10 จุดต่อนิ้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอักษรใช้ทดสอบที่ Hidden Node = 50	ตัวอักษรที่สามารถระบุได้																								รวม	
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	จำนวน	ร้อยละ
Noise 10%	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/		/	25	96.15
AngsanaUPC16							/	/	/	/	/	/	/	/	/	/				/	/			/	/	/	15	57.69
AngsanaUPC16B							/	/	/	/	/	/		/		/				/					/	/	11	42.31
Arial10	/		/	/	/	/	/	/	/		/	/	/	/		/			/	/	/	/		/	/	/	20	76.92
BrowalliaUPC16	/		/		/	/	/	/	/		/	/	/	/	/	/		/	/	/	/	/			/		19	73.08
DelleniaUPC16	/		/	/					/	/	/	/		/	/				/	/	/	/				/	14	53.85
EucrosiaUPC16				/	/		/		/	/	/			/		/				/	/						10	38.46
FreesiaUPC16			/	/	/	/		/	/	/		/	/	/		/	/	/		/	/						15	57.69
Garamond11							/		/	/		/	/	/	/				/	/		/					10	38.46
JasmineUPC16	/		/	/	/		/	/	/	/	/	/		/	/	/		/	/	/	/	/			/	/	20	76.92
MSSansSerif10	/		/		/		/	/	/			/		/		/			/	/	/	/			/	/	15	57.69
รวม	6	1	8	7	7	5	9	7	10	7	7	9	5	10	5	8	2	4	7	11	8	7	1	2	5	7	165	57.69

/ หมายถึงตัวอักษรที่โครงข่ายสามารถระบุได้

ตารางที่ 4.9 แสดงผลการทดลองแบบจำลองโครงข่ายซึ่งมีจำนวนนิวรอนในชั้นข้อมูลเข้าจำนวน 144 นิวรอน และจำนวนนิวรอนในชั้นซ่อน จำนวน 50 นิวรอน ซึ่งมีผลการทดลองดังนี้

ผลที่ได้จากแสดงผลการทดลองแบบจำลองโครงข่ายซึ่งมีจำนวนนิวรอนในชั้นข้อมูลเข้าจำนวน 144 นิวรอน และจำนวนนิวรอนในชั้นซ่อน จำนวน 50 นิวรอน ในการทดลองพบว่า ซึ่งมีผลการทดลอง

1. ตัวอักษร CordiaUPC ขนาด 16 จุดต่อนิ้ว ตัวหนา ที่มีความผิดพลาดของตัวอักษร (Noise) 10% โครงข่ายสามารถรู้จำได้เพียง 25 ตัวอักษร คิดเป็น 96.15%

2. ตัวอักษร AngsanaUPC ขนาด 16 จุดต่อนิ้ว โครงข่ายสามารถรู้จำได้เพียง 6 ตัวอักษร คิดเป็น 23.08%

3. ตัวอักษร AngsanaUPC ขนาด 16 จุดต่อนิ้ว ตัวหนา โครงข่ายสามารถรู้จำได้เพียง 11 ตัวอักษร คิดเป็น 42.31%

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. ตัวอักษร Arial ขนาด 10 จุดต่อนิ้ว โครงข่ายสามารถรู้จำได้เพียง 20 ตัวอักษร คิดเป็น 76.92%

5. ตัวอักษร BrowalliaUPC ขนาด 16 จุดต่อนิ้ว โครงข่ายสามารถรู้จำได้เพียง 19 ตัวอักษร คิดเป็น 73.08%

6. ตัวอักษร DelleniaUPC ขนาด 16 จุดต่อนิ้ว โครงข่ายสามารถรู้จำได้เพียง 14 ตัวอักษร คิดเป็น 53.85%

7. ตัวอักษร EucrosiaUPC ขนาด 16 จุดต่อนิ้ว โครงข่ายสามารถรู้จำได้เพียง 10 ตัวอักษร คิดเป็น 38.46%

8. ตัวอักษร FreesiaUPC ขนาด 16 จุดต่อนิ้ว โครงข่ายสามารถรู้จำได้เพียง 15 ตัวอักษร คิดเป็น 57.69%

9. ตัวอักษร Garamond ขนาด 11 จุดต่อนิ้วโครงข่ายสามารถรู้จำได้เพียง 10 ตัวอักษร คิดเป็น 38.46%

10.ตัวอักษร JasmineUPC ขนาด 16 จุดต่อนิ้วโครงข่ายสามารถรู้จำได้เพียง 20 ตัวอักษร คิดเป็น 76.92%

11.ตัวอักษร MSSansSerif ขนาด 10 จุดต่อนิ้วโครงข่ายสามารถรู้จำได้เพียง 15 ตัวอักษร คิดเป็น 57.69%

12.ผลรวมของจำนวนตัวอักษรทั้งหมดที่โครงข่ายสามารถรู้จำได้เพียง 165 ตัวอักษร คิดเป็น 57.69% ของตัวอักษรที่ทำการทดสอบทั้งหมด 286 ตัวอักษร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอักษรใช้ทดสอบที่ Hidden Node = 100	ตัวอักษรที่สามารถระบุได้																								รวม	
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	จำนวน	ร้อยละ
Noise 10%		/	/	/	/	/	/	/	/	/	/	/	/	/		/	/	/	/	/	/		/	/	/	/	23	88.46
AngsanaUPC16		/	/	/		/	/																				5	19.23
AngsanaUPC16B							/		/	/	/	/	/	/		/			/	/	/			/	/		13	50.00
Arial10			/	/	/	/	/	/	/	/	/	/	/			/			/	/	/	/		/	/		18	69.23
BrowalliaUPC16	/					/		/	/	/	/	/		/	/	/		/	/	/	/	/	/		/	/	18	69.23
DelleniaUPC16	/		/	/	/				/	/		/	/			/			/	/	/	/		/	/	/	16	61.54
EucrosiaUPC16				/	/		/		/	/			/			/			/	/	/						10	38.46
FreesiaUPC16			/	/	/	/		/	/	/		/	/	/		/				/	/						13	50.00
Garamond11					/				/			/	/	/	/				/	/			/	/			10	38.46
JasmineUPC16	/			/	/		/	/	/	/		/	/			/		/	/	/	/	/	/			/	17	65.38
MSSansSerif10					/			/	/	/		/	/	/		/			/	/	/	/			/	/	14	53.85
รวม	3	2	5	7	8	5	6	6	10	9	4	9	8	7	2	9	1	3	9	10	9	5	4	5	6	5	157	54.90

/ หมายถึงตัวอักษรที่โครงข่ายสามารถระบุได้

ตารางที่ 4.10 แสดงผลการทดลองแบบจำลองโครงข่ายซึ่งมีจำนวนนิวรอนในชั้นข้อมูลเข้าจำนวน 144 นิวรอน และจำนวนนิวรอนในชั้นซ่อน จำนวน 100 นิวรอน ซึ่งมีผลการทดลองดังนี้

ผลที่ได้จากแสดงผลการทดลองแบบจำลองโครงข่ายซึ่งมีจำนวนนิวรอนในชั้นข้อมูลเข้าจำนวน 144 นิวรอน และจำนวนนิวรอนในชั้นซ่อน จำนวน 100 นิวรอน ในการทดลองพบว่า ซึ่งมีผลการทดลอง

1. ตัวอักษร CordiaUPC ขนาด 16 จุดต่อนิ้ว ตัวหนา ที่มีความผิดพลาดของตัวอักษร (Noise) 10% โครงข่ายสามารถรู้จำได้เพียง 23 ตัวอักษร คิดเป็น 88.46%
2. ตัวอักษร AngsanaUPC ขนาด 16 จุดต่อนิ้ว โครงข่ายสามารถรู้จำได้เพียง 5 ตัวอักษร คิดเป็น 19.23%
3. ตัวอักษร AngsanaUPC ขนาด 16 จุดต่อนิ้ว ตัวหนา โครงข่ายสามารถรู้จำได้เพียง 13 ตัวอักษร คิดเป็น 50.00%

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. ตัวอักษร Arial ขนาด 10 จุดต่อนิ้ว โครงข่ายสามารถรู้จำได้เพียง 18 ตัวอักษร คิดเป็น 69.23%
5. ตัวอักษร BrowalliaUPC ขนาด 16 จุดต่อนิ้ว โครงข่ายสามารถรู้จำได้เพียง 18 ตัวอักษร คิดเป็น 69.23%
6. ตัวอักษร DelleniaUPC ขนาด 16 จุดต่อนิ้ว โครงข่ายสามารถรู้จำได้เพียง 16 ตัวอักษร คิดเป็น 61.54%
7. ตัวอักษร EucrosiaUPC ขนาด 16 จุดต่อนิ้ว โครงข่ายสามารถรู้จำได้เพียง 10 ตัวอักษร คิดเป็น 38.46%
8. ตัวอักษร FreesiaUPC ขนาด 16 จุดต่อนิ้ว โครงข่ายสามารถรู้จำได้เพียง 13 ตัวอักษร คิดเป็น 50.00%
9. ตัวอักษร Garamond ขนาด 11 จุดต่อนิ้วโครงข่ายสามารถรู้จำได้เพียง 10 ตัวอักษร คิดเป็น 38.46%
10. ตัวอักษร JasmineUPC ขนาด 16 จุดต่อนิ้วโครงข่ายสามารถรู้จำได้เพียง 17 ตัวอักษร คิดเป็น 65.38%
11. ตัวอักษร MSSansSerif ขนาด 10 จุดต่อนิ้วโครงข่ายสามารถรู้จำได้เพียง 14 ตัวอักษร คิดเป็น 53.90%
12. ผลรวมของจำนวนตัวอักษรทั้งหมดที่โครงข่ายสามารถรู้จำได้เพียง 157 ตัวอักษร คิดเป็น 54.90% ของตัวอักษรที่ทำการทดสอบทั้งหมด 286 ตัวอักษร

จากผลการทดลองถึงขั้นนี้ ผู้ศึกษามีความเห็นว่าน่าจะมีการศึกษาถึงขนาดของข้อมูลที่ชั้นข้อมูลเข้า (Input Layer) ที่แตกต่างกันซึ่งอาจได้ผลที่แตกต่างกัน เพื่อหาถึงขนาดที่เหมาะสม โดยผู้ศึกษาได้ทำการศึกษาแบบจำลองโครงข่ายที่มีขนาดใหญ่ขึ้นเป็น 20 x 20 Pixel แล้วนำผลที่ได้มาจากการทดลองว่าขนาดของชั้นข้อมูลเข้าเท่าใด ให้ค่าความถูกต้องมากว่ากัน ดังนั้นในขั้นตอนต่อไปผู้ศึกษาได้ทดลองแบบจำลองโครงข่ายเพื่อใช้ในการเรียนรู้โดยมีขนาดของนิวรอนในชั้นข้อมูลเข้าเท่ากับ 400 นิวรอน เรียนรู้ตัวอักษร CordiaUPC ขนาด 26 ตัวต่อนิ้ว ตัวหนา ณ ที่มีนิวรอนในชั้นซ่อน (Hidden Layer) จำนวน 50, 100, 150, 200 และ 250 นิวรอนตามลำดับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อมูล	รายละเอียด
ตัวอักษรที่ใช้	CordiaUPC ขนาด 26 จุดต่อนิ้ว ตัวหนา
จำนวนตัวอักษรที่ใช้ในการเรียนรู้	26
ขนาดข้อมูลของแต่ละตัวอักษรใน Text File	20 x 20
จำนวนนิวรอนในชั้นข้อมูลเข้า	400
จำนวนนิวรอนในชั้นซ่อน	50
จำนวนนิวรอนในชั้นข้อมูลออก	26
อัตราการเรียนรู้ (Learning Rate)	0.7 – 0.1
ค่าความผิดพลาดรวม (ค่าความผิดพลาด^2 x 10^5)	0.000054
ทดสอบด้วยข้อมูลที่ใช้ในการเรียนรู้	26 ตัวอักษร (100.00%)

ตารางที่ 4.11 แสดงผลการทดลองกับตัวอักษร CordiaUPC ขนาด 26 จุดต่อนิ้ว ตัวหนา ด้วยจำนวนนิวรอนในชั้นซ่อน 50 นิวรอน โดยมีการเปลี่ยนค่าอัตราการเรียนรู้

แบบจำลองโครงข่ายที่ใช้ในการเรียนรู้ในการทดลองครั้งนี้ มีจำนวนนิวรอนในชั้นข้อมูลเข้าจำนวน 400 นิวรอน จำนวนนิวรอนในชั้นซ่อนจำนวน 50 นิวรอน จำนวนนิวรอนในข้อมูลออกจำนวน 26 นิวรอน ทำการเรียนรู้ตัวอักษรโดยใช้ฟอนต์ CordiaUPC ขนาด 26 จุดต่อนิ้ว ตัวหนา จำนวน 26 ตัวอักษร มีอัตราการเรียนรู้ (Learning Rate) อยู่ในช่วง 0.7 - 0.1 ผลที่ได้มีค่าความผิดพลาดรวม 0.000054 จึงหยุดการเรียนรู้แล้วนำโครงข่าย (Neural Network) ที่ได้ไปใช้ทดสอบ (Testing) ผลการทดลองปรากฏว่าสามารถรู้จำตัวอักษรได้ทั้งหมด 26 ตัวอักษร คิดเป็น 100.00%

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อมูล	รายละเอียด
ตัวอักษรที่ใช้	CordiaUPC ขนาด 26 จุดต่อนิ้ว ตัวหนา
จำนวนตัวอักษรที่ใช้ในการเรียนรู้	26
ขนาดข้อมูลของแต่ละตัวอักษรใน Text File	20 x 20
จำนวนนิวรอนในชั้นข้อมูลเข้า	400
จำนวนนิวรอนในชั้นซ่อน	100
จำนวนนิวรอนในชั้นข้อมูลออก	26
อัตราการเรียนรู้ (Learning Rate)	0.7 - 0.1
ค่าความผิดพลาดรวม (ค่าความผิดพลาด^2 x 10^5)	0.000312
ทดสอบด้วยข้อมูลที่ใช้ในการเรียนรู้	26 ตัวอักษร (100.00%)

ตารางที่ 4.12 แสดงผลการทดลองกับตัวอักษร CordiaUPC ขนาด 26 จุดต่อนิ้ว ตัวหนา ด้วยจำนวนนิวรอนในชั้นซ่อน 100 นิวรอน โดยมีการเปลี่ยนค่าอัตราการเรียนรู้

แบบจำลองโครงข่ายที่ใช้ในการเรียนรู้ในการทดลองครั้งนี้ มีจำนวนนิวรอนในชั้นข้อมูลเข้าจำนวน 400 นิวรอน จำนวนนิวรอนในชั้นซ่อนจำนวน 100 นิวรอน จำนวนนิวรอนในข้อมูลออกจำนวน 26 นิวรอน ทำการเรียนรู้ตัวอักษรโดยใช้ฟอนต์ CordiaUPC ขนาด 26 จุดต่อนิ้ว ตัวหนา จำนวน 26 ตัวอักษร มีอัตราการเรียนรู้ (Learning Rate) อยู่ในช่วง 0.7 - 0.1 ผลที่ได้มีค่าความผิดพลาดรวม 0.000312 จึงหยุดการเรียนรู้แล้วนำโครงข่าย (Neural Network) ที่ได้ไปใช้ทดสอบ (Testing) ผลการทดลองปรากฏว่าสามารถรู้จำตัวอักษรได้ทั้งหมด 26 ตัวอักษร คิดเป็น 100.00%

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อมูล	รายละเอียด
ตัวอักษรที่ใช้	CordiaUPC ขนาด 26 จุดต่อนิ้ว ตัวหนา
จำนวนตัวอักษรที่ใช้ในการเรียนรู้	26
ขนาดข้อมูลของแต่ละตัวอักษรใน Text File	20 x 20
จำนวนนิวรอนในชั้นข้อมูลเข้า	400
จำนวนนิวรอนในชั้นซ่อน	150
จำนวนนิวรอนในชั้นข้อมูลออก	26
อัตราการเรียนรู้ (Learning Rate)	0.7 - 0.1
ค่าความผิดพลาดรวม (ค่าความผิดพลาด^2 x 10^8)	0.000107
ทดสอบด้วยข้อมูลที่ใช้ในการเรียนรู้	26 ตัวอักษร (100.00%)

ตารางที่ 4.13 แสดงผลการทดลองกับตัวอักษร CordiaUPC ขนาด 26 จุดต่อนิ้ว ตัวหนา ด้วยจำนวนนิวรอนในชั้นซ่อน 150 นิวรอน โดยมีการเปลี่ยนค่าอัตราการเรียนรู้

แบบจำลองโครงข่ายที่ใช้ในการเรียนรู้ในการทดลองครั้งนี้ มีจำนวนนิวรอนในชั้นข้อมูลเข้าจำนวน 400 นิวรอน จำนวนนิวรอนในชั้นซ่อนจำนวน 150 นิวรอน จำนวนนิวรอนในข้อมูลออกจำนวน 26 นิวรอน ทำการเรียนรู้ตัวอักษรโดยใช้ฟอนต์ CordiaUPC ขนาด 26 จุดต่อนิ้ว ตัวหนา จำนวน 26 ตัวอักษร มีอัตราการเรียนรู้ (Learning Rate) อยู่ในช่วง 0.7 - 0.1 ผลที่ได้มีค่าความผิดพลาดรวม 0.000107 จึงหยุดการเรียนรู้แล้วนำโครงข่าย (Neural Network) ที่ได้ไปใช้ทดสอบ (Testing) ผลการทดลองปรากฏว่าสามารถรู้จำตัวอักษรได้ทั้งหมด 26 ตัวอักษร คิดเป็น 100.00%

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อมูล	รายละเอียด
ตัวอักษรที่ใช้	CordiaUPC ขนาด 26 จุดต่อนิ้ว ตัวหนา
จำนวนตัวอักษรที่ใช้ในการเรียนรู้	26
ขนาดข้อมูลของแต่ละตัวอักษรใน Text File	20 x 20
จำนวนนิวรอนในชั้นข้อมูลเข้า	400
จำนวนนิวรอนในชั้นซ่อน	250
จำนวนนิวรอนในชั้นข้อมูลออก	26
อัตราการเรียนรู้ (Learning Rate)	0.7 - 0.1
ค่าความผิดพลาดรวม (ค่าความผิดพลาด^2 x 10^8)	0.002139
ทดสอบด้วยข้อมูลที่ใช้ในการเรียนรู้	26 ตัวอักษร (100.00%)

ตารางที่ 4.14 แสดงผลการทดลองกับตัวอักษร CordiaUPC ขนาด 26 จุดต่อนิ้ว ตัวหนา ด้วยจำนวนนิวรอนในชั้นซ่อน 250 นิวรอน โดยมีการเปลี่ยนค่าอัตราการเรียนรู้

แบบจำลองโครงข่ายที่ใช้ในการเรียนรู้ในการทดลองครั้งนี้ มีจำนวนนิวรอนในชั้นข้อมูลเข้าจำนวน 400 นิวรอน จำนวนนิวรอนในชั้นซ่อนจำนวน 250 นิวรอน จำนวนนิวรอนในข้อมูลออกจำนวน 26 นิวรอน ทำการเรียนรู้ตัวอักษรโดยใช้ฟอนต์ CordiaUPC ขนาด 26 จุดต่อนิ้ว ตัวหนา จำนวน 26 ตัวอักษร มีอัตราการเรียนรู้ (Learning Rate) อยู่ในช่วง 0.7 - 0.1 ผลที่ได้มีค่าความผิดพลาดรวม 0.002139 จึงหยุดการเรียนรู้แล้วนำโครงข่าย (Neural Network) ที่ได้ไปใช้ทดสอบ (Testing) ผลการทดลองปรากฏว่าสามารถรู้จำตัวอักษรได้ทั้งหมด 26 ตัวอักษร คิดเป็น 100.00%

การทดลองในขั้นต่อไป เป็นการนำแบบจำลองใช้ทดสอบกับตัวอักษรชนิดอื่น ๆ ซึ่งได้แก่ ตัวอักษร CordiaUPC ขนาด 26 จุดต่อนิ้ว ตัวหนาที่มีความผิดพลาด (Noise) 5% และ 10% ตัวอักษร AngsanaUPC ขนาด 26 จุดต่อนิ้ว ตัวอักษร Arial ขนาด 17 จุดต่อนิ้ว ตัวอักษร BookAntiqua ขนาด 18 จุดต่อนิ้ว ตัวอักษร BrowalliaUPC ขนาด 26 จุดต่อนิ้ว ตัวอักษร DelleniaUPC ขนาด 26 จุดต่อนิ้ว ตัวอักษร EucrosiaUPC ขนาด 26 จุดต่อนิ้ว ตัวอักษร FreesiaUPC ขนาด 26 จุดต่อนิ้ว ตัวอักษร Garamond ขนาด 20 จุดต่อนิ้ว JasmineUPC ขนาด 26 จุดต่อนิ้ว และ ตัวอักษร MSSansSerif ขนาด 17 จุดต่อนิ้ว (รายละเอียดตัวอักษรปรากฏในภาคผนวก)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอักษรใช้ทดสอบที่ Hidden Node = 50	ตัวอักษรที่สามารถระบุได้																								รวม	
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	จำนวน	ร้อยละ
Noise 10%	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	26	100.00
AngsanaUPC26	/	/		/		/					/	/	/		/	/			/	/				/	/	/	14	53.85
Arial17	/	/	/	/	/	/		/	/	/	/		/	/		/			/		/		/		/	/	18	69.23
BookAntiqua18	/	/				/			/	/	/	/	/					/	/	/		/			/	/	14	53.85
BrowalliaUPC26	/	/	/	/	/	/	/		/	/	/	/	/	/	/	/	/		/	/	/	/			/		21	80.77
DelleniaUPC26	/								/	/	/	/	/	/	/				/	/					/		11	42.31
EucrosiaUPC26	/					/			/	/	/	/	/	/		/			/						/	/	12	46.15
FreesiaUPC26	/	/				/			/	/	/	/	/	/		/			/	/			/		/	/	15	57.69
Garamond20	/		/	/		/	/		/	/	/	/	/	/					/	/				/	/	/	17	65.38
JasmineUPC26	/		/	/		/		/	/	/	/								/	/	/	/			/	/	15	57.69
MSSansSerif17	/	/	/	/	/	/	/		/	/	/	/	/		/	/			/	/	/		/		/	/	20	76.92
รวม	11	7	6	7	4	10	4	3	8	10	11	10	10	9	5	7	3	2	11	9	5	4	4	3	11	9	183	63.99

/ หมายถึงตัวอักษรที่โครงข่ายสามารถระบุได้

ตารางที่ 4.15 แสดงผลการทดลองแบบจำลองโครงข่ายซึ่งมีจำนวนนิวรอนในชั้นข้อมูลเข้าจำนวน 400 นิวรอน และจำนวนนิวรอนในชั้นซ่อน จำนวน 50 นิวรอน ซึ่งมีผลการทดลองดังนี้

ผลที่ได้จากการทดลองโครงข่ายที่มีนิวรอนในชั้นซ่อนจำนวน 50 นิวรอน ในการทดลองพบว่า

1. ตัวอักษร CordiaUPC ขนาด 26 จุดต่อนิ้ว ตัวหนา ที่มีความผิดพลาดของตัวอักษร (Noise) 10% โครงข่ายสามารถรู้จำได้ทั้งหมด 26 ตัวอักษร คิดเป็น 100.00%

2. ตัวอักษร AngsanaUPC ขนาด 26 จุดต่อนิ้ว โครงข่ายสามารถรู้จำได้เพียง 14 ตัวอักษร คิดเป็น 56.85%

3. ตัวอักษร Arial ขนาด 17 จุดต่อนิ้ว โครงข่ายสามารถรู้จำได้เพียง 18 ตัวอักษร คิดเป็น 69.23%

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. ตัวอักษร BookAntiqua ขนาด 18 จุดต่อนิ้ว โครงข่ายสามารถรู้จำได้เพียง 14 ตัวอักษร คิดเป็น 53.85%

5. ตัวอักษร BrowalliaUPC ขนาด 26 จุดต่อนิ้ว โครงข่ายสามารถรู้จำได้เพียง 21 ตัวอักษร คิดเป็น 80.77%

6. ตัวอักษร DelleniaUPC ขนาด 26 จุดต่อนิ้ว โครงข่ายสามารถรู้จำได้เพียง 11 ตัวอักษร คิดเป็น 42.31%

7. ตัวอักษร EucrosiaUPC ขนาด 26 จุดต่อนิ้ว โครงข่ายสามารถรู้จำได้เพียง 12 ตัวอักษร คิดเป็น 46.15%

8. ตัวอักษร FreesiaUPC ขนาด 26 จุดต่อนิ้ว โครงข่ายสามารถรู้จำได้เพียง 15 ตัวอักษร คิดเป็น 57.69%

9. ตัวอักษร Garamond ขนาด 20 จุดต่อนิ้วโครงข่ายสามารถรู้จำได้เพียง 17 ตัวอักษร คิดเป็น 65.38%

10. ตัวอักษร JasmineUPC ขนาด 26 จุดต่อนิ้วโครงข่ายสามารถรู้จำได้เพียง 15 ตัวอักษร คิดเป็น 57.69%

11. ตัวอักษร MSSansSerif ขนาด 17 จุดต่อนิ้วโครงข่ายสามารถรู้จำได้เพียง 20 ตัวอักษร คิดเป็น 76.92%

12. ผลรวมของจำนวนตัวอักษรทั้งหมดที่โครงข่ายสามารถรู้จำได้เพียง 183 ตัวอักษร คิดเป็น 63.99% ของตัวอักษรที่ทำการทดสอบทั้งหมด 286 ตัวอักษร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอักษรใช้ทดสอบที่ Hidden Node = 100	ตัวอักษรที่สามารถระบุได้																								รวม	
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	จำนวน	ร้อยละ
Noise 10%	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	26	100.00
AngsanaUPC26	/	/		/		/				/	/	/			/	/			/	/	/			/	/	/	15	57.69
Arial17	/	/	/	/	/	/	/	/	/	/		/	/		/	/			/		/	/				/	19	73.08
BookAntiqua18	/	/	/						/	/	/	/		/					/	/					/	/	12	46.15
BrowalliaUPC26	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/			/		23	88.46
DelleniaUPC26		/						/	/	/	/	/	/						/				/		/		10	38.46
EucrosiaUPC26	/							/	/	/	/	/	/						/		/			/	/	/	12	46.15
FreesiaUPC26	/	/			/			/	/	/	/	/	/		/				/	/	/			/	/	/	16	61.54
Garamond20	/	/	/				/	/	/		/	/	/						/	/				/	/	/	14	53.85
JasmineUPC26	/	/	/	/			/	/	/	/	/				/				/	/	/	/		/	/	/	17	65.38
MSSansSerif17	/	/	/	/	/	/	/		/	/	/	/	/	/	/	/			/	/	/		/		/	/	22	84.62
รวม	10	10	7	6	4	6	4	6	8	11	10	10	9	8	5	7	4	2	11	8	8	4	3	6	10	9	186	65.03

/ หมายถึงตัวอักษรที่โครงข่ายสามารถระบุได้

ตารางที่ 4.16 แสดงผลการทดลองแบบจำลองโครงข่ายซึ่งมีจำนวนนิวรอนในชั้นข้อมูลเข้าจำนวน 400 นิวรอน และจำนวนนิวรอนในชั้นซ่อน จำนวน 100 นิวรอน ซึ่งมีผลการทดลองดังนี้

ผลที่ได้จากการทดลองโครงข่ายที่มีนิวรอนในชั้นซ่อนจำนวน 100 นิวรอน ในการทดลองพบว่า

1. ตัวอักษร CordiaUPC ขนาด 26 จุดต่อนิ้ว ตัวหนา ที่มีความผิดพลาดของตัวอักษร (Noise) 10% โครงข่ายสามารถรู้จำได้เพียง 26 ตัวอักษร คิดเป็น 100.00%

2. ตัวอักษร AngsanaUPC ขนาด 26 จุดต่อนิ้ว โครงข่ายสามารถรู้จำได้เพียง 15 ตัวอักษร คิดเป็น 57.69%

3. ตัวอักษร Arial ขนาด 17 จุดต่อนิ้ว โครงข่ายสามารถรู้จำได้เพียง 19 ตัวอักษร คิดเป็น 73.08%

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. ตัวอักษร BookAntiqua ขนาด 18 จุดต่อนิ้ว โครงข่ายสามารถรู้จำได้เพียง 12 ตัวอักษร คิดเป็น 46.15%

5. ตัวอักษร BrowalliaUPC ขนาด 26 จุดต่อนิ้ว โครงข่ายสามารถรู้จำได้เพียง 23 ตัวอักษร คิดเป็น 88.46%

6. ตัวอักษร DelleniaUPC ขนาด 26 จุดต่อนิ้ว โครงข่ายสามารถรู้จำได้เพียง 10 ตัวอักษร คิดเป็น 38.46%

7. ตัวอักษร EucrosiaUPC ขนาด 26 จุดต่อนิ้ว โครงข่ายสามารถรู้จำได้เพียง 12 ตัวอักษร คิดเป็น 46.15%

8. ตัวอักษร FreesiaUPC ขนาด 26 จุดต่อนิ้ว โครงข่ายสามารถรู้จำได้เพียง 16 ตัวอักษร คิดเป็น 61.54%

9. ตัวอักษร Garamond ขนาด 20 จุดต่อนิ้วโครงข่ายสามารถรู้จำได้เพียง 14 ตัวอักษร คิดเป็น 53.85%

10. ตัวอักษร JasmineUPC ขนาด 26 จุดต่อนิ้วโครงข่ายสามารถรู้จำได้เพียง 17 ตัวอักษร คิดเป็น 65.38%

11. ตัวอักษร MSSansSerif ขนาด 17 จุดต่อนิ้วโครงข่ายสามารถรู้จำได้เพียง 22 ตัวอักษร คิดเป็น 84.62%

12. ผลรวมของจำนวนตัวอักษรทั้งหมดที่โครงข่ายสามารถรู้จำได้เพียง 186 ตัวอักษร คิดเป็น 65.03% ของตัวอักษรที่ทำการทดสอบทั้งหมด 286 ตัวอักษร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอักษรใช้ทดสอบที่ Hidden Node = 150	ตัวอักษรที่สามารถระบุได้																								รวม	
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	จำนวน	ร้อยละ
Noise 10%	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	26	100.00
AngsanaUPC26	/	/		/	/	/						/			/	/			/	/				/	/	/	13	50.00
Arial17	/	/	/	/	/	/		/	/	/		/	/	/		/			/		/	/	/			/	18	69.23
BookAntiqua18	/	/	/					/		/	/	/		/	/	/			/	/	/				/	/	15	57.69
BrowalliaUPC26	/	/	/		/	/	/	/	/	/	/	/	/	/	/	/	/		/		/	/			/		20	76.92
DelleniaUPC26	/								/	/		/	/		/				/				/	/	/		10	38.46
EucrosiaUPC26	/								/	/	/	/	/	/		/			/						/	/	11	42.31
FreesiaUPC26	/					/			/	/		/	/	/		/					/		/	/	/	/	13	50.00
Garamond20	/		/				/		/	/			/			/			/	/					/		10	38.46
JasmineUPC26	/		/	/				/	/	/		/		/		/			/	/		/		/	/	/	15	57.69
MSSansSerif17	/		/	/	/	/	/					/	/	/	/	/	/	/	/	/	/		/		/	/	19	73.08
รวม	11	5	7	5	5	6	4	5	8	9	4	10	8	8	6	10	3	2	10	7	5	4	5	5	10	8	170	59.44

/ หมายถึงตัวอักษรที่โครงข่ายสามารถระบุได้

ตารางที่ 4.17 แสดงผลการทดลองแบบจำลองโครงข่ายซึ่งมีจำนวนนิวรอนในชั้นข้อมูลเข้าจำนวน 400 นิวรอน และจำนวนนิวรอนในชั้นซ่อน จำนวน 150 นิวรอน ซึ่งมีผลการทดลองดังนี้

ผลที่ได้จากการทดลองโครงข่ายที่มีนิวรอนในชั้นซ่อนจำนวน 150 นิวรอน ในการทดลองพบว่า

1. ตัวอักษร CordiaUPC ขนาด 26 จุดต่อนิ้ว ตัวหนาที่มีความผิดพลาดของตัวอักษร (Noise) 10% โครงข่ายสามารถรู้จำได้เพียง 26 ตัวอักษร คิดเป็น 100.00%

2. ตัวอักษร AngsanaUPC ขนาด 26 จุดต่อนิ้ว โครงข่ายสามารถรู้จำได้เพียง 13 ตัวอักษร คิดเป็น 50.00%

3. ตัวอักษร Arial ขนาด 17 จุดต่อนิ้ว โครงข่ายสามารถรู้จำได้เพียง 18 ตัวอักษร คิดเป็น 69.23%

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. ตัวอักษร BookAntiqua ขนาด 18 จุดต่อนิ้ว โครงข่ายสามารถรู้จำได้เพียง 15 ตัวอักษร คิดเป็น 57.69%

5. ตัวอักษร BrowalliaUPC ขนาด 26 จุดต่อนิ้ว โครงข่ายสามารถรู้จำได้เพียง 20 ตัวอักษร คิดเป็น 76.92%

6. ตัวอักษร DelleniaUPC ขนาด 26 จุดต่อนิ้ว โครงข่ายสามารถรู้จำได้เพียง 10 ตัวอักษร คิดเป็น 38.46%

7. ตัวอักษร EucrosiaUPC ขนาด 26 จุดต่อนิ้ว โครงข่ายสามารถรู้จำได้เพียง 11 ตัวอักษร คิดเป็น 42.31%

8. ตัวอักษร FreesiaUPC ขนาด 26 จุดต่อนิ้ว โครงข่ายสามารถรู้จำได้เพียง 13 ตัวอักษร คิดเป็น 50.00%

9. ตัวอักษร Garamond ขนาด 20 จุดต่อนิ้วโครงข่ายสามารถรู้จำได้เพียง 10 ตัวอักษร คิดเป็น 38.46%

10.ตัวอักษร JasmineUPC ขนาด 26 จุดต่อนิ้วโครงข่ายสามารถรู้จำได้เพียง 15 ตัวอักษร คิดเป็น 57.69%

11.ตัวอักษร MSSansSerif ขนาด 17 จุดต่อนิ้วโครงข่ายสามารถรู้จำได้เพียง 19 ตัวอักษร คิดเป็น 73.08%

12.ผลรวมของจำนวนตัวอักษรทั้งหมดที่โครงข่ายสามารถรู้จำได้เพียง 170 ตัวอักษร คิดเป็น 59.44% ของตัวอักษรที่ทำการทดสอบทั้งหมด 286 ตัวอักษร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอักษรใช้ทดสอบที่ Hidden Node = 200	ตัวอักษรที่สามารถระบุได้																								รวม	
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	จำนวน	ร้อยละ
Noise 10%	/	/	/	/	/	/	/	/		/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	25	96.15
AngsanaUPC26	/	/		/		/					/	/			/	/			/	/				/	/	/	13	50.00
Arial17	/	/	/	/	/	/		/	/	/	/		/	/		/			/		/	/				/	17	65.38
BookAntiqua18	/	/	/							/	/	/	/		/				/	/	/				/	/	13	50.00
BrowalliaUPC26	/	/	/	/	/	/	/			/	/		/		/	/	/		/		/	/			/		17	65.38
DelleniaUPC26	/		/	/					/	/	/	/	/	/	/							/		/	/	/	14	53.85
EucrosiaUPC26	/								/	/	/	/	/	/		/			/		/			/	/	/	13	50.00
FreesiaUPC26	/					/			/	/		/	/	/		/			/	/				/		/	12	46.15
Garamond20	/		/					/	/	/					/				/	/				/	/		10	38.46
JasmineUPC26	/		/	/				/	/	/	/	/									/	/	/	/	/	/	14	53.85
MSSansSerif17	/		/	/	/	/	/			/	/	/	/	/	/	/	/		/	/	/		/		/	/	20	76.92
รวม	11	5	8	7	4	6	3	4	6	10	9	8	8	6	7	7	3	1	9	7	7	5	2	7	9	9	168	58.74

/ หมายถึงตัวอักษรที่โครงข่ายสามารถระบุได้

ตารางที่ 4.18 แสดงผลการทดลองแบบจำลองโครงข่ายซึ่งมีจำนวนนิวรอนในชั้นข้อมูลเข้าจำนวน 400 นิวรอน และจำนวนนิวรอนในชั้นซ่อน จำนวน 200 นิวรอน ซึ่งมีผลการทดลองดังนี้

ผลที่ได้จากการทดลองโครงข่ายที่มีนิวรอนในชั้นซ่อนจำนวน 200 นิวรอน ในการทดลองพบว่า

1. ตัวอักษร CordiaUPC ขนาด 26 จุดต่อนิ้ว ตัวหนาที่มีความผิดพลาดของตัวอักษร (Noise) 10% โครงข่ายสามารถรู้จำได้เพียง 25 ตัวอักษร คิดเป็น 96.15%

2. ตัวอักษร AngsanaUPC ขนาด 26 จุดต่อนิ้ว โครงข่ายสามารถรู้จำได้เพียง 13 ตัวอักษร คิดเป็น 50.00%

3. ตัวอักษร Arial ขนาด 17 จุดต่อนิ้ว โครงข่ายสามารถรู้จำได้เพียง 17 ตัวอักษร คิดเป็น 65.38%

4. ตัวอักษร BookAntiqua ขนาด 18 จุดต่อนิ้ว โครงข่ายสามารถรู้จำได้เพียง 13 ตัวอักษร คิดเป็น 50.00%

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. ตัวอักษร BrowalliaUPC ขนาด 26 จุดต่อนิ้ว โครงข่ายสามารถรู้จำได้เพียง 17 ตัวอักษร คิดเป็น 65.38%

6. ตัวอักษร DelleniaUPC ขนาด 26 จุดต่อนิ้ว โครงข่ายสามารถรู้จำได้เพียง 14 ตัวอักษร คิดเป็น 53.85%

7. ตัวอักษร EucrosiaUPC ขนาด 26 จุดต่อนิ้ว โครงข่ายสามารถรู้จำได้เพียง 13 ตัวอักษร คิดเป็น 50.00%

8. ตัวอักษร FreesiaUPC ขนาด 26 จุดต่อนิ้ว โครงข่ายสามารถรู้จำได้เพียง 12 ตัวอักษร คิดเป็น 46.15%

9. ตัวอักษร Garamond ขนาด 20 จุดต่อนิ้วโครงข่ายสามารถรู้จำได้เพียง 10 ตัวอักษร คิดเป็น 38.46%

10.ตัวอักษร JasmineUPC ขนาด 26 จุดต่อนิ้วโครงข่ายสามารถรู้จำได้เพียง 14 ตัวอักษร คิดเป็น 53.85%

11.ตัวอักษร MSSansSerif ขนาด 17 จุดต่อนิ้วโครงข่ายสามารถรู้จำได้เพียง 20 ตัวอักษร คิดเป็น 76.92%

12.ผลรวมของจำนวนตัวอักษรทั้งหมดที่โครงข่ายสามารถรู้จำได้เพียง 168 ตัวอักษร คิดเป็น 58.74% ของตัวอักษรที่ทำการทดสอบทั้งหมด 286 ตัวอักษร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอักษรใช้ทดสอบที่ Hidden Node = 250	ตัวอักษรที่สามารถระบุได้																									รวม	
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	จำนวน	ร้อยละ
Noise 10%	/	/	/	/	/	/	/	/		/	/	/	/	/	/	/	/	/	/		/	/	/	/		/	23	88.46
AngsanaUPC26	/			/		/		/				/	/		/	/			/	/				/		/	12	46.15
Arial17	/	/	/	/	/	/		/	/	/	/		/	/		/			/		/	/				/	17	65.38
BookAntiqua18	/	/	/			/				/	/	/	/		/	/			/	/	/		/			/	15	57.69
BrowalliaUPC26	/	/	/	/	/	/	/	/		/	/	/	/	/	/	/	/		/	/	/	/			/		21	80.77
DelleniaUPC26	/	/							/	/		/	/		/									/	/		9	34.62
EucrosiaUPC26	/					/				/		/	/	/					/							/	8	30.77
FreesiaUPC26	/					/			/	/		/	/			/			/	/						/	10	38.46
Garamond20									/	/		/	/	/	/				/	/				/		/	10	38.46
JasmineUPC26	/	/	/	/		/		/		/		/							/	/	/	/			/	/	14	53.85
MSSansSerif17	/		/	/	/	/	/			/	/	/	/	/	/	/	/		/	/			/		/	/	19	73.08
รวม	10	6	6	6	4	9	3	5	4	10	5	10	10	6	7	7	3	1	10	7	5	4	3	4	4	9	158	55.24

/ หมายถึงตัวอักษรที่โครงข่ายสามารถระบุได้

ตารางที่ 4.19 แสดงผลการทดลองแบบจำลองโครงข่ายซึ่งมีจำนวนนิวรอนในชั้นข้อมูลเข้าจำนวน 400 นิวรอน และจำนวนนิวรอนในชั้นซ่อน จำนวน 250 นิวรอน ซึ่งมีผลการทดลองดังนี้

ผลที่ได้จากการทดลองโครงข่ายที่มีนิวรอนในชั้นซ่อนจำนวน 250 นิวรอน ในการทดลองพบว่า

1. ตัวอักษร CordiaUPC ขนาด 26 จุดต่อนิ้ว ตัวหนาที่มีความผิดพลาดของตัวอักษร (Noise) 10% โครงข่ายสามารถรู้จำได้เพียง 23 ตัวอักษร คิดเป็น 88.46%

2. ตัวอักษร AngsanaUPC ขนาด 26 จุดต่อนิ้ว โครงข่ายสามารถรู้จำได้เพียง 12 ตัวอักษร คิดเป็น 43.15%

3. ตัวอักษร Arial ขนาด 17 จุดต่อนิ้ว โครงข่ายสามารถรู้จำได้เพียง 17 ตัวอักษร คิดเป็น 65.38%

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. ตัวอักษร BookAntiqua ขนาด 18 จุดต่อนิ้ว โครงข่ายสามารถรู้จำได้เพียง 15 ตัวอักษร คิดเป็น 57.69%

5. ตัวอักษร BrowalliaUPC ขนาด 26 จุดต่อนิ้ว โครงข่ายสามารถรู้จำได้เพียง 21 ตัวอักษร คิดเป็น 80.77%

6. ตัวอักษร DelleniaUPC ขนาด 26 จุดต่อนิ้ว โครงข่ายสามารถรู้จำได้เพียง 9 ตัวอักษร คิดเป็น 34.62%

7. ตัวอักษร EucrosiaUPC ขนาด 26 จุดต่อนิ้ว โครงข่ายสามารถรู้จำได้เพียง 8 ตัวอักษร คิดเป็น 30.77%

8. ตัวอักษร FreesiaUPC ขนาด 26 จุดต่อนิ้ว โครงข่ายสามารถรู้จำได้เพียง 10 ตัวอักษร คิดเป็น 38.46%

9. ตัวอักษร Garamond ขนาด 20 จุดต่อนิ้วโครงข่ายสามารถรู้จำได้เพียง 10 ตัวอักษร คิดเป็น 38.46%

10. ตัวอักษร JasmineUPC ขนาด 26 จุดต่อนิ้วโครงข่ายสามารถรู้จำได้เพียง 14 ตัวอักษร คิดเป็น 53.85%

11. ตัวอักษร MSSansSerif ขนาด 17 จุดต่อนิ้วโครงข่ายสามารถรู้จำได้เพียง 19 ตัวอักษร คิดเป็น 73.08%

12. ผลรวมของจำนวนตัวอักษรทั้งหมดที่โครงข่ายสามารถรู้จำได้เพียง 158 ตัวอักษร คิดเป็น 55.24% ของตัวอักษรที่ทำการทดสอบทั้งหมด 286 ตัวอักษร

จากการทดลองการใช้แบบจำลองโครงข่ายที่มีจำนวนนิวรอนของชั้นข้อมูลเข้า (Input Layer) ระหว่าง 144 นิวรอน กับ 400 นิวรอน ผลปรากฏว่าแบบจำลองโครงข่ายที่มีจำนวนนิวรอนของชั้นข้อมูลเข้า (Input Later) จำนวน 400 นิวรอนและมีจำนวนนิวรอนของชั้นซ่อน (Hidden Layer) มีประสิทธิภาพสูงสุดในการระบุตัวอักษร มีความถูกต้องโดยรวม 65.03% และจากการสังเกตตัวอักษรที่ใช้ทดสอบพบว่า ตัวอักษรที่ใช้มีลักษณะที่คล้ายคลึงกัน 2 กลุ่มได้แก่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. ตัวอักษร CordiaUPC ตัวหนา ( ) BrowalliaUPC ( ) MS Sanserif ( )

   และ FreesiaUPC ( )

2. ตัวอักษร AngsanaUPC ( ) JasmineUPC ( ) Garamond ( )

   BookAntiqua ( ) DelleniaUPC ( ) และ EucrosiaUPC ()

ดังนั้นในขั้นตอนต่อไป ผู้ศึกษาจึงได้นำตัวแทนของตัวอักษรของทั้ง 2 เข้าไปให้แบบจำลองโครงข่ายในการเรียนรู้ โดยจะทำการทดลองกับแบบจำลองโครงข่ายที่มีจำนวนนิวรอนในชั้นซ่อน (Hidden Layer) ขนาด 50, 100 และ 150 เท่านั้นทั้งนี้เพราะจากการทดลองครั้งก่อนในการเรียนรู้และระบุตัวอักษรของแบบจำลองโครงข่ายที่มีจำนวนนิวรอนในชั้นซ่อนในระดับนี้ ให้ผลความถูกต้องในระดับที่สูง 3 อันดับแรก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อมูล	รายละเอียด
ตัวอักษรที่ใช้	CordiaUPC ขนาด 26 จุดต่อนิ้ว ตัวหนา และ AngsanaUPC ขนาด 26 จุดต่อนิ้ว
จำนวนตัวอักษรที่ใช้ในการเรียนรู้	52
ขนาดข้อมูลของแต่ละตัวอักษรใน Text File	20 x 20
จำนวนนิวรอนในชั้นข้อมูลเข้า	400
จำนวนนิวรอนในชั้นซ่อน	50
จำนวนนิวรอนในชั้นข้อมูลออก	26
อัตราการเรียนรู้ (Learning Rate)	0.7 – 0.1
ค่าความผิดพลาดรวม (ค่าความผิดพลาด^2 x 10^6)	0.001925
ทดสอบด้วยข้อมูลที่ใช้ในการเรียนรู้	52 ตัวอักษร (100.00%)

ตารางที่ 4.20 แสดงผลการทดลองกับตัวอักษร CordiaUPC ขนาด 26 จุดต่อนิ้ว ตัวหนา และ AngsanaUPC ขนาด 26 จุดต่อนิ้ว ด้วยจำนวนนิวรอนในชั้นซ่อน 50 นิวรอน โดยมีการเปลี่ยนค่าอัตราการเรียนรู้

แบบจำลองโครงข่ายที่ใช้ในการเรียนรู้ในการทดลองครั้งนี้ มีจำนวนนิวรอนในชั้นข้อมูลเข้าจำนวน 400 นิวรอน จำนวนนิวรอนในชั้นซ่อนจำนวน 50 นิวรอน จำนวนนิวรอนในข้อมูลออกจำนวน 26 นิวรอน ทำการเรียนรู้ตัวอักษรโดยใช้ฟอนต์ CordiaUPC ขนาด 26 จุดต่อนิ้ว ตัวหนา จำนวน 26 ตัวอักษร และ AngsanaUPC ขนาด 26 จุดต่อนิ้ว จำนวน 26 ตัวอักษร อัตราการเรียนรู้ (Learning Rate) อยู่ในช่วง 0.7 - 0.1 ผลที่ได้มีค่าความผิดพลาดรวม 0.001925 จึงหยุดการเรียนรู้แล้วนำโครงข่าย (Neural Network) ที่ได้ไปใช้ทดสอบ (Testing) ผลการทดลองปรากฏว่าสามารถรู้จำตัวอักษรได้ทั้งหมด 52 ตัวอักษร คิดเป็น 100.00%

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อมูล	รายละเอียด
ตัวอักษรที่ใช้	CordiaUPC ขนาด 26 จุดต่อนิ้ว ตัวหนา และ AngsanaUPC ขนาด 26 จุดต่อนิ้ว
จำนวนตัวอักษรที่ใช้ในการเรียนรู้	52
ขนาดข้อมูลของแต่ละตัวอักษรใน Text File	20 x 20
จำนวนนิวรอนในชั้นข้อมูลเข้า	400
จำนวนนิวรอนในชั้นซ่อน	100
จำนวนนิวรอนในชั้นข้อมูลออก	26
อัตราการเรียนรู้ (Learning Rate)	0.7 – 0.1
ค่าความผิดพลาดรวม (ค่าความผิดพลาด^2 x 10^8)	0.000274
ทดสอบด้วยข้อมูลที่ใช้ในการเรียนรู้	52 ตัวอักษร (100.00%)

ตารางที่ 4.21 แสดงผลการทดลองกับตัวอักษร CordiaUPC ขนาด 26 จุดต่อนิ้ว ตัวหนา และ AngsanaUPC ขนาด 26 จุดต่อนิ้ว ด้วยจำนวนนิวรอนในชั้นซ่อน 100 นิวรอน โดยมีการเปลี่ยนค่าอัตราการเรียนรู้

แบบจำลองโครงข่ายที่ใช้ในการเรียนรู้ในการทดลองครั้งนี้ มีจำนวนนิวรอนในชั้นข้อมูลเข้าจำนวน 400 นิวรอน จำนวนนิวรอนในชั้นซ่อนจำนวน 100 นิวรอน จำนวนนิวรอนในข้อมูลออกจำนวน 26 นิวรอน ทำการเรียนรู้ตัวอักษรโดยใช้ฟอนต์ CordiaUPC ขนาด 26 จุดต่อนิ้ว ตัวหนา จำนวน 26 ตัวอักษร และ AngsanaUPC ขนาด 26 จุดต่อนิ้ว จำนวน 26 ตัวอักษร อัตราการเรียนรู้ (Learning Rate) อยู่ในช่วง 0.7 - 0.1 ผลที่ได้มีค่าความผิดพลาดรวม 0.000274 จึงหยุดการเรียนรู้แล้วนำโครงข่าย (Neural Network) ที่ได้ไปใช้ทดสอบ (Testing) ผลการทดลองปรากฏว่าสามารถรู้จำตัวอักษรได้ทั้งหมด 52 ตัวอักษร คิดเป็น 100.00%

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อมูล	รายละเอียด
ตัวอักษรที่ใช้	CordiaUPC ขนาด 26 จุดต่อนิ้ว ตัวหนา และ AngsanaUPC ขนาด 26 จุดต่อนิ้ว
จำนวนตัวอักษรที่ใช้ในการเรียนรู้	52
ขนาดข้อมูลของแต่ละตัวอักษรใน Text File	20 x 20
จำนวนนิวรอนในชั้นข้อมูลเข้า	400
จำนวนนิวรอนในชั้นซ่อน	150
จำนวนนิวรอนในชั้นข้อมูลออก	26
อัตราการเรียนรู้ (Learning Rate)	0.7 – 0.1
ค่าความผิดพลาดรวม (ค่าความผิดพลาด^2 x 10^8)	0.000460
ทดสอบด้วยข้อมูลที่ใช้ในการเรียนรู้	52 ตัวอักษร (100.00%)

ตารางที่ 4.22 แสดงผลการทดลองกับตัวอักษร CordiaUPC ขนาด 26 จุดต่อนิ้ว ตัวหนา และ AngsanaUPC ขนาด 26 จุดต่อนิ้ว ด้วยจำนวนนิวรอนในชั้นซ่อน 150 นิวรอน โดยมีการเปลี่ยนค่าอัตราการเรียนรู้

แบบจำลองโครงข่ายที่ใช้ในการเรียนรู้ในการทดลองครั้งนี้ มีจำนวนนิวรอนในชั้นข้อมูลเข้าจำนวน 400 นิวรอน จำนวนนิวรอนในชั้นซ่อนจำนวน 150 นิวรอน จำนวนนิวรอนในข้อมูลออกจำนวน 26 นิวรอน ทำการเรียนรู้ตัวอักษรโดยใช้ฟอนต์ CordiaUPC ขนาด 26 จุดต่อนิ้ว ตัวหนา จำนวน 26 ตัวอักษร และ AngsanaUPC ขนาด 26 จุดต่อนิ้ว จำนวน 26 ตัวอักษร อัตราการเรียนรู้ (Learning Rate) อยู่ในช่วง 0.7 - 0.1 ผลที่ได้มีค่าความผิดพลาดรวม 0.000460 จึงหยุดการเรียนรู้แล้วนำโครงข่าย (Neural Network) ที่ได้ไปใช้ทดสอบ (Testing) ผลการทดลองปรากฏว่าสามารถรู้จำตัวอักษรได้ทั้งหมด 52 ตัวอักษร คิดเป็น 100.00%

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอักษรใช้ทดสอบที่ Hidden Node = 50	ตัวอักษรที่สามารถระบุได้																								รวม	
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	จำนวน	ร้อยละ
CordiaUPC26B Noise 10%	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	26	100.00
CordiaUPC26B	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	26	100.00
AngsanaUPC26	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	26	100.00
Arial17	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	21	80.77
BookAntiqua18	/	/	/			/			/	/	/				/	/	/		/	/	/		/		/	/	16	61.54
BrowalliaUPC26	/	/		/	/	/	/	/	/		/	/	/	/		/	/		/	/			/		/		18	69.23
DelleniaUPC26	/		/	/				/		/	/	/	/						/	/			/	/	/		13	50.00
EucrosiaUPC26	/		/			/		/	/	/	/	/	/		/				/					/	/	/	14	53.85
FreesiaUPC26	/	/	/			/			/	/	/	/	/	/		/	/		/		/		/	/	/	/	18	69.23
Garamond20	/	/		/		/			/		/		/	/					/					/	/	/	12	46.15
JasmineUPC26	/		/	/				/	/	/	/	/	/	/		/			/	/	/	/		/		/	17	65.38
MSSansSerif17	/	/	/	/	/	/	/		/		/	/	/	/	/	/	/		/	/	/		/		/	/	21	80.77
รวม	12	9	10	9	6	9	7	6	11	8	12	10	11	11	5	10	7	4	12	9	8	5	8	9	10	10	228	73.08

/ หมายถึงตัวอักษรที่โครงข่ายสามารถระบุได้

ตารางที่ 4.23 แสดงผลการทดลองแบบจำลองโครงข่ายซึ่งมีจำนวนนิวรอนในชั้นข้อมูลเข้าจำนวน 400 นิวรอน และจำนวนนิวรอนในชั้นซ่อน จำนวน 50 นิวรอน กับตัวอักษร Cordia และ AngsanaUPC ซึ่งมีผลการทดลองดังนี้

ผลที่ได้จากการทดลองโครงข่ายที่มีนิวรอนในชั้นซ่อนจำนวน 50 นิวรอน ในการทดลองพบว่า

1. ตัวอักษร CordiaUPC ขนาด 26 จุดต่อนิ้ว ตัวหนาที่มีความผิดพลาดของตัวอักษร (Noise) 10% โครงข่ายสามารถรู้จำได้ทั้งหมด 26 ตัวอักษร คิดเป็น 100.00%

2. ตัวอักษร CordiaUPC ขนาด 26 จุดต่อนิ้ว ตัวหนา โครงข่ายสามารถรู้จำได้ทั้งหมด 26 ตัวอักษร คิดเป็น 100.00%

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. ตัวอักษร AngsanaUPC ขนาด 26 จุดต่อนิ้ว โครงข่ายสามารถรู้จำได้ทั้งหมด 26 ตัวอักษร คิดเป็น 100.00%

4. ตัวอักษร Arial ขนาด 17 จุดต่อนิ้ว โครงข่ายสามารถรู้จำได้เพียง 21 ตัวอักษร คิดเป็น 80.77%

5. ตัวอักษร BookAntiqua ขนาด 18 จุดต่อนิ้ว โครงข่ายสามารถรู้จำได้เพียง 16 ตัวอักษร คิดเป็น 61.54%

6. ตัวอักษร BrowalliaUPC ขนาด 26 จุดต่อนิ้ว โครงข่ายสามารถรู้จำได้เพียง 18 ตัวอักษร คิดเป็น 69.23%

7. ตัวอักษร DelleniaUPC ขนาด 26 จุดต่อนิ้ว โครงข่ายสามารถรู้จำได้เพียง 13 ตัวอักษร คิดเป็น 50.00%

8. ตัวอักษร EucrosiaUPC ขนาด 26 จุดต่อนิ้ว โครงข่ายสามารถรู้จำได้เพียง 14 ตัวอักษร คิดเป็น 53.85%

9. ตัวอักษร FreesiaUPC ขนาด 26 จุดต่อนิ้ว โครงข่ายสามารถรู้จำได้เพียง 18 ตัวอักษร คิดเป็น 69.23%

10. ตัวอักษร Garamond ขนาด 20 จุดต่อนิ้วโครงข่ายสามารถรู้จำได้เพียง 12 ตัวอักษร คิดเป็น 46.15%

11. ตัวอักษร JasmineUPC ขนาด 26 จุดต่อนิ้วโครงข่ายสามารถรู้จำได้เพียง 17 ตัวอักษร คิดเป็น 65.38%

12. ตัวอักษร MSSansSerif ขนาด 17 จุดต่อนิ้วโครงข่ายสามารถรู้จำได้เพียง 21 ตัวอักษร คิดเป็น 80.77%

13. ผลรวมของจำนวนตัวอักษรทั้งหมดที่โครงข่ายสามารถรู้จำได้เพียง 228 ตัวอักษร คิดเป็น 73.08% ของตัวอักษรที่ทำการทดสอบทั้งหมด 312 ตัวอักษร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอักษรใช้ทดสอบที่ Hidden Node = 100	ตัวอักษรที่สามารถระบุได้																							รวม	
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	จำนวน	ร้อยละ
CordiaUPC26B Noise 10%	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	26	100.00
CordiaUPC26B	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	26	100.00
AngsanaUPC26	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	26	100.00
Arial17	/	/	/	/	/		/	/	/			/	/	/		/		/	/		/	/				/	17	65.38
BookAntiqua18	/	/	/			/		/		/	/	/			/			/	/	/			/		/	/	15	57.69
BrowalliaUPC26	/	/		/	/	/	/	/	/	/	/	/	/	/	/	/	/		/	/					/		19	73.08
DelleniaUPC26	/		/	/					/	/	/	/	/	/					/	/			/	/	/	/	15	57.69
EucrosiaUPC26	/		/						/	/	/		/	/				/	/				/		/	/	12	46.15
FreesiaUPC26	/	/	/			/			/		/		/	/		/	/		/						/	/	/	14	53.85
Garamond20	/	/	/			/		/		/	/		/			/	/		/		/				/	/		14	53.85
JasmineUPC26	/	/	/	/				/	/	/	/	/		/					/	/	/				/	/	/	16	61.54
MSSansSerif17	/	/	/	/	/	/	/	/			/	/	/	/	/	/	/		/	/	/		/			/	/	21	80.77
รวม	12	10	11	8	6	8	6	9	9	9	11	9	10	10	6	8	7	6	12	8	7	4	7	7	11	10	221	70.83

/ หมายถึงตัวอักษรที่โครงข่ายสามารถระบุได้

ตารางที่ 4.24 แสดงผลการทดลองแบบจำลองโครงข่ายซึ่งมีจำนวนนิวรอนในชั้นข้อมูลเข้า จำนวน 400 นิวรอน และจำนวนนิวรอนในชั้นซ่อน จำนวน 100 นิวรอน กับตัวอักษร Cordia และ AngsanaUPC ซึ่งมีผลการทดลองดังนี้

ผลที่ได้จากการทดลองโครงข่ายที่มีนิวรอนในชั้นซ่อนจำนวน 100 นิวรอน ในการทดลองพบว่า

1. ตัวอักษร CordiaUPC ขนาด 26 จุดต่อนิ้ว ตัวหนาที่มีความผิดพลาดของตัวอักษร (Noise) 10% โครงข่ายสามารถรู้จำได้ทั้งหมด 26 ตัวอักษร คิดเป็น 100.00%

2. ตัวอักษร CordiaUPC ขนาด 26 จุดต่อนิ้ว ตัวหนา โครงข่ายสามารถรู้จำได้ทั้งหมด 26 ตัวอักษร คิดเป็น 100.00%

3. ตัวอักษร AngsanaUPC ขนาด 26 จุดต่อนิ้ว โครงข่ายสามารถรู้จำได้ทั้งหมด 26 ตัวอักษร คิดเป็น 100.00%

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. ตัวอักษร Arial ขนาด 17 จุดต่อนิ้ว โครงข่ายสามารถรู้จำได้เพียง 17 ตัวอักษร คิดเป็น 65.38%

5. ตัวอักษร BookAntiqua ขนาด 18 จุดต่อนิ้ว โครงข่ายสามารถรู้จำได้เพียง 15 ตัวอักษร คิดเป็น 57.69%

6. ตัวอักษร BrowalliaUPC ขนาด 26 จุดต่อนิ้ว โครงข่ายสามารถรู้จำได้เพียง 19 ตัวอักษร คิดเป็น 73.08%

7. ตัวอักษร DelleniaUPC ขนาด 26 จุดต่อนิ้ว โครงข่ายสามารถรู้จำได้เพียง 15 ตัวอักษร คิดเป็น 57.69%

8. ตัวอักษร EucrosiaUPC ขนาด 26 จุดต่อนิ้ว โครงข่ายสามารถรู้จำได้เพียง 12 ตัวอักษร คิดเป็น 46.15%

9. ตัวอักษร FreesiaUPC ขนาด 26 จุดต่อนิ้ว โครงข่ายสามารถรู้จำได้เพียง 14 ตัวอักษร คิดเป็น 53.85%

10. ตัวอักษร Garamond ขนาด 20 จุดต่อนิ้วโครงข่ายสามารถรู้จำได้เพียง 14 ตัวอักษร คิดเป็น 53.85%

11. ตัวอักษร JasmineUPC ขนาด 26 จุดต่อนิ้วโครงข่ายสามารถรู้จำได้เพียง 16 ตัวอักษร คิดเป็น 61.54%

12. ตัวอักษร MSSansSerif ขนาด 17 จุดต่อนิ้วโครงข่ายสามารถรู้จำได้เพียง 21 ตัวอักษร คิดเป็น 80.77%

13. ผลรวมของจำนวนตัวอักษรทั้งหมดที่โครงข่ายสามารถรู้จำได้เพียง 221 ตัวอักษร คิดเป็น 70.83% ของตัวอักษรที่ทำการทดสอบทั้งหมด 312 ตัวอักษร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอักษรใช้ทดสอบที่ Hidden Node = 150	ตัวอักษรที่สามารถระบุได้																							รวม	
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	จำนวน	ร้อยละ
CordiaUPC26B Noise 10%	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	26	100.00
CordiaUPC26B	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	26	100.00
AngsanaUPC26	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	26	100.00
Arial17	/	/	/	/	/	/	/	/	/	/	/	/	/	/		/			/	/	/	/				/	20	76.92
BookAntiqua18	/	/	/			/				/	/			/	/	/			/	/	/				/	/	14	53.85
BrowalliaUPC26	/	/	/	/	/	/	/			/	/	/	/	/	/	/			/	/	/				/		18	69.23
DelleniaUPC26	/		/	/					/		/	/	/	/	/			/	/	/				/		/	14	53.85
EucrosiaUPC26	/		/						/	/	/	/	/	/				/	/		/					/	/	13	50.00
FreesiaUPC26	/	/	/			/			/	/	/		/	/		/	/		/		/			/	/	/	/	17	65.38
Garamond20		/	/			/				/	/	/	/	/		/			/		/				/	/	/	14	53.85
JasmineUPC26	/	/	/	/				/	/	/	/	/	/	/		/			/		/	/			/	/	/	18	69.23
MSSansSerif17	/		/	/	/	/	/			/	/	/	/	/	/	/			/	/	/	/	/	/	/	/	/	21	80.77
รวม	11	9	12	8	6	9	6	5	8	7	12	11	11	12	8	9	7	5	12	8	11	6	6	7	11	10	227	72.76

/ หมายถึงตัวอักษรที่โครงข่ายสามารถระบุได้

ตารางที่ 4.25 แสดงผลการทดลองแบบจำลองโครงข่ายซึ่งมีจำนวนนิวรอนในชั้นข้อมูลเข้า จำนวน 400 นิวรอน และจำนวนนิวรอนในชั้นซ่อน จำนวน 150 นิวรอน กับตัวอักษร Cordia และ AngsanaUPC ซึ่งมีผลการทดลองดังนี้

ผลที่ได้จากการทดลองโครงข่ายที่มีนิวรอนในชั้นซ่อนจำนวน 150 นิวรอน ในการทดลองพบว่า

1. ตัวอักษร CordiaUPC ขนาด 26 จุดต่อนิ้ว ตัวหนาที่มีความผิดพลาดของตัวอักษร (Noise) 10% โครงข่ายสามารถรู้จำได้ทั้งหมด 26 ตัวอักษร คิดเป็น 100.00%

2. ตัวอักษร CordiaUPC ขนาด 26 จุดต่อนิ้ว ตัวหนา โครงข่ายสามารถรู้จำได้ทั้งหมด 26 ตัวอักษร คิดเป็น 100.00%

3. ตัวอักษร AngsanaUPC ขนาด 26 จุดต่อนิ้ว โครงข่ายสามารถรู้จำได้ทั้งหมด 26 ตัวอักษร คิดเป็น 100.00%

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. ตัวอักษร Arial ขนาด 17 จุดต่อนิ้ว โครงข่ายสามารถรู้จำได้เพียง 20 ตัวอักษร คิดเป็น 76.92%

5. ตัวอักษร BookAntiqua ขนาด 18 จุดต่อนิ้ว โครงข่ายสามารถรู้จำได้เพียง 14 ตัวอักษร คิดเป็น 53.85%

6. ตัวอักษร BrowalliaUPC ขนาด 26 จุดต่อนิ้ว โครงข่ายสามารถรู้จำได้เพียง 18 ตัวอักษร คิดเป็น 69.23%

7. ตัวอักษร DelleniaUPC ขนาด 26 จุดต่อนิ้ว โครงข่ายสามารถรู้จำได้เพียง 14 ตัวอักษร คิดเป็น 53.85%

8. ตัวอักษร EucrosiaUPC ขนาด 26 จุดต่อนิ้ว โครงข่ายสามารถรู้จำได้เพียง 13 ตัวอักษร คิดเป็น 50.00%

9. ตัวอักษร FreesiaUPC ขนาด 26 จุดต่อนิ้ว โครงข่ายสามารถรู้จำได้เพียง 17 ตัวอักษร คิดเป็น 65.38%

10. ตัวอักษร Garamond ขนาด 20 จุดต่อนิ้วโครงข่ายสามารถรู้จำได้เพียง 14 ตัวอักษร คิดเป็น 53.85%

11. ตัวอักษร JasmineUPC ขนาด 26 จุดต่อนิ้วโครงข่ายสามารถรู้จำได้เพียง 18 ตัวอักษร คิดเป็น 69.23%

12. ตัวอักษร MSSansSerif ขนาด 17 จุดต่อนิ้วโครงข่ายสามารถรู้จำได้เพียง 21 ตัวอักษร คิดเป็น 80.77%

13. ผลรวมของจำนวนตัวอักษรทั้งหมดที่โครงข่ายสามารถรู้จำได้เพียง 227 ตัวอักษร คิดเป็น 72.76% ของตัวอักษรที่ทำการทดสอบทั้งหมด 312 ตัวอักษร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

สรุปผลการทดลอง

5.1 สรุปผลการทดลอง

จากการศึกษาถึงสถาปัตยกรรมโครงข่ายประสาทเทียม แบบแพร่ย้อนกลับ (Backpropagation Network) ว่าสามารถนำมาใช้ในการเรียนรู้และจดจำลักษณะตัวอักษรได้นั้นผู้ศึกษาจึงได้พัฒนาแบบจำลองโครงข่ายประสาทเทียมแบบแพร่ย้อนกลับมาทดลอง โดยแบบจำลองที่ให้มีการกำหนดค่าน้ำหนักเริ่มต้นแบบสุ่ม (Random) และการกำหนดอัตราการเรียนรู้ ซึ่งจากการทดลอง สามารถสรุปได้ดังตารางที่ 5.1

ข้อมูล	รายละเอียด			
จำนวนนิวรอนในชั้นซ่อน	50		100	
อัตราการเรียนรู้ (Learning Rate)	0.1	0.7 - 0.1	0.1	0.7 – 0.1
ค่าความผิดพลาดรวม (ค่าความผิดพลาด^2 x 10^8)	0.001715	0.000086	0.000846	0.00006
ทดสอบด้วยข้อมูลที่ใช้ในการเรียนรู้	26 (100.00)	26 (100.00)	26 (100.00)	26 (100.00)
ทดสอบด้วยข้อมูลที่ใช้ในการเรียนรู้ที่มีความผิดพลาด 10%	23 (88.46)	24 (92.31)	17 (65.38)	23 (88.46)

ตารางที่ 5.1 แสดงผลการทดลองการเปลี่ยนแปลงอัตราการเรียนรู้

จากตารางที่ 5.1 แสดงให้เห็นว่าการอัตราการเรียนรู้ (Learning Rate) จะมีผลต่อการเรียนรู้ของโครงข่าย จากการทดลองพบว่าหากมีการระบุอัตราการเรียนรู้แบบคงที่ แบบจำลองโครงข่ายจะมีประสิทธิภาพในการเรียนรู้ได้น้อยกว่าการที่มีการเปลี่ยนแปลงอัตราการเรียนรู้ (Learning Rate) ระหว่างที่โครงข่ายกำลังเรียนรู้ลักษณะตัวอักษร ซึ่งอัตราการเรียนรู้ที่ทดลองมีการเปลี่ยนแปลงมีค่าเริ่มต้นและสุดท้ายเท่ากับ 0.7 และ 0.1 ดังเช่น แบบจำลองโครงข่ายที่มีนิวรอนในชั้นซ่อนเท่ากับ 50 มีอัตราการเรียนรู้คงที่ 0.1 ผลที่ได้จากการทดลองกับตัวอักษรที่ใช้ในการเรียนรู้ที่มีความผิดพลาด 10% ปรากฏว่าสามารถระบุได้เพียง 88.46% แต่ถ้ามีการเปลี่ยนแปลงอัตราการเรียนรู้ (Learning

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Rate) ระหว่างที่แบบจำลองโครงข่ายกำลังเรียนรู้ (Learning) จะทำให้สามารถระบุได้ถึง 92.31% นอกจากนี้การเรียนรู้ของแบบจำลองโครงข่ายยังสามารถเรียนรู้ได้รวดเร็วขึ้นอีกด้วย

เมื่อทำการทดลองกับแบบจำลองซึ่งมีจำนวนนิวรอนในชั้นข้อมูลเข้า (Input Layer) และ และจำนวนนิวรอนในชั้นซ่อน (Hidden Layer) ผลที่ได้จากการทดลองสรุปได้ดังนี้

จำนวนนิวรอนในชั้นซ่อน	50	100
ตัวอักษรที่ใช้ในการทดสอบ	จำนวน (ร้อยละ)	จำนวน (ร้อยละ)
Noise 10%	25 (96.15)	23 (88.46)
AngsanaUPC16	15 (57.69)	5 (19.23)
AngsanaUPC16B	11 (42.31)	13 (50.00)
Arial10	20 (76.92)	18 (69.23)
BrowalliaUPC16	19 (73.08)	18 (69.23)
DelleniaUPC16	14 (53.85)	16 (61.54)
EucrosiaUPC16	10 (38.46)	10 (38.46)
FreesiaUPC16	15 (57.69)	13 (50.00)
Garamond11	10 (38.46)	10 (38.46)
JasmineUPC16	20 (76.92)	17 (65.38)
MSSansSerif10	15 (57.69)	14 (53.85)
รวม	165 (57.69)	157 (54.90)

ตารางที่ 5.2 ตารางสรุปผลการทดสอบแบบจำลองโครงข่ายประสาทเทียมที่ใช้ข้อมูลในชั้นข้อมูลเข้าจำนวน 144 นิวรอน

จากการทดลองแบบจำลองโครงข่ายประสาทเทียมที่มีนิวรอนในชั้นข้อมูลเข้า (Input Layer) จำนวน 144 นิวรอน โดยมีนิวรอนในชั้นซ่อนจำนวน 50, 100, 150, 200 และ 250 นิวรอน ผลปรากฏว่าแบบจำลองโครงข่ายที่มีนิวรอนในชั้นซ่อน (Hidden Layer) จำนวน 50 และ 100 นิวรอนเท่านั้นที่สามารถระบุตัวอักษรทั้งหมดที่ใช้ในการเรียนรู้ได้ (CordiaUPC ขนาด 16 จุดต่อนิ้ว ตัวหนา) จึงนำแบบจำลองดังกล่าวมาใช้ทดสอบกับตัวอักษรที่มีความผิดเพี้ยน 10% และตัวอักษรอื่นๆ ผลปรากฏว่าแบบจำลองโครงข่ายที่มีนิวรอนในชั้นซ่อนจำนวน 50 นิวรอน สามารถระบุตัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อักษรได้ถูกต้องเพียง 57.69% และแบบจำลองโครงข่ายที่มีนิวรอนในชั้นซ่อนจำนวน 100 นิวรอน สามารถระบุตัวอักษรได้ถูกต้องเพียง 54.90% เท่านั้น

จำนวนนิวรอนในชั้นซ่อน	50	100	150	200	250
ตัวอักษรที่ใช้ในการทดสอบ	จำนวน (ร้อยละ)	จำนวน (ร้อยละ)	จำนวน (ร้อยละ)	จำนวน (ร้อยละ)	จำนวน (ร้อยละ)
Noise 10%	26 (100.00)	26 (100.00)	26 (100.00)	25 (96.15)	23 (88.46)
AngsanaUPC26	14 (53.85)	15 (57.69)	13 (50.00)	13 (50.00)	12 (46.15)
Arial17	18 (69.23)	19 (73.08)	18 (69.23)	17 (65.38)	17 (65.38)
BookAntiqua18	14 (53.85)	12 (46.15)	15 (57.69)	13 (50.00)	15 (57.69)
BrowalliaUPC26	21 (80.77)	23 (88.46)	20 (76.92)	17 (65.38)	21 (80.77)
DelleniaUPC26	11 (42.31)	10 (38.46)	10 (38.46)	14 (53.85)	9 (34.62)
EucrosiaUPC26	12 (46.15)	12 (46.15)	11 (42.31)	13 (50.00)	8 (30.77)
FreesiaUPC26	15 (57.69)	16 (61.54)	13 (50.00)	12 (46.15)	10 (38.46)
Garamond20	17 (65.38)	14 (53.85)	10 (38.46)	10 (38.46)	10 (38.46)
JasmineUPC26	15 (57.69)	17 (65.38)	15 (57.69)	14 (53.85)	14 (53.85)
MสสSSansSerif17	20 (76.92)	22 (84.62)	19 (73.08)	20 (76.92)	19 (73.08)
รวม	183 (63.99)	186 (65.03)	170 (59.44)	168 (58.74)	158 (55.24)

ตารางที่ 5.3 ตารางสรุปผลการทดสอบแบบจำลองโครงข่ายประสาทเทียมที่ใช้ข้อมูลในชั้นข้อมูลเข้าจำนวน 400 นิวรอน โดยใช้ตัวอักษร CordiaUPC ขนาด 26 จุดต่อนิ้ว ตัวหนาในการเรียนรู้

จากการทดลองแบบจำลองโครงข่ายประสาทเทียมที่มีนิวรอนในชั้นข้อมูลเข้า (Input Layer) จำนวน 400 นิวรอน โดยมีนิวรอนในชั้นซ่อนจำนวน 50, 100, 150, 200 และ 250 นิวรอน ผลปรากฏว่าแบบจำลองโครงข่ายที่มีนิวรอนในชั้นซ่อน (Hidden Layer) ทั้งหมดที่ใช้ทดสอบสามารถระบุตัวอักษรทั้งหมดที่ใช้ในการเรียนรู้ได้ (CordiaUPC ขนาด 26 จุดต่อนิ้ว ตัวหนา) จึงนำแบบจำลองดังกล่าวมาใช้ทดสอบกับตัวอักษรที่มีความผิดเพี้ยน 10% และตัวอักษรอื่นๆ ผลปรากฏว่าแบบจำลองโครงข่ายที่มีนิวรอนในชั้นซ่อนจำนวน 50 นิวรอน สามารถระบุตัวอักษรได้ถูกต้องเพียง 63.99% นิวรอนในชั้นซ่อนจำนวน 100 นิวรอน สามารถระบุตัวอักษรได้ถูกต้องเพียง 65.03% นิวรอนในชั้นซ่อนจำนวน 150 นิวรอน สามารถระบุตัวอักษรได้ถูกต้องเพียง 59.44% นิวรอนใน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชั้นซ่อนจำนวน 200 นิวรอน สามารถระบุตัวอักษรได้ถูกต้องเพียง 58.74% นิวรอนในชั้นซ่อนจำนวน 200 นิวรอน สามารถระบุตัวอักษรได้ถูกต้องเพียง 55.24% เท่านั้น

จำนวนนิวรอนในชั้นซ่อน	50	100	150
ตัวอักษรที่ใช้ในการทดสอบ	จำนวน (ร้อยละ)	จำนวน (ร้อยละ)	จำนวน (ร้อยละ)
CordiaUPC26B Noise 10%	26 (100.00)	26 (100.00)	26 (100.00)
CordiaUPC26B	26 (100.00)	26 (100.00)	26 (100.00)
AngsanaUPC26	26 (100.00)	26 (100.00)	26 (100.00)
Arial17	21 (80.77)	17 (65.38)	20 (76.92)
BookAntiqua18	16 (61.54)	15 (57.69)	14 (53.85)
BrowalliaUPC26	18 (69.23)	19 (73.08)	18 (69.23)
DelleniaUPC26	13 (50.00)	15 (57.69)	14 (53.85)
EucrosiaUPC26	14 (53.85)	12 (46.15)	13 (50.00)
FreesiaUPC26	18 (69.23)	14 (53.85)	17 (65.38)
Garamond20	12 (46.15)	14 (53.85)	14 (53.85)
JasmineUPC26	17 (65.38)	16 (61.54)	18 (69.23)
MSSansSerif17	21 (80.77)	21 (80.77)	21 (80.77)
รวม	228 (73.08)	221 (70.83)	227 (72.76)

ตารางที่ 5.4 ตารางสรุปผลการทดสอบแบบจำลองโครงข่ายประสาทเทียมที่ใช้ข้อมูลในชั้นข้อมูลเข้าจำนวน 400 นิวรอน โดยใช้ตัวอักษร CordiaUPC ขนาด 26 จุดต่อนิ้ว ตัวหนา และ AngsanaUPC ขนาด 26 จุดต่อนิ้ว ในการเรียนรู้

จากการทดลองแบบจำลองโครงข่ายประสาทเทียมที่มีนิวรอนในชั้นข้อมูลเข้า (Input Layer) จำนวน 400 นิวรอน โดยมีนิวรอนในชั้นซ่อนจำนวน 50, 100, 150 และ 200 นิวรอน ผลปรากฏว่าแบบจำลองโครงข่ายที่มีนิวรอนในชั้นซ่อน (Hidden Layer) ทั้งหมดที่ใช้ทดสอบสามารถระบุตัวอักษรทั้งหมดที่ใช้ในการเรียนรู้ได้ (CordiaUPC ขนาด 26 จุดต่อนิ้ว ตัวหนา และ AngsanaUPC ขนาด 26 จุดต่อนิ้ว) จึงนำแบบจำลองดังกล่าวมาใช้ทดสอบกับตัวอักษรที่มีความผิดเพี้ยน 10% และตัวอักษรอื่นๆ ผลปรากฏว่าแบบจำลองโครงข่ายที่มีนิวรอนในชั้นซ่อนจำนวน 50 นิวรอน สามารถระบุตัวอักษรได้ถูกต้องเพียง 73.08% นิวรอนในชั้นซ่อนจำนวน 100 นิวรอน สามารถระบุตัวอักษร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ได้ถูกต้องเพียง 70.83% และนิวรอนในชั้นซ่อนจำนวน 150 นิวรอน สามารถระบุตัวอักษรได้ถูกต้องเพียง 72.76% เท่านั้น

ผลการทดลองพบว่าแบบจำลองโครงข่ายประสาทเทียมแบบแพร่ย้อนกลับที่ดีที่สุด คือ มีจำนวนนิวรอนในชั้นข้อมูลเข้า (Input Layer) เท่ากับ 20 x 20 = 400 นิวรอน จำนวนนิวรอนในชั้นซ่อน (Hidden Layer) เท่ากับ 50 นิวรอน จะทำให้โครงข่ายมีความสามารถในการระบุตัวอักษรที่ใช้ในการเรียนรู้ได้ทั้งหมด และเมื่อทดสอบโดยใช้ตัวอักษร CordiaUPC ขนาด 26 ตัวอักษรต่อนิ้ว ตัวหนา และ ตัวอักษร AngsanaUPC ขนาด 26 ตัวอักษรต่อนิ้ว ในการเรียนรู้จำนวน 52 ตัวอักษร หลังจากการเรียนรู้จึงนำแบบจำลองไปทดสอบกับตัวอักษรที่ใช้ในการเรียนรู้แม้จะมีความผิดพลาด 10% ก็สามารถระบุได้ทั้งหมด และสามารถระบุตัวอักษรอื่น ๆ ที่ใช้ในการทดลองได้ 228 ตัวอักษร จาก 312 ตัวอักษร คิดเป็น 73.08% นอกจากนี้การทดลองทำให้ทราบว่าจำนวนนิวรอนในชั้นข้อมูลเข้า (Input Layer) ชั้นซ่อน (Hidden Layer) และอัตราการเรียนรู้ (Learning Rate) มีความสำคัญต่อการเรียนรู้และจดของแบบจำลองโครงข่าย ดังนี้

1. จำนวนนิวรอนในชั้นข้อมูลเข้า (Input Layer) ที่มีขนาดไม่เท่ากันจะมีผลต่อการเรียนรู้ที่แตกต่างกัน ซึ่งจากการทดลองพบว่าการกำหนดจำนวนนิวรอนในชั้นข้อมูลเข้า (Input Layer) ที่มีขนาด 20 x 20 = 400 นิวรอน จะให้ผลต่อการเรียนรู้ที่ดีกว่าขนาด 12 x 12 = 144 นิวรอน

2. จำนวนนิวรอนในชั้นซ่อน (Hidden Layer) ที่มากเกินไปจะมีผลต่อการเรียนรู้ของแบบจำลอง ซึ่งจากการทดลองจำนวนนิวรอนในชั้นซ่อนที่มากกว่า 150 นิวรอนมีผลทำให้แบบจำลองโครงข่ายมีประสิทธิภาพในการระบุตัวอักษรของแบบจำลองลดน้อยลง ซึ่งจำนวนนิวรอนในชั้นซ่อนที่เหมาะสมคือ 50 และ 150 นิวรอน

3. การอัตราการเรียนรู้ (Learning Rate) จะมีผลต่อการเรียนรู้ของโครงข่าย จากการทดลองพบว่าหากมีการระบุอัตราการเรียนรู้แบบคงที่ แบบจำลองโครงข่ายจะมีประสิทธิภาพในการเรียนรู้ได้น้อยกว่าการที่มีการเปลี่ยนแปลงอัตราการเรียนรู้ (Learning Rate) ระหว่างที่โครงข่ายกำลังเรียนรู้ลักษณะตัวอักษร

5.2 ปัญหาและข้อจำกัดของการทำงานของแบบจำลอง

การเรียนรู้ของแบบจำลองโครงข่ายประสาทเทียมแบบแพร่ย้อนกลับ (Backpropagation Network) เป็นรูปแบบการเรียนรู้แบบแบบมีครูสอน (Supervised Learning Algorithm) ดังนั้นในการเรียนรู้จึงต้องเสียเวลาในการเรียนรู้ก่อนจึงจะนำแบบจำลองนั้นไปใช้งนได้ และยิ่งถ้าจำนวนนิวรอนในชั้นซ่อน (Hidden Layer) และจำนวนตัวอักษรที่ใช้ในการเรียนรู้ยิ่งมีมากเท่าไร ยิ่งต้องใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เวลาในการเรียนรู้มากขึ้นเท่านั้น แม้ว่าจะมีการเปลี่ยนแปลงอัตราการเรียนรู้ (Learning Rate) ในเวลาที่โครงข่ายกำลังเรียนรู้ก็ตาม

5.3 ข้อเสนอแนะ

1. สำหรับการพัฒนาในอนาคตควรมีการปรับปรุงแบบจำลองโครงข่ายให้ประสิทธิภาพมากกว่านี้ เพื่อให้แบบจำลองโครงข่ายสามารถที่จะระบุตัวอักษรได้หลากหลายมากขึ้น หรือวิเคราะห์ตัวอักษรได้หลายฟอนต์และหลายขนาด ซึ่งอาจจะให้เทคนิคอื่นเข้ามาช่วยในการเรียนรู้ของโครงข่าย

2. ในการเรียนรู้ของแบบจำลองโครงข่าย ควรมีการปรับปรุงให้แบบจำลองโครงข่ายสามารถที่จะเรียนรู้ได้รวดเร็วขึ้นกว่าปัจจุบัน ซึ่งอาจใช้เทคนิคอื่นเข้ามาช่วย เช่น การนำ Fussy Logic หรือมี Function พิเศษมาช่วยในการหาอัตราการเรียนรู้ (Learning Rate) ที่เหมาะสมของแบบจำลองโครงข่าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

ชม กิ้มปาน. 2525. "ทฤษฎีการรู้จำรูปแบบเบื้องต้น". ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมคอมพิวเตอร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง กรุงเทพ. พิมพ์ครั้งที่ 1

ทวี เปรมรัตนชัย. 2538 . "การรู้จำอักขระตัวพิมพ์ภาษาไทยด้วยวิธีโครงข่ายนิวรอนแบบแบคพรอบพาเกชั่น". วิทยานิพนธ์วิทยาศาสตร์มหาบัณฑิต สาขาวิทยาการคอมพิวเตอร์และเทคโนโลยีสารสนเทศ บัณฑิตวิทยาลัย, สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง.

เพ็ญพรรณ ใช้ฮวดเจริญ. 2541. "การจดจำอักขระภาษาไทยตัวพิมพ์โดยโครงข่ายประสาทเทียมแบบแบบอะแดปทีฟรีโซแนนซ์เทียรี่". วิทยานิพนธ์วิทยาศาสตร์มหาบัณฑิต สาขาวิทยาการคอมพิวเตอร์และเทคโนโลยีสารสนเทศ บัณฑิตวิทยาลัย, สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง.

วาภิรมย์ มนัสรังสี. 2540. "การจดจำรูปแบบใบหน้ามนุษย์". วิทยานิพนธ์วิทยาศาสตร์มหาบัณฑิต สาขาวิทยาการคอมพิวเตอร์และเทคโนโลยีสารสนเทศ บัณฑิตวิทยาลัย, สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง.

สมศักดิ์ วลัยรัชต์ และ รศ.ดร.ชม กิ้มปาน. 2534. "การแยกแยะและจัดกลุ่มตัวอักษรตัวพิมพ์หลายรูปแบบ" การประชุมทางวิชาการทางวิศวกรรมไฟฟ้า ครั้งที่ 14, คณะวิศวกรรมคอมพิวเตอร์ มหาวิทยาลัยสงขลานครินทร์ วิทยาเขตหาดใหญ่.

อภิรักษ์ จิรายุสกุล. 2539. "การวิเคราะห์ตัวพิมพ์อักษรภาษาไทยโดยใช้ CPN". วิทยานิพนธ์วิทยาศาสตร์มหาบัณฑิต สาขาวิทยาการคอมพิวเตอร์และเทคโนโลยีสารสนเทศ บัณฑิตวิทยาลัย, สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง.

อัญชลี วานิชทวีวัฒน์. 2540. "การจดจำอักขระภาษาไทยตัวพิมพ์โดยโครงข่ายประสาทเทียมแบบเซลท์ออร์แกนไนซิ่งแมบซ์". วิทยานิพนธ์วิทยาศาสตร์มหาบัณฑิต สาขาวิทยาการคอมพิวเตอร์และเทคโนโลยีสารสนเทศ บัณฑิตวิทยาลัย, สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

David M.Skapura. 1996. "Building Neural Network". ACM Press. New York.

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

Fausett Laurene, 1994. "Fundamentals of Neural Networks", Prentice Hall.

Igor Aleksander. n.d. "An Introduction to Neural Computing" Helen Morton. Chapman & Hall.

Jame A.Freeman and Devid. 1991. "Neural Networks Algorithms, Application, and Programming Techniques. Addison-Wesley.

Richard K.Miller. 1987. "Neural Networks", SEAI Technical Publications, 1987

Robert Essenreiter. 2000. "Neural Networks". [Online] Available :URL:HTTP://www-gpi.physik.uni-karlsruhe.de/pub/rebert/Diplom/ node7.htm

Robert Hecht-Nielsen. 1990 . "Neurocomputing" Addison-Wesley. Reading.MA.

Russel C. Eberhart and Roy W. n.d."Neural Network and Pc tools a Pratical Guide". Dobbins. Academic Press, Inc.

Schuberger Lab For Computer Science. n.d . "Artificial Intelligence in Process Eng" Academic Press, Inc.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

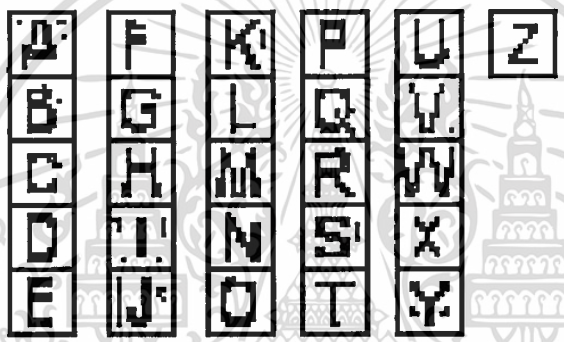
ภาคผนวก

แบบตัวอักษรที่ใช้ในการทดลอง

1. ตัวอักษร CordiaUPC ขนาด 16 จุดต่อนิ้ว ตัวหนา

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

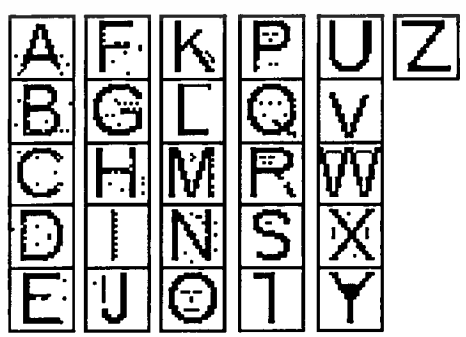
2. ตัวอักษร CordiaUPC ขนาด 16 จุดต่อนิ้ว ตัวหนาที่มีความผิดพลาด (Noise) 10%



3. ตัวอักษร CordiaUPC ขนาด 26 จุดต่อนิ้ว ตัวหนา

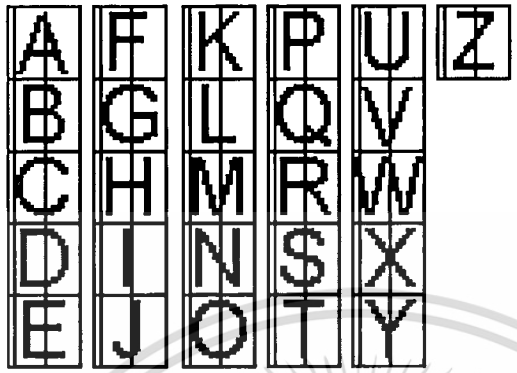
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

4. ตัวอักษร CordiaUPC ขนาด 26 จุดต่อนิ้ว ตัวหนาที่มีความผิดพลาด (Noise) 5%



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. ตัวอักษร CordiaUPC ขนาด 26 จุดต่อนิ้ว ตัวหนาที่มีความผิดพลาด (Noise) 10%



6. ตัวอักษร AngsanaUPC ขนาด 26 จุดต่อนิ้ว

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

7. ตัวอักษร Arial ขนาด 17 จุดต่อนิ้ว

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

8. ตัวอักษร BookAntiqua ขนาด 18 จุดต่อนิ้ว

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

9. ตัวอักษร BrowalliaUPC ขนาด 26 จุดต่อนิ้ว

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

10. ตัวอักษร DelleniaUPC ขนาด 26 จุดต่อนิ้ว

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

11. ตัวอักษร EucrosiaUPC ขนาด 26 จุดต่อนิ้ว

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

12. ตัวอักษร MS Sansrif ขนาด 17 จุดต่อนิ้ว

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

13. ตัวอักษร FreesiaUPC ขนาด 26 จุดต่อนิ้ว

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

14. ตัวอักษร Garamond ขนาด 20 จุดต่อนิ้ว

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

15. JasmineUPC ขนาด 26 จุดต่อนิ้ว

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

16. ตัวอักษร MSSansSerif ขนาด 17 จุดต่อนิ้ว

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมแบบจำลองโครงข่ายสำหรับการเรียนรู้ (Learning)

```
//Program Learning Charecter
#include <stdio.h>
#include <stdlib.h>
#include <conio.h>
#include <ctype.h>
#include <math.h>

#define NoOfInputNode 400
#define NoOfHiddenNode 200
#define NoOfOutputNode 26
#define MinValue .2
#define MaxValue .8

void GetInput();
void PrintInput();
void PrintTarget();
void GetWij();
void PrintWij();
void GetWjk();
void PrintWjk();
void GetLearn();
void CalNodeY();
void PrintHidden();
void CalNodeO();
void PrintOutput();
void CalDifWjk();
void AdjWjk();
void CalDifWij();
void AdjWij();
void PrtToFileWij();
void PrtToFileWjk();
void PrtToFileLearn();

//Declare Public Variable
   FILE *ABC;
   float NodeI[NoOfInputNode];
   float NodeU[NoOfHiddenNode];
   float NodeY[NoOfHiddenNode];
   float NodeV[NoOfOutputNode];
   float NodeO[NoOfOutputNode];
   float NodeT[NoOfOutputNode];
   float Wij[NoOfInputNode]
  [NoOfHiddenNode];
   float Wjk[NoOfHiddenNode]
  [NoOfOutputNode];
   float DifWij[NoOfInputNode]
  [NoOfHiddenNode];
   float Dk[NoOfOutputNode];
   float DifWjk[NoOfHiddenNode]
  [NoOfOutputNode];
   float Dj[NoOfHiddenNode];
   float LEARN,MinLearn,MaxLearn;
   float SumError,TotalError,TError;
  int NoOfData;

void main()
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
{
//Check file is exist
   ABC = fopen("ABC.DAT","rt");
   if (ABC == NULL)
   {
      printf("Input File does not exist\n");
      getch();
      return;
   }
  float MinError= 0;
   int i,k,MELoop;
   char quit;

  printf("No of Data : ");
   scanf("%d", &NoOfData);
  printf("Min Learning Rate : ");
   scanf("%f", &MinLearn);
  printf("Max Learning Rate : ");
   scanf("%f", &MaxLearn);

  //Get Weight Wij & Wjk
   GetWij();//clrscr();PrintWij();getch();
   GetWjk();//PrintWjk();getch();

  GetLearn();
  MELoop=1;
  k = 0;
  i = 0;
   while (quit != 'q')
   {
      TotalError = 0;
      for(int j=0;j<NoOfData;j++)
      {
         //clrscr();//PrintWij();//PrintWjk();
         GetInput();//PrintInput
();PrintTarget();getch();
         CalNodeY();//PrintHidden();getch
();
         CalNodeO();//PrintOutput();getch
();
         CalDifWjk();
         CalDifWij();
         AdjWjk();
         AdjWij();
         //clrscr();PrintWij();//PrintWjk
();quit=getch();
         TotalError = TotalError +
SumError;
      TError = TotalError * 100000000;
      }

      //If Total Error is change,write weight
to file.
    if (TError < MinError && MinError <
0.005)
    {
         PrtToFileWij();
         PrtToFileWjk();
    }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    if (MinError == 0)
        MinError = TError;

    if (TError < MinError)
    {
        MELoop = 0;
        MinError = TError;
    }
    else
        MELoop++;

    //If Min Error isn't change for long
time,reduce LEARNING RATE.
        if (MELoop > 50 && LEARN >
MinLearn)
    {
        if (MinError > 100 && LEARN >
0.2)
            LEARN = LEARN - 0.05;
       else
            LEARN = LEARN - 0.01;
            MELoop = 0;
            PrtToFileLearn();
       MinError = TError;
    }

        printf("TError = %f  k= %d MError =
%f M=%d
L=%f\n",TError,k++,MinError,MELoop,LE
ARN);
        if(i==500 && TError > 0.05)
        {
            PrtToFileWij();
            PrtToFileWjk();
            i=1;
        }
        else
        {
            i++;
        }
        }
        return;
    }
    void GetInput()
    {
        char ch;
        fpos_t filepos = 0;
        int i=0,TARGET;
        ch =fgetc(ABC);
    //READ Data to InputNode and TargetNode
        do
        {
        //Set pointer to begin of file if EOF
        if(ch == EOF)
        {
            fsetpos(ABC, &filepos);
            ch =fgetc(ABC);
        }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    //Select READ Data A-Z for TargetNode
        if(ch >= 'A' && ch <= 'Z')
        {
            //Initial Target
            for(int j =
0;j<NoOfOutputNode;j++)
                NodeT[j] = MinValue;
            TARGET = toascii(ch)-65;  //Map
Charecter A->1,B->2,...,Z->26
            NodeT[TARGET] = MaxValue;
        }
    //Select READ Data 0,1 for InputNode
        switch(ch)
        {
            case '0':
        NodeI[i] = MinValue;
         i++;
         break;
            case '1':
        NodeI[i] = MaxValue;
         i++;
         break;
        }
/*      if(ch == '0' || ch == '1')
        {
            NodeI[i] = toascii(ch)-48;
            ;i++;
        }*/
    // Read charecter from File
        ch = fgetc(ABC);
    }while(i < NoOfInputNode);
}

void PrintInput()
{
    //Print Input Node
    for(int i=0;i < NoOfInputNode;i++)
    {
        if(i%12 == 0)
            printf("\n");
        printf("%f",NodeI[i]);
    }
}

void PrintTarget()
{
    //Print Output Node
    printf("\nTarget is : ");
    for(int i=0;i < NoOfOutputNode;i++)
        printf("%f",NodeT[i]);
    printf("\n");
}

void GetWij()
{
    FILE *WeightIJ;
    WeightIJ = fopen("Wij.txt","rt");
    float Weight=0;
    for(int i=0;i<NoOfInputNode;i++)
        for(int j=0;j<NoOfHiddenNode;j++)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        {
            if
(fscanf(WeightIJ,"%f",&Weight) == -1)
                Weight = random(10000);
            //Wij[i][j] = random
(10000)*.0001;
            //printf("Weight =
%f\n",Weight);getch();
            Wij[i][j] = Weight * .00001;
            fgetc(WeightIJ);
        }
  fclose(WeightIJ);
}

void PrintWij()
{
    //Print WeightIJ
    for(int i=0;i<NoOfInputNode;i++)
        for(int j=0;j<NoOfHiddenNode;j++)
            printf("Wij[%d][%d] =
%f\n",i,j,Wij[i][j]);
}

void GetWjk()
{
    FILE *WeightJK;
    WeightJK = fopen("Wjk.txt","rt");
    float Weight;
    for(int i=0;i<NoOfHiddenNode;i++)
        for(int j=0;j<NoOfOutputNode;j++)
        {
            if
(fscanf(WeightJK,"%f",&Weight) == -1)
                Weight = random(10000);
            //Wjk[i][j] = random
(10000)*.0001;
            Wjk[i][j] = Weight*.00001;
            fgetc(WeightJK);
        }
  fclose(WeightJK);
}
void PrintWjk()
{
  printf("Print WeightJK\n");
  for(int i=0;i<NoOfHiddenNode;i++)
    for(int j=0;j<NoOfOutputNode;j++)
      printf("Wjk[%d][%d] =
%f\n",i,j,Wjk[i][j]);
  }

void GetLearn()
{
    FILE *FileLearn;
    FileLearn = fopen("Learn.txt","rt");
  if (FileLearn == NULL)
      LEARN = MaxLearn;
  else
  {
  fscanf(FileLearn,"%f",&LEARN);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        fclose(FileLearn);
    }
}

void CalNodeY()
{
    int i,j,k = 0;
    for(i=0;i<NoOfHiddenNode;i++)
        {
            NodeU[i] = 0;
            NodeY[i] = 0;
        }
    for(i=0;i<NoOfInputNode;i++)
        for(j=0;j<NoOfHiddenNode;j++)
        {
        //Sum(XiWij)
            NodeU[j] = NodeU[j] + (NodeI[i]
* Wij[i][j]);
//          printf("X[%d] - W[%d][%d] = %f
x %f = %f\n",i,i,j,NodeI[i],Wij[i][j],NodeI[i]
* Wij[i][j]);
        k++;
    }
    for(j=0;j<NoOfHiddenNode;j++)
  {
        //f(Ui) = 1/(1+e^-Ui)
//      printf("NodeU[%d] = %f\n",j,NodeU
[j]);
        NodeY[j] = 1/(1+exp(NodeU[j]*-1));
//      getch();
    }
}

void PrintHidden()
{
    printf("Print Hidden\n");
    for(int i=0;i<NoOfHiddenNode;i++)
        printf("NodeU[%d] = %f,NodeY[%d]
= %f\n",i,NodeU[i],i,NodeY[i]);
}

void CalNodeO()
{
    int i,j = 0;
    for(i=0;i<NoOfOutputNode;i++)
    {
        NodeV[i] = 0;
        NodeO[i] = 0;
    }
    for(i=0;i<NoOfHiddenNode;i++)
        for(j=0;j<NoOfOutputNode;j++)
        {
        //Sum(YiWjk)
            NodeV[j] = NodeV[j] + (NodeY[i]
* Wjk[i][j]);
//      printf("Y[%d]W[%d][%d] = %f x %f
= %f\n",i,i,j,NodeY[i],Wjk[i][j],NodeY[i] *
Wjk[i][j]);
    }
  for(j=0;j<NoOfOutputNode;j++)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    {
      //f(V) = /(1+e^-Vi)
            NodeO[j] = 1/(1+exp(NodeV[j]*-
1));
//          printf("[%d][%d] NodeV = %f
,NodeO = %f \n",i,j,NodeV[j],NodeO
[j]);getch();
        }
}

void PrintOutput()
{
    printf("Print Output\n");
    for(int i=0;i<NoOfOutputNode;i++)
        printf("NodeV[%d] = %f,NodeO[%d]
= %f\n",i,NodeV[i],i,NodeO[i]);
}

void CalDifWjk()
{
    int i,j=0;
    SumError = 0;
    float Error,Error2;
    for(i=0;i<NoOfOutputNode;i++)
    {
        //DkI = (Tk-Ok) * (Ok(1-Ok));
        //f'(v) = Ok(1-Ok)
        // Error = Tk-Ok
        Error = (NodeT[i] - NodeO[i]);
        Error2 = Error * Error;
        SumError = SumError + Error2;
//
    printf("%f\n",SumError*100000000000);
getch();
//      printf("f'(v) = %f  ",NodeO[i]*(1 -
NodeO[i]));
        Dk[i] = Error * (NodeO[i] *(1 -
NodeO[i]));
//      printf("%d = T-O = [%f] - [%f] = %f
\n",i,NodeT[i],NodeO[i],Error);
//      printf("Dk = %f\n",Dk[i]);
    }
    for(i=0;i<NoOfHiddenNode;i++)
    {
        for(j=0;j<NoOfOutputNode;j++)
        {
//          printf("DifWjk[%d][%d]= %f x
%f = ",i,j,Dk[j],NodeY[i]);
            DifWjk[i][j] = LEARN * Dk[j] *
NodeY[i];
//          printf(" %f\n",DifWjk[i]
[j]*10000000000);
        }
//    getch();
    }
}

void AdjWjk()
{
    int i,j=0;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
//   printf("WjkOLD + DifWjk =
WjkNEW\n");
    for(i=0;i<NoOfHiddenNode;i++)
        for(j=0;j<NoOfOutputNode;j++)
        {
//          printf("[%d][%d] %f + %f =
",i,j,Wjk[i][j],DifWjk[i][j]);
            Wjk[i][j] = Wjk[i][j] + DifWjk[i]
[j];
//          printf("%f\n",Wjk[i][j]);getch();
        }
}

void CalDifWij()
{
    int i,j=0;
    float DkWjk[NoOfHiddenNode];
    // Calculate Sum(DkWjk)
    for(i=0;i<NoOfHiddenNode;i++)
    {
        DkWjk[i] = 0;
        for(j=0;j<NoOfOutputNode;j++)
        {
            DkWjk[i] = DkWjk[i] + (Dk[j] *
Wjk[i][j]);
//          printf("Dk[%d] x Wjk[%d][%d] =
%f x %f = %f\n",j,i,j,Dk[j],Wjk[i][j],Dk[j] *
Wjk[i][j]);
        }
    }
    // Calculate DeltaJ  = Y(1-Y) x Sum
(DkWjk)
    for(i=0;i<NoOfHiddenNode;i++)
    {
        Dj[i] = (NodeY[i] * (1-NodeY[i]))  *
DkWjk[i];
//    printf("%f\n",DkWjk[i]);
//    getch();
//      printf("f'(U) x DkWjk = %f x %f =
%f\n",NodeY[i]*(1-NodeY[i]),DkWjk[i],Dj
[i]);
    }
    for(i=0;i<NoOfInputNode;i++)
    for(j=0;j<NoOfHiddenNode;j++)
    {
//          printf("DifWij[%d][%d]= %f x %f
= ",i,j,Dj[j],NodeI[(i*NoOfInputNode)+j]);
        DifWij[i][j] = LEARN * Dj[j] *
NodeI[i];
//          printf(" %f\n",DifWij[i]
[j]*1000000000000);getch();
    }
}

void AdjWij()
{
    int i,j=0;
//  printf("WijOLD + DifWij =
WijNEW\n");
    for(i=0;i<NoOfInputNode;i++)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        for(j=0;j<NoOfHiddenNode;j++)
        {
//          printf("[%d][%d] %f + %f =
",i,j,Wij[i][j],DifWij[i][j]);
            Wij[i][j] = Wij[i][j] + DifWij[i][j];
//          printf("%f\n",Wij[i][j]);
        }
}

void PrtToFileWij()
{
    FILE *FileWij;
    FileWij = fopen("Wij.txt","w");
    for(int i=0;i<NoOfInputNode;i++)
    {
        for(int j=0;j<NoOfHiddenNode;j++)
        {
            fprintf(FileWij,"%f", Wij[i][j] *
100000);
            fprintf(FileWij,"%s",",");
        }
        fprintf(FileWij,"%c",'\n');
    }
    fclose(FileWij);
}

void PrtToFileWjk()
{
    FILE *FileWjk;
    FileWjk = fopen("Wjk.txt","w");
    for(int i=0;i<NoOfHiddenNode;i++)
    {
        for(int j=0;j<NoOfOutputNode;j++)
        {
            fprintf(FileWjk,"%f",Wjk[i]
[j]*100000);
            fprintf(FileWjk,"%s",",");
        }
        fprintf(FileWjk,"%c",'\n');
    }
    fclose(FileWjk);
}

void PrtToFileLearn()
{
    FILE *FileLearn;
    FileLearn = fopen("Learn.txt","w");
    fprintf(FileLearn,"%f",LEARN);
    fclose(FileLearn);
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมสำหรับการทดสอบ (Testing) แบบจำลองโครงข่าย

```
//Program Test Neural

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

#define NoOfInputNode 400
#define NoOfHiddenNode 250
#define NoOfOutputNode 26
#define MinValue 0
#define MaxValue 1

void GetInput();
void PrintInput();
void PrintTarget();
void GetWij();
void PrintWij();
void GetWjk();
void PrintWjk();
void CalNodeY();
void PrintHidden();
void CalNodeO();
void CalError();
void PrintOutput();

FILE *ABC;

float NodeI[NoOfInputNode];
float NodeU[NoOfHiddenNode];
float NodeY[NoOfHiddenNode];
float NodeV[NoOfOutputNode];
float NodeO[NoOfOutputNode];
float NodeT[NoOfOutputNode];
float Wij[NoOfInputNode]
[NoOfHiddenNode];
float Wjk[NoOfHiddenNode]
[NoOfOutputNode];

void main()
{
  char FileName[20];
//Check file is exist
  printf("Input File : ");
  scanf("%20s",&FileName);
  ABC = fopen(FileName,"rt");
  if (ABC == NULL)
  {
     printf("Input File does not exist\n");
     getch();
     return;
  }
//Declare Array
  char quit;

//Get Weight Wij & Wjk
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    GetWij();//clrscr();PrintWij();getch();
    GetWjk();//PrintWjk();getch();

  clrscr();
    for(int i=0;i < 26;i++)
    {
        //clrscr();//PrintWij();//PrintWjk();
        GetInput();
        //PrintInput();
        PrintTarget();
    CalNodeY();//PrintHidden();getch();
        CalNodeO();
        PrintOutput();
    CalError();
    if (i == 12)
        quit = getch();
    }
    return;
}

void GetInput()
{
    char ch;
    fpos_t filepos = 0;
    int i=0,TARGET;
    ch =fgetc(ABC);
//READ Data to InputNode and TargetNode
    do
    {
    //Set pointer to begin of file if EOF
    if(ch == EOF)
    {
        fsetpos(ABC, &filepos);
    }
    //Select READ Data A-Z for TargetNode
        if(ch >= 'A' && ch <= 'Z')
        {
            //Initial Target
            for(int j =
0;j<NoOfOutputNode;j++)
                NodeT[j] = MinValue;
            TARGET = toascii(ch)-65;  //Map
Charecter A->1,B->2,...,Z->26
            NodeT[TARGET] = MaxValue;
        }
    //Select READ Data 0,1 for InputNode
        switch(ch)
        {
            case '0':
        NodeI[i] = MinValue;
         i++;
         break;
            case '1':
        NodeI[i] = MaxValue;
         i++;
         break;
        }
/*      if(ch == '0' || ch == '1')
        {
            NodeI[i] = toascii(ch)-48;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            ;i++;
        }*/
    // Read charecter from File
        ch = fgetc(ABC);
    }while(i < NoOfInputNode);
}

void PrintInput()
{
    //Print Input Node
    for(int i=0;i < NoOfInputNode;i++)
    {
        if(i%12 == 0)
            printf("\n");
        printf("%f",NodeI[i]);
    }
}

void PrintTarget()
{
    int i;
  float Max = MaxValue;
    //Print Target
    for(i=0; NodeT[i] != Max;i++);
    printf("Target is : %c ",65+i);
}

void GetWij()
{
    FILE *WeightIJ;
    WeightIJ = fopen("Wij.txt","rt");
    float Weight=0;
    for(int i=0;i<NoOfInputNode;i++)
        for(int j=0;j<NoOfHiddenNode;j++)
        {
            if
(fscanf(WeightIJ,"%f",&Weight) == -1)
                Weight = random(10000);
                //Wij[i][j] = random
(10000)*.0001;
//              printf("Weight =
%f\n",Weight);getch();
                Wij[i][j] = Weight * .0001;
                fgetc(WeightIJ);
            }
        fclose(WeightIJ);
    }

void PrintWij()
{
    for(int i=0;i<NoOfInputNode;i++)
        for(int j=0;j<NoOfHiddenNode;j++)
            printf("Wij[%d][%d] =
%f\n",i,j,Wij[i][j]);
}

void GetWjk()
{
    FILE *WeightJK;
    WeightJK = fopen("Wjk.txt","rt");
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    float Weight;
    for(int i=0;i<NoOfHiddenNode;i++)
        for(int j=0;j<NoOfOutputNode;j++)
        {
            if
(fscanf(WeightJK,"%f",&Weight) == -1)
                Weight = random(10000);
            //Wjk[i][j] = random
(10000)*.0001;
            Wjk[i][j] = Weight*.0001;
            fgetc(WeightJK);
        }
    fclose(WeightJK);
}

void PrintWjk()
{
    printf("Print WeightJK\n");
    for(int i=0;i<NoOfHiddenNode;i++)
        for(int j=0;j<NoOfOutputNode;j++)
            printf("Wjk[%d][%d] =
%f\n",i,j,Wjk[i][j]);
}

void CalNodeY()
{
    int i,j;
    for(i=0;i<NoOfHiddenNode;i++)
        {
            NodeU[i] = 0;
            NodeY[i] = 0;
        }
    for(i=0;i<NoOfInputNode;i++)
        for(j=0;j<NoOfHiddenNode;j++)
        {
        //Sum(XiWij)
            NodeU[j] = NodeU[j] + (NodeI[i]
* Wij[i][j]);
//          printf("X[%d] - W[%d][%d] = %f
x %f = %f\n",i,i,j,NodeI[i],Wij[i][j],NodeI[i]
* Wij[i][j]);
        }
    for(j=0;j<NoOfHiddenNode;j++)
    {
        //f(Ui) = 1/(1+e^-Ui)
//      printf("NodeU[%d] = %f\n",j,NodeU
[j]);
        NodeY[j] = 1/(1+exp(NodeU[j]*-1));
//      getch();
    }
}

void PrintHidden()
{
    printf("Print Hidden\n");
    for(int i=0;i<NoOfHiddenNode;i++)
        printf("NodeU[%d] = %f,NodeY[%d]
= %f\n",i,NodeU[i],i,NodeY[i]);
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
void CalNodeO()
{
    int i,j;
    for(i=0;i<NoOfOutputNode;i++)
        {
            NodeV[i] = 0;
            NodeO[i] = 0;
        }
    for(i=0;i<NoOfHiddenNode;i++)
        for(j=0;j<NoOfOutputNode;j++)
        {
        //Sum(YiWjk)
            NodeV[j] = NodeV[j] + (NodeY
[i] * Wjk[i][j]);
//      printf("Y[%d]W[%d][%d] = %f x %f
= %f\n",i,i,j,NodeY[i],Wjk[i][j]),NodeY[i] *
Wjk[i][j]);
    }
  for(j=0;j<NoOfOutputNode;j++)
    {
      //f(V) = /(1+e^-Vi)
            NodeO[j] = 1/(1+exp(NodeV[j]*-
1));
//          printf("[%d][%d] NodeV = %f
,NodeO = %f \n",i,j,NodeV[j],NodeO[j]);
//          printf("NodeO[%d] = %f
",j,NodeO[j]*100);
//      if (j%2 == 1) printf("\n");
        }
}
```

```
void CalError()
{
    float SumError,WkError;
  SumError = 0;
    for(int i=0;i<NoOfOutputNode;i++)
  {
    if (NodeT[i] >= NodeO[i])
            WkError = NodeT[i] - NodeO[i];
    else
            WkError = NodeO[i] - NodeT[i];
        SumError = SumError + WkError;
    }
    printf("  Error : %f\n",SumError*100);
}

void PrintOutput()
{
    float Max=0;
  int MaxPos=-1;
    for(int i=0;i<NoOfOutputNode;i++)
  {
    if(NodeO[i] == Max)
    {
          printf("I do not known");
      break;
    }
      else
    {
      if(NodeO[i] > Max)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    {
     Max = NodeO[i];
    MaxPos = i;
    }
  }
```

```
    }
  //   printf("Print Output :
  %c\n",65+MaxPos);
      printf("-> %c",65+MaxPos);
  }
```


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประวัติผู้เขียน

ชื่อ-นามสกุล	นายสมชาย อิทธิพลกังวาล
วัน/เดือน/ปี ที่เกิด	18 ธันวาคม 2517
ที่อยู่	7/205 ซอย 53 ถนนโชคชัย 4 ลาดพร้าว 10130
ประวัติการศึกษา	ระดับชั้นประถมศึกษา โรงเรียนราษฎร์วิทยา
	ระดับชั้นมัธยมศึกษา โรงเรียนพรหมานุสรณ์
	ระดับปริญญาตรี มหาวิทยาลัยธรรมศาสตร์
ประวัติการทำงาน	เม.ย. 2539 – เม.ย. 2543 ผู้พัฒนาระบบสารสนเทศ
	บริษัท เอ็ม ซี ซอฟท์แวร์ (ประเทศไทย) จำกัด
	เม.ย. 2543 – ปัจจุบัน ผู้วิเคราะห์ระบบสารสนเทศ
	บริษัท เอ็ม ซี ซอฟท์แวร์ (ประเทศไทย) จำกัด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้