การประยุกต์ใช้นิวรอลเน็ตเวอร์กเพื่อจำแนกลายวงจรบนแผ่นวงจรพิมพ์

Application of a Neural Network for Classification of PCB's

ยุทธชัย นิลแพทย์

Yutthachai Nilpat

วิทยานิพนธ์วิศวกรรมศาสตรมหาบัณฑิต สาขาวิชาวิศวกรรมไฟฟ้า

มหาวิทยาลัยสงขลานครินทร์

Master of Engineering Thesis in Electrical Engineering

Prince of Songkla University

2544

ชื่อวิทยานิพนธ์ การประยุกต์ใช้นิวรอลเน็ตเวอร์กเพื่อจำแนกลายวงจรบนแผ่นวงจรพิมพ์

ผู้เขียน นายยุทธชัย นิลแพทย์

สาขาวิชา วิศวกรรมไฟฟ้า

คณะกรรมการที่ปรึกษา

.....ประธานกรรมการ
(อาจารย์ปราโมทย์ จูฑาพร)

.....กรรมการ
(ผู้ช่วยศาสตราจารย์ ดร.ชูศักดิ์ ลิ่มสกุล)

.....กรรมการ
(ผู้ช่วยศาสตราจารย์ เลียง คูบุรัตถ์)

คณะกรรมการสอบ

.....ประธานกรรมการ
(อาจารย์ปราโมทย์ จูฑาพร)

.....กรรมการ
(ผู้ช่วยศาสตราจารย์ ดร.ชูศักดิ์ ลิ่มสกุล)

.....กรรมการ
(ผู้ช่วยศาสตราจารย์ เลียง คูบุรัตถ์)

.....กรรมการ
(ผู้ช่วยศาสตราจารย์ ดร.กิตติพัฒน์ ตันตระรุ่งโรจน์)

.....กรรมการ
(อาจารย์ วีระพันธุ์ มุสิกสาร)

บัณฑิตวิทยาลัย มหาวิทยาลัยสงขลานครินทร์ อนุมัติให้นับวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรมหาบัณฑิต สาขาวิชาวิศวกรรมไฟฟ้า

.....
(รองศาสตราจารย์ ดร.ปิติ ทฤษฎิคุณ)
คณบดีบัณฑิตวิทยาลัย

ชื่อวิทยานิพนธ์ การประยุกต์ใช้นิวรอลเน็ตเวอร์กเพื่อจำแนกลายวงจรบนแผ่นวงจรพิมพ์

ผู้เขียน นายยุทธชัย นิลแพทย์

สาขาวิชา วิศวกรรมไฟฟ้า

ปีการศึกษา 2543

บทคัดย่อ

วิทยานิพนธ์ฉบับนี้ เป็นรายงานการศึกษาวิจัยวิธีการจำแนกลายวงจรพิมพ์ โดยนำนิวรอลเน็ตเวอร์กมาประยุกต์ใช้ในการจำแนกลายวงจรพิมพ์ ในการศึกษาวิจัยมีการลำดับการทำงานดังนี้คือ รับภาพลายวงจรพิมพ์ โดยจะรับภาพเข้ามาแบบ bitmap file โดยกล้อง CCD ขนาดของภาพที่ได้คือ 320 x 240 จุด แล้วจึงนำภาพที่ได้ผ่านกระบวนการปรับปรุงภาพอย่างง่ายนั่นคือทำการเพิ่มหรือลดความสว่างของภาพ การกระทำดังกล่าวจะขึ้นอยู่กับภาพที่รับมาได้ว่าเป็นอย่างไร จากนั้นจึงทำการตัดภาพพื้นหลังทิ้งให้เหลือภาพในแบบขาว-ดำ โดยใช้ภาพลายวงจรทั้งหมด 8 แบบ แบ่งเป็น 2 ประเภทคือลายวงจรที่เป็นลายทองแดงกับแบบที่ทำการเคลือบลายทองแดงแล้ว โดยใช้ใช้นิวรอลเน็ตเวอร์กแบบแพร่กลับทำการเรียนรู้ และสุดท้ายให้นิวรอลเน็ตเวอร์กทำการจำแนก การใช้นิวรอลเน็ตเวอร์กในการศึกษานี้ได้ทำการปรับจำนวนของชั้นซ่อนและจำนวนเซลล์ในแต่ละชั้นซ่อนโดยใช้ค่าที่ต่างๆ กันด้วย การปรับค่าต่าง ๆ ภายในนิวรอลเน็ตเวอร์กเพื่อศึกษาว่าค่าองค์ประกอบของนิวรอลเน็ตเวอร์กแบบใดที่จะให้ผลการจำแนกได้ดี

ก่อนที่จะใช้นิวรอลเน็ตเวอร์กกับลายวงจรพิมพ์จริงได้ทำการทดลองรับภาพลายวงจรพิมพ์บนกระดาษแล้วพบว่าอัลกอริทึมสามารถจำแนกได้ ผลที่ได้จากการจำแนกพบว่านิวรอลเน็ตเวอร์กแบบแพร่กลับ โดยใช้ชั้นซ่อน 1 ชั้น มีจำนวนเซลล์ในชั้นซ่อน 10,000 และ 50,000 เซลล์ ในกรณีที่ภาพลายวงจรเป็นลายเส้นทองแดงขนาดไม่น้อยกว่า 1/32 นิ้ว สามารถจำแนกลายวงจรพิมพ์ได้ค่าความถูกต้องได้ 83.3 เปอร์เซ็นต์ ส่วนภาพลายวงจรที่ทำการเคลือบสารสะท้อนแสงแล้ว ที่ 1 ชั้นซ่อน มีจำนวนเซลล์ 10,000 และ 50,000 เซลล์ จะได้ความถูกต้องอยู่ที่ 56.7 เปอร์เซ็นต์ เพื่อให้มีความถูกต้องสูงขึ้น ต้องทำการเรียนรู้กับจำนวนจุดผิดพลาดสูงขึ้น โดยต้องใช้จุดผิดพลาดในการเรียนรู้สูงกว่า 23040 จุด

Thesis Title Application of a Neural Network for Classification of PCB's

Author Mr.Yutthachai Nilpat

Major Program Electrical Engineering

Academic Year 2000

Abstract

This thesis presents a procedure for classification of printed circuit boards (PCB's) through the application of neural networks. A PCB image would first be captured through a CCD and input to a computer as a bitmap file with a resolution of 320x240 pixels. A simple enhancement was then applied to the image using contrast adjustment. The image was then extracted for its copper traces, abandoning the image background, and saved as a black and white image. Eight samples of both coated (glossy) and non-coated (non-glossy) PCB's were used in the study. A back-propagation neural network was used in the training stage. Experiments were carried out using neural network with many different hidden layers and attempts were made to fine-tune the neural network for best performance.

It was found success when we test with image in paper before used neural network with PCB's. It was found that a 10,000-cells, 50,000-cells neural network with a single hidden layer gave the best performance. For non-coated (non-glossy) PCB's, a classification success rate of 83.3% was obtained. For coated (glossy) PCB's, a classification success rate of only 56.7% could be achieved. You should be used incorrect pixel more than 23040 pixel for high classification success rate.

กิตติกรรมประกาศ

ขอแสดงคำขอบพระคุณ อาจารย์ปราโมทย์ จูฑาพร ประธานกรรมการที่ปรึกษาที่ได้กรุณาให้การสนับสนุนในด้านต่างๆเป็นอย่างดีไม่ว่าจะเป็นการให้คำปรึกษา การแนะนำความรู้ในด้านต่างๆ เอกสารข้อมูล อุปกรณ์ในการทำวิจัยต่างๆ รวมทั้งกำลังใจในการแก้ปัญหาตลอดจนช่วยตรวจแก้ไขวิทยานิพนธ์ให้ดำเนินไปอย่างสมบูรณ์

ขอขอบพระคุณ ผู้ช่วยศาสตราจารย์ ดร.ชูศักดิ์ ลิ่มสกุล และผู้ช่วยศาสตราจารย์ เลียง คูบุรัตถ์ ที่ได้ให้คำแนะนำที่เป็นประโยชน์ต่อการวิจัยและการช่วยเหลือในการจัดหาอุปกรณ์ต่างๆ สำหรับการทำวิจัยตลอดจนช่วยตรวจแก้ไขวิทยานิพนธ์ให้ดำเนินไปอย่างสมบูรณ์

ขอบพระคุณอาจารย์ผู้สอบวิทยานิพนธ์ทุกท่านที่ได้ให้ข้อคิดเห็นและข้อแนะนำที่ดีเพื่อจะนำไปใช้ในโอกาสต่อไป

ขอขอบพระคุณ อาจารย์ และ บุคลากรในภาควิชาวิศวกรรมไฟฟ้าทุกท่านที่ให้คำปรึกษาและความช่วยเหลือในด้านต่างๆที่สำคัญจนสำเร็จลุล่วง

ขอขอบพระคุณ บัณฑิตวิทยาลัย มหาวิทยาลัยสงขลานครินทร์ วิทยาเขตหาดใหญ่ ที่ให้การสนับสนุนทุนในการทำวิจัย

ขอขอบคุณ เพื่อนและรุ่นพี่นักศึกษาปริญญาโทภาควิชาวิศวกรรมไฟฟ้าทุกท่านที่ได้ให้คำแนะนำ คำปรึกษาและกำลังใจเป็นอย่างดีมาโดยตลอด

และที่สำคัญที่สุด ข้าพเจ้าขอน้อมรำลึกถึงพระคุณของ บิดามารดา และครอบครัวที่ส่งเสริมและสนับสนุนข้าพเจ้าในทุกๆเรื่องตลอดมาจนสำเร็จการศึกษา

ยุทธชัย นิลแพทย์

สารบาญ

หน้า

รายการตาราง

ตาราง หน้า

รายการภาพประกอบ

ภาพประกอบ หน้า

ตัวย่อและสัญลักษณ์

α	=	อัตราการเรียนรู้
δ_k	=	ค่าความไวที่จะนำไปคำนวณในส่วนของการแพร่กลับ
Δv_{ij}	=	ค่าน้ำหนักของชั้นซ่อนที่ 1
Δw_{jk}	=	ค่าน้ำหนักของชั้นเอาต์พุต
CCD	=	classical chromodynamics
$F(x)$	=	ค่าเฉลี่ยของความผิดพลาดกำลังสอง
G_0	=	ค่าระดับเทาของวัตถุหลังจากทำ thresholding
G_b	=	ค่าระดับเทาของพื้นหลัง หลังจากทำ thresholding
LMS	=	least mean square
PCB's	=	printed circuit boards
T	=	ค่า threshold
W	=	เมตริกซ์น้ำหนัก
a	=	ค่าเอาต์พุตที่ได้จากการ feedforward ของรอบนั้นๆ
b	=	ค่าไบอัส
brightness	=	ค่าความสว่างที่เพิ่มให้กับภาพ
contrast	=	ค่าตัวเลขที่ใช้ในการเพิ่มความแตกต่างของค่าระดับเทาในภาพให้มีค่าความแตกต่างกันมากขึ้น
e	=	ค่าความผิดพลาด
f'	=	การทำอนุพันธ์
g_i	=	ค่าระดับเทาของ pixel ก่อนผ่านกระบวนการ brightness
S^m_i	=	ความไวที่เปลี่ยนเทอมที่ i ของอินพุตชั้นที่ m
t	=	ค่าเป้าหมาย
V_{0j}	=	ค่าไบอัส
V_{ij}	=	ค่าน้ำหนักในชั้นนั้นนั้น
X_i	=	คือค่าของเซลล์ในชั้นอินพุตจำนวน i เซลล์
Z_in_j	=	ค่าที่ได้จากการคูณน้ำหนักกับอินพุต
Z_j	=	ค่าในชั้นซ่อนที่ต้องการแพร่กลับมีจำนวน j เซลล์

บทที่ 1

บทนำ

1.1 บทนำต้นเรื่อง

การทำงานทุกประเภท จะต้องมีการตรวจสอบคุณภาพของผลผลิตที่ได้จากการทำงานเช่น กุ้งก็ต้องมีการคัดขนาด คอมพิวเตอร์ก็ต้องมีการผ่านขั้นตอนการตรวจสอบคุณภาพ การเรียนก็มีการสอบประเมินผล ฯลฯ จะเห็นได้ว่าการตรวจสอบคุณภาพผลของงานนั้นๆ เป็นสิ่งที่ผู้ผลิตโดยทั่วไปให้ความสำคัญเป็นอันดับแรก ๆ ในระบบการผลิตลายวงจรบนแผ่นวงจรพิมพ์ของบริษัทและ โรงงานต่าง ๆ ก็เช่นกัน ต้องมีการตรวจเช็คความถูกต้องของผลงานที่ได้ผลิตมา ในที่นี้ก็คือลายทองแดงของวงจรไฟฟ้าบนแผ่นวงจรพิมพ์ ซึ่งจะนำไปใช้ในการประกอบกับอุปกรณ์ทางไฟฟ้าและอิเล็กทรอนิกส์หรือสิ่งอื่น เพื่อจะได้ประกอบเป็นชิ้นงานสำเร็จรูปต่อไป ในการตรวจหาจุดบกพร่องของลายวงจรพิมพ์ที่สามารถยอมรับได้นี้ก็เพื่อที่จะได้รู้ว่ามีส่วนใดของลายวงจรบ้างที่แตกต่างไปจากลายวงจรต้นแบบ เช่น เชื่อมต่อกันผิดไปจากต้นแบบ หรือลายวงจรบางแห่งขาดไป ไม่มีการเชื่อมต่อเหมือนต้นแบบ ในการตรวจเช็คตรงนี้ยังใช้มนุษย์ในการทำงานอยู่ ซึ่งสามารถเกิดความผิดพลาดได้ง่าย ผลงานจากการทำวิทยานิพนธ์นี้จะทำให้มีแนวทางที่จะช่วยในการจำแนกนี้ จะทำให้การจำแนกรวดเร็ว และมีประสิทธิภาพมากยิ่งขึ้น

1.2 ความสำคัญและที่มาของหัวข้อวิจัย

การศึกษานี้ เป็นการออกแบบวิธีการที่จะช่วยในการจำแนกความถูกต้องของลายวงจรบนแผ่นวงจรพิมพ์ (printed circuit boards : PCBs) ที่ผลิตในปริมาณที่มากพอสมควร โดยส่วนใหญ่จะเป็นการผลิตแผ่นวงจรพิมพ์เพื่อใช้ในงานอุตสาหกรรม ในการวิจัยจะรับข้อมูลของลายวงจรพิมพ์เข้ามาในแบบของข้อมูลภาพ (bitmap file) ซึ่งจะเป็นไฟล์ภาพที่มีนามสกุล BMP จากนั้นจึงนำข้อมูลภาพเหล่านั้นมาทำการปรับปรุงภาพโดยการตัดพื้นหลังของภาพออก ให้เหลือเฉพาะข้อมูลภาพที่เป็นเส้นของลายวงจรพิมพ์ แล้วจึงนำข้อมูลภาพที่เหลือไปทำการสอนให้นิวรอลเน็ตเวอร์กแบบแพร่กลับ (backpropagation neural network) รู้จัก โดยที่นิวรอลเน็ตเวอร์กจะทำการคำนวณหาค่าน้ำหนักในแต่เซลล์ของทุกๆ ชั้นซ่อน (hidden layer) โดยการเรียนรู้ด้วยตัวเอง เมื่อถึงเวลาที่จะนำไปใช้งานจริง ก็จะนำค่าของน้ำหนักที่ผ่านการเรียนรู้แล้ว ไปใช้ในการจำแนกว่าข้อมูลภาพลายวงจรที่เข้ามาใหม่แตกต่างไปจากต้นแบบที่ได้รับการเรียนรู้มาหรือไม่ ถ้าแตกต่างก็ต้องพิจารณาว่าค่าของความแตกต่างนั้นสามารถยอมรับได้หรือไม่ ผลที่ได้รับจาก

งานวิจัยนี้สามารถนำไปช่วยในการทำการตรวจสอบความถูกต้องของลายวงจรบนแผ่นวงจรพิมพ์ในระบบการผลิตแผ่นวงจรพิมพ์ได้

1.3 การตรวจเอกสาร

1.3.1 ในงานของ Henry A. Rowley (1998) ได้นำเสนอการตรวจหาภาพใบหน้าที่เหมือนกัน โดยการใช้นิวรอลเน็ตเวอร์กในการทำงานเพื่อค้นหารูปต้นแบบว่าคือรูปหน้าของใคร ในขั้นตอนการรับภาพเข้ามาจะเป็นรับภาพระดับเทาขนาด 20X20 จุดต่อภาพ นำภาพที่ได้ไปทำการปรับปรุงภาพด้วยการเพิ่มความแตกต่างระหว่างจุดขาวและดำที่มีในภาพ จากนั้นจึงนำภาพไปแบ่งเป็นขนาดต่างๆ คือ 10X10 , 5X5 และ 20X5 เพื่อใช้ในการเรียนรู้และตรวจหาของนิวรอลเน็ตเวอร์ก จุดเด่นของงานวิจัยนี้คือสามารถค้นหาภาพที่มีลักษณะของหน้ามนุษย์ทั่วไปคือมีองค์ประกอบของใบหน้า (ตา , จมูก , ปาก) ครบถ้วน แม้ว่าจะเป็นภาพวาดรูปคล้ายคน ข้อจำกัดที่ผู้วิจัยบอกไว้คือเวลาที่ใช้ในการเรียนรู้และการตรวจหารูปหน้านั้นยังใช้เวลามากอยู่ และยังมีความผิดพลาดในการตรวจหาที่สูง

1.3.2 จากงานของ Kraison Aunchaleevarapan (1999) มีการนำเสนอการตรวจดูว่ามีลายวงจรพิมพ์ส่วนใดบนแผ่นวงจรพิมพ์ที่ผิดพลาดไปในการทำการผลิต โดยการวัดค่าของสนามแม่เหล็กที่เกิดจากลายวงจรพิมพ์แบบต่างๆ ในแบบของสเปคตรัม แล้วนำไปผ่านนิวรอลเน็ตเวอร์กเพื่อตรวจหาว่ามีส่วนใดของลายวงจรที่ผลิตมาไม่สมบูรณ์ จุดเด่นของการวิจัยนี้คือการประยุกต์ใช้นิวรอลเน็ตเวอร์กเพื่อนำมาใช้ในการทำงานจริง แต่บอกเพียงว่าสามารถใช้นิวรอลเน็ตเวอร์กในการใช้งานได้จริงเท่านั้น ไม่ได้กล่าวว่าขนาดของเส้นลายวงจรพิมพ์มีขนาดเท่าใด

1.3.3 การประมวลผลภาพเชิงตัวเลข (กิตติ ไพฑูรย์วัฒนกิจ และ สาธิต อินทจักร์, 2538) ในบทความนี้ได้นำเสนอวิธีการหาขอบภาพ ซึ่งเป็นขั้นตอนหนึ่งในหลายขั้นตอนของการประมวลผลภาพในระดับต่ำที่มีความสำคัญมากขั้นตอนหนึ่ง ในการที่จะทำให้คอมพิวเตอร์สามารถเข้าใจข้อมูลภาพที่ต้องเข้าไปใช้งานนั้นนั้นได้ ในงานวิจัยนี้ผู้ทำวิจัยได้นำเสนอวิธีการหาขอบภาพด้วยวิธีต่างๆ ที่เป็นที่นิยมใช้กันทั่วไปในการประมวลผลภาพ เช่นการหาขอบภาพโดยวิธี Sobel, Marr-Hildreth, Canny, Gabor, Relaxation เป็นต้น ซึ่งวิธีการเหล่านี้ผู้วิจัยได้กล่าวถึงผลที่ได้จากการวิจัยรวมทั้งมีการเปรียบเทียบข้อดีและข้อเสียของแต่ละวิธีด้วย

1.3.4 จากการทดสอบทางทฤษฎีการปรับปรุงภาพของ กนกศักดิ์ เอี่ยมโอภาส (2535) โดยมีวัตถุประสงค์เพื่อใช้ในงานที่เกี่ยวกับการเกษตร แต่ได้มีบางส่วนของงานวิจัยที่ต้องใช้วิธีการทาง *image processing* เข้ามาช่วย จึงมีการบรรยายถึงทฤษฎีและหลักการเบื้องต้นที่เกี่ยวกับ

image processing ซึ่งมีขั้นตอนของการปรับปรุงภาพที่ดีและมีการบรรยายถึงผลการใช้งานของการปรับปรุงภาพในลักษณะต่างๆ

1.4 วัตถุประสงค์การวิจัย

1.4.1 เพื่อศึกษาและนำเทคโนโลยีทางด้านนิวรอลเน็ตเวอร์ก มาประยุกต์ใช้ในการจำแนกความถูกต้องของลายวงจรบนแผ่นวงจรพิมพ์

1.4.2 เพื่อหาจุดบกพร่องของลายวงจรและจำแนกแผ่นวงจรพิมพ์ที่ใช้งานได้กับใช้งานไม่ได้

1.5 ขอบเขตของการวิจัย

1.5.1 งานวิจัยนี้จะใช้โปรแกรมคอมพิวเตอร์ นิวรอลเน็ตเวอร์ก แบบ backpropagation ในการจำแนกความถูกต้องของลายวงจรบนแผ่นวงจรพิมพ์

1.5.2 ลายวงจรที่อยู่บนแผ่นวงจรพิมพ์สามารถจำแนกได้ครั้งละ 1 ด้าน

1.5.3 การรับภาพลายวงจรจากแผ่นวงจรพิมพ์จะใช้กล้อง CCD ที่มีความละเอียด 320x240 pixel

1.5.4 ภาพลายวงจรบนแผ่นวงจรพิมพ์จะถูกมองแบบ 2 มิติ

1.5.5 ขนาดของเส้นลายวงจรที่เล็กที่สุดที่สามารถจำแนกได้คือ 1/32 นิ้ว

1.6 ขั้นตอนและวิธีดำเนินการวิจัย

1.6.1 ออกแบบโปรแกรมในการรับภาพลายวงจร

1.6.2 เขียนโปรแกรมในการปรับปรุงภาพ

1.6.3 ออกแบบโปรแกรมนิวรอลเน็ตเวอร์ก

1.6.4 เขียนโปรแกรมในการจัดระเบียบข้อมูลภาพเพื่อให้สามารถใช้กับนิวรอลเน็ตเวอร์กได้

1.6.5 ทดลองใช้โปรแกรมกับลายวงจรที่มีอยู่บนกระดาษ และปรับปรุงให้ได้ผลที่ดี โดยปรับปรุงความสว่างของแสง ความละเอียดของภาพ ข้อจำกัดของขนาดของแผ่น pcb

1.6.6 ทดลองใช้โปรแกรมกับลายวงจรที่มีอยู่บนแผ่น pcb และปรับปรุงให้ได้ผลที่ดีพอ

1.6.7 ทดลองใช้โปรแกรมกับแผ่น pcb สีอื่น และรูปแบบอื่นๆ

1.6.8 วิเคราะห์และสรุปผลการทดลอง

1.7 ประโยชน์ที่คาดว่าจะได้รับจากการวิจัย

1.7.1 ได้รับความรู้ในการนำนิวรอลเน็ตเวอร์กมาประยุกต์ใช้ในงานจริง

1.7.2 สามารถนำระบบการทำงานของงานวิจัยนี้ไปปรับใช้ได้กับงานวิจัยอื่นที่เกี่ยวข้องกับภาพ 2 มิติ

บทที่ 2

ภาพและการปรับปรุงภาพลายวงจรพิมพ์

2.1 ความเบื้องต้น

เนื่องจากข้อจำกัดต่างๆ ของอุปกรณ์รับภาพต่างๆ ที่มีอยู่ ทำให้ภาพที่รับเข้ามาจากอุปกรณ์รับภาพทั้งหลาย เป็นภาพที่มีองค์ประกอบอื่นๆ ที่ไม่ต้องการในการใช้งาน เช่น ภาพพื้นหลังของลายวงจรพิมพ์ สิ่งรบกวนที่เกิดจากการถ่ายภาพ ฯลฯ สิ่งเหล่านี้ไม่เป็นที่ต้องการในการจำแนกความถูกต้องของลายวงจรบนแผ่นวงจรพิมพ์ ในบทนี้จะนำเสนอทฤษฎีและวิธีการปรับปรุงภาพลายวงจรพิมพ์ ซึ่งการปรับปรุงภาพลายวงจรพิมพ์ก่อนที่จะนำไปใช้ในการจำแนกความถูกต้องนั้นเป็นขั้นตอนที่สำคัญมากในการที่จะช่วยให้การจำแนกมีความถูกต้องมากที่สุด

2.2 ภาพลายวงจรพิมพ์ในรูปแบบของ BMP file

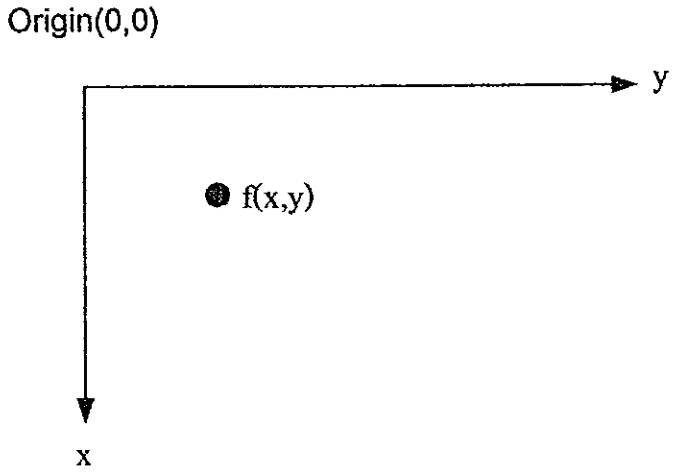
ในการรับภาพลายวงจรพิมพ์นั้นจะใช้อุปกรณ์รับภาพใดๆ ก็ได้ที่สามารถรับภาพเป็นภาพระดับเทา (gray scale) และทำการเก็บภาพที่ได้ในแบบ Bitmap File (.BMP) จากนั้นจึงสามารถนำไปทำการปรับปรุงภาพต่อไป ภาพที่รับมาเพื่อนำมาประมวลผลด้วยคอมพิวเตอร์นั้นจะถูกแทนด้วยตัวเลขที่อยู่ในรูปของเมทริกซ์ขนาด n x m ดังภาพประกอบ 2-1

$$image = \begin{bmatrix} i(1,1) & i(1,2) & \cdots & i(1,M) \\ i(2,1) & i(2,2) & \cdots & i(2,M) \\ \vdots & \vdots & & \vdots \\ i(N,1) & i(N,2) & \cdots & i(N,M) \end{bmatrix}$$

ภาพประกอบ 2-1 แสดงรูปแบบของภาพที่จะนำมาประมวลผลในแบบเมทริกซ์

ซึ่งภาพที่ได้จากอุปกรณ์รับภาพจะอยู่ในรูปแบบระนาบสองมิติคือแกน x และ y สิ่งที่แสดงอยู่ในแต่ละตำแหน่งของคู่อันดับในเมทริกซ์ภาพคือค่าความเข้มของภาพในจุดนั้นนั้น ในกรณีที่เป็นภาพสีก็สามารถแสดงได้ถึง 256 สี โดยทั่วไปแล้วจะมีค่าความเข้มของภาพระดับเทาที่เป็นไปได้อยู่ 256 ระดับ ซึ่งจะทำให้ค่าของระดับเทาที่แสดงในภาพจะอยู่ในช่วง 0 ถึง 255 โดยใช้ 8 บิต (2^8 = 256) สำหรับเก็บข้อมูลภาพในแต่ละจุด ในกรณีที่ต้องการภาพที่มีความละเอียดสูงๆ ก็จะต้องการจำนวนบิตสำหรับเก็บข้อมูลเพิ่มขึ้นไปคืออาจจะเป็น 16 หรือ 24 บิต

โดยค่าระดับความเข้มในแต่ละจุดของภาพก็จะเป็น 2^{16} และ 2^{24} ตามลำดับ โดยที่ภาพที่รับเข้ามามีจุดเริ่มต้นของคู่ลำดับอยู่ที่มุมบนด้านซ้าย ดังภาพประกอบ 2-2

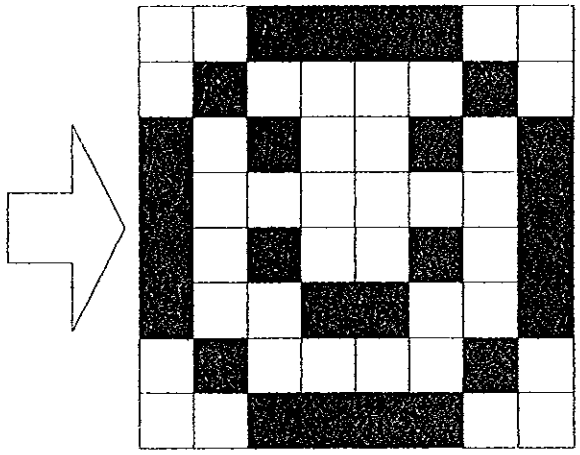


ภาพประกอบ 2-2 แสดงลักษณะการอ้างอิงตำแหน่งแต่ละจุดบนภาพ

สิ่งที่สำคัญมากในการที่ผู้ใช้คอมพิวเตอร์จะสามารถสื่อสารกับคอมพิวเตอร์ได้ง่ายขึ้นก็โดยการใช้สิ่งที่เราเรียกว่าไอคอน (icon) ไม่ว่าจะเป็นภาพสีหรือภาพระดับเทาก็ตาม ไอคอนเหล่านี้จะเป็นไฟล์ภาพในแบบ bitmap file ซึ่งสามารถแสดงได้หลายรูปแบบเช่น .RLE, .TIFF, .TGA, .PCX, .PNG, .BMP, .PCD และ .GIF ในงานวิจัยนี้จะขอกล่าวถึงเฉพาะรูปแบบของ .BMP ที่เป็นภาพระดับเทาเท่านั้น

ภาพแบบ .BMP ที่อยู่ในแบบขาวดำนั้นมีรูปแบบที่ต้องทำความเข้าใจเพื่อที่จะนำไปใช้ในการทำงานหรือประมวลผลด้วยคอมพิวเตอร์ต่อไป คือข้อมูลภาพที่ได้จะมีลักษณะเป็นภาพขาว-ดำ ที่มีความเข้มของสีดำและสีขาวไม่เท่ากันเราเรียกว่า ระดับเทา(gray level) พิกเซลที่มีค่าระดับเทาเป็น 0 คือสีดำ เมื่อเพิ่มค่าระดับเทาขึ้นเรื่อยๆ ค่าระดับเทาที่เรามองว่าเป็นจุดที่ขาวที่สุดคือค่าระดับเทาที่ 255 ซึ่งก็จะเห็นเป็นสีขาว ซึ่งจะแสดงเปรียบเทียบกันดังภาพประกอบ 2-3 รูปแบบของไฟล์ .BMP นี้ไม่สามารถทำการเก็บข้อมูลโดยมีการบีบอัดได้ เนื่องจากเป็นไฟล์ที่มีรูปแบบของข้อมูลภาพในแต่ละจุดที่อยู่ในภาพนั้นโดยมีส่วนที่เรียกว่า header ของภาพ จำนวน 54 ไบต์ อยู่ที่ส่วนแรกของไฟล์ภาพ ในส่วน header นี้จะเป็นการแสดงถึงลักษณะและรายละเอียดของภาพ เช่น เป็นภาพชนิดใด มีขนาดของภาพแบบกว้างเท่าไรและยาวเท่าไร ฯลฯ ส่วนต่างๆเหล่านี้ แสดงในตาราง 2-1

255	255	0	0	0	0	255	255
255	0	255	255	255	255	0	255
0	255	0	255	255	0	255	0
0	255	255	255	255	255	255	0
0	255	0	255	255	0	255	0
0	255	255	0	0	255	255	0
255	0	255	255	255	255	0	255
255	255	0	0	0	0	255	255



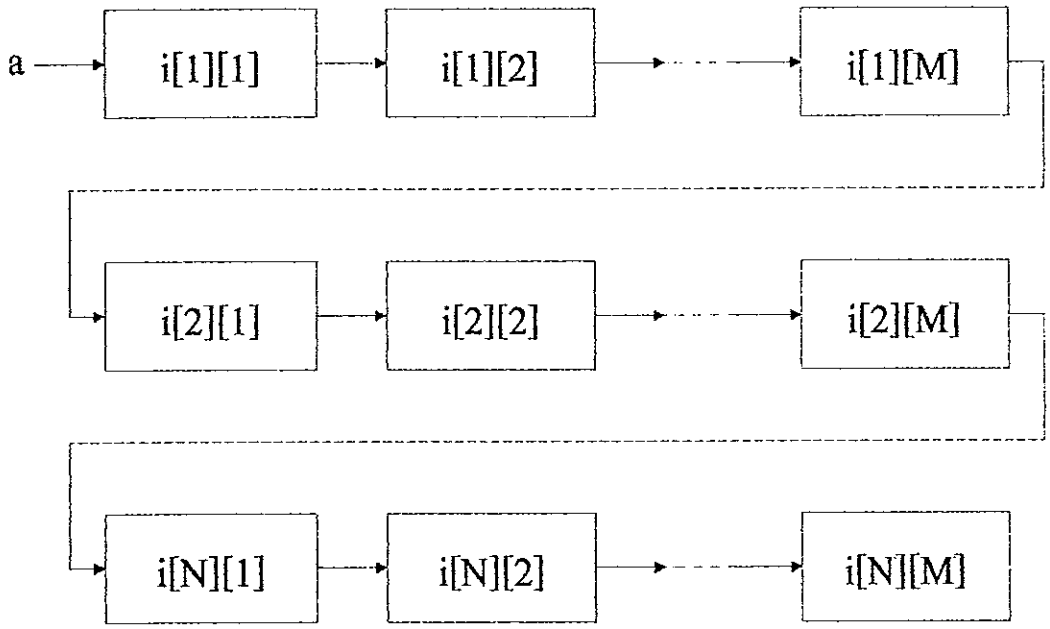
ภาพประกอบ 2-3 แสดงภาพขาวดำเมื่อจะนำข้อมูลภาพไปประมวลผล

ตาราง 2-1 แสดงรูปแบบของ header ของข้อมูลภาพแบบ BMP

Data	Description
WORD Type;	File type. Set "BMP"
DWORD Size;	Size in BYTES of the file.
DWORD Reserved;	Reserved. Set to zero.
DWORD Offset;	Offset to the data.
DWORD headerSize;	Size of rest of header. Set to 40.
DWORD Width;	Width of bitmap in pixels.
DWORD Height;	Height of bitmap in pixels.
WORD Planes;	Number of planes. Set to 1.
WORD BitsPerPixel;	Number of bits per pixel.
DWORD Compression;	Compression. Usually set to 0.
DWORD SizeImage;	Size in Bytes of the bitmap.
DWORD XpixelsPerMeter;	Horizontal pixels per meter.
DWORD YpixelsPerMeter;	Vertical pixels per meter.
DWORD ColorsUsed;	Number of colors used.
DWORD ColorsImportant;	Number of "important" colors.

(ที่มา : Rimmer , 1993 : 116)

การอ่านข้อมูลภาพที่มีอยู่ในแบบ .BMP ก็สามารถทำได้สะดวก ด้วยการอ่านข้อมูลที่มีอยู่ใน header จากนั้นก็ทำการอ่านข้อมูลภาพตามลำดับที่เรียงกันอยู่ในไฟล์ ตำแหน่งของแต่ละจุดบนภาพเราจะรู้ได้จากขนาดของภาพในแนวตั้งและแนวนอน ขนาดของแต่ละจุดของภาพแล้วจึงทำการอ่านข้อมูลภาพทีละตำแหน่งตามลำดับดังภาพประกอบ 2-4 แล้วจึงนำข้อมูลภาพที่อ่านได้นั้นเก็บไว้ในหน่วยความจำแบบเมทริกซ์ขนาด n x m ดังที่ได้กล่าวไว้ข้างต้น



ภาพประกอบ 2-4 แสดงลำดับการอ่านข้อมูลภาพจากไฟล์

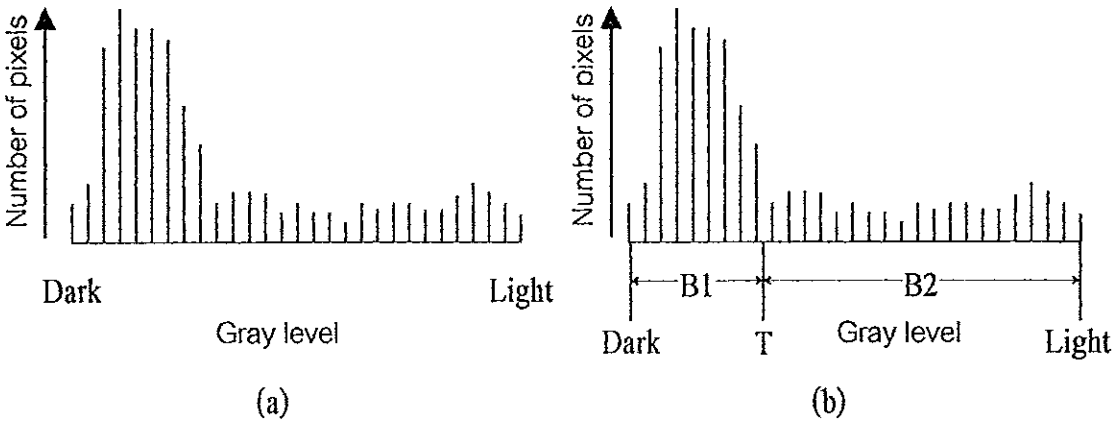
2.3 การปรับปรุงภาพ

2.3.1 การแบ่งภาพ

ในงานวิจัยนี้ คำว่า segmentation หมายถึงการแบ่งแยกส่วนที่เป็นวัตถุที่อยู่ภายในรูปภาพนั้นออกมาจากส่วนที่เป็นพื้นหลังของภาพ วิธีการแบ่งภาพหรือ image segmentation นั้นถือได้ว่าเป็นขั้นตอนที่สำคัญมากของงาน image processing เพราะจะทำให้เราได้รูปวัตถุที่เราสนใจหรือต้องการเพื่อนำไปใช้งานและสามารถทำงานในขั้นต่อไปได้ หลักการเบื้องต้นของ image segmentation คือการใช้คุณสมบัติของค่าระดับเทาในแต่ละจุดของภาพมาใช้งาน คุณสมบัติที่กล่าวถึงคือ ความไม่ต่อเนื่อง (discontinuity) และความคล้ายคลึงกัน (similarity)

วิธีการที่จะใช้ในการทำ image segmentation ในงานวิจัยนี้คือการทำ thresholding การทำงานของวิธีนี้คือการแบ่ง histogram ของภาพระดับเทานั้นเป็นส่วนๆ โดย

เลือกใช้ค่า threshold (T) ที่เหมาะสมดังแสดงในภาพประกอบ 2-5 จะเห็นว่า histogram ที่แสดงนั้นจะแบ่งแยกให้เห็นชัดว่ามีส่วนใดของภาพบ้างที่เป็นวัตถุและมีส่วนใดที่เป็นพื้นหลังของภาพ



ภาพประกอบ 2-5 (a) แสดง histogram ของภาพ
(b) แสดงการเลือกค่า threshold ของภาพ

จากภาพประกอบ 2-5 แสดง histogram thresholding ซึ่งมีจุดภายในภาพในส่วนที่เป็นพื้นหลังจำนวนมากที่เป็นส่วนมืด และส่วนน้อยที่เหลือจะเป็นส่วนสว่างของภาพ ค่า T ที่เลือกนั้นจะถูกเลือกเพื่อให้ส่วน B2 ในภาพประกอบ 2-5 มีค่าระดับเทาใกล้เคียงกับส่วนที่เป็นพื้นหลังมากที่สุด และส่วน B1 ซึ่งเป็นส่วนของวัตถุ การเปลี่ยนแปลงค่าระดับเทาในภาพโดยใช้ค่า threshold เป็นจุดแบ่งแยกนั้น ถือว่าเป็นขอบเขตของวัตถุในรูปภาพ การหาวัตถุที่มีอยู่ในภาพทั้งในแนวตั้งและแนวนอนสามารถทำได้โดยใช้รูปแบบของการคำนวณดังต่อไปนี้

$$g(x,y) = \begin{cases} G_0 & if \quad f(x,y) > T \\ G_b & if \quad f(x,y) \le T \end{cases} \qquad (2\text{-}1)$$

โดยที่ f(x,y) คือค่าระดับเทาของภาพต้นฉบับ

g(x,y) คือค่าระดับเทาของภาพหลังจากการทำ thresholding

T คือค่า threshold

G_0 คือค่าระดับเทาของวัตถุหลังจากทำ thresholding

G_b คือค่าระดับเทาของพื้นหลัง หลังจากทำ thresholding

วิธีการปรับปรุงภาพวิธีนี้ จะเป็นการนำค่าระดับเทาของทุก pixel มาเปรียบเทียบกับค่า threshold ที่ตั้งไว้ ถ้าค่าระดับเทามีค่ามากกว่าค่า threshold ค่าระดับเทาที่ pixel นั้นจะถูกเปลี่ยนค่าเป็น G_0 ถ้าน้อยกว่าหรือเท่ากับค่า threshold ให้เปลี่ยนค่าที่ pixel นั้นเป็น G_b

2.3.2 การปรับความสว่างของภาพ

การปรับความสว่างของภาพหรือ brightness correction วิธีนี้จะเป็นการเพิ่มความสว่างให้กับภาพ เพื่อให้สามารถมองเห็นระดับต่างๆ ของภาพ ซึ่งจะมีรูปแบบในการคำนวณดังนี้

$$S_i = g_i + \text{brightness} \qquad \ldots(2\text{-}2)$$

โดยที่ S_i คือค่าระดับเทาของ pixel หลังจากผ่านกระบวนการ brightness

g_i คือค่าระดับเทาของ pixel ก่อนผ่านกระบวนการ brightness

brightness คือค่าความสว่างที่เพิ่มให้กับภาพ

2.3.3 การปรับค่าความต่างกันในภาพ

การปรับค่าความต่างกันในภาพหรือ contrast correction วิธีนี้เป็นวิธีการที่จะเพิ่มค่าความสว่างและความมืดให้กับภาพ เมื่อเปรียบเทียบกับค่าเฉลี่ยของระดับเทาภายในภาพ โดยมีวิธีการคำนวณดังนี้

$$S_i = \text{contrast} * (g_i - \text{average}) + \text{average} \qquad \ldots(2\text{-}3)$$

โดยที่ S_i คือค่าระดับเทาของ pixel หลังจากผ่านกระบวนการ contrast

g_i คือค่าระดับเทาของ pixel ก่อนผ่านกระบวนการ contrast

average สามารถคำนวณได้จาก

$$\overline{g_i} = \frac{1}{N_x N_y} \sum_{X=0}^{N_x} \sum_{Y=0}^{N_y} F(X, Y) \qquad \ldots(2\text{-}4)$$

contrast คือค่าตัวเลขที่ใช้ในการเพิ่มความแตกต่างของค่าระดับเทาในภาพให้มีค่าความแตกต่างกันมากขึ้น

จะเห็นว่าวิธีการคำนวณก็คือ จะหาค่าเฉลี่ยของระดับเทาทั้งภาพ นำค่าเฉลี่ยที่ได้ไปลบกับค่าระดับเทาในทุก pixel จากนั้นนำมาคูณกับค่า contrast (ซึ่งการคูณจะเป็นการเพิ่มค่าจากการลบให้มากขึ้น จะมีผลต่อความสว่างและความมืดของ pixel) แล้วจึงนำมารวมกับค่าเฉลี่ย

บทที่ 3

ทฤษฎีนิวรอลเน็ตเวอร์กแบบแพร่กลับ

3.1 ความเบื้องต้น

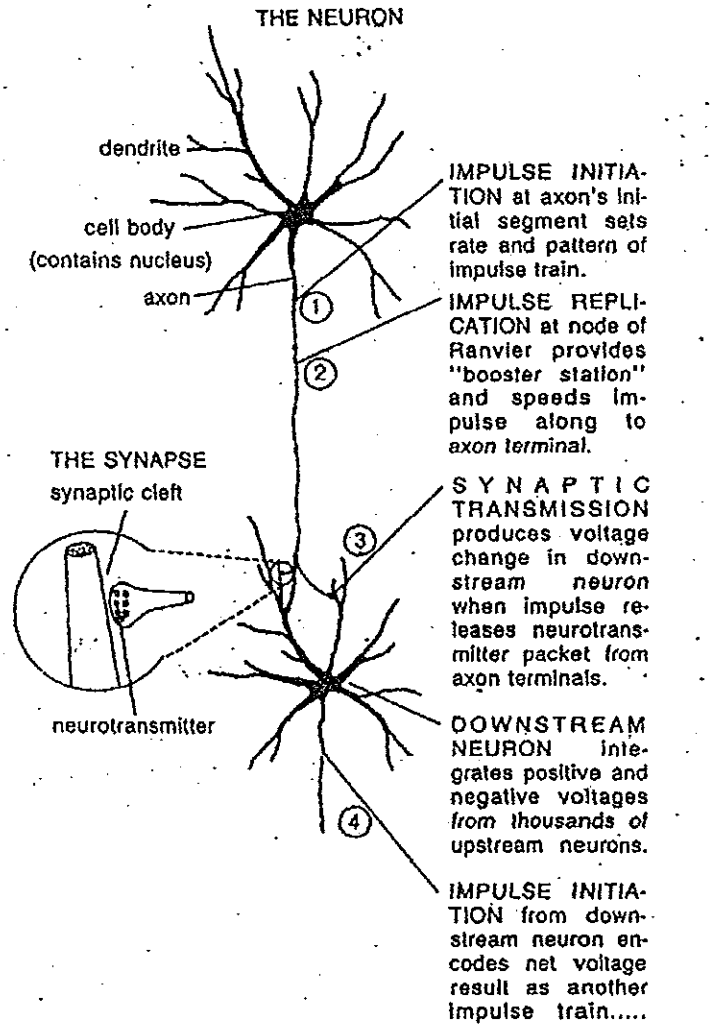
ในปัจจุบันถ้ามีความต้องการที่จะใช้งานคอมพิวเตอร์ที่มีความสามารถในการคำนวณที่รวดเร็วมาก ก็จะต้องนึกถึง super computer, คอมพิวเตอร์ที่มีความเร็วสูงๆ หรือไม่ก็เป็นเครื่องมือที่มีชิพประมวลผลที่มีความสามารถทำงานในด้านการคำนวณโดยเฉพาะ แต่ยังมีงานบางอย่างที่ต้องการความยืดหยุ่นในการทำงานสูง ระบบนี้ยังสามารถที่จะปรับตัวให้สามารถรับรู้สภาพที่เปลี่ยนแปลงได้ สิ่งนี้เป็นประโยชน์อย่างมากในการแก้ปัญหาอีกหลายประเภทที่คอมพิวเตอร์ไม่เหมาะสมจะนำมาใช้งาน วิธีการนี้จะทำหน้าที่คล้ายกับสมองของมนุษย์ในรูปแบบของอาร์ติฟีเชียลนิวรอลเน็ตเวอร์ก (artificial neural network หรือ ANN) อันเป็นการลอกเลียนการทำงานของเซลล์ประสาท (nerve cells) ซึ่งมีชื่อเรียกอีกอย่างหนึ่งว่านิวรอน (neuron) จึงมีความจำเป็นที่จะต้องกล่าวถึงพื้นฐานอย่างสั้นๆ ว่านิวรอนนั้นทำงานอย่างไรในรูปแบบของสรีรวิทยาประสาท (neurophysiology) เพื่อเป็นความรู้เบื้องต้นที่สำคัญก่อนที่จะมีการนำไปประยุกต์ใช้งานต่อไป

3.2 นิวรอลเน็ตเวอร์ก

เรื่องราวของนิวรอลเน็ตเวอร์กมีพื้นฐานมาจากนิวรอนที่เป็นของจริงในสมองมนุษย์ ดังนั้นจึงควรจะทราบว่าในการทำงานจริงของนิวรอนหรือเซลล์ประสาทภายในสมองนั้นทำงานกันอย่างไรเสียก่อน

นิวรอนหรือเซลล์ประสาทในสมองของมนุษย์นั้นจะมีหน้าที่หลักสองประการคือทำการคำนวณ และทำการส่งผลที่ได้จากการคำนวณไปยังอีกปลายหนึ่งของเซลล์ประสาทอย่างรวดเร็ว เพื่อให้สามารถส่งผลดังกล่าวไปยังเซลล์อื่นๆ ได้อย่างทันท่วงที ซึ่งทั้งหมดนี้จะทำงานโดยอาศัยหลักการทางไฟฟ้า รูปร่างลักษณะของนิวรอนจะแสดงในภาพประกอบ 3-1 ซึ่งจะมีลักษณะคล้ายกับต้นไม้ที่ปราศจากใบ แต่มีกิ่งและรากที่เชื่อมโยงด้วยลำต้น ส่วนของลำต้นนี้เรียกว่าแอ็กซอน (axon) ในแอ็กซอนของเซลล์ประสาทจะมีความยาวเพียงหนึ่งมิลลิเมตรเท่านั้น แต่ก็มีบางนิวรอนที่มีแอ็กซอนที่ยาวมากๆ ตั้งแต่ท้ายทอยจนถึงส้นเท้าก็มี การส่งความรู้สึกต่างๆ ก็จะอยู่ในรูปของสัญญาณไฟฟ้า เมื่อร่างกายสัมผัสถูกสิ่งใดก็จะเกิดสัญญาณไฟฟ้าในรูปแบบของอิมพัลส์

(impulse) ที่มีแรงเคลื่อนไฟฟ้าเพียง 1/10 โวลต์ มีช่วงกว้างของพัลส์เท่ากับ 1/1000 วินาที โดยจะวิ่งไปในเอ็กซอนด้วยความเร็ว 500 กิโลเมตรต่อชั่วโมง

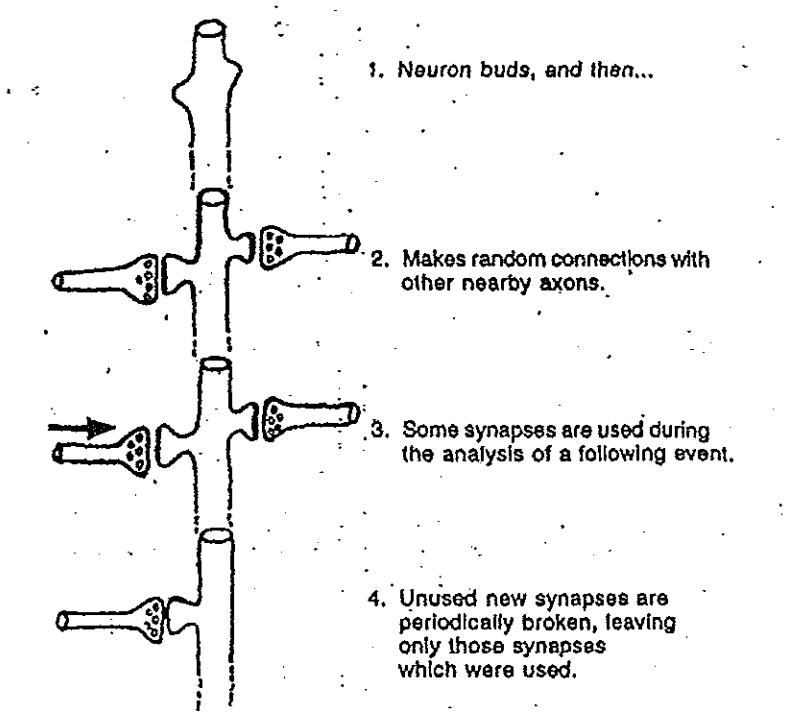


ภาพประกอบ 3-1 แสดงรายละเอียดของเซลล์ประสาท
พร้อมด้วยหน้าที่ของส่วนต่างๆ ในเซลล์ประสาท
(ที่มา : Chester, 1993 : 37)

การทำงานของการส่งสัญญาณไฟฟ้าของเซลล์ประสาทจากภาพประกอบ 3-1 จะเริ่มที่ส่วนแรกของซกเมนต์เซต ซึ่งจะทำหน้าที่กำหนดอัตราความเร็วและรูปลักษณะของกระแสอิมพัลส์ดังแสดงไว้ด้วยหมายเลข 1 จากนั้นส่วนต่างๆ จะเป็นเสมือนสถานีทวนสัญญาณพร้อม

กับเร่งความเร็วของอิมพัลส์ให้ไปสู่จุดหมายปลายทางที่เรียกว่าแอ็กซอนเทอร์มินอล (หมายเลข 2) ส่วนต่อมาคือ synaptic transmission ดังแสดงในหมายเลข 3 เป็นส่วนที่ทำการเปลี่ยนแปลงแรงดันที่ตอนปลายของนิวรอน เมื่ออิมพัลส์ปล่อยนิวโรทรานสมิตเตอร์แพคเก็ต (neurotransmitter packet) ออกจากแอ็กซอนเทอร์มินอล ในขั้นตอนสุดท้าย จะเป็นการเริ่มสัญญาณใหม่ตรงส่วนปลายของนิวรอนในรูปแรงดันสุทธิ (net voltage result) เพื่อที่จะเป็นจุดเริ่มต้นของนิวรอนถัดไป ในส่วนของการส่งสัญญาณอิมพัลส์ภายในร่างการนั้นจะอาศัยการส่งสัญญาณต่อกันเป็นทอดๆ ของเซลล์ต่อไปเรื่อยจากเซลล์หนึ่งไปยังอีกเซลล์หนึ่งโดยมีประสาทที่เชื่อมโยงตัวทรานสดิวเซอร์ (transducer) ในผิวหนังเข้ากับสมอง อิมพัลส์จะส่งต่อกันได้จะอาศัยสารเคมีที่มีประจุไฟฟ้าเป็นตัวกระตุ้น ในส่วนที่เป็นรอยต่อระหว่างเซลล์ประสาทสองเซลล์เรียกว่าซินแนปส์ (synapse) ส่วนสิ่งที่ก่อให้เกิดอาการเช่นนี้คือ การเปลี่ยนแปลงของแรงดันในขณะใดขณะหนึ่งที่ส่งผลทางเคมีอีกที ส่วนนิวโรทรานสมิตเตอร์จะเป็นสารเคมีอย่างหนึ่งที่จะเป็นตัวก่อให้เกิดการเปลี่ยนแปลงแรงดันในตอนปลายของเซลล์ประสาท ด้วยเหตุนี้การทำงานของยาระงับปวดหรือยานอนหลับจะอาศัยหลักของการเข้ามาแทรกแซงกระบวนการทางเคมีที่มีหน้าที่ในการเชื่อมโยงเซลล์ประสาทสองเซลล์ตรงจุดที่เรียกว่าซินแนปส์ ด้วยการลดหรือเพิ่มความเข้มในการเชื่อมโยง (strength of connection) ดังกล่าว

การเรียนรู้และความทรงจำเป็นคุณสมบัติที่สำคัญที่สุดในการนำมาใช้งาน การเรียนรู้ของมนุษย์ไม่ได้เกิดจากการเปลี่ยนแปลงทางเคมีชีวะภาพเพียงอย่างเดียว แต่ต้องมีการเปลี่ยนแปลงทางด้านกายวิภาค (anatomical atterations) ควบคู่ไปด้วย ซึ่งการเปลี่ยนแปลงทางกายวิภาคนี้จะไม่เกิดกับนิวรอนหรือซินแนปส์เพียงตัวเดียวหรือจุดเดียว คือจะเกิดกับหลายๆนิวรอนพร้อมๆ กัน เมื่อถึงเวลาหนึ่งนิวรอนก็จะหมดหน้าที่ไป แต่การลบเลือนของความทรงจำจะไม่เหมือนกับการสลายไปของนิวรอน นิวรอนตัวหนึ่งจะร่วมงานกับนิวรอนอีกหลายตัวเพื่อก่อให้เกิดรูปแบบของความทรงจำสำหรับเรื่องใดเรื่องหนึ่งขึ้นในสมอง หมายความว่าความจำในเรื่องหนึ่งๆ มักเกิดจากการเปลี่ยนแปลงของนิวรอนหลายๆ ตัว และจะเกิดในส่วนใดส่วนหนึ่งของสมอง โดยแต่ละส่วนของสมองจะแบ่งหน้าที่และความถนัดในเรื่องที่ต่างกันออกไป



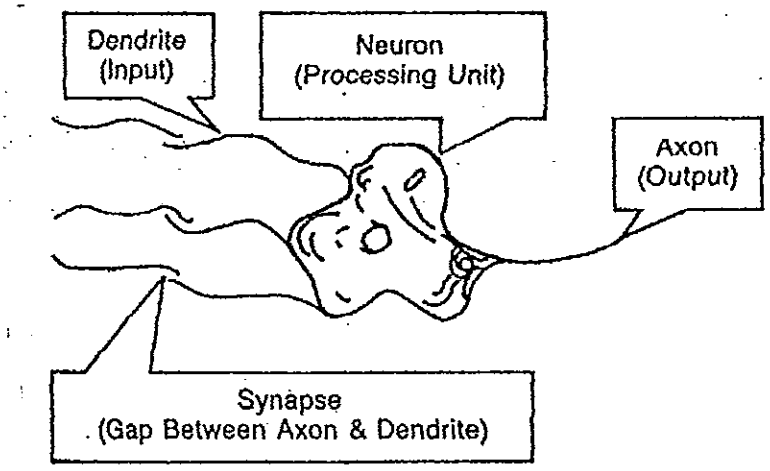
ภาพประกอบ 3-2 แสดงรูปแบบของความทรงจำระยะยาวในรูปของการจดจำแบบภาพถ่าย
(ที่มา : Chester, 1993 : 40)

จากภาพประกอบ 3-2 จะเป็นรูปแบบของความทรงจำระยะยาวในการจดจำภาพถ่าย ซึ่งต้องอาศัยตัวนิวรอนเป็นกลไกสำคัญ 4 ขั้นตอนด้วยกันคือ

1. นิวรอนแตกหน่อออกมา
2. หน่อที่แตกออกมาจะต่อกับแอ็กซอนที่อยู่ใกล้ โดยไม่มีกฎเกณฑ์ที่แน่นอน (random connections)
3. ซินแนปส์บางตัวจะวิเคราะห์เหตุการณ์ในขณะใดขณะหนึ่ง
4. แสดงการหลุดร่วงของซินแนปส์ที่ไม่ได้ใช้งาน ซึ่งการหลุดร่วงนี้จะเกิดขึ้นเป็นครั้งคราวไป คงเหลือแต่ซินแนปส์ที่ยังคงใช้งานเท่านั้น

ทั้งหมดนี้เป็นการมองถึงระบบของนิวรอนในทางสรีรวิทยาประสาทอย่างคร่าวๆ ต่อมามีผู้ที่พยายามที่จะลอกเลียนพฤติกรรมของนิวรอนทั้งในรูปของฮาร์ดแวร์และซอร์ฟแวร์ในรูปของคอมพิวเตอร์ เพราะพฤติกรรมทางสมองแบบนี้สามารถทำให้สิ่งที่คอมพิวเตอร์ทำได้ยากหรือทำไม่ได้เลยให้เป็นไปได้ โดยการมองนิวรอนในแบบที่แสดงดังภาพประกอบ 3- 3 จะเป็นนิวรอนตัว

เดียวที่มีส่วนเด็นไดรต์ (dendrites) เป็นส่วนรับข้อมูลแบบหลายช่องทาง และมีแอ็กซอนเป็นส่วนที่นำข้อมูลออกเพียงทางเดียว ในทางปฏิบัตินั้นนิวรอนตัวนี้จะทำงานเชื่อมโยงกับนิวรอนตัวอื่นๆ อีกหลายตัวโดยใช้พัลส์ทางไฟฟ้าเพื่อทำให้เอาต์พุตมีอิทธิพลต่อไปยังนิวรอนตัวอื่นๆ การมองนิวรอนในทัศนะของคอมพิวเตอร์นั้นจะประกอบไปด้วย อินพุต โปรเซสซิ่งยูนิต และเอาต์พุต พร้อมทั้งช่องซินแนปส์ที่เป็นช่องว่างระหว่างแอ็กซอนกับเด็นไดรต์ นอกจากนี้ยังสามารถฝึกนิวรอนแต่ละตัวให้เพิ่มหรือลดหรือเป็นทางผ่านของอินพุตต่างๆ รวมทั้งปิดกั้นไม่ให้อินพุผ่านไปยังเซลล์หรือนิวรอนอื่นๆ ได้ ส่วนกิจกรรมของนิวรอนนั้นจะขึ้นกับความเป็นมาในอดีตว่าได้รับการฝึกฝนมาเช่นไร



ภาพประกอบ 3-3 แสดงรูปแบบการมองนิวรอนของคอมพิวเตอร์

(ที่มา : Tager, 1994 : 22)

สรุปได้ว่านิวรอนเหล่านี้คือรากฐานระบบประสาทของมนุษย์รวมทั้งการทำหน้าที่ในด้านการคิดคำนึงต่างๆด้วย จำนวนนิวรอนที่มีอยู่ในสมองของมนุษย์นั้นมีมากมายเหลือเกิน แต่ถ้าจะดูเฉพาะบริเวณที่เรียกว่าคอร์เทกซ์นั้นจะมีนิวรอนประมาณ 10^{10} ตัว โดยนิวรอนแต่ละตัวจะเชื่อมโยงกับนิวรอนตัวอื่นอีกนับพันตัว เชื่อกันว่าความรู้ทั้งหมดของมนุษย์จะเก็บไว้ในจุดเชื่อมโยง (connections) ต่างๆ และขึ้นอยู่กับความเข้มของการเชื่อมโยงด้วย สิ่งเหล่านี้ทำให้มนุษย์สามารถเรียน คิด จดจำ หรือระลึกถึงสิ่งที่จำไว้ เพื่อที่จะสามารถนำไปใช้ประโยชน์ได้อย่างฉับพลัน จึงเป็นความมหัศจรรย์ที่คอมพิวเตอร์ไม่สามารถทำได้

นิวรอลเน็ตเวอร์กในทัศนะของคอมพิวเตอร์นั้นจะประกอบไปด้วย หน่วยโปรเซสซิ่งที่เชื่อมโยงกันหลายตัว ทำงานในลักษณะขนานกับไปคล้ายกับนิวรอนในสมองของมนุษย์ เพื่อ

แปลงข้อมูลจากรูปแบบหนึ่งไปสู่อีกรูปแบบหนึ่ง การใช้นิวรอลเน็ตเวอร์กนี้จะเป็นไปรูปแบบของการสอนให้คอมพิวเตอร์เรียนรู้แทนที่จะเป็นการป้อนโปรแกรมให้กับคอมพิวเตอร์ จุดมุ่งหมายของการสอนนิวรอลเน็ตเวอร์กหรือการป้อนข้อมูลที่ต้องการให้คอมพิวเตอร์เรียนรู้คือการทำให้ระบบคอมพิวเตอร์นี้สามารถแสดงคำตอบในรูปแบบที่ต้องการได้ ซึ่งก็จะเป็นการป้อนข้อมูลที่ถือว่ารู้อยู่แล้วเข้าไปให้นิวรอลเน็ตเวอร์ก พร้อมด้วยค่าเอาต์พุตที่ต้องการให้นิวรอลเน็ตเวอร์กนั้นแสดงออกมา จากนั้นนิวรอลเน็ตเวอร์กจะทำการคำนวณและปรับค่าตัวเลขน้ำหนักเองโดยใช้กฎเกณฑ์ต่างๆ เข้าช่วย จนกระทั่งเอาต์พุตที่ได้ออกมานั้นถูกต้องแม่นยำอยู่ในเกณฑ์ที่น่าพอใจ ส่วนตัวเลขที่เป็นกลไกที่ทำให้นิวรอลเน็ตเวอร์กสามารถเรียนรู้และจดจำได้นั้นจะอยู่ในรูปที่เรียกว่า เมตริซ์น้ำหนักของรอยต่อระหว่างเซลล์ประสาท ลักษณะการทำงานแบบนี้จะทำให้ผู้ใช้นิวรอลเน็ตเวอร์กสามารถที่จะป้อนข้อมูลใหม่ๆ เข้าไปแล้วปล่อยให้เป็นหน้าที่ของนิวรอลเน็ตเวอร์กชนิดนั้นๆ ที่จะหาทางที่จะจัดการปรับตัวเพื่อที่จะหาคำตอบนั้นเอาเอง ความก้าวหน้าของนิวรอลเน็ตเวอร์กในปัจจุบันเป็นไปอย่างรวดเร็ว ทำให้สามารถแก้ปัญหาให้กับงานบางประเภทที่ไม่สามารถทำได้ในอดีตเช่น ความสามารถในการรับข้อมูลที่ไม่สมบูรณ์แล้วสามารถทำงานได้จนสัมฤทธิ์ผลและให้คำตอบในแบบที่ผู้ใช้งานพึงพอใจ คุณสมบัติที่สำคัญของนิวรอลเน็ตเวอร์กมี 3 ประการคือ การทำงานในหน่วยโปรเซสซิ่งแบบขนาน ทำให้เกิดความรวดเร็วในการคำนวณและสามารถฝึกฝนเป็นโครงข่ายแบบนี้ได้, ความสมบุกสมบันในการทำงานในแง่ที่ว่าถ้ามีส่วนใดส่วนหนึ่งเสียไปก็ยังมีส่วนที่เหลือทำหน้าที่คำนวณต่อไปได้ (fault tolerant) และความสามารถในการจัดการข้อมูลที่ละมากๆ ได้ แต่จะต่างกับการทำงานของคอมพิวเตอร์คือ คอมพิวเตอร์สามารถให้คำตอบจากการคำนวณที่มีความละเอียดแม่นยำสูง ดังเปรียบเทียบไว้ในตาราง 3-1

รูปแบบของนิวรอลเน็ตเวอร์กส่วนใหญ่จะมีองค์ประกอบที่สำคัญเช่น รูปแบบการเรียนรู้ภายในส่วนโปรเซสซิ่ง มีการบวก การคูณ การลบตัวเลขบรรจุอยู่ หน่วยโปรเซสซิ่งในนิวรอลเน็ตเวอร์กจะทำงานสัมพันธ์กันโดยขึ้นกับการเชื่อมโยงกันหมดทุกส่วนหรือเชื่อมโยงเพียงบางส่วนก็ได้ การสร้างนิวรอลเน็ตเวอร์กต้องเริ่มด้วยการพิจารณาจาก เรื่องของการเชื่อมต่อกันภายใน (interconnection) และ สถาปัตยกรรมที่สนับสนุนการสร้างนิวรอลเน็ตเวอร์กให้เป็นไปได้

ตาราง 3-1 แสดงข้อแตกต่างของการคิดแบบนิวรอลเน็ตเวอร์กกับการคิดของคอมพิวเตอร์

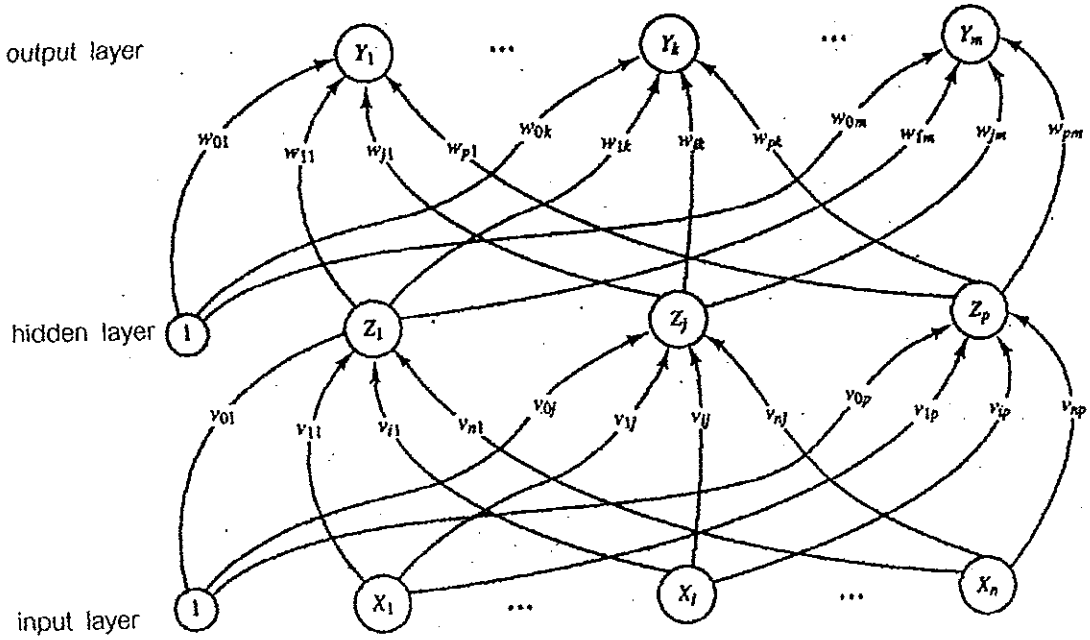
Computer	Neural Networks
Process digital data in binary form.	Process analog signals that flucttuate continuously.
Make yes/no decisions based on mathematical or logical functions.	Make weighted decisions based on fuzzy, incomplete and contradictory data.
Rigidly structured sequence of operations with predictable results.	Independently formulated methods of data processing.
Definitive answers, given enough time.	Approximate answers to highly complex problems.
Sort large data bases for exact matches.	Sort large data bases for close matches.
Specific data storage.	Associative data storage.

(ที่มา : Nelson, 1991 : 68)

3.3 นิวรอลเน็ตเวอร์กแบบแพร่กลับ (Backpropagation Neural Network) [Fausett , 1994]

3.3.1 ที่มาของนิวรอลเน็ตเวอร์กแบบแพร่กลับ

ในปี ค.ศ.1970 ได้เริ่มมีการนำนิวรอลเน็ตเวอร์กมาใช้กันแต่เป็นแบบที่มีชั้นซ่อนเพียงชั้นเดียว จากนั้นได้มีผู้วิจัยได้ทำการเรียนรู้แบบเป็นวงรอบขึ้นมา โดยในช่วงแรกมีชื่อที่ใช้เรียกรูปแบบนี้ว่า backpropagation of errors หรือ generalized delta rule ซึ่งวิธีนี้จะเป็นการนำค่าความผิดพลาดที่เอาต์พุตกลับมาใช้ในการคำนวณภายในระบบอีกครั้ง โดยทั่วไปธรรมชาติของการเรียนรู้ในนิวรอลเน็ตเวอร์กแบบแพร่กลับนี้จะประกอบไปด้วยสองส่วนหลักคือ ส่วนของการเรียนรู้ และส่วนของการทดลองนำไปใช้งานจริง



ภาพประกอบ 3-4 แสดงนิวรอลเน็ตเวอร์กแบบแพร่กลับ 1 ชั้นซ่อน

3.3.2 ฟังก์ชันโอนย้าย (Transfer Function)

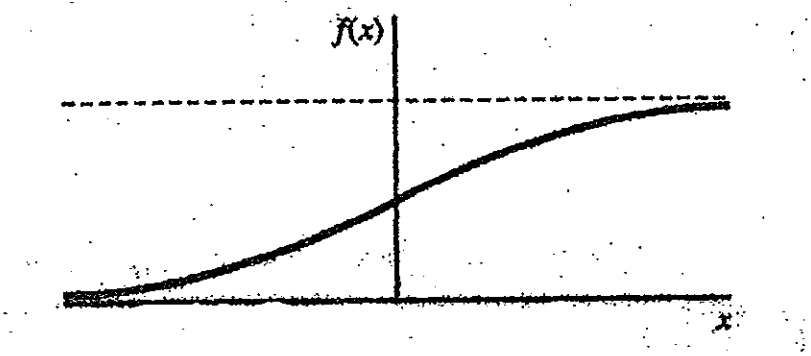
ฟังก์ชันโอนย้ายเป็นขั้นตอนหนึ่งที่ต้องใช้ในการผ่านค่า เพื่อที่จะปรับค่าให้อยู่ในช่วงใดช่วงหนึ่งก่อนที่จะนำไปใช้งานในขั้นต่อไป ฟังก์ชันโอนย้ายที่ใช้สำหรับนิวรอลเน็ตเวอร์กแบบแพร่กลับนี้ มีอยู่ 2 ประเภทคือ binary sigmoid function และ bipolar sigmoid function

3.3.2.1 Binary Sigmoid Transfer Function เป็นหนึ่งในหลายๆ ชนิดของฟังก์ชันโอนย้ายที่ใช้งานกันอยู่ในนิวรอลเน็ตเวอร์ก ซึ่งผลที่ได้จากการผ่านค่าเข้าไปในฟังก์ชันโอนย้ายนั้นจะอยู่ในช่วง 0 ถึง 1 โดยมีขั้นตอนการทำงานภายในฟังก์ชันโอนย้ายดังนี้

$$f_1(x) = \frac{1}{1 + \exp(-x)} \qquad \ldots(3\text{-}1)$$

และ หลังจากที่ทำอนุพันธ์สมการ (3-1) แล้วจะได้

$$f_1'(x) = f_1(x)[1 - f_1(x)] \qquad \ldots(3\text{-}2)$$



ภาพประกอบ 3-5 แสดงการตอบสนองของ binary sigmoid transfer function

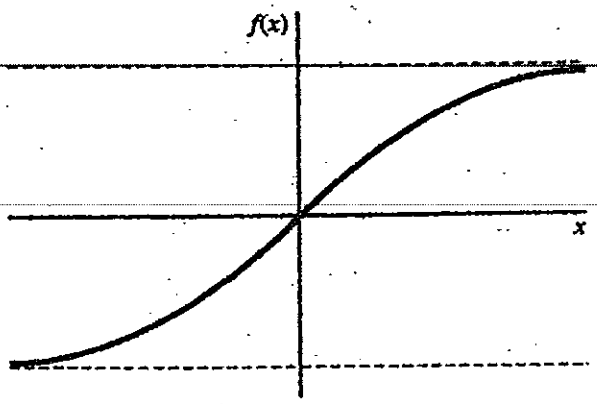
(ที่มา : Fausett, 1994 : 293)

3.3.2.2 Bipolar Sigmoid Transfer Function เป็นฟังก์ชันโอนย้ายอีกชนิดหนึ่งที่เป็นที่นิยมใช้กันมากในนิวรอลเน็ตเวอร์กแบบแพร่กลับ ซึ่งผลที่ได้จากการผ่านค่าเข้าไปในฟังก์ชันโอนย้ายนั้นจะอยู่ในช่วง -1 ถึง 1 โดยมีขั้นตอนการทำงานภายในฟังก์ชันโอนย้ายดังนี้

$$f_2(x) = \frac{2}{1 + \exp(-x)} - 1 \qquad \ldots(3\text{-}3)$$

และ หลังจากที่ทำอนุพันธ์สมการ (3-3) แล้วจะได้

$$f_2'(x) = \frac{1}{2}[1 + f_2(x)][1 - f_2(x)] \qquad \ldots(3\text{-}4)$$



ภาพประกอบ 3-6 แสดงการตอบสนองของ bipolar sigmoid transfer function

(ที่มา : Fausett, 1994 : 294)

3.3.3 ดัชนีชี้สมรรถนะ

ในนิวรอลเน็ตเวอร์กแบบแพร่กลับนี้จะใช้อัลกอริทึมการประมาณค่า และดัชนีชี้สมรรถนะคือค่าเฉลี่ยกำลังสองของความผิดพลาด ในการแพร่กลับนั้น การหาค่าอนุพันธ์ของความผิดพลาดเมื่อเทียบกับน้ำหนักสำหรับนิวรอลเน็ตเวอร์กแบบชั้นเดียวจะไม่ซับซ้อนมากนัก แต่ในนิวรอลเน็ตเวอร์กที่มากกว่า 1 ชั้นขึ้นไป อนุพันธ์ระหว่างน้ำหนักกับค่าความผิดพลาดของโครงข่ายจะมีความซับซ้อนมากขึ้น การคำนวณจึงต้องมีการใช้กฎลูกโซ่ (chain rule) เข้าช่วย อัลกอริทึมการแพร่กลับสำหรับโครงข่ายหลายชั้น จะได้ดัชนีสมรรถนะเช่นเดียวกับอัลกอริทึม LMS (least mean square algorithm) คือค่าเฉลี่ยของความผิดพลาดกำลังสอง ซึ่งจะมีวิธีการคำนวณดังนี้

เมื่อให้กลุ่มข้อมูลในการเรียนรู้และเป้าหมายคือ $\{p_1,t_1\}$, $\{p_2,t_2\}$, ...,$\{p_q,t_q\}$ เมื่อป้อนอินพุตให้กับโครงข่าย เอาต์พุตของโครงข่ายจะถูกเปรียบเทียบกับค่าเป้าหมาย อัลกอริทึมจะทำการปรับค่าต่างๆ ของโครงข่าย เพื่อให้ค่าเฉลี่ยของความผิดพลาดกำลังสองมีค่าน้อยที่สุด โดยที่ค่าเฉลี่ยของความผิดพลาดกำลังสองคือ

$$F(x) = E\left[e^2\right] = E\left[(t-a)^2\right] \qquad \ldots(3\text{-}5)$$

ถ้าโครงข่ายมีหลายเอาต์พุต สมการทั่วไปของโครงข่ายจาก (3-5) จะได้

$$F(x) = E\left[e^T e\right] = E\left[(t-a)^T (t-a)\right] \qquad \ldots(3\text{-}6)$$

เหมือนกับ LMS อัลกอริทึม จะสามารถประมาณค่าเฉลี่ยของความผิดพลาดกำลังสองได้เท่ากับ

$$F(x) = (t(k)-a(k))^T (t(k)-a(k)) = e^T(k)e(k) \qquad \ldots(3\text{-}7)$$

เมื่อทราบดังนี้ การประมาณค่าเฉลี่ยของความผิดพลาดกำลังสองของน้ำหนักคือ

$$w^m_{i,j}(k+1) = w^m_{i,j}(k) - \alpha \frac{\partial F}{\partial w^m_{i,j}} \qquad \ldots(3\text{-}8)$$

ในส่วนของค่าไบอัสคือ

$$b_i^m(k+1) = b_i^m(k) - \alpha \frac{\partial F}{\partial b_i^m} \qquad \text{...(3-9)}$$

กฎลูกโซ่ (chain rule) เป็นกฎที่ใช้สำหรับการคำนวณอนุพันธ์สำหรับโครงข่ายหลายชั้น เพราะค่าความผิดพลาดเป็นฟังก์ชันโดยอ้อมของน้ำหนักในชั้นซ่อน การคำนวณค่าอนุพันธ์สามารถทำได้โดยใช้กฎลูกโซ่นี้ ถ้าฟังก์ชัน f มีความชัดเจนว่าขึ้นอยู่กับตัวแปร n แล้ว จะต้องมีการหาค่าอนุพันธ์ของ f เมื่อเทียบกับตัวแปรตัวที่สามคือ w จะได้ว่า

$$\frac{\partial f(n(w))}{\partial w} = \frac{\partial f(n)}{\partial n} \times \frac{\partial n(w)}{\partial w} \qquad \text{...(3-10)}$$

เราสามารถใช้หลักการนี้กับสมการที่ (3-8) และ (3-9) จะได้เป็นสมการที่ (3-11) และ (3-12) ตามลำดับดังนี้

$$\frac{\partial F}{\partial w_{i,j}^m} = \frac{\partial F}{\partial n_i^m} \times \frac{\partial n_i^m}{\partial w_{i,j}^m} \qquad \text{...(3-11)}$$

$$\frac{\partial F}{\partial b_i^m} = \frac{\partial F}{\partial n_i^m} \times \frac{\partial n_i^m}{\partial b_i^m} \qquad \text{...(3-12)}$$

สำหรับเทอมที่สองของสมการที่ (3-11) และ (3-12) สามารถหาได้ง่ายเพราะชัดเจนว่าอินพุตของชั้นที่ m จะขึ้นอยู่กับค่าน้ำหนักและค่าไบอัสของชั้นนั้น

$$n_i^m = \sum_{j=1}^{s^{m-1}} w_{i,j}^m a_j^{m-1} + b_i^m \qquad \text{...(3-13)}$$

ดังนั้น

$$\frac{\partial n_i^m}{\partial w_{i,j}^m} = a_j^{m-1}, \frac{\partial n_i^m}{\partial b_i^m} = 1 \qquad \text{...(3-14)}$$

ถ้าจะให้ว่า S^m_i คือความไวของ F ที่เปลี่ยนเทอมที่ i ของอินพุตชั้นที่ m จะได้

$$s_i^m = \frac{\partial F}{\partial n_i^m} \qquad \text{...(3-15)}$$

และสมการที่ (3-11) และ (3-12) จะเขียนได้เป็น

$$\frac{\partial F}{\partial w^{m}_{i,j}} = s^{m}_{i} a^{m-1}_{j} \qquad \text{...(3-16)}$$

$$\frac{\partial F}{\partial b^{m}_{i}} = s^{m}_{i} \qquad \text{...(3-17)}$$

เราจะได้สมการของ steepest descent ดังนี้

$$w^{m}_{i,j}(k+1) = w^{m}_{i,j}(k) - \alpha s^{m}_{i} a^{m-1}_{j} \qquad \text{...(3-18)}$$

$$b^{m}_{i}(k+1) = b^{m}_{i}(k) - \alpha s^{m}_{i} \qquad \text{...(3-19)}$$

เมื่อ S^m หมายถึงความไวของชั้นที่ m นั้นจะสามารถคำนวณความไวได้จากชั้นที่ $m+1$ ได้ดังนี้

$$\frac{\partial n^{m+1}_{i}}{\partial n^{m}_{j}} = \frac{\partial\left(\sum_{l=1}^{s^{m}} w^{m+1}_{i,l} a^{m}_{l} + b^{m+1}_{i}\right)}{\partial n^{m}_{j}} = w^{m+1}_{i,j} \frac{\partial a^{m}_{j}}{\partial n^{m}_{j}}$$

$$= w^{m+1}_{i,j} \frac{\partial f^{m}(n^{m}_{j})}{\partial n^{m}_{j}} = w^{m+1}_{i,j} \dot{f}^{m}(n^{m}_{j}) \qquad \text{...(3-20)}$$

เมื่อ

$$\dot{f}^{m}(n^{m}_{j}) = \frac{\partial f^{m} n^{m}_{j}}{\partial n^{m}_{j}} \qquad \text{...(3-21)}$$

ดังนั้นจะสามารถเขียนสมการของความไวโดยใช้กฎลูกโซ่ได้ดังนี้

$$s^{m} = \frac{\partial F}{\partial n^{m}} = \left(\frac{\partial n^{m+1}}{\partial n^{m}}\right)^{T} \frac{\partial F}{\partial n^{m+1}}$$

$$= \dot{f}^{m}(n^{m})(W^{m+1})^{T} \frac{\partial F}{\partial n^{m+1}}$$

$$= \dot{F}^{m}(n^{m})(W^{m+1})^{T} s^{m+1} \qquad \ldots(3\text{-}22)$$

จะพบว่าความไวของนิวรอลเน็ตเวอร์กแบบแพร่กลับจะเริ่มแพร่จากชั้นสุดท้ายแล้วจึงย้อนไปเรื่อยๆ จนถึงชั้นแรกสุดคือ

$$s^{m} \rightarrow s^{m-1} \rightarrow \cdots \rightarrow s^{2} \rightarrow s^{1} \qquad \ldots(3\text{-}23)$$

เพื่อให้การแพร่กลับสมบูรณ์จึงจะต้องหาค่าความไวของการแพร่กลับในชั้นสุดท้าย ซึ่งหาได้จาก

$$s_{i}^{M} = \frac{\partial F}{\partial n_{i}^{M}} = \frac{\partial (t-a)^{T}(t-a)}{\partial n_{i}^{M}}$$

$$= \frac{\partial \sum_{j=1}^{S^{M}} (t_{j}-a_{j})^{2}}{\partial n_{i}^{M}} = -2(t_{i}-a_{i})\frac{\partial a_{i}}{\partial n_{i}^{M}} \qquad \ldots(3\text{-}24)$$

เนื่องจาก

$$\frac{\partial a_{i}}{\partial n_{i}^{M}} = \frac{\partial a_{i}^{M}}{\partial n_{i}^{M}} = \frac{\partial f^{m}(n_{i}^{m})}{\partial n_{i}^{M}} = \dot{f}^{M}(n_{i}^{M}) \qquad \ldots(3\text{-}25)$$

จะได้ว่า

$$s_{i}^{M} = -2(t_{i}-a_{i})\dot{f}^{M}(n_{i}^{M}) \qquad \ldots(3\text{-}26)$$

ดังนั้นจะสามารถสรุปขั้นตอนการแพร่กลับของโครงข่ายได้ดังนี้ เมื่อป้อนอินพุตให้กับโครงข่ายจะได้

$$a_{0} = p \qquad \ldots(3\text{-}27)$$

$$a^{m+1} = f^{m+1}(W^{m+1}a^{m} + b^{m+1}) \qquad \ldots(3\text{-}28)$$

จากสมการ (3-28) สำหรับ m = 0, 2, ..., m-1 ฉะนั้น

$$a = a^{m} \qquad \text{...(3-29)}$$

ขั้นตอนต่อไปเป็นการแพร่ค่าความไวกลับเข้าสู่โครงข่าย

$$s^{M} = -2F^{M}(n^{M})(t - a) \qquad \text{...(3-30)}$$

$$s^{m} = F^{m}(n^{m})(W^{m+1})^{T} s^{m+1} \qquad \text{...(3-31)}$$

จากสมการ (3-31) สำหรับ m = M-1, ..., 2, 1 และท้ายที่สุดจะได้ค่าน้ำหนักและไบอัสเท่ากับ

$$W^{m}(k+1) = W^{m}(k) - \alpha s^{m}(a^{m-1})^{T} \qquad \text{...(3-32)}$$

$$b^{m}(k+1) = b^{m}(k) - \alpha s^{m} \qquad \text{...(3-33)}$$

โดยที่ S^{m}_{i} คือความไวที่เปลี่ยนเทอมที่ i ของอินพุตชั้นที่ m

α คือ อัตราการเรียนรู้

$F(x)$ คือค่าเฉลี่ยของความผิดพลาดกำลังสอง

e คือค่าความผิดพลาด

t คือค่าเป้าหมาย

a คือค่าเอาต์พุตที่ได้จากการ feedforward ของรอบนั้นๆ

W คือเมตริกซ์น้ำหนัก

b คือค่าไบอัส

3.3.4 โครงสร้างและสถาปัตยกรรม

นิวรอลเน็ตเวอร์กแบบนี้เป็นโครงข่ายที่สามารถที่จะมีชั้นซ่อนได้มากกว่าหนึ่งชั้นซ่อน เป็นนิวรอลเน็ตเวอร์กที่ประกอบไปด้วยส่วนหลักๆ 3 ส่วนคือ ชั้นอินพุต (X) ชั้นซ่อน (Z) และชั้นเอาต์พุต (Y) ดังภาพประกอบ 3-4 โครงสร้างการทำงานในส่วนของการเรียนรู้จะแบ่งออกเป็น 3 ขั้นตอนใหญ่ๆ คือ ขั้นตอนการส่งข้อมูลแพร่ไปข้างหน้า (feedforward) โดยมีข้อมูลอินพุตคือข้อมูลที่ต้องการให้โครงข่ายเรียนรู้ ต่อมาเป็นขั้นตอนของการแพร่กลับโดยนำค่าความผิดพลาดที่ได้

จากการคำนวณครั้งสุดท้ายนำกลับมาใช้คำนวณในการแพร่กลับอีกครั้ง และขั้นตอนสุดท้ายจะเป็นการปรับค่าของน้ำหนักในแต่ละชั้นที่ได้จากการคำนวณในรอบนั้น เพื่อให้เป็นค่าน้ำหนักของโครงข่ายในปัจจุบัน

การทำงานของนิวรอลเน็ตเวอร์กแบบแพร่กลับนี้จะต้องมีการเตรียมการหรือต้องการข้อมูลพื้นฐานบางอย่างก่อน จึงจะสามารถทำงานได้ การเตรียมการพื้นฐานดังกล่าวคือ

- ต้องทราบจำนวนของชั้นซ่อน (hidden layer) ที่จะใช้งานก่อน
- ต้องทราบจำนวนเซลล์ในแต่ละชั้นซ่อนว่ามีจำนวนเท่าใด
- ต้องทราบจำนวนเซลล์ของชั้นเอาต์พุต
- ต้องทราบค่าความผิดพลาด (error) เพื่อให้เป็นจุดสิ้นสุดการเรียนรู้ของโครงข่าย

การเรียนรู้ของนิวรอลเน็ตเวอร์กแบบแพร่กลับนี้จะมีการทำงานโดยเริ่มจาก ระบบจะทำการสุ่มค่าน้ำหนักเริ่มต้นสำหรับทุกๆ ชั้นเพื่อใช้ในการคำนวณเสียก่อน จากนั้นจึงรับอินพุต (X_i) เข้ามา แล้วจึงทำการแพร่อินพุตผ่านค่าน้ำหนักที่มีอยู่แล้วส่งผลไปยังแต่ละเซลล์ของชั้นซ่อนที่ 1 (Z_j) ก่อนที่จะส่งไปถึงชั้นซ่อนที่หนึ่งได้นั้นต้องมีการผ่านฟังก์ชันโอนย้าย (transfer function) อันหนึ่งก่อน ในกรณีที่มาชั้นซ่อนมากกว่า 1 ชั้นซ่อน ก็ให้ทำแบบเดิมโดยใช้ข้อมูลเข้าเป็นข้อมูลที่มาจากเอาต์พุตของชั้นก่อนหน้า เมื่อทำการแพร่ไปถึงชั้นซ่อนชั้นสุดท้าย ผลที่ได้ออกมาจากชั้นซ่อนชั้นสุดท้ายนั้นก็จะเป็นชั้นเอาต์พุต (Y_k) ของระบบ จากนั้นจึงนำค่าจากชั้นเอาต์พุตไปเปรียบเทียบกับค่าเป้าหมาย (t_k) ในแต่ละเซลล์ ซึ่งค่าความแตกต่างที่ได้จะหมายถึงค่าความผิดพลาด (error) ของรอบการคำนวณนั้น ค่าความผิดพลาดที่ได้นั้นจะถูกใช้ในการคำนวณในขั้นตอนของการแพร่กลับในภายหลัง โดยใช้ค่าความผิดพลาดมาคำนวณร่วมกับค่าของเซลล์ในแต่ละชั้น ผลที่ได้จากการคำนวณจะนำไปใช้ในการปรับน้ำหนัก w_{jk} และ v_{ij} ดังภาพประกอบ 3-4

บทที่ 4

การออกแบบ

4.1 ความเบื้องต้น

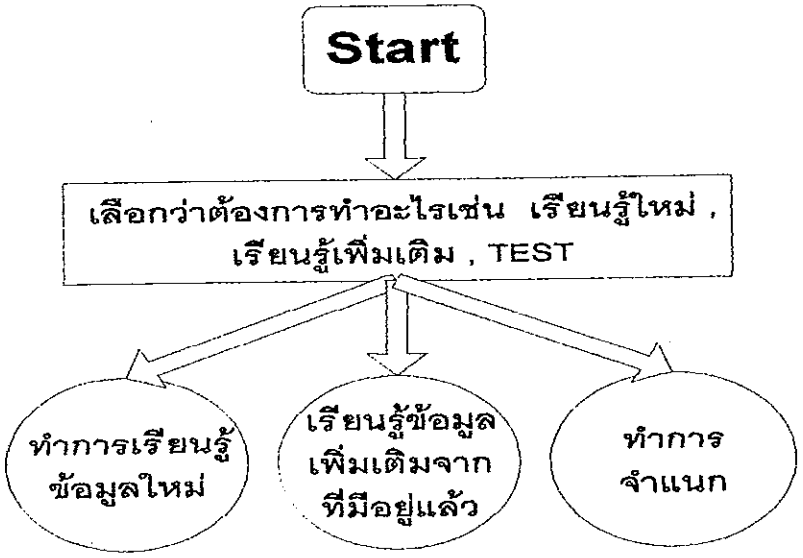
การใช้งานกระบวนการจำแนกลายวงจรพิมพ์ให้ได้ประสิทธิภาพที่ดีนั้น จะขึ้นอยู่กับอัลกอริทึมของนิวรอลเน็ตเวอร์ก การปรับปรุงภาพลายวงจรพิมพ์ และความละเอียดของอุปกรณ์รับภาพที่ใช้ในการรับภาพลายวงจรเข้ามา เนื่องจากในการวิจัยนี้ได้จำกัดอุปกรณ์รับภาพเป็นกล้อง CCD ในการออกแบบกระบวนการทำงานทั้งหมดจึงต้องเริ่มที่รับภาพและทำการปรับปรุงภาพลายวงจรพิมพ์นั้น

4.2 อุปกรณ์ที่ใช้ในการทดลอง

- คอมพิวเตอร์รุ่น เพนเทียม 150 เมกะเฮิร์ซ ที่มีหน่วยความจำแรม 32 เมกะไบต์
- capture TV ยี่ห้อ Tekram รุ่น M205
- กล้อง CCD ที่สามารถรับภาพได้ขนาด 320 x 240 จุด สามารถรับภาพเข้ามาเป็นภาพระดับเทา ซึ่งมีค่าความละเอียดอยู่ที่ 96 จุดต่อนิ้ว

4.3 การออกแบบกระบวนการจำแนก

ในขั้นตอนการออกแบบกระบวนการทั้งหมดมีรูปแบบของการทำงานที่สามารถสรุปได้ 3 รูปแบบคือ ทำการเรียนรู้ข้อมูลใหม่ ทำการเรียนรู้ข้อมูลเพิ่มเติมจากที่มีอยู่แล้วและการทำการจำแนก ดังภาพประกอบ 4-1 ซึ่งจะแสดงให้เห็นถึงลักษณะงานโดยรวมของการจำแนกลายวงจรพิมพ์ โดยการทำงานของทั้งสามรูปแบบก็ต้องอาศัยขั้นตอนที่สำคัญมาประกอบกัน โดยมีขั้นตอนในการทำงานที่คล้ายกันดังนี้คือ การรับภาพ ปรับปรุงภาพลายวงจรพิมพ์ การใช้นิวรอลเน็ตเวอร์กเรียนรู้ภาพลายวงจรพิมพ์ และการใช้นิวรอลเน็ตเวอร์กในการจำแนก



ภาพประกอบ 4-1 แสดงรูปแบบการทำงานโดยรวมของการจำแนกลาย วงจรพิมพ์

4.3.1 การออกแบบการรับภาพลายวงจรพิมพ์จากกล้อง CCD

การออกแบบกระบวนการทำงานทั้งหมดจึงต้องเริ่มที่รับภาพและทำการปรับปรุงภาพลายวงจรพิมพ์นั้น จากนั้นจึงจะเริ่มทำงานในส่วนของนิวรอลเน็ตเวอร์กเพื่อการเรียนรู้และจำแนก ในขั้นตอนการรับภาพและปรับปรุงภาพนี้เป็นส่วนที่ต้องพิจารณาด้วยตนเองจะให้ประสิทธิภาพที่ดีกว่าการทำเป็นอัตโนมัติ ด้วยการพิจารณาว่าภาพที่ได้สามารถนำไปใช้เรียนรู้หรือจำแนกได้หรือยัง ถ้ายังนำไปใช้ไม่ได้ก็จะต้องผ่านการปรับปรุงภาพด้วยวิธีต่างๆ วิธีใดบ้างที่ต้องใช้ในการปรับปรุงภาพเพื่อให้ภาพนั้นสามารถนำไปใช้ในขั้นตอนต่อไปได้ ความรู้ที่ได้จากบทที่ 2 เกี่ยวกับการรับและปรับปรุงภาพ ได้นำมาสู่การออกแบบในส่วนนี้ ในการรับภาพลายวงจรพิมพ์ต้องรับในขนาด 320 x 240 จุด ซึ่งเป็นภาพระดับเทา เราจึงออกแบบให้เมตริกซ์ที่ใช้ในการทำงานมีขนาดที่เท่ากัน แล้วนำไปเก็บในรูปของไฟล์ข้อมูลหนึ่ง ในส่วนของการรับภาพลายวงจรพิมพ์จะมีขั้นตอนทั้งหมดดังนี้

(1) เปิดแฟ้มข้อมูลภาพ

(2) อ่านข้อมูล header ของภาพและเก็บข้อมูลที่ต้องใช้ในการทำงาน เช่น ขนาดของภาพ (กว้าง,ยาว), ชนิดของภาพ (bmp, tif ฯลฯ), บิตต่อจุด ฯลฯ

(3) อ่านข้อมูลภาพครั้งละแถว นั่นคือครั้งละ 240 จุดภาพ จนครบทั้ง 320 แถว นำไปเก็บในอาเรย์แบบเมตริกซ์ เพื่อใช้ในการปรับปรุงภาพต่อไป

(4) ปิดแฟ้มข้อมูลภาพ โดยข้อมูลภาพที่รับเข้ามามีจุดเริ่มต้นของคู่ลำดับอยู่ที่มุมบนด้านซ้าย ซึ่งข้อมูลภาพที่เก็บเข้ามาจะเรียงจากมุมบนซ้ายไปทางขวา และจากบนลงล่าง

โปรแกรมที่พัฒนาขึ้นนี้เราจะใช้ชื่อว่า โปรแกรมการรับภาพลายวงจรพิมพ์ โดยโปรแกรมจะมีขั้นตอนการทำงานดังที่กล่าวไว้แล้ว

4.3.2 การออกแบบการปรับปรุงภาพลายวงจรพิมพ์

4.3.2.1 Image Thresholding

ขั้นตอนนี้เป็นการตัดพื้นหลังของลายวงจรพิมพ์ เพื่อให้เหลือเพียงลายวงจรพิมพ์ที่เป็นลายทองแดงเท่านั้น ในขั้นตอนนี้อาจจะต้องมีการ thresholding 2 ระดับใน 1 ภาพ เนื่องจากค่าความสว่างที่ให้กับภาพในขณะรับภาพมีค่าไม่เท่ากัน นั่นคือ บางที่มืดมาก แต่บางที่สว่างมาก ซึ่งเป็นสิ่งที่เกิดขึ้นได้ ขั้นตอนการทำ image thresholding มีดังนี้

1) เปิดไฟล์ภาพที่ต้องการใช้งาน

2) เลือกค่า T เพื่อใช้ในการแบ่งภาพพื้นหลังกับลายวงจรพิมพ์ออกจากกัน

3) แสดงภาพหลังจากทำ thresholding แล้ว เมื่อเห็นว่าภาพที่ได้สามารถแสดงถึงลายวงจรพิมพ์ได้ชัดที่สุดแล้วทำข้อ 4) ถ้ายังไม่ดีพอ ทำข้อ 2) อีกครั้ง

4) เขียนภาพที่ได้ลงไฟล์

5) ปิดไฟล์ที่เปิดทั้งหมด

วิธีการปรับปรุงภาพวิธีนี้ จะเป็นการนำค่าระดับเทาของทุก pixel มาเปรียบเทียบกับค่า threshold ที่ตั้งไว้ ถ้าค่าระดับเทามีค่ามากกว่าค่า threshold ค่าระดับเทาที่ pixel นั้นจะถูกเปลี่ยนค่าเป็น 1 ถ้าน้อยกว่าหรือเท่ากับค่า threshold ให้เปลี่ยนค่าที่ pixel นั้นเป็น 0 จะเห็นว่าผลที่ได้จากการใช้งานจะเป็นภาพแบบไบนารี คือมีเพียงสีขาวและดำเท่านั้นที่แสดงอยู่ในภาพ โปรแกรมที่พัฒนาขึ้นนี้เราจะใช้ชื่อว่า โปรแกรม threshold โดยโปรแกรมจะมีขั้นตอนการทำงานดังที่กล่าวไว้แล้ว

4.3.2.2 Brightness

ขั้นตอนนี้เป็นการเพิ่มหรือลดค่าความสว่างให้กับภาพลายวงจรพิมพ์ที่มีอยู่ ใช้สำหรับกรณีที่ภาพที่รับเข้ามามีค่าความสว่างไม่เพียงพอ ขึ้นอยู่กับการสังเกตของผู้ใช้งานเอง

ว่าความสว่างที่มีอยู่นั้นเพียงพอกับการใช้งานหรือไม่ ขั้นตอนนี้ควรทำก่อนการทำ image thresholding ซึ่งมีขั้นตอนการทำงานดังนี้

1) เปิดไฟล์ภาพที่ต้องการใช้งาน

2) เลือกค่าความสว่างที่ต้องการเพิ่มหรือลดในภาพนั้น

3) แสดงภาพหลังจากทำการลดหรือเพิ่มความสว่างให้กับภาพนั้นๆ ถ้าเห็นว่าภาพที่ได้สามารถแสดงถึงลายวงจรพิมพ์ได้ชัดที่สุดแล้วทำข้อ 4) ถ้ายังไม่ดีพอ ทำข้อ 2) อีกครั้ง

4) เขียนภาพที่ได้ลงไฟล์

5) ปิดไฟล์ที่เปิดทั้งหมด

โปรแกรมที่พัฒนาขึ้นนี้เราจะใช้ชื่อว่า โปรแกรม brightness โดยโปรแกรมจะมีขั้นตอนการทำงานดังที่กล่าวไว้แล้ว

4.3.2.3 Contrast Correction

ขั้นตอนนี้เป็นเครื่องมือที่จะช่วยให้ภาพลายวงจรพิมพ์เด่นชัดขึ้น โดยสามารถแบ่งแยกลายวงจรกับภาพพื้นหลังให้ชัดเจนขึ้น โดยใช้การคูณทุกๆ จุดในภาพด้วยค่าค่าหนึ่ง การกระทำต่อภาพด้วยวิธีนี้ ค่าของทุกจุดในภาพจะมีค่าเพิ่มขึ้นเป็นทวีคูณตามค่าที่ได้คูณเข้าไป ซึ่งจะส่งผลให้ค่าระดับเทาในภาพมีความแตกต่างกันมากขึ้น มีขั้นตอนการทำงานดังนี้

1) เปิดไฟล์ภาพที่ต้องการใช้งาน

2) เลือกค่าตัวคูณที่ต้องการใช้ (ควรใช้ค่าไม่เกิน 2)

3) แสดงภาพหลังจากทำการคูณให้กับภาพนั้นๆ เมื่อเห็นว่าภาพที่ได้สามารถแสดงถึงลายวงจรพิมพ์ได้ชัดที่สุดแล้วทำข้อ 4) ถ้ายังไม่ดีพอ ทำข้อ 2) อีกครั้ง

4) เขียนภาพที่ได้ลงไฟล์

5) ปิดไฟล์ที่เปิดทั้งหมด

โปรแกรมที่พัฒนาขึ้นนี้เราจะใช้ชื่อว่า โปรแกรม contrast โดยโปรแกรมจะมีขั้นตอนการทำงานดังที่กล่าวไว้แล้ว

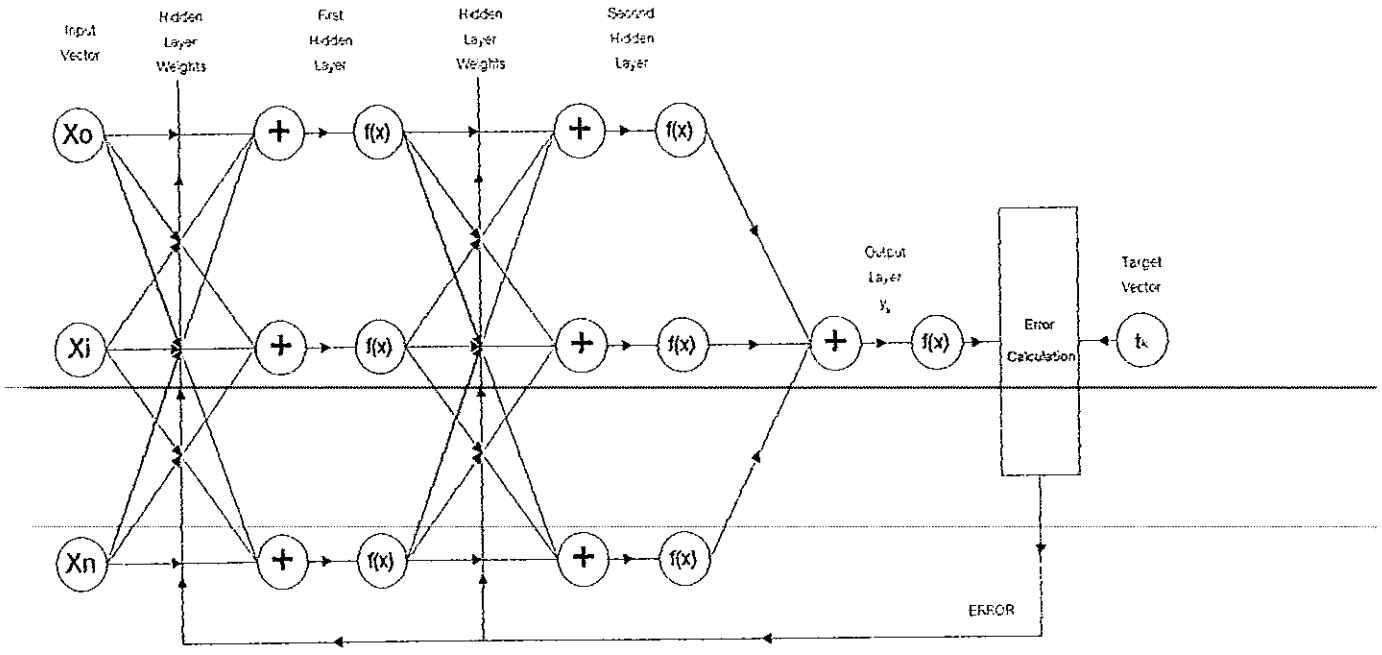
4.4 การออกแบบนิวรอลเน็ตเวอร์กแบบแพร่กลับ

4.4.1 นิวรอลเน็ตเวอร์กในส่วนการส่งค่าไปข้างหน้า

การออกแบบการทำงานในส่วนนี้จะเป็นส่วนหนึ่งของการทำงานในนิวรอลเน็ตเวอร์กแบบแพร่กลับ มีหน้าที่ในการคำนวณด้วยการคูณ การบวกและทำการส่งค่าที่ได้ผ่านฟังก์ชันโอน

ย้าย ทำเช่นนี้ไปจนกระทั่งส่งค่าถึงชั้นเอาต์พุตก็เป็นอันสิ้นสุดขั้นตอนการทำงานของการส่งค่าไปข้างหน้า (feedforward) โดยโปรแกรมที่พัฒนาขึ้นนี้จะเป็นโครงข่ายประสาทที่สามารถใช้จำนวนของชั้นซ่อนได้ 1-5 ชั้น แต่ในการทดลองจะใช้จำนวนชั้นซ่อน 1-2 ชั้น ดังภาพประกอบ 4-2 ในแต่ละชั้นซ่อนสามารถกำหนดจำนวนเซลล์ได้ 1-120,000 เซลล์ โดยที่มีการรับค่าองค์ประกอบในการทำงานของนิวรอนเน็ตเวอร์กในช่วงของการเรียนรู้ข้อมูลใหม่ คือ ค่าความผิดพลาด (ในการทำการศึกษานี้ใช้ค่าความผิดพลาดที่ 0.13) จำนวนชั้นซ่อน จำนวนเซลล์ในแต่ละชั้นซ่อน

ก่อนที่จะมีการส่งข้อมูลภาพที่ได้จากการรับภาพและปรับปรุงภาพไปให้นิวรอลเน็ตเวอร์กแบบแพร่กลับทำการเรียนรู้และทดสอบนั้น จะต้องมีการจัดเรียงข้อมูลภาพที่ได้ให้อยู่ในรูปแบบของเมตริกซ์ 1 หลักเสียก่อนแล้วจึงส่งไปให้นิวรอลเน็ตเวอร์กทำงานต่อไป เอาต์พุตที่ออกจากนิวรอลเน็ตเวอร์กจะมีขนาด 1 เซลล์เป็นตัวเลขจำนวนจริงที่มีค่าอยู่ในช่วง 0-1 โดยที่ 1 แสดงถึงแผ่นวงจรพิมพ์ที่ดีเหมือนต้นแบบที่ได้เรียนรู้ที่สุด ส่วนค่าเอาต์พุตที่เป็น 0 แสดงถึงแผ่นวงจรพิมพ์ที่ต่างจากต้นแบบที่เรียนรู้ไว้ เมื่อค่าเอาต์พุตที่ได้เกินกว่า 0.6 (เนื่องจาก ที่ค่าเอาต์พุตต่ำกว่า 0.6 อาจจะมีแผ่นลายวงจรพิมพ์ที่เสียถูกจำแนกเป็นแผ่นที่ดี) จึงถือว่าเป็นแผ่นลายวงจรพิมพ์ที่ใช้ได้



ภาพประกอบ 4-2 แสดงรูปแบบการส่งค่าไปข้างหน้าของนิวรอลเน็ตเวอร์ก 2 ชั้นซ่อน

การทำงานของนิวรอลเน็ตเวอร์กแบบแพร่กลับจะมีการออกแบบขั้นตอนในการทำงานดังนี้

1) นำทุกๆ ค่าอินพุตในชั้นนั้น คูณกับค่าน้ำหนักในชั้นนั้น แล้วนำผลที่ได้ทั้งหมดมารวมกันดังสมการที่ (4-1)

$$z\_in_j = v_{0j} + \sum_{i=1}^{n} x_i v_{ij} \qquad \text{...(4-1)}$$

โดยที่ z_in_j หมายถึงค่าที่ได้จากการคูณน้ำหนักกับอินพุต

v_{0j} หมายถึงค่าไบอัส

x_i หมายถึงอินพุตที่ใส่เข้าไปคำนวณ

v_{ij} หมายถึงค่าน้ำหนักในชั้นนั้นนั้น

2) นำค่าที่ได้จาก 1) ในทุกๆ ค่าไปผ่านฟังก์ชันโอนย้ายเพื่อให้ได้เป็นค่าเอาต์พุตของการส่งค่าไปข้างหน้าของชั้นนั้นๆ

3) เก็บค่าที่ได้จากทุกชั้นไว้ในไฟล์เพื่อนำไปใช้ในการแพร่กลับในครั้งต่อไป

โปรแกรมที่พัฒนาขึ้นนี้เราจะใช้ชื่อว่า โปรแกรมนิวรอลเน็ตเวอร์กในส่วนการส่งค่าไปข้างหน้า โดยโปรแกรมจะมีขั้นตอนการทำงานดังที่กล่าวไว้แล้ว

4.4.2 นิวรอลเน็ตเวอร์กในส่วนการแพร่กลับของค่าความผิดพลาด

เป็นส่วนที่จะทำการคำนวณค่าความไว และค่าความผิดพลาดและส่งองค์ประกอบของค่าความผิดพลาดนั้นไปใช้ในการหาค่าเพื่อการแพร่กลับซึ่งจะมีขั้นตอนดังนี้

1) นำค่าเป้าหมาย (t_k) ไปลบกับค่าที่ได้จากการ feedforward ของชั้นสุดท้าย (y_k)

2) คำนวณหา δ_k จากสมการ (4-2)

$$\delta_k = (t_k - y_k) f'(y\_in_k) \qquad \text{...(4-2)}$$

3) คำนวณหาค่าน้ำหนักที่เปลี่ยนแปลงไปจากสมการ (4-3)

$$\Delta w_{jk} = \alpha \delta_k z_j \qquad \text{...(4-3)}$$

4) จากนั้นจึงจะคำนวณค่าของน้ำหนักที่เปลี่ยนแปลงในชั้นต่อมาได้จาก

$$\delta\_in_j = \sum_{k=1}^{m} \delta_k w_{jk} \qquad \text{...(4-4)}$$

$$\delta_j = \delta\_in_j f'(z\_in_j) \quad \ldots(4\text{-}5)$$

น้ำหนักที่เปลี่ยนแปลงในชั้นนี้คือ

$$\Delta v_{ij} = \alpha \delta_j x_i \quad \ldots(4\text{-}6)$$

5) ทำขั้นตอนที่ 4) จนกระทั่งถึงชั้นที่มีค่าน้ำหนักชั้นอินพุต

6) เก็บค่าน้ำหนักที่เปลี่ยนไปที่ได้จากทุกชั้นไว้ในไฟล์เพื่อนำไปใช้ในปรับค่าน้ำหนักก่อนที่จะทำการ feedforward ในครั้งต่อไป

โดยที่ f' คือการทำอนุพันธ์

δ_j คือค่าความไวที่จะนำไปคำนวณในส่วนของการแพร่กลับ

Z_j คือค่าในชั้นซ่อนที่ต้องการแพร่กลับมีจำนวน j เซลล์

X_i คือค่าของเซลล์ในชั้นอินพุตจำนวน i เซลล์

โปรแกรมที่พัฒนาขึ้นนี้เราจะใช้ชื่อว่า โปรแกรมนิวรอลเน็ตเวอร์กในส่วนการแพร่กลับของค่าความผิดพลาด โดยโปรแกรมจะมีขั้นตอนการทำงานดังที่กล่าวไว้แล้ว

4.4.3 นิวรอลเน็ตเวอร์กในส่วนการปรับค่าน้ำหนัก

ในส่วนนี้จะเป็นการนำค่าน้ำหนักที่คำนวณได้มาทำการปรับเปลี่ยนร่วมกับค่าน้ำหนักเดิมในแต่ละชั้นซ่อน เพื่อให้ได้เป็นค่าน้ำหนักใหม่ในการนำไปใช้คำนวณในรอบต่อไป เนื่องจากในการปรับค่าน้ำหนักจะเป็นการปรับข้อมูลทั้งหมด จำนวนค่าน้ำหนักที่เปลี่ยนไปในแต่ละชั้นซ่อนจะต้องมีจำนวนเท่ากับค่าน้ำหนักเดิม ซึ่งจะมีขั้นตอนการทำงานดังนี้

1) อ่านค่าน้ำหนักจากไฟล์ และค่าน้ำหนักที่เปลี่ยนแปลงจากผลของการแพร่กลับค่าความผิดพลาด

2) ทำการปรับค่าน้ำหนักในแต่ละชั้น ดังสมการ 4-7 และ4-8

$$v_{ij}(new) = v_{ij}(old) + \Delta v_{ij} \quad \ldots(4\text{-}7)$$

$$w_{jk}(new) = w_{jk}(old) + \Delta w_{jk} \quad \ldots(4\text{-}8)$$

โดยที่ Δv_{ij} คือค่าน้ำหนักของชั้นซ่อนที่ 1

Δw_{jk} คือค่าน้ำหนักของชั้นเอาต์พุต

3) เขียนค่าน้ำหนักที่ได้ใหม่ลงไฟล์เพื่อใช้ในการ feedforward ในครั้งต่อไป

โปรแกรมที่พัฒนาขึ้นนี้เราจะใช้ชื่อว่า โปรแกรมนิวรอลเน็ตเวอร์กในส่วนการปรับค่าน้ำหนัก โดยโปรแกรมจะมีขั้นตอนการทำงานดังที่กล่าวไว้แล้ว

4.5 สรุปการออกแบบระบบการทำงาน

เนื่องจากการทำงานของระบบจำแนกโดยรวมมีทางเลือกให้ทำได้ 3 อย่างคือ ทำการเรียนรู้ข้อมูลใหม่ ทำการเรียนรู้ข้อมูลเพิ่มเติมจากที่มีอยู่แล้วและการทำการจำแนก ดังที่กล่าวมาแล้วในตอนต้นของบทนี้ ซึ่งใน 3 ทางเลือกนั้นจะประกอบไปด้วยขั้นตอนบางขั้นตอนของเนื้อหาในบท ดังนี้

4.5.1 การทำการเรียนรู้ข้อมูลใหม่จะประกอบด้วย

1) โปรแกรมการรับภาพลายวงจรพิมพ์

2) โปรแกรมการปรับปรุงภาพลายวงจรพิมพ์

3) โปรแกรมนิวรอลเน็ตเวอร์กในส่วนการส่งค่าไปข้างหน้า

4) โปรแกรมนิวรอลเน็ตเวอร์กในส่วนการแพร่กลับของค่าความผิดพลาด

5) โปรแกรมนิวรอลเน็ตเวอร์กในส่วนการปรับค่าน้ำหนัก

4.5.2 การทำการเรียนรู้ข้อมูลเพิ่มเติมจากที่มีอยู่แล้ว

1) โปรแกรมการออกแบบการรับภาพลายวงจรพิมพ์

2) โปรแกรมการปรับปรุงภาพลายวงจรพิมพ์

3) เก็บข้อมูลภาพใหม่ที่ต้องการเรียนรู้ลงไฟล์ข้อมูลภาพที่ใช้ในการเรียนรู้

4) อ่านข้อมูลไฟล์น้ำหนักเดิมที่ได้เรียนรู้เอาไว้ก่อนแล้ว

5) โปรแกรมนิวรอลเน็ตเวอร์กในส่วนการส่งค่าไปข้างหน้า

6) โปรแกรมนิวรอลเน็ตเวอร์กในส่วนการแพร่กลับของค่าความผิดพลาด

7) โปรแกรมนิวรอลเน็ตเวอร์กในส่วนการปรับค่าน้ำหนัก

4.5.3 การทำการจำแนก

1) โปรแกรมการรับภาพลายวงจรพิมพ์

2) โปรแกรมการปรับปรุงภาพลายวงจรพิมพ์

3) จัดข้อมูลภาพเพื่อใช้ในการจำแนก

4) อ่านข้อมูลไฟล์น้ำหนักเดิมที่ได้เรียนรู้เอาไว้ก่อนแล้ว

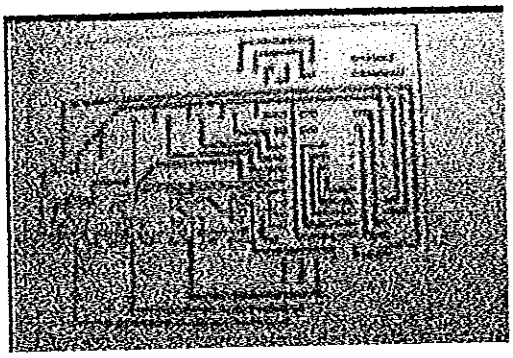
5) โปรแกรมนิวรอลเน็ตเวอร์กในส่วนการส่งค่าไปข้างหน้า

บทที่ 5

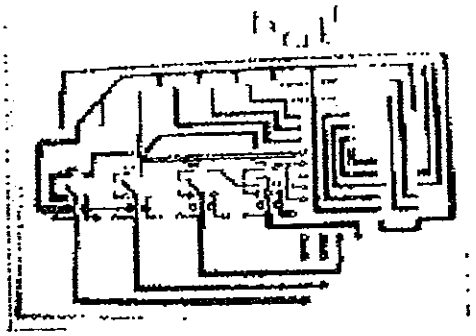
ผลการทดลอง

5.1 ผลการปรับปรุงภาพ

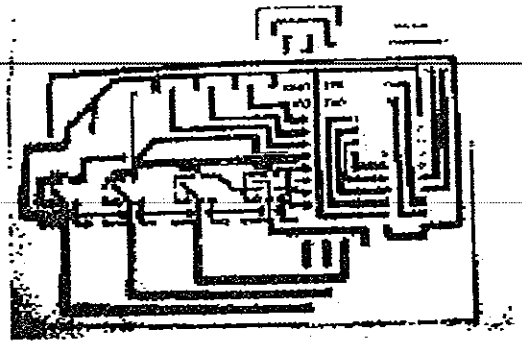
เนื่องจากกระบวนการจำแนกได้ใช้วิธีการเดียวในการปรับปรุงภาพคือ thresholding เพื่อที่จะทำการตัดในส่วนพื้นหลังของภาพออกไป ให้เหลือเพียงเส้นลายวงจรพิมพ์ ผลจากการทดสอบในส่วนของการปรับปรุงภาพด้วยวิธีการ thresholding ทั้งในแบบที่ใช้ค่า threshold ค่าเดียวและหลายค่าในภาพ จะพบว่าวิธีที่ใช้สามารถที่จะตัดภาพพื้นหลังของภาพลายวงจรพิมพ์ได้ ทำให้ข้อมูลภาพที่ได้จากการทำ thresholding เหลือเพียงข้อมูล 1 คือพิกเซลที่เป็นสีดำ กับ 0 คือพิกเซลที่เป็นสีขาวเท่านั้น ซึ่งข้อมูลภาพที่เหลือที่ได้เป็นสีดำนั้นก็คือลายวงจรพิมพ์ที่ต้องการนำไปให้นิวรอลเน็ตเวอร์กเรียนรู้ และทำการจำแนก



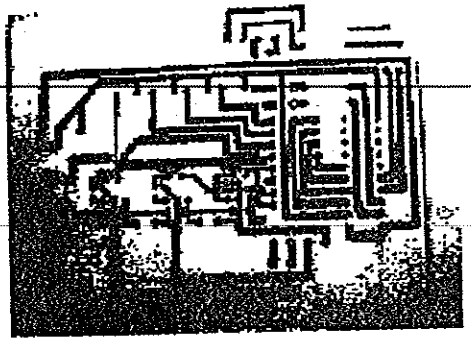
(ก)



(ข)



(ค)



(ง)

ภาพประกอบ 5-1 แสดงผลจากการทำ thresholding ค่าเดียวต่อภาพ จากภาพบนกระดาษ

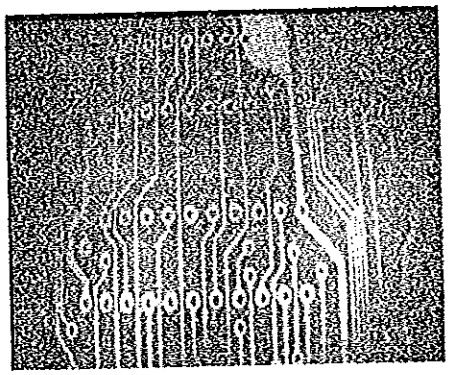
(ก) ภาพที่อ่านได้จากกล้อง CCD ก่อนทำการ thresholding

(ข) ภาพที่ได้หลังจากการทำ thresholding เมื่อค่า threshold = 120

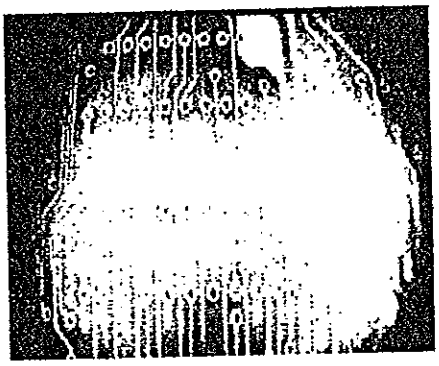
(ค) ภาพที่ได้หลังจากการทำ thresholding เมื่อค่า threshold = 140

(ง) ภาพที่ได้หลังจากการทำ thresholding เมื่อค่า threshold = 160

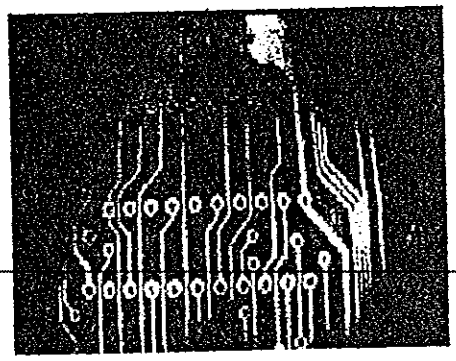
เมื่อทดลองใช้กับภาพลายวงจรพิมพ์ที่รับภาพมาจากลายวงจรบนกระดาษ และเมื่อทำการรับภาพลายวงจรจริงจากแผ่นวงจรพิมพ์ จะพบว่าเมื่อรับภาพจากแผ่นวงจรพิมพ์จริง จะมีปัญหาเกี่ยวกับความสม่ำเสมอของแสงที่ให้กับภาพในตอนที่รับภาพเข้ามา รวมทั้งเกิดการสะท้อนแสงที่ไม่เท่ากันทั้งภาพ ดังภาพประกอบ 5-2



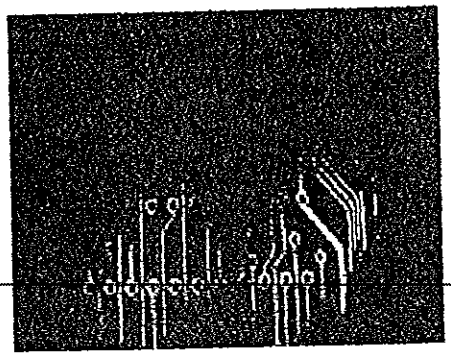
(ก)



(ข)



(ค)



(ง)

ภาพประกอบ 5-2 แสดงผลจากการทำ thresholding ค่าเดียวต่อภาพ จากภาพลายวงจรจริง

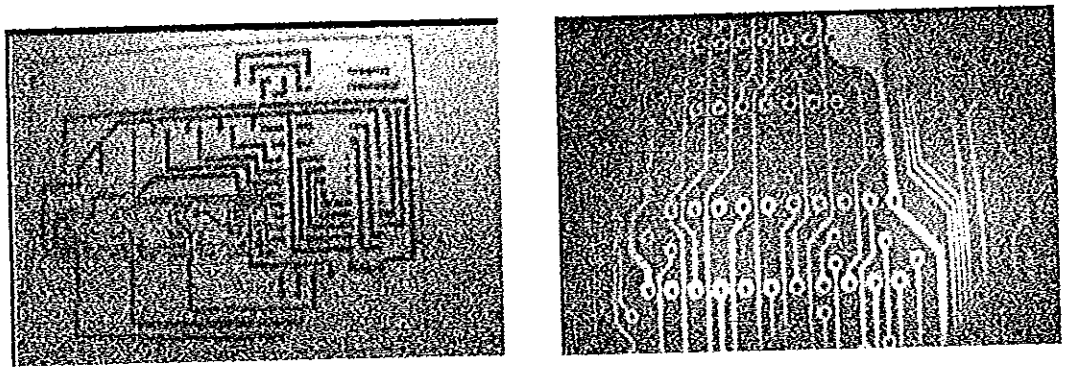
(ก) ภาพที่อ่านได้จากกล้อง CCD ก่อนทำการ thresholding

(ข) ภาพที่ได้หลังจากการทำ thresholding เมื่อค่า threshold = 150

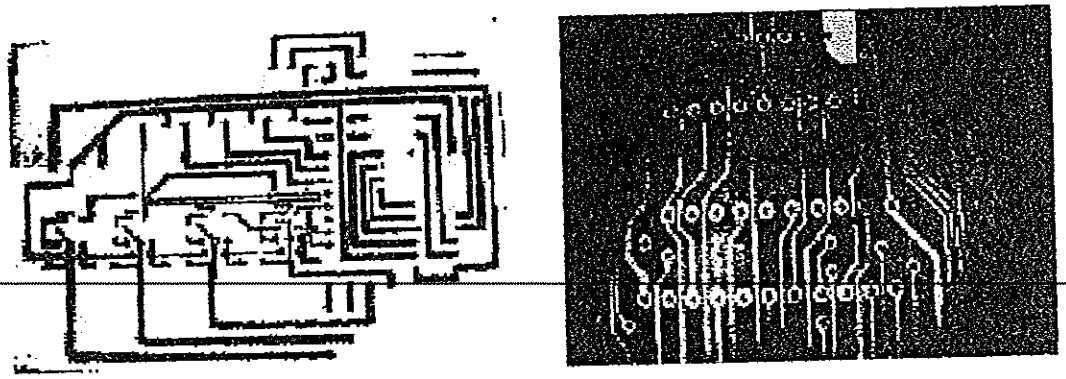
(ค) ภาพที่ได้หลังจากการทำ thresholding เมื่อค่า threshold = 180

(ง) ภาพที่ได้หลังจากการทำ thresholding เมื่อค่า threshold = 250

จากผลที่ได้จากการทำ thresholding ในแบบที่ใช้ค่า threshold เดียวกันทั้งภาพจะพบว่าภาพที่ได้เพื่อใช้ในการเรียนรู้ของนิวรอลเน็ตเวอร์กมีข้อมูลบางช่วงขาดหายไป อันเนื่องมาจากการให้แสงสว่างในภาพลายวงจรพิมพ์เพื่อใช้กับกล้อง ccd ให้ไม่ทั่วถึงทั้งแผ่นลายวงจรพิมพ์ ความสว่างจึงเป็นช่วงๆ ในขั้นตอนการ thresholding จึงต้องมีการใช้ค่า threshold หลายๆ ค่าในภาพเดียว ซึ่งจะได้ผลดังภาพประกอบ 5-3



(ก)



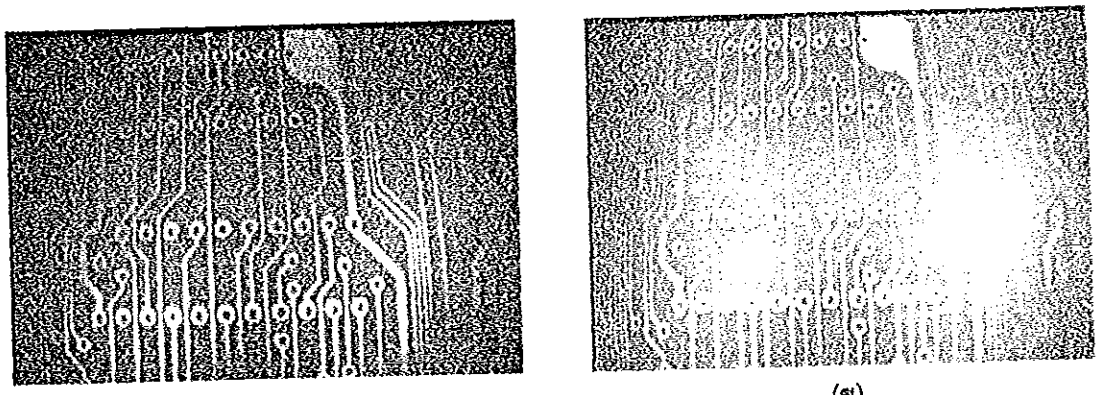
(ข)

ภาพประกอบ 5-3 แสดงผลจากการทำ thresholding สองค่าต่อภาพ

(ก) ภาพที่อ่านได้จากกล้อง CCD ก่อนทำการ thresholding

(ข) ภาพที่ได้หลังจากการทำ thresholding สองค่าต่อภาพ

ผลที่ได้จากการปรับปรุงภาพลายวงจรพิมพ์ด้วยวิธีการเพิ่มค่าความสว่างให้กับภาพ จะพบว่าสามารถทำให้เห็นรายละเอียดภายในภาพได้ชัดเจนขึ้น ดังภาพประกอบ 5-4 แต่ในการเพิ่มค่าความสว่างให้กับภาพนั้น ไม่สมควรที่จะเพิ่มค่าความสว่างมากจนเกินไป เพราะจะทำให้รายละเอียดหรือองค์ประกอบบางอย่างของภาพหายไปได้ โดยเฉพาะในส่วนที่อยู่ด้านที่ใกล้ขอบภาพ เนื่องจากภาพที่ได้รับมาจากกล้อง ccd ก็มีความผิดเพี้ยนของการมองอันเนื่องมาจากความโค้งของเลนส์มากพอสมควรอยู่แล้ว ถ้าทำการเพิ่มความสว่างมากเกินไปจะทำให้ในส่วนกลางของภาพกลายเป็นบริเวณที่เป็นสีขาวจนหมด ข้อมูลภาพจะเสียไปทันที



(ก) (ข)

ภาพประกอบ 5-4 แสดงผลจากการทำ brightness

(ก) ภาพที่อ่านได้จากกล้อง CCD ก่อนทำการ brightness

(ข) ภาพที่ได้หลังจากการทำ brightness

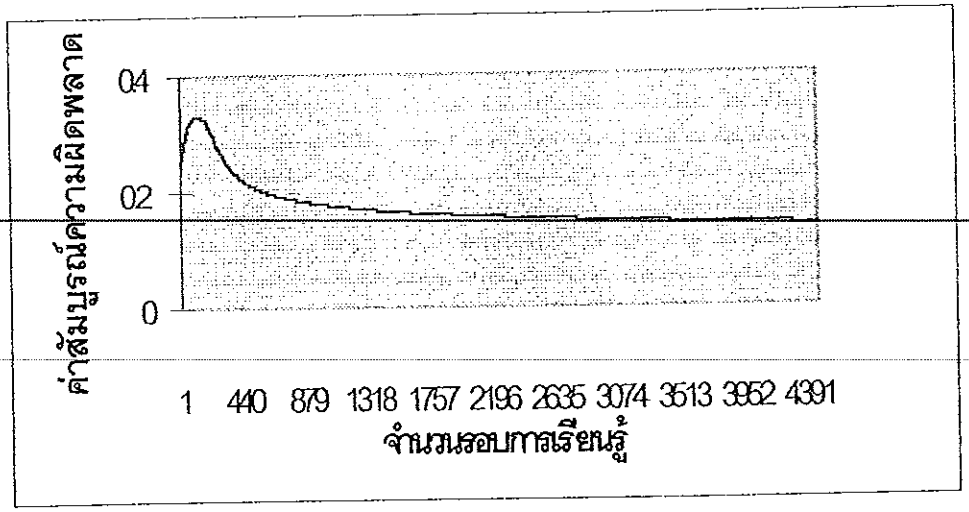
5.2 ผลการใช้นิวรอลเน็ตเวอร์ก

5.2.1 ผลจากการใช้นิวรอลเน็ตเวอร์กในส่วนของการเรียนรู้

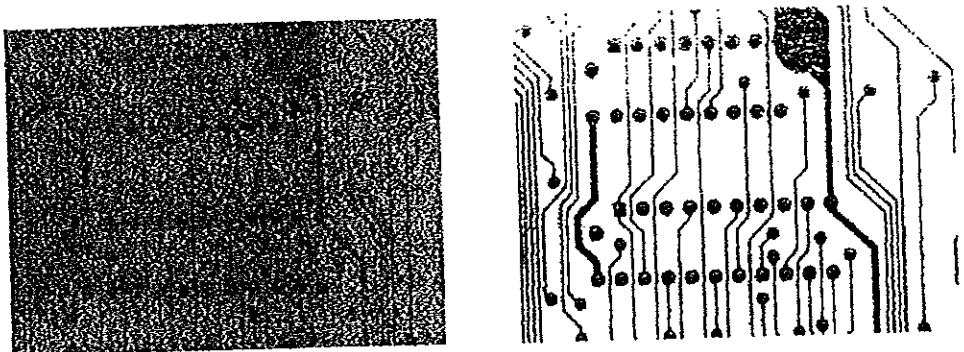
ในการป้อนข้อมูลที่สำคัญบางอย่างเพื่อให้นิวรอลเน็ตเวอร์กนำไปใช้ในการประมวลผลนั้นจะต้องประกอบด้วย จำนวนของชั้นซ่อน จำนวนเซลล์ในแต่ละชั้นซ่อน ค่าความผิดพลาดที่ต้องการ และอัตราการเรียนรู้ ในการทดลองใช้นิวรอลเน็ตเวอร์กแบบแพร่กลับเพื่อสังเกตพฤติกรรมบางอย่างของโครงข่าย พบว่าลักษณะการตอบสนองของนิวรอลเน็ตเวอร์กที่เป็นค่าสัมบูรณ์ความผิดพลาดจะเป็นดังภาพประกอบ 5-5 โดยให้อินพุตเป็นภาพลายวงจรพิมพ์ขนาด 10x10 จุด ใช้นิวรอลเน็ตเวอร์กแบบแพร่กลับ 1 ชั้นซ่อน มีจำนวนเซลล์ 15 เซลล์ ให้ค่าความผิดพลาดที่ 0.13

ตาราง 5-1 ถึง 5-3 เป็นการแสดงถึงจำนวนรอบที่นิวรอลเน็ตเวอร์กแบบแพร่กลับใช้ในการเรียนรู้ เพื่อสังเกตพฤติกรรมของนิวรอลเน็ตเวอร์กและนำผลที่ได้ไปใช้ในการทดลอง โดยที่ภาพลายวงจรพิมพ์ที่เป็นอินพุตให้กับนิวรอลเน็ตเวอร์กใช้ภาพลายวงจรพิมพ์ที่ยังไม่ได้เคลือบ ภาพลายวงจรพิมพ์ที่ใช้ในการเรียนรู้นั้นจะเป็นภาพที่ดีเพียง 1 ภาพ (ภาพขนาด 320x 240 นั่นคือมีจำนวน 76,800 พิกเซลต่อภาพ) โดยมีอินพุตเป็นค่าระดับเทาที่อยู่ในทุกพิกเซลของภาพลายวงจรพิมพ์จากนั้นจึงทำการทดลองกับนิวรอลเน็ตเวอร์กแบบแพร่กลับที่ทำขึ้นมา ส่วนภาพที่จะให้ผลเป็นภาพลายวงจรพิมพ์ที่เสียนั้นจะมี 2 ภาพ ซึ่งได้มาจากการสุ่มสัญญาณรบกวนลงในภาพลายวงจรพิมพ์ที่ดี หลังจากนั้นจึงนำไปให้นิวรอลเน็ตเวอร์กแบบแพร่กลับเรียนรู้ ซึ่งผลที่ได้จากการเรียนรู้นี้จะนำไปใช้ในการจำแนก ดังแสดงไว้ในหัวข้อ 5.2.2 โดยที่กำหนดค่าความผิดพลาดที่ 0.13 (โดยให้อัตราการเรียนรู้คงที่คือ 0.85) ส่วนจำนวนเซลล์ที่ใช้ในชั้นซ่อนที่ใช้ในการเรียนรู้ของนิวรอลเน็ตเวอร์กแบบแพร่กลับนี้ ในแบบ 1 ชั้นซ่อนจะใช้ที่ 100, 500, 1000, 5000, 10000, 50000, 70000 และ 90000 ส่วนในแบบ 2 ชั้นซ่อนจะใช้ที่ 100, 500, 1000, 5000, 10000, 50000, 70000 และ 90000

การใช้งานนิวรอลเน็ตเวอร์กแบบแพร่กลับ ได้ทำการทดลองใช้โครงข่ายตั้งแต่ 1 ชั้นซ่อน ไปจนถึง 2 ชั้นซ่อน โดยแบ่งเป็นลายวงจรพิมพ์ที่ยังไม่ได้ทำการเคลือบกับลายวงจรพิมพ์ในแบบที่ทำการเคลือบแล้ว



ภาพประกอบ 5-5 แสดงผลการลู่เข้าของค่าสัมบูรณ์ความผิดพลาด
จากการเรียนรู้ของนิวรอลเน็ตเวอร์ก 1 ชั้นซ่อน เมื่อกำหนดค่าความผิดพลาด = 0.13



(ก) (ข)

ภาพประกอบ 5-6 แสดงภาพลายวงจรพิมพ์ที่ใช้ในการทดสอบกับโครงข่าย

(ก) แสดงภาพที่ยังไม่ได้ผ่านกระบวนการการปรับปรุงภาพ

(ข) แสดงภาพที่ผ่านกระบวนการปรับปรุงภาพแล้ว

ลักษณะของภาพลายวงจรพิมพ์ที่ใช้ในการเรียนรู้และทดสอบจำแนก จะเป็นลายวงจรพิมพ์ที่มีอยู่ทั่วไป โดยที่เส้นลายวงจรพิมพ์มีทั้งแบบที่เป็นเส้นเล็กๆ หรือเส้นใหญ่ไปจนกระทั่งเป็นแบบพื้นที่ลายวงจรพิมพ์

ตาราง 5-1 แสดงจำนวนรอบการเรียนรู้ของนิวรอลเน็ตเวอร์ก 1 ชั้นซ่อน

จำนวนเซลล์	จำนวนรอบในการเรียนรู้
100	ไม่ลู่เข้า
500	ไม่ลู่เข้า
1,000	ไม่ลู่เข้า
5,000	1,800,000
10,000	530,000
50,000	360,000
70,000	200,000
90,000	230,000
100,000	300,000

ตาราง 5-2 แสดงจำนวนรอบการเรียนรู้ของนิวรอลเน็ตเวอร์ก 2 ชั้นซ่อน

จำนวนเซลล์ในชั้นซ่อนที่ 1	จำนวนเซลล์ในชั้นซ่อนที่ 2	จำนวนรอบในการเรียนรู้
100	100	ไม่ลู่เข้า
500	500	430,000
1,000	1,000	320,000
5,000	5,000	380,000
10,000	10,000	843,000
50,000	50,000	ไม่ลู่เข้า
70,000	70,000	ไม่ลู่เข้า
90,000	90,000	ไม่ลู่เข้า
100,000	100,000	ไม่ลู่เข้า

ตาราง 5-3 แสดงจำนวนรอบการเรียนรู้ของนิวรอลเน็ตเวอร์ก 3 ชั้นซ่อน

จำนวนเซลล์ในชั้นซ่อนที่ 1	จำนวนเซลล์ในชั้นซ่อนที่ 2	จำนวนเซลล์ในชั้นซ่อนที่ 3	จำนวนรอบในการเรียนรู้
100	100	100	ไม่ลู่เข้า
500	500	500	740,000
1,000	1,000	1,000	1,100,000
5,000	5,000	5,000	1,830,000
10,000	10,000	10,000	ไม่ลู่เข้า
50,000	50,000	50,000	ไม่ลู่เข้า
70,000	70,000	70,000	ไม่ลู่เข้า
90,000	90,000	90,000	ไม่ลู่เข้า
100,000	100,000	100,000	ไม่ลู่เข้า

จากการเรียนรู้ของนิวรอลเน็ตเวอร์กแบบแพร่กลับจะพบว่า

1) ที่นิวรอลเน็ตเวอร์ก 1 ชั้นซ่อน

- นิวรอลเน็ตเวอร์กแบบ 1 ชั้นซ่อน เมื่อใช้จำนวนเซลล์น้อยกว่า 5,000 เซลล์ จะทำให้โครงข่ายไม่สามารถลู่เข้าหาค่าความผิดพลาดคงที่ที่กำหนดไว้หรือเรียกว่าไม่ลู่เข้า

- จะมีช่วงของจำนวนเซลล์ในชั้นซ่อนค่าหนึ่งที่จะทำให้จำนวนรอบของการลู่เข้าของค่าความผิดพลาดมีค่าน้อยที่สุดช่วงหนึ่ง

2) ที่นิวรอลเน็ตเวอร์ก 2 ชั้นซ่อน

- เมื่อจำนวนเซลล์ในโครงข่ายแบบ 2 ชั้นซ่อน มีจำนวนน้อยหรือมากเกินไป นิวรอลเน็ตเวอร์กก็จะไม่สามารถปรับตัวเพื่อให้ค่าความผิดพลาดลู่เข้าได้

- จำนวนรอบที่ใช้ในการลู่เข้าจะสูงกว่าโครงข่ายแบบ 1 ชั้นซ่อน

3) ที่นิวรอลเน็ตเวอร์ก 3 ชั้นซ่อน

- โครงข่ายแบบนี้ มีช่วงของเซลล์ในชั้นซ่อนที่สามารถทำให้ค่าความผิดพลาดลู่เข้าได้เพียงช่วง 500 – 10,000 จุด

- จำนวนรอบที่ใช้ในการลู่เข้าจะสูงกว่าโครงข่ายแบบ 1 และ 2 ชั้นซ่อน

5.2.2 ผลจากการใช้นิวรอลเน็ตเวอร์กในส่วนของการทดสอบจำแนก

ผลที่ได้จากการใช้งานนิวรอลเน็ตเวอร์กหลังจากที่ได้ทำการเรียนรู้แล้วนั้น จะเป็นการป้อนภาพลายวงจรพิมพ์ที่ไม่ได้ให้นิวรอลเน็ตเวอร์กใช้ในการเรียนรู้ จำนวน 30 ภาพ โดยที่ทั้งหมดจะมีทั้งภาพลายวงจรที่ให้ผลการจำแนกดีและบกพร่อง เพื่อที่จะทำการสังเกตผลจากการทำงานของโครงข่าย จึงได้ทำการทดสอบการจำแนกดังนี้

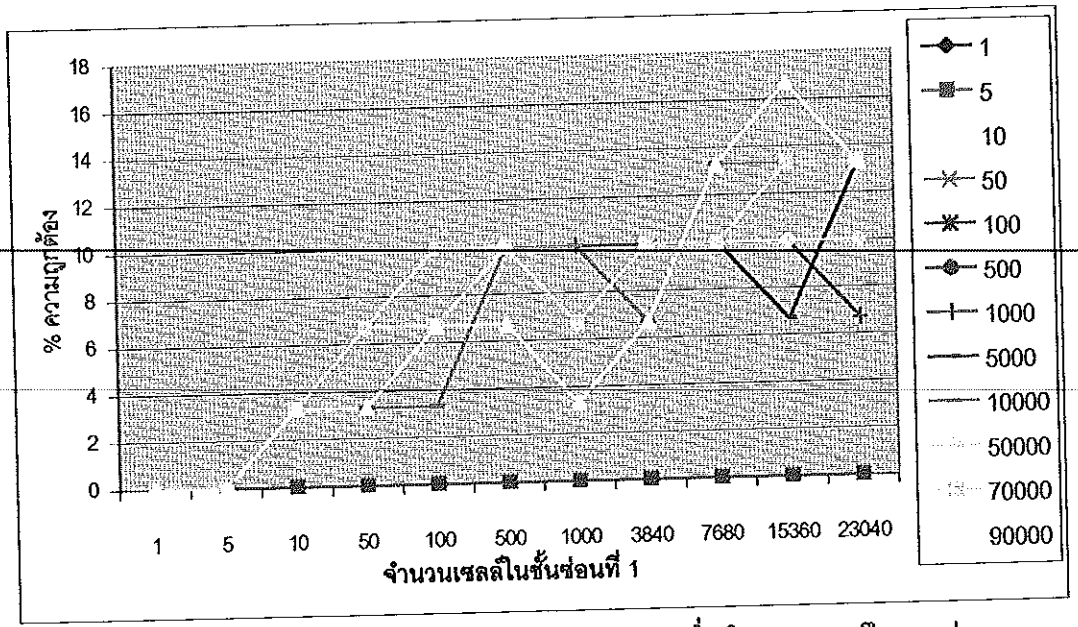
- ที่ 1 ชั้นซ่อน มีกลุ่มของลายวงจรพิมพ์ 2 แบบคือแบบที่ทำการเคลือบลายวงจรแล้วกับแบบที่ยังไม่ได้ทำการเคลือบลายวงจรพิมพ์ ใช้จำนวนจุดผิดพลาดในการทดลองให้นิวรอลเน็ตเวอร์กทำการจำแนกที่ 1, 5, 10, 50, 100, 500, 1000, 3840, 7680, 15360 และ 23040 จุด จำนวนเซลล์ในชั้นซ่อนคือ 1, 5, 10, 50, 100, 500, 1000, 5000, 10000, 50000, 70000 และ 90000

- ที่ 2 ชั้นซ่อน ได้ทำการทดสอบเฉพาะลายวงจรพิมพ์ที่ยังไม่ได้ทำการเคลือบใช้จำนวนจุดผิดพลาดในการทดลองให้นิวรอลเน็ตเวอร์กทำการจำแนกที่ 100, 500, 1000, 3840, 7680, 15360 และ 23040 จุด จำนวนเซลล์ในชั้นซ่อนคือ 100, 500, 1000, 5000, 10000, 50000, 70000 และ 90000

มีผลการทดลองดังแสดงในตารางที่ 5-4 ถึง 5-28

ตาราง 5-4 แสดงความถูกต้องในการจำแนกในแบบ 1 ชั้นซ่อน
เมื่อใช้จำนวนจุดผิดพลาดในการเรียนรู้คือ 1 พิกเซล
กับลายวงจรพิมพ์ที่ยังไม่ได้ทำการเคลือบ

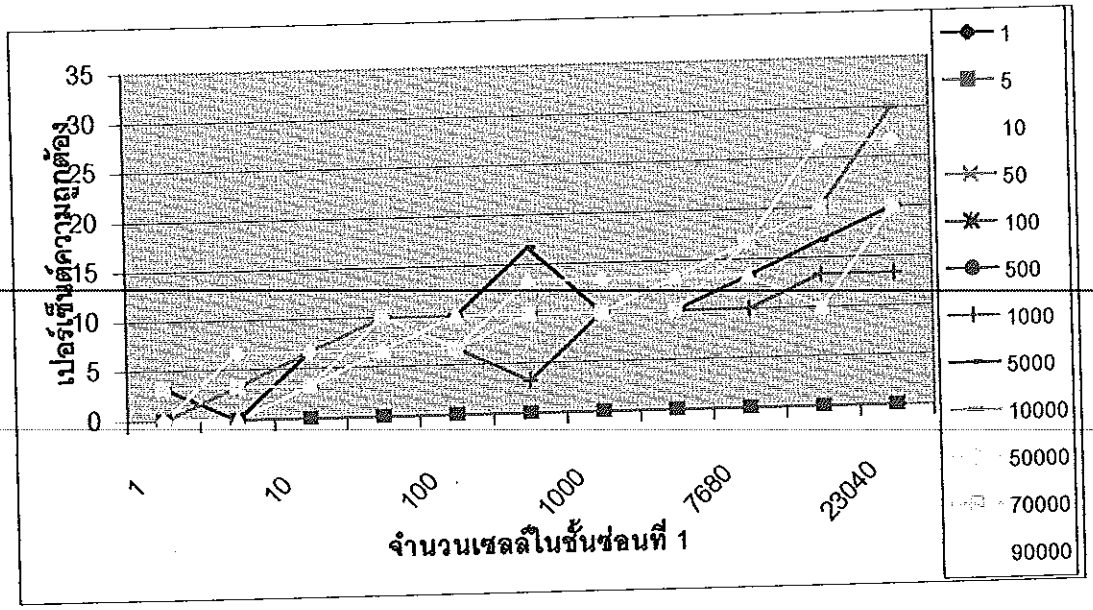
จำนวนเซลล์ชั้นที่ 1	จำนวนจุดที่ผิดพลาดที่ใช้ในการทดสอบ										
	1	5	10	50	100	500	1000	3840	7680	15360	23040
1	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	0
50	0	0	0	0	0	0	0	0	0	0	0
100	0	0	0	0	0	0	0	0	0	0	0
500	0	0	0	0	0	0	0	0	0	0	0
1000	0	0	3.3	6.7	10	10	10	10	10	10	6.7
5000	0	0	3.3	6.7	6.7	10	10	10	10	6.7	13.3
10000	0	0	3.3	3.3	3.3	10	10	6.7	13.3	13.3	13.3
50000	0	0	3.3	6.7	10	10	6.7	10	10	10	10
70000	0	0	3.3	6.7	6.7	10	6.7	10	10	13.3	13.3
90000	0	0	3.3	3.3	6.7	6.7	3.3	6.7	13.3	16.7	13.3



ภาพประกอบ 5-7 แสดงข้อมูลจากตาราง 5-4 เมื่อนำมาแสดงเป็นกราฟ

ตาราง 5-5 แสดงความถูกต้องในการจำแนกในแบบ 1 ชั้นซ่อน
เมื่อใช้จำนวนจุดผิดพลาดในการเรียนรู้ 5 พิกเซล
กับลายวงจรพิมพ์ที่ยังไม่ได้ทำการเคลือบ

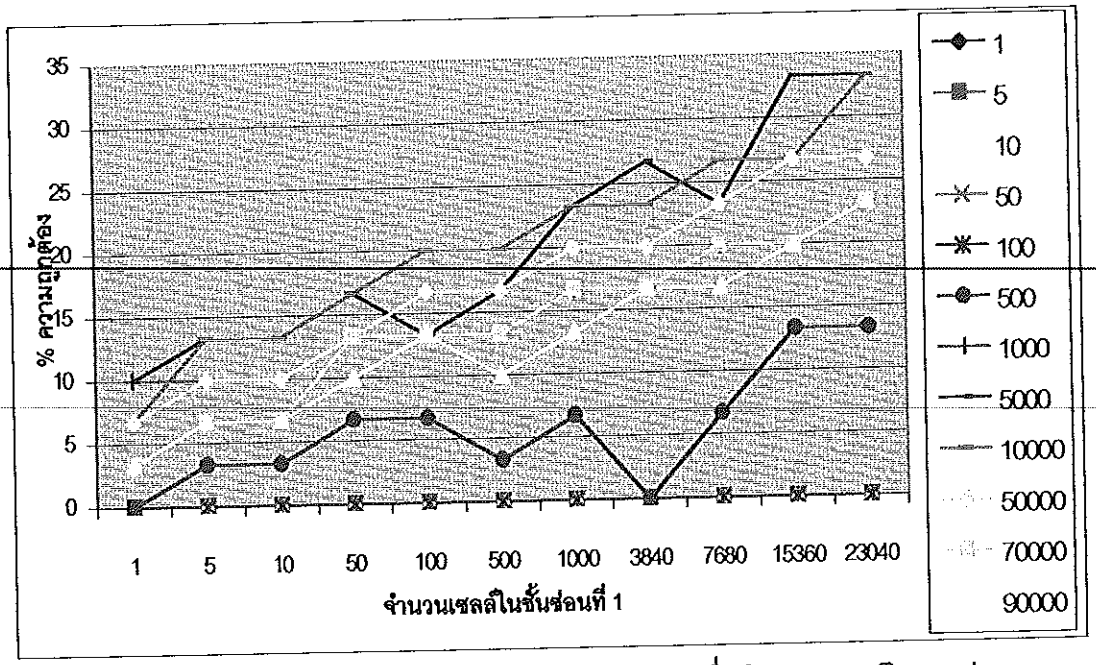
จำนวนเซลล์ชั้นที่ 1	จำนวนจุดที่ผิดพลาดที่ใช้ในการทดสอบ										
	1	5	10	50	100	500	1000	3840	7680	15360	23040
1	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	0
50	0	0	0	0	0	0	0	0	0	0	0
100	0	0	0	0	0	0	0	0	0	0	0
500	0	0	0	0	0	0	0	0	0	0	0
1000	3.3	3.3	3.3	6.7	6.7	3.3	10	10	10	13.3	13.3
5000	3.3	0	6.7	6.7	10	16.7	10	10	13.3	16.7	20
10000	0	3.3	6.7	10	10	10	10	13.3	16.7	20	30
50000	0	6.7	6.7	6.7	6.7	13.3	13.3	13.3	16.7	26.7	26.7
70000	3.3	3.3	3.3	10	6.7	10	10	10	16.7	20	20
90000	0	0	3.3	6.7	10	10	10	13.3	13.3	10	20



ภาพประกอบ 5-8 แสดงข้อมูลจากตาราง 5-5 เมื่อนำมาแสดงเป็นกราฟ

ตาราง 5-6 แสดงความถูกต้องในการจำแนกในแบบ 1 ชั้นซ่อน
เมื่อใช้จำนวนจุดผิดพลาดในการเรียนรู้ 10 พิกเซล
กับลายวงจรพิมพ์ที่ยังไม่ได้ทำการเคลือบ

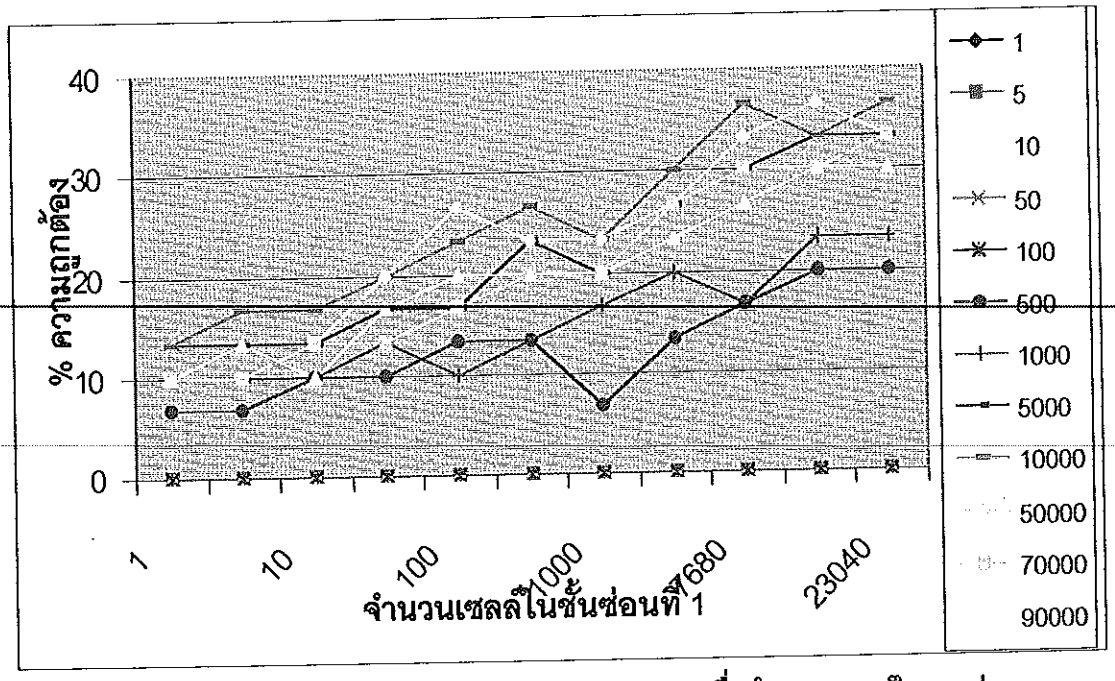
จำนวนเซลล์ชั้นที่ 1	จำนวนจุดที่ผิดพลาดที่ใช้ในการทดสอบ										
	1	5	10	50	100	500	1000	3840	7680	15360	23040
1	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	0
50	0	0	0	0	0	0	0	0	0	0	0
100	0	0	0	0	0	0	0	0	0	0	0
500	0	3.3	3.3	6.7	6.7	3.3	6.7	0	6.7	13.3	13.3
1000	10	10	10	13.3	13.3	13.3	16.7	16.7	20	20	23.3
5000	10	13.3	13.3	16.7	13.3	16.7	23.3	26.7	23.3	33.3	33.3
10000	6.7	13.3	13.3	16.7	20	20	23.3	23.3	26.7	26.7	33.3
50000	6.7	10	10	13.3	16.7	16.7	20	20	23.3	26.7	26.7
70000	3.3	6.7	6.7	13.3	13.3	13.3	16.7	16.7	20	20	23.3
90000	3.3	6.7	6.7	10	13.3	10	13.3	16.7	16.7	20	23.3



ภาพประกอบ 5-9 แสดงข้อมูลจากตาราง 5-6 เมื่อนำมาแสดงเป็นกราฟ

ตาราง 5-7 แสดงความถูกต้องในการจำแนกในแบบ 1 ชั้นซ่อน
เมื่อใช้จำนวนจุดผิดพลาดในการเรียนรู้ 50 พิกเซล
กับลายวงจรพิมพ์ที่ยังไม่ได้ทำการเคลือบ

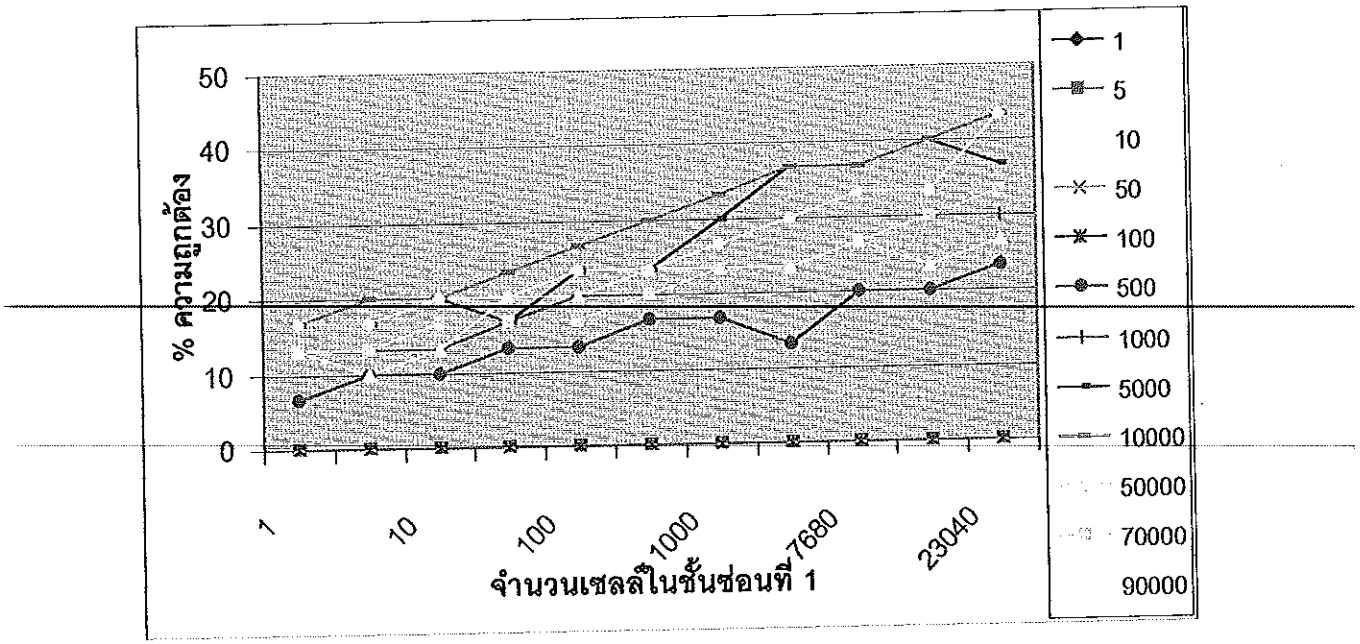
จำนวนเซลล์ชั้นที่ 1	จำนวนจุดที่ผิดพลาดที่ใช้ในการทดสอบ										
	1	5	10	50	100	500	1000	3840	7680	15360	23040
1	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	0
50	0	0	0	0	0	0	0	0	0	0	0
100	0	0	0	0	0	0	0	0	0	0	0
500	6.7	10	10	13.3	13.3	16.7	16.7	13.3	20	20	23.3
1000	13.3	13.3	13.3	16.7	20	20	23.3	23.3	26.7	30	30
5000	16.7	16.7	20	16.7	23.3	23.3	30	36.7	36.7	40	36.7
10000	16.7	20	20	23.3	26.7	30	33.3	36.7	36.7	40	43.3
50000	16.7	16.7	20	20	23.3	23.3	26.7	30	33.3	33.3	43.3
70000	13.3	13.3	16.7	16.7	16.7	20	23.3	23.3	26.7	30	33.3
90000	13.3	10	13.3	20	20	23.3	23.3	23.3	26.7	23.3	26.7



ภาพประกอบ 5-10 แสดงข้อมูลจากตาราง 5-7 เมื่อนำมาแสดงเป็นกราฟ

ตาราง 5-8 แสดงความถูกต้องในการจำแนกในแบบ 1 ชั้นซ่อน
เมื่อใช้จำนวนจุดผิดพลาดในการเรียนรู้ 100 พิกเซล
กับลายวงจรพิมพ์ที่ยังไม่ได้ทำการเคลือบ

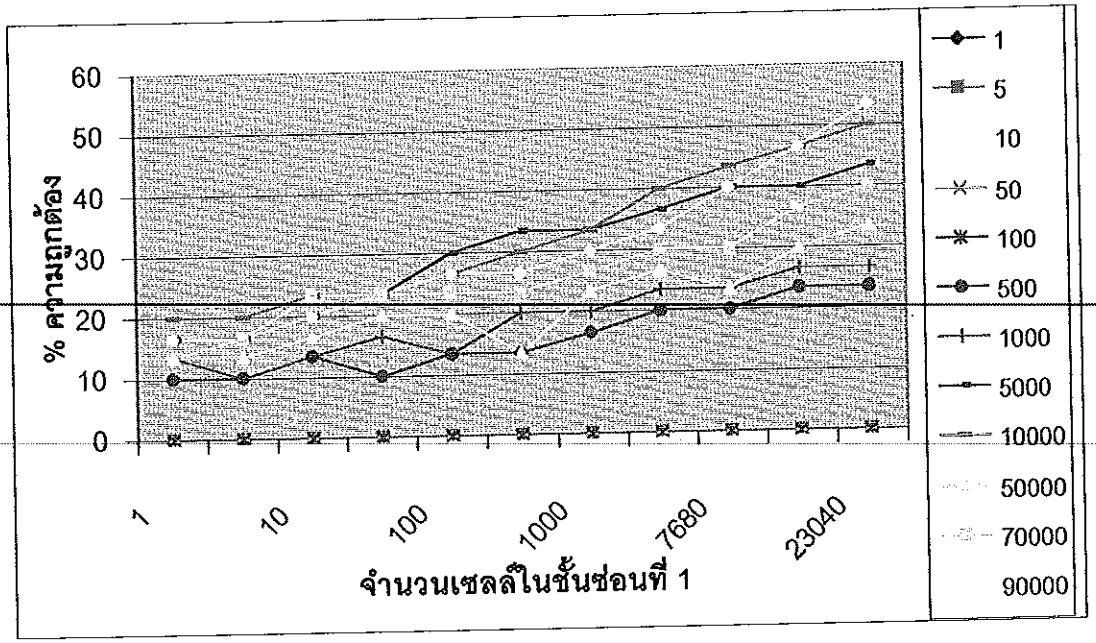
จำนวนเซลล์ชั้นที่ 1	จำนวนจุดที่ผิดพลาดที่ใช้ในการทดสอบ										
	1	5	10	50	100	500	1000	3840	7680	15360	23040
1	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	0
50	0	0	0	0	0	0	0	0	0	0	0
100	0	0	0	0	0	0	0	0	0	0	0
500	6.7	6.7	10	10	13.3	13.3	6.7	13.3	16.7	20	20
1000	10	10	10	13.3	10	13.3	16.7	20	16.7	23.3	23.3
5000	13.3	13.3	13.3	16.7	16.7	23.3	20	26.7	30	33.3	33.3
10000	13.3	16.7	16.7	20	23.3	26.7	23.3	30	36.7	33.3	36.7
50000	10	10	13.3	20	26.7	23.3	23.3	26.7	33.3	36.7	33.3
70000	10	10	13.3	13.3	16.7	20	20	26.7	30	30	33.3
90000	10	13.3	10	16.7	20	20	20	23.3	26.7	30	30



ภาพประกอบ 5-11 แสดงข้อมูลจากตาราง 5-8 เมื่อนำมาแสดงเป็นกราฟ

ตาราง 5-9 แสดงความถูกต้องในการจำแนกในแบบ 1 ชั้นซ่อน
เมื่อใช้จำนวนจุดผิดพลาดในการเรียนรู้ 500 พิกเซล
กับลายวงจรพิมพ์ที่ยังไม่ได้ทำการเคลือบ

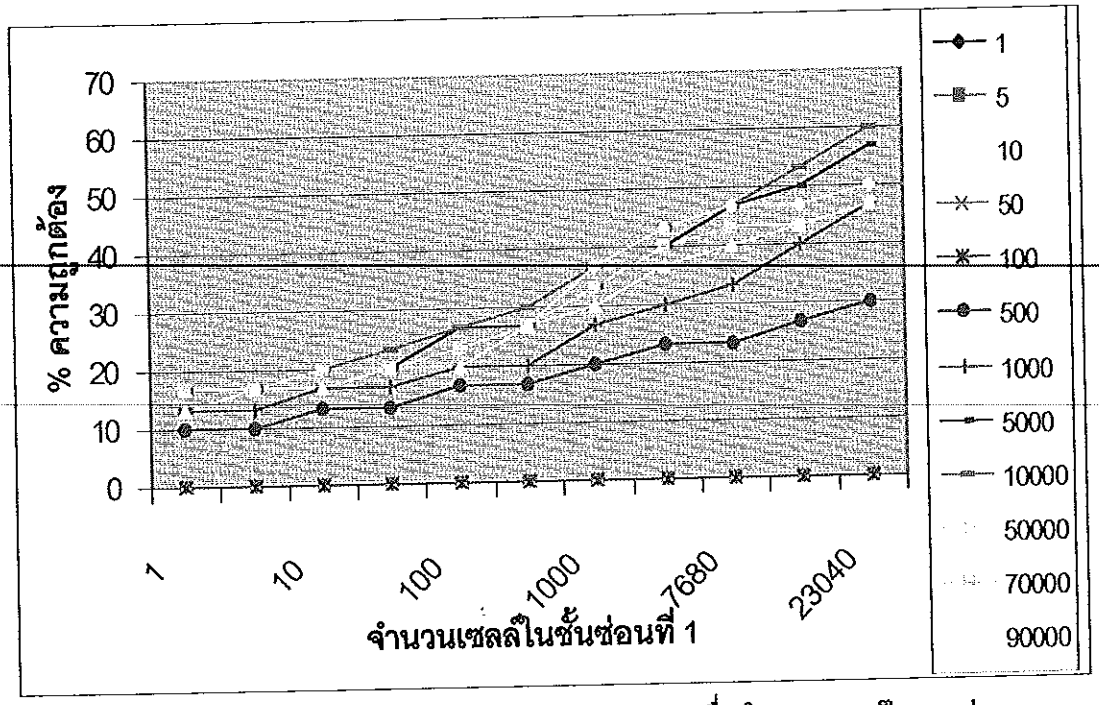
จำนวนเซลล์ชั้นที่ 1	จำนวนจุดที่ผิดพลาดที่ใช้ในการทดสอบ										
	1	5	10	50	100	500	1000	3840	7680	15360	23040
1	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	0
50	0	0	0	0	0	0	0	0	0	0	0
100	0	0	0	0	0	0	0	0	0	0	0
500	10	10	13.3	10	13.3	13.3	16.7	20	20	23.3	23.3
1000	13.3	10	13.3	16.7	13.3	20	20	23.3	23.3	26.7	26.7
5000	16.7	16.7	20	23.3	30	33.3	33.3	36.7	40	40	43.3
10000	20	20	23.3	23.3	26.7	30	33.3	40	43.3	46.7	50
50000	16.7	16.7	20	23.3	26.7	26.7	30	33.3	40	46.7	53.3
70000	13.3	16.7	23.3	23.3	23.3	23.3	26.7	30	30	36.7	40
90000	13.3	13.3	16.7	20	20	13.3	23.3	26.7	23.3	30	33.3



ภาพประกอบ 5-12 แสดงข้อมูลจากตาราง 5-9 เมื่อนำมาแสดงเป็นกราฟ

ตาราง 5-10 แสดงความถูกต้องในการจำแนกในแบบ 1 ชั้นซ่อน
เมื่อใช้จำนวนจุดผิดพลาดในการเรียนรู้ 1000 พิกเซล
กับลายวงจรพิมพ์ที่ยังไม่ได้ทำการเคลือบ

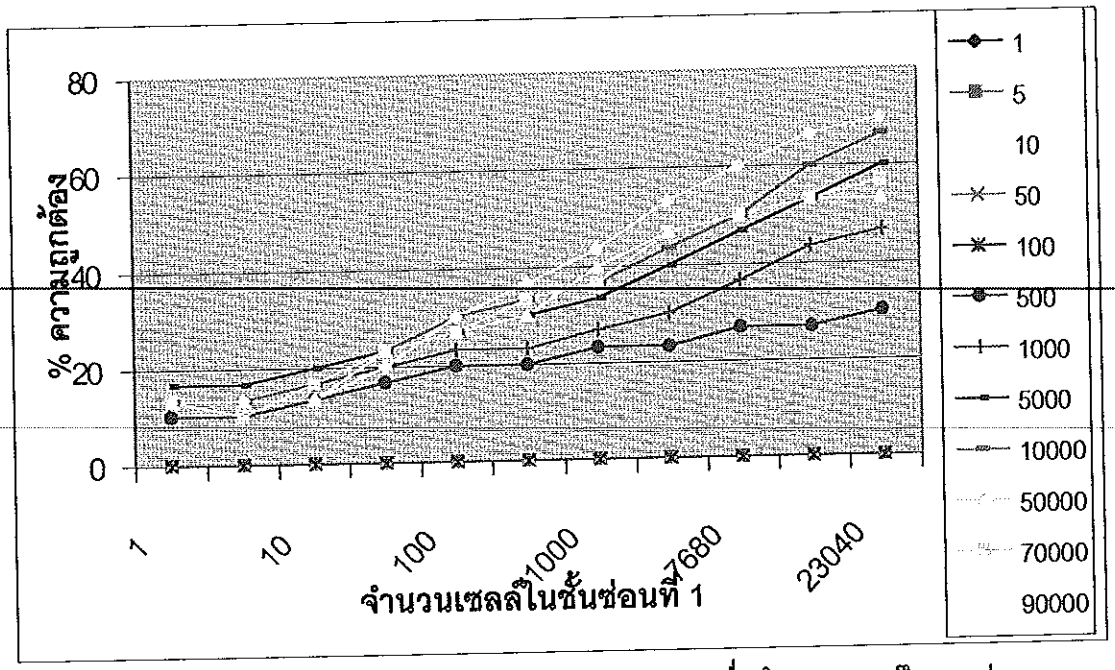
จำนวนเซลล์ชั้นที่ 1	จำนวนจุดที่ผิดพลาดที่ใช้ในการทดสอบ										
	1	5	10	50	100	500	1000	3840	7680	15360	23040
1	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	0
50	0	0	0	0	0	0	0	0	0	0	0
100	0	0	0	0	0	0	0	0	0	0	0
500	10	10	13.3	13.3	16.7	16.7	20	23.3	23.3	26.7	30
1000	13.3	13.3	16.7	16.7	20	20	26.7	30	33.3	40	46.7
5000	16.7	16.7	20	20	26.7	26.7	33.3	40	46.7	50	56.7
10000	13.3	16.7	20	23.3	26.7	30	36.7	43.3	46.7	53.3	60
50000	16.7	16.7	20	20	23.3	26.7	36.7	43.3	46.7	46.7	50
70000	16.7	16.7	20	20	23.3	26.7	33.3	40	43.3	43.3	46.7
90000	13.3	16.7	16.7	20	20	26.7	30	36.7	40	43.3	46.7



ภาพประกอบ 5-13 แสดงข้อมูลจากตาราง 5-10 เมื่อนำมาแสดงเป็นกราฟ

ตาราง 5-11 แสดงความถูกต้องในการจำแนกในแบบ 1 ชั้นซ่อน
เมื่อใช้จำนวนจุดผิดพลาดในการเรียนรู้ 5000 พิกเซล
กับลายวงจรพิมพ์ที่ยังไม่ได้ทำการเคลือบ

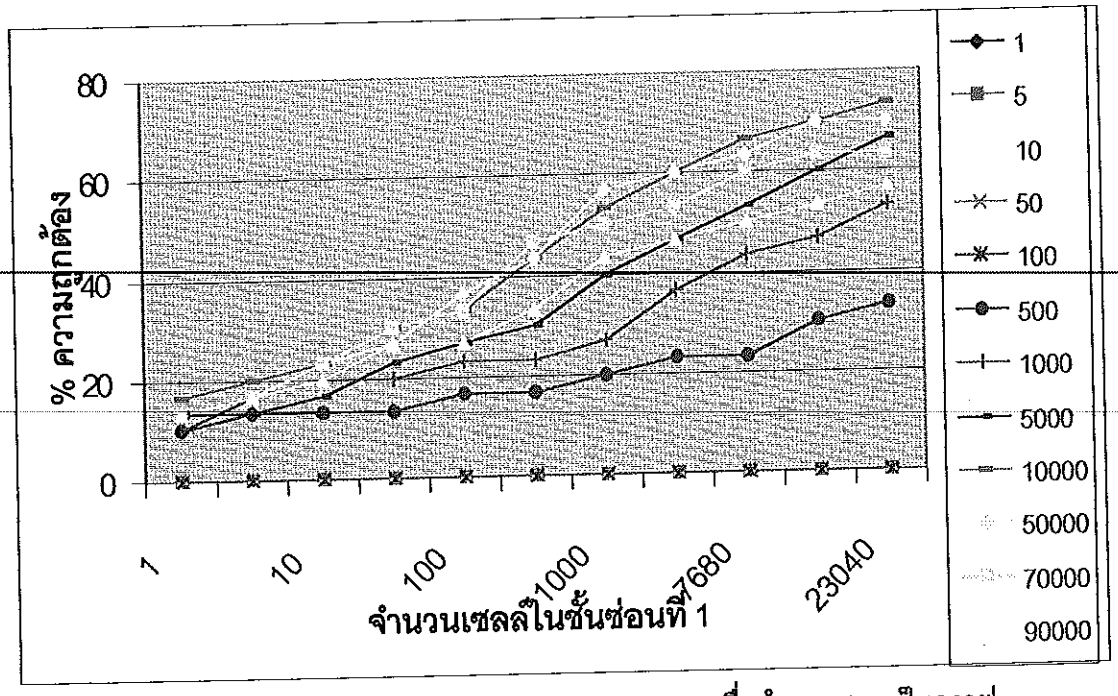
จำนวนเซลล์ชั้นที่ 1	จำนวนจุดที่ผิดพลาดที่ใช้ในการทดสอบ										
	1	5	10	50	100	500	1000	3840	7680	15360	23040
1	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	0
50	0	0	0	0	0	0	0	0	0	0	0
100	0	0	0	0	0	0	0	0	0	0	0
500	10	10	13.3	16.7	20	20	23.3	23.3	26.7	26.7	30
1000	13.3	13.3	16.7	20	23.3	23.3	26.7	30	36.7	43.3	46.7
5000	16.7	16.7	20	23.3	26.7	30	33.3	40	46.7	53.3	60
10000	13.3	13.3	16.7	23.3	30	33.3	36.7	43.3	50	60	66.7
50000	13.3	13.3	13.3	20	30	36.7	43.3	53.3	60	66.7	70
70000	13.3	10	16.7	23.3	26.7	33.3	36.7	46.7	50	53.3	56.7
90000	13.3	13.3	13.3	23.3	26.7	30	40	46.7	50	53.3	53.3



ภาพประกอบ 5-14 แสดงข้อมูลจากตาราง 5-11 เมื่อนำมาแสดงเป็นกราฟ

ตาราง 5-12 แสดงความถูกต้องในการจำแนกในแบบ 1 ชั้นซ่อน
เมื่อใช้จำนวนจุดผิดพลาดในการเรียนรู้ 10000 พิกเซล
กับลายวงจรพิมพ์ที่ยังไม่ได้ทำการเคลือบ

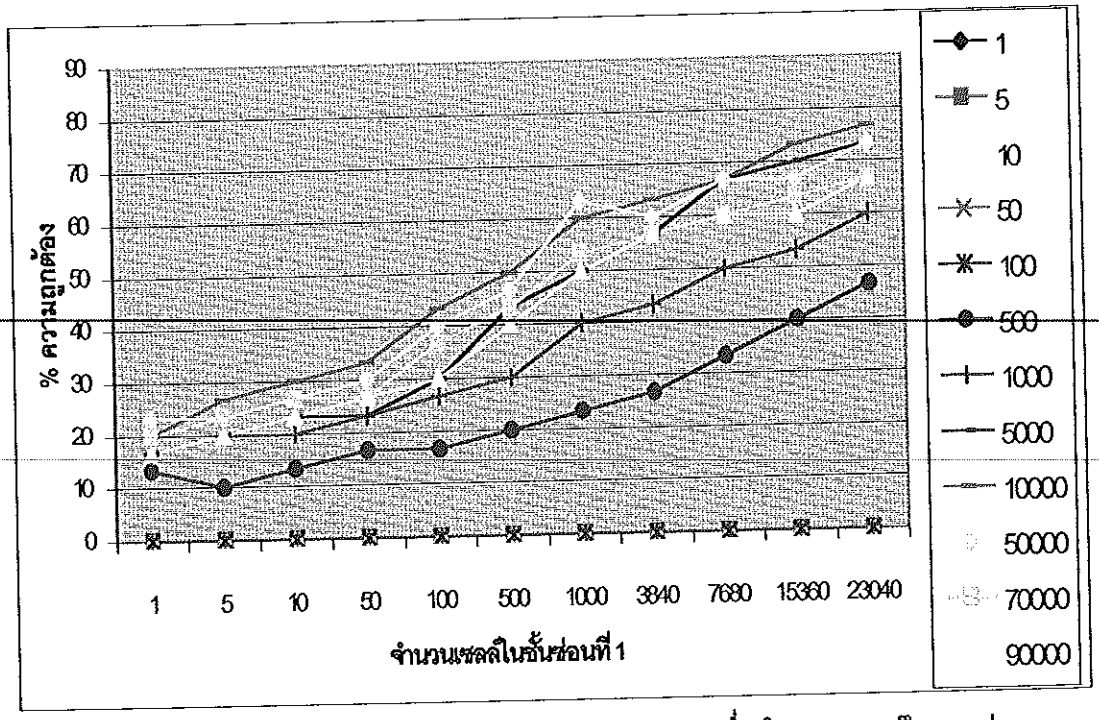
จำนวนเซลล์ชั้นที่ 1	จำนวนจุดที่ผิดพลาดที่ใช้ในการทดสอบ										
	1	5	10	50	100	500	1000	3840	7680	15360	23040
1	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	0
50	0	0	0	0	0	0	0	0	0	0	0
100	0	0	0	0	0	0	0	0	0	0	0
500	10	13.3	13.3	13.3	16.7	16.7	20	23.3	23.3	30	33.3
1000	10	16.7	20	20	23.3	23.3	26.7	36.7	43.3	46.7	53.3
5000	13.3	13.3	16.7	23.3	26.7	30	40	46.7	53.3	60	66.7
10000	16.7	20	23.3	30	33.3	43.3	53.3	60	66.7	70	73.3
50000	13.3	16.7	23.3	26.7	36.7	43.3	56.7	60	63.3	70	70
70000	13.3	16.7	23.3	30	33.3	46.7	50	53.3	60	63.3	63.3
90000	13.3	16.7	20	26.7	26.7	33.3	43.3	46.7	50	53.3	56.7



ภาพประกอบ 5-15 แสดงข้อมูลจากตาราง 5-12 เมื่อนำมาแสดงเป็นกราฟ

ตาราง 5-13 แสดงความถูกต้องในการจำแนกในแบบ 1 ชั้นซ่อน
เมื่อใช้จำนวนจุดผิดพลาดในการเรียนรู้ 50000 พิกเซล
กับลายวงจรพิมพ์ที่ยังไม่ได้ทำการเคลือบ

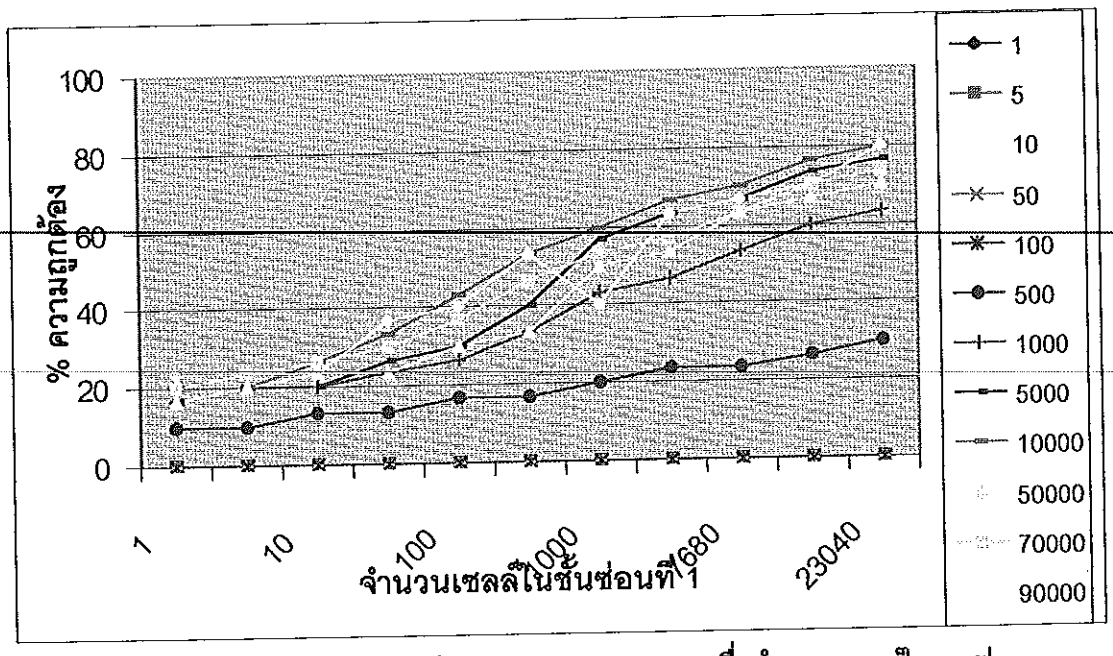
จำนวนเซลล์ชั้นที่ 1	จำนวนจุดที่ผิดพลาดที่ใช้ในการทดสอบ										
	1	5	10	50	100	500	1000	3840	7680	15360	23040
1	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	0
50	0	0	0	0	0	0	0	0	0	0	0
100	0	0	0	0	0	0	0	0	0	0	0
500	13.3	10	13.3	16.7	16.7	20	23.3	26.7	33.3	40	46.7
1000	16.7	20	20	23.3	26.7	30	40	43.3	50	53.3	60
5000	16.7	20	23.3	23.3	30	43.3	50	56.7	66.7	70	73.3
10000	20	26.7	30	33.3	43.3	50	60	63.3	66.7	73.3	76.7
50000	23.3	23.3	26.7	30	40	46.7	63.3	60	66.7	66.7	73.3
70000	20	23.3	26.7	26.7	36.7	43.3	53.3	56.7	60	63.3	66.7
90000	16.7	20	23.3	26.7	30	40	50	56.7	60	60	66.7



ภาพประกอบ 5-16 แสดงข้อมูลจากตาราง 5-13 เมื่อนำมาแสดงเป็นกราฟ

ตาราง 5-14 แสดงความถูกต้องในการจำแนกในแบบ 1 ชั้นซ่อน
เมื่อใช้จำนวนจุดผิดพลาดในการเรียนรู้ 70000 พิกเซล
กับลายวงจรพิมพ์ที่ยังไม่ได้ทำการเคลือบ

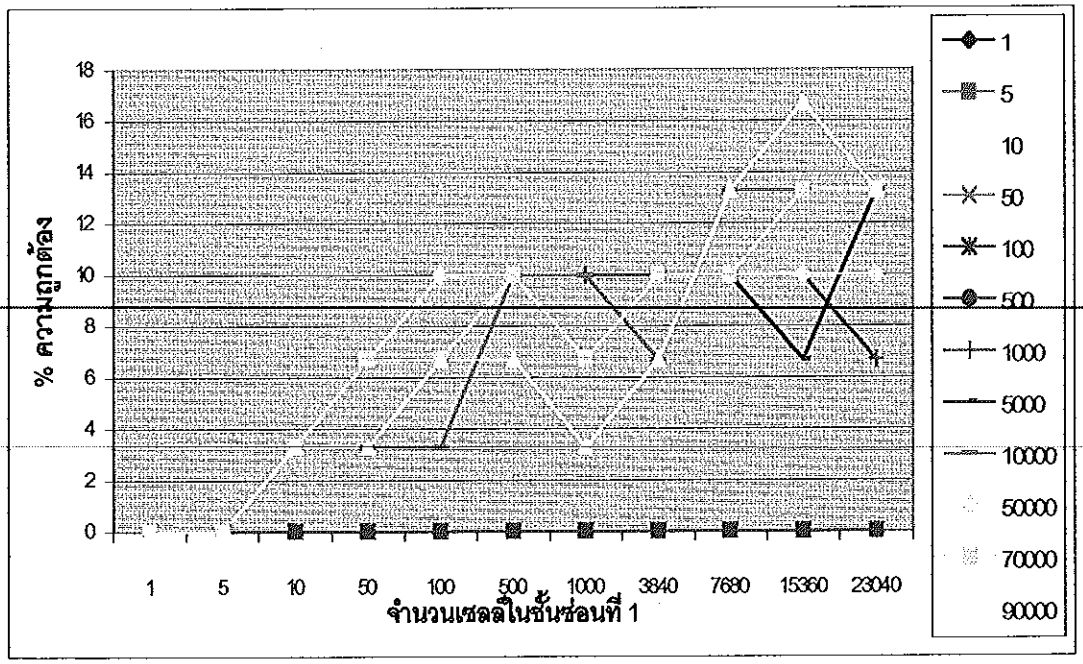
จำนวนเซลล์ชั้นที่ 1	จำนวนจุดที่ผิดพลาดที่ใช้ในการทดสอบ										
	1	5	10	50	100	500	1000	3840	7680	15360	23040
	1	5	10	50	100	500	1000	3840	7680	15360	23040
1	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	0
50	0	0	0	0	0	0	0	0	0	0	0
100	0	0	0	0	0	0	0	0	0	0	0
500	10	10	13.3	13.3	16.7	16.7	20	23.3	23.3	26.7	30
1000	16.7	20	20	23.3	26.7	33.3	43.3	46.7	53.3	60	63.3
5000	16.7	20	20	26.7	30	40	56.7	63.3	66.7	73.3	76.7
10000	20	20	26.7	33.3	43.3	53.3	60	66.7	70	76.7	80
50000	23.3	23.3	23.3	36.7	40	53.3	40	63.3	66.7	70	80
70000	20	23.3	26.7	30	36.7	46.7	53.3	56.7	63.3	70	73.3
90000	16.7	20	23.3	23.3	30	33.3	50	53.3	63.3	66.7	70



ภาพประกอบ 5-17 แสดงข้อมูลจากตาราง 5-14 เมื่อนำมาแสดงเป็นกราฟ

ตาราง 5-15 แสดงความถูกต้องในการจำแนกในแบบ 1 ชั้นซ่อน
เมื่อใช้จำนวนจุดผิดพลาดในการเรียนรู้1 พิกเซล
กับลายวงจรพิมพ์ที่ทำการเคลือบแล้ว

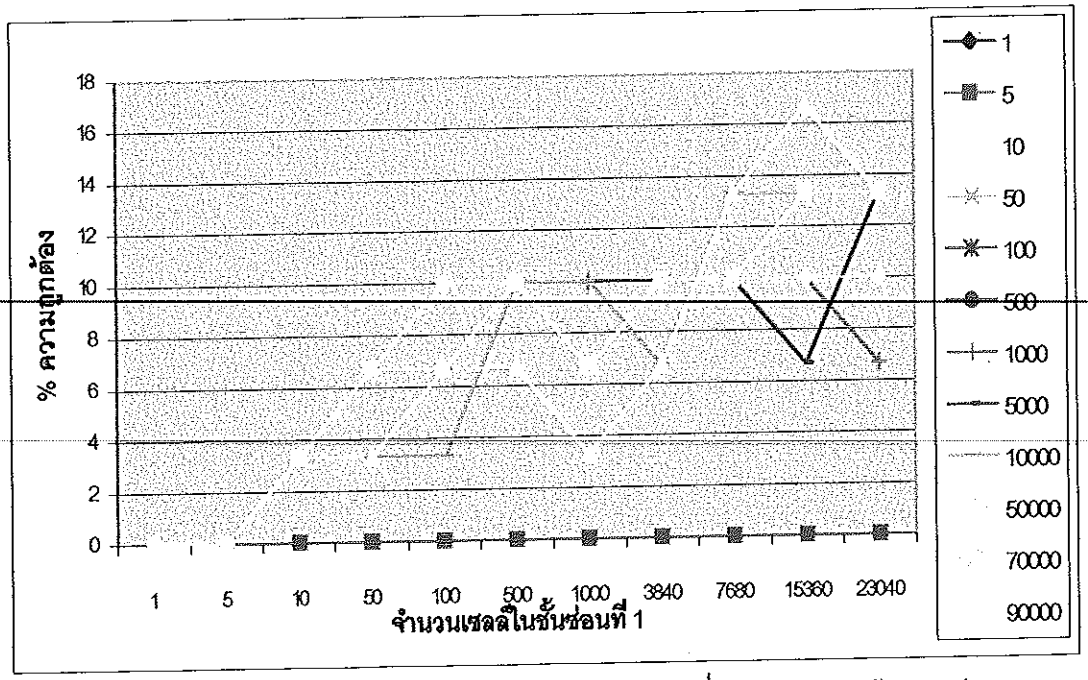
จำนวนเซลล์ชั้นที่ 1	จำนวนจุดที่ผิดพลาดที่ใช้ในการทดสอบ										
	1	5	10	50	100	500	1000	3840	7680	15360	23040
1	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	0
50	0	0	0	0	0	0	0	0	0	0	0
100	0	0	0	0	0	0	0	0	0	0	0
500	0	0	0	0	0	0	0	0	0	0	0
1000	0	0	3.3	6.7	10	10	10	10	10	10	6.7
5000	0	0	3.3	6.7	6.7	10	10	10	10	6.7	13.3
10000	0	0	3.3	3.3	3.3	10	10	6.7	13.3	13.3	13.3
50000	0	0	3.3	6.7	10	10	6.7	10	10	10	10
70000	0	0	3.3	6.7	6.7	10	6.7	10	10	13.3	13.3
90000	0	0	3.3	3.3	6.7	6.7	3.3	6.7	13.3	16.7	13.3



ภาพประกอบ 5-18 แสดงข้อมูลจากตาราง 5-15 เมื่อนำมาแสดงเป็นกราฟ

ตาราง 5-15 แสดงความถูกต้องในการจำแนกในแบบ 1 ชั้นซ่อน
เมื่อใช้จำนวนจุดผิดพลาดในการเรียนรู้ 1 พิกเซล
กับลายวงจรพิมพ์ที่ทำการเคลือบแล้ว

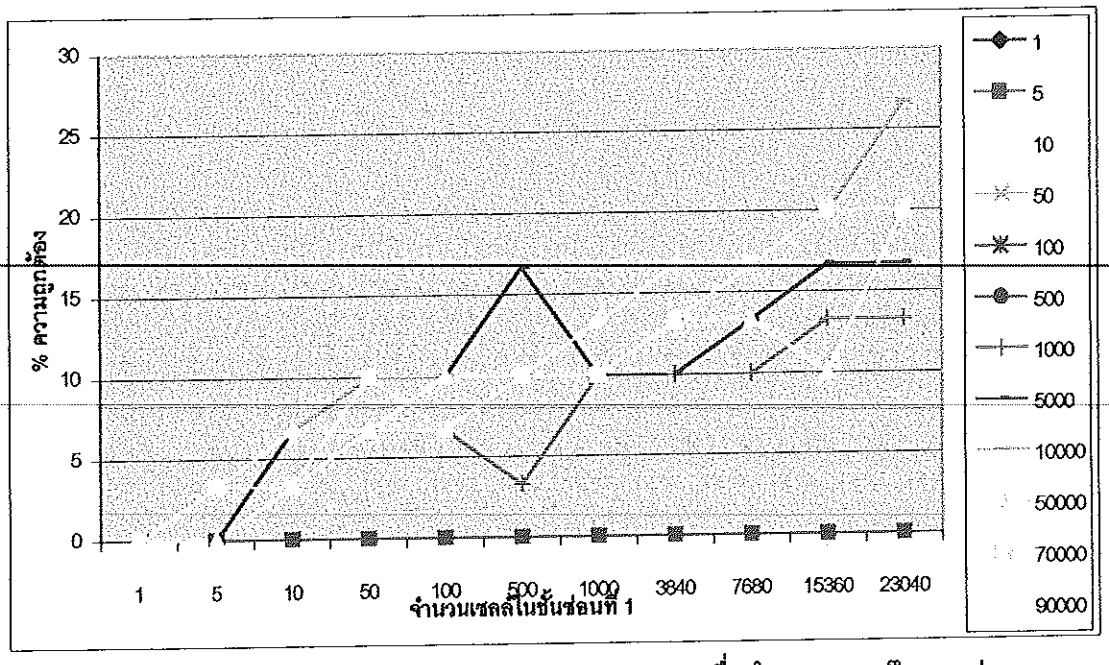
จำนวนเซลล์ชั้นที่ 1	จำนวนจุดที่ผิดพลาดที่ใช้ในการทดสอบ										
	1	5	10	50	100	500	1000	3840	7680	15360	23040
1	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	0
50	0	0	0	0	0	0	0	0	0	0	0
100	0	0	0	0	0	0	0	0	0	0	0
500	0	0	0	0	0	0	0	0	0	0	0
1000	0	0	3.3	6.7	10	10	10	10	10	10	6.7
5000	0	0	3.3	6.7	6.7	10	10	10	10	6.7	13.3
10000	0	0	3.3	3.3	3.3	10	10	6.7	13.3	13.3	13.3
50000	0	0	3.3	6.7	10	10	6.7	10	10	10	10
70000	0	0	3.3	6.7	6.7	10	6.7	10	10	13.3	13.3
90000	0	0	3.3	3.3	6.7	6.7	3.3	6.7	13.3	16.7	13.3



ภาพประกอบ 5-18 แสดงข้อมูลจากตาราง 5-15 เมื่อนำมาแสดงเป็นกราฟ

ตาราง 5-16 แสดงความถูกต้องในการจำแนกในแบบ 1 ชั้นซ่อน
เมื่อใช้จำนวนจุดผิดพลาดในการเรียนรู้ 5 พิกเซล
กับลายวงจรพิมพ์ที่ทำการเคลือบแล้ว

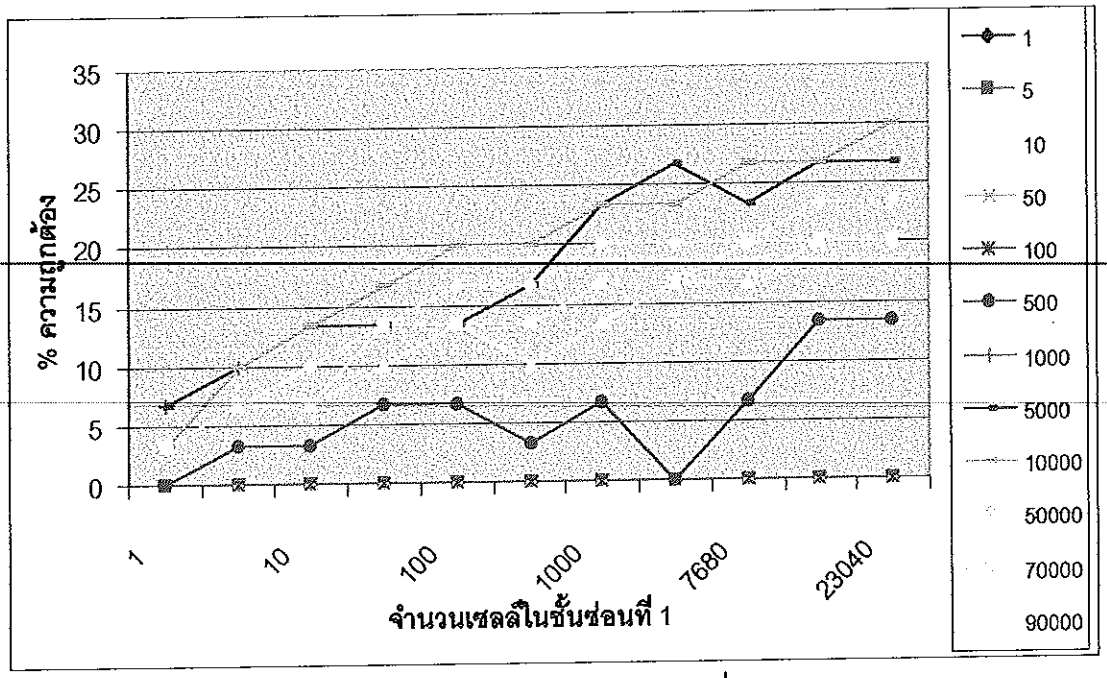
จำนวนเซลล์ชั้นที่ 1	จำนวนจุดที่ผิดพลาดที่ใช้ในการทดสอบ										
	1	5	10	50	100	500	1000	3840	7680	15360	23040
1	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	0
50	0	0	0	0	0	0	0	0	0	0	0
100	0	0	0	0	0	0	0	0	0	0	0
500	0	0	0	0	0	0	0	0	0	0	0
1000	0	3.3	3.3	6.7	6.7	3.3	10	10	10	13.3	13.3
5000	0	0	6.7	6.7	10	16.7	10	10	13.3	16.7	16.7
10000	0	3.3	6.7	10	10	10	10	13.3	16.7	20	26.7
50000	0	3.3	6.7	6.7	6.7	10	13.3	16.7	16.7	20	20
70000	0	3.3	3.3	10	6.7	10	10	13.3	16.7	20	20
90000	0	0	3.3	6.7	10	10	10	13.3	13.3	10	20



ภาพประกอบ 5-19 แสดงข้อมูลจากตาราง 5-16 เมื่อนำมาแสดงเป็นกราฟ

ตาราง 5-17 แสดงความถูกต้องในการจำแนกในแบบ 1 ชั้นซ่อน
เมื่อใช้จำนวนจุดผิดพลาดในการเรียนรู้ 10 พิกเซล
กับลายวงจรพิมพ์ที่ทำการเคลือบแล้ว

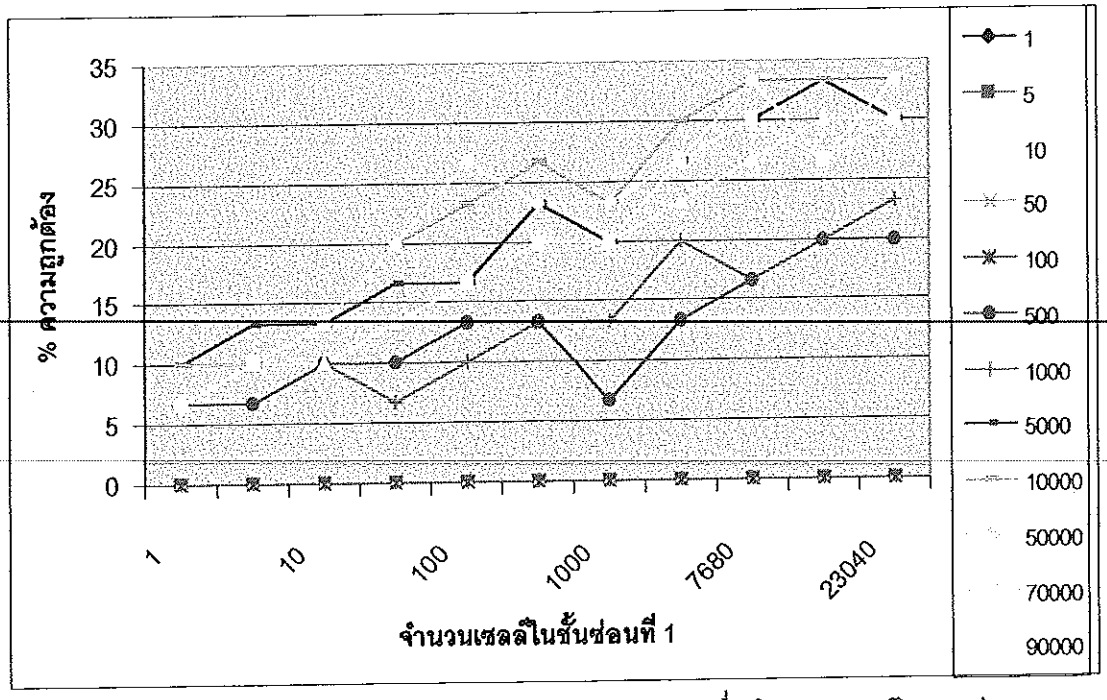
จำนวนเซลล์ชั้นที่ 1	จำนวนจุดที่ผิดพลาดที่ใช้ในการทดสอบ										
	1	5	10	50	100	500	1000	3840	7680	15360	23040
1	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	0
50	0	0	0	0	0	0	0	0	0	0	0
100	0	0	0	0	0	0	0	0	0	0	0
500	0	3.3	3.3	6.7	6.7	3.3	6.7	0	6.7	13.3	13.3
1000	6.7	10	10	13.3	13.3	13.3	16.7	16.7	20	20	20
5000	6.7	10	13.3	13.3	13.3	16.7	23.3	26.7	23.3	26.7	26.7
10000	3.3	10	13.3	16.7	20	20	23.3	23.3	26.7	26.7	30
50000	3.3	6.7	10	13.3	16.7	16.7	20	20	20	23.3	23.3
70000	3.3	6.7	6.7	13.3	13.3	13.3	16.7	16.7	20	20	20
90000	3.3	6.7	6.7	10	13.3	10	13.3	16.7	16.7	20	20



ภาพประกอบ 5-20 แสดงข้อมูลจากตาราง 5-17 เมื่อนำมาแสดงเป็นกราฟ

ตาราง 5-18 แสดงความถูกต้องในการจำแนกในแบบ 1 ชั้นซ่อน
เมื่อใช้จำนวนจุดผิดพลาดในการเรียนรู้ 50 พิกเซล
กับลายวงจรพิมพ์ที่ทำการเคลือบแล้ว

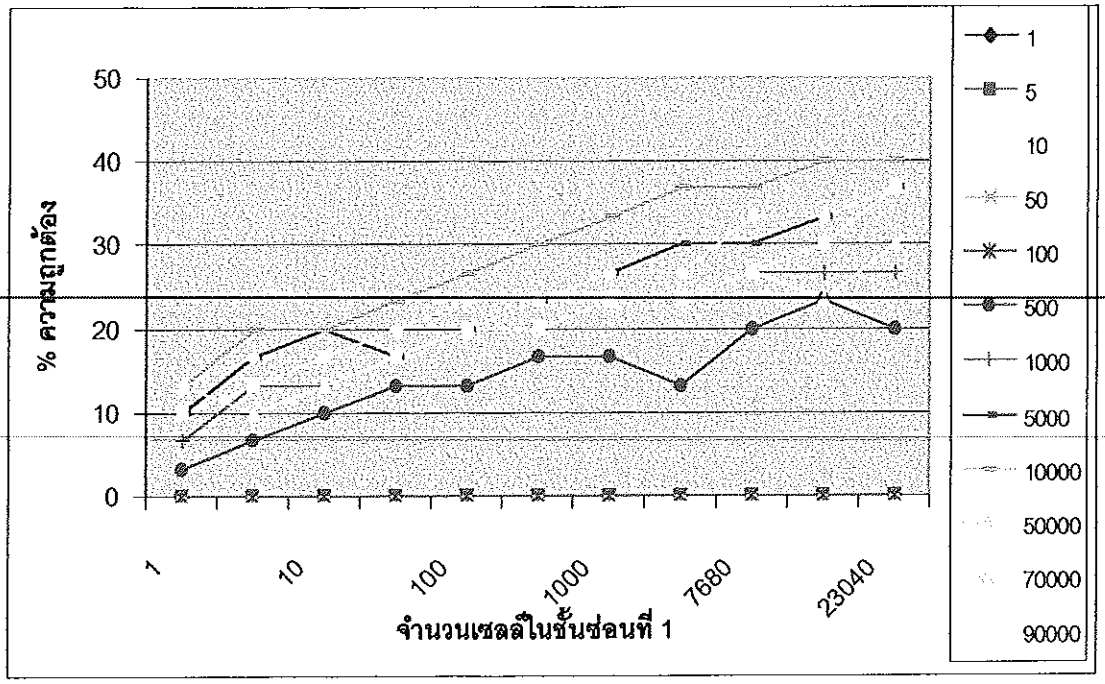
จำนวนเซลล์ชั้นที่ 1	จำนวนจุดที่ผิดพลาดที่ใช้ในการทดสอบ										
	1	5	10	50	100	500	1000	3840	7680	15360	23040
1	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	0
50	0	0	0	0	0	0	0	0	0	0	0
100	0	0	0	0	0	0	0	0	0	0	0
500	6.7	6.7	10	10	13.3	13.3	6.7	13.3	16.7	20	20
1000	6.7	10	10	6.7	10	13.3	13.3	20	16.7	20	23.3
5000	10	13.3	13.3	16.7	16.7	23.3	20	26.7	30	33.3	30
10000	10	10	13.3	20	23.3	26.7	23.3	30	33.3	33.3	33.3
50000	6.7	10	13.3	20	26.7	23.3	23.3	26.7	33.3	26.7	33.3
70000	6.7	10	13.3	13.3	16.7	20	20	26.7	30	26.7	30
90000	6.7	10	10	13.3	16.7	20	20	23.3	26.7	30	30



ภาพประกอบ 5-21 แสดงข้อมูลจากตาราง 5-18 เมื่อนำมาแสดงเป็นกราฟ

ตาราง 5-19 แสดงความถูกต้องในการจำแนกในแบบ 1 ชั้นซ่อน
เมื่อใช้จำนวนจุดผิดพลาดในการเรียนรู้ 100 พิกเซล
กับลายวงจรพิมพ์ที่ทำการเคลือบแล้ว

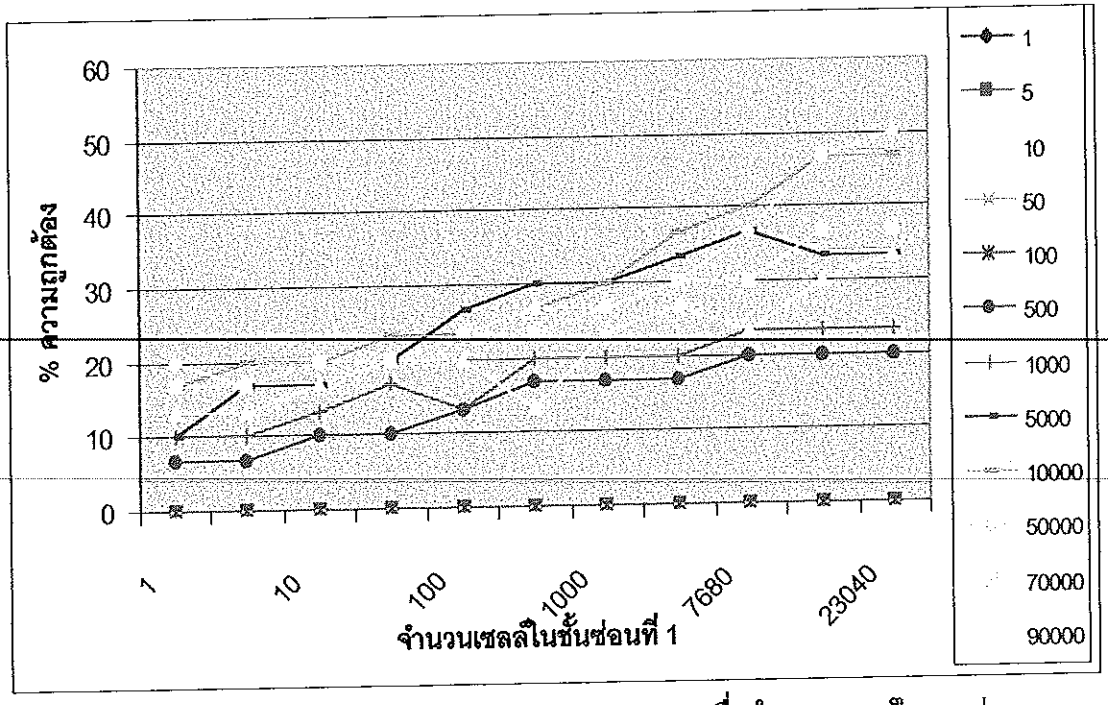
จำนวนเซลล์ชั้นที่ 1	จำนวนจุดที่ผิดพลาดที่ใช้ในการทดสอบ										
	1	5	10	50	100	500	1000	3840	7680	15360	23040
1	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	0
50	0	0	0	0	0	0	0	0	0	0	0
100	0	0	0	0	0	0	0	0	0	0	0
500	3.3	6.7	10	13.3	13.3	16.7	16.7	13.3	20	23.3	20
1000	6.7	13.3	13.3	16.7	20	20	23.3	23.3	26.7	26.7	26.7
5000	10	16.7	20	16.7	20	23.3	26.7	30	30	33.3	36.7
10000	13.3	20	20	23.3	26.7	30	33.3	36.7	36.7	40	40
50000	13.3	16.7	16.7	20	23.3	23.3	26.7	26.7	33.3	33.3	36.7
70000	10	13.3	16.7	16.7	20	20	23.3	23.3	26.7	30	36.7
90000	13.3	10	13.3	16.7	20	23.3	23.3	23.3	26.7	23.3	30



ภาพประกอบ 5-22 แสดงข้อมูลจากตาราง 5-19 เมื่อนำมาแสดงเป็นกราฟ

ตาราง 5-20 แสดงความถูกต้องในการจำแนกในแบบ 1 ชั้นซ่อน
เมื่อใช้จำนวนจุดผิดพลาดในการเรียนรู้ 500 พิกเซล
กับลายวงจรพิมพ์ที่ทำการเคลือบแล้ว

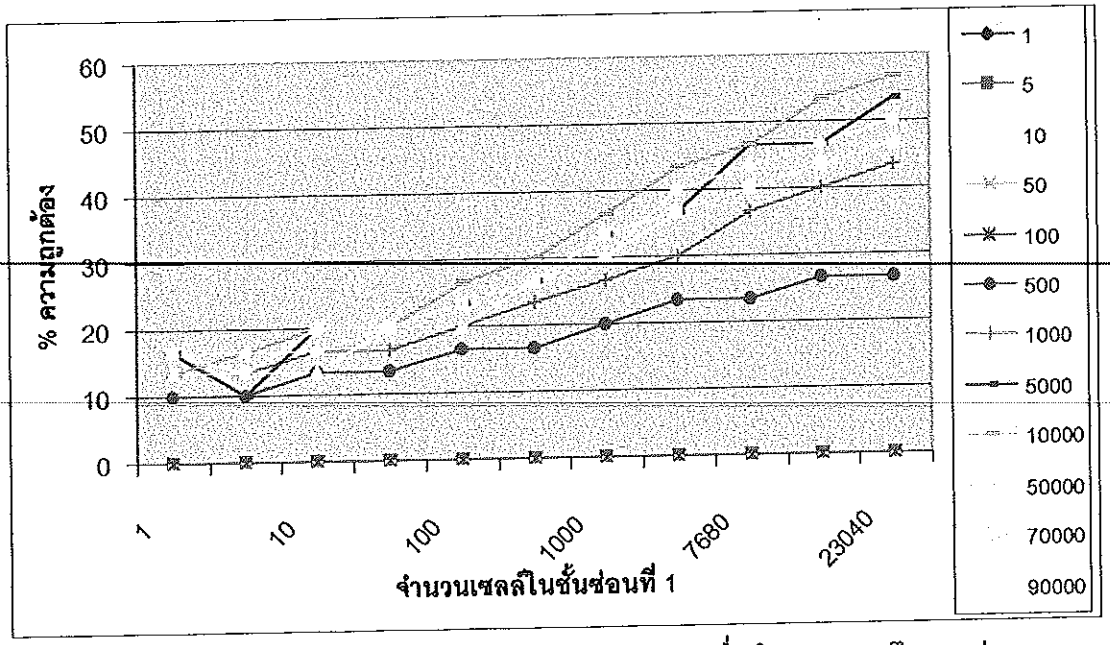
จำนวนเซลล์ชั้นที่ 1	จำนวนจุดที่ผิดพลาดที่ใช้ในการทดสอบ										
	1	5	10	50	100	500	1000	3840	7680	15360	23040
1	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	0
50	0	0	0	0	0	0	0	0	0	0	0
100	0	0	0	0	0	0	0	0	0	0	0
500	6.7	6.7	10	10	13.3	16.7	16.7	16.7	20	20	20
1000	10	10	13.3	16.7	13.3	20	20	20	23.3	23.3	23.3
5000	10	16.7	16.7	20	26.7	30	30	33.3	36.7	33.3	33.3
10000	16.7	20	20	23.3	23.3	26.7	30	36.7	40	46.7	46.7
50000	20	16.7	20	20	23.3	26.7	26.7	30	36.7	46.7	50
70000	16.7	16.7	20	20	23.3	23.3	26.7	30	30	36.7	36.7
90000	13.3	13.3	16.7	20	20	13.3	23.3	26.7	23.3	30	33.3



ภาพประกอบ 5-23 แสดงข้อมูลจากตาราง 5-20 เมื่อนำมาแสดงเป็นกราฟ

ตาราง 5-21 แสดงความถูกต้องในการจำแนกในแบบ 1 ชั้นซ่อน
เมื่อใช้จำนวนจุดผิดพลาดในการเรียนรู้ 1000 พิกเซล
กับลายวงจรพิมพ์ที่ทำการเคลือบแล้ว

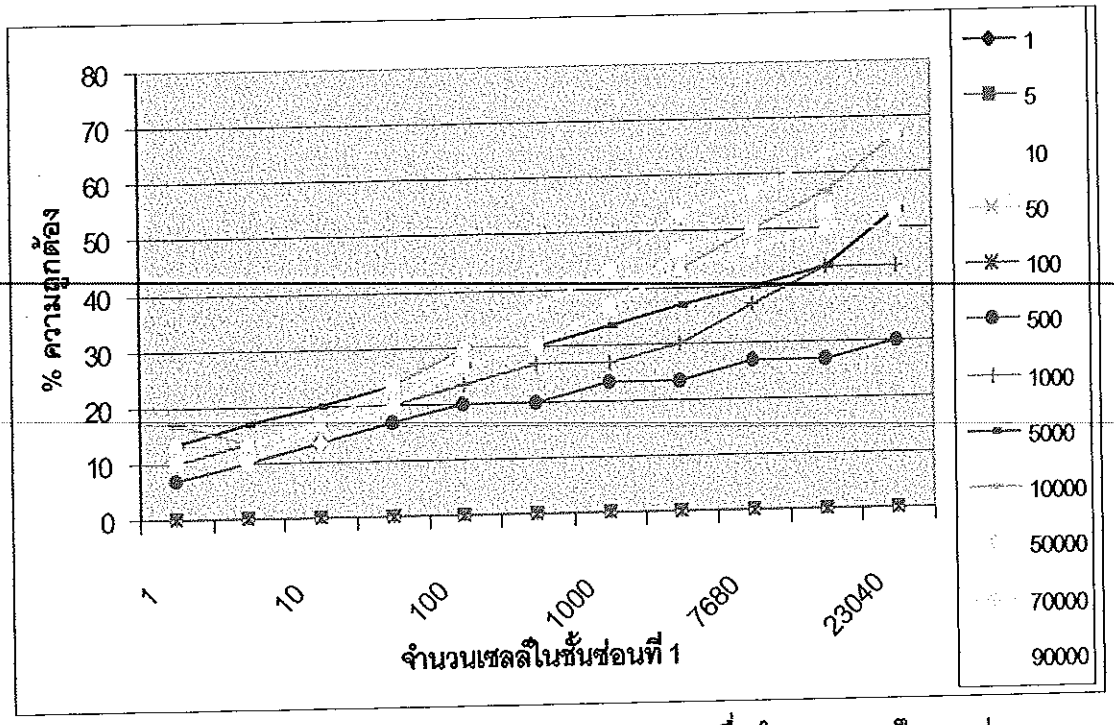
จำนวนเซลล์ชั้นที่ 1	จำนวนจุดที่ผิดพลาดที่ใช้ในการทดสอบ										
	1	5	10	50	100	500	1000	3840	7680	15360	23040
1	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	0
50	0	0	0	0	0	0	0	0	0	0	0
100	0	0	0	0	0	0	0	0	0	0	0
500	10	10	13.3	13.3	16.7	16.7	20	23.3	23.3	26.7	26.7
1000	13.3	13.3	16.7	16.7	20	23.3	26.7	30	36.7	40	43.3
5000	16.7	10	20	20	23.3	26.7	33.3	36.7	46.7	46.7	53.3
10000	13.3	16.7	20	20	26.7	30	36.7	43.3	46.7	53.3	56.7
50000	16.7	16.7	16.7	20	23.3	26.7	30	40	43.3	46.7	50
70000	16.7	13.3	20	20	23.3	26.7	33.3	36.7	40	43.3	46.7
90000	13.3	13.3	13.3	20	20	26.7	30	36.7	40	43.3	46.7



ภาพประกอบ 5-24 แสดงข้อมูลจากตาราง 5-21 เมื่อนำมาแสดงเป็นกราฟ

ตาราง 5-22 แสดงความถูกต้องในการจำแนกในแบบ 1 ชั้นซ่อน
เมื่อใช้จำนวนจุดผิดพลาดในการเรียนรู้ 5000 พิกเซล
กับลายวงจรพิมพ์ที่ทำการเคลือบแล้ว

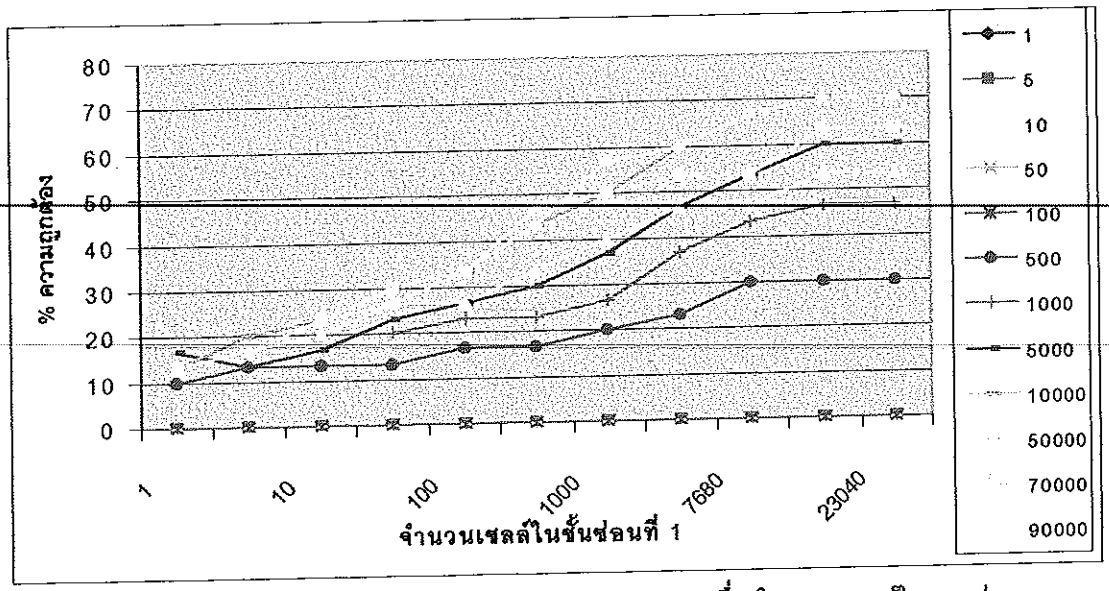
จำนวนเซลล์ชั้นที่ 1	จำนวนจุดที่ผิดพลาดที่ใช้ในการทดสอบ										
	1	5	10	50	100	500	1000	3840	7680	15360	23040
1	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	0
50	0	0	0	0	0	0	0	0	0	0	0
100	0	0	0	0	0	0	0	0	0	0	0
500	6.7	10	13.3	16.7	20	20	23.3	23.3	26.7	26.7	30
1000	10	13.3	16.7	20	23.3	26.7	26.7	30	36.7	43.3	43.3
5000	13.3	16.7	20	23.3	26.7	30	33.3	36.7	40	43.3	53.3
10000	16.7	13.3	16.7	23.3	30	30	36.7	43.3	50	56.7	66.7
50000	13.3	13.3	13.3	20	30	36.7	43.3	53.3	56.7	63.3	66.7
70000	10	10	16.7	23.3	26.7	30	36.7	46.7	50	53.3	53.3
90000	13.3	13.3	16.7	20	26.7	30	36.7	43.3	46.7	50	50



ภาพประกอบ 5-25 แสดงข้อมูลจากตาราง 5-22 เมื่อนำมาแสดงเป็นกราฟ

ตาราง 5-23 แสดงความถูกต้องในการจำแนกในแบบ 1 ชั้นซ่อน
เมื่อใช้จำนวนจุดผิดพลาดในการเรียนรู้ 10000 พิกเซล
กับลายวงจรพิมพ์ที่ทำการเคลือบแล้ว

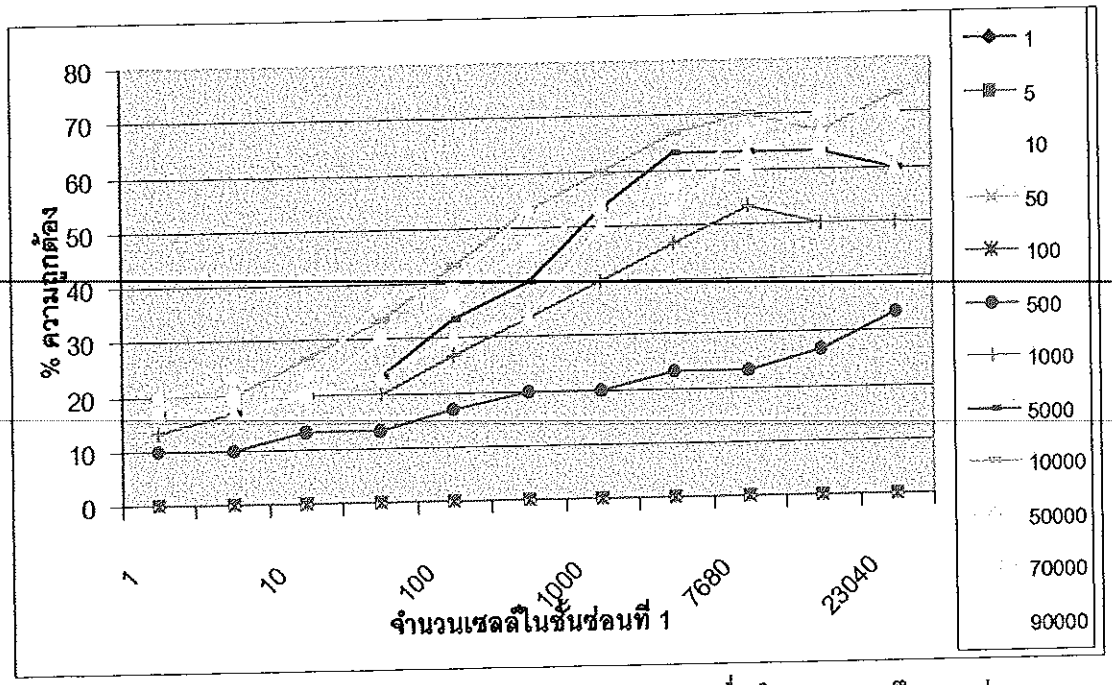
จำนวนเซลล์ชั้นที่ 1	จำนวนจุดที่ผิดพลาดที่ใช้ในการทดสอบ										
	1	5	10	50	100	500	1000	3840	7680	15360	23040
1	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	0
50	0	0	0	0	0	0	0	0	0	0	0
100	0	0	0	0	0	0	0	0	0	0	0
500	10	13.3	13.3	13.3	16.7	16.7	20	23.3	30	30	30
1000	13.3	16.7	20	20	23.3	23.3	26.7	36.7	43.3	46.7	46.7
5000	16.7	13.3	16.7	23.3	26.7	30	36.7	46.7	53.3	60	60
10000	13.3	20	23.3	30	33.3	43.3	50	60	66.7	70	70
50000	13.3	16.7	23.3	26.7	33.3	43.3	56.7	60	66.7	70	70
70000	13.3	16.7	23.3	30	33.3	46.7	50	53.3	53.3	63.3	63.3
90000	13.3	16.7	20	26.7	26.7	33.3	40	46.7	46.7	53.3	53.3



ภาพประกอบ 5-26 แสดงข้อมูลจากตาราง 5-23 เมื่อนำมาแสดงเป็นกราฟ

ตาราง 5-24 แสดงความถูกต้องในการจำแนกในแบบ 1 ชั้นซ่อน
เมื่อใช้จำนวนจุดผิดพลาดในการเรียนรู้ 50000 พิกเซล
กับลายวงจรพิมพ์ที่ทำการเคลือบแล้ว

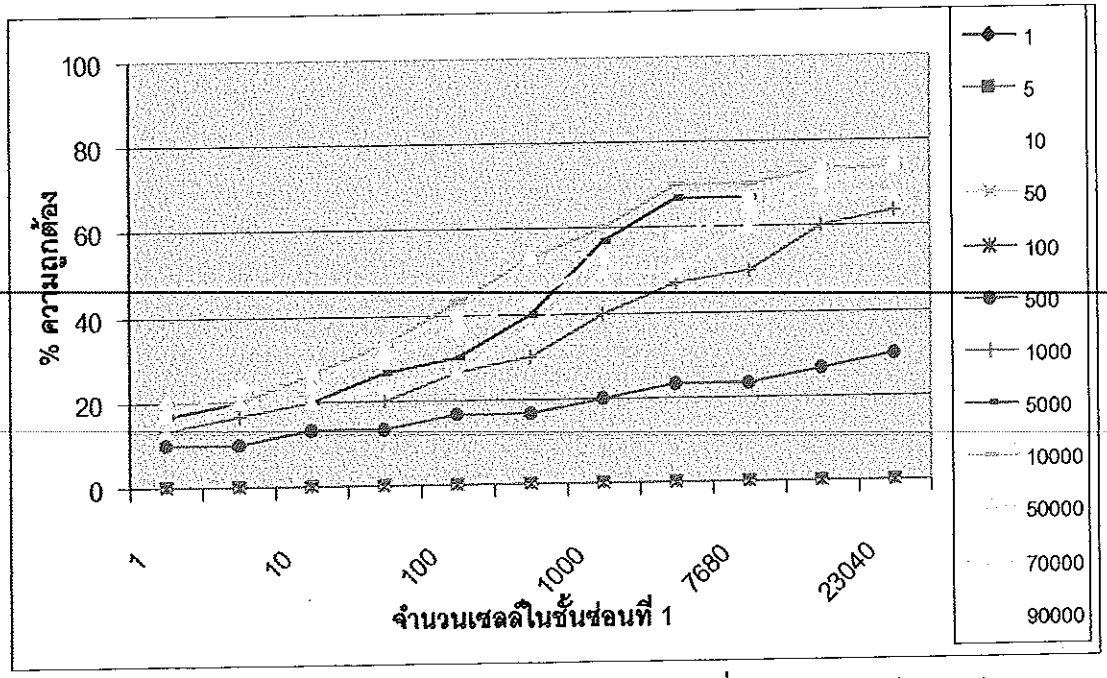
จำนวนเซลล์ชั้นที่ 1	จำนวนจุดที่ผิดพลาดที่ใช้ในการทดสอบ										
	1	5	10	50	100	500	1000	3840	7680	15360	23040
1	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	0
50	0	0	0	0	0	0	0	0	0	0	0
100	0	0	0	0	0	0	0	0	0	0	0
500	10	10	13.3	13.3	16.7	20	20	23.3	23.3	26.7	33.3
1000	13.3	16.7	20	20	26.7	33.3	40	46.7	53.3	50	50
5000	16.7	16.7	20	23.3	33.3	40	53.3	63.3	63.3	63.3	60
10000	20	20	26.7	33.3	43.3	53.3	60	66.7	70	66.7	73.3
50000	20	23.3	23.3	36.7	40	53.3	53.3	56.7	66.7	70	70
70000	16.7	20	20	30	36.7	46.7	53.3	56.7	60	63.3	63.3
90000	16.7	16.7	20	23.3	30	33.3	50	53.3	63.3	66.7	60



ภาพประกอบ 5-27 แสดงข้อมูลจากตาราง 5-24 เมื่อนำมาแสดงเป็นกราฟ

ตาราง 5-25 แสดงความถูกต้องในการจำแนกในแบบ 1 ชั้นซ่อน
เมื่อใช้จำนวนจุดผิดพลาดในการเรียนรู้ 70000 พิกเซล
กับลายวงจรพิมพ์ที่ทำการ เคลือบแล้ว

จำนวนเซลล์ชั้นที่ 1	จำนวนจุดที่ผิดพลาดที่ใช้ในการทดสอบ										
	1	5	10	50	100	500	1000	3840	7680	15360	23040
1	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	0
50	0	0	0	0	0	0	0	0	0	0	0
100	0	0	0	0	0	0	0	0	0	0	0
500	10	10	13.3	13.3	16.7	16.7	20	23.3	23.3	26.7	30
1000	13.3	16.7	20	20	26.7	30	40	46.7	50	60	63.3
5000	16.7	20	20	26.7	30	40	56.7	66.7	66.7	73.3	73.3
10000	20	20	26.7	33.3	43.3	53.3	60	70	70	73.3	73.3
50000	20	23.3	23.3	33.3	40	53.3	53.3	56.7	66.7	73.3	76.7
70000	16.7	23.3	26.7	30	36.7	43.3	53.3	56.7	63.3	70	73.3
90000	13.3	20	20	23.3	26.7	36.7	50	50	60	66.7	66.7



ภาพประกอบ 5-28 แสดงข้อมูลจากตาราง 5-25 เมื่อนำมาแสดงเป็นกราฟ

ตาราง 5-26 แสดงความถูกต้องในการจำแนกในแบบ 2 ชั้นซ่อน
เมื่อใช้จำนวนจุดผิดพลาดในการเรียนรู้คือ 7680 จุด (10 เปอร์เซ็นต์)

จำนวนเซลล์ชั้นที่ 1	จำนวนเซลล์ชั้นที่ 2	จำนวนจุดที่ผิดพลาดที่ใช้ในการทดสอบ										
		1	5	10	50	100	500	1000	3840	7680	15360	23040
100	100	0	0	0	0	0	0	0	0	0	0	0
100	500	0	0	0	0	0	0	0	0	0	0	0
100	1,000	-	3.3	6.7	6.7	6.7	10	10.3	13.3	16.7	20	26.7
100	5,000	-	3.3	6.7	6.7	10	13.3	13.3	16.7	17.2	20	24.1
100	10,000	3.3	6.7	3.3	10	13.3	16.7	20	20	23.3	23.3	23.3
100	50,000	3.3	6.7	6.7	10	10.3	16.7	16.7	17.2	23.3	20	23.3
100	70,000	0	0	0	0	0	0	0	0	0	0	0
100	90,000	0	0	0	0	0	0	0	0	0	0	0
500	100	0	0	0	0	0	0	0	0	0	0	0
500	500	16.7	16.7	20	23.3	30	33.3	34.5	41.4	48.3	46.7	53.3
500	1,000	10	16.7	20	24.1	27.6	30	33.3	43.3	43.3	44.8	51.7
500	5,000	10	20	16.7	20	23.3	23.3	26.7	30	33.3	33.3	36.7
500	10,000	6.7	10	16.7	16.7	20	24.1	26.7	27.6	31	31	33.3
500	50,000	6.7	10.3	10.3	16.7	17.2	16.7	20	23.3	26.7	26.7	30
500	70,000	0	0	0	0	0	0	0	0	0	0	0
500	90,000	0	0	0	0	0	0	0	0	0	0	0
1,000	100	10	16.7	20	23.3	23.3	30	33.3	43.3	43.3	44.8	51.7
1,000	500	16.7	17.2	20.7	24.1	26.7	30	30	37.9	46.7	48.3	48.3
1,000	1,000	10	10	10.3	16.7	20.7	33.3	36.7	43.3	51.7	60	65.5
1,000	5,000	6.7	6.7	10	13.3	13.3	20	26.7	33.3	40	43.3	50
1,000	10,000	3.3	6.7	6.7	10	13.3	20.7	20.7	26.7	30	33.3	40
1,000	50,000	3.3	3.3	6.7	10	10.3	16.7	20	23.3	26.7	30	36.7
1,000	70,000	0	0	0	0	0	0	0	0	0	0	0
1,000	90,000	0	0	0	0	0	0	0	0	0	0	0

ตาราง 5-26 (ต่อ)

จำนวนเซลล์ชั้นที่ 1	จำนวนเซลล์ชั้นที่ 2	จำนวนจุดที่ผิดพลาดที่ใช้ในการทดสอบ										
		1	5	10	50	100	500	1000	3840	7680	15360	23040
5,000	100	-	3.3	3.3	6.7	10	10.3	13.3	16.7	20	23.3	26.7
5,000	500	3.3	10	16.7	20	20	23.3	26.7	33.3	36.7	36.7	40
5,000	1,000	3.3	6.7	13.3	16.7	20	23.3	26.7	40	46.7	53.3	60
5,000	5,000	6.7	10	16.7	20	23.3	33.3	40	56.6	70	73.3	73.3
5,000	10,000	6.7	6.7	10	16.7	23.3	23.3	30	43.3	50	63.3	70
5,000	50,000	3.3	3.3	6.7	10	16.7	20	30	36.7	43.3	50	53.3
5,000	70,000	3.3	6.7	6.7	13.3	13.3	20	26.7	26.7	33.3	43.3	46.7
5,000	90,000	0	0	0	0	0	0	0	0	0	0	0
10,000	100	3.3	6.7	6.7	10	13.3	16.7	16.7	20	23.3	26.7	30
10,000	500	6.7	10	16.7	16.7	20	23.3	26.7	30	33.3	36.7	43.3
10,000	1,000	6.7	10	16.7	16.7	20	23.3	26.7	30	33.3	40	46.7
10,000	5,000	6.7	6.7	10	16.7	23.3	30	36.7	40	46.7	56.7	63.3
10,000	10,000	3.3	3.3	6.7	10	20	26.7	33.3	46.7	56.7	66.7	70
10,000	50,000	0	0	0	0	0	0	0	0	0	0	0
10,000	70,000	0	0	0	0	0	0	0	0	0	0	0
10,000	90,000	0	0	0	0	0	0	0	0	0	0	0
50,000	100	3.3	6.7	6.7	10	10.3	16.7	16.7	20	23.3	26.7	26.7
50,000	500	6.7	10	10.3	16.7	16.7	16.7	20	23.3	26.7	26.7	30
50,000	1,000	3.3	3.3	6.7	10	10.3	16.7	23.3	23.3	26.7	30	36.7
50,000	5,000	3.3	3.3	6.7	10	16.7	20	23.3	30	36.7	40	46.7
50,000	10,000	0	0	0	0	0	0	0	0	0	0	0
50,000	50,000	0	0	0	0	0	0	0	0	0	0	0
50,000	70,000	0	0	0	0	0	0	0	0	0	0	0
50,000	90,000	0	0	0	0	0	0	0	0	0	0	0

ตาราง 5-26 (ต่อ)

จำนวนเซลล์ชั้นที่ 1	จำนวนเซลล์ชั้นที่ 2	จำนวนจุดที่ผิดพลาดที่ใช้ในการทดสอบ										
		1	5	10	50	100	500	1000	3840	7680	15360	23040
70,000	100	0	0	0	0	0	0	0	0	0	0	0
70,000	500	0	0	0	0	0	0	0	0	0	0	0
70,000	1,000	0	0	0	0	0	0	0	0	0	0	0
70,000	5,000	3.3	6.7	6.7	10	13.3	20	36.7	36.7	40	40	40
70,000	10,000	0	0	0	0	0	0	0	0	0	0	0
70,000	50,000	0	0	0	0	0	0	0	0	0	0	0
70,000	70,000	0	0	0	0	0	0	0	0	0	0	0
70,000	90,000	0	0	0	0	0	0	0	0	0	0	0

ตาราง 5-27 แสดงความถูกต้องในการจำแนกในแบบ 2 ชั้นซ่อน
เมื่อใช้จำนวนจุดผิดพลาดในการเรียนรู้คือ 15360 จุด (20 เปอร์เซ็นต์)

จำนวนเซลล์ชั้นที่ 1	จำนวนเซลล์ชั้นที่ 2	จำนวนจุดที่ผิดพลาดที่ใช้ในการทดสอบ										
		1	5	10	50	100	500	1000	3840	7680	15360	23040
100	100	0	0	0	0	0	0	0	0	0	0	0
100	500	0	0	0	0	0	0	0	0	0	0	0
100	1,000	3.3	3.3	6.7	6.7	10	13.3	20	26.7	30	33.3	33.3
100	5,000	3.3	3.3	6.7	6.7	10	13.3	16.7	16.7	17.2	20	26.7
100	10,000	6.7	6.7	3.3	10	13.3	16.7	20	20	23.3	23.3	26.7
100	50,000	6.7	6.7	10	13.3	16.7	16.7	16.7	17.2	20	23.3	23.3
100	70,000	0	0	0	0	0	0	0	0	0	0	0
100	90,000	0	0	0	0	0	0	0	0	0	0	0
500	100	0	0	0	0	0	0	0	0	0	0	0
500	500	13.3	16.7	20	26.7	30	33.3	36.7	40	43.3	46.7	50
500	1,000	6.7	13.3	20	26.7	26.7	30	33.3	43.3	43.3	46.7	53.3
500	5,000	10	13.3	16.7	20	20	23.3	26.7	30	30	33.3	40
500	10,000	3.3	10	16.7	16.7	20	23.3	26.7	27.6	30	33.3	36.7
500	50,000	3.3	13.3	10.3	16.7	13.3	16.7	20	23.3	23.3	26.7	26.7
500	70,000	0	0	0	0	0	0	0	0	0	0	0
500	90,000	0	0	0	0	0	0	0	0	0	0	0
1,000	100	6.7	10	20	23.3	23.3	30	36.7	43.3	46.7	50	56.7
1,000	500	10	13.3	20.7	24.1	26.7	30	33.3	40	46.7	53.3	56.7
1,000	1,000	10	13.3	10	16.7	20	30	36.7	43.3	50	56.7	63.3
1,000	5,000	3.3	6.7	10	10	13.3	20	33.3	40	43.3	53.3	60
1,000	10,000	3.3	3.3	6.7	10	13.3	23.3	20	30	33.3	43.3	53.3
1,000	50,000	3.3	3.3	6.7	6.7	10	16.7	20	23.3	26.7	36.7	43.3
1,000	70,000	0	0	0	0	0	0	0	0	0	0	0
1,000	90,000	0	0	0	0	0	0	0	0	0	0	0

ตาราง 5-27 (ต่อ)

จำนวนเซลล์ชั้นที่ 1	จำนวนเซลล์ชั้นที่ 2	จำนวนจุดที่ผิดพลาดที่ใช้ในการทดสอบ										
		1	5	10	50	100	500	1000	3840	7680	15360	23040
5,000	100	3.3	3.3	6.7	6.7	10	10.3	13.3	16.7	20	23.3	26.7
5,000	500	3.3	10	16.7	20	20	23.3	26.7	33.3	36.7	36.7	43.3
5,000	1,000	3.3	6.7	13.3	16.7	20	23.3	26.7	40	46.7	53.3	60
5,000	5,000	6.7	10	16.7	20	23.3	33.3	40	56.6	70	76.7	83.3
5,000	10,000	6.7	6.7	10	16.7	23.3	23.3	30	36.7	43.3	53.3	66.7
5,000	50,000	3.3	3.3	6.7	10	16.7	20	23.3	30	36.7	40	50
5,000	70,000	3.3	6.7	6.7	13.3	13.3	20	26.7	26.7	33.3	43.3	46.7
5,000	90,000	0	0	0	0	0	0	0	0	0	0	0
10,000	100	3.3	6.7	6.7	10	13.3	16.7	20	20	23.3	23.3	23.3
10,000	500	6.7	10	16.7	16.7	20	23.3	26.7	27.6	30	36.7	40
10,000	1,000	6.7	10	16.7	16.7	20	24.1	26.7	30	36.7	40	46.7
10,000	5,000	3.3	6.7	10	16.7	20	23.3	30	36.7	40	43.3	50
10,000	10,000	3.3	3.3	6.7	10	20	23.3	26.7	30	40	40	53.3
10,000	50,000	0	0	0	0	0	0	0	0	0	0	0
10,000	70,000	0	0	0	0	0	0	0	0	0	0	0
10,000	90,000	0	0	0	0	0	0	0	0	0	0	0
50,000	100	3.3	3.3	6.7	10	13.3	16.7	16.7	20	23.3	30	30
50,000	500	6.7	3.3	10	16.7	16.7	20	20	23.3	26.7	26.7	33.3
50,000	1,000	3.3	3.3	6.7	10	10.3	16.7	23.3	23.3	26.7	33.3	40
50,000	5,000	3.3	6.7	10	10	16.7	20	23.3	30	36.7	43.3	46.7
50,000	10,000	6.7	6.7	10.0	10.0	13.3	20.0	26.7	30.0	33.3	43.3	50.0
50,000	50,000	0	0	0	0	0	0	0	0	0	0	0
50,000	70,000	0	0	0	0	0	0	0	0	0	0	0
50,000	90,000	0	0	0	0	0	0	0	0	0	0	0

ตาราง 5-27 (ต่อ)

จำนวนเซลล์ชั้นที่ 1	จำนวนเซลล์ชั้นที่ 2	จำนวนจุดที่ผิดพลาดที่ใช้ในการทดสอบ										
		1	5	10	50	100	500	1000	3840	7680	15360	23040
70,000	100	0	0	0	0	0	0	0	0	0	0	0
70,000	500	0	0	0	0	0	0	0	0	0	0	0
70,000	1,000	0	0	0	0	0	0	0	0	0	0	0
70,000	5,000	3.3	6.7	10	10	16.7	23.3	26.7	30	33.3	40	46.7
70,000	10,000	0	0	0	0	0	0	0	0	0	0	0
70,000	50,000	0	0	0	0	0	0	0	0	0	0	0
70,000	70,000	0	0	0	0	0	0	0	0	0	0	0
70,000	90,000	0	0	0	0	0	0	0	0	0	0	0

ตาราง 5-28 แสดงความถูกต้องในการจำแนกในแบบ 2 ชั้นซ่อน
เมื่อใช้จำนวนจุดผิดพลาดในการเรียนรู้คือ 23040 จุด (30 เปอร์เซ็นต์)

จำนวนเซลล์ชั้นที่ 1	จำนวนเซลล์ชั้นที่ 2	จำนวนจุดที่ผิดพลาดที่ใช้ในการทดสอบ										
		1	5	10	50	100	500	1000	3840	7680	15360	23040
100	100	0	0	0	0	0	0	0	0	0	0	0
100	500	0	0	0	0	0	0	0	0	0	0	0
100	1,000	6.7	6.7	10	10	13.3	10	16.7	16.7	20	30	36.7
100	5,000	6.7	6.7	10	6.7	10	13.3	16.7	20	26.7	30	33.3
100	10,000	6.7	6.7	10	13.3	16.7	20	20	23.3	26.7	26.7	30
100	50,000	6.7	6.7	10	10	13.3	16.7	16.7	20	23.3	30	30
100	70,000	0	0	0	0	0	0	0	0	0	0	0
100	90,000	0	0	0	0	0	0	0	0	0	0	0
500	100	0	0	0	0	0	0	0	0	0	0	0
500	500	10	16.7	20	26.7	30	33.3	36.7	40	43.3	46.7	56.7
500	1,000	10	16.7	20	26.7	30	36.7	40	46.7	50	50	53.3
500	5,000	10	13.3	16.7	20	20	23.3	26.7	30	33.3	36.7	40
500	10,000	10	13.3	16.7	20	20	23.3	26.7	27.6	30	33.3	36.7
500	50,000	6.7	13.3	13.3	16.7	16.7	20	20	23.3	26.7	26.7	30
500	70,000	0	0	0	0	0	0	0	0	0	0	0
500	90,000	0	0	0	0	0	0	0	0	0	0	0
1,000	100	6.7	10	13.3	16.7	20	30	36.7	46.7	46.7	50	60
1,000	500	10	13.3	16.7	23.3	26.7	30	36.7	40	46.7	50	53.3
1,000	1,000	10	13.3	13.3	16.7	20	33.3	40	43.3	50	60	66.7
1,000	5,000	6.7	10	13.3	13.3	20	26.7	30	36.7	40	46.7	53.3
1,000	10,000	6.7	6.7	10	10	13.3	20	30	30	36.7	40	46.7
1,000	50,000	3.3	6.7	10	6.7	10	16.7	20	23.3	23.3	30	40
1,000	70,000	0	0	0	0	0	0	0	0	0	0	0
1,000	90,000	0	0	0	0	0	0	0	0	0	0	0

ตาราง 5-28 (ต่อ)

จำนวนเซลล์ชั้นที่ 1	จำนวนเซลล์ชั้นที่ 2	จำนวนจุดที่ผิดพลาดที่ใช้ในการทดสอบ										
		1	5	10	50	100	500	1000	3840	7680	15360	23040
5,000	100	6.7	6.7	3.3	10	10	20	23.3	30	33.3	36.7	46.7
5,000	500	10	10	16.7	20	23.3	26.7	30	36.7	40	46.7	50
5,000	1,000	6.7	10	13.3	20	20	26.7	33.3	40	50	56.7	66.7
5,000	5,000	6.7	10	16.7	20	26.7	33.3	43.3	60	70	76.7	83.3
5,000	10,000	6.7	6.7	10	16.7	20	30	36.7	40	50	56.7	66.7
5,000	50,000	6.7	6.7	10	10	16.7	23.3	30	33.3	40	53.3	60
5,000	70,000	0	0	0	0	0	0	0	0	0	0	0
5,000	90,000	0	0	0	0	0	0	0	0	0	0	0
10,000	100	6.7	6.7	13.3	10	16.7	16.7	20	26.7	30	30	33.3
10,000	500	6.7	13.3	16.7	16.7	20	26.7	30	33.3	36.7	36.7	43.3
10,000	1,000	6.7	10	16.7	20	23.3	26.7	26.7	30	36.7	40	46.7
10,000	5,000	6.7	10	10	16.7	23.3	23.3	30	43.3	46.7	50	60
10,000	10,000	6.7	10	13.3	16.7	20	26.7	30	36.7	46.7	56.7	63.3
10,000	50,000	0	0	0	0	0	0	0	0	0	0	0
10,000	70,000	0	0	0	0	0	0	0	0	0	0	0
10,000	90,000	0	0	0	0	0	0	0	0	0	0	0
50,000	100	6.7	6.7	10	13.3	13.3	16.7	20	26.7	30	36.7	36.7
50,000	500	6.7	3.3	10	13.3	16.7	20	23.3	26.7	33.3	36.7	40
50,000	1,000	6.7	6.7	10	16.7	13.3	20	26.7	30	36.7	40	53.3
50,000	5,000	3.3	6.7	10	13.3	16.7	20	23.3	30	40	46.7	56.7
50,000	10,000	6.7	10	10	13.3	16.7	23.3	30	33.3	36.7	46.7	50
50,000	50,000	0	0	0	0	0	0	0	0	0	0	0
50,000	70,000	0	0	0	0	0	0	0	0	0	0	0
50,000	90,000	0	0	0	0	0	0	0	0	0	0	0

ตาราง 5-28 (ต่อ)

จำนวนเซลล์ชั้นที่ 1	จำนวนเซลล์ชั้นที่ 2	จำนวนจุดที่ผิดพลาดที่ใช้ในการทดสอบ										
		1	5	10	50	100	500	1000	3840	7680	15360	23040
70,000	100	0	0	0	0	0	0	0	0	0	0	0
70,000	500	0	0	0	0	0	0	0	0	0	0	0
70,000	1,000	0	0	0	0	0	0	0	0	0	0	0
70,000	5,000	10	10	10	13.3	16.7	26.7	26.7	30	33.3	36.7	40
70,000	10,000	0	0	0	0	0	0	0	0	0	0	0
70,000	50,000	0	0	0	0	0	0	0	0	0	0	0
70,000	70,000	0	0	0	0	0	0	0	0	0	0	0
70,000	90,000	0	0	0	0	0	0	0	0	0	0	0

จากตารางที่ 5-4 และ 5-25 จะเป็นการแสดงถึงเปอร์เซ็นต์ความถูกต้องของการจำแนกลายวงจรพิมพ์ระหว่างแบบที่ยังไม่ได้ทำการเคลือบกับแบบที่ได้ทำการเคลือบลายวงจรพิมพ์แล้วในโครงข่ายแบบ 1 ชั้นซ่อน เพื่อใช้ในการสังเกตว่ามีความแตกต่างกันหรือไม่ในการจำแนกระหว่างลายวงจรพิมพ์ที่ยังไม่ได้เคลือบกับลายวงจรพิมพ์ที่ทำการเคลือบแล้ว ก็จะสามารถได้ข้อสังเกตดังนี้

- ลายวงจรพิมพ์ที่ได้เคลือบแล้วจะสามารถจำแนกได้ถูกต้องน้อยกว่าลายวงจรพิมพ์แบบที่ยังไม่ได้เคลือบ
- จำนวนเซลล์ในการจำแนกที่ให้ผลดีที่สุดจะอยู่ในช่วง 10,000 ถึง 50,000 เซลล์
- เมื่อใช้จำนวนพิกเซลความผิดพลาด 1 - 5 จุด ในการเรียนรู้และจำแนกจะมีความถูกต้องในการจำแนกต่ำมากจนถึงไม่สามารถจำแนกได้

จากตารางที่ 5-26 และ 5-28 จะทดลองเฉพาะลายวงจรพิมพ์ที่ไม่ได้เคลือบ ซึ่งแสดงถึงเปอร์เซ็นต์ความถูกต้องของการจำแนก ในโครงข่ายแบบ 2 ชั้นซ่อน (เนื่องจากในนิวรอลเน็ตเวอร์กแบบ 1 ชั้นซ่อนได้พบว่าลายวงจรพิมพ์แบบที่ยังไม่ได้เคลือบให้ผลในการจำแนกได้ดีกว่าในแบบที่ทำการเคลือบแล้ว) ซึ่งจะสามารถได้ข้อสังเกตจากการจำแนกดังนี้

- จากตาราง 5-26 จะสามารถจำแนกได้ถูกต้องสูงสุด 73.3 เปอร์เซ็นต์ โดยใช้จำนวนเซลล์ในชั้นที่ 1 และ 2 เป็น 5,000 และ 5,000 ตามลำดับ

- จากตาราง 5-26 ความถูกต้องน้อยที่สุดที่จำแนกได้เป็นศูนย์

- จากตาราง 5-26 เมื่อใช้จำนวนเซลล์ในชั้นซ่อน 500 และ 100 โครงข่ายจะไม่สามารถลู่เข้าได้

- จากตาราง 5-26 เมื่อใช้จำนวนเซลล์ในชั้นซ่อนที่ 2 เป็น 90,000 โครงข่ายจะไม่สามารถลู่เข้าได้

- จากตาราง 5-27 จะสามารถจำแนกได้ถูกต้องสูงสุด 83.3 เปอร์เซ็นต์ โดยใช้จำนวนเซลล์ในชั้นที่ 1 และ 2 เป็น 5,000 และ 5,000 ตามลำดับ

- จากตาราง 5-27 ความถูกต้องน้อยที่สุดที่จำแนกได้เป็น 0 เปอร์เซ็นต์

- จากตาราง 5-27 เมื่อใช้จำนวนเซลล์ในชั้นซ่อน 500 และ 100 โครงข่ายจะไม่สามารถลู่เข้าได้

- จากตาราง 5-27 เมื่อใช้จำนวนเซลล์ในชั้นซ่อนที่ 2 เป็น 70,000 กับ 90,000 โครงข่ายจะไม่สามารถลู่เข้าได้

- จากตาราง 5-28 จะสามารถจำแนกได้ถูกต้องสูงสุด 83.3 เปอร์เซ็นต์ โดยใช้จำนวนเซลล์ในชั้นที่ 1 และ 2 เป็น 5,000 และ 5,000 ตามลำดับ

- จากตาราง 5-28 ความถูกต้องน้อยที่สุดที่จำแนกได้เป็น 3.3 เปอร์เซ็นต์

- จากตาราง 5-28 เมื่อใช้จำนวนเซลล์ในชั้นซ่อน 500 และ 100 ค่าความผิดพลาดไม่สามารถลู่เข้าได้

- จากตาราง 5-28 เมื่อใช้จำนวนเซลล์ในชั้นซ่อนที่ 2 เป็น 70,000 กับ 90,000 ค่าความผิดพลาดไม่สามารถลู่เข้าได้เลย

บทที่ 6

วิจารณ์และสรุป

จากการทดลองใช้นิวรอลเน็ตเวอร์กแบบแพร่กลับแบบ 1 และ 2 ชั้นซ่อน ในการจำแนกลายวงจรพิมพ์ ทำให้ทราบถึงวิธีการทำงานของโครงข่ายและประสิทธิภาพในการทำงานของการจำแนกลายวงจรพิมพ์ สรุปได้ดังนี้

6.1 วิจารณ์และข้อเสนอแนะ

- ในการรับภาพควรคำนึงถึงความสว่างของแสง ต้องให้ความสว่างอย่างเพียงพอและสม่ำเสมอกันทั้งภาพ เพื่อให้ข้อมูลภาพลายวงจรพิมพ์ในการป้อนเข้านิวรอลเน็ตเวอร์กสมบูรณ์ที่สุด

- การทำ thresholding, brightness และ contrast มีความสำคัญ ซึ่งถ้าปรับได้ดีก็จะสามารถให้ผลในการจำแนกดีตามไปด้วย ถ้ามีอัลกอริทึมที่เป็นอัตโนมัติก็จะสามารถทำการจำแนกได้เร็วมากขึ้นไปอีก

6.2 สรุป

- นิวรอลเน็ตเวอร์กแบบ 2 ชั้นซ่อนจะให้ผลการจำแนกถูกต้องที่สุด

- เมื่อใช้จุดผิดพลาดในการเรียนรู้มากกว่า 15360 จุดขึ้นไป จะให้ค่าความถูกต้องในการจำแนกที่ดี

- นิวรอลเน็ตเวอร์กแบบ 1 ชั้นซ่อน เมื่อใช้จำนวนเซลล์ 10000 – 50000 เซลล์จะให้ความถูกต้องในการจำแนกสูงที่สุด

- นิวรอลเน็ตเวอร์กแบบ 2 ชั้นซ่อน เมื่อใช้จำนวนเซลล์ในชั้นซ่อนที่ 1 และ 2 คือ 5000 และ 5000 เซลล์ตามลำดับ จะให้ความถูกต้องในการจำแนกสูงที่สุด

บรรณานุกรม

บทความจากวารสาร

Iren, Valova. 1995. "Image Decomposition by Answer-in-Weights Neural Network", IEICE TRANS. INF. & SYST. (September, 1995), E79-D9

Jain, Anil K. 1996. "Object Maching Using Deformable Templates", IEEE TRANS on Pattern Analysis and Machine Intelligence, (Mar. 1996), Vol. 18, No. 3, 267-278

Kosugi, Yukio. 1997. "Definition of Artificial Neural Networks", Basis of Neural Networks, Tokyo : Interdisciplinary Graduate School of Science and Engineering Tokyo Institute of Technology, 1997, 2-15.

วิทยานิพนธ์

จรูญ ไชยนิตย์. 2542. "เครื่องตรวจขวดอัตโนมัติ (Automatic Bottle Inspecting Machine)". วิทยานิพนธ์วิศวกรรมศาสตรมหาบัณฑิต สาขาวิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์ มหาวิทยาลัยสงขลานครินทร์. (สำเนา)

Books

Craig, Lindley. 1991. Practical Image Processing In C. 1st ed. Toronto : John Wiley & Sons. Inc.,

Fausett, Laurence V. 1994. Fundamentals of Neural Networks. Englewood Cliffs, New Jersey : Prentice-Hall, Inc.,

Freeman, James A. 1994. Simulating Neural Networks. Atlanta : Addison – Wesley Publishing Company, Inc.,

Gottfriend, Byron S. 1990. Schaum's Theory and Problems of Programming with C Outline Series. Newyork : McGraw-Hill, Inc.,

Harley, R. Myler and Arthur R. Weeks. 1993. The Pocket Handbook of Imaging Processing Algorithms in C. Englewood Cliffs, New Jersey : PTR Prentice Hall, Inc.,

Ioannis, Pitas. 1993. Digital Image Processing Algorithms. Englewood Cliffs, New Jersey: Prentice-Hall, Inc.,

Kulkarni, A. D. 1994. Artificial Neural Networks for Image Understanding. Newyork : Van Nostrand Reinhold.

Marilyn, McCord Nelson and W.T. Illingworth. 1991. A Partial Guide To Neural Nets. Atlanta : Addison – Wesley Publishing Company, Inc.,

Michael, Chester. 1993. NEURAL NETWORKS A Tutorial. Englewood Cliffs, New Jersey : Prentice-Hall, Inc.,

Rimmer, Steve. 1993. Windows Bitmapped Graphics. Newyork : McGraw-Hill, Inc.,

Tager, Ranold R. and Zadeh, Lofti A. 1994. Fuzzy Sets, Neural Networks and Soft Computing. Newyork : Van Nostrand Reinhold.

Veelenturf, L.P.J. 1995. Analysis and Applications of Artificial Neural Networks. Englewood Cliffs, New Jersey : Prentice-Hall, Inc.,

มนตรี พจนารถลาวัณย์. 2540. การเขียนโปรแกรมคอมพิวเตอร์ด้วยเทอร์โบซี. กรุงเทพฯ :
บริษัทซีเอ็ดยูเคชั่นจำกัด (มหาชน)

ภาคผนวก

ภาคผนวก ก
โปรแกรมการทำงานในส่วนต่างๆ ของนิวรอลเน็ตเวอร์ก

```
#include<string.h>
#include<stdio.h>
#include<dos.h>
#include<alloc.h>
#include<conio.h>
#include<math.h>
#include<stdlib.h>
#include<stdarg.h>

        typedef struct {
        char    id[2];
        long    filesize;
        int     reserve[2];
        long    headersize;
        long    infosize;
        long    width;
        long    depth;
        int     biplanes;
        int     bits;
        long    biCompression;
        long    biSizeImage;
        long    biXPelsPerMeter;
        long    biYPelsPerMeter;
        long    biClrUsed;
        long    ClrImportant;
        } BMPHEAD;
        BMPHEAD bmp;
        char  *buffer = NULL;
        int width,depth,bytes,bits,h,i,j,k,l,count;
        int k1, k2, k3, k4, k5, k6;
        long int kk, errt, errort;
```

```
        int no_hid, no_hidt, no_hidb, no_hidup, no_help, pic, pictu;
        unsigned char * image[321];
        float image1[41][321];
        float wp, iisum, iwsum, sumk, n, error, err, alpha;
        float past, past1, past2, past3;
        float yk, tk, sk[6], del[6], delt[6][80000], dj[6][80000], bias[6];
        char buff[10];
        double pat;
        int tar[10];
        int no_file, node[7], w[7];
        int target[100] = {1, 1, 0, 0};

// Transfer Function Log Sigmoid
  double Logsig(float n);

void main(void){
        FILE *fp;           FILE *hp;
        FILE *hid1;         FILE *hid2;
        FILE *hid3;         FILE *hid4;         FILE *hid5;
        FILE *s1;           FILE *s2;           FILE *s3;
        FILE *s4;           FILE *s5;           FILE *s6;
        FILE *hidc1;        FILE *hidc2;        FILE *hidc3;
        FILE *hidc4;        FILE *hidc5;

        no_hid = 1;         /* Number of hidden layer */
        node[1] = 5;        /* Number of node in first layer */
        node[2] = 3;        /* Number of node in second layer */
        node[3] = 4;        /* Number of node in third layer */
        node[4] = 2;        /* Number of node in fourth layer */
        node[5] = 3;        /* Number of node in fifth layer */
        node[6] = 1;        /* Number of node in sixth layer */
```

```
        pic = 3;                    /* Number image to train */

clrscr();
for (pictu=0;  pictu<pic;  pictu++){
        // Open Image file from *.BMP
        printf("%s",listfile[pictu]);
        if((fp = fopen(listfile[pictu],"rb")) == NULL){
                printf("error in open file\n");
                exit(1);
        }
        fread(&bmp,sizeof bmp,1,fp);
        width  =  (int)bmp.width;
        depth  =  (int)bmp.depth;

        printf("\nwidth = %5d : depth = %5d header_size = %5d\n", width, depth, bmp.headersize);
        printf("\nbits  = %5d : bitplans = %5d\n", bmp.bits, bmp.biplanes);

        fseek(fp,(long)bmp.headersize,SEEK_SET);

        /* OPEN FILE TO COLLECT DATA OF IMAGE */
        if((s1 = fopen("c:\\lang\\turboc\\s1.nn","wb+")) == NULL){
                printf("error in open file s1\n");
                exit(1);
        }
        past  =  1;
        fwrite(&past,sizeof(float), 1, s1);
        printf("%f \n",  past);

        for (i=0;  i<depth;  i++){
          for (j=0;  j<width;  j++){
```

```
            fgetc(fp);
            past = fgetc(fp);
            printf("%f ",past);
            past = past / 255;
            printf("%f \n", past);
            fwrite(&past,sizeof(float),1,s1);
            fgetc(fp);
            fgetc(fp);
          }
        }
        fclose(fp);
        fclose(s1);

/*****************************************/
/* To insert Image Processing here */
/*****************************************/
        if((s1 = fopen("c:\\lang\\turboc\\s1.nn","rb")) == NULL){
          printf("error in open file s1\n");
          exit(1);
        }
  for (i=0; i<((width*depth)+1); i++){
        fread(&past,4,1,s1);
        ss[pictu][i] = past ;
        printf("%8.6f ", past);
        }
  fclose(s1);
} //finish for read picture and transform to array ss[pictu][i]
```

```
// start small value for initial weight by random value
randomize();

// THE No. OF WEIGHT IN 1_st HIDDEN LAYER is node[1]
        if((hid1 = fopen("c:\\lang\\turboc\\h1.nn","wb+")) == NULL){
          printf("error in open file h1\n");
          exit(1);
        }
        node[1] = (width * depth ) + 1;
        k1 = node[1] * node[2];
        for (i=0; i<k1; i++){
                past = random (100);
                past = past/100;
                fwrite(&past,sizeof(float),1,hid1);
                printf("%f \n", past);
        }
        fclose(hid1);
        if (no_hid == 1) goto train;

/* THE No. OF WEIGHT IN 2_nd HIDDEN LAYER is node[2] */
        if((hid2 = fopen("c:\\lang\\turboc\\h2.nn","wb+")) == NULL){
                printf("error in open file h2\n");
                exit(1);
        }
        k1 = (node[2] + 1) * node[3] ;
        for (i=0; i<k1; i++){
                past = random (100);
                past = past/100;
                fwrite(&past,sizeof(float),1,hid2);
        }
        fclose(hid2);
```

```
        if (no_hid == 2) goto train;

/* THE No. OF WEIGHT IN 3_rd HIDDEN LAYER is node[3] */
        if((hid3 = fopen("c:\\lang\\turboc\\h3.nn","wb+")) == NULL){
                printf("error in open file h3\n");
                exit(1);
        }
        k1 = (node[3] + 1) * node[4] ;
        for (i=0; i<k1; i++){
                past = random (100);
                past = past/100;
                fwrite(&past,sizeof(float),1,hid3);
        }
        fclose(hid3);
        if (no_hid == 3) goto train;

/* THE No. OF WEIGHT IN 4_th HIDDEN LAYER is node[4] */
        if((hid4 = fopen("c:\\lang\\turboc\\h4.nn","wb+")) == NULL){
                printf("error in open file h4\n");
                exit(1);
        }
        k1 = (node[4] + 1) * node[5] ;
        for (i=0; i<k1; i++){
                past = random (100);
                past = past/100;
                fwrite(&past,sizeof(float),1,hid4);
        }
        fclose(hid4);
        if (no_hid == 4) goto train;
```

```
/* THE No. OF WEIGHT IN 5_th HIDDEN LAYER is node[5] */
        if((hid5 = fopen("c:\\lang\\turboc\\h5.nn","wb+")) == NULL){
                printf("error in open file h5\n");
                exit(1);
        }
        k1 = (node[5] + 1) * node[6] ;
        for (i=0; i<k1; i++){
                past = random (100);
                past = past/100;
                write(&past,sizeof(float),1,hid5);
        }
        fclose(hid5);
        if (no_hid == 5) goto train;

/*****************************************/
/*  TO TRAIN NEURAL NETWORK  */
/*****************************************/
train:  printf("\nSTART TO TRAIN\n");

count = 0;
do {

 for(pictu=0; pictu<pic; pictu++){
        count += 1;

  if((hid1 = fopen("c:\\lang\\turboc\\h1.nn","rb+")) == NULL){
          printf("\nError in open file h1_train\n");
          exit(1);}
        if((s2 = fopen("c:\\lang\\turboc\\s2.nn","wb+")) == NULL){
          printf("\nError in open file\n");
          exit(1);}
```

```
  past = 1 ;
  fwrite(&past,sizeof(float),1,s2);
for (i=0; i<node[2]; i++){
  sumk = 0;
  for (j=0; j<(node[1]); j++){
    fread(&past1 , 4, 1, hid1);
    sumk += ss[pictu][j] * past1;
  }
  past = Logsig(sumk);
  fwrite(&past,sizeof(float),1,s2);
}
fclose(hid1);
fclose(s2);
if (no_hid == 1) goto back1;

if((hid2 = fopen("c:\\lang\\turboc\\h2.nn","rb+")) == NULL){
  printf("\nError in open file\n");
  exit(1);}
if((s3 = fopen("c:\\lang\\turboc\\s3.nn","wb+")) == NULL){
  printf("\nError to open file\n");
  exit(1);}
past = 1 ;
fwrite(&past,sizeof(float),1,s3);
for (i=0; i<node[3] ; i++){
  if((s2 = fopen("c:\\lang\\turboc\\s2.nn","rb+")) == NULL){
          printf("\nError to open file\n");
          exit(1); }
  sumk = 0;
  for (j=0; j<(node[2]+1); j++){
    fread(&past, 4, 1, s2);
    fread(&past1, 4, 1, hid2);
```

```
      sumk += past * past1 ;
    }
    fclose(s2);
    past = Logsig(sumk);
    fwrite(&past,sizeof(float),1,s3);
  }
  fclose(hid2);
  fclose(s3);
  if (no_hid == 2) goto back2;

  if((hid3 = fopen("c:\\lang\\turboc\\h3.nn","rb+")) == NULL){
    printf("\nError in open file\n");
    exit(1);
  }
  if((s4 = fopen("c:\\lang\\turboc\\s4.nn","wb+")) == NULL){
    printf("\nError to open file\n");
    exit(1);
  }
  past = 1 ;
  fwrite(&past,sizeof(float),1,s4);
  for (i=0; i<node[4] ; i++){
    if((s3 = fopen("c:\\lang\\turboc\\s3.nn","rb+")) == NULL){
      printf("\nError to open file\n");
      exit(1);
    }
    sumk = 0;
    for (j=0; j<(node[3]+1); j++){
      fread(&past, 4, 1, s3);
      fread(&past1, 4, 1, hid3);
      sumk += past * past1 ;
    }
```

```
    fclose(s3);
    past = Logsig(sumk);
    fwrite(&past,sizeof(float),1,s4);
  }
  fclose(hid3);
  fclose(s4);
  if (no_hid == 3) goto back3;

  if((hid4 = fopen("c:\\lang\\turboc\\h4.nn","rb+")) == NULL){
    printf("\nError in open file\n");
    exit(1);
  }
  if((s5 = fopen("c:\\lang\\turboc\\s5.nn","wb+")) == NULL){
    printf("\nError to open file\n");
    exit(1);
  }
  past = 1 ;
  fwrite(&past,sizeof(float),1,s5);
  for (i=0; i<node[5] ; i++){
    if((s4 = fopen("c:\\lang\\turboc\\s4.nn","rb+")) == NULL){
      printf("\nError to open file\n");
      exit(1);
}
    sumk = 0;
    for (j=0; j<(node[4]+1); j++){
      fread(&past, 4, 1, s4);
      fread(&past1, 4, 1, hid4);
      sumk += past * past1 ;
    }
    fclose(s4);
    past = Logsig(sumk);
```

```
    fwrite(&past,sizeof(float),1,s5);
  }
  fclose(hid4);
  fclose(s5);
  if (no_hid == 4) goto back4;
  if((hid5 = fopen("c:\\lang\\turboc\\h5.nn","rb+")) == NULL){
    printf("\nError in open file\n");
    exit(1);
  }
  if((s6 = fopen("c:\\lang\\turboc\\s6.nn","wb+")) == NULL){
    printf("\nError to open file\n");
    exit(1);
  }
  for (i=0; i<node[6] ; i++){
    if((s5 = fopen("c:\\lang\\turboc\\s5.nn","rb+")) == NULL){
      printf("\nError to open file\n");
      exit(1);
}
    sumk = 0;
    for (j=0; j<(node[5]+1); j++){
      fread(&past, 4, 1, s5);
      fread(&past1 ,4, 1, hid5);
      sumk += past * past1;
    }
    fclose(s5);
    past = Logsig(sumk);
    fwrite(&past,sizeof(float),1,s6);
  }
  fclose(hid5);
  fclose(s6);
  if (no_hid == 5) goto back5;
```

```
/*************************************/
/*       BACK PROPAGATION       */
/*************************************/
/*************************************/
/*   Back propagation 3 layers    */
/*************************************/
back3:

          if((s4 = fopen("c:\\lang\\turboc\\s4.nn","rb")) == NULL){
            printf("\nError to open file 1\n");
            exit(1);}
          if((hidc3 = fopen("c:\\lang\\turboc\\hc3.nn","wb+")) == NULL){
            printf("\nError to open file 3\n");
            exit(1);}
  fread(&past, 4, 1, s4);
          err = target[pictu] - past;
          del[4] = err * past * (1 - past);
          for (k=1; k<=node[4]; k++){
            if((s3 = fopen("c:\\lang\\turboc\\s3.nn","rb")) == NULL){
              printf("\nError to open file s3.nn\n");
              exit(1); }
            for (j=0; j<=node[3]; j++){
              fread(&past, 4, 1, s3);
              past = alpha * del[4] * past ; /* for weight and bias layer 3 */
              fwrite(&past,sizeof(float),1,hidc3);
            }
            fclose(s3);
          }
          fclose(s4);
          fclose(hidc3);
// End change hidc3
```

```
//BPN 2

        for(j=0; j<node[3]; j++){
        if((hid3 = fopen("c:\\lang\\turboc\\h3.nn","rb")) == NULL){
          printf("\nError in open file\n");
                exit(1);   }
    fread(&past,4,(1+j),hid3);
    sumk = 0;
        for (h=0; h<node[4]; h++){
                        fread(&past,4,1,hid3);
        sumk += delt[4][h] * past;
          fseek(hid3,(4*node[3]),SEEK_CUR);
          }
    delt[3][j] = sumk;
          fclose(hid3);
 }

        if((s3 = fopen("c:\\lang\\turboc\\s3.nn","rb")) == NULL){
    printf("\nError to open file\n");
    exit(1); }
 for (j=0; j<node[3]; j++){
        fread(&past, 4, 1, s3);
                        delt[3][j] = delt[3][j] * past * (1 - past);
        }
 fclose(s3);


        if((hidc2 = fopen("c:\\lang\\turboc\\hc2.nn","wb+")) == NULL){
          printf("\nError in open file \n");
          exit(1);}
        for (h=0; h<node[3]; h++){
```

```
        if((s2 = fopen("c:\\lang\\turboc\\s2.nn","rb")) == NULL){
          printf("\nError to open file\n");
          exit(1);}
        for (j=0; j<=node[2]; j++){
                    fread(&past,4,1,s2);
                    past1 = alpha * delt[3][h] * past;
                  fwrite(&past1,sizeof(float),1,hidc2);
    }
    fclose(s2);
        }
        fclose(hidc3);
// End change hidc2

//BPN 1

        for(j=0; j<node[2]; j++){
        if((hid2 = fopen("c:\\lang\\turboc\\h2.nn","rb")) == NULL){
          printf("\nError in open file\n");
                exit(1);   }
    fread(&past,4,(1+j),hid2);
    sumk = 0;
        for (h=0; h<node[3]; h++){
                          fread(&past,4,1,hid2);
        sumk += delt[3][h] * past;
          fseek(hid2,(4*node[2]),SEEK_CUR);
          }
    delt[2][j] = sumk;
          fclose(hid2);
  }

        if((s2 = fopen("c:\\lang\\turboc\\s2.nn","rb")) == NULL){
```

```
    printf("\nError to open file\n");
    exit(1); }
  for (j=0; j<node[2]; j++){
        fread(&past, 4, 1, s2);
                        delt[2][j] = delt[2][j] * past * (1 - past);
        }
  fclose(s2);

        if((hidc1 = fopen("c:\\lang\\turboc\\hc1.nn","wb+")) == NULL){
          printf("\nError in open file \n");
          exit(1);}
        for (h=0; h<node[2]; h++){
                if((s1 = fopen("c:\\lang\\turboc\\s1.nn","rb")) == NULL){
                printf("\nError to open file\n");
                exit(1);   }
        for (j=0; j<=node[1]; j++){
                        fread(&past, 4, 1, s1);
                        past1 = alpha * delt[3][h] * past;
                fwrite(&past1,sizeof(float),1,hidc1);
    }
        fclose(s1);
        }
        fclose(hidc1);
  goto Upweight;
  // End change hidc1

/*************************************/
/*   Back propagation 2 layers    */
/*************************************/

back2:
```

```
        if((s3 = fopen("c:\\lang\\turboc\\s3.nn","rb")) == NULL){
          printf("\nError to open file 1\n");
          exit(1);}
        if((hidc2 = fopen("c:\\lang\\turboc\\hc2.nn","wb+")) == NULL){
          printf("\nError to open file 3\n");
          exit(1);}
fread(&past, 4, 1, s3);
        err = target[pictu] - past;
        del[3] = err * past * (1 - past);

        for (k=1; k<=node[3]; k++){
          if((s2 = fopen("c:\\lang\\turboc\\s2.nn","rb")) == NULL){
            printf("\nError to open file s2.nn\n");
            exit(1); }
          for (j=0; j<=node[2]; j++){
            fread(&past, 4, 1, s2);
            past = alpha * del[3] * past ; /* for weight and bias layer 2 */
            fwrite(&past,sizeof(float),1,hidc2);
          }
          fclose(s2);
        }
        fclose(s3);
        fclose(hidc2);
// End change hidc2

//BPN 1

        for(j=0; j<node[2]; j++){
        if((hid2 = fopen("c:\\lang\\turboc\\h2.nn","rb")) == NULL){
          printf("\nError in open file\n");
                exit(1);   }
```

```
  fread(&past,4,(1+j),hid2);
  sumk = 0;
        for (h=0; h<node[3]; h++){
                        fread(&past,4,1,hid2);
        sumk += delt[3][h] * past;
          fseek(hid2,(4*node[2]),SEEK_CUR);
          }
  delt[2][j] = sumk;
          fclose(hid2);
}

        if((s2 = fopen("c:\\lang\\turboc\\s2.nn","rb")) == NULL){
  printf("\nError to open file\n");
  exit(1); }
for (j=0; j<node[2]; j++){
        fread(&past, 4, 1, s2);
                        delt[2][j] = delt[2][j] * past * (1 - past);
        }
fclose(s2);

        if((hidc1 = fopen("c:\\lang\\turboc\\hc1.nn","wb+")) == NULL){
          printf("\nError in open file \n");
          exit(1);}
        for (h=0; h<node[2]; h++){
                if((s1 = fopen("c:\\lang\\turboc\\s1.nn","rb")) == NULL){
                printf("\nError to open file\n");
                exit(1);   }
        for (j=0; j<=node[1]; j++){
                        fread(&past, 4, 1, s1);
                        past1 = alpha * delt[3][h] * past;
                fwrite(&past1,sizeof(float),1,hidc1);
```

```
    }
        fclose(s1);
        }
        fclose(hidc1);
  goto Upweight;
// End change hidc1

/*************************************/
/*   Back propagation 1 layer    */
/*************************************/
back1:

        if((s2 = fopen("c:\\lang\\turboc\\s2.nn","rb")) == NULL){
          printf("\nError to open file \n");
          exit(1);}
        if((hidc1 = fopen("c:\\lang\\turboc\\hc1.nn","wb+")) == NULL){
          printf("\nError to open file \n");
          exit(1);}
  fread(&past, 4, 1, s2);
        err = target[pictu] - past;
        del[2] = err * past * (1 - past);

        for (k=1; k<=node[2]; k++){
          if((s1 = fopen("c:\\lang\\turboc\\s1.nn","rb")) == NULL){
            printf("\nError to open file s1.nn\n"); exit(1); }
          for (j=0; j<=node[1]; j++){
            fread(&past, 4, 1, s1);
            past = alpha * del[2] * past ; /* for weight and bias layer 1 */
            fwrite(&past,sizeof(float),1,hidc1);
          }
          fclose(s1);
```

```
        }
        fclose(s2);
        fclose(hidc1);
  goto Upweight;
// End change hidc1

/*********************************************************/
/*    FEED FORWARD TO UPDATE WEIGHT        */
/*********************************************************/

Upweight:

/* UPDATE WEIGHT LAYER 1 */
        hid1 = fopen("c:\\lang\\turboc\\h1.nn","rb+");
        hidc1 = fopen("c:\\lang\\turboc\\hc1.nn","rb");
        hp = fopen("c:\\lang\\turboc\\hp.nn","wb+");
        for (i=0; i<(node[1]*node[2]); i++){
                fread(&past,4,1,hidc1);
                fread(&past1,4,1,hid1);
                past1 = past1 + past;
                fwrite(&past1,sizeof(float),1,hp);
        }
        fclose(hidc1);
        fclose(hid1);
        fclose(hp);
        hid1 = fopen("c:\\lang\\turboc\\h1.nn","wb+");
        hp = fopen("c:\\lang\\turboc\\hp.nn","rb+");
        for (i=0; i<((node[1]+1)*node[2]); i++){
                fread(&past,4,1,hp);
                past1 = past;
                fwrite(&past1,sizeof(float),1,hid1);
```

```
        }
        fclose(hid1);
        fclose(hp);
        if (no_hid == 1) goto final;

/* UPDATE WEIGHT LAYER 2 */
        hidc2 = fopen("c:\\lang\\turboc\\hc2.nn","rb");
        hid2 = fopen("c:\\lang\\turboc\\h2.nn","rb+");
        hp = fopen("c:\\lang\\turboc\\hp.nn","wb+");
        for (i=0; i<((node[2]+1)*node[3]); i++){
                fread(&past,4,1,hidc2);
                fread(&past1,4,1,hid2);
                past1 = past1 + past;
                fwrite(&past1,sizeof(float),1,hp);
        }
        fclose(hidc2);
        fclose(hid2);
        fclose(hp);
        hid2 = fopen("c:\\lang\\turboc\\h2.nn","wb+");
        hp = fopen("c:\\lang\\turboc\\hp.nn","rb+");
        for (i=0; i<((node[2]+1)*node[3]); i++){
                fread(&past,4,1,hp);
                past1 = past;
                fwrite(&past1,sizeof(float),1,hid2);
        }
        fclose(hid2);
        fclose(hp);
        if (no_hid == 2) goto final;

/* UPDATE WEIGHT LAYER 3 */
        hidc3 = fopen("c:\\lang\\turboc\\hc3.nn","rb");
```

```
        hid3 = fopen("c:\\lang\\turboc\\h3.nn","rb+");
        hp = fopen("c:\\lang\\turboc\\hp.nn","wb+");
        for (i=0; i<((node[3]+1)*node[4]); i++){
                fread(&past,4,1,hidc3);
                fread(&past1,4,1,hid3);
                past1 = past1 + past;
                fwrite(&past1,sizeof(float),1,hp);
        }
        fclose(hidc3);
        fclose(hid3);
        fclose(hp);
        hid3 = fopen("c:\\lang\\turboc\\h3.nn","wb+");
        hp = fopen("c:\\lang\\turboc\\hp.nn","rb+");
        for (i=0; i<((node[3]+1)*node[4]); i++){
                fread(&past,4,1,hp);
                past1 = past;
                fwrite(&past1,sizeof(float),1,hid3);
        }
        fclose(hid3);
        fclose(hp);
        if (no_hid == 3) goto final;

final:
printf("\nround =  %d    err = %f ",count,err);
 }

} while (err > error);
printf("\n%d",count);
getch();
}               /* THE END MAIN OF FILE */
```

SR=NL;BST;


```
/* Transfer function Logsig */

double Logsig(float n)
{
return(1 /(1+exp(-n)));
}
```

ภาคผนวก ข
ศัพท์บัญญัติ

ศัพท์บัญญัติ

ภาษาอังกฤษ		ภาษาไทย
axon	=>	ส่วนที่อยู่ระหว่าง cell body กับ synaptic transmission ดังภาพประกอบ 3-1 หน้า 13
backpropagation	=>	การแพร่กลับ
brightness correction	=>	การปรับค่าความสว่างภาพในภาพ
chain rule	=>	กฎลูกโซ่
contrast correction	=>	การปรับค่าความต่างกันภาพในภาพ
dendrites	=>	ส่วนที่รับสัญญาณเข้าของเซลล์ประสาท
feed forward	=>	การส่งค่าไปข้างหน้า
gray scall	=>	ระดับเทา
hidden layer	=>	ชั้นซ่อน
histogram	=>	การแสดงจำนวนพิกเซลว่าที่ระดับเทาค่าหนึ่งมีจำนวนกี่พิกเซล
input layer	=>	ชั้นที่รับข้อมูลเข้า
interconnection	=>	การเชื่อมต่อกันภายใน
matrix	=>	เมตริกซ์
neural network	=>	นิวรอลเน็ตเวอร์ก, โครงข่ายประสาท
neuron, nerve cells	=>	นิวรอน หรือ เซลล์ประสาท
output layer	=>	ชั้นที่ส่งข้อมูลออก
pixel	=>	จุดข้อมูลภาพ
segmentation	=>	การแบ่งภาพ
synapse	=>	ส่วนที่เป็นรอยต่อระหว่างเซลล์ประสาทสองเซลล์
transfer function	=>	ฟังก์ชั่นโอนย้าย

ประวัติผู้เขียน

ชื่อ นายยุทธชัย นิลแพทย์

วัน เดือน ปี เกิด 12 กันยายน 2515

วุฒิการศึกษา

วุฒิ	ชื่อสถาบัน	ปีที่สำเร็จการศึกษา
วิศวกรรมศาสตรบัณฑิต (สาขาวิศวกรรมอิเล็กทรอนิกส์)	มหาวิทยาลัยเทคโนโลยีมหานคร	2536