

การประยุกต์เครือข่ายนิวรอลเพื่อใช้ในการรู้จำตัวอักษรไทย

ผศ.ดร. วิเชียร เปรมชัยสวัสดิ์

ผศ. นุชรี เปรมชัยสวัสดิ์

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยขอนแก่น

บทคัดย่อ

เทคนิคของเครือข่ายนิวรอลได้ถูกนำมาพัฒนาเพื่อใช้ในการรู้จำตัวอักษรไทย โดยการป้อนข้อมูลตัวเข้าไปในลักษณะของภาพ โดยที่แต่ละตัวอักษรมีลักษณะเป็นแมทริกซ์ขนาด 20x8 จากนั้นเครือข่ายนิวรอลจะทำการประมวลผลเพื่อบอกว่าภาพที่ป้อนเข้าไปในนั้นคือตัวอักษรอะไรโดยใช้หลักการของ Back-propagation ผลการทดลองพบว่า ระบบเครือข่ายนิวรอลนี้สามารถให้ความถูกต้องได้เฉลี่ยถึง 98.7 เปอร์เซ็นต์

Application of Artificial Neural Network for Thai Character Recognition

Dr. W. Premchaiswadi

N. Premchaiswadi

Assistant Professor

Department of Computer Engineering

Faculty of Engineering Khon Kaen University

Abstract

Artificial Neural Network technique is used for thai character recognition. The input of this network is in the form of picture which size of 20x8. Then, the neural network processes this picture and give the result in the form of ASCII code of that character by using Back-propagation technique. The results shown that correctness is about 98.7 percents.

บทนำ

ถึงแม้ว่าในปัจจุบันนี้การประมวลผลด้วยเครื่องคอมพิวเตอร์สามารถทำได้ด้วยความสะดวกรวดเร็ว และมีแนวโน้มว่าความเร็วของเครื่องคอมพิวเตอร์จะสูงขึ้นได้อีกโดยการใช้ตัวประมวลผลหลายตัวมาทำงานร่วมกันอย่างขนาน แต่อย่างไรก็ตามนอกเหนือจากความสามารถในการคำนวณอย่างรวดเร็วแล้ว เครื่องคอมพิวเตอร์ยังไม่สามารถช่วยงานมนุษย์ได้อย่างเต็มที่ โดยเฉพาะอย่างยิ่งในงานบางอย่างที่ยังต้องอาศัยแต่กลับเป็นสิ่งที่ยากสำหรับเครื่องคอมพิวเตอร์ เช่น การจดจำใบหน้าคน การจดจำตัวอักษร การเรียนรู้ที่จะพูดและเดิน เป็นต้น.

ในงานหลายอย่างที่ใช้กันอยู่ในปัจจุบัน ก็ได้มีความพยายามที่จะนำเอาคอมพิวเตอร์มาใช้เพื่อการแก้ปัญหาของการจดจำรูปแบบต่างๆ แต่เนื่องจากเทคนิคต่างๆ ที่ใช้กันอยู่ในปัจจุบันไม่เหมาะที่จะแก้ปัญหาเหล่านี้ จึงได้มีความพยายามที่จะพัฒนาวิธีการใหม่ๆ ขึ้น โดยเฉพาะการเลียนแบบการทำงานของสมองเพื่อนำมาใช้ในการจดจำรูปแบบต่างๆ เทคนิคนี้ก็ได้ถูกเรียกกันว่า Artificial Neural Network (ANS) หรือบางทีก็เรียกสั้นๆ ว่า Neural Network ด้วยจุดเด่นของเครือข่ายนิวรอลในด้านการจดจำรูปแบบ ในการทดลองนี้จึงได้นำ เทคนิคของเครือข่ายนิวรอลมาใช้ในการรู้จำตัวอักษรไทย

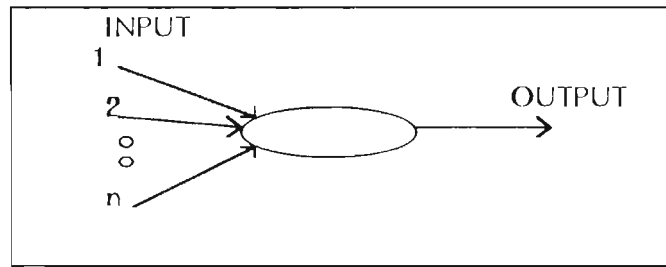
การทำงานของเครือข่ายนิวรอล

ในหัวข้อนี้จะได้อธิบายถึงหลักการทำงานเบื้องต้นของเครือข่ายนิวรอลเพื่อให้เกิดความเข้าใจในหลักการทั่วไป โดยเริ่มต้นจากตัวประมวลผลเพียงตัวเดียว (นิวรอล 1 ตัว) หลังจากนั้นก็จะกล่าวถึงนิวรอลหลายตัวซึ่งอยู่ในชั้นเดียวกัน และท้ายสุดก็จะเป็นการทำงานของเครือข่ายนิวรอลหลายตัวที่มีหลายชั้น

นิวรอลตัวเดียว

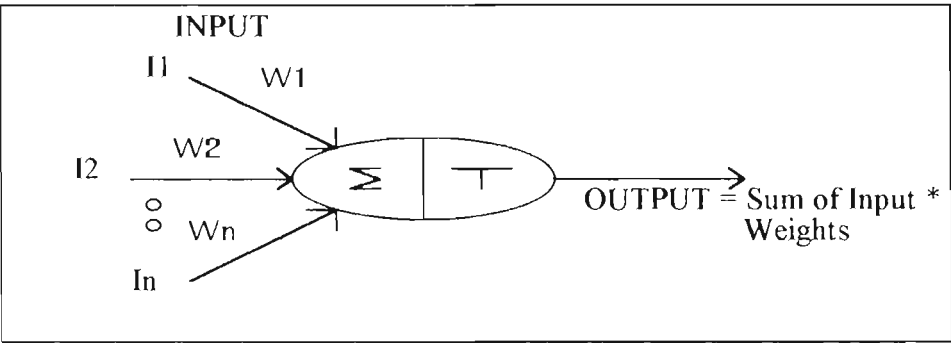
นิวรอลนี้เราจะเรียกว่าเป็น processing element (PE) หรือบางครั้งก็จะเรียกว่าโนด (node) และแทนได้ด้วยวงกลมดังแสดงในรูปที่ 1

นิวรอล 1 ตัวอาจจะมีข้อมูลเข้า (inputs) เข้ามาพร้อม ๆ กันได้หลายตัว ซึ่งนิวรอลจะทำการตอบสนองต่อข้อมูลเข้าเหล่านี้ด้วยการส่งสัญญาณหรือไม่ส่งสัญญาณ ("fire" or "doesn't fire")ต่อไปทางด้านข้อมูลออก การตอบสนองนี้จะขึ้นอยู่กับระดับของ threshold. จากรูปที่ 1 จะเห็นว่าข้อมูลเข้าหลายตัวทำให้เกิดข้อมูลออกเพียงตัวเดียว และนอกจากข้อมูลจริงที่เข้ามาในระบบแล้ว การตอบสนองของนิวรอลยังมีผลมาจากข้อมูลเข้าอย่างอื่นที่เรียกว่า bias term หรือ forcing term ซึ่งตัวแปรนี้อาจจะเป็นตัวที่ทำให้นิวรอลเกิดการลืมหรือไม่สนใจที่จะจดจำ



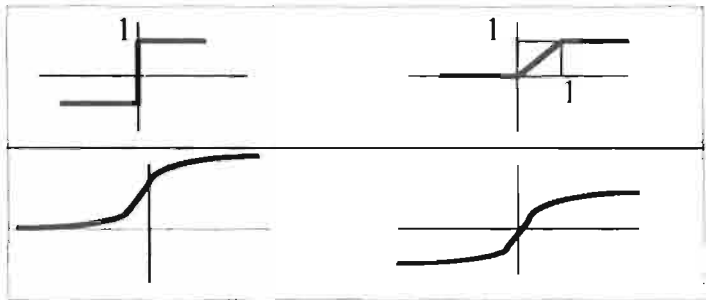
รูปที่ 1 นิวรอนหนึ่งตัว

ข้อมูลเข้าแต่ละตัวจะมีผลต่อนิวรอนไม่เท่ากันซึ่งถูกกำหนดโดยใช้ค่าน้ำหนัก (weight) ของข้อมูลเข้านั้นๆ ดังแสดงในรูปที่ 2



รูปที่ 2 นิวรอน 1 ตัวพร้อมด้วยค่าน้ำหนักของข้อมูลเข้า

ในทางคณิตศาสตร์เราสามารถเขียนข้อมูลเข้าและค่าน้ำหนักของมันให้อยู่ในรูปของเวกเตอร์ได้
เป็น $(I_1, I_2, \dots, I_n)$ และ $(W_1, W_2, \dots, W_n)$ เป็นต้น สัญญาณที่เข้าไปในนิวรอนแต่ละตัว สามารถที่จะ
คำนวณได้จากการทำ dot หรือ Inner product ของเวกเตอร์ทั้งสองนี้
ผลรวมที่ได้จากการคำนวณนี้จะถูกส่งผ่านไปยัง transfer function ซึ่งแทนด้วย T ในรูปที่ 2
Transfer function หรือบางทีเรียกว่า Threshold โดยทั่วไปแล้วมีลักษณะไม่เป็นเชิงเส้น
(nonlinear) ข้อมูลออกของนิวรอนก็จะขึ้นอยู่กับผลการเปรียบเทียบระหว่างผลลัพธ์ของ dot product
กับ Transfer function ซึ่ง transfer function ที่ใช้กันอยู่ก็มีหลายรูปแบบด้วยกันดังแสดงในรูปที่3

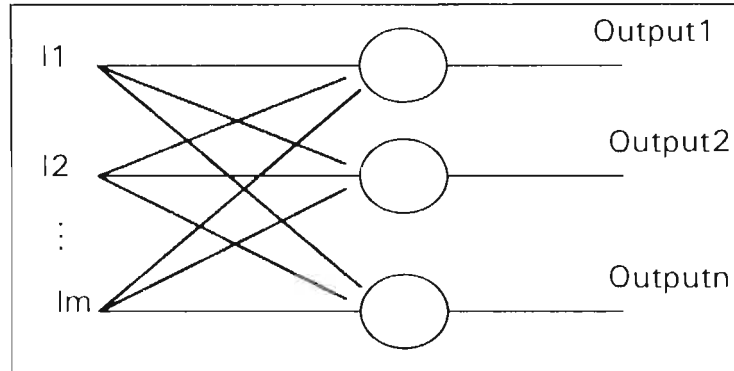


รูปที่ 3 Transfer function

ผลของการคำนวณนี้จะทำให้นิวรอนมีการตอบสนองคือจะส่งข้อมูลต่อไปที่นิวรอนตัวอื่นหรือไม่
ส่งข้อมูลออกไป และถ้าเราเพิ่มหน่วยความจำ ให้แต่ละนิวรอนจะทำให้มันสามารถจดจำ ผลของการคำนวณ
ครั้งก่อนและนำไปปรับค่าน้ำหนักได้ ซึ่งลักษณะเช่นนี้เรียกว่า เครือข่ายเกิดการเรียนรู้

นิวรอลหลายตัวในชั้นเดียวกัน

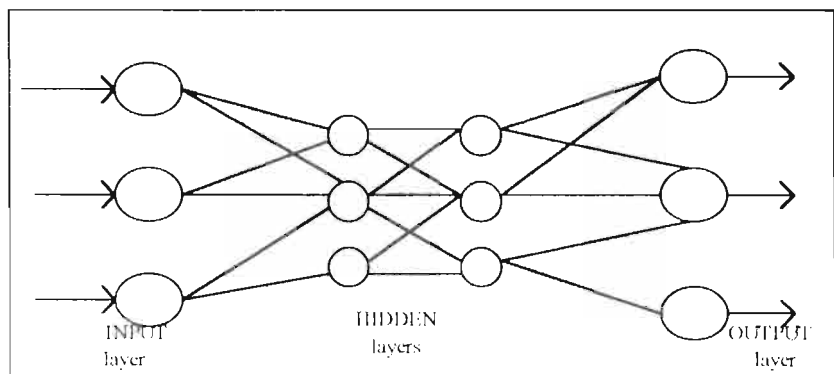
เป็นการที่เรานำ PE หลาย ๆ ตัวมารวมกัน เพื่อให้ข้อมูลผ่านไป ดังแสดงในรูปที่ 4 ข้อมูลเข้าสามารถต่อกับโหนดได้หลายโหนดโดยที่มีค่าของน้ำหนักที่แตกต่างกัน ผลลัพธ์ที่ได้ก็คือกลุ่มของข้อมูลออกของโหนดแต่ละตัว



รูปที่ 4 เครือข่ายนิวรอลแบบชั้นเดียว

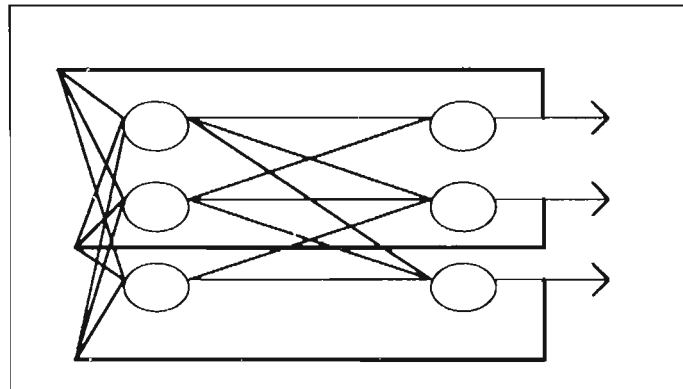
นิวรอลแบบหลายชั้น

จากข้อ 2.2 เราสามารถนำนิวรอลแบบชั้นเดียวแต่ละชั้นมาต่อรวมกัน ซึ่งเราเรียกว่าเป็นนิวรอลแบบหลายชั้น โดยที่ชั้นที่รับข้อมูลเข้าเราจะเรียกว่า ชั้นข้อมูลเข้า (Input layer) ซึ่งในชั้นนี้ก็จะทำหน้าที่เป็น buffer สำหรับรับสัญญาณเข้าเท่านั้น ส่วนผลลัพธ์ของเครือข่ายนั้นจะถูกผลิตออกมาทางชั้นข้อมูลออก (Output layer) และชั้นอื่นๆ นอกจากชั้นทั้งสองที่กล่าวมาแล้วเราจะเรียกว่าเป็น Hidden layer ซึ่งเปรียบเสมือนกล่องดำ (Black Box) ในระบบ ดังแสดงในรูปที่ 5



รูปที่ 5 ตัวอย่างของนิวรอลที่มีหลายชั้น

การที่ข้อมูลออกของแต่ละชั้นถูกส่งต่อไปยังชั้นถัดไปเรื่อยๆเช่นนี้เราเรียกว่า feedforward network และถ้าข้อมูลออกสามารถถูกดึงนำมาใช้เป็นข้อมูลเข้าของเครือข่ายได้อีกเราเรียกว่า feedback network ดังแสดงในรูปที่ 6



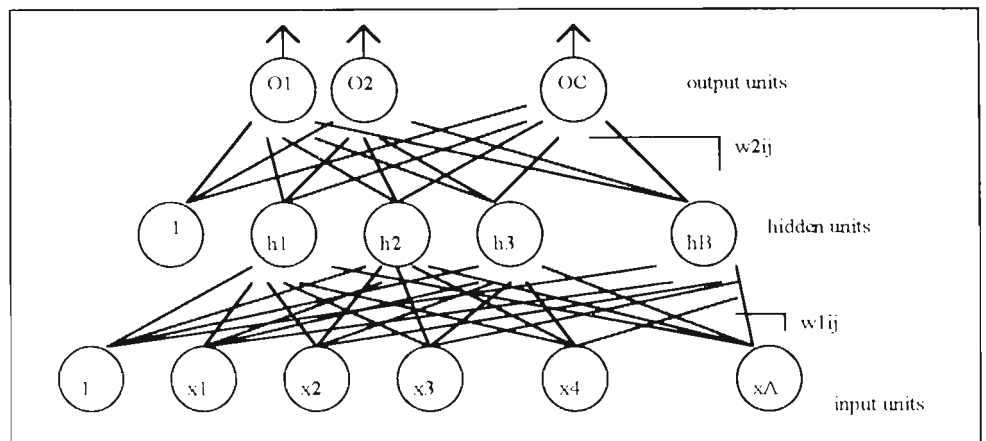
รูปที่ 6 เครือข่ายที่มีการ feedback

Back-Propagation อัลกอริทึม และการรู้จำตัวอักษรไทย

Back-propagation จัดได้ว่าเป็นเทคนิคที่มีประสิทธิภาพและได้รับความนิยมเป็นอย่างมาก มีลักษณะเป็นเครือข่ายแบบหลายชั้นดังแสดงในรูปที่ 7. เทคนิคนี้ได้ถูกค้นพบโดย Werbos; Parker; Rumelhart, hilton และ willians ซึ่งทั้งหมดนี้ได้ทำการค้นคว้าโดยอิสระต่อกันแต่ได้สร้างผลงานออกมาในเวลาใกล้เคียงกัน

Back-propagation สามารถเรียนรู้จากความผิดพลาดโดยใช้ Delta Rule ซึ่งมีการทำงานประกอบไปด้วยสองส่วนใหญ่ว่าด้วยกันคือ

1. ขณะเมื่อนำข้อมูลเข้าสู่ระบบ จะมีการคำนวณในลักษณะที่เคลื่อนที่ไปข้างหน้า (feedforward) เพื่อหาผลลัพธ์ของแต่ละนิวรอน ผลลัพธ์ที่ได้จากการคำนวณนี้จะถูกนำมาเปรียบเทียบกับเป้าหมายที่ต้องการเพื่อหาค่าความผิดพลาด
2. ค่าความผิดพลาดที่คำนวณได้จากข้อ 1 จะถูกนำมาแพร่กระจายย้อนกลับไปเพื่อทำการปรับค่าน้ำหนักทั้งหมดของระบบเครือข่ายนิวรอนนี้



รูปที่ 7. เครือข่ายแบบ Back-propagation

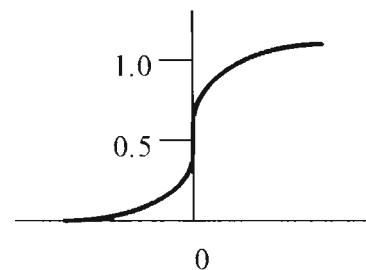
การพัฒนาโปรแกรมสำหรับการรู้จำตัวอักษรไทย

ในการที่จะนำหลักการของ Back-propagation มาประยุกต์ใช้กับการรู้จำตัวอักษรไทยนั้น ได้ทำการกำหนดส่วนประกอบต่าง ๆ ดังนี้

- ก) ตัวอักษรทุกตัวที่ใช้เป็นข้อมูลเข้าจะถูกเก็บอยู่ในรูปของเมทริกซ์ ขนาด 8×20
- ข) ใช้ sigmoid function (S-shaped) ในการหาผลลัพธ์ของระบบ โดยสมการของผลลัพธ์ของแต่ละโหนดเขียนได้ดังนี้

$$\text{output} = 1 / (1 + e^{-\text{sum}})$$

โดยที่ sum คือ ผลรวมของ (weight*input) ที่เชื่อมอยู่กับแต่ละโหนด



รูปที่ 8. Sigmoid function

- ค) ผลลัพธ์ที่ได้จากเครือข่าย ถูกกำหนดให้เป็นไปตามรหัสภาษาไทยของสำนักงานมาตรฐานอุตสาหกรรม (สม.อ.) ในรหัสเลขฐาน 2 เช่น ก->10100001 (1A₁₆ หรือ 161)
จากข้อกำหนดข้างต้นได้นำมาสู่การพัฒนาโปรแกรมหาดังนี้

- 1) input layer ซึ่งเป็นส่วนที่รับข้อมูลจะมีทั้งหมด 161 โหนด $((8*20)+1)$ กำหนดให้เป็น X_j เมื่อ $j = 0, \dots, 160$ ส่วน hidden layer จะมีทั้งหมด 41 โหนด $(40+1)$ กำหนดให้เป็น h_j เมื่อ $j = 0, \dots, 40$ และ output layer จะมีทั้งหมด 8 โหนด กำหนดให้เป็น O_j เมื่อ $j = 1, \dots, 8$ และให้ $W1_{ij}$ เป็นค่าน้ำหนักที่เชื่อมระหว่าง input layer กับ hidden layer

โดยที่ $i =$ อันดับของ input units

$j =$ อันดับของ hidden units

ในทำนองเดียวกัน $W2_{ij}$ เป็นค่าน้ำหนักที่เชื่อมระหว่าง hidden layer กับ output layer

โดยที่ $i =$ อันดับของ hidden units

$j =$ อันดับของ output units

- 2) กำหนดค่าเริ่มต้นของ ค่าน้ำหนักที่เชื่อมอยู่ระหว่างโหนดในเครือข่ายนิวรอล โดยการสุ่มตัวเลขระหว่าง -0.1 ถึง 0.1

$$W1_{ij} = \text{random}(-0.1, 0.1) \quad i = 0, \dots, 160 \quad j = 1, \dots, 40$$

$$W2_{ij} = \text{random}(-0.1, 0.1) \quad i = 0, \dots, 40 \quad j = 1, \dots, 8$$

- 3) กำหนดค่า threshold มีค่าเป็น 1 ทั้งใน input และ hidden layer
 4) เลือกคู่ Input-Output ที่จะทำการสอน
 5) หาค่าของแต่ละโหนดใน hidden layer โดยใช้สมการข้างล่างนี้

$$h_j = \frac{1}{1 + e^{-\sum_{i=0}^{160} W1_{ij} X_i}} \quad j = 1, \dots, 40$$

($W1_{0j}$ คือค่าน้ำหนักของ threshold unit สำหรับ hidden unit j)

- 6) หาค่าของแต่ละโหนดใน output layer โดยใช้สมการข้างล่างนี้

$$O_j = \frac{1}{1 + e^{-\sum_{i=0}^{40} W2_{ij} h_i}} \quad j = 1, \dots, 8$$

($W2_{0j}$ คือค่าตัวคูณของ threshold unit สำหรับ output unit j)

- 7) คำนวณค่า error ของแต่ละโหนดใน output layer ตามสมการ

$$E2_j = O_j (1 - O_j)(Y_j - O_j) \quad j = 1, \dots, 8$$

โดยที่ $O_j =$ actual output

$Y_j =$ target output

- 8) คำนวณค่า error ของแต่ละโหนดใน hidden layer

$$E1_j = h_j (1-h_j) e^{-\sum_{i=1}^8 E2_j * W2_{ij}} \quad j = 1, \dots, 40$$

โดยที่ h_j = actual output (ของ hidden layer)

- 9) ปรับค่าน้ำหนักระหว่าง hidden กับ Output layer

$$\Delta W2_{ij} (t+1) = n.E2_j. h_j + \alpha \Delta W2_{ij} (t) \quad \begin{matrix} i = 0, \dots, 40 \\ j = 1, \dots, 8 \end{matrix}$$

โดยที่ n = learning rate

α = momentum term

- 10) ปรับค่าน้ำหนักระหว่าง input layer กับ hidden layer

$$\Delta W1_{ij} (t+1) = n.E1_j. X_j + \alpha \Delta W1_{ij} (t) \quad \begin{matrix} i = 0, \dots, 160 \\ j = 1, \dots, 40 \end{matrix}$$

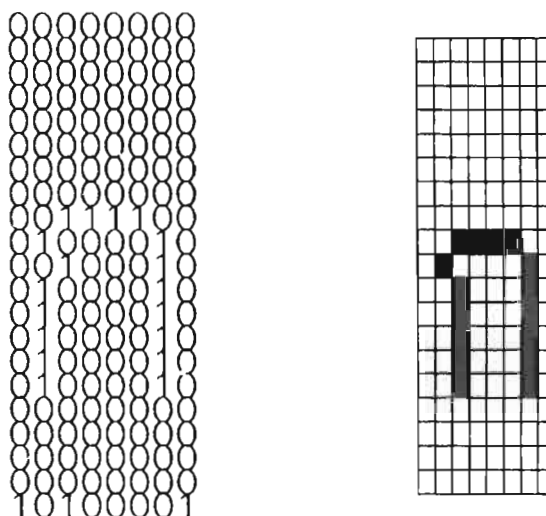
- 11) กลับไปทำขั้นตอนที่ 4 จนกระทั่งคู่ Input-output ที่ต้องการได้รับการสอนเสร็จครบหมดทุกคู่ ก็ถือว่าเป็นการสอน 1 ครั้ง ซึ่งในการสอนนี้จะต้องทำการสอนหลายๆครั้งด้วยข้อมูลที่แตกต่างกันจึงจะได้ผลลัพธ์ของเครือข่ายที่สามารถทำงานได้ตามความต้องการ

ผลการทดลอง

ในหัวข้อนี้จะได้กล่าวถึงการทดสอบระบบเครือข่ายนิวรอลที่ถูกพัฒนาขึ้นโดยใช้อัลกอริทึมของ Back-propagation ที่ได้อธิบายไปแล้วในข้อ 3 ในการทดลองนี้จะประกอบไปด้วย 2 ส่วนด้วยกันคือ การสอนเครือข่ายนิวรอลให้รู้ตัวอักษรไทย และการทดสอบระบบเครือข่ายที่ได้รับการสอนเรียบร้อยแล้ว

การสอนเครือข่ายนิวรอล

เพื่อให้เครือข่ายนิวรอลสามารถเรียนรู้ตัวอักษรไทยจึงได้มีการสอนเครือข่ายนิวรอลด้วยข้อมูลภาพตัวอักษรไทยทั้งหมด 46 ตัวซึ่งแต่ละตัวมีขนาดเท่ากับ 20x8 ดังตัวอย่างที่แสดงในรูปที่ 9.



รูปที่ 9. ตัวอย่างรูปแบบข้อมูลเข้า

แถวสุดท้าย (ล่างสุด) ที่แสดงในรูปที่ 9 เป็นตัวบอกผลลัพธ์ที่ต้องการจากระบบ เพื่อให้เครือข่ายนิวรอลทำการปรับค่าน้ำหนักที่เชื่อมต่อกันระหว่างโหนดต่างๆ ให้ได้ผลลัพธ์ตรงตามแถวสุดท้ายนี้ ข้อมูลที่ใช้ในการสอนระบบเครือข่ายนิวรอลนี้มีด้วยกันหลายกลุ่มด้วยกันดังนี้

1. แบบที่ 1 ตัวอักษรที่สมบูรณ์ปราศจากสัญญาณรบกวนดังตัวอย่างที่แสดงใน รูปที่ 10
2. แบบที่ 2 ตัวอักษรที่มีสัญญาณรบกวนที่ทำให้ตำแหน่งที่เป็น 1 หายไป 1 ตำแหน่ง ดังตัวอย่างที่แสดงในรูปที่ 11

3. แบบที่ 3 ตัวอักษรที่มีสัญญาณรบกวนที่ทำให้ตำแหน่งที่เป็น 1 หายไป 2 ตำแหน่ง ดังตัวอย่างที่แสดงในรูปที่ 12
4. แบบที่ 4 ตัวอักษรที่มีสัญญาณรบกวนที่ทำให้ตำแหน่งที่เป็น 1 หายไป 3 ตำแหน่ง ดังตัวอย่างที่แสดงในรูปที่ 13
5. แบบที่ 5 ตัวอักษรที่มีสัญญาณรบกวนที่ทำให้ตำแหน่งที่เป็น 1 เพิ่มขึ้น 1 ตำแหน่ง ดังตัวอย่างที่แสดงในรูปที่ 14
6. แบบที่ 6 ตัวอักษรที่มีสัญญาณรบกวนที่ทำให้ตำแหน่งที่เป็น 1 เพิ่มขึ้น 2 ตำแหน่ง ดังตัวอย่างที่แสดงในรูปที่ 15
7. แบบที่ 7 ตัวอักษรที่มีสัญญาณรบกวนที่ทำให้ตำแหน่งที่เป็น 1 เพิ่มขึ้น 3 ตำแหน่ง ดังตัวอย่างที่แสดงในรูปที่ 16

การทดสอบ

เมื่อได้ทำการสอนเครือข่ายนิวรอลเรียบร้อยแล้วก็ได้ทำการทดสอบ ดังนี้

1. ใช้ข้อมูลที่สอนนิวรอลเน็ตเวิร์กทั้งหมดในข้อ 4.1 คือแบบที่ 1 ถึง 7
2. ใช้ข้อมูลใหม่คือ N1 และ N2 ดังตัวอย่างที่แสดงในรูปที่ 16 และ 17 ตามลำดับ

00000000	00000000	00000000	00000000	00000000	00000000	00000000
00000000	00000000	00000000	00000000	00000000	00000000	00000000
00000000	00000000	00000000	00000000	00000000	00000000	00000000
00000000	00000000	00000000	00000000	00000000	00000000	00000000
00000000	00000000	00000000	00000000	00000000	00000000	00000000
00000000	00000000	00000000	00000000	00000000	00000000	00000000
00000000	00000000	00000000	00000000	00000000	00000000	00000000
00000000	00000000	00000000	00000000	00000000	00000000	00000000
00000000	00000000	00000000	00000000	00000000	00000000	00000000
00100100	10110100	00111100	01101100	10100010	00001000	
01010100	01010100	10000010	10000010	11010010	00010100	
00110100	00010100	10000010	10000010	00110010	00001100	
00010100	00010100	10010010	10010010	00100010	00000100	
00010100	00010100	10101010	10101010	01110010	01000100	
00010100	00010100	11010010	11010010	10101010	00100100	
00010100	00010100	10000010	10000010	10100110	00010100	
00010000	00011000	10000010	10000010	01100010	00001000	
00000000	00000000	00000000	00000000	00000000	00000000	
00000000	00000000	00000000	00000000	00000000	00000000	
00000000	00000000	00000000	00000000	00000000	00000000	
00000000	00000000	00000000	00000000	00000000	00000000	
10100010	10100011	10100100	10100101	10100110	10100111	
00000000	00000000	00000000	00000000	00000000	00000000	
00000000	00000000	00000000	00000000	00000000	00000000	
00000000	00000000	00000000	00000000	00000000	00000000	
00000000	00000000	00000000	00000000	00000000	00000000	
00000000	00000000	00000000	00000000	00000000	00000000	
00000010	00000010	00000000	00000000	00000000	00000000	
00000100	00000100	00000000	00000000	00000000	00000000	
00101000	10101000	01110010	01110010	01111100	01111100	
01010100	01010100	10001010	10001010	10000010	10000010	
00010100	00010100	01001010	01001010	01100010	01000010	
00010100	00100100	10001010	10001010	00100010	00100010	
00010100	00100100	10001010	10001010	00100010	00100010	
00010100	00100100	10011010	11001010	01100010	01100010	
00010100	00101000	10101110	10101010	10100010	10100010	
00011000	00110000	11010010	01101110	01000011	01000010	
00000000	00000000	00000000	00000000	00010110	00000010	
00000000	00000000	00000000	00001000	00101010	10101010	
00000000	00000000	00000000	00001110	00110110	01010110	
00000000	00000000	00000000	00000000	00000000	00000000	
10101010	10101011	10101100	10101101	10101110	10101111	

รูปที่ 11 ตัวอย่างตัวอักษรแบบที่ 2

	00000000	00000000	00000000	00000000	00000000	00000000
	00000000	00000000	00000000	00000000	00000000	00000000
	00000000	00000000	00000000	00000000	00000000	00000000
	00000000	00000000	00000000	00000000	00000000	00000000
	00000000	00000000	00000000	00000000	00000000	00000000
	00000000	00000000	00000000	00000000	00000000	00000000
	00000000	00000000	00000000	00000000	00000000	00000000
	00000000	00000000	00000000	00000000	00000000	00000000
	00000000	00000000	00000000	00000000	00000000	00000000
	00000100	10100100	00111100	01101100	10110010	00001000
	01010100	11010100	10000010	10000010	10010010	00010100
	00110100	00010100	10000010	10000010	00110010	00000000
	00010100	00010100	10010010	10000010	00100010	00000100
	00010100	00010100	10101010	10101010	01110010	01000100
	00010100	00010100	11010010	11010010	10101010	00100100
	00010100	00010100	10000010	10000010	10100110	00010100
	00001000	00001000	00000010	00000010	01000010	00001100
	00000000	00000000	00000000	00000000	00000000	00000000
	00000000	00000000	00000000	00000000	00000000	00000000
	00000000	00000000	00000000	00000000	00000000	00000000
	00000000	00000000	00000000	00000000	00000000	00000000
	00000000	00000000	00000000	00000000	00000000	00000000
	10100010	10100011	10100100	10100101	10100110	10100111

	00000000	00000000	00000000	00000000	00000000	00000000
	00000000	00000000	00000000	00000000	00000000	00000000
	00000000	00000000	00000000	00000000	00000000	00000000
	00000000	00000000	00000000	00000000	00000000	00000000
	00000000	00000000	00000000	00000000	00000000	00000000
	00000000	00000000	00000000	00000000	00000000	00000000
	00000010	00000010	00000000	00000000	00000000	00000000
	00000100	00000100	00000000	00000000	00000000	00000000
	00100000	10101000	01110010	01110010	01111100	01111100
	01010100	10010100	10001010	00001010	10000010	10000010
	00110100	00010100	01001010	01001010	01100010	01000010
	00010100	00100100	10001010	10001010	00100010	00100010
	00010100	00100100	10001010	10001010	00100010	00100010
	00010100	00100100	10011010	11001010	01100010	01100010
	00010100	00101000	10101110	10001010	10100010	10100010
	00001000	00010000	10011000	11101110	01000011	01000010
	00000000	00000000	00000000	00000000	00010110	00000010
	00000000	00000000	00000000	00001000	00101010	10101010
	00000000	00000000	00000000	00001110	00010110	01010100
	00000000	00000000	00000000	00000000	00000000	00000000
	10101010	10101011	10101100	10101101	10101110	10101111

รูปที่ 12 ตัวอย่างตัวอักษรแบบที่ 3

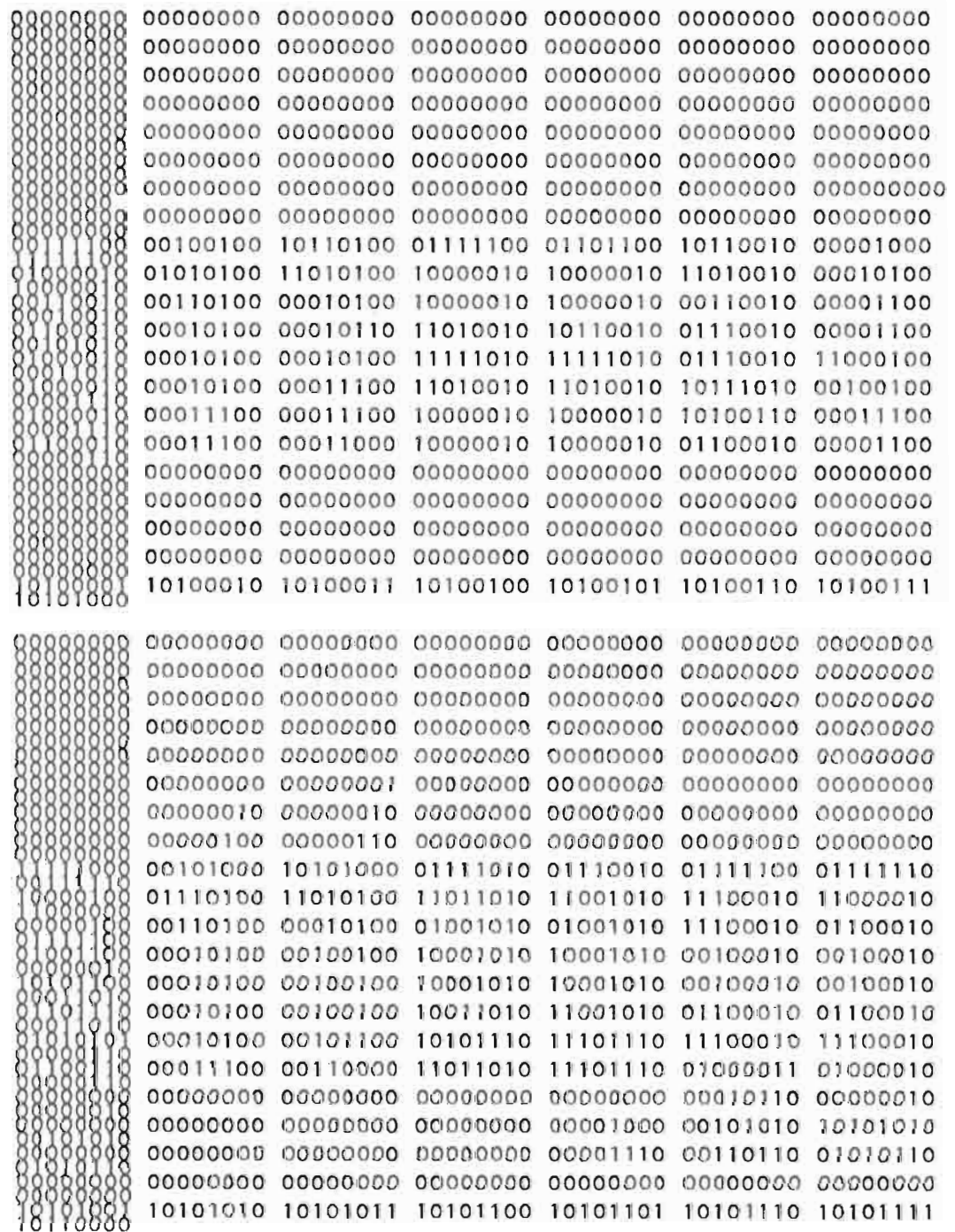
0	00000000	00000000	00000000	00000000	00000000	00000000
0	00000000	00000000	00000000	00000000	00000000	00000000
0	00000000	00000000	00000000	00000000	00000000	00000000
0	00000000	00000000	00000000	00000000	00000000	00000000
0	00000000	00000000	00000000	00000000	00000000	00000000
0	00000000	00000000	00000000	00000000	00000000	00000000
0	00000000	00000000	00000000	00000000	00000000	00000000
0	00000000	00000000	00000000	00000000	00000000	00000000
0	00000000	00000000	00000000	00000000	00000000	00000000
0	00100000	10100100	01111100	01101100	10110010	00001000
0	01010100	11010100	10000010	10010010	01010010	00010100
0	00100100	00010100	10000010	10000010	00010010	00000100
0	00010100	00010100	10010010	10000010	00100010	00000100
0	00010100	00010100	10101010	10101010	01100010	01000100
0	00010100	00010100	01000010	01010010	10101010	00000100
0	00010100	00010100	10000010	10000010	10100110	00010100
0	00001000	00000000	10000000	10000000	01100010	00001000
0	00000000	00000000	00000000	00000000	00000000	00000000
0	00000000	00000000	00000000	00000000	00000000	00000000
0	00000000	00000000	00000000	00000000	00000000	00000000
0	00000000	00000000	00000000	00000000	00000000	00000000
1	10100010	10100011	10100100	10100101	10100110	10100111
0	00000000	00000000	00000000	00000000	00000000	00000000
0	00000000	00000000	00000000	00000000	00000000	00000000
0	00000000	00000000	00000000	00000000	00000000	00000000
0	00000000	00000000	00000000	00000000	00000000	00000000
0	00000000	00000000	00000000	00000000	00000000	00000000
0	00000010	00000010	00000000	00000000	00000000	00000000
0	00000100	00000100	00000000	00000000	00000000	00000000
0	00101000	00100000	01110010	01110010	01111100	01111100
0	01000100	11010100	10001010	10001010	10000010	10000010
0	00010100	00010100	01001010	01001010	01000010	01000010
0	00010100	00100100	10001010	10001010	00100010	00100010
0	00010100	00100100	10001010	10001010	00100010	00100010
0	00010100	00100100	10001010	01001010	00100010	00100010
0	00010100	00101000	10101010	10101010	10100010	10100010
0	00001000	00010000	11010010	01001110	01000011	01000010
0	00000000	00000000	00000000	00000000	00010110	00000010
0	00000000	00000000	00000000	00001000	00101010	10001010
0	00000000	00000000	00000000	00001110	00101010	01010110
0	00000000	00000000	00000000	00000000	00000000	00000000
1	10101010	10101011	10101100	10101101	10101110	10101111

รูปที่ 13 ตัวอย่างตัวอักษรแบบที่ 4

รูปที่ 14 ตัวอย่างตัวอักษรแบบที่ 5

00000000	00000000	00000000	00000000	00000000	00000000	00000000
00000000	00000000	00000000	00000000	00000000	00000000	00000000
00000000	00000000	00000000	00000000	00000000	00000000	00000000
00000000	00000000	00000000	00000000	00000000	00000000	00000000
00000000	00000000	00000000	00000000	00000000	00000000	00000000
00000000	00000000	00000000	00000000	00000000	00000000	00000000
00000000	00000000	00000000	00000000	00000000	00000000	00000000
00000000	00000000	00000000	00000000	00000000	00000000	00000000
00000000	00000000	00000000	00000000	00000000	00000000	00000000
00100100	10110100	01111100	01101100	10110010	00001000	
01010100	11010100	10000010	10000010	11010010	00010100	
00110100	00010100	10000010	10000010	00110010	00001100	
00010100	00010100	11010010	10110010	01110010	00001100	
00010100	00010100	11101010	11101010	01110010	01000100	
00010100	00011100	11010010	11010010	10101010	00100100	
00011100	00011100	10000010	10000010	10100110	00011100	
00011100	00011000	10000010	10000010	01100010	00001100	
00000000	00000000	00000000	00000000	00000000	00000000	
00000000	00000000	00000000	00000000	00000000	00000000	
00000000	00000000	00000000	00000000	00000000	00000000	
00000000	00000000	00000000	00000000	00000000	00000000	
10100010	10100011	10100100	10100101	10100110	10100111	
00000000	00000000	00000000	00000000	00000000	00000000	
00000000	00000000	00000000	00000000	00000000	00000000	
00000000	00000000	00000000	00000000	00000000	00000000	
00000000	00000000	00000000	00000000	00000000	00000000	
00000000	00000000	00000000	00000000	00000000	00000000	
00000010	00000010	00000000	00000000	00000000	00000000	
00000100	00000110	00000000	00000000	00000000	00000000	
00101100	10101000	01111010	01110010	01111100	01111110	
01010100	11010100	10011010	11001010	11100010	11000010	
00110100	00010100	01001010	01001010	11100010	01100010	
00010100	00100100	10001010	10001010	00100010	00100010	
00010100	00100100	10001010	10001010	00100010	00100010	
00010100	00100100	10011010	11001010	01100010	01100010	
00010100	00101100	10101110	11101010	10100010	10100010	
00011100	00110000	11011010	11101110	01000011	01000010	
00000000	00000000	00000000	00000000	00010110	00000010	
00000000	00000000	00000000	00001000	00101010	10101010	
00000000	00000000	00000000	00001110	00110110	01010110	
00000000	00000000	00000000	00000000	00000000	00000000	
10101010	10101011	10101100	10101101	10101110	10101111	

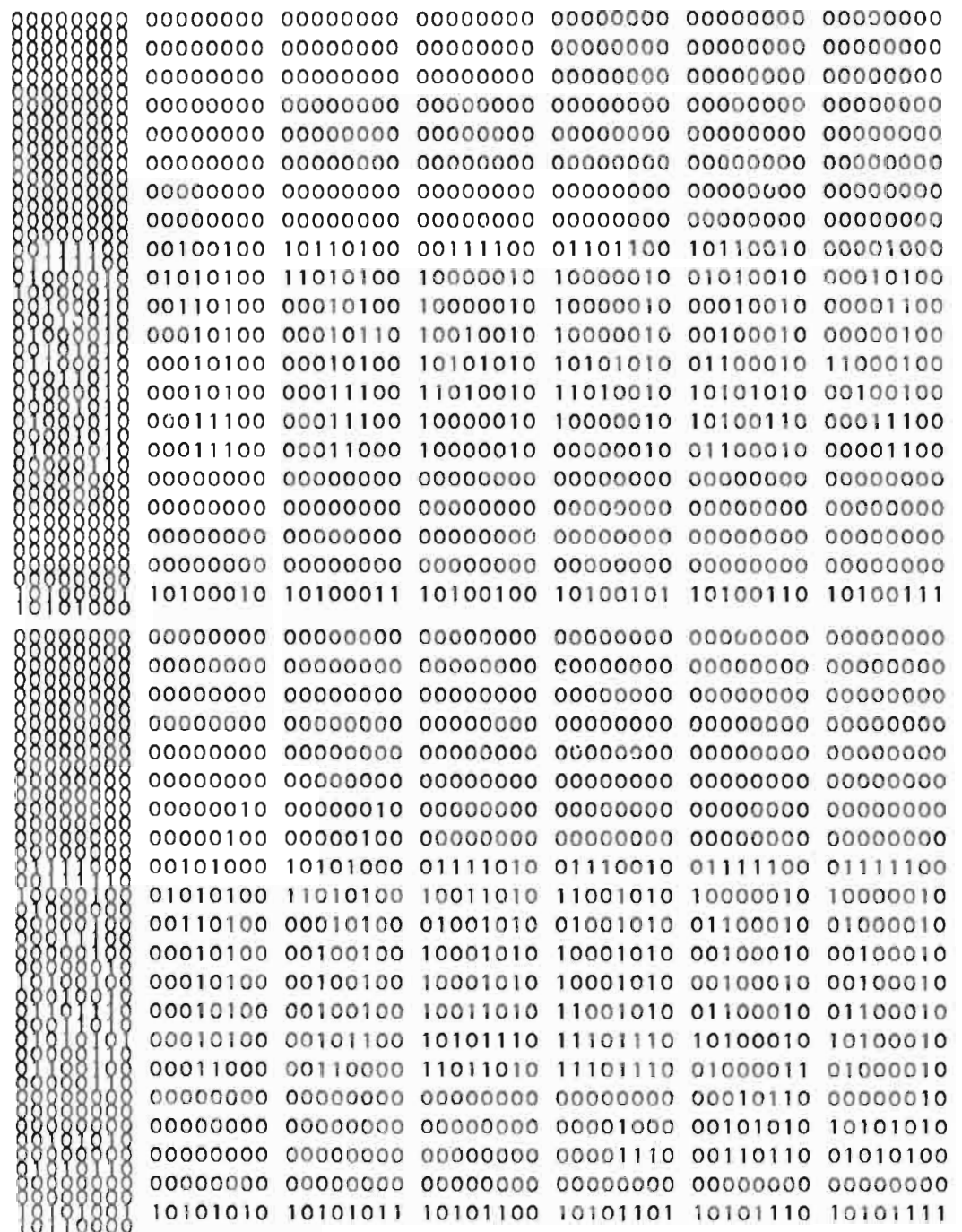
รูปที่ 15 ตัวอย่างตัวอักษรแบบที่ 6



รูปที่ 16 ตัวอย่างตัวอักษรแบบที่ 7

00000000	00000000	00000000	00000000	00000000	00000000
00000000	00000000	00000000	00000000	00000000	00000000
00000000	00000000	00000000	00000000	00000000	00000000
00000000	00000000	00000000	00000000	00000000	00000000
00000000	00000000	00000000	00000000	00000000	00000000
00000000	00000000	00000000	00000000	00000000	00000000
00000000	00000000	00000000	00000000	00000000	00000000
00000000	00000000	00000000	00000000	00000000	00000000
00000000	00000000	00000000	00000000	00000000	00000000
00100100	10110100	01111100	01101100	10110010	00001000
01010100	11010100	10000010	10000010	11010010	00010100
00110100	00010100	10000010	10000010	00110010	00001100
00110100	00010100	10010010	10010010	00100010	00000100
00010100	00010100	10101010	10101010	01110010	11000100
00010100	00010100	11010010	11010010	10101010	00100100
00010100	00010100	11000010	10000010	10100110	00011100
00011000	00011100	10000010	11000010	01100110	00001100
00000000	00000000	00000000	00000000	00000000	00000000
00000000	00000000	00000000	00000000	00000000	00000000
00000000	00000000	00000000	00000000	00000000	00000000
00000000	00000000	00000000	00000000	00000000	00000000
10100010	10100011	10100100	10100101	10100110	10100111
00000000	00000000	00000000	00000000	00000000	00000000
00000000	00000000	00000000	00000000	00000000	00000000
00000000	00000000	00000000	00000000	00000000	00000000
00000000	00000000	00000000	00000000	00000000	00000000
00000000	00000000	00000000	00000000	00000000	00000000
00000000	00000000	00000000	00000000	00000000	00000000
00000010	00000010	00000000	00000000	00000000	00000000
00000100	00000100	00000000	00000000	00000000	00000000
00101100	10101000	01110010	01110010	01111100	01111100
01010100	11010100	10001010	10001010	10000010	10000010
00110100	00010100	01001010	01001010	11100010	01100010
00010100	00100100	10001010	10001010	00100010	00100010
00010100	00100100	10001010	10001010	00100010	00100010
00010100	00100100	11011010	11001010	01100010	01100010
00010100	00101000	10101110	10101010	10100010	10100010
00011000	00111000	11011010	11101110	01000011	01100010
00000000	00000000	00000000	00000000	00010110	00000010
00000000	00000000	00000000	00001000	01101010	10101010
00000000	00000000	00000000	00001111	00110110	01010110
00000000	00000000	00000000	00000000	00000000	00000000
10101010	10101011	10101100	10101101	10101110	10101111

รูปที่ 17 ตัวอย่างตัวอักษรแบบที่ N1



รูปที่ 18 ตัวอย่างตัวอักษรแบบที่ N2

ผลการทดสอบโดยใช้ค่า learing rate=0.35 momentum=0.9 และ error=0.1 ได้แสดงในตารางที่ 1

ตารางที่1 ผลของการทดลอง

ข้อมูลที่ใช้ในการทดสอบ	ความถูกต้อง (%)
แบบที่ 1	100
แบบที่ 2	100
แบบที่ 3	100
แบบที่ 4	97.82
แบบที่ 5	100
แบบที่ 6	100
แบบที่ 7	100
ข้อมูลใหม่ 1	95.65
ข้อมูลใหม่ 2	100
ค่าเฉลี่ย	98.7

สรุปและข้อเสนอแนะ

ผลการทดลองพบว่า ระบบเครือข่ายนิวรอลที่สามารถตรวจรู้ตัวอักษรไทยได้ความถูกต้องเฉลี่ยถึง 98.7% และให้ความถูกต้องถึง 99.6% สำหรับข้อมูลที่ได้รับการสอนไว้แล้ว และ 97.8% สำหรับข้อมูลที่ยังไม่ได้สอนให้กับเครือข่าย จะเห็นได้ว่าผลที่ได้รับเป็นที่น่าพอใจ แต่ในการที่จะนำไปใช้ในการประยุกต์ใช้งานจริงนั้น ยังจะต้องพัฒนาส่วนอื่น ๆ เพิ่มเข้ามาด้วย โดยเฉพาะอย่างยิ่งส่วนของ preprocessing คือ การรับค่าจากเครื่องกวาดภาพ (scanner) โดยตรง จะต้องสามารถแยกข้อมูลของแต่ละบรรทัดออกจากกัน จากนั้นก็จะต้องทำการแยกตัวอักษรออกจากกันทีละตัว เพื่อนำมาแปลงให้อยู่ในรูปที่จะสามารถป้อนให้กับเครือข่ายนิวรอลได้ นอกจากนั้นแล้วถ้าหากบรรทัดที่ได้จากเครื่องกวาดภาพนั้นเองก็จะต้องมีการปรับภาพกลับให้อยู่ในแนวระดับเสียก่อน ซึ่งส่วนที่กล่าวมาแล้วนี้กำลังอยู่ในระหว่างการพัฒนา เพื่อให้สามารถนำมาใช้งานได้อย่างมีประสิทธิภาพต่อไป

บรรณานุกรม

1) M.M. Nelson and W.T. Illingworth, "A Practical Guide to Neural Nets", Addison Wesley, 1991

2) J. Hertz, A. Krogh and R.G. Palmer, "Introduction to the Theory of Neural Computation", Addison Wesley, 1991

- 3) S.Y. Kung, "***Digital Neural Networks***", Prentice Hall, 1993
- 4) J.A. Freeman and D.M. Skapura, "***Neural Networks - Algorithms, Applications, and Programming Techniques***", Addison Wesley, 1992
- 5) J.A. Freeman, "***Simulating Neural Networks with Mathematica***", Addison Wesley, 1994