สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

การจำลองการทำงานโครงข่ายประสาทเทียม

ARTIFICIAL NEURAL NETWORK SIMULATION



นายสุทธิชัย มณีรัตนรุ่งโรจน์

MR.SUTTICHAI MANIRATRONGROJ

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต

สาขาวิชาวิทยาการคอมพิวเตอร์และเทคโนโลยีสารสนเทศ

บัณฑิตวิทยาลัย

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เลขหมู่.....
เลขทะเบียน.....27266
วัน, เดือน, ปี.....18 มี.ค. 2540

พ.ศ.2539

ISBN 974-621-780-1

ลิขสิทธิ์ของบัณฑิตวิทยาลัย สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ARTIFICIAL NEURAL NETWORK SIMULATION

SUTTICHAI MANIRATRONGROJ

A THESIS SUBMITTED IN PARTIAL FULFILLMENT

OF THE REQUIREMENT FOR THE DEGREE

MASTER OF SCIENCE PROGRAM IN COMPUTER SCIENCE

AND INFORMATION TECHNOLOGY

SCHOOL OF GRADUATE STUDIES

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

1996

ISBN 974-621-780-1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อวิทยานิพนธ์	การจำลองการทำงานโครงข่ายประสาทเทียม
ชื่อนักศึกษา	นายสุทธิชัย มณีรัตนรุ่งโรจน์
อาจารย์ผู้ควบคุมวิทยานิพนธ์	อาจารย์สุรสิทธิ์ วรรณไกรโรจน์
ระดับการศึกษา	วิทยาศาสตรมหาบัณฑิต สาขาวิชาวิทยาการคอมพิวเตอร์ และเทคโนโลยีสารสนเทศ
ภาควิชา	คณิตศาสตร์และวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
พ.ศ.	2539

บทคัดย่อ

ผลงานวิจัยนี้ได้กล่าวถึงทฤษฎีโครงข่ายประสาทเทียมสองรูปแบบคือทฤษฎีอะดาไลน์ที่กล่าวถึงการจำลองการทำงานของเซลล์ประสาทเดี่ยวและทฤษฎีแพร่ย้อนกลับที่ได้นำเซลล์ประสาทหลายๆเซลล์มาเชื่อมโยงเป็นโครงข่าย พร้อมกฎเกณฑ์การเรียนรู้ทางคณิตศาสตร์ นอกจากนี้ยังได้จำลองการทำงานตามทฤษฎีทั้งสองด้วยโปรแกรมคอมพิวเตอร์ที่สามารถแสดงผลการทำงานด้วยภาพกราฟิก พร้อมทั้งจัดสร้างฟังก์ชั่นไลบรารี่เพื่อให้ผู้ใช้สามารถสร้างโปรแกรมประยุกต์โครงข่ายประสาทเทียมกับโครงงานอื่นๆ ตามต้องการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Thesis Title	Artificial Neural Network Simulation
Student	Mr. Suttichai Maniratrongroj
Thesis Advisor	Mr. Surasit Vannakrairojn
Level of study	Master of Science Program in Information Technology
Department	Mathematics and Computer Science Department Faculty of Science King Mongkut' s Institute of Technology Ladkrabang
Year	1996

ABSTRACT

This research is about two neural network theories. One is Adaline theory which is about the simulation of single neuron cell. The other is about Back-propagation theory which is the linking of neuron cell to form the network, including mathematics learning rules. Both theories are then simulated by computer program to show graphical results. Futhermore, this research has built some library functions for the user to create their own application program in neural networks as required.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ในการทำวิทยานิพนธ์ฉบับนี้ ผู้เขียนขอกราบขอบพระคุณ บิดา มารดา ผู้ให้กำเนิดอีกทั้งยังให้การสนับสนุน เป็นกำลังใจและยังมอบอุปกรณ์ที่จำเป็นสำหรับการศึกษา

ขอกราบขอบพระคุณอาจารย์สุรสิทธ์ วรรณไกรโรจน์ ซึ่งเป็นอาจารย์ผู้ควบคุมการวิจัย ที่กรุณาให้ข้อคิด คำปรึกษา คำแนะนำ ตลอดจนความรู้ที่เป็นประโยชน์ และได้กรุณาตรวจสอบแก้ไขวิทยานิพนธ์จนสำเร็จเรียบร้อย

สุดท้ายนี้ผู้วิจัยขอขอบพระคุณสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ที่ได้มอบทุนอุดหนุนการวิจัยจนวิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงลงด้วยดี

สุทธิชัย มณีรัตนรุ่งโรจน์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

หน้า

บทคัดย่อภาษาไทย.....I

บทคัดย่อภาษาอังกฤษ.....II

กิตติกรรมประกาศ.....III

สารบัญ.....IV

สารบัญภาพ.....VII

คัดย่อ/คำนิยามของศัพท์ที่ใช้.....XII

บทที่

1 บทนำ.....1

วัตถุประสงค์ของการวิจัย.....1

แนวความคิดที่นำมาใช้.....2

ขอบเขตของการวิจัย.....3

เนื้อหาของวิทยานิพนธ์.....4

2 ทฤษฎีโครงข่ายประสาทเทียม.....5

การทำงานของเซลล์ประสาทของสิ่งมีชีวิต.....7

สถาปัตยกรรมพื้นฐานของโครงข่ายประสาทเทียม.....8

ความสามารถในการจัดกลุ่ม.....9

ทฤษฎีอะดาไลน์และมาดาไลน์.....11

สถาปัตยกรรมของอะดาไลน์.....11

ทฤษฎีการคำนวณของอะดาไลน์.....12

โครงสร้างของมาดาไลน์.....14

ทฤษฎีหลักการแพร่ย้อนกลับ.....16

สถาปัตยกรรมของโครงข่ายบีพีเอ็น.....16

การฝึกหัดโครงข่ายด้วยวิธีการแพร่ย้อนกลับ.....18

การหาค่าผิดพลาดและการปรับน้ำหนักที่ชั้นเอาต์พุท.....18

การหาค่าผิดพลาดและการปรับน้ำหนักที่ชั้นซ่อน.....20

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

บทที่		หน้า
	สรุปขั้นตอนการฝึกหัดโครงข่าประสาทเทียมโดยวิธีการแพร่ย้อนกลับ	21
	หัวข้ออื่นๆของหลักการแพร่ย้อนกลับ	22
	การเพิ่มเทอมโมเมนตัม	22
	ฟังก์ชั่นกระตุ้น	23
	การกำหนดค่าน้ำหนักเริ่มต้น	24
	การกำหนดจำนวนชั้นซ่อน	26
	บทสรุป	26
3	การออกแบบโปรแกรมจำลองโครงข่ายประสาทเทียม	27
	การออกแบบโปรแกรมจำลองอะดาไลน์	27
	โครงสร้างหลักของโปรแกรม	27
	โครงสร้างแบบจำลองอะดาไลน์	28
	อัลกอริธึมในการสร้างส่วนจำลองอะดาไลน์	29
	อัลกอริธึม calculate net	31
	อัลกอริธึม calculate output	31
	อัลกอริธึม update weights	32
	การออกแบบโปรแกรมจำลองโครงข่ายบีพีเอ็น	33
	ระบบปฏิบัติการและภาษาที่เลือกใช้	33
	โครงสร้างหลักของโปรแกรม	34
	โครงสร้างแบบจำลองโครงข่ายประสาทเทียม	40
	อัลกอริธึมในการสร้างส่วนจำลองการทำงานโครงข่ายประสาทเทียม	42
	อัลกอริธึม forward propagate	44
	อัลกอริธึม backward propagate	44
	อัลกอริธึม update weights	46
	บทสรุป	47
4	การใช้งานโปรแกรมจำลองโครงข่ายประสาทเทียม	48
	การใช้งานโปรแกรมจำลองอะดาไลน์	48
	การฝึกหัดอะดาไลน์	48

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

บทที่		หน้า
	การทดสอบหาผลลัพธ์ทางเอาต์พุทของหน่วยอะดาไลน์	55
	การใช้งานโปรแกรมจำลองการทำงานโครงข่ายประสาทเทียมแบบแพร่ย้อนกลับ	59
	การฝึกหัดโครงข่ายประสาทเทียม	59
	การทดสอบหาผลลัพธ์ทางเอาต์พุทโครงข่ายประสาทเทียม	74
	ส่วนเพิ่มเติมของโปรแกรม	84
	บทสรุป	85
	ข้อจำกัดของโปรแกรมจำลองอะดาไลน์	85
	ข้อจำกัดของโปรแกรมจำลองโครงข่ายประสาทเทียมโดยหลักการแพร่ย้อนกลับ	85
5	การทดลอง	86
	การประยุกต์ใช้กับการรู้จำรูปแบบอักขระ	86
	การฝึกหัดโครงข่ายประสาทเทียม	86
	การทดสอบโครงข่ายประสาทเทียม	88
	ฟังก์ชั่นไลบรารี่	88
	การประยุกต์ใช้กับการรู้จำรูปทรงเรขาคณิต	111
	การฝึกหัดโครงข่ายประสาทเทียม	111
	การทดสอบโครงข่ายประสาทเทียม	113
	บทสรุป	125
6	สรุปผลการวิจัยและข้อเสนอแนะ	126
	เอกสารอ้างอิง	128
	ภาคผนวก	129
	แฟ้มข้อมูล CHAR.TRN	129
	แฟ้มข้อมูล CHAR.TES	138
	แฟ้มข้อมูล GEOME.TRN	147
	แฟ้มข้อมูล GEOME.TES	150
	บทความที่ได้รับการตีพิมพ์	155
	ประวัติผู้เขียน	166

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

สารบัญภาพ

หน้า

1 แสดงการฝึกหัดโครงข่ายโดยอาศัยโปรแกรมจำลองโครงข่ายประสาทเทียม.....2
2 แสดงการประยุกต์ใช้งานโปรแกรมโครงงานของผู้ใช้กับฟังก์ชั่นไลบรารี่.....3
3 โครงสร้างและส่วนประกอบของเซลล์ประสาทโดยทั่วไป.....7
4 สถาปัตยกรรมพื้นฐานของโครงข่ายประสาทเทียม.....9
5 แสดงกราฟแบ่งกลุ่มเชิงเส้นและไม่เป็นเชิงเส้น.....11
6 แสดงหน่วยอะดาไลน์.....12
7 แสดงโครงสร้างมาดาไลน์.....14
8 แสดงลักษณะของกราฟที่มาดาไลน์สามารถจัดกลุ่มตรรก XOR ได้.....15
9 แสดงโครงข่ายบีพีเอ็น.....17
10 แสดงกราฟในลักษณะ S-shape ของฟังก์ชั่นซิกมอยด์.....20
11 แสดงกราฟของฟังก์ชั่นไบนารี่ซิกมอยด์.....23
12 แสดงกราฟของฟังก์ชั่นไบโพล่าซิกมอยด์.....24
13 แสดงแผนภูมิโครงสร้างหลักของโปรแกรม.....28
14 แสดงโครงสร้างแบบจำลองอะดาไลน์.....29
15 แสดงลำดับชั้นของคลาส.....35
16 แสดงการทำงานโดยอาศัยเทคนิคเอ็มดีไอ.....36
17 แสดงส่วนประกอบหลักของโปรแกรมที่เป็นเอ็มดีไอ.....37
18 แสดงโครงสร้างแบบจำลองโครงข่ายประสาทเทียม.....41
19 การแสดงผลเริ่มต้นของโปรแกรม.....49
20 แสดงการเปิดแฟ้มข้อมูลฝึกหัด.....49
21 ใส่ชื่อแฟ้มข้อมูลฝึกหัด.....50
22 แสดงหน่วยอะดาไลน์ที่โปรแกรมสร้างขึ้น.....50
23 แสดงการเลือกฝึกหัดอะดาไลน์.....51
24 แสดงค่าต่างๆเปลี่ยนแปลงขณะทำการฝึกหัด.....51
25 แสดงการเลือกจัดเก็บค่าน้ำหนัก.....52
26 ใส่ชื่อแฟ้มข้อมูลน้ำหนัก.....52
27 แสดงการเลือกดูรายละเอียดทฤษฎีอะดาไลน์.....53

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ(ต่อ)

หน้า

28 แสดงรายละเอียดทฤษฎีอะดาไลน์ 53
29 แสดงการเลือกดูรายละเอียดรูปแบบแฟ้มข้อมูล 54
30 แสดงรายละเอียดรูปแบบแฟ้มข้อมูล 54
31 แสดงการเลือกออกจากโปรแกรม 55
32 การแสดงผลเริ่มต้นของโปรแกรม 56
33 แสดงการเปิดแฟ้มข้อมูลทดสอบ 56
34 ใส่ชื่อแฟ้มข้อมูลทดสอบ 57
35 ใส่ชื่อแฟ้มข้อมูลน้ำหนัก 57
36 แสดงการเลือกทดสอบหน่วยอะดาไลน์ 57
37 แสดงหน่วยอะดาไลน์ที่โปรแกรมสร้างขึ้น 58
38 แสดงการเลือกออกจากโปรแกรม 58
39 การแสดงผลเริ่มต้นของโปรแกรม 62
40 แสดงการเปิดแฟ้มข้อมูลโครงสร้าง 62
41 แสดงไดอะล็อกบ๊อกเปิดแฟ้มข้อมูล 63
42 หน้าต่างลูกแสดงโครงข่าย 63
43 แสดงการเปิดแฟ้มข้อมูลฝึกหัด 64
44 แสดงไดอะล็อกบ๊อกเปิดแฟ้มข้อมูล 64
45 หน้าต่างลูกแสดงรายละเอียดการทำงาน 65
46 แสดงการเลือกฝึกหัดโครงข่าย 66
47 แสดงการเลือกเปิดหน้าต่างแสดงกราฟค่าผิดพลาด 66
48 หน้าต่างลูกแสดงระดับค่าผิดพลาด 67
49 การเลือกเปิดหน้าต่างแสดง I/O 67
50 หน้าต่างลูกแสดง I/O 68
51 การเลือกกำหนดค่าพารามิเตอร์ใหม่ 68
52 แสดงไดอะล็อกบ๊อกกำหนดพารามิเตอร์ 69
53 แสดงเมนูเลือกรูปแบบการแสดงผล 69

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ(ต่อ)

หน้า

54 แสดงการเลือกดูตรรชนีช่วยเหลือ....70
55 แสดงตรรชนีช่วยเหลือ....70
56 แสดงการเลือกดูเวอร์ชั่นของโปรแกรม....71
57 แสดงอินฟอร์เมชั่นไดอะล็อกบ๊อก....71
58 เมนูเลือกการหยุดการฝึกหัดชั่วคราว....72
59 แสดงการเลือกจัดเก็บค่าน้ำหนัก....72
60 แสดงไดอะล็อกบ๊อกจัดเก็บค่าน้ำหนัก....73
61 แสดงการเลือกออกจากโปรแกรม....73
62 การแสดงผลเริ่มต้นของโปรแกรม....75
63 แสดงการเปิดแฟ้มข้อมูลโครงสร้าง....75
64 แสดงไดอะล็อกบ๊อกเปิดแฟ้มข้อมูล....76
65 หน้าต่างลูกแสดงโครงข่าย....76
66 แสดงการเปิดแฟ้มข้อมูลทดสอบ....77
67 แสดงไดอะล็อกบ๊อกเปิดแฟ้มข้อมูล....77
68 แสดงการเปิดแฟ้มข้อมูลน้ำหนัก....78
69 แสดงไดอะล็อกบ๊อกเปิดแฟ้มข้อมูลน้ำหนัก....78
70 แสดงการเลือกทดสอบโครงข่าย....79
71 หน้าต่างลูกแสดงลำดับข้อมูลทดสอบ....79
72 หน้าต่างลูกแสดง I/O....80
73 แสดงการเลือกการทำงานแบบซิงเกิลสเตป....80
74 แสดงการเลือกลำดับข้อมูลทดสอบถัดไปเมื่ออยู่ในโหมดซิงเกิลสเตป....81
75 หน้าต่างลูกแสดงลำดับข้อมูลทดสอบ....81
76 หน้าต่างลูกแสดง I/O ที่ลำดับข้อมูลทดสอบที่ 1....82
77 แสดงการเลือกลำดับข้อมูลทดสอบถัดไป....82
78 หน้าต่างลูกแสดงลำดับข้อมูลทดสอบ....83
79 หน้าต่างลูกแสดง I/O ที่ลำดับข้อมูลทดสอบที่ 2....83
80 แสดงการทำงานของโปรแกรมตัวอย่าง....97

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ(ต่อ)

หน้า

81 แสดงผลลัพธ์ของอักขระ B....98
82 แสดงผลลัพธ์ของอักขระ C....98
83 แสดงผลลัพธ์ของอักขระ D....99
84 แสดงผลลัพธ์ของอักขระ E....99
85 แสดงผลลัพธ์ของอักขระ F....100
86 แสดงผลลัพธ์ของอักขระ G....100
87 แสดงผลลัพธ์ของอักขระ H....101
88 แสดงผลลัพธ์ของอักขระ I....101
89 แสดงผลลัพธ์ของอักขระ J....102
90 แสดงผลลัพธ์ของอักขระ K....102
91 แสดงผลลัพธ์ของอักขระ L....103
92 แสดงผลลัพธ์ของอักขระ M....103
93 แสดงผลลัพธ์ของอักขระ N....104
94 แสดงผลลัพธ์ของอักขระ O....104
95 แสดงผลลัพธ์ของอักขระ P....105
96 แสดงผลลัพธ์ของอักขระ Q....105
97 แสดงผลลัพธ์ของอักขระ R....106
98 แสดงผลลัพธ์ของอักขระ S....106
99 แสดงผลลัพธ์ของอักขระ T....107
100 แสดงผลลัพธ์ของอักขระ U....107
101 แสดงผลลัพธ์ของอักขระ V....108
102 แสดงผลลัพธ์ของอักขระ W....108
103 แสดงผลลัพธ์ของอักขระ X....109
104 แสดงผลลัพธ์ของอักขระ Y....109
105 แสดงผลลัพธ์ของอักขระ Z....110
106 แสดงผลลัพธ์ของข้อมูลทดสอบที่ 1....120
107 แสดงผลลัพธ์ของข้อมูลทดสอบที่ 2....120

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ(ต่อ)

หน้า

108 แสดงผลลัพธ์ของข้อมูลทดสอบที่ 3.....121
109 แสดงผลลัพธ์ของข้อมูลทดสอบที่ 4.....121
110 แสดงผลลัพธ์ของข้อมูลทดสอบที่ 5.....121
111 แสดงผลลัพธ์ของข้อมูลทดสอบที่ 6.....122
112 แสดงผลลัพธ์ของข้อมูลทดสอบที่ 7.....122
113 แสดงผลลัพธ์ของข้อมูลทดสอบที่ 8.....122
114 แสดงผลลัพธ์ของข้อมูลทดสอบที่ 9.....123
115 แสดงผลลัพธ์ของข้อมูลทดสอบที่ 10.....123
116 แสดงผลลัพธ์ของข้อมูลทดสอบที่ 11.....123
117 แสดงผลลัพธ์ของข้อมูลทดสอบที่ 12.....124
118 แสดงผลลัพธ์ของข้อมูลทดสอบที่ 13.....124
119 แสดงผลลัพธ์ของข้อมูลทดสอบที่ 14.....124

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คำนิยามศัพท์

การจำลอง	simulation
กฎเมเจอริตี้โวท	majority vote rule
กฎเดลต้า	delta rule
กรอบโต้ตอบ	dialog box
การโปรแกรมเชิงวัตถุ	Oject-Oriented Programming
การรู้จำรูปแบบอักขระ	character pattern recognition
โครงข่ายประสาทเทียม	artificial neural network
คำสั่งเทียม	pseudo code
โครงข่ายบีพีเอ็น	Back-Propagation Neural Network(BPN)
ไคลเอ็นท์วินโดว์	client window
เชิงเส้น	linear
ชั้นอินพุท	input layer
ชั้นซ่อน	hidden layer
ชั้นเอาต์พุท	output layer
เทรสโฮลด์	threshold
นิวรอน	neuron
น้ำหนัก	weight
แผนภูมิโครงสร้าง	structure chart
ไบนารี่ซิกมอยด์	binary sigmoid
ไบโพล่าร์ซิกมอยด์	bipolar sigmoid
ไบแอส	bias
เป้าหมาย	target
เพอร์เซพตรอน	perceptron
แพร่ย้อนกลับ	back-propagation
โพลิมอฟิค	polymorphic
พูลดาวน์เมนู	pull down menu
ฟังก์ชั่นกระตุ้น	activation function

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แฟ้มข้อมูลโครงสร้าง	structure file
แฟ้มข้อมูลฝึกหัด	training file
แฟ้มข้อมูลทดสอบ	testing file
แฟ้มข้อมูลน้ำหนัก	weight file
มาดาไลน์	madaline
ไม่เป็นเชิงเส้น	nonlinear
โมเมนตัม	momentum
โมดูล	module
ระบบปฏิบัติการ	operating system
ไลบรารี่	library
หน้าต่างลูก	child window
อะดาไลน์	adaline
อัลกอริธึม	algorithm
อัตราการเรียนรู้	learning rate
อินพุท-เอาต์พุท เวคเตอร์	input-output vector
เอ็มดีไอ	Mutiple Document Interface(MDI)
อักขระ	character

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

ปัจจุบันโครงข่ายประสาทเทียมเริ่มได้รับความสนใจที่จะนำไปใช้ในโครงงานด้านต่างๆ เช่นนำไปประยุกต์ใช้กับการจดจำรูปแบบ(pattern recognition)การประมวลผลสัญญาณภาพ(image processing) ควบคุมการเคลื่อนไหวของหุ่นยนต์(robot movement control) และอื่นๆ ซึ่งทำให้การพัฒนาศาสตร์ทางด้านโครงข่ายประสาทเทียมนี้มีความสำคัญยิ่งเพราะเป็นแนวทางหนึ่งของความก้าวหน้าทางเทคโนโลยีที่ก่อให้เกิดการประดิษฐ์เครื่องประมวลผลที่มีความชาญฉลาดโดยอาศัยความรู้เรื่องกระบวนการทำงานของเซลล์ประสาททางชีววิทยาและได้รับการพัฒนาให้อยู่ในรูปทางคณิตศาสตร์ที่สามารถจำลองการทำงานของเซลล์ประสาทได้ในระดับหนึ่ง ดังนั้นจึงเล็งเห็นประโยชน์ที่จะทำการวิจัยและสร้างโปรแกรมจำลองการทำงานของโครงข่ายประสาทเทียมเพื่อประยุกต์ใช้กับโครงงานด้านต่างๆ ข้าพเจ้าจึงได้ดำเนินงานวิจัยมาอย่างต่อเนื่องตั้งแต่บทความที่ได้จากงานวิจัยในการสัมมนา 1 และสัมมนา 2 บทความที่ได้นำเสนอในสัมมนา 1 นั้นข้าพเจ้าได้นำความรู้จากการศึกษาทฤษฎีเบื้องต้นของโครงข่ายประสาทเทียมมาสร้างเป็นโปรแกรมคอมพิวเตอร์จำลองการทำงานของระบบมาดาไลน์เพื่อใช้ในการควบคุมการเปิด-ปิด สวิตช์ไฟฟ้าอิเล็กทรอนิกส์โดยใช้ทรานซิสเตอร์ทำหน้าที่เป็นสวิตช์และใช้หลอดไฟฟ้าเป็นตัวแสดงผล ส่วนบทความที่ได้นำเสนอในสัมมนา 2 นั้น ข้าพเจ้าได้นำทฤษฎีการเรียนรู้แบบแพร่ย้อนกลับมาประยุกต์ใช้กับการรู้จำอักขระภาษาอังกฤษ ประสบการณ์ที่ได้จากการสัมมนาทั้ง 2 ครั้งที่ผ่านมาทำให้ข้าพเจ้ามีแนวคิดที่จะดำเนินงานวิจัยทางด้านโครงข่ายประสาทเทียมต่อไปโดยปรึกษาแนวทางที่จะทำวิทยานิพนธ์กับอาจารย์ สุรสิทธิ์ วรรณไกรโรจน์ ซึ่งเป็นอาจารย์ที่ปรึกษาของข้าพเจ้า ท่านก็ให้คำแนะนำและเห็นสมควรว่าน่าจะสร้างแบบจำลองโครงข่ายประสาทเทียมที่สามารถนำไปประยุกต์ใช้กับโครงงานต่างๆ ได้อย่างสะดวก ดังนั้นจึงดำเนินงานวิจัยเพื่อที่จะสร้างโปรแกรมจำลองโครงข่ายประสาทเทียมที่สามารถแสดงผลเป็นภาพกราฟิกและมีความยืดหยุ่นเหมาะสำหรับการใช้งานอีกทั้งยังสร้างโปรแกรมในลักษณะของฟังก์ชั่นไลบรารี่เพื่อให้ผู้ใช้สามารถเขียนโปรแกรมเรียกใช้ฟังก์ชั่นเหล่านี้ในการสร้างและประยุกต์โครงข่ายประสาทเทียมกับโครงงานอื่นๆ ตามต้องการ

วัตถุประสงค์ของการวิจัย

- สร้างโปรแกรมจำลองโครงข่ายประสาทเทียมที่สามารถกำหนดคุณสมบัติของโครงข่าย และมีความคล่องตัวในการใช้งาน

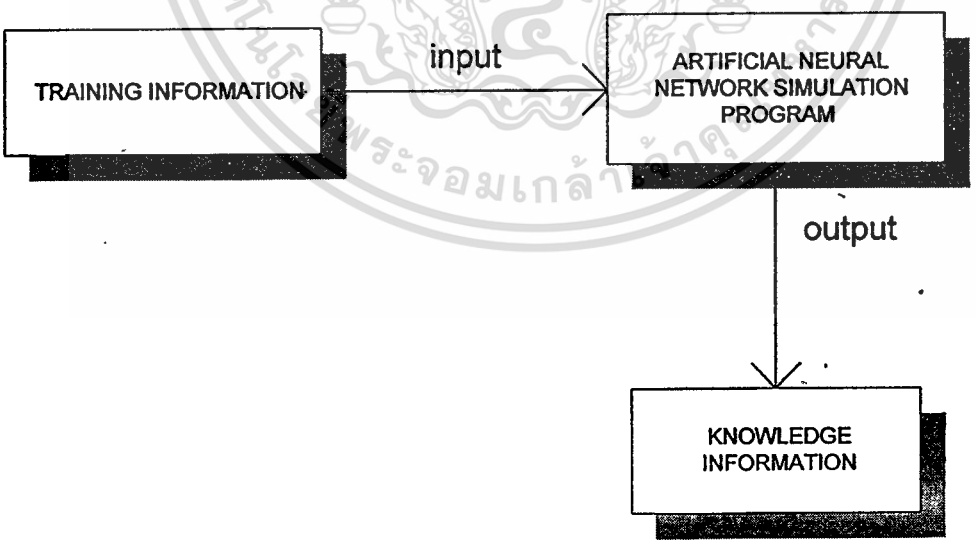
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- สร้างฟังก์ชั่นไลบรารี่ที่ผู้ใช้สามารถเขียนโปรแกรมติดต่อเพื่อเรียกใช้ฟังก์ชั่นเหล่านี้ในการประยุกต์ใช้กับโครงงานต่าง ๆ

- เผยแพร่แนวคิดและศาสตร์ทางด้านโครงข่ายประสาทเทียมซึ่งถือว่าเป็นศาสตร์ที่ค่อนข้างเป็นที่รู้จักกันน้อยในปัจจุบัน

<u>แนวความคิดที่นำมาใช้</u>

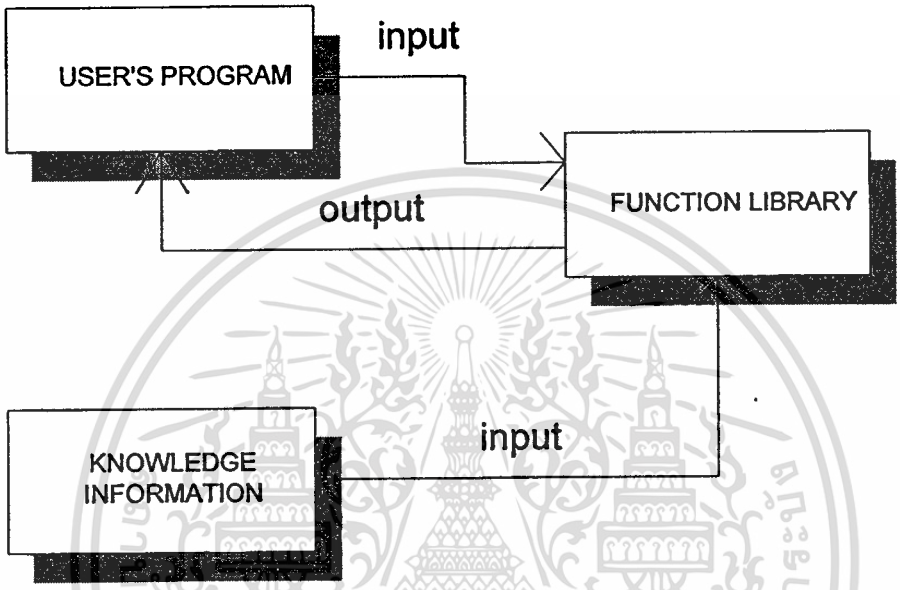
จากการวิจัยในการสัมมนา 1 และสัมมนา 2 ที่ผ่านมาได้จำลองการทำงานของโครงข่ายประสาทเทียมเพื่อใช้กับการควบคุมการเปิด-ปิดสวิตช์ไฟฟ้าอิเลคทรอนิกส์และการรู้จำอักขระภาษาอังกฤษ ในขณะนั้นได้สร้างโปรแกรมจำลองโครงข่ายประสาทเทียมเพื่อใช้เฉพาะโครงงานทำให้ไม่สะดวกในการใช้งานเพราะต้องรวมกระบวนการทุกขั้นตอนภายในโปรแกรมเดียวกัน จึงมีแนวคิดที่จะสร้างโปรแกรมจำลองโครงข่ายประสาทเทียมที่สามารถนำไปประยุกต์ใช้กับโครงงานต่างๆ โดยมีความประสงค์ที่จะลดขั้นตอนต่างๆ ทำให้เกิดความคล่องตัวในการใช้งาน โปรแกรมจำลองโครงข่ายประสาทเทียมที่สร้างขึ้นจากการวิจัยนี้แบ่งออกเป็นสองลักษณะคือ โปรแกรมจำลองโครงข่ายประสาทเทียมและโปรแกรมที่จัดทำเป็นฟังก์ชั่นไลบรารี่ ถ้าผู้ใช้ต้องการประยุกต์ใช้กับโครงงานของตนเองแล้วจะต้องใช้โปรแกรมจำลองทั้งสองลักษณะโดยแสดงการใช้งานคร่าวๆ ดังภาพที่ 1 และ ภาพที่ 2



ภาพที่ 1 แสดงการฝึกหัดโครงข่ายโดยอาศัยโปรแกรมจำลองโครงข่ายประสาทเทียม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาพที่ 1 แสดงขั้นตอนแรกของการใช้งานโปรแกรมจำลองโครงข่ายประสาทเทียม ผู้ใช้จะต้องจัดเตรียมข้อมูลฝึกหัดเพื่อใช้สอนโครงข่ายประสาทเทียมจากนั้นโปรแกรมจำลองจะให้ข้อมูลที่เป็นความรู้ซึ่งเก็บไว้เป็นแฟ้มข้อมูล



ภาพที่ 2 แสดงการประยุกต์ใช้งานโปรแกรมโครงงานของผู้ใช้กับฟังก์ชั่นไลบรารี่

จากภาพที่ 2 แสดงการสร้างโปรแกรมโครงงานที่ผู้ใช้สร้างขึ้นโดยมีการเรียกใช้ฟังก์ชั่นไลบรารี่เพื่อให้ผู้ใช้สามารถสร้างโครงข่ายประสาทเทียมได้ตามต้องการและใช้ข้อมูลความรู้ที่ได้จากการฝึกหัดโครงข่ายมาเป็นความรู้ของโครงข่ายที่ได้จัดสร้างขึ้นมาและนี่คือแนวคิดหลักของงานวิจัยนี้โดยจะได้กล่าวรายละเอียดในบทต่อๆ ไป

<u>ขอบเขตของการวิจัย</u>

- ศึกษาทฤษฎีอะดาไลน์ ซึ่งเป็นกฎเกณฑ์การเรียนรู้ของนิวรอนเดี่ยว
- ศึกษาทฤษฎีแพร่ย้อนกลับ ซึ่งเป็นกฎเกณฑ์การเรียนรู้ของโครงข่ายประสาทเทียมแบบหลายชั้น
- สร้างโปรแกรมจำลองการทำงานอะดาไลน์ โดยติดต่อกับผู้ใช้ในลักษณะของภาพกราฟิก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- สร้างโปรแกรมจำลองโครงข่ายประสาทเทียมโดยอาศัยหลักการแพร่ย้อนกลับและติดต่อกับผู้ใช้ในลักษณะของภาพกราฟิก

- สร้างฟังก์ชั่นไลบรารี่ของโครงข่ายประสาทเทียม เพื่อให้ผู้ใช้สามารถนำไปประยุกต์ใช้กับโครงงานอื่น ๆ โดยการเขียนโปรแกรมเรียกใช้งานฟังก์ชั่นเหล่านี้

- ทำการทดลองโดยนำโปรแกรมจำลองโครงข่ายประสาทเทียมที่สร้างขึ้นไปประยุกต์ใช้กับการรู้จำอักขระ

เนื้อหาของวิทยานิพนธ์

ในเอกสารวิจัยเล่มนี้กล่าวถึง บทที่ 1 กล่าวถึงวัตถุประสงค์ของการวิจัย ขอบเขตของการวิจัยและแนวคิดที่นำมาใช้ บทที่ 2 กล่าวถึงทฤษฎีการทำงานของอะดาไลน์ มาดาไลน์และหลักการแพร่ย้อนกลับ บทที่ 3 กล่าวถึงการวิเคราะห์และออกแบบโปรแกรมจำลองโครงข่ายประสาทเทียม โดยแบ่งออกเป็น 2 ส่วนคือ การวิเคราะห์และออกแบบโปรแกรมจำลองอะดาไลน์ซึ่งเป็นการจำลองการทำงานของนิวรอนเดี่ยว การวิเคราะห์และออกแบบโปรแกรมจำลองโครงข่ายประสาทเทียมโดยใช้หลักการแพร่ย้อนกลับซึ่งเป็นการจำลองนิวรอนหลาย ๆ นิวรอนเชื่อมโยงกันเป็นโครงข่าย บทที่ 4 กล่าวถึงการใช้งานโปรแกรมจำลองโครงข่ายประสาทเทียม แบ่งเป็น 2 ส่วนเช่นกันคือการใช้งานโปรแกรมจำลองอะดาไลน์และการใช้งานโปรแกรมจำลองโครงข่ายประสาทเทียมแบบแพร่ย้อนกลับโดยได้กล่าวถึงการใช้งานอย่างเป็นขั้นตอนผู้อ่านสามารถปฏิบัติตามขั้นตอนต่างๆ ได้โดยง่าย บทที่ 5 การทดลอง ได้นำโครงข่ายประสาทเทียมไปประยุกต์ใช้กับการรู้จำอักขระภาษาอังกฤษ A ถึง Z โดยเน้นถึงการเรียกใช้ฟังก์ชั่นไลบรารี่ที่มีมาให้และยังมีตัวอย่างโปรแกรมเพื่อแสดงการใช้งานฟังก์ชั่นเหล่านี้ด้วย บทที่ 6 สรุปผลการวิจัยและข้อเสนอแนะ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎีโครงข่ายประสาทเทียม

ศาสตร์ทางด้านโครงข่ายประสาทเทียมได้เริ่มมีการคิดค้นและวิจัยกันมาตั้งแต่ปี ค.ศ.1940 โดยจะขอลำดับเหตุการณ์ในช่วงเวลาที่ก่อให้เกิดสถาปัตยกรรมของโครงข่ายประสาทเทียมชนิดต่างๆ ในปี ค.ศ.1943 วอร์เรน แมคคูลล็อค(Warren McCulloch) และวอลเตอร์ พิทท์ (Walter Pitts) ได้ออกแบบนิวรอนที่ถือเป็นต้นกำเนิดและพื้นฐานของโครงข่ายประสาทเทียมในปัจจุบัน โครงข่ายของแมคคูลล็อคและพิทท์ ประกอบด้วยชั้นของโครงข่ายจำนวน 2 ชั้นคือชั้นอินพุทและชั้นเอาต์พุท โครงข่ายจะมีการเชื่อมโยงจากชั้นอินพุทไปยังชั้นเอาต์พุทโดยผ่านน้ำหนัก (weight)ซึ่งค่าน้ำหนักจะถูกกำหนดค่าไว้ตายตัว การทำงานเบื้องต้นของโครงข่ายคือนำไปใช้จำลองฟังก์ชั่นทางลอจิกพื้นฐานคือฟังก์ชั่น AND ฟังก์ชั่น OR ฟังก์ชั่น NOT และฟังก์ชั่น XOR โดยนำเสนอการกำหนดค่าน้ำหนักเพื่อให้โครงข่ายทำงานได้กับการจำลองตามฟังก์ชั่นดังกล่าว

ในปี ค.ศ.1949 โดแนลด์ เฮบบ์(Donald Hebb) ได้ออกแบบกฎการเรียนรู้สำหรับโครงข่ายประสาทเทียมเป็นคนแรก กฎการเรียนรู้นี้ก็คือขั้นตอนของกระบวนการปรับแต่งค่าน้ำหนักเพื่อทำให้เกิดค่าเอาต์พุทที่ต้องการซึ่งเป็นแนวทางในการพัฒนากฎการเรียนรู้รูปแบบอื่นๆตามมา

ในปี ค.ศ.1957 แฟรงค์ โรเซนบลัทท์(Frank Rosenblatt) ได้นำเสนอรูปแบบที่สำคัญรูปแบบหนึ่งของโครงข่ายประสาทเทียมนั่นก็คือรูปแบบที่เรียกว่าเพอร์เซพตรอน เป็นสถาปัตยกรรมอันประกอบไปด้วยชั้นอินพุทต่อผ่านน้ำหนักไปยังชั้นนิวรอนที่ชั้นเอาต์พุท และใช้กฎการเรียนรู้ของเพอร์เซพตรอนในการปรับแต่งค่าน้ำหนักที่มีประสิทธิภาพดีกว่ากฎการเรียนรู้ของเฮบบ์

ในปี ค.ศ.1960 เบอร์นาร์ด วิดโรว์(Bernard Widrow) และมาเซียน ฮอฟฟ์(Marcian Hoff) ได้พัฒนากฎการเรียนรู้ซึ่งใกล้เคียงกับการเรียนรู้ของเพอร์เซพตรอนและนำกฎนี้ไปใช้กับระบบหรือรูปแบบที่ชื่อว่าอะดาไลน์ รูปแบบมาดาไลน์คือรูปแบบที่พัฒนาจากอะดาไลน์โดยวิดโรว์เช่นกัน

ในปี ค.ศ.1972 ทูโว คาโคเนน(Teuvo Kohonen) ได้เริ่มพัฒนาการเรียนรู้แบบจัดตัวเอง (self-organized) สถาปัตยกรรมของโครงข่ายคาโคเนนนี้ประกอบด้วยจำนวนชั้น 2 ชั้น คือชั้นอินพุทและชั้นเอาต์พุท การเชื่อมโยงที่ชั้นอินพุทไปยังนิวรอนที่ชั้นเอาต์พุทนั้นจะต่อผ่านน้ำหนัก ข้อแตกต่างจากโครงข่ายอื่นๆคือกระบวนการเรียนรู้ที่สามารถจัดกลุ่มข้อมูลอินพุทได้ด้วยตัวของมันเองและเรียกการเรียนรู้นี้ว่าการเรียนรู้แบบแข่งขัน (competitive learning)

หลังจากปี ค.ศ.1980 เป็นต้นมาได้มีการพัฒนารูปแบบของโครงข่ายประสาทเทียมชนิดต่างๆเช่น รูปแบบจัดตัวเองที่เรียกว่า ART คิดค้นโดย สตีเฟน กรอสเบิร์ก(Stephen Grossberg)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และ เกียล คาร์เพนเตอร์(Gail Carpenter) รูปแบบโครงข่ายฮอปฟิลด์ คิดค้นโดย จอน ฮอปฟิลด์ (John Hopfield) นีโอคอกนิตรอน คิดค้นโดย คูนิฮิโกะ ฟูกูชิมา(Kunihiko Fukushima) รูปแบบโบลสแมนน์แมชชีน(Boltzmann machine) คิดค้นโดยนักวิจัยหลายคน และรูปแบบที่นิยมใช้กันมากคือรูปแบบแพร่ย้อนกลับ(Back-Propagation) คิดค้นโดยเดวิด ปาร์คเคอร์(David Parker)และเดวิด รูเมลฮาร์ท(David Rumelhard) ในบทนี้จะกล่าวถึงรูปแบบอะดาไลน์ มาดาไลน์และรูปแบบแพร่ย้อนกลับเท่านั้น

ตารางที่ 1 แสดงแบบจำลองโครงข่ายประสาทเทียมในช่วงปี ค.ศ. 1950-1986[3]

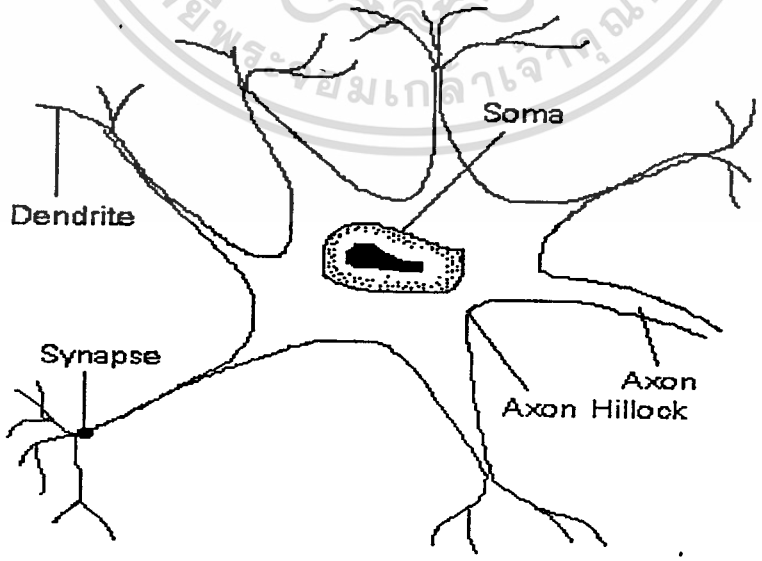
ชื่อแบบจำลอง	ผู้คิดค้น	ปีที่พัฒนา	การใช้งาน
1. Perceptron	F. Rosenblatt	1957	อ่านตัวพิมพ์ดีด
2. ADALINE และ MADALINE	B. Widrow	1960-1962	adaptive equalizer
3. Avalanche	S. Grossberg	1967	เข้าใจคำพูด
4. Cerebellatron	D. Mar, J. Albus และ A. Pellionez	1969-1982	ควบคุมการเคลื่อนไหวของแขนหุ่นยนต์
5. Back Propagation	P. Werbos, D.Parker และ D. Rumelhart	1974-1985	เลียนแบบฟังก์ชั่นทางคณิตศาสตร์, อ่านตัวเขียน, คำพูด, ทำนายหุ้น
6. Brain-State-in-a-box	J. Anderson	1977	สกัดข้อมูลบางอย่างจากฐานข้อมูล
7. Neocognitron	K. Fukushima	1978-1984	อ่านตัวเลขและตัวเขียน
8. Adaptive Resonance	G. Carpenter และ S. Grossberg	1978-1986	pattern recognition
9. Self-Organizing Map	T. Kohonen	1980	จำลองลักษณะพื้นผิวของวัตถุ เช่น ปีกเครื่องบิน

ตารางที่ 1(ต่อ)

10. Hopfield	J. Hopfield	1982	ทำนายส่วนของข้อมูลที่หายไป เช่นบางส่วนของหน้าคน และทำ optimization
11. Bidirectional Associative Memory	B. Kosko	1985	content-addressable associative memory
12. Boltzmann และ Cauchy machine	J. Hinton, T. Sejnowsky	1985-1986	pattern recognition
13. Counterpropagation	และ H. Szu R.Hencht-Nielsen	1986	อัดข้อมูลของภาพให้น้อยลง, table-lookup

การทำงานของเซลล์ประสาทของสิ่งมีชีวิต

ในธรรมชาติเซลล์ประสาทของสิ่งมีชีวิตมีหลายประเภทแล้วแต่หน้าที่ของมัน เซลล์ประสาทในตัวของคนเราก็เช่นกันมีอยู่หลายประเภทตามตำแหน่งและหน้าที่ของมันเช่นเซลล์ประสาทของกล้ามเนื้อ เซลล์ประสาทในสมอง เซลล์ประสาทที่ลิ้นและจมูกเป็นต้น โครงสร้างและส่วนประกอบของเซลล์ประสาทโดยทั่ว ๆ ไปแสดงได้ดังภาพที่ 3



ภาพที่ 3 โครงสร้างและส่วนประกอบของเซลล์ประสาทโดยทั่วไป[4]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เซลล์ประสาทประกอบด้วยส่วนใหญ่ๆ 4 ส่วนคือตัวเซลล์ประสาทหรือนิวรอน ซึ่งมีนิวเครียสอยู่ตรงกลาง รอบๆตัวเซลล์ประสาทมีสิ่งที่ยื่นออกไปเพื่อรับและส่งสัญญาณจากเซลล์ประสาทอื่นๆ สิ่งดังกล่าวเรียกว่าแอกซอน(axon) ที่ปลายกิ่งจะแตกออกเป็นก้านย่อยๆ เรียกว่าเดนดริท(dendrite)รอยต่อระหว่างก้านของเซลล์ประสาทที่ต่างกันเรียกว่าซีนแนพส์(synapse)ซึ่งสามารถเปลี่ยนค่าความต้านทานได้ตามสัญญาณที่ส่งระหว่างกันของเซลล์ประสาท การส่งสัญญาณระหว่างเซลล์ประสาททำได้โดยการถ่ายเทสารประกอบโซเดียมและโพแทสเซียม

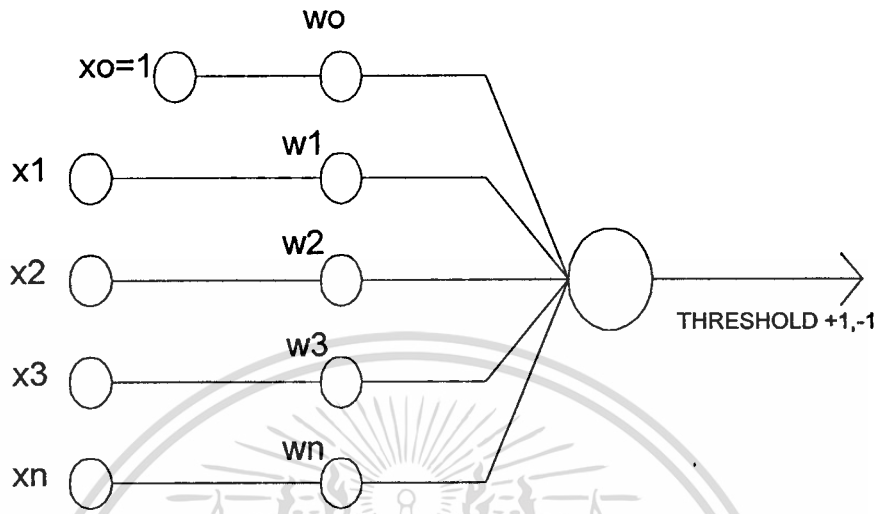
ฮอดกิ้น(Hodkin) และ ฮักลีย์(Huxley) ซึ่งได้รับรางวัลโนเบลทางชีววิทยาได้ค้นพบว่าการไหลของสารประกอบโซเดียมและโพแทสเซียมของเซลล์ประสาทของปลาหมึกได้ทำให้เกิดความต่างศักย์ จะอยู่ระหว่าง 50 ถึง 70 มิลลิโวลต์ จากผลการศึกษาดังกล่าวทำให้เราสามารถจำลองการทำงานของเซลล์ประสาทโดยอาศัยวงจรอิเลคทรอนิกส์ได้[3]

โครงสร้างของโครงข่ายประสาทเทียมมีลักษณะที่คล้ายคลึงกับสมองและระบบประสาท ซึ่งประกอบด้วยส่วนของการประมวลผลต่างๆที่เรียกว่านิวรอน(neuron) ทุกๆนิวรอนสามารถมีอินพุทได้หลายอินพุทแต่มีเอาต์พุทเพียงเอาต์พุทเดียว และทุกๆเอาต์พุทจะแยกไปยังอินพุทของนิวรอนอื่นๆ ภายในโครงข่าย การติดต่อกันภายในระหว่างนิวรอนไม่ใช่ลักษณะของการต่อแบบธรรมดา ทุกๆ อินพุทจะมีน้ำหนักเป็นตัวกำหนดกำลังของการติดต่อภายในและช่วยในการตัดสินใจ การทำงานของนิวรอนในบางโครงข่ายจะถูกกำหนดไว้ตายตัวแต่บางโครงข่ายสามารถที่จะปรับแต่งได้ ซึ่งอาจจะเป็นการปรับแต่งจากภายนอกโครงข่ายหรือนิวรอนสามารถปรับได้ด้วยตัวของมันเอง ในจุดนี้เองแสดงถึงความสามารถในการเรียนรู้และจดจำของโครงข่ายประสาทเทียม

<u>สถาปัตยกรรมพื้นฐานของโครงข่ายประสาทเทียม</u>[8]

จากลักษณะและการทำงานของเซลล์ประสาทหรือนิวรอนดังที่กล่าวมาข้างต้นได้ถูกนำมาสร้างเป็นทฤษฎีทางคณิตศาสตร์และจำลองการทำงานในรูปแบบพื้นฐานโดยใช้ชื่อว่านิวรอนเทียม หรือ AN (Artificial Neuron) ซึ่งมีสถาปัตยกรรมพื้นฐานแสดงดังภาพที่ 4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 4 สถาปัตยกรรมพื้นฐานของโครงข่ายประสาทเทียม

จากภาพที่ 4 โครงข่ายประกอบด้วยชั้นของโครงข่ายจำนวน 2 ชั้นคือชั้นอินพุทและชั้นเอาต์พุท ที่ชั้นอินพุทจะมีหน่วยอินพุทจำนวน n หน่วยที่จะคอยรับสัญญาณอินพุท x_1ถึงx_n และจะส่งสัญญาณอินพุทนี้ผ่านน้ำหนัก w_1 ถึง w_n ไปยังหน่วยเอาต์พุท ส่วนที่ชั้นเอาต์พุทจะมีหน่วยเอาต์พุทหรือหน่วยประมวลผลเพียงหน่วยเดียว

<u>ความสามารถในการจัดกลุ่ม</u>

ความสามารถในการจัดกลุ่มของ AN จะจัดกลุ่มข้อมูลอินพุทในลักษณะเชิงเส้นเดี่ยวเท่านั้นที่เป็นเช่นนี้เพราะสมการที่ใช้คำนวณเป็นสมการเชิงเส้น พิจารณาภาพที่ 4 จะได้สมการคือ

$$x_0w_0 + x_1w_1 + x_2w_2 + x_3w_3 + ... + x_nw_n = 0$$

จะเห็นว่าที่เทอม x_0w_0 เป็นเทอมที่นำมาใช้ที่บอกถึงการตั้งค่าเทรสโฮลด์ของระบบ ซึ่งเรียกเทอมนี้ว่าเทอมไบแอส โดยกำหนดให้ x_0 มีค่าคงที่เท่ากับ 1 ดังนั้นจึงเขียนสมการใหม่ได้

$$w_0 + x_1w_1 + x_2w_2 + x_3w_3 + ... + x_nw_n = 0$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ลักษณะเช่นนี้ทำให้การคำนวณค่าเอาต์พุทของ AN ที่มีเทอมไบแอสได้ดังนี้

$$f(net) = 1 \quad \text{ถ้า} \quad net >= 0$$
$$= -1 \quad \text{ถ้า} \quad net < 0$$

เมื่อ $$net = \sum_{i=1}^{n} x_i w_i + w_0$$

ถ้าตัดเทอมไบแอสออกไป นั่นหมายถึงจะต้องมีการกำหนดค่าเทรสโฮลด์ภายนอกเพื่อใช้ตัดสินการคำนวณค่าทางเอาต์พุท แสดงได้โดยสมการ

$$x_1w_1 + x_2w_2 + x_3w_3 + ... + x_nw_n = T$$

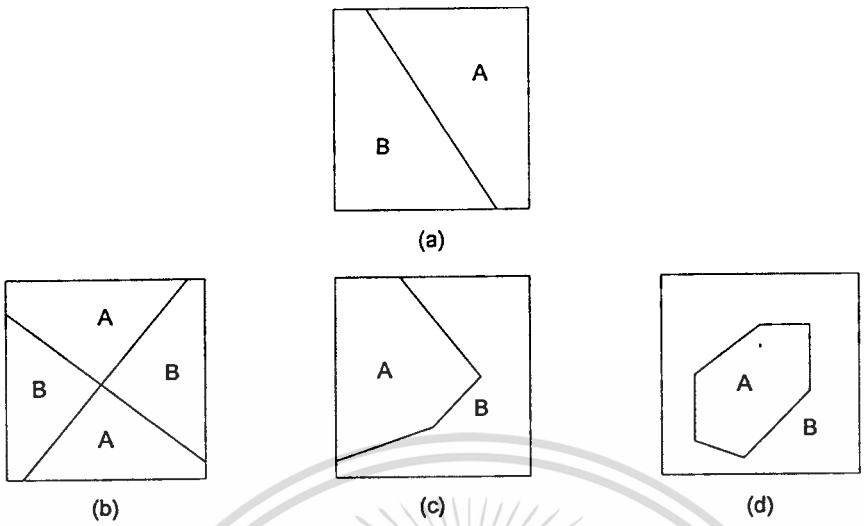
การคำนวณค่าทางเอาต์พุทของ AN ที่ไม่มีเทอมไบแอสแสดงได้ดังนี้

$$f(net) = 1 \quad \text{ถ้า} \quad net >= T$$
$$= -1 \quad \text{ถ้า} \quad net < T$$

เมื่อ $$net = \sum_{i=1}^{n} x_i w_i$$

ความสามารถในการแบ่งกลุ่มแบบเชิงเส้นเดี่ยวแสดงได้ดังภาพที่ 5(a) ส่วนภาพ 5(b) ภาพ 5(c) และภาพ 5(d) นั้นไม่ใช่การแบ่งกลุ่มแบบเชิงเส้นเดี่ยวที่ AN สามารถจะทำได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 5 ภาพ(a)แสดงกราฟแบ่งกลุ่มเชิงเส้นเดี่ยวส่วนภาพอื่นไม่เป็นกราฟแบ่งกลุ่มเชิงเส้นเดี่ยว

<u>ทฤษฎีอะดาไลน์และมาดาไลน์</u> [7]

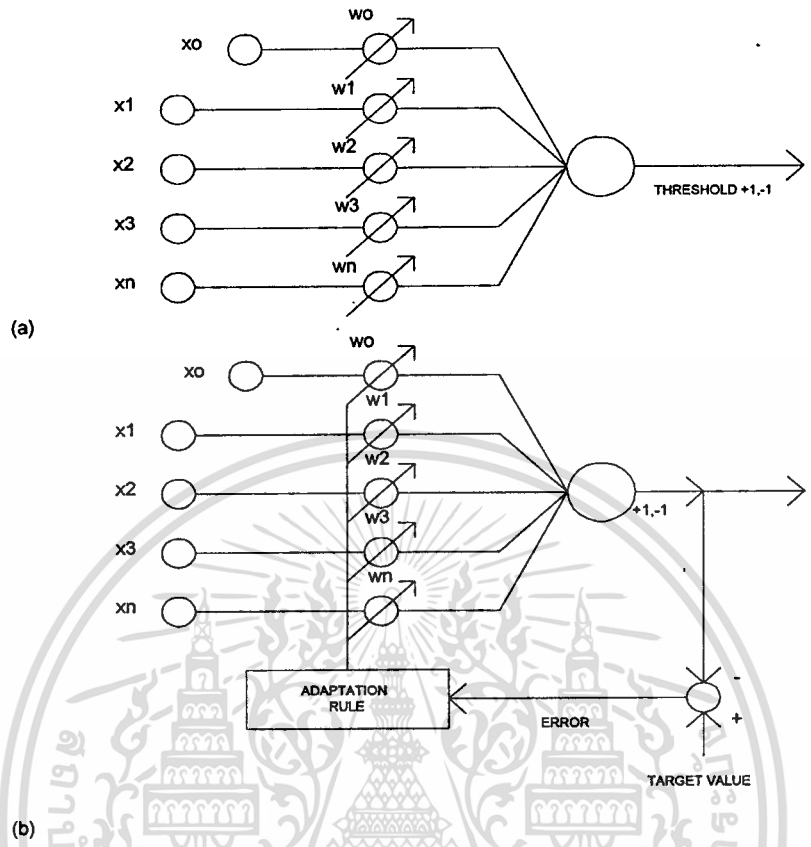
คำว่า ADALINE มาจาก <u>ADAPTIVE</u> LINEAR <u>NEURON</u> โดยมีการทำงานคล้ายกับเซลล์ประสาทเดี่ยวทางชีววิทยา (single biological nerve cell) ซึ่งคิดค้นขึ้นมาโดยวิดโรว์ ในปี ค.ศ.1960 อะดาไลน์มีลักษณะคล้ายคลึงกับเพอร์เซพตรอนที่มีการกำหนดเทรสโฮลด์เพื่อใช้กำหนดค่าเอาต์พุทและยังสามารถปรับแต่งค่าน้ำหนักได้ด้วยตัวของมันเอง

คำว่า MADALINE มาจาก <u>MULTILAYER ADALINE</u> คิดค้นขึ้นโดยวิดโรว์ เช่นกัน มาดาไลน์จะมีองค์ประกอบของอะดาไลน์ตั้งแต่ 2 หน่วยขึ้นไป อะดาไลน์แต่ละหน่วยจะส่งเอาต์พุทไปเป็นอินพุทให้กับหน่วยตัดสินใจ ดังนั้นสถาปัตยกรรมของมาดาไลน์จึงมีลักษณะเป็นโครงข่ายที่มีจำนวนชั้น 3 ชั้น

<u>สถาปัตยกรรมของอะดาไลน์</u>

ภาพ 6(a) แสดงให้เห็นถึงโครงสร้างของหน่วยอะดาไลน์ ซึ่งมีองค์ประกอบสำคัญอยู่สามส่วนคือหน่วยประมวลผลในภาพคือวงกลมทางด้านขวา หน่วยอินพุทในภาพคือวงกลมทางด้านซ้ายและน้ำหนักในภาพคือวงกลมตรงกลาง ทุกๆการเชื่อมโยงจากหน่วยอินพุทไปยังหน่วยประมวลผลจะผ่านน้ำหนัก ภาพ6(b)แสดงให้เห็นถึงโครงสร้างของหน่วยอะดาไลน์กับกลไกการปรับแต่งค่าน้ำหนักโดยการนำเอาเอาต์พุทมาเปรียบเทียบกับค่าเป้าหมาย ผลต่างของทั้งสองค่านี้คือค่าผิดพลาดซึ่งจะต้องใช้กฎการปรับแต่ง(adaptation rule) มาใช้ในการปรับแต่งค่าน้ำหนัก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 6 ภาพ(a)แสดงหน่วยอะดาไลน์ ภาพ(b)แสดงการปรับแต่งค่าน้ำหนัก

<u>ทฤษฎีการคำนวณของอะดาไลน์</u>

ทฤษฎีการคำนวณของอะดาไลน์ได้นำหลักการของสถาปัตยกรรมพื้นฐานที่ได้กล่าวผ่านมา มาใช้ในการคำนวณหาค่าเอาต์พุทของอะดาไลน์ การฝึกหัดอะดาไลน์จะอาศัยข้อมูลฝึกหัดซึ่งเป็นคู่อันดับอินพุท-เอาต์พุท ค่าเอาต์พุทนี้ก็คือค่าเป้าหมายนั่นเอง ระหว่างการฝึกหัดจะมีการป้อนอินพุทเวคเตอร์ไปยังอะดาไลน์ ซึ่งอะดาไลน์จะทำการคำนวณค่าเน็ทโดยใช้สมการ

$$net = \sum_{i=1}^{n} x_i w_i + \bar{w}_0$$

เมื่อ x_i คือ ค่าอินพุท i

w_i คือ ค่าน้ำหนักที่เชื่อมต่อจากอินพุท i

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ค่าไบแอสถูกกำหนดไว้ที่ x_0 โดยให้มีค่าเท่ากับ 1 หลังจากคำนวณค่าผลรวมจากสมการหาค่าเน็ทแล้วจะต้องทำการหาค่าเอาต์พุทโดยการกำหนดเทรสโฮลด์ โดยทั่ว ๆ ไปแล้วจะใช้กฎ

$$O_j = +1 \quad \text{if net} >= 0$$
$$= -1 \quad \text{if net} < 0$$

จากนั้นจึงหาค่าเปลี่ยนแปลงของน้ำหนักเพื่อปรับแต่งค่าน้ำหนักโดยสมการ

$$\Delta w_i = \eta x_i(\text{target} - \text{net})$$

เมื่อ η คือ อัตราการเรียนรู้ (learning rate)

x_i คือ ค่าอินพุท i

target คือ ค่าเป้าหมาย

แล้วจึงค่อยปรับแต่งค่าน้ำหนักโดยสมการ

$$w_i = w_i + \Delta w_i$$

การเรียนรู้ของอะดาไลน์ใช้ได้กับเฉพาะปัญหาเชิงเส้นที่สามารถจัดแบ่งกลุ่มแพทเทิร์นได้เพียงเส้นเดี่ยว ดังแสดงในภาพที่ 5(a) ส่วนภาพอื่นๆของภาพ 5 นั้นไม่เป็นเชิงเส้นเดี่ยวซึ่งอะดาไลน์ไม่สามารถแบ่งกลุ่มแพทเทิร์นในลักษณะนี้ได้

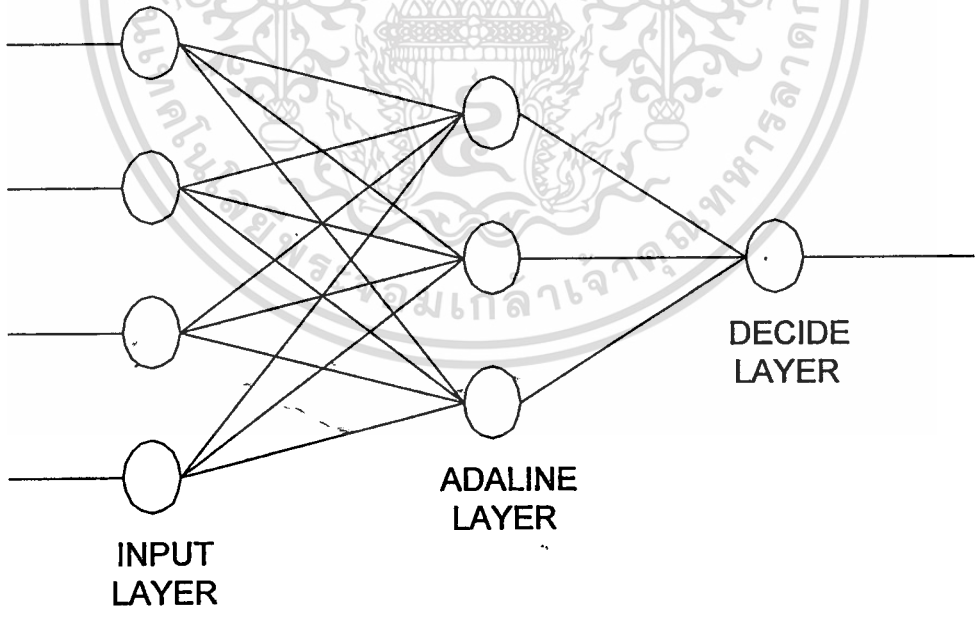
การแก้ไขปัญหาที่ไม่เป็นเชิงเส้นเดี่ยวนี้สามารถแก้ไขได้ ในบางกรณีอาจจะใช้มาดาไลน์ซึ่งยังมีขีดจำกัดอยู่แต่ก็ยังมีรูปแบบของโครงข่ายที่มีลักษณะหลายชั้นที่พัฒนาขึ้นมาเช่น โครงข่าย BPN (Back-Propagation Network)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โครงสร้างของมาดาไลน์

ระบบของมาดาไลน์จะมีชั้นของหน่วยอะดาไลน์ที่เชื่อมโยงไปยังหน่วยตัดสินใจ ที่หน่วยตัดสินใจจะใช้กฎการตัดสินใจแบบเมเจอริติ้โวท (majority vote rule) โดยการพิจารณาค่าเอาต์พุทของหน่วยอะดาไลน์เช่น ถ้าเอาต์พุทของอะดาไลน์ที่มีค่า +1 มากกว่าครึ่งหนึ่งของจำนวนเอาต์พุทของอะดาไลน์ทั้งหมดหน่วยตัดสินก็จะให้เอาต์พุทมีค่า +1 ในทางกลับกันจะให้ค่า -1

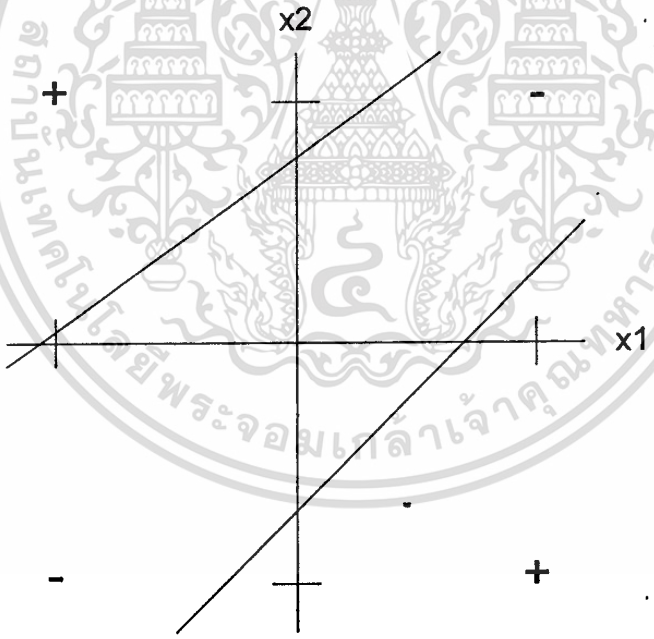
ภาพที่ 7 แสดงโครงสร้างระบบมาดาไลน์ซึ่งเป็นโครงข่ายที่มีขนาด 3 ชั้น โดยมีชั้นอินพุท ชั้นของหน่วยอะดาไลน์และชั้นของหน่วยตัดสินใจ การเชื่อมโยงในแต่ละชั้นเป็นแบบฟูลลี่ (fully) คือทุกๆนิวรอนในที่เป็นอินพุทจะส่งผ่านค่าไปยังทุกๆนิวรอนที่เป็นเอาต์พุท · การเชื่อมต่อระหว่างชั้นอินพุทและชั้นของหน่วยอะดาไลน์จะต่อผ่านน้ำหนัก แต่จะไม่มีน้ำหนักต่ออยู่ระหว่างชั้นของหน่วยอะดาไลน์กับชั้นของหน่วยตัดสินใจ ทุกๆหน่วยอะดาไลน์จะส่งผ่านค่าเอาต์พุท (+1 หรือ -1) ไปยังหน่วยตัดสินใจ ซึ่งหน่วยตัดสินใจจะใช้กฎแบบเมเจอร์ตี้โวท เพื่อกำหนดค่าเอาต์พุทของระบบมาดาไลน์



ภาพที่ 7 แสดงโครงสร้างมาดาไลน์

กระบวนการในการฝึกหัดโครงข่ายของมาดาไลน์นี้คล้ายคลึงกับอะดาไลน์ คือเมื่อกลุ่มข้อมูลทดสอบถูกป้อนเข้าสู่โครงข่ายแล้วมาดาไลน์จะให้ค่าเอาต์พุทจากการทำเมเจอร์ตี้โวทนำค่าเอาต์พุทนี้ไปเปรียบเทียบกับค่าเป้าหมายที่ต้องการจากนั้นก็เข้าสู่กระบวนการปรับแต่งค่าน้ำหนัก

กระบวนการปรับแต่งค่าน้ำหนักนี้เริ่มต้นด้วย การเปรียบเทียบค่าเอาต์พุทของมาดาไลน์กับค่าเป้าหมายซึ่งเป็นไปได้สองกรณีคือ มีค่าเท่ากัน หรือมีค่าไม่เท่ากัน ถ้าค่าเอาต์พุทกับค่าเป้าหมายมีค่าเท่ากันจะไม่มีการปรับแต่งค่าน้ำหนักแต่ถ้าไม่เท่ากันจะต้องทำการปรับแต่ง โดยจะเลือกหน่วยอะดาไลน์ ที่จะทำการปรับแต่งค่าน้ำหนักจากค่าเอาต์พุทของอะดาไลน์ที่ไม่เท่ากับค่าเป้าหมายและค่าผลรวมหรือค่าเน็ทของหน่วยอะดาไลน์นั้นเข้าใกล้ศูนย์มากที่สุด ซึ่งจะเลือกหน่วยอะดาไลน์มาเพียงหน่วยเดียว การคำนวณนี้จะใช้สมการเดียวกับอะดาไลน์ที่กล่าวผ่านมาแล้วกระบวนการเหล่านี้จะกระทำต่อเนื่องกันไปจนกระทั่งการฝึกหัดเสร็จสมบูรณ์



ภาพที่ 8 แสดงลักษณะของกราฟที่มาดาไลน์สามารถจัดกลุ่มตรรก XOR ได้

จากภาพที่ 8 แสดงให้เห็นกราฟที่มาดาไลน์สามารถสร้างเพื่อจัดกลุ่มเวคเตอร์ที่ไม่เป็นเชิงเส้นเดี่ยวได้โดยแสดงตัวอย่างกราฟแบ่งกลุ่มอินพุทที่ให้ค่าเอาต์พุทตามตรรก XOR

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

<u>ทฤษฎีหลักการแพร่ย้อนกลับ</u>

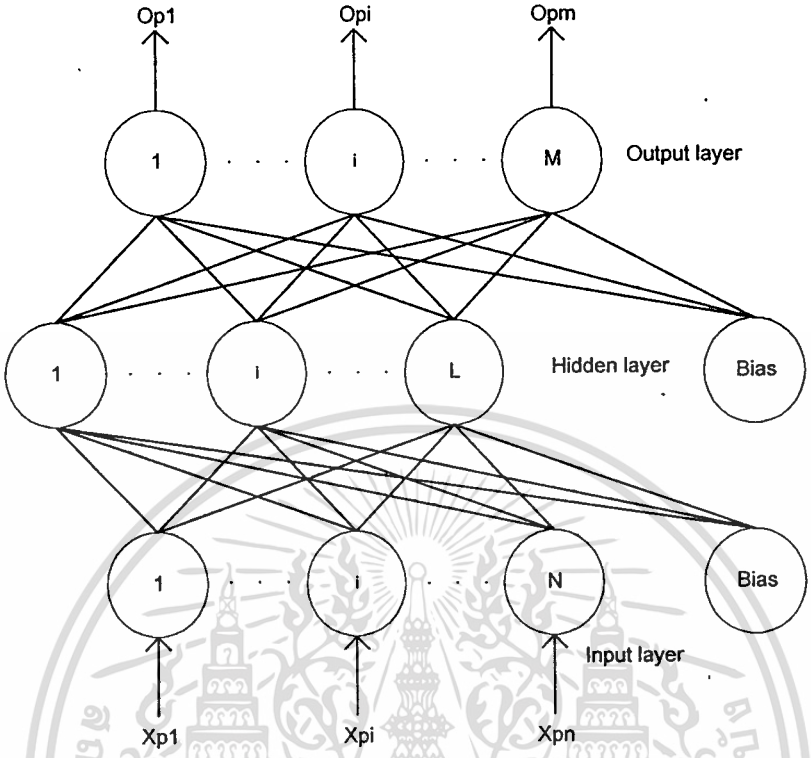
หลักการแพร่ย้อนกลับได้รับการพัฒนาโดยรูเมลฮาร์ทและถูกนำเสนอในช่วงปี ค.ศ.1986 หลักการแพร่ย้อนกลับนี้สามารถนำไปใช้แก้ปัญหาในลักษณะเชิงเส้น(linear) และปัญหาที่ไม่เชิงเส้นได้(nonlinear) ทำให้ประยุกต์เข้ากับโครงงานต่างๆได้อย่างหลากหลายและได้รับความนิยมมากที่สุดในปัจจุบัน หลักการแพร่ย้อนกลับมีพื้นฐานมาจากกฎเดลต้า (delta-rule) ที่พัฒนาขึ้นโดยวิดโรว์ และ ฮอฟฟ์ ในช่วงปี ค.ศ.1960-1962 ซึ่งกระบวนการของกฎเดลต้าจะลดค่าผิดพลาดที่เอาต์พุทของโครงข่ายเปรียบเทียบกับค่าเป้าหมายโดยค่าผิดพลาดที่ลดลงนี้เกิดจากการปรับแต่งค่าน้ำหนักที่อินพุทของแต่ละนิวรอน

<u>สถาปัตยกรรมของโครงข่ายบีพีเอ็น</u>[5]

โครงข่ายบีพีเอ็น (BPN,Back-Propagation Neural Network) เป็นโครงข่ายที่มีจำนวนชั้นมากกว่า 2 ชั้นคือมีได้ตั้งแต่ 3 ชั้นขึ้นไปคือ ชั้นอินพุท(input layer) ชั้นซ่อน(hidden layer) และชั้นเอาต์พุท(output layer) จากภาพที่ 9 แสดงให้เห็นถึงลักษณะของโครงข่ายบีพีเอ็น ซึ่งเป็นฟีดฟอร์เวิร์ดเน็ทเวิร์ค (feedforword network) โดยมีการเชื่อมโยงในแต่ละชั้นแบบฟูลลี่ นั่นคือทุกๆนิวรอนในชั้นอินพุทจะส่งสัญญาณไปยังทุกๆ นิวรอนในชั้นซ่อนชั้นแรก และทุกๆนิวรอนในชั้นซ่อนชั้นแรกจะส่งสัญญาณไปยังทุกๆนิวรอนในชั้นซ่อนถัดไปจนในที่สุดทุกๆ นิวรอนในชั้นซ่อนชั้นสุดท้ายจะส่งสัญญาณไปยังทุกๆนิวรอนในชั้นเอาต์พุท จากภาพแสดงโครงข่ายที่มี 3 ชั้นคือชั้นอินพุท ชั้นซ่อน และชั้นเอาต์พุท ในทางปฏิบัติที่ชั้นซ่อนสามารถที่จะมีได้มากกว่า 1 ชั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง



ภาพที่ 9 แสดงโครงข่ายบีพีเอ็น

โครงข่ายประสาทเทียมเรียกได้ว่าเป็นแมพปิ้งเน็ทเวิร์ค (mapping network) ถ้าหากโครงข่ายสามารถคำนวณฟังก์ชั่นในบางลักษณะที่กำเนิดความสำพันธ์ระหว่างอินพุทและเอาต์พุทในโครงข่าย ตัวอย่างเช่น ถ้ามีอินพุทป้อนให้กับโครงข่ายซึ่งมีค่าเป็นมุม (angle) และเอาต์พุทคือค่าโคซายน์ (cosine) ของมุมที่อินพุท โครงข่ายจะมีการแมพปิ้งคือ $\theta \rightarrow \cos(\theta)$ สิ่งสำคัญที่ทำให้การแมพปิ้งถูกต้องนั้นจำเป็นต้องมีการฝึกหัดโครงข่าย และการฝึกหัดโครงข่ายโดยอาศัยหลักการแพร่ย้อนกลับนั้น แบ่งออกได้เป็นสามขั้นตอนคือ

1. แพร่สัญญานที่ชั้นอินพุทไปยังชั้นซ่อนแรก และแพร่สัญญาณไปยังชั้นซ่อนถัดไปจนถึงชั้นเอาต์พุทโดยผ่านฟังก์ชั่นกระตุ้น

2. คำนวณค่าผิดพลาดของค่าเอาต์พุทกับค่าเป้าหมายและแพร่ย้อนกลับค่าผิดพลาดเหล่านี้ไปยังชั้นซ่อน

3. ปรับแต่งค่าน้ำหนักที่ชั้นเอาต์พุทและชั้นซ่อนเพื่อลดค่าผิดพลาดโดยทำให้ค่าเอาต์พุทมีค่าใกล้เคียงหรือเท่ากับค่าเป้าหมาย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การฝึกหัดโครงข่ายด้วยวิธีการแพร่ย้อนกลับ

เริ่มต้นด้วยการพิจารณาภาพที่ 9 ซึ่งเป็นโครงข่ายขนาด 3 ชั้น ที่มีอินพุทเวกเตอร์ $x_p = (x_{p1}, x_{p2}, \ldots, x_{pn})t$ อินพุท x_p นี้จะถูกป้อนให้กับชั้นอินพุทของโครงข่ายและที่ชั้นอินพุทนี้จะกระจายค่าไปยังนิวรอนในชั้นซ่อน ที่ชั้นซ่อนนี้เองจะทำการคำนวณค่าเน็ท สมการหาค่าเน็ทที่นิวรอน j ในชั้นซ่อนคือ

$$net^{h}_{pj} = \sum_{i=1}^{N} w^{h}_{ji} x_{pi} + \theta_{h_j}$$

เมื่อ w^{h}_{ji} คือน้ำหนักที่ชั้นซ่อนตรงจุดต่อจากโหนดอินพุท i ไปยังนิวรอน j และ θ_{h_j} คือค่าเทอมไบแอส เมื่อได้ค่าเน็ทแล้วขั้นถัดไปคือคำนวณค่าเอาต์พุทของแต่ละนิวรอนในชั้นซ่อนโดยคำนวณได้ดังนี้

$$i_{pj} = f^{h}_{j}(net^{h}_{pj})$$

ใช้สมการทั้งสองคำนวณค่าในชั้นซ่อนทุกๆชั้น จากนั้นจึงทำการคำนวณค่าเน็ทและค่าเอาต์พุทที่ชั้นเอาต์พุทโดยใช้สมการที่คล้ายคลึงกันคือ

$$net^{o}_{pk} = \sum_{j=1}^{L} w^{o}_{kj} i_{pj} + \theta_{o_k}$$

$$o_{pk} = f^{o}_{k}(net^{o}_{pk})$$

การหาค่าผิดพลาดและการปรับน้ำหนักที่ชั้นเอาต์พุท

สมการปรับค่าน้ำหนักที่ชั้นเอาต์พุทแสดงได้ดังนี้

$$w^{o}_{kj}(t+1) = w^{o}_{kj}(t) + \Delta_p w^{o}_{kj}(t)$$

$$\Delta_p w^{o}_{kj} = \eta\,(y_{pk} - o_{pk})\, f^{o}_{k}{}'\,(net^{o}_{pk}) i_{pj}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$i_{pj} = f^{h}_{j}(net^{h}_{pj})$$

องค์ประกอบ η คืออัตราการเรียนรู้ โดยปกติค่านี้จะมีค่าเป็นบวกและน้อยกว่า 1 ส่วน y_{pk} คือค่าเป้าหมาย ลองกลับไปดูฟังก์ชั่นกระตุ้น f^{o}_{k} และ f^{h}_{j} ซึ่งมี 2 รูปแบบที่น่าสนใจคือ

$$f^{o}_{k}(net^{o}_{jk}) = net^{o}_{jk}$$

$$f^{o}_{k}(net^{o}_{jk}) = (1 + \exp(-net^{o}_{jk}))^{-1}$$

ฟังก์ชั่นกระตุ้นรูปแบบแรกเป็นการกำหนดเอาต์พุทเป็นแบบลิเนียร์ ส่วนฟังก์ชั่นรูปแบบหลังเรียกว่าไบนารี่ซิกมอยด์ (binary sigmoid) ภาพที่ 10 แสดงกราฟของฟังก์ชั่นนี้ การเลือกใช้ฟังก์ชั่นนั้นขึ้นอยู่กับว่าต้องการเลือกลักษณะข้อมูลที่เอาต์พุทอย่างไร ถ้าเลือกฟังก์ชั่นรูปแบบแรก $f^{o}_{k}{}' = 1$ ถ้าเลือกรูปแบบที่ 2 ฟังก์ชั่น $f^{o}_{k}{}' = f^{o}_{k}(1 - f^{o}_{k}) = o_{pk}(1 - o_{pk})$ ดังนั้นการปรับน้ำหนักในกรณีของลิเนียร์จะกำหนดได้ดังนี้

$$w^{o}_{kj}(t+1) = w^{o}_{kj}(t) + \eta(y_{pk} - o_{pk})\, i_{pj}$$

ส่วนฟังก์ชั่นไบนารีซิกมอยด์กำหนดได้ดังนี้

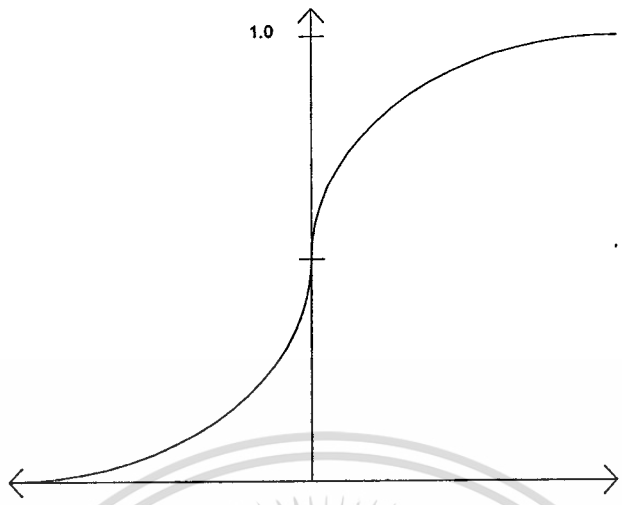
$$w^{o}_{kj}(t+1) = w^{o}_{kj}(t) + \eta(y_{pk} - o_{pk})\, o_{pk}\,(1-o_{pk})\, i_{pj}$$

เขียนสมการให้ดูง่ายขึ้นได้ดังนี้

$$\sigma^{o}_{pk} = (y_{pk} - o_{pk})\, f^{o}_{k}{}'(net^{o}_{pk})$$

$$w^{o}_{kj}(t+1) = w^{o}_{kj}(t) + \eta\, \sigma^{o}_{pk}\, i_{pj}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 10 แสดงกราฟในลักษณะ S-shape ของฟังก์ชั่นไบนารีซิกมอยด์

<u>การหาค่าผิดพลาดและการปรับน้ำหนักที่ชั้นซ่อน</u>

การปรับแต่งค่าน้ำหนักที่ชั้นซ่อนจะต้องทำการคำนวณหาค่าผิดพลาดที่ชั้นซ่อน (hidden-layer error term) ก่อนโดยใช้สมการ

$$\sigma^{h}_{pj} = f^{h}_{j}{}'(net^{h}_{pj}) \sum_{k} \sigma^{o}_{pk} w^{o}_{kj}$$

จากนั้นจึงสามารถปรับแต่งค่าน้ำหนักโดยสมการที่คล้ายคลึงกับสมการปรับแต่งค่าน้ำหนักที่ชั้นเอาต์พุทคือ

$$w^{h}_{ji}(t+1) = w^{h}_{ji}(t) + \eta \sigma^{h}_{pj} x_{i}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สรุปขั้นตอนการฝึกหัดโครงข่ายประสาทเทียมโดยวิธีการแพร่ย้อนกลับ

1. กำหนดอินพุทเวคเตอร์ $x_n = (x_{p1}, x_{p2}, \ldots, x_{pn})$ แก่โหนดอินพุท
2. คำนวณหาค่าเน็ทอินพุทของแต่ละนิวรอนในชั้นซ่อนโดยสมการ

$$net^{h}_{pj} = \sum_{i=1}^{N} w^{h}_{ji} x_{pi} + \theta_{h_j}$$

3. คำนวณค่าเอาต์พุทของแต่ละนิวรอนในชั้นซ่อน

$$i_{pj} = f^{h}_{j}(net^{h}_{pj})$$

4. คำนวณค่าเน็ทของแต่ละนิวรอนในชั้นเอาต์พุท

$$net^{o}_{pk} = \sum_{j=1}^{L} w^{o}_{kj} i_{pj} + \theta_{o_k}$$

5. คำนาณค่าเอาต์พุทของแต่ละนิวรอนในชั้นเอาต์พุท

$$o_{pk} = f^{o}_{k}(net^{o}_{pk})$$

6. คำนวณเทอมค่าผิดพลาดของแต่ละนิวรอนในชั้นเอาต์พุท

$$\sigma^{o}_{pk} = (y_{pk} - o_{pk}) f^{o}_{k}{}'(net^{o}_{pk})$$

7. คำนวณเทอมค่าผิดพลาดของแต่ละนิวรอนในชั้นซ่อน

$$\sigma^{h}_{pj} = f^{h}_{j}{}'(net^{h}_{pj}) \sum_{k} \sigma^{o}_{pk} w^{o}_{kj}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หมายเหตุ ค่าเทอมผิดพลาดของแต่ละนิวรอนในชั้นซ่อนต้องคำนวณก่อนที่จะมีการปรับแต่งน้ำหนักที่ชั้นเอาต์พุท

8. ปรับแต่งค่าน้ำหนักที่ชั้นเอาต์พุท

$$w^{o}_{kj}(t+1) = w^{o}_{kj}(t) + \eta \sigma^{o}_{pk} i_{pj}$$

9. ปรับแต่งค่าน้ำหนักที่ชั้นซ่อน

$$w^{h}_{ji}(t+1) = w^{h}_{ji}(t) + \eta \sigma^{h}_{pj} x_i$$

และต้องคำนวณเทอมค่าผิดพลาด

$$Ep = \left(\sum_{p=1}^{M} \left(\sum_{k=1} (y_{pk} - o_{pk})^2\right)^{1/2}\right)/p$$

ซึ่งเป็นปริมาณการวัดว่าโครงข่ายเรียนรู้ได้ดีเพียงไร เมื่อค่าผิดพลาดมีค่าต่ำลงจนสามารถยอมรับได้(Error tolerance)สำหรับทุก ๆ คู่อันดับ อินพุท-เอาต์พุท ของกลุ่มข้อมูลทดสอบการฝึกหัดนี้ก็ยุติลงได้

<u>หัวข้ออื่น ๆ ของหลักการแพร่ย้อนกลับ</u>

การเพิ่มประสิทธิภาพในการเรียนรู้โดยอาศัยวิธีการแพร่ย้อนกลับนั้น สามารถทำได้โดยพิจารณาในแง่ความเหมาะสมโดยการนำเอาเทอมของการคำนวณที่จะกล่าวถึงต่อไปมาใช้ร่วมกับหลักการแพร่ย้อนกลับ ซึ่งอาจจะทำให้การเรียนรู้ของโครงข่ายรวดเร็วขึ้นในแง่ของเวลาที่ใช้ในการฝึกหัดโครงข่าย

<u>การเพิ่มเทอมโมเมนตัม</u>

การลดช่วงเวลาในการเรียนรู้วิธีหนึ่งคือการนำเทคนิคของโมเมนตัมซึ่งเป็นวิธีการปรับค่าน้ำหนัก โดยการคำนวณค่าน้ำหนักเปลี่ยนแปลงในปัจจุบันรวมกับเศษส่วนของค่าน้ำหนักที่มีการเปลี่ยนแปลงก่อนหน้านี้ซึ่งเป็นเทอมของโมเมนตัม การเพิ่มเทอมนี้ขึ้นมาก็เพื่อจะให้การ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เปลี่ยนแปลงของค่าน้ำหนักไปในทิศทางเดียวกัน สมการปรับค่าน้ำหนักเมื่อเพิ่มเทอมโมเมนตัมเข้าไปกำหนดได้ดังนี้

$$w^{\circ}_{kj}(t+1) = w^{\circ}_{kj}(t) + \eta\sigma^{\circ}_{pk} i_{pj} + \alpha\Delta w^{\circ}_{kj}(t-1)$$

จากสมการเป็นการปรับค่าน้ำหนักที่ชั้นเอาต์พุท ซึ่งก็ใช้ได้กับการปรับค่าน้ำหนักที่ชั้นซ่อนด้วย พารามิเตอร์ α โดยทั่วไปถูกกำหนดให้มีค่าเป็นบวกและน้อยกว่า 1

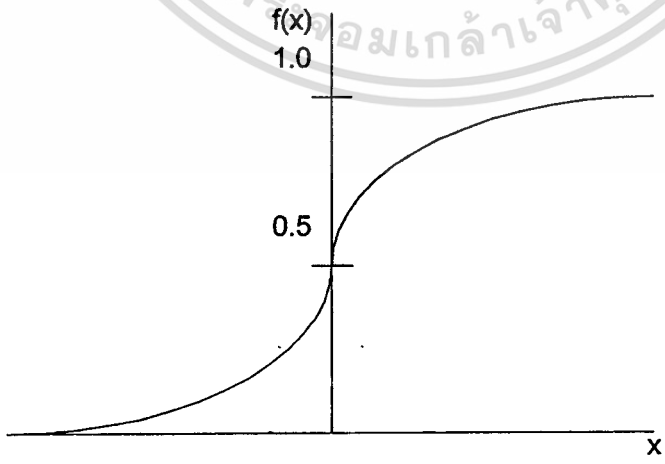
<u>ฟังก์ชั่นกระตุ้น</u>[8]

ฟังก์ชั่นกระตุ้นสำหรับหลักการแพร่ย้อนกลับนี้มีความสำคัญมากที่จะเป็นตัวกำหนดคุณสมบัติให้กับโครงข่าย ฟังก์ชั่นกระตุ้นที่นิยมใช้กันมากคือฟังก์ชั่นซิกมอยด์ได้แก่ไบนารีซิกมอยด์ (binary sigmoid) ซึ่งได้เคยกล่าวถึงมาแล้ว ค่าเอาต์พุทที่ได้จากฟังก์ชั่นนี้อยู่ในช่วงระหว่าง 0 และ 1 ฟังก์ชั่นไบนารีซิกมอยด์กำหนดได้ดังนี้

$$f_1(x) = (1 + e^{-x})^{-1}$$

จะได้ฟังก์ชั่นอนุพันธ์ คือ

$$f_1'(x) = f_1(x)(1 - f_1(x))$$



ภาพที่ 11 แสดงกราฟของฟังก์ชั่นไบนารีซิกมอยด์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ฟังก์ชั่นซิกมอยด์อีกชนิดหนึ่งที่นิยมใช้กันมากก็คือไบโพล่าซิกมอยด์ (bipolar sigmoid) ซึ่งค่าเอาต์พุทที่ได้จากฟังก์ชั่นอยู่ในช่วงระหว่าง -1 และ 1 ฟังก์ชั่นไบโพล่าซิกมอยด์กำหนดได้ดังนี้

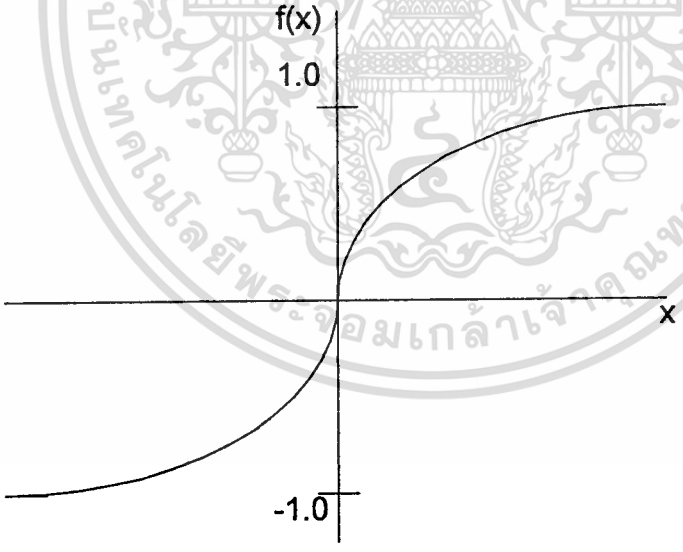
$$f_2(x) = (2/(1 + e^{-x})) -1$$

จะได้ฟังก์ชั่นอนุพันธ์ คือ

$$f_2'(x) = 1/2(1+f_2(x))(1 - f_2(x))$$

ฟังก์ชั่นไบโพล่าซิกมอยด์นี้มีลักษณะคล้ายคลึงกับฟังก์ชั่น

$$\tanh(x) = (e^{x} - e^{-x}) / (e^{x} + e^{-x})$$



ภาพที่ 12 แสดงกราฟของฟังก์ชั่นไบโพล่าซิกมอยด์

<u>การกำหนดค่าน้ำหนักเริ่มต้น</u>[8]

การกำหนดค่าน้ำหนักเริ่มต้นโดยทั่ว ๆ ไปแล้วจะใช้วิธีการสุ่มค่า ซึ่งจะมีค่าอยู่ในช่วงระหว่าง -0.5 และ 0.5 หรือช่วงระหว่าง-1และ1 หรืออาจจะอยู่ในช่วงระหว่างค่าใดๆที่เหมาะสมตามคุณสมบัติการออกแบบโครงข่ายนั้นๆ แต่ยังมีวิธีการกำหนดค่าน้ำหนักอีกวิธีหนึ่ง ที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

พัฒนาขึ้นโดย แน็กกายเอ็นและวิดโรว์ (Nguyen and Widrow)ในปี 1990 วิธีการดังกล่าวมีพื้นฐานมาจากการวิเคราะห์เชิงเรขาคณิตของการตอบสนองต่อสัญญาณอินพุทเดี่ยวที่เข้ามายังนิวรอนในชั้นซ่อนและขยายการวิเคราะห์ออกไปเป็นกรณีของจำนวนอินพุทหลายๆอินพุทโดยใช้การแปลงฟูเรียร์ (fourier transform)

วิธีการของแน็กกายเอ็นและวิดโรว์นี้จะกำหนดค่าน้ำหนักที่เชื่อมต่อระหว่างชั้นซ่อนและชั้นเอาต์พุทให้มีค่าสุ่มอยู่ระหว่าง -0.5 และ 0.5 ส่วนค่าน้ำหนักที่เชื่อมต่อระหว่างชั้นอินพุทและชั้นซ่อนนั้นกำหนดได้ดังนี้

n คือ จำนวนของอินพุท

p คือ จำนวนของนิวรอนในชั้นซ่อน

β คือ สเกลแฟคเตอร์

โดยที่ $\beta = 0.7(p)^{1/n}$

กระบวนการของการกำหนดค่าน้ำหนักทำได้ดังนี้

for each hidden neuron (j=1,..,p);

Initialize its weight vector (from the input units):

w_{ij}(old) random number between -0.5 and 0.5 (or between $-\Upsilon$ and Υ)

Compute $||w_j(old)||$

Reinitialize weights:

$w_{ij} = \beta w_{ij}(old) / ||w_j(old)||$

Set bias:

w_{0j} = random number between $-\beta$ and β

การกำหนดค่าน้ำหนักเริ่มต้นของแน็กกายเอ็นและวิดโรว์นี้ได้ทดสอบบนโครงข่ายที่ใช้ฟังก์ชั่นกระตุ้นไบโพล่าซิกมอยด์

$$\tanh(x) = (e^{x} - e^{-x}) / (e^{x} + e^{-x})$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซึ่งผลที่ได้คือทำให้การเรียนรู้ในชั้นซ่อนมีประสิทธิภาพดีขึ้น ภาพรวมก็คือใช้ระยะเวลาการเรียนรู้ของโครงข่ายน้อยลงนั่นเอง

การกำหนดจำนวนชั้นซ่อน

ยังไม่มีวิธีการใดที่สามารถกำหนดจำนวนของชั้นซ่อนให้มีค่าที่แน่นอนได้ แต่ก็มีทฤษฎีที่กล่าวไว้ว่า ชั้นซ่อนเพียงชั้นเดียวก็พอเพียงแล้วสำหรับการนำไปใช้กับปัญหาที่ไม่เป็นเชิงเส้นใด ๆ ก็ได้ แต่ในบางกรณีการเพิ่มชั้นซ่อนก็สามารถช่วยให้การเรียนรู้ง่ายขึ้น รวดเร็วขึ้นและอีกปัญหาหนึ่งคือในชั้นซ่อนควรจะมีนิวรอนกี่นิวรอนซึ่งเช่นเดียวกันคือยังไม่มีกฎเกณฑ์ตายตัวใดๆ กำหนดให้อย่างแน่นอนได้ เนื่องจากจำนวนนิวรอนในชั้นซ่อนนี้ขึ้นอยู่กับความซับซ้อนของปัญหา มีหลายเทคนิคถูกนำเสนอออกมาเพื่อใช้หาจำนวนนิวรอนที่เหมาะสมในชั้นซ่อน ซึ่งวิธีที่นับว่าแพร่หลายพอสมควรคือ ใช้นิวรอนให้มีจำนวนมากไว้ก่อนแล้วค่อย ๆ ลดจำนวนลงมาจนได้จำนวนที่น้อยที่สุดที่โครงข่ายประสาทเทียมยังคงเรียนรู้ได้ เทคนิคดังกล่าวเรียกว่า พรุนนิ่ง (Pruning)

บทสรุป

ในบทนี้ได้กล่าวถึงประวัติความเป็นมาของการวิจัยศาสตร์ทางด้านโครงข่ายประสาทเทียม พร้อมทั้งแสดงทฤษฎีการทำงานของรูปแบบอะดาไลน์ มาดาไลน์และหลักการแพร่ย้อนกลับ ซึ่งเป็นความรู้พื้นฐานของงานวิจัยนี้ ในบทที่ 3 หรือบทถัดไปจะแสดงแนวคิดที่จะแปลงรูปแบบทฤษฎีการทำงานในบทนี้เป็นภาษาโปรแกรมคอมพิวเตอร์เพื่อทำให้ไมโครคอมพิวเตอร์สามารถจำลองการทำงานตามทฤษฎีเหล่านี้ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
การออกแบบโปรแกรมจำลองโครงข่ายประสาทเทียม

ในบทนี้จะกล่าวถึงการออกแบบโปรแกรมจำลองโครงข่ายประสาทเทียม โดยแบ่งเนื้อหาตามชนิดของโปรแกรมซึ่งมีอยู่ 2 แบบคือ การออกแบบโปรแกรมจำลองอะดาไลน์และการออกแบบโปรแกรมจำลองโครงข่ายประสาทเทียมโดยใช้หลักการแพร่ย้อนกลับ

<u>การออกแบบโปรแกรมจำลองอะดาไลน์</u>

จากทฤษฎีของอะดาไลน์ดังที่ได้กล่าวมาแล้วได้ถูกนำมาสร้างเป็นโปรแกรมเพื่อจำลองการทำงานตามทฤษฎีโดยได้นำหลักทฤษฎีทางคณิตศาสตร์มาดัดแปลงให้อยู่ในรูปของอัลกอริธึมเพื่อลำดับขั้นตอนในการสร้างโปรแกรมที่สามารถทำงานได้โดยอาศัยเครื่องไมโครคอมพิวเตอร์ ในขั้นแรกของการออกแบบนั้นได้มีการกำหนดคุณสมบัติของโปรแกรมที่จะสร้างโดยอิงการทำงานบนระบบปฏิบัติการของดอส(DOS)และใช้ภาษาซีในการโปรแกรม เนื่องจากหลักการของอะดาไลน์เป็นการจำลองการทำงานของนิวรอนเดี่ยวซึ่งไม่ซับซ้อนมากนักต่อการโปรแกรม ดังนั้นในสภาวะแวดล้อมบนระบบปฏิบัติการของดอสที่ยอมให้ใช้หน่วยความจำคอนเวนชั่นแนลแมมโมรี่ซึ่งจะใช้หน่วยความจำภายใต้ 640 กิโลไบท์ก็เพียงพอต่อการทำงานของโปรแกรม ภาษาที่เลือกใช้ในการจำลองทฤษฎีอะดาไลน์ได้เลือก ไมโครซอฟท์ควิคซีเวอร์ชั่น 2.5 (Microsoft Quick C V2.5) ในหัวข้อต่อไปจะกล่าวถึงโครงสร้างและอัลกอริธึมของโปรแกรมที่ได้สร้างขึ้นตามลำดับ

<u>โครงสร้างหลักของโปรแกรม</u>

โปรแกรมที่สร้างขึ้นมานี้ถูกออกแบบในลักษณะของโปรแกรมโครงสร้าง โดยได้ออกแบบการทำงานในส่วนต่างๆไว้เป็นโมดูล (module) ประกอบด้วยโมดูลต่างๆ ดังนี้

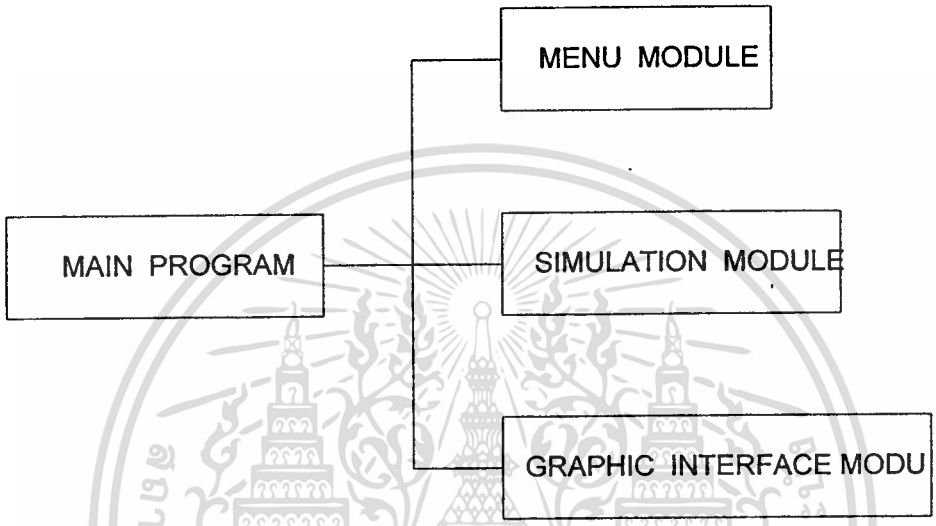
1. เมนูโมดูล (menu module) เป็นโมดูลในการทำงานเกี่ยวกับเมนูของโปรแกรม ผู้ใช้สามารถเลือกเมนูในขั้นตอนการใช้งานโปรแกรมเพื่อเลือกการทำงานด้านต่างๆ ในลักษณะของภาพกราฟิก

2. ซิมมูเลชั่นโมดูล (simulation module) เป็นโมดูลการทำงานในการจำลองทฤษฎีอะดาไลน์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. กราฟิกอินเตอร์เฟสโมดูล (graphic interface module) เป็นโมดูลการทำงานเกี่ยวกับการแสดงภาพกราฟิกต่างๆในโปรแกรม

การสร้างโปรแกรมนั้นจะเรียกใช้ฟังก์ชั่นต่างๆอย่างสำพันธ์กันตามหน้าที่และลำดับการทำงานโดยแสดงแผนภูมิโครงสร้าง (structure chart) หลักของโปรแกรมได้ คือ



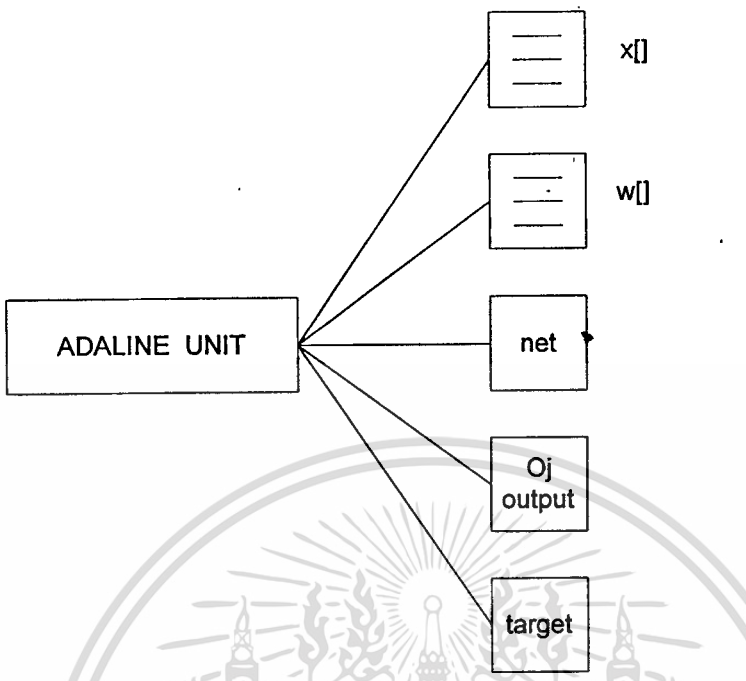
ภาพที่ 13 แสดงแผนภูมิโครงสร้างหลักของโปรแกรม

ในหัวข้อถัดไปจะของกล่าวเฉพาะในส่วนซิมมูเลชั่นโมดูลหรือส่วนจำลองอะดาไลน์ที่ได้ออกแบบและสร้างขึ้นมา

<u>โครงสร้างแบบจำลองอะดาไลน์</u>

สิ่งสำคัญที่สุดของการออกแบบโปรแกรมจำลองอะดาไลน์คือ การออกแบบโครงสร้างอะดาไลน์เพราะถ้ามีโครงสร้างที่ดีแล้วการออกแบบโปรแกรมก็สามารถทำให้กระชับและง่ายต่อการเขียนโปรแกรม โครงสร้างของอะดาไลน์ที่ออกแบบไว้มีลักษณะดังภาพที่ 14

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 14 แสดงโครงสร้างแบบจำลองอะดาไลน์

ต่อไปจะกล่าวถึงโครงสร้างแบบจำลองอะดาไลน์เพื่อที่จะได้เกิดความเข้าใจกับอัลกอริธึมการเรียนรู้ของอะดาไลน์ ลักษณะโครงสร้างอะดาไลน์ประกอบด้วยตัวแปรหมวด x[ ] ซึ่งใช้เก็บค่าอินพุทเวคเตอร์ที่ป้อนให้กับหน่วยอะดาไลน์ ตัวแปรหมวด w[ ] ใช้เก็บค่าน้ำหนักที่เชื่อมต่อระหว่างหน่วยอินพุทกับหน่วยประมวลผล ตัวแปร net ใช้เก็บค่าเน็ทที่ได้จากการคำนวณ ตัวแปร O_j หรือ output นี้จะใช้เก็บค่าเอาต์พุทของอะดาไลน์ และตัวแปร target ใช้เก็บค่าเป้าหมายเพื่อใช้ในการคำนวณ

<u>อัลกอริธึมในการสร้างส่วนจำลองอะดาไลน์</u>

ในหัวข้อนี้จะกล่าวถึงการสร้างส่วนจำลองอะดาไลน์ซึ่งเป็นองค์ประกอบสำคัญของฟังก์ชั่นในซิมมูเลชั่นโมดูล อัลกอริธึมที่จะแสดงในหัวข้อนี้ได้มาจากการแปลงหลักการของอะดาไลน์มาเป็นรูปแบบที่ทำงานได้โดยโปรแกรมคอมพิวเตอร์และเนื่องจากโปรแกรมจำลองอะดาไลน์ถูกพัฒนาขึ้นมาจากภาษาซีดังนั้นจึงแสดงอัลกอริธึมการทำงานในรูปแบบของภาษาที่ใช้อยู่นี้ การดูอัลกอริธึมหรือโปรแกรมให้พิจารณาร่วมกับโครงสร้างแบบจำลองอะดาไลน์ที่ได้กล่าวไปในหัวข้อที่ผ่านมาร่วมกับทฤษฎีการคำนวณของอะดาไลน์เพื่อทำให้ง่ายต่อความเข้าใจอีกทั้งผู้อ่านต้องมีความรู้ด้านภาษาซีมาบ้าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ก่อนที่จะกล่าวถึงอัลกอริธึมจะขอกล่าวถึงการจัดแบ่งหน้าที่หรือลักษณะของการจำลองอะดาไลน์โดยสามารถแบ่งได้เป็นสองส่วนหลัก ๆ คือ

1. การจำลองในการฝึกหัดอะดาไลน์ หรือการสอนอะดาไลน์นั่นเอง

2. การจำลองในการทดสอบอะดาไลน์เพื่อหาค่าเอาต์พุทจากกลุ่มข้อมูลทดสอบ โดยอะดาไลน์ที่ได้รับการฝึกหัดมาแล้ว

วิธีจำลองในการฝึกอะดาไลน์สามารถแสดงได้โดยคำสั่งเทียม (pseudo code) โดยมีลำดับขั้นตอนดังนี้

```
begin
    initialize weights
    while output error
    do
     set input to apply a training input
     calculate net
     calculate output using threshold
     if  output ≠ target
      update weights
    end do
    save weights to file
end
```

วิธีจำลองในการทดสอบอะดาไลน์แสดงได้ดังนี้

```
begin
    set weights value from file
    set input to stimulate the adaline with an input
    calculate net
    calculate output using threshold
end
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ลำดับต่อไปจะแสดงอัลกอริธึมที่สำคัญคือ calculate net, calculate output และ update weights

อัลกอริธึม calculate net

ฟังก์ชั่น CalculateNet ทำหน้าที่คำนวณหาค่า net ของ หน่วยอะดาไลน์

```
void CalculateNet ( )
{
  int i
  net = 0;
  for(i=0; i< NumberOfInputs; i++)
    net = net + w[i] * x[i];
}
```

อัลกอริธึม calculate output

ฟังก์ชั่น CalculateOutput ทำหน้าที่คำนวณค่าเอาต์พุทของหน่วยอะดาไลน์ โดยใช้กฎเทรสโฮลด์

```
void CalculateOutput ( )
{
  if(net < 0) output = -1;
  else output = 1;
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อัลกอริธึม update weights

ฟังก์ชั่น UpdateWeights ทำหน้าที่ในการปรับค่าน้ำหนักของหน่วยอะดาไลน์

```
void UpdateWeights ( )
{
  int i;
  long delta_w;
  for(i=0;i<NumberOfInputs;i++){
    delta_w=eta*x[i]*(target-net);
    w[i]=w[i]+delta_w;
  }
}
```

การออกแบบโปรแกรมจำลองอะดาไลน์ที่ได้นำเสนอนี้ได้กล่าวเฉพาะในส่วนของซิมมูเลชั่นโมดูลเพราะเกี่ยวข้องกับหัวข้องานวิจัยโดยตรงและได้ข้ามในส่วนของเมนูโมดูลและกราฟิกอินเตอร์เฟสโมดูลไปทั้งนี้เพราะเป็นรายละเอียดที่ไม่เกี่ยวกับโครงข่ายประสาทเทียม ถ้านำเสนอทั้งหมดคงจะต้องกล่าวในรายละเอียดมาก อีกทั้งจะกลายเป็นการนำเสนอเทคนิคการเขียนโปรแกรมในส่วนอื่นๆที่ไม่เกี่ยวข้องกับการจำลองโครงข่ายประสาทเทียม ในหัวข้อถัดไปจะกล่าวถึงการออกแบบโปรแกรมจำลองโครงข่ายประสาทเทียมโดยใช้หลักการแพร่ย้อนกลับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การออกแบบโปรแกรมจำลองโครงข่ายบีพีเอ็น

จากทฤษฎีของหลักการแพร่ย้อนกลับดังที่กล่าวมาแล้วได้ถูกนำมาสร้างเป็นโปรแกรมเพื่อใช้งานจริงโดยได้นำหลักทฤษฎีทางคณิตศาสตร์ข้างต้นมาดัดแปลงให้อยู่ในรูปของอัลกอริธึมเพื่อลำดับขั้นตอนในการสร้างโปรแกรมที่สามารถทำงานได้โดยอาศัยเครื่องไมโครคอมพิวเตอร์ ในขั้นแรกของการออกแบบนั้นได้มีการกำหนดคุณสมบัติของโปรแกรมที่จะสร้างโดยพิจารณาตั้งแต่ทฤษฎีของหลักการแพร่ย้อนกลับว่ามีความเหมาะสมที่จะจำลองการทำงานโดยอาศัยสภาวะแวดล้อมของระบบปฏิบัติการใด(Operating System)และควรใช้ภาษาคอมพิวเตอร์ใดในการพัฒนาโปรแกรม จากนั้นจึงออกแบบโครงสร้างหลักการทำงานของโปรแกรม ออกแบบโครงสร้างแบบจำลองโครงข่ายประสาทเทียมตามสถาปัตยกรรมของโครงข่ายบีพีเอ็นที่ได้กล่าวมาแล้วและในขั้นตอนสุดท้ายได้มีการแยกแยะกลุ่มฟังก์ชั่นต่างๆ เพื่อใช้ในการทำงานของโปรแกรม โดยจะขอกล่าวตามหัวข้อเพื่อพิจารณาในรายละเอียดต่อไป

ระบบปฏิบัติการและภาษาที่เลือกใช้

โปรแกรมจำลองโครงข่ายประสาทเทียมที่ได้สร้างขึ้นมานี้แบ่งได้เป็น 2 ประเภท คือโปรแกรมจำลองนิวรอนเดี่ยวโดยใช้หลักการของอะดาไลน์ซึ่งได้กล่าวผ่านมาแล้ว โปรแกรมดังกล่าวทำงานภายใต้ระบบปฏิบัติการของดอส(DOS)และใช้ไมโครซอฟท์ควิคซี (Microsoft Quick C V2.5) เป็นภาษาที่ใช้ในการสร้างโปรแกรม การเขียนโปรแกรมให้ทำงานภายใต้ระบบปฏิบัติการของดอสนั้นจะทำให้การสร้างโปรแกรมทำได้ง่ายเพราะไม่ต้องคำนึงถึงสภาพแวดล้อมของระบบปฏิบัติการมากนัก การสร้างโปรแกรมจึงดำเนินในลักษณะของการแปลงฟังก์ชั่นทางคณิตศาสตร์ให้เป็นอัลกอริธึมเชิงโปรแกรมเป็นส่วนใหญ่แต่การพัฒนาบนดอสก็มีปัญหาอยู่บ้างเป็นปัญหาไม่มากถ้าเป็นโปรแกรมขนาดเล็กแต่จะเป็นปัญหามากถ้าสร้างโปรแกรมที่มีขนาดใหญ่ ปัญหาหลักของการเขียนโปรแกรมบนดอสนั้นคือในเรื่องของการจัดการหน่วยความจำและการสร้างยูสเซอร์อินเตอร์เฟส (user interface) ปัญหาเรื่องการจัดการหน่วยความจำนั้นบนระบบปฏิบัติการของดอสสามารถใช้หน่วยความจำได้เฉพาะในส่วนของคอนเวนชั่นแนลเมมโมรี่ ซึ่งจะใช้หน่วยความจำภายใต้ 640 กิโลไบท์เท่านั้น ถึงแม้ว่าภายในเครื่องไมโครคอมพิวเตอร์จะมีหน่วยความจำมากเกิน 640 กิโลไบท์ก็ตาม ถ้าจะใช้หน่วยความจำเกินกว่านี้ก็ต้องมีไดรเวอร์สนับสนุนหรือไม่ก็สร้างโปรแกรมจัดการหน่วยความจำที่มากเกินกว่า 640 กิโลไบท์ ซึ่งเป็นเรื่องที่ลำบากและอาศัยเวลาในการออกแบบโปรแกรมเพื่อให้ใช้งานหน่วยความจำให้ได้เกิน 640 กิโลไบท์ โปรแกรมจำลองอะดาไลน์ทำงานบนดอสได้ดีก็เพราะโปรแกรมมีขนาดเล็ก และโครงสร้างของนิวรอนที่จำลองก็มีเพียงนิวรอนเดียวจึงไม่มีความจำเป็นต้องใช้หน่วยความจำมากมายนัก เมื่อกลับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มาพิจารณาถึงโครงสร้างการจำลองโครงข่ายประสาทเทียมโดยอาศัยหลักการแพร่ย้อนกลับ ซึ่งมีลักษณะเป็นโครงข่ายประกอบด้วยนิวรอนหลายนิวรอนต่อเชื่อมกันระหว่างชั้นต่างๆอีกทั้งยังสามารถเปลี่ยนแปลงจำนวนนิวรอนหรือจำนวนชั้นของโครงข่ายได้ ทำให้บางกรณีต้องใช้หน่วยความจำมากเพื่อรองรับกับโครงสร้างที่ถูกกำหนดขึ้น ดังนั้นถ้าออกแบบให้โปรแกรมทำงานภายใต้ระบบปฏิบัติการของดอสแล้วเป็นเรื่องค่อนข้างลำบาก จึงเกิดความสนใจที่จะพัฒนาโปรแกรมนี้บนระบบปฏิบัติการของไมโครซอฟท์วินโดว์ (Microsoft Window 3.1) ซึ่งทำให้การจัดการเกี่ยวกับหน่วยความจำนั้นง่ายขึ้น สามารถใช้หน่วยความจำได้เต็มที่ตามขีดจำกัดของขนาดหน่วยความจำของเครื่องไมโครคอมพิวเตอร์และการทำยูสเซอร์อินเตอร์เฟสเช่นระบบพูลดาวน์เมนู(pull down menu) ไดอะล็อกบ๊อกหรือกรอบโต้ตอบกับผู้ใช้ (dialog box) และฟังก์ชั่นกราฟิกก็มีให้ใช้ได้โดยสะดวก ถึงแม้จะมีปัญหาเรื่องการที่ต้องคำนึงถึงระบบปฏิบัติการ เช่นการเขียนโปรแกรมจะต้องมีรูปแบบเฉพาะ แต่เมื่อมองภาพรวมแล้วการพัฒนาโปรแกรมบนไมโครซอฟท์วินโดว์สะดวกกว่าบนดอสมาก (ในกรณีจำลองโครงข่ายประสาทเทียมโดยอาศัยหลักการแพร่ย้อนกลับ) จึงเลือกที่จะสร้างโปรแกรมบนระบบปฏิบัติการวินโดว์

ภาษาที่เลือกใช้นั้นในขณะที่ข้าพเจ้าเริ่มต้นที่จะสร้างโปรแกรม คอมไพเลอร์ (compiler) ที่นิยมในขณะนั้นคือบอร์แลนด์ซีพลัสพลัส (Borland C ++) กำลังเป็นที่นิยม เพราะมีหนังสือเกี่ยวกับบอร์แลนด์ซีเป็นจำนวนมาก อีกทั้งภาษาซีพลัสพลัสเป็นภาษาที่ข้าพเจ้าให้ความสนใจเพราะเป็นภาษาที่ใช้ในการโปรแกรมเชิงวัตถุ (Object-Oriented programming) ซึ่งเป็นวิธีการเขียนโปรแกรมที่มีลักษณะเป็นโครงสร้างและเป็นโมดูลได้ดีมากกว่าภาษาซี ซึ่งหมายความว่าโปรแกรมเชิงวัตถุนี้จะง่ายต่อการบำรุงรักษาและแก้ไขได้สะดวกกว่าภาษาซี ดังนั้นจึงเลือกภาษาซีพลัสพลัสในการสร้างโปรแกรมนี้ และใช้บอร์แลนด์ซีพลัสพลัสเวอร์ชั่น 4 เป็นตัวแปรภาษา

<u>โครงสร้างหลักของโปรแกรม</u>

โปรแกรมที่สร้างขึ้นมานี้ถูกออกแบบในลักษณะของโปรแกรมเชิงวัตถุจึงได้มีการจัดแบ่งหน้าที่การทำงานในส่วนต่างๆ เป็นคลาส (class) ซึ่งสามารถจำแนกคลาสต่าง ๆ ได้ดังนี้

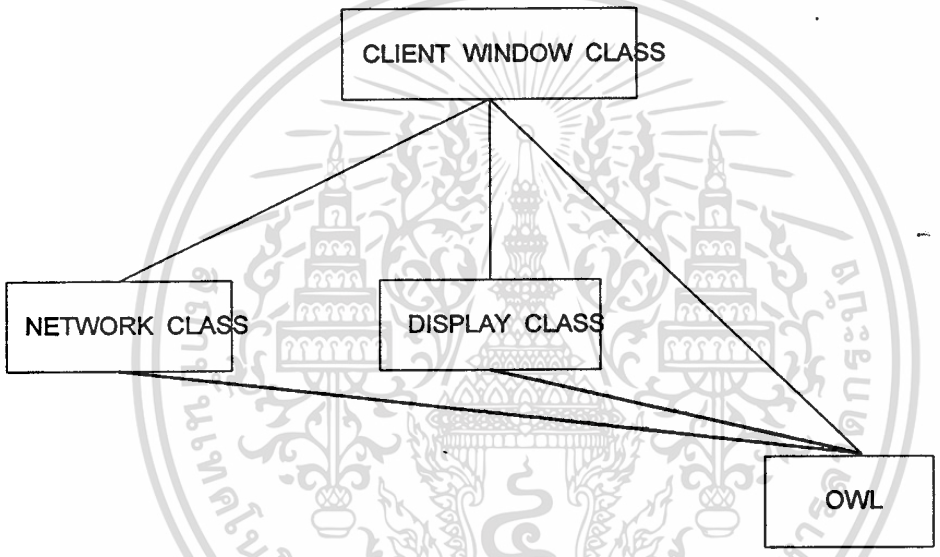
1. คลาสไคลเอนท์วินโดว์ (client window class) เป็นคลาสหลักของการทำงานภายในคลาสประกอบด้วยตัวแปรและกลุ่มฟังก์ชั่นสมาชิกที่ตอบสนองคำสั่งจากเมนู การทำงานของฟังก์ชั่นบางฟังก์ชั่นภายในคลาสยังเรียกใช้คลาสย่อยอื่น ๆ ที่กำหนดไว้ด้วย

2. คลาสโครงข่าย (network class) ภายในคลาสประกอบด้วยตัวแปรและฟังก์ชั่นสมาชิกที่ใช้ในการทำงานของการจำลองโครงข่ายประสาทเทียม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. คลาสแสดงผล (display class) ภายในคลาสประกอบด้วยตัวแปรและฟังก์ชั่นสมาชิกที่ใช้ในการแสดงผลการทำงานทางจอภาพ

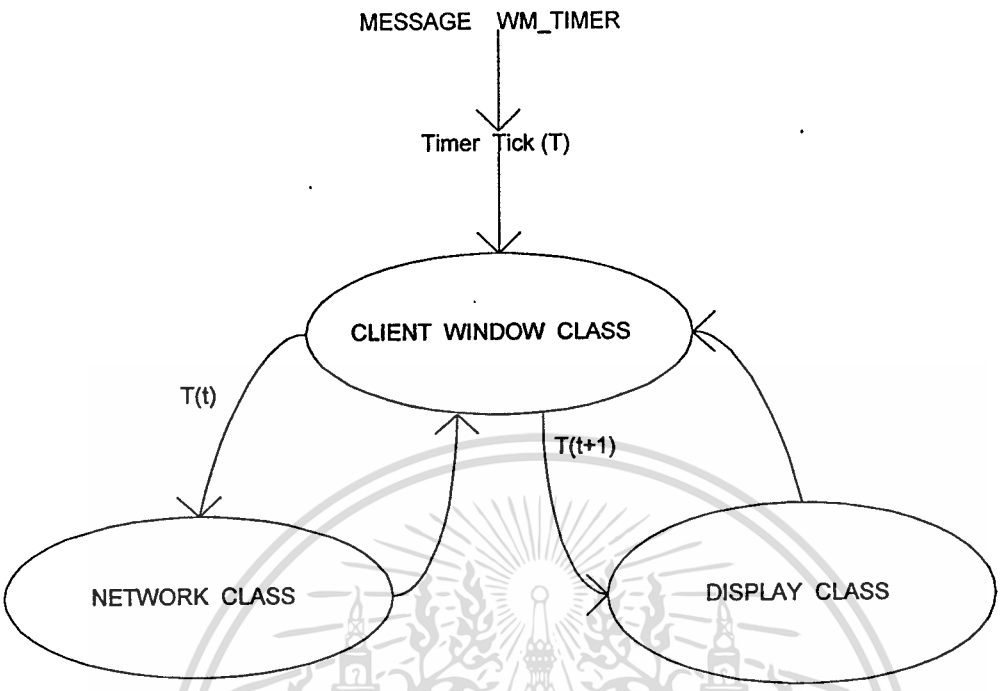
4. ออปเจคท์วินโดว์ไลบรารี่ (Object Window Library , OWL) บรรจุคลาสของบอร์แลนด์ซี คลาสต่างๆ ภายใน OWL นี้ได้ถูกออกแบบไว้เป็นพิเศษเพื่อให้ใช้งานร่วมกับระบบปฏิบัติการวินโดว์ได้ การใช้ออปเจคท์จากคลาสใน OWL จะทำให้การทำโอเปอเรชั่นต่างๆ บนวินโดว์ได้สะดวกและง่ายดายมาก เช่น การสร้างหน้าต่าง การสร้างตัวควบคุมต่างๆ เช่น สกรอลล์บาร์ (scroll bar) และปุ่มกด (push button) การโต้ตอบกับผู้ใช้และอื่น ๆ



ภาพที่ 15 แสดงลำดับชั้นของคลาส

เนื่องจากโปรแกรมทำงานภายใต้ระบบปฏิบัติการวินโดว์จึงมีเทคนิคการสร้างโปรแกรมเรียกว่า มัลติเปิลดอคูเมนท์อินเตอร์เฟส (Multiple Document Interface) หรือเอ็มดีไอ (MDI) ซึ่งการนำเทคนิคนี้มาใช้ทำให้การทำงานของโปรแกรมสามารถที่จะทำงานได้หลายๆงานพร้อมๆกัน โดยอาศัยฐานเวลาเป็นตัวกระตุ้นการทำงานของโปรแกรม ที่ระดับการทำงานในชั้นของคลาสไคลเอ็นท์วินโดว์จะได้รับการกระตุ้นของฐานเวลาโดยค่าฐานเวลาสามารถกำหนดได้ภายในโปรแกรม จากนั้นที่ไคลเอนท์วินโดว์ก็จะส่งต่อการทำงานให้กับคลาสอื่น ๆ ที่กำหนดไว้ คือ คลาสโครงข่ายและคลาสแสดงผลตามลำดับ ดังแสดงในภาพที่ 16

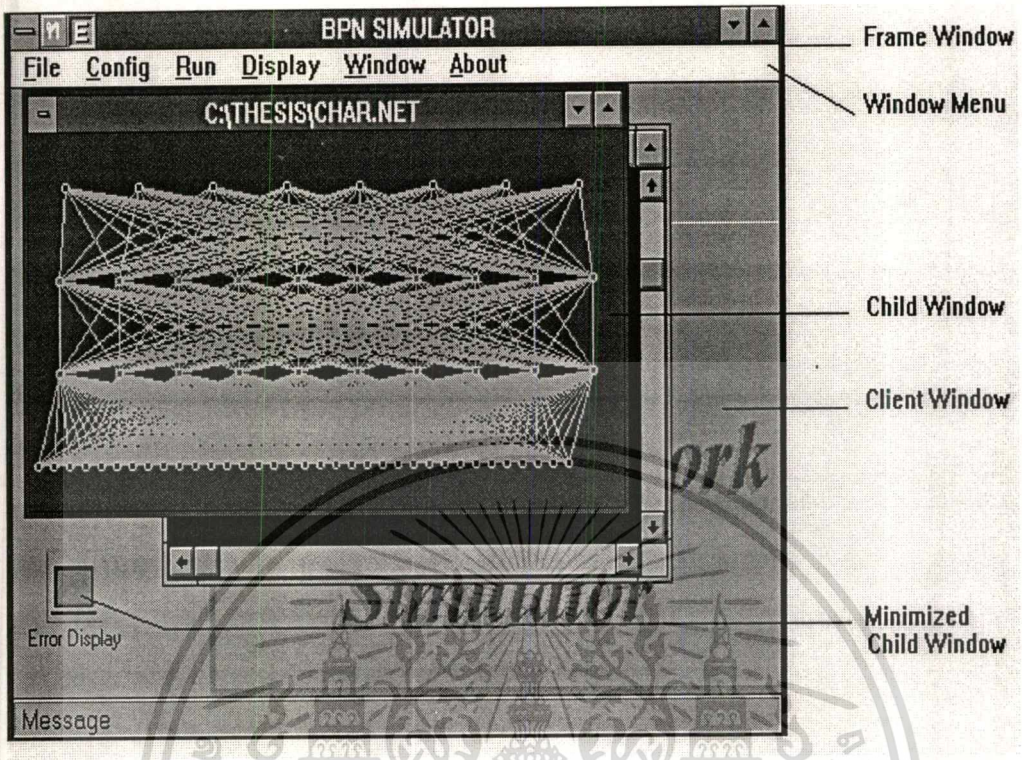
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 16 แสดงการทำงานโดยอาศัยเทคนิคเอ็มดีไอ

การทำงานที่ชั้นของคลาสไคลเอ็นท์วินโดว์จะทำหน้าที่ในการกำหนดเฟรมวินโดว์ (frame window) ซึ่งภายในเฟรมวินโดว์จะมีพื้นที่การใช้งานโดยแบ่งเป็นส่วนของเมนูและพื้นที่ใช้งานที่มีลักษณะเหมือนกับเดสท็อป (desktop) ซึ่งเป็นพื้นที่ทำงานของไชด์วินโดว์ (child window) หรือหน้าต่างลูกหรืออาจจะใช้พื้นที่นี้ในการแสดงผลด้วยก็ได้ ภายในคลาสไคลเอนท์วินโดว์จะบรรจุตัวแปรและฟังก์ชั่นสมาชิกที่ตอบสนองต่อการทำงานในชั้นนี้ โดยฟังก์ชั่นส่วนใหญ่จะเป็นฟังก์ชั่นการทำงานที่ตอบสนองต่อเมสเสจของระบบปฏิบัติการวินโดว์และฟังก์ชั่นที่ตอบสนองต่อการทำงานของเมนู การเรียกใช้คลาสย่อยจะดำเนินการในชั้นคลาสไคลเอนท์วินโดว์นี้ โดยคลาสย่อยที่ถูกเรียกใช้ก็จะมีการทำงานหรือโปรเซสของคลาสย่อยเองซึ่งได้รับเวลาในการทำงานจากคลาสไคลเอ็นท์วินโดว์โดยอาศัยเมจเสจ WM_TIMER ที่ส่งมาจากระบบปฏิบัติการวินโดว์เป็นตัวกระตุ้นการทำงานให้แก่คลาสย่อยต่าง ๆ ที่กำหนดไว้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 17 แสดงส่วนประกอบหลักของโปรแกรมที่เป็น MDI

การทำงานที่ชั้นของคลาสโครงข่ายนั้นจะทำงานได้ก็ต่อเมื่อมีการกำหนดให้คลาสโครงข่ายทำงาน การกำหนดนั้นกระทำในชั้นของคลาสไคลเอ็นท์วินโดว์โดยที่ภายในคลาสโครงข่ายจะบรรจุตัวแปรและฟังก์ชั่นที่ทำงานเกี่ยวกับการจำลองโครงข่ายประสาทเทียมจะมีฟังก์ชั่นหนึ่งที่ถูกเรียกทำงานจากชั้นคลาสไคลเอ็นท์วินโดว์โดยอาศัยเมจเสจ WM_TIMER จากระบบปฏิบัติการวินโดว์ ฟังก์ชั่นนี้มีชื่อว่า TimerTick ซึ่งเป็นฟังก์ชั่นหลักในการทำงานจำลองโครงข่ายประสาทเทียม เมื่อคลาสโครงข่ายถูกกำหนดให้มีการทำงานแล้วคลาสโครงข่ายจะกลายเป็นไชด์(Child) หรือลูกของคลาสไคลเอ็นท์วินโดว์และก็จะมีส่วนแสดงผลเป็นของตนเองคือ ไชด์วินโดว์ ลักษณะการทำงานของระบบปฏิบัติการวินโดว์จะมีลักษณะเป็นเมจเสจไดรเว่น (message driven) ดังนั้นที่คลาสโครงข่ายก็จะมีส่วนของการรับเมจเสจเช่นเดียวกับคลาสไคลเอ็นท์วินโดว์เช่นกัน การแสดงข้อมูลขณะที่คลาสโครงข่ายกำลังทำงานจึงอาศัยเมจเสจ WM_PAINT เป็นหลัก

ขั้นตอนของการกำหนดคลาสโครงข่ายในชั้นของคลาสไคล์เอ็นท์วินโดว์แสดงด้วยโปรแกรมดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
void ClientWnd :: DefineNetworkClass( )
{
  TMDIChild*    child = new  TMDIChild(*this, "Network", new NetworkClass);
  child -> SetIcon (0,0);
  child -> Create ( );
}
```

เมื่อมีเมจเสจ WM_TIMER ในชั้นของคลาสไคลเอ็นท์วินโดว์ ฟังก์ชั่น EvTimer ในชั้นคลาสไคลเอ็นท์วินโดว์ก็จะทำงาน และจะส่งผ่านการทำงานไปให้คลาสย่อยที่กำหนดไว้

```
void ClientWnd :: EvTimer(UINT)
{
  ForEach (ChildTimers,0);
}
```

พารามิเตอร์ ChildTimers คือ ตัวแปรพอยท์เตอร์ชี้ไปยังฟังก์ชั่น ChildTimers ซึ่งมีหน้าที่เรียกฟังก์ชั่น TimerTick ในคลาสย่อยให้ทำงาน

```
void ChildTimers (TWindow* p, void*)
{
  TFameWindow*  frame = TYPESAFE_DOWNCAST(p, TFrameWindow);
  CHECK(frame);
  TBaseDemoWindow*  demoWin  =
        TYPESAFE_DOWNCAST(frame->GetClientWindow(),TBaseDemoWindow);
  CHECK(demoWin);
  demoWin -> TimerTick( );
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำงานของฟังก์ชั่น TimerTick ในคลาสโครงข่ายมีการทำงานคร่าว ๆ ดังนี้

```
void NetworkClass :: TimerTick( )
{
        .
        .
        .
  Artificial  Neural Network Simulation
        .
        .
        .
}
```

ต่อไปจะกล่าวถึงการทำงานในชั้นของคลาสแสดงผลซึ่งในคลาสนี้ทำหน้าที่ในการแสดงผลการทำงานโครงข่ายประสาทเทียมเป็นภาพกราฟิกซึ่งก็จะถูกกำหนดให้ทำงานที่ชั้นคลาสไคลเอ็นท์วินโดว์เช่นเดียวกับคลาสโครงข่ายและก็มีฟังก์ชั่น TimerTick เหมือนกับคลาสโครงข่ายเช่นกัน โดยสามารถกำหนดคลาสแสดงผลในชั้นของคลาสไคล์เอ็นวินโดว์ได้ดังนี้

```
void ClientWnd :: DefineDisplayClass ( )
{
  TMDIChid*  child = new  TMDIChild(*this, "Display", new DisplayClass);
  child -> SetIcon(0,0);
  child -> Create( );
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

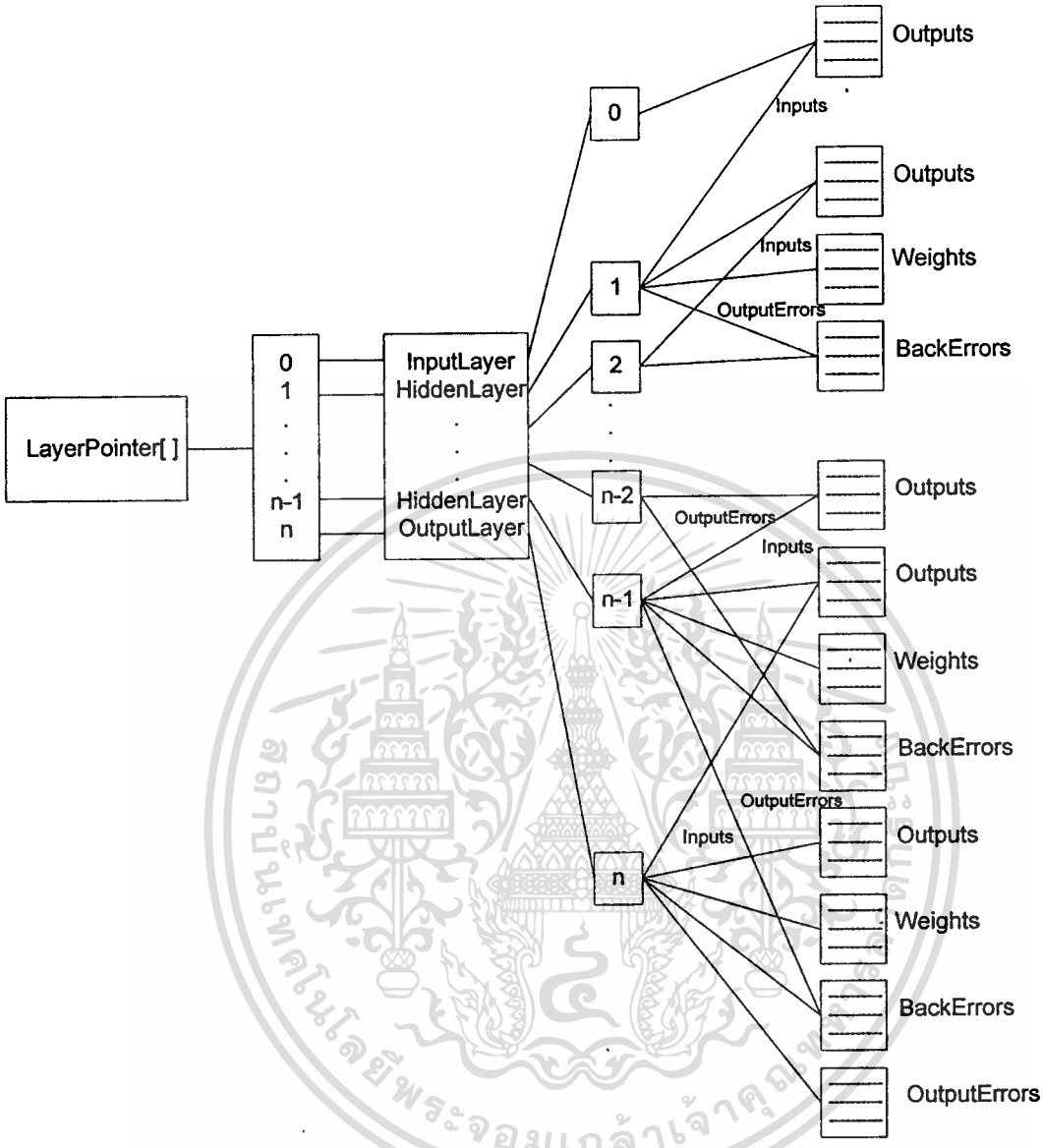
ส่วนการเรียกใช้คลาสแสดงผลให้ทำงานนั้นก็อาศัยฟังก์ชั่น EvTimer และ ChildTimers ที่ได้กล่าวผ่านมาแล้วเรียกฟังก์ชั่น TimerTick ในคลาสแสดงผลให้ทำงาน การทำงานของ TimerTick แสดงคร่าว ๆ ได้ดังนี้

```
void DisplayClass :: TimerTick( )
{
.
.
.
   Artificial Neural Network Simulate Display
.
.
.
}
```

<u>โครงสร้างแบบจำลองโครงข่ายประสาทเทียม</u>

สิ่งสำคัญที่สุดของการออกแบบโปรแกรมจำลองโครงข่ายประสาทเทียมได้แก่การออกแบบโครงสร้างโครงข่ายประสาทเทียม เพราะหมายความว่าถ้าโครงข่ายมีโครงสร้างที่ดีแล้วการออกแบบโปรแกรมก็สามารถทำให้กระชับและง่ายต่อการเขียนโปรแกรม โครงสร้างโครงข่ายที่ออกแบบไว้มีลักษณะดังภาพที่ 18

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 18 แสดงโครงสร้างแบบจำลองโครงข่ายประสาทเทียม

ต่อไปจะขอกล่าวถึงรายละเอียดของโครงสร้างโครงข่ายประสาทเทียมเพื่อที่จะได้เกิดความเข้าใจกับอัลกอริธึมของการเรียนรู้แบบแพร่ย้อนกลับที่จะกล่าวในหัวข้อถัดไป ลักษณะของโครงสร้างโครงข่ายประกอบด้วยตัวแปรหมวดพอยท์เตอร์ชื่อว่า LayerPointer [ ] เป็นตัวแปรที่ใช้เก็บแอดเดรส (address) ของชั้นต่าง ๆ ของโครงข่ายบีพีเอ็น ซึ่งประกอบด้วย ชั้นอินพุท (Input Layer) ชั้นซ่อน (Hidden Layer) ชั้นเอาต์พุท (Output Layer) โดยที่ชั้นอินพุทจะมีตัวแปร Outputs ซึ่งเป็นตัวแปรหมวดเก็บค่าอินพุทเวคเตอร์ที่ป้อนเข้ามายังโครงข่าย ที่ชั้นซ่อนจะมีตัวแปร Inputs ซึ่งเป็นตัวแปรพอยท์เตอร์ชี้ไปยังแอดเดรส Outputs ของชั้นที่อยู่ต่ำกว่า ตัวแปร Outputs เป็นตัวแปร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หมวดใช้เก็บค่าเอาต์พุทของนิวรอนใช้ชั้นซ่อน ตัวแปร Weights เป็นตัวแปรหมวดใช้เก็บค่าน้ำหนักของชั้นซ่อน ตัวแปร BackErrors เป็นตัวแปรหมวดใช้เก็บค่าผิดพลาดเพื่อส่งกลับไปยังชั้นที่อยู่ต่ำกว่า ตัวแปร OutputErrors เป็นตัวแปรพอยท์เตอร์ชี้ไปยังแอสเดรสของตัวแปร BackErrors ของชั้นที่อยู่สูงกว่า ที่ชั้นเอาต์พุทจะมีตัวแปรเช่นเดียวกับชั้นซ่อนจะต่างกันตรงตัวแปร OutputErrors เป็นตัวแปรหมวดของชั้นเอาต์พุทเอง ข้อกำหนดอีกอย่างหนึ่งของโครงสร้างโครงข่ายนี้ก็คือการกำหนดจำนวนชั้นอย่างน้อยที่สุดคือ 3 ชั้น มีอินพุท 1 ชั้น ชั้นซ่อน 1 ชั้น และชั้นเอาต์พุท 1 ชั้น กำหนดจำนวนชั้นได้มากที่สุดคือ 5 ชั้น มีอินพุท 1 ชั้น ชั้นซ่อน 3 ชั้น และชั้นเอาต์พุท 1 ชั้น จะกล่าวถึงการกำหนดนี้ในบทที่ว่าด้วยเรื่องการใช้งานโปรแกรม

<u>อัลกอริธึมในการสร้างส่วนจำลองการทำงานโครงข่ายประสาทเทียม</u>

ในหัวข้อนี้จะกล่าวถึงการสร้างส่วนจำลองการทำงานโครงข่ายประสาทเทียมซึ่งเป็นส่วนสำคัญที่ใช้ในคลาสโครงข่าย อัลกอริธึมที่จะแสดงในหัวข้อนี้ได้มาจากการแปลงหลักการแพร่ย้อนกลับที่อยู่ในรูปแบบทางคณิตศาสตร์มาเป็นรูปแบบที่ทำงานได้โดยโปรแกรมคอมพิวเตอร์ และเนื่องจากโปรแกรมจำลองการทำงานโครงข่ายประสาทเทียมนี้ถูกพัฒนาขึ้นมาจากภาษาซีพลัสพลัส ดังนั้นจึงแสดงอัลกอริทึมการทำงานในรูปแบบของภาษาที่ใช้อยู่นี้ การดูอัลกอริธึมหรือโปรแกรมให้พิจารณาร่วมกับโครงสร้างแบบจำลองโครงข่ายประสาทเทียมที่ได้กล่าวไปในหัวข้อที่แล้วร่วมกับทฤษฎีหลักการแพร่ย้อนกลับเพื่อทำให้ง่ายต่อความเข้าใจ อีกทั้งผู้อ่านต้องมีความรู้ด้านภาษาซีหรือภาษาซีพลัสพลัสมาบ้าง

ก่อนที่จะกล่าวถึงอัลกอริธึมจะของกล่าวถึงการจัดแบ่งหน้าที่หรือลักษณะของการจำลองโครงข่ายประสาทเทียมโดยสามารถแบ่งได้เป็นสองส่วนหลัก ๆ คือ

1. การจำลองในการฝึกหัดโครงข่าย ซึ่งก็คือการ สอนโครงข่ายนั่นเอง
2. การจำลองในการทดสอบโครงข่าย ซึ่งก็คือการทดสอบหาค่าเอาต์พุท จากกลุ่มข้อมูลทดสอบโดยโครงข่ายประสาทเทียมที่ได้รับการฝึกหัดมาแล้ว

วิธีจำลองในการฝึกหัดโครงข่ายสามารถแสดงได้โดยคำสั่งเทียม (pseudo code) โดยมีลำดับขั้นตอนดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
begin
  random weights
  while network error is larger than some predefine limit
  do.
    set input to apply a training input
    forward propagate to generate an outputs
    backward propagate to update error values
    update weights
  end do
  save weights to file
end.
```

วิธีจำลองในการทดสอบโครงข่ายสามารถแสดงได้ดังนี้

```
begin
  set weights value from file
  set input to stimulate the network with an input
  forward propagate to generate an outputs
  save output to file
end.
```

ลำดับต่อไปจะแสดงอัลกอริธึมที่สำคัญสามส่วน คือ foward propagate , backward propagate และ update weights

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อัลกอริธึม forward propagate

ฟังก์ชั่น ForwardPropagate ทำหน้าที่คำนวณค่าเอาต์พุทที่ชั้นซ่อนและชั้นเอาต์พุท ฟังก์ชั่นกระตุ้นที่เลือกใช้คือฟังก์ชั่นไบนารีซิกมอยด์

```
void NetworkClass :: ForwardPropagate ( )
{
  int i;
  for (i=1; i<NumberOfLayers; i++)
    LayerPointer[i] -> CalculateOutput( );
}
void CalculateOutput( )
{
  int i, j, k;
  float Net;
  for(j=0; j<NumberOfOutputs; j++){
    for(i=0; i<NumberOfInputs; i++){
      k = i* NumberOfOutputs;
      Outputs[j] = Weights[k+j] * Inputs[i];
      Net +=Outputs[j];
    }
    Outputs[j] = (float) (1 / 1+exp(-Net));
  }
}
```

อัลกอริธึม backward propagate

ฟังก์ชั่น BackwardPropagate ทำหน้าที่คำนวณค่าผิดพลาดที่ชั้นเอาต์พุทและชั้นซ่อน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
void NetworkClass :: BackwardPropagate ( )
{
  int i;
  for(i=NumberOfLayers; i>0; i--)
    LayerPointer[i] -> CalculateError( );
}
```

ฟังก์ชั่นย่อยที่เรียกใช้คือ CalculateError ซึ่งเป็นฟังก์ชั่นโพลิมอฟิค (polymorphic) ที่ชั้นเอาต์พุทและชั้นซ่อนดังนี้

```
void OutputLayer :: CalculateError( )
{
  int i, j, k;
  float Summation = 0;
  for(j=0; j<NumberOfOutputs; j++)
    OutputErrors[j] = (TargetValue[j] - Outputs[j]) *
                      (Outputs[j]*(1-Outputs[j]));
  for(i=0; i<NumberOfInputs; i++){
    k = i* NumberOfOutputs;
    for(j=0; j<NumberOfOutputs; j++){
      BackErrors[i] = Weights[k+j] * OutputErrors[j];
      Summation += BackErrors[i];
    }
    BackErrors[i] = Summation;
    Summation=0;
    BackError[i] *= Inputs[i] * (1-Inputs[i]);
  }
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
void HiddenLayer :: CalculateError ( )
{
  int i, j, k;
  float  Summation = 0;
  for(i=0; i<NumberOfInputs; i++){
    k = i* NumberOfOutputs;
    for(j=0; j<NumberOfOutputs; j++){
      BackErrors[i] = Weights[k+j] * OutputErrors[j];
      Summation += BackErrors[i];
    }
    BackErrors[i] = Summation;
    Summation = 0;
    BackErrors[i] *= Inputs[i] * (1-Inputs[i]);
  }
}
```

อัลกอริธึม update weights

ฟังก์ชั่น UpdateWeights ทำหน้าที่ในการปรับค่าน้ำหนักที่ชั้นซ่อนและชั้นเอาต์พุท

```
void NetworkClass :: UpdateWeights ( )
{
  int i;
  for(i=1; i<NumberOfLayers; i++)
    LayerPointer[i] -> Update(Eta);
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
void Update(float Eta)
{
  int i, j, k;
  for(i=0; i<NumberOfInputs; i++){
    k = i* NumberOfOutputs;
    for(j=0; j<NumberOfOutputs; j++)
      Weights[k+j] += Eta * OutputErrors[j] * Inputs[i];
  }
}
```

<u>บทสรุป</u>

ขั้นตอนของการออกแบบโปรแกรมที่ผ่านมานั้นได้กล่าวเฉพาะในส่วนของการจำลองโครงข่ายประสาทเทียมโดยดัดแปลงรูปแบบทางคณิตศาสตร์ในบทที่ 2 เป็นอัลกอริธึมเชิงโปรแกรมและได้ข้ามรายละเอียดการออกแบบในส่วนอื่น ๆ ไป เช่นในส่วนของภาคแสดงผลที่อินเตอร์เฟสกับผู้ใช้เพราะว่าไม่ใช่ประเด็นตามหัวข้อของการวิจัยอีกทั้งมีรายละเอียดข้อปลีกย่อยค่อนข้างมากถ้านำมากล่าวทั้งหมดคงจะเป็นการกล่าวเทคนิคการเขียนโปรแกรมมากกว่าที่จะเป็นเนื้อหาตามจุดมุ่งหมายของงานวิจัย ในบทที่ 4 หรือบทถัดไปจะกล่าวถึงการใช้งานโปรแกรมจำลองโครงข่ายประสาทเทียมที่ได้สร้างขึ้นมาโดยจะแสดงการใช้งานเป็นขั้นตอนซึ่งผู้อ่านสามารถปฏิบัติตามขั้นตอนนี้ได้โดยสะดวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4
การใช้งานโปรแกรมจำลองโครงข่ายประสาทเทียม

ในบทนี้จะกล่าวถึงการใช้งานโปรแกรมจำลองโครงข่ายประสาทเทียมที่ได้สร้างขึ้นมา โดยแบ่งลักษณะการใช้ตามชนิดของการจำลองคือโปรแกรมจำลองอะดาไลน์และโปรแกรมจำลองโครงข่ายประสาทเทียมโดยใช้หลักการแพร่ย้อนกลับ ดังจะได้กล่าวตามลำดับต่อไป

การใช้งานโปรแกรมจำลองอะดาไลน์

หัวข้อนี้จะกล่าวถึงวิธีการใช้โปรแกรมจำลองอะดาไลน์ที่ได้สร้างขึ้นมา การใช้งานจะแบ่งออกเป็น 2 ลักษณะคือใช้เพื่อการฝึกหัดหน่วยอะดาไลน์และใช้เพื่อการทดสอบหาผลลัพธ์ทางเอาต์พุทของหน่วยอะดาไลน์โดยจะกล่าวถึงรายละเอียดถึงวิธีการใช้ของแต่ละลักษณะในหัวข้อถัดไป

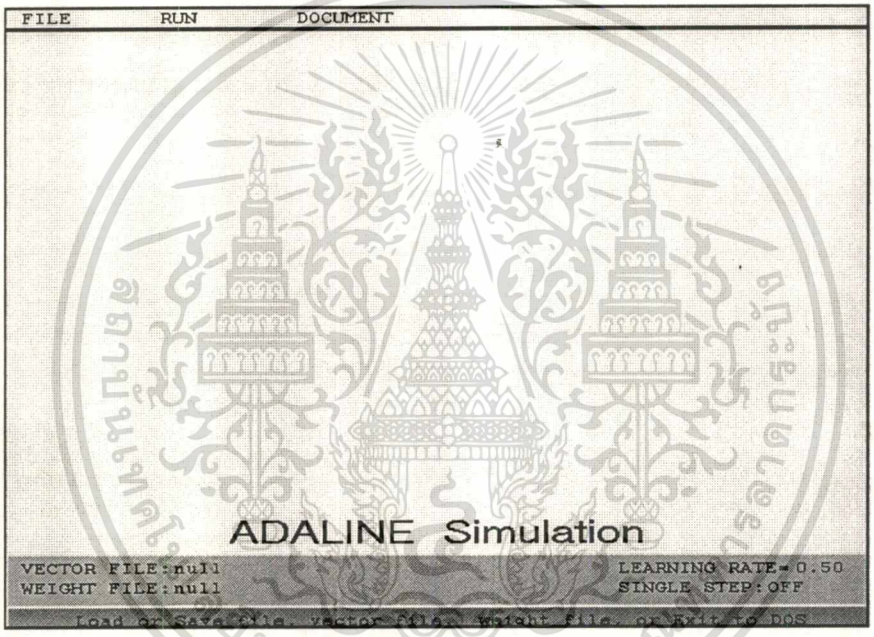
การฝึกหัดอะดาไลน์

การใช้งานโปรแกรมจำลองอะดาไลน์ในการฝึกหัดนั้นก็คือ การนำกลุ่มข้อมูลฝึกหัดซึ่งประกอบด้วยคู่ลำดับ อินพุท-เอาต์พุทเวคเตอร์ (input-output vector) ป้อนให้กับหน่วยอะดาไลน์ เพื่อให้หน่วยอะดาไลน์ทำการเรียนรู้และก่อนที่จะใช้งานโปรแกรมเพื่อฝึกหัดนั้นจะต้องมีการสร้างแฟ้มข้อมูลฝึกหัดเพื่อใช้ในการฝึกหัดหน่วยอะดาไลน์ (training file) จะขอยกตัวอย่างแฟ้มและข้อมูลฝึกหัดโดยให้ชื่อว่า TRAIN3 ซึ่งมีข้อมูลกำหนดโครงสร้างและข้อมูลฝึกหัดดังนี้

```
3  <-------------------  กำหนดจำนวนอินพุทของอะดาไลน์
5  <-------------------  กำหนดจำนวนของข้อมูลฝึกหัดหรือจำนวนเวคเตอร์นั่นเอง
10      10      10      1
10      20      30      1
20      10      20      1
30      20      10      -1
20      30      10      -1
↑_____↑____↑   ↑__  กำหนดค่าเป้าหมาย
      |
กำหนดอินพุทเวคเตอร์
```

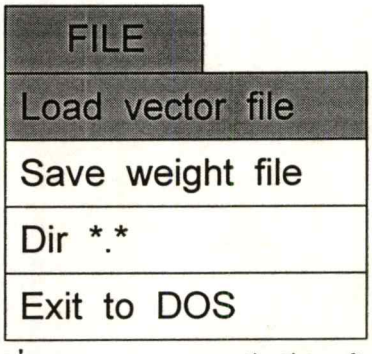
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากข้อมูลภายในแฟ้ม TRAIN3 ได้กำหนดให้ข้อมูลแรกคือ 3 เป็นค่ากำหนดจำนวนอินพุทแก่หน่วยอะดาไลน์ซึ่งข้อมูลนี้มีค่าได้ตั้งแต่ 1-19 และข้อมูลลำดับถัดมาคือ 5 เป็นจำนวนเวคเตอร์ที่ใช้ฝึกหัดหน่วยอะดาไลน์ ข้อมูลที่เหลือคือ อินพุท-เอาต์พุทเวคเตอร์โดยคอลัมท์ที่ 1 คอลัมท์ที่ 2 และคอลัมท์ที่ 3 เป็นอินพุทเวคเตอร์ส่วนคอลัมท์ที่ 4 เป็นค่าเอาต์พุทเป้าหมาย (target) หลังจากได้สร้างแฟ้มข้อมูลฝึกหัด TRAIN3 แล้วก็พร้อมที่จะใช้งานโปรแกรมจำลองอะดาไลน์ การเรียกใช้โปรแกรมสามารถเรียกได้ภายใต้ระบบปฏิบัติการของดอสโดยเรียกโปรแกรม ADASIM.EXE เมื่อโปรแกรมเริ่มทำงานจะปรากฏภาพขึ้นที่จอภาพดังนี้



ภาพที่ 19 การแสดงผลเริ่มต้นของโปรแกรม

จากนั้นให้เลือกการทำงานที่หัวข้อ FILE ที่เมนูบาร์จะปรากฏเป็นพูลดาวน์เมนูให้เลือก Load vector file เพื่อเปิดแฟ้มข้อมูลฝึกหัด



ภาพที่ 20 แสดงการเปิดแฟ้มข้อมูลฝึกหัด

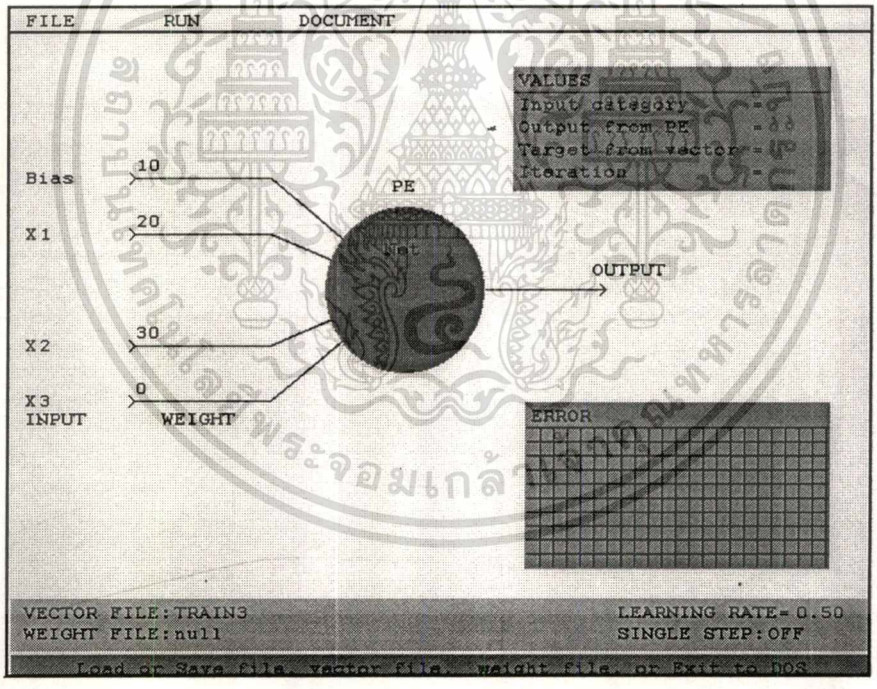
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อเลือก Load vector file แล้วจะปรากฏช่องว่างเพื่อให้ใส่ชื่อแฟ้มข้อมูลให้คีย์ชื่อแฟ้มข้อมูลฝึกหัด TRAIN3 ที่ได้สร้างขึ้นไว้แล้ว จากนั้นกดคีย์ ENTER.



ภาพที่ 21 ใส่ชื่อแฟ้มข้อมูลฝึกหัด

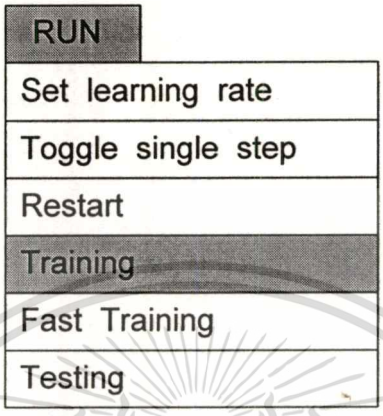
โปรแกรมจำลองอะดาไลน์จะสร้างภาพของหน่วยอะดาไลน์และช่องหน้าต่างรายงานผล ซึ่งมี 2 หน้าต่างคือหน้าต่างแวลู(VALUES)ใช้แสดงรายละเอียดของการทำงานและหน้าต่างผิดพลาด (ERROR) ใช้แสดงกราฟรายงานผลผิดพลาด



ภาพที่ 22 แสดงหน่วยอะดาไลน์ที่โปรแกรมสร้างขึ้น

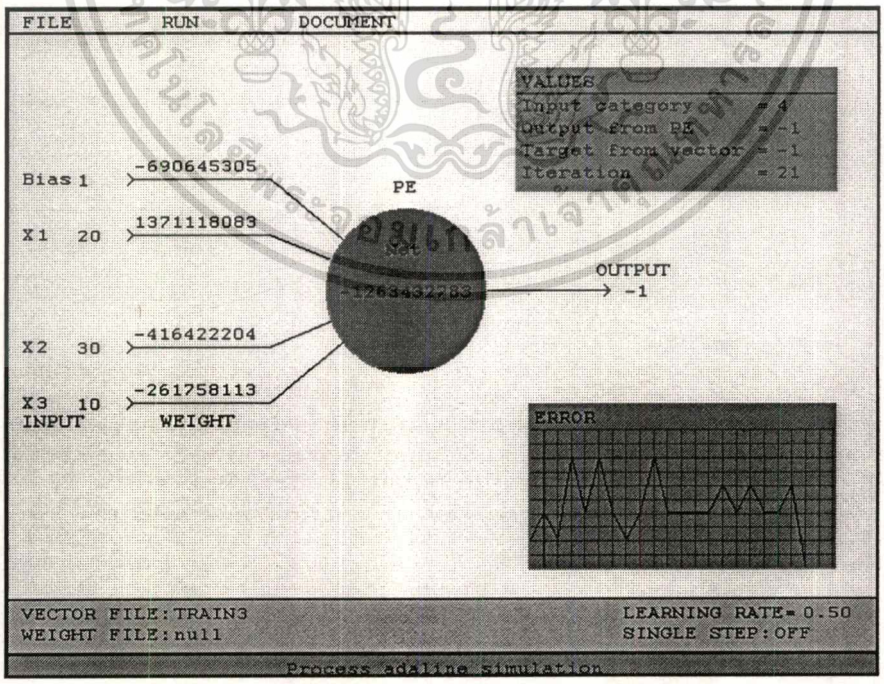
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ลำดับต่อไปจะทำการฝึกหัดหน่วยอะดาไลน์ โดยเลือกการทำงานที่หัวข้อ RUN บนเมนูบาร์แล้วเลือก Training



ภาพที่ 23 แสดงการเลือกฝึกหัดอะดาไลน์

โปรแกรมจะเริ่มทำงานในการเรียนรู้โดยอาศัยข้อมูลฝึกหัด ในระหว่างการฝึกหัดนั้นจะเห็นค่าต่าง ๆ เปลี่ยนแปลงปรากฏอยู่บนจอภาพ



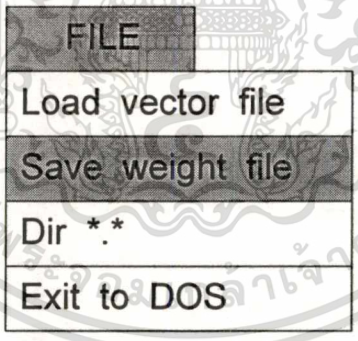
ภาพที่ 24 แสดงค่าต่าง ๆ เปลี่ยนแปลงขณะทำการฝึกหัด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากภาพที่ 24 จะเห็นการเปลี่ยนแปลงของค่าต่างๆ ที่หน่วยอะดาไลน์จะมีอินพุทเวคเตอร์ต่อผ่านน้ำหนักมายังหน่วยประมวลผล ซึ่งจะแสดงค่าอินพุท ค่าน้ำหนัก ค่าเน็ทและค่าเอาต์พุทของหน่วยอะดาไลน์ ที่หน้าต่างแวลูจะแสดง Input category หรือเป็นค่าแสดงลำดับของอินพุทเวคเตอร์ที่ป้อนให้แก่หน่วยอะดาไลน์ Output from PE แสดงค่าเอาต์พุทของหน่วยอะดาไลน์ Target from vector แสดงค่าเป้าหมายที่ต้องการของลำดับเวคเตอร์นั้น Iteration แสดงจำนวนรอบที่ทำการฝึกหัดอยู่ ส่วนหน้าต่างผิดพลาดจะแสดงกราฟของค่าผิดพลาดขณะทำการฝึกหัดหน่วยอะดาไลน์

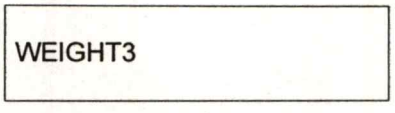
การฝึกหัดอะดาไลน์ นั้นสามารถจะตั้งค่าอัตราการเรียนรู้ได้โดยเลือก RUN บนเมนูบาร์ แล้วเลือก Set learning rate ซึ่งค่านี้สามารถกำหนดได้ในช่วงระหว่าง 0 ถึง 1.0 นอกจากนั้นยังสามารถให้การฝึกหัดกระทำในลักษณะซิงเกิลสเตปได้โดยเลือก Toggle single step หรือทำให้การฝึกหัดเป็นไปในลักษณะรวดเร็วได้โดยเลือก Fast Training ส่วน Restart เป็นการเริ่มต้นการฝึกหัดใหม่

เมื่อการฝึกหัดสิ้นสุดลง สามารถทำการจัดเก็บค่าน้ำหนักได้โดยเลือก FILE บนเมนูบาร์ แล้วเลือก Save weight file



ภาพที่ 25 แสดงการเลือกจัดเก็บค่าน้ำหนัก

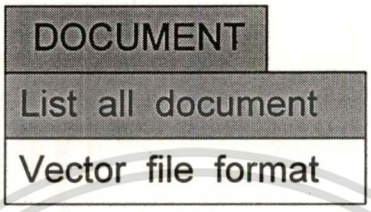
จะปรากฏช่องว่างให้คีย์ชื่อแฟ้มข้อมูล WEIGHT3 เพื่อจัดเก็บค่าน้ำหนักลงในแฟ้มนี้



ภาพที่ 26 ใส่ชื่อแฟ้มข้อมูลน้ำหนัก

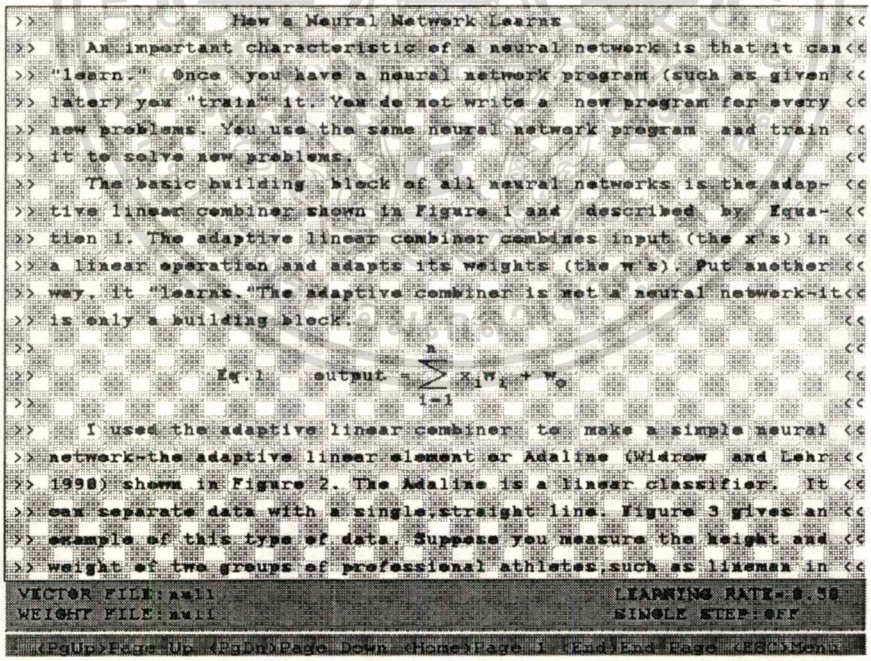
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมนูบาร์ DOCUMENT จะเป็นเมนูที่ทำหน้าที่อธิบายรายละเอียดทฤษฎีอะดาไลน์และแฟ้มข้อมูล โดยแบ่งเป็นเมนูย่อยคือ List all document และ Vector file format ถ้าต้องการดูรายละเอียดทฤษฎีอะดาไลน์ให้เลือก List all document ถ้าต้องการดูรายละเอียดของรูปแบบแฟ้มข้อมูลฝึกหัดและแฟ้มข้อมูลทดสอบให้เลือก Vector file format



ภาพที่ 27 แสดงการเลือกดูรายละเอียดทฤษฎีอะดาไลน์

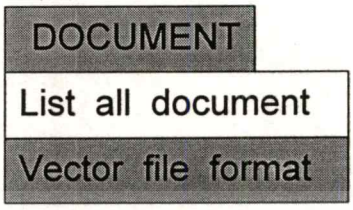
โปรแกรมจะแสดงรายละเอียดทฤษฎีอะดาไลน์ซึ่งสามารถดูหน้าต่างๆได้โดยกดคีย์เพจอัพ(PgUp)หรือเพจดาวน์(PgDn) เมื่อต้องการกลับสู่การทำงานตามปรกติให้กดคีย์เอสเคป(Esc)



ภาพที่ 28 แสดงรายละเอียดทฤษฎีอะดาไลน์

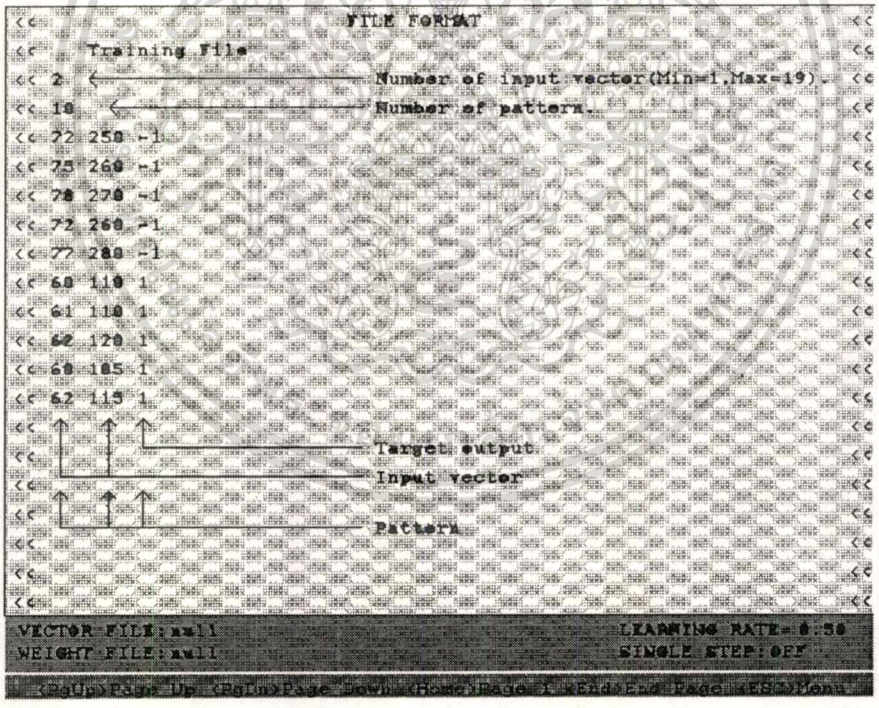
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เลือกเมนู Vector file format เมื่อต้องการดูรายละเอียดรูปแบบแฟ้มข้อมูลฝึกหัดและแฟ้มข้อมูลทดสอบ



ภาพที่ 29 แสดงการเลือกดูรายละเอียดรูปแบบแฟ้มข้อมูล

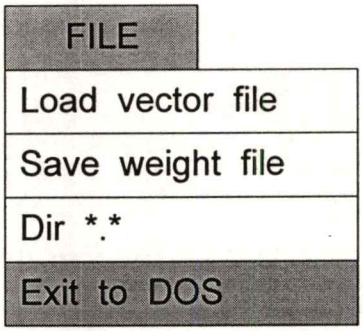
โปรแกรมจะแสดงรายละเอียดรูปแบบแฟ้มข้อมูลซึ่งสามารถดูหน้าต่างๆได้โดยกดคีย์เพจอัพ(PgUp)หรือเพจดาวน์(PgDn) เมื่อต้องการกลับสู่การทำงานตามปรกติให้กดคีย์เอสเคป(Esc)



ภาพที่ 30 แสดงรายละเอียดรูปแบบแฟ้มข้อมูล

เมื่อต้องการออกจากโปรแกรมจำลองอะดาไลน์ให้เลือก FILE บนเมนูบาร์แล้วเลือก Exit to DOS

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 31 แสดงการเลือกออกจากโปรแกรม

การทดสอบหาผลลัพธ์ทางเอาต์พุทของหน่วยอะดาไลน์

การทดสอบหาผลลัพธ์ทางเอาต์พุทของหน่วยอะดาไลน์นั้นก็คือ การนำหน่วยอะดาไลน์ที่ผ่านการฝึกหัดแล้วมาใช้กับกลุ่มข้อมูลทดสอบเพื่อดูผลลัพธ์ทางเอาต์พุท ก่อนที่จะใช้งานโปรแกรมเพื่อการทดสอบนั้น จะต้องมีการเตรียมการล่วงหน้าก่อน สิ่งที่ต้องจัดเตรียมคือ

1. เตรียมแฟ้มข้อมูลน้ำหนัก (weight file)
2. สร้างแฟ้มข้อมูลทดสอบหน่วยอะดาไลน์ (testing file)

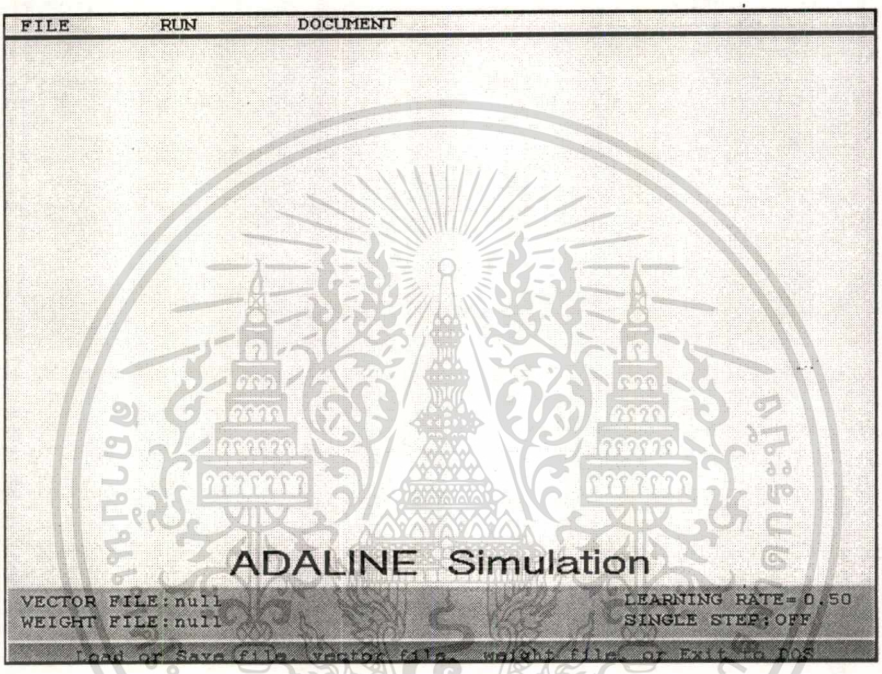
จากการฝึกหัดหน่วยอะดาไลน์ในหัวข้อที่ผ่านมาได้ทำการจัดเก็บค่าน้ำหนักไว้ที่แฟ้มข้อมูลน้ำหนักโดยให้ชื่อว่า WEIGHT3 ในการทดสอบก็จะใช้แฟ้มข้อมูลน้ำหนักนี้ ส่วนแฟ้มที่ต้องสร้างขึ้นใหม่นั้นคือแฟ้มข้อมูลทดสอบ ภายในแฟ้มข้อมูลนี้จะมีข้อมูลกำหนดโครงสร้างของหน่วยอะดาไลน์และข้อมูลทดสอบเฉพาะส่วนที่เป็นอินพุท ไม่ต้องมีส่วนที่เป็นค่าเอาต์พุทหรือค่าเป้าหมาย เพราะจะให้หน่วยอะดาไลน์ทำการหาค่าเอาต์พุทให้ จะขอยกตัวอย่างแฟ้มข้อมูลทดสอบชื่อว่า TEST3 โดยกำหนดให้มีข้อมูลภายในแฟ้มดังนี้

```
3   <-------- กำหนดจำนวนอินพุทของอะดาไลน์
5   <-------- กำหนดจำนวนของข้อมูลทดสอบหรือจำนวนเวคเตอร์นั่นเอง
10    10    10
10    20    30
20    10    20
30    20    10
20    30    10
↑_____↑____↑
      |
กำหนดอินพุทเวคเตอร์
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

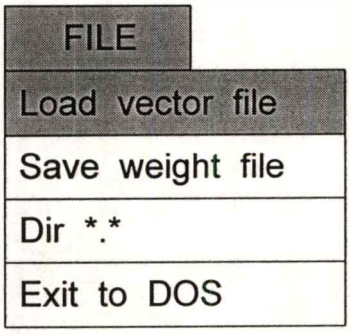
ข้อมูลภายในแฟ้ม TEST3 ได้กำหนดให้ข้อมูลแรกคือ 3 เป็นค่ากำหนดจำนวนอินพุทแก่หน่วยอะดาไลน์ และข้อมูลถัดไปคือ 5 เป็นจำนวนเวคเตอร์ที่ใช้ทดสอบหน่วยอะดาไลน์ข้อมูลที่เหลือคือ อินพุทเวคเตอร์ โดยทั้ง 3 คอลัมท์คือข้อมูลอินพุทที่จะนำไปทดสอบหาค่าเอาต์พุท

เมื่อได้แฟ้มต่างๆครบแล้วก็พร้อมที่จะใช้โปรแกรมจำลองอะดาไลน์ ให้เรียกโปรแกรม ADASIM.EXE เมื่อโปรแกรมเริ่มทำงานจะปรากฏภาพขึ้นที่จอภาพดังนี้



ภาพที่ 32 การแสดงผลเริ่มต้นของโปรแกรม

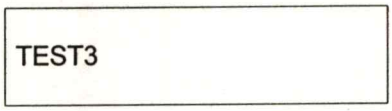
จากนั้นให้เลือกการทำงานที่หัวข้อ FILE บนเมนูบาร์จะปรากฏพูลดาวน์เมนูให้เลือก Load vector file



ภาพที่ 33 แสดงการเปิดแฟ้มข้อมูลทดสอบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะปรากฏช่องว่างใช้คีย์ชื่อแฟ้มข้อมูลทดสอบ TEST3 แล้วกดคีย์ ENTER.



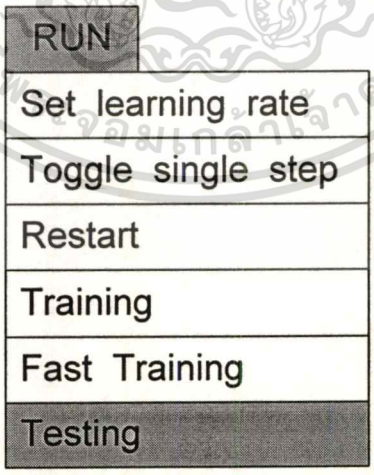
ภาพที่ 34 ใส่ชื่อแฟ้มข้อมูลทดสอบ

ก็จะปรากฏช่องว่างขึ้นอีกให้คีย์ชื่อแฟ้มข้อมูลน้ำหนัก WEIGHT3 แล้วกดคีย์ ENTER.



ภาพที่ 35 ใส่ชื่อแฟ้มข้อมูลน้ำหนัก

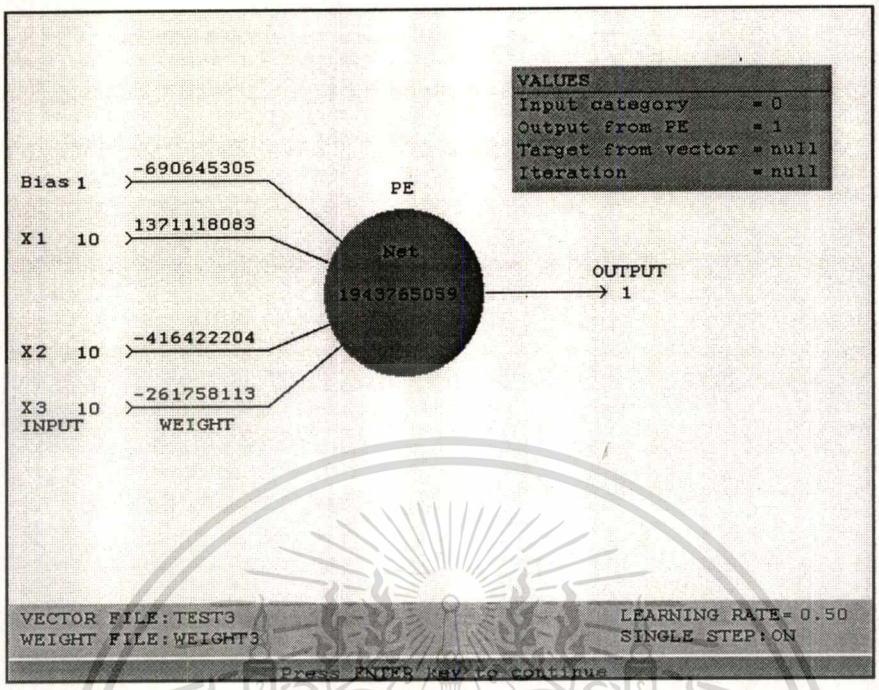
ลำดับต่อไปจะทำการทดสอบหาค่าเอาต์พุทของหน่วยอะดาไลน์โดยเลือกการทำงานที่หัวข้อ RUN บนเมนูบาร์แล้วเลือก Toggle single step เพื่อให้การทำงานเป็นไปทีละลำดับเวคเตอร์ แล้วจึงเลือก Testing



ภาพที่ 36 แสดงการเลือกทดสอบหน่วยอะดาไลน์

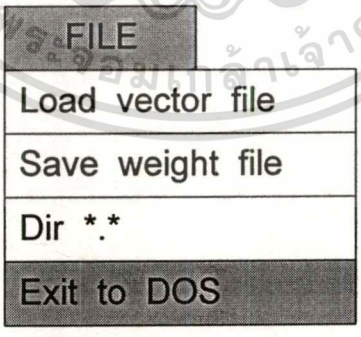
โปรแกรมจะเริ่มทำงานในการทดสอบหาค่าเอาต์พุทของหน่วยอะดาไลน์โดยแสดงภาพที่จอภาพดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 37 แสดงหน่วยอะดาไลน์ที่โปรแกรมสร้างขึ้น

บนจอภาพจะมีรายละเอียดต่างๆ ครบถ้วนเช่น ค่าอินพุท ค่าน้ำหนัก ส่วนที่ช่องหน้าต่างแวลู จะแสดงค่าเอาต์พุทของหน่วยอะดาไลน์ สามารถดูค่าที่เอาต์พุทตามลำดับอินพุทเวคเตอร์ไปได้เรื่อยๆด้วยการกดคีย์ ENTER เมื่อต้องการออกจากโปรแกรมให้เลือก FILE บนเมนูบาร์แล้วเลือก Exit to DOS



ภาพที่ 38 แสดงการเลือกออกจากโปรแกรม

ในหัวข้อถัดไปจะกล่าวถึงการใช้งานโปรแกรมจำลองโครงข่ายประสาทเทียมแบบแพร่ย้อนกลับที่เป็นโครงข่ายประกอบด้วยนิวรอนหลายนิวรอนต่อเชื่อมโยงกันระหว่างชั้นต่างๆ โดยการใช้งานจะกล่าวเป็นขั้นตอนเช่นเดียวกันกับการใช้งานโปรแกรมจำลองอะดาไลน์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การใช้งานโปรแกรมจำลองการทำงานโครงข่ายประสาทเทียมแบบแพร่ย้อนกลับ

ลำดับต่อไปจะกล่าวถึงวิธีการใช้โปรแกรมจำลองโครงข่ายประสาทเทียมที่ได้สร้างขึ้นมา การใช้งานจะแบ่งออกเป็น 2 ลักษณะคือใช้เพื่อการฝึกหัดโครงข่ายประสาทเทียม และใช้เพื่อการทดสอบหาผลลัพธ์ทางเอาต์พุทของโครงข่ายประสาทเทียม โดยจะกล่าวรายละเอียดถึงวิธีการใช้ในแต่ละลักษณะในหัวข้อถัดไป

การฝึกหัดโครงข่ายประสาทเทียม

การใช้งานโปรแกรมจำลองโครงข่ายประสาทเทียมในการฝึกหัดนั้นก็คือ การนำกลุ่มข้อมูลฝึกหัดซึ่งประกอบด้วยคู่ลำดับ อินพุท-เอาต์พุทเวคเตอร์ ป้อนให้กับโครงข่ายเพื่อให้โครงข่ายทำการเรียนรู้และก่อนที่จะใช้งานโปรแกรมเพื่อการฝึกหัดนั้นจะต้องมีการเตรียมการล่วงหน้าก่อนสิ่งที่ต้องตระเตรียมคือ

1. สร้างแฟ้มข้อมูลโครงสร้างโครงข่ายประสาทเทียม (structure file)
2. สร้างแฟ้มข้อมูลฝึกหัดโครงข่ายประสาทเทียม (training file)

การกำหนดหรือสร้างแฟ้มข้อมูลโครงสร้างที่มีนามสกุล .NET นั้นก็เพื่อกำหนดโครงสร้างของโครงข่ายประสาทเทียมที่ต้องการ โดยขอตัวอย่างแฟ้มที่สร้างขึ้นให้ชื่อว่า XOR.NET มีข้อมูลกำหนดโครงสร้างดังนี้

```
LAYER          : 1          ← กำหนดชั้นอินพุทของโครงข่าย
NODE           : 2          ← กำหนดจำนวนหน่วยอินพุท
LAYER          : 2          ← กำหนดชั้นซ่อนของโครงข่าย
NODE           : 5          ← กำหนดจำนวนนิวรอนในชั้นซ่อน
LAYER          : 3          ← กำหนดชั้นเอาต์พุทของโครงข่าย
NODE           : 1          ← กำหนดจำนวนนิวรอนในชั้นเอาต์พุท
LEARN-RATE          : 0.5       ← กำหนดอัตราการเรียนรู้
ERROR-TOLERANCE     : 0.01      ← กำหนดค่าผิดพลาดต่ำสุดที่ยอมรับได้
MOMENTUM            : 0.0← กำหนดเทอมโมเมนตัม
MAX-CYCLE           : 10000     ← กำหนดจำนวนรอบที่ใช้ฝึกหัด
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ความหมายของข้อมูลกำหนดโครงสร้างมีดังนี้

LAYER: n ใช้กำหนดลำดับชั้นของโครงข่าย

NODE : x ใช้กำหนดจำนวนของอินพุทถ้าชั้นที่ระบุก่อนหน้าคือชั้นที่ 1 และใช้กำหนดจำนวนนิวรอนถ้าชั้นที่ระบุก่อนหน้าไม่ใช้ชั้นที่ 1 ซึ่งหมายถึงชั้นที่ระบุก่อนหน้าอาจจะเป็นชั้นซ่อนหรือชั้นเอาต์พุท

LEARN-RATE : f ให้กำหนดอัตราการเรียนรู้โดยที่ f มีค่าได้ตั้งแต่ 0.01 ถึง 1.0

ERROR-TOLERANCE : e ใช้กำหนดค่าผิดพลาดต่ำสุดที่ยอมรับได้ในการฝึกหัดโครงข่าย โดยที่ e สามารถกำหนดค่าได้ในช่วง 0.001 ถึง 100.0

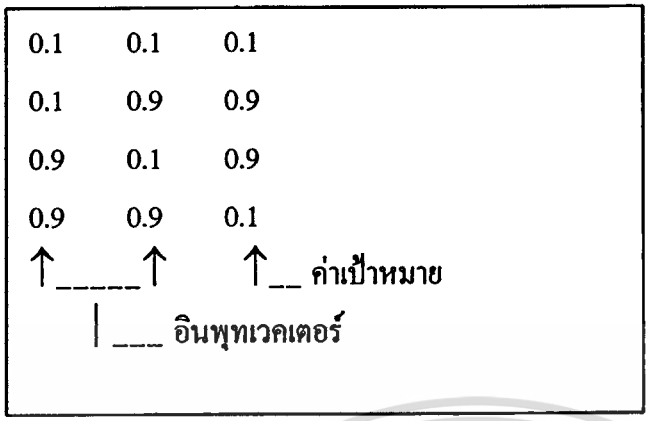
MOMENTUM : a ใช้กำหนดค่าโมเมนตัมโดยที่ a สามารถกำหนดได้ตั้งแต่ 0.0 ถึง 1.0 ถ้ากำหนดให้มีค่าเป็น 0.0 หมายถึงการเรียนรู้จะไม่นำเทอมโมเมนตัมมาใช้ในการปรับแต่งน้ำหนักของโครงข่าย

MAX-CYCLE : m ใช้กำหนดจำนวนรอบสูงสุดในการเรียนรู้โดยที่ m สามารถกำหนดได้ตั้งแต่ 1 ถึง 1,000,000

จำนวนชั้นต่ำสุดของการกำหนดจำนวนชั้นของโครงข่ายจะต้องไม่ต่ำกว่า 3 ชั้น และไม่เกิน 5 ชั้น ซึ่งเป็นข้อกำหนดของโปรแกรมจำลองโครงข่ายประสาทเทียม จากแฟ้มข้อมูลโครงสร้าง XOR.NET ที่ได้สร้างขึ้นนี้ มีคุณสมบัติดังนี้ ที่ชั้นที่ 1 (LAYER : 1) หรือชั้นอินพุท มีจำนวนอินพุท 2 อินพุท (NODE : 2) ชั้นที่ 2 (LAYER : 2) หรือชั้นซ่อนมีจำนวนนิวรอน 5 นิวรอน (NODE : 5) ที่ชั้นที่ 3 (LAYER : 3) หรือชั้นเอาต์พุทมีจำนวนนิวรอน 1 นิวรอน (NODE : 1) ดังนั้นจึงเป็นโครงข่ายประสาทเทียม 3 ชั้น และกำหนดให้มีอัตราการเรียนรู้เท่ากับ 0.5 (LEARN-RATE : 0.5) มีค่าผิดพลาดเฉลี่ยต่ำสุด 0.01 (ERROR-TOLERANCE : 0.01) ค่าผิดพลาดเฉลี่ยนี้ก็คือค่าผิดพลาดเฉลี่ยที่เอาต์พุท ต่อ 1 เวคเตอร์เทียบกับจำนวนเวคเตอร์ทั้งหมดที่ใช้ฝึกหัดโครงข่าย กำหนดค่าโมเมนตัมเป็น 0.0 (MOMENTUM : 0.0) และจำนวนรอบที่ใช้ฝึกหัดสูงสุดคือ 1000 รอบ (MAX-CYLLE : 1000)

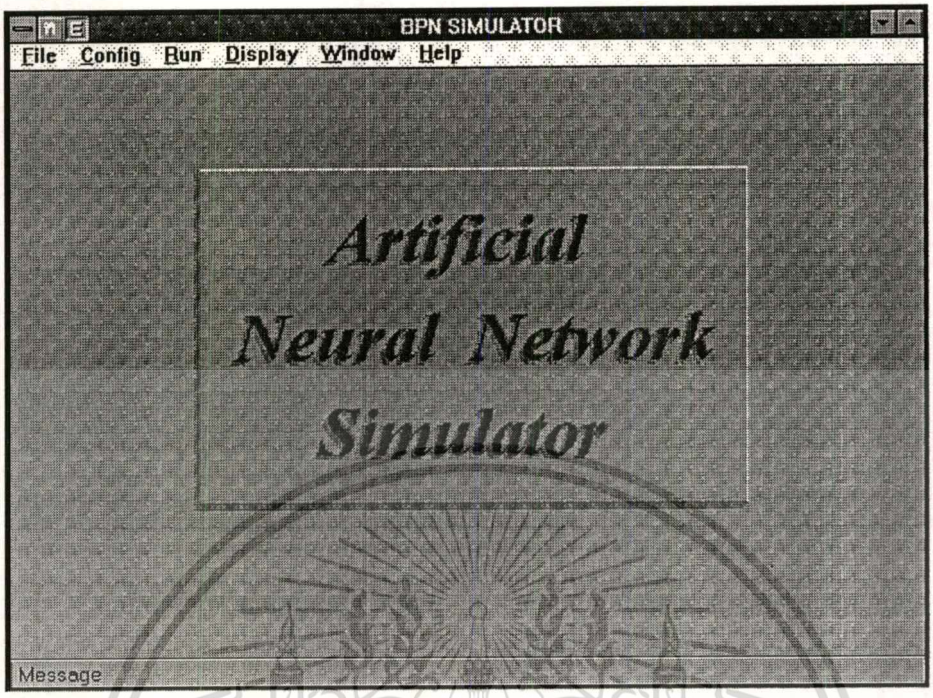
การกำหนดหรือสร้างแฟ้มข้อมูลอีกแฟ้มหนึ่งคือแฟ้มข้อมูลฝึกหัด ที่มีนามสกุล .TRN ภายในแฟ้มนี้ประกอบด้วยข้อมูล อินพุท-เอาต์พุทเวคเตอร์ ที่ใช้ในการฝึกหัดโครงข่ายจะขอยกตัวอย่างของการสร้างแฟ้มข้อมูลฝึกหัดโดยให้ชื่อว่า XOR.TRN ซึ่งมีข้อมูลที่ใช้ฝึกหัดเป็นตรรก "XOR" อันเป็นปัญหาไม่เป็นเชิงเส้นมาตรฐานของวงการโครงข่ายประสาทเทียม ข้อมูลภายในแฟ้ม XOR.TRN มีดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



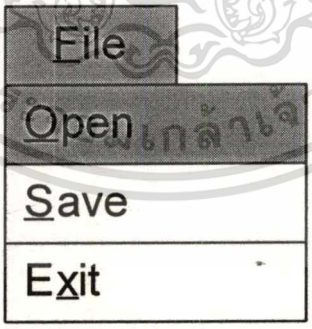
จากข้อมูลภายในแฟ้ม XOR.TRN ได้กำหนดให้คอลัมท์ที่ 1 และคอลัมท์ที่ 2 เป็นข้อมูลอินพุท ส่วนคอลัมท์ที่ 3 คือข้อมูลเอาต์พุทที่เป็นค่าเป้าหมาย จำนวนอินพุท-เอาต์พุทที่ใช้ในการฝึกหัดนี้มีจำนวน 4 เวคเตอร์ หรือเรียกอีกอย่างหนึ่งว่า 4 แพทเทิร์น (pattern) ข้อกำหนดจำนวนเวคเตอร์สูงสุดของโปรแกรมจำลองการทำงานโครงข่ายประสาทเทียม ได้กำหนดให้มีจำนวนเวคเตอร์ได้สูงสุดคือ 100 เวคเตอร์ ข้อสังเกตุอีกอย่างหนึ่งคือ เนื่องจากการจำลองโครงข่ายประสาทเทียมแบบแพร่ย้อนกลับนี้ได้นำฟังก์ชั่นกระตุ้นไบนารี่ซิกมอยด์มาใช้ดังนั้นค่าที่เอาต์พุทจะอยู่ในช่วงระหว่าง 0.0 ถึง 1.0 การกำหนดค่าเป้าหมายก็ต้องกำหนดอยู่ในช่วงนี้เช่นกัน หลังจากที่ได้สร้างแฟ้มข้อมูล XOR.NET และ XOR.TRN เรียบร้อยแล้ว ก็พร้อมที่จะใช้งานโปรแกรมจำลองการทำงานโครงข่ายประสาทเทียม เนื่องจากโปรแกรมทำงานภายใต้ระบบปฏิบัติการวินโดว์ ดังนั้นการเรียกใช้โปรแกรมสามารถกระทำได้โดยการเรียกโปรแกรม BPNSIM.EXE ที่ไฟล์แมนเนจเจอร์ (file manager) ในระบบปฏิบัติการวินโดว์ เมื่อโปรแกรมเริ่มทำงานจะปรากฏภาพขึ้นที่จอภาพดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 39 การแสดงผลเริ่มต้นของโปรแกรม

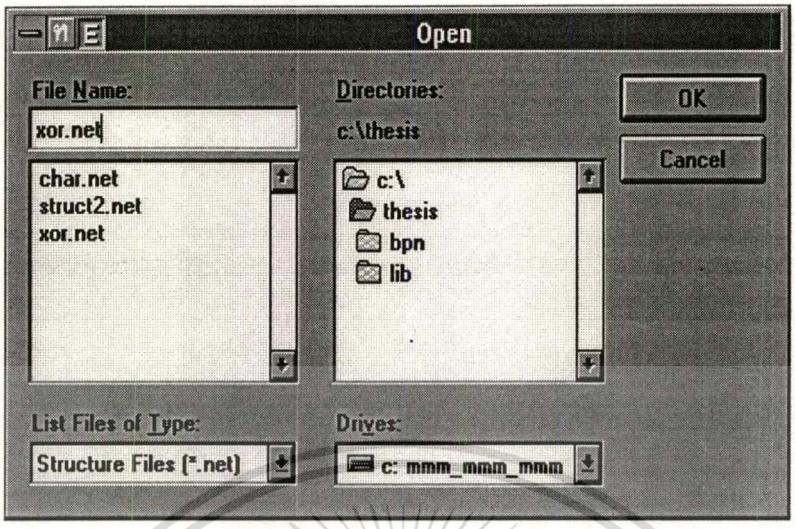
จากนั้นให้เลือกการทำงานที่หัวข้อ File ที่เมนูบาร์ จะปรากฏเป็นพูลดาวน์เมนูให้เลือก Open เพื่อที่จะเปิดแฟ้มข้อมูลโครงสร้าง



ภาพที่ 40 แสดงการเปิดแฟ้มข้อมูลโครงสร้าง

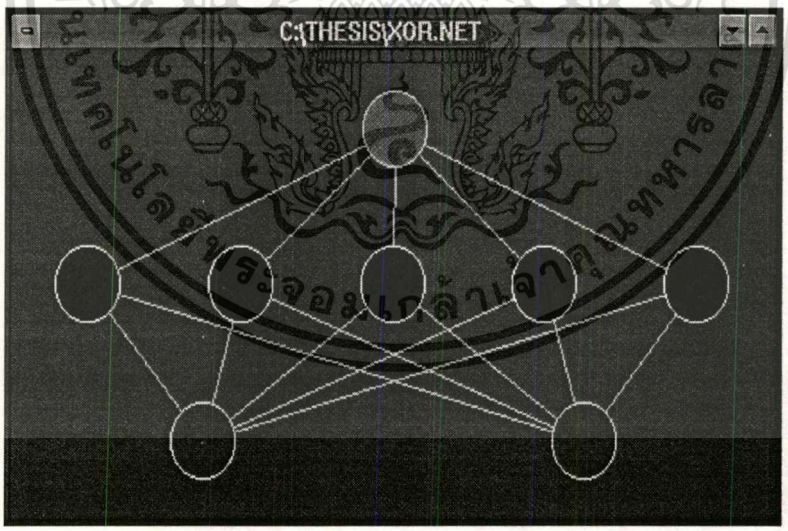
เมื่อเลือก Open แล้วจะปรากฏไดอะล็อกบ๊อกสำหรับเปิดแฟ้มข้อมูลให้คีย์ชื่อแฟ้มข้อมูล XOR.NET ที่ได้สร้างไว้แล้วจากนั้นกดปุ่ม OK.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 41 แสดงไดอะล็อกบ๊อกเปิดแฟ้มข้อมูล

โปรแกรมจำลองการทำงานจะสร้างหน้าต่างลูก โดยภายในหน้าต่างแสดงให้เห็นถึงลักษณะโครงข่ายที่ได้ทำการกำหนดไว้ในแฟ้มข้อมูลโครงสร้าง XOR.NET

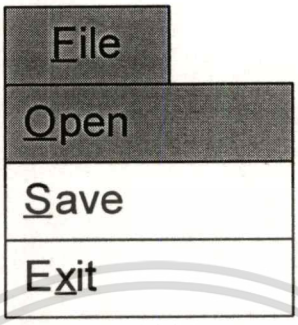


ภาพที่ 42 หน้าต่างลูกแสดงโครงข่าย

ภาพที่ 42 แสดงให้เห็นโครงข่ายที่โปรแกรมได้สร้างขึ้นตามรูปแบบที่กำหนดไว้ในแฟ้มกำหนดข้อมูลโครงสร้าง XOR.NET ซึ่งจะมีชั้นอินพุท 2 อินพุท ชั้นซ่อน 1 ชั้นมี 5 นิวรอน ชั้นเอาต์พุท 1 เอาต์พุท

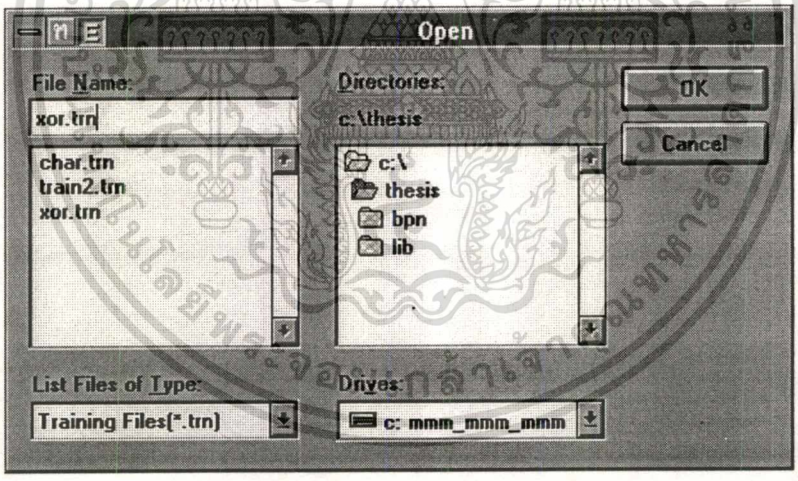
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นตอนต่อไปให้ทำการเปิดแฟ้มข้อมูลอีกครั้งหนึ่งโดยครั้งนี้จะเปิดแฟ้มข้อมูลฝึกหัด XOR.TRN เลือกการทำงานที่หัวข้อ File แล้วเลือก Open



ภาพที่ 43 แสดงการเปิดแฟ้มข้อมูลฝึกหัด

จากนั้นคีย์ชื่อแฟ้มข้อมูล XOR.TRN กดปุ่ม OK.



ภาพที่ 44 แสดงไดอะล็อกบ๊อกเปิดแฟ้มข้อมูล

โปรแกรมจะสร้างหน้าต่างลูกขึ้นอีกหนึ่งหน้าต่างโดยหน้าต่างที่สร้างขึ้นใหม่นี้จะแสดงข้อมูลรายละเอียดในระหว่างการฝึกหัดโครงข่าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 45 หน้าต่างลูกแสดงรายละเอียดการทำงาน

ภายในจอภาพจะแสดงรายละเอียดการทำงานดังนี้

Input Pattern: บอกลำดับของแพทเทิร์นที่ทำการฝึกหัดอยู่

Total cycles: บอกจำนวนรอบที่กำลังฝึกหัดอยู่

Average error per pattern: บอกค่าผิดพลาดเฉลี่ยต่อ 1 แพทเทิร์นขณะกำลังฝึกหัด

Average error per cycle: บอกค่าผิดพลาดเฉลี่ยต่อ 1 รอบการทำงาน

Error last cycle: บอกค่าผิดพลาดในรอบการทำงานครั้งสุดท้าย

Error last cycle per pattern: บอกค่าผิดพลาดเฉลี่ยต่อ 1 แพทเทิร์นในรอบการทำงานครั้งสุดท้าย

Total cycles: บอกจำนวนรอบที่ใช้ฝึกหัดทั้งหมด

Total patterns: บอกจำนวนแพทเทิร์นที่ใช้ฝึกหัดทั้งหมด

Learning process: บอกสถานะของการทำงาน

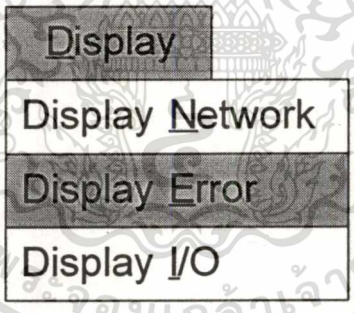
ลำดับต่อไปจะทำการฝึกหัดโครงข่ายโดยเลือกการทำงานที่หัวข้อ Run บนเมนูบาร์แล้วเลือก Training

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 46 แสดงการเลือกฝึกหัดโครงข่าย

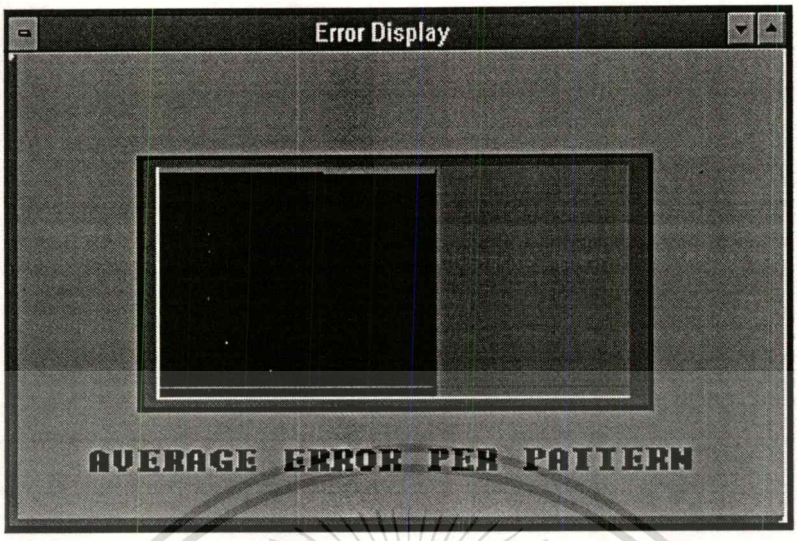
โปรแกรมจะเริ่มทำงานในการเรียนรู้ตรรก "XOR" ที่กำหนดไว้ในแฟ้มข้อมูลฝึกหัด ในระหว่างที่ทำการฝึกหัดโครงข่ายนี้สามารถที่จะเปิดหน้าต่างแสดงกราฟค่าผิดพลาดได้ โดยการเลือกการทำงานที่หัวข้อ Display แล้วเลือก Display Error โปรแกรมจะสร้างหน้าต่างลูกเพื่อแสดงกราฟค่าผิดพลาดอีกหนึ่งหน้าต่าง



ภาพที่ 47 แสดงการเลือกเปิดหน้าต่างแสดงกราฟค่าผิดพลาด

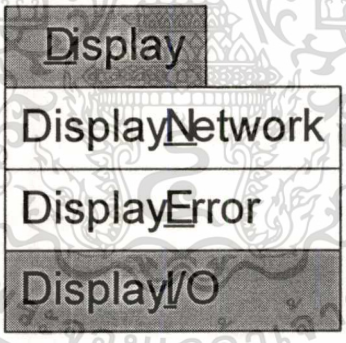
ภายในหน้าต่างแสดงกราฟระดับค่าผิดพลาดนี้จะมีเส้นเส้นสแกน(scan)ที่แสดงระดับค่าผิดพลาดและเส้นระดับของค่าผิดพลาดต่ำสุดที่ได้กำหนดไว้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 48 หน้าต่างลูกแสดงระดับค่าผิดพลาด

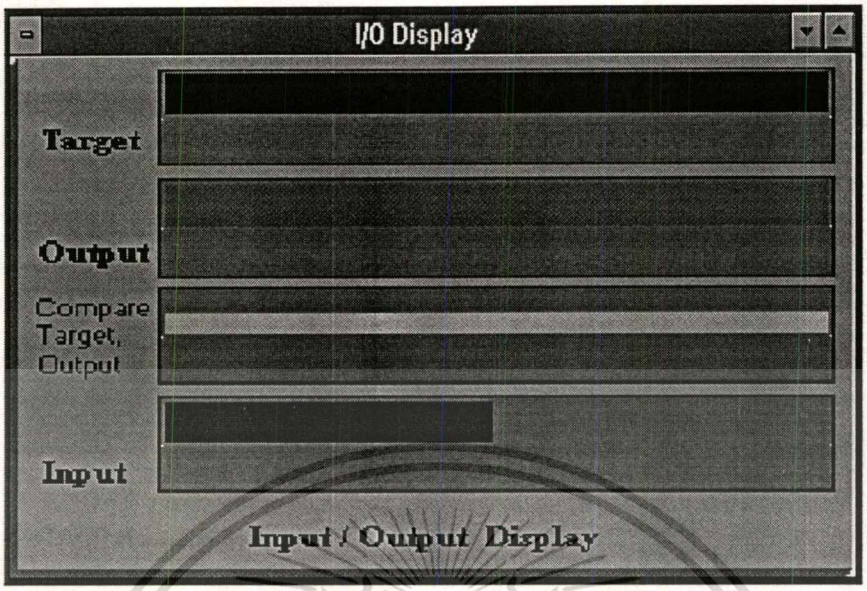
หน้าต่างแสดงผลการทำงานนั้นสามารถเปิดได้อีกหนึ่งหน้าต่าง โดยการเลือก Display บนเมนูบาร์แล้วเลือก Display I/O



ภาพที่ 49 การเลือกเปิดหน้าต่างแสดง I/O

จะเห็นหน้าต่างลูกแสดงค่าระดับอินพุท ระดับค่าเอาต์พุท ระดับค่าเป้าหมาย และระดับค่าผิดพลาดของเอาต์พุทเปรียบเทียบกับค่าเป้าหมาย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 50 หน้าต่างลูกแสดง I/O

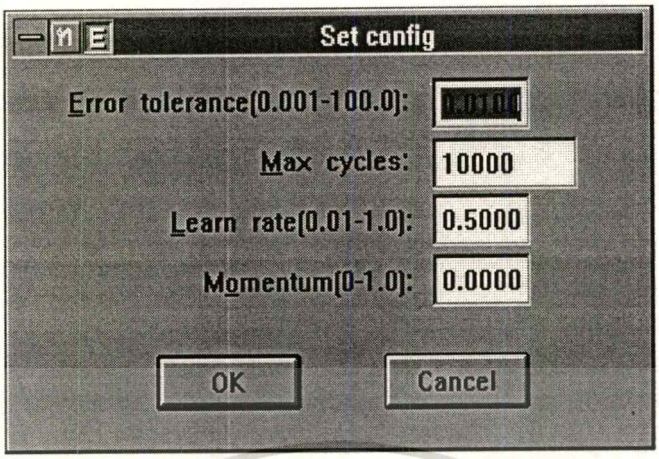
การฝึกหัดโครงข่ายประสาทเทียมนั้นสามารถจะตั้งค่าพารามิเตอร์ใหม่ให้กับโปรแกรมได้ โดยเลือก Config บนเมนูบาร์แล้วเลือก Set Config



ภาพที่ 51 การเลือกกำหนดค่าพารามิเตอร์ใหม่

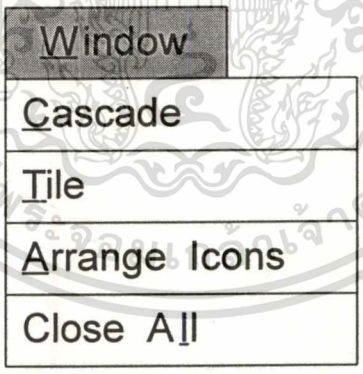
จะปรากฏไดอะล็อกบ๊อกสำหรับใส่ข้อมูลพารามิเตอร์ เพื่อให้สามารถกำหนดค่าได้ใหม่ตามต้องการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 52 แสดงไดอะล็อกบ๊อกกำหนดพารามิเตอร์

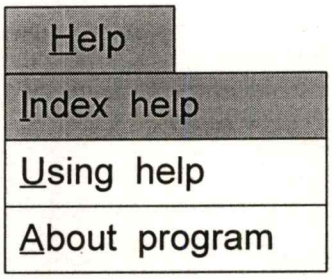
เมนูบาร์ Window จะเป็นการเลือกรูปแบบต่าง ๆ คือ เลือกการแสดงผลแบบคาสเคด (Cascade) เลือกการแสดงผลแบบไทล์ (Tile) หรือถ้าได้ มินิไมซ์ (Minimize)หน้าต่างลูก ทาสส์ (Task) หรืองานของหน้าต่างลูกจะกลายเป็นไอคอน (Icon) ซึ่งสามารถจัดไอคอนนี้ได้โดยการเลือก Arrange Icons หรือเราต้องการปิดหน้าต่างลูกทั้งหมดก็สามารถทำได้โดยการเลือก Close All



ภาพที่ 53 แสดงเมนูเลือกรูปแบบการแสดงผล

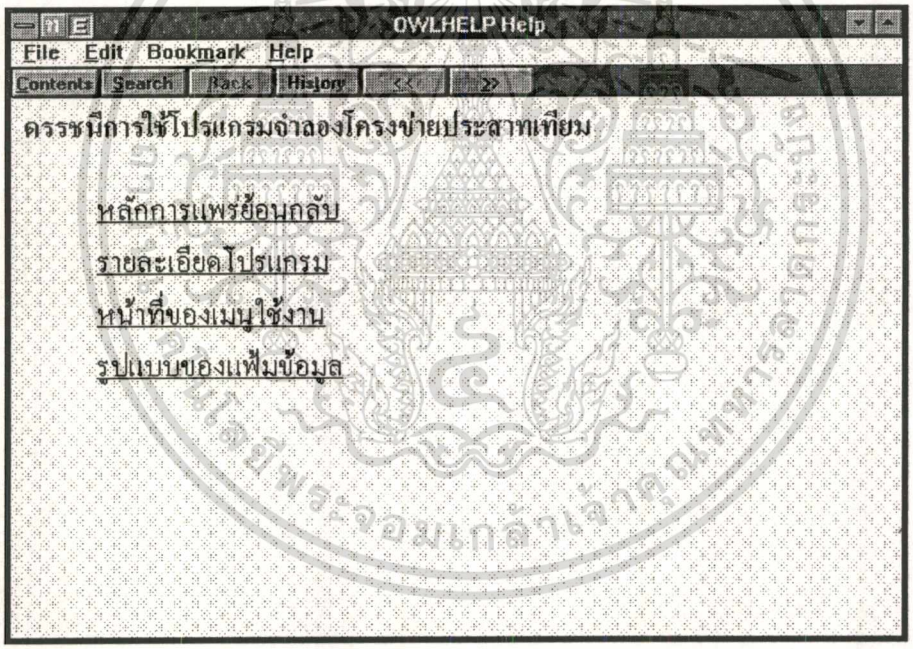
เมนูบาร์ Help จะเป็นการดูรายละเอียดของการใช้งานโปรแกรมหรือเวอร์ชั่นของโปรแกรม โดยแบ่งเป็นเมนูย่อยคือ Index help เป็นดรรชนีรายละเอียดในส่วนต่างๆของการใช้โปรแกรม Using helpเป็นส่วนของการใช้โปรแกรมช่วยเหลือและAbout programจะแสดงเวอร์ชั่นของโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 54 แสดงการเลือกดูดรรชนีช่วยเหลือ

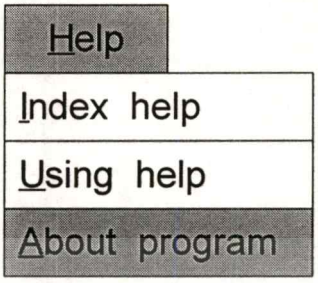
โปรแกรมจะสร้างส่วนช่วยเหลือเพื่อแสดงรายละเอียดส่วนต่างๆของการใช้โปรแกรมโดยจะแสดงผลเป็นภาษาไทย



ภาพที่ 55 แสดงดรรชนีช่วยเหลือ

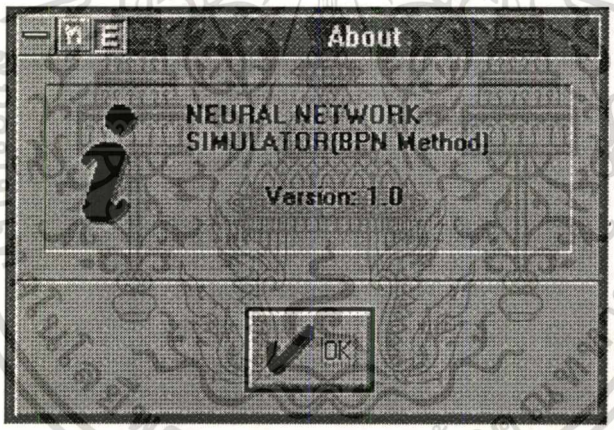
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เลือก About program จะแสดงเวอร์ชั่นของโปรแกรมจำลองโครงข่ายประสาทเทียม



ภาพที่ 56 แสดงการเลือกดูเวอร์ชั่นของโปรแกรม

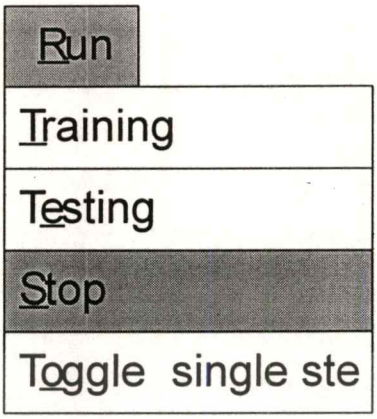
แสดงเวอร์ชั่นของโปรแกรม



ภาพที่ 57 แสดงอินฟอร์เมชั่นไดอะล็อกบ๊อก

ในระหว่างการฝึกหัดโครงข่ายนั้นสามารถที่จะหยุดการฝึกหัดไว้ชั่วคราวได้ โดยเลือก Run บนเมนูบาร์แล้วเลือก Stop

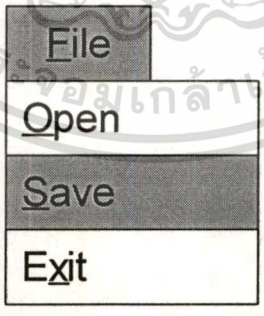
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 58 เมนูเลือกการหยุดการฝึกหัดชั่วคราว

ถ้าต้องการทำฝึกหัดโครงข่ายต่อไปก็ให้เลือก Run บนเมนูบาร์แล้วเลือก Training และยังสามารถฝึกหัดโครงข่ายในลักษณะซิงเกิลสเตปได้โดยการเลือก Run บนเมนูบาร์แล้วเลือก Toggle single step จากนั้นเลือก Training ในการฝึกหัดโครงข่ายเป็นซิงเกิลสเตป ถ้าต้องการฝึกหัดแบบปรกติอีกครั้งก็เลือก Toggle single step อีกครั้งแล้วจึงเลือก Training ซึ่งทุก ๆ ขั้นตอนจะปรากฏเมจเสจที่หน้าต่างลูกแสดงรายละเอียดเพื่อป้องกันการสับสน

เมื่อการฝึกหัดโครงข่ายสิ้นสุดลง สามารถจัดเก็บค่าน้ำหนักได้โดยการเลือก File บนเมนูบาร์ แล้วเลือก Save

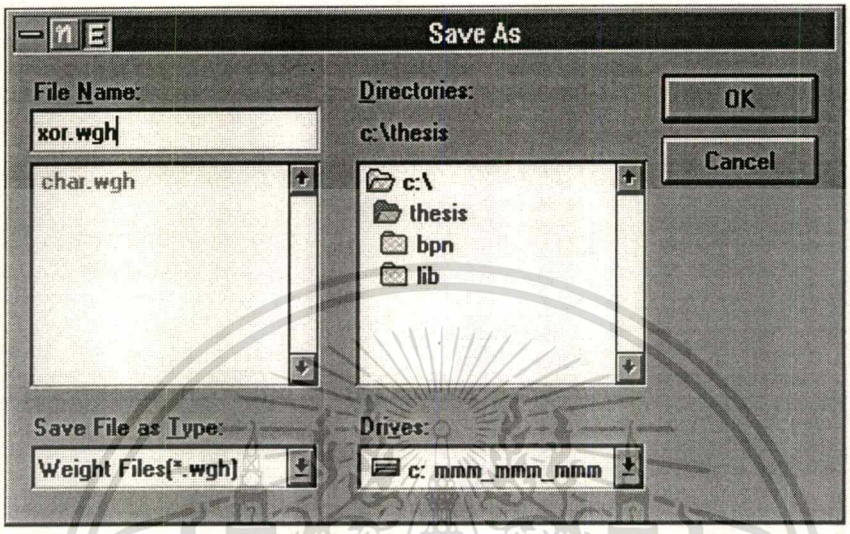


ภาพที่ 59 แสดงการเลือกจัดเก็บค่าน้ำหนัก

จากนั้นโปรแกรมจะแสดงไดอะล็อกบ็อกสำหรับการจัดเก็บ ซึ่งสามารถจัดเก็บค่าน้ำหนักได้โดยให้คีย์ชื่อแฟ้มข้อมูลที่ต้องการจัดเก็บ ในตัวอย่างนี้ให้คีย์ชื่อแฟ้มข้อมูลที่ต้องการชื่อ XOR จะใส่นามสกุล .WGH หรือไม่ต้องใส่ก็ได้เพราะโปรแกรมจะแทรกนามสกุล .WGH ให้โดย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อัตโนมัติ ดังนั้นจึงหมายความว่าแฟ้มข้อมูลที่มีนามสกุล .WGH คือแฟ้มข้อมูลน้ำหนัก (weight file)



ภาพที่ 60 แสดงไดอะล็อกบ๊อกจัดเก็บค่าน้ำหนัก

เมื่อต้องการออกจากโปรแกรมจำลองโครงข่ายประสาทเทียมให้เลือก File บนเมนูบาร์แล้วเลือก Exit



ภาพที่ 61 แสดงการเลือกออกจากโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทดสอบหาผลลัพธ์ทางเอาต์พุทโครงข่ายประสาทเทียม

การทดสอบหาผลลัพธ์ทางเอาต์พุทโครงข่ายประสาทเทียมนั้นก็คือการนำโครงข่ายที่ผ่านการฝึกหัดแล้วมาใช้กับกลุ่มข้อมูลทดสอบเพื่อดูผลลัพธ์ทางเอาต์พุท ก่อนที่จะใช้งานโครงข่ายประสาทเทียมเพื่อการทดสอบนั้น จะต้องมีการเตรียมการล่วงหน้าก่อนสิ่งที่ต้องจัดเตรียมคือ

1. เตรียมแฟ้มข้อมูลโครงสร้างโครงข่ายประสาทเทียม (structure file)
2. เตรียมแฟ้มข้อมูลน้ำหนัก (weight file)
3. สร้างแฟ้มข้อมูลทดสอบโครงข่ายประสาทเทียม (testing file)

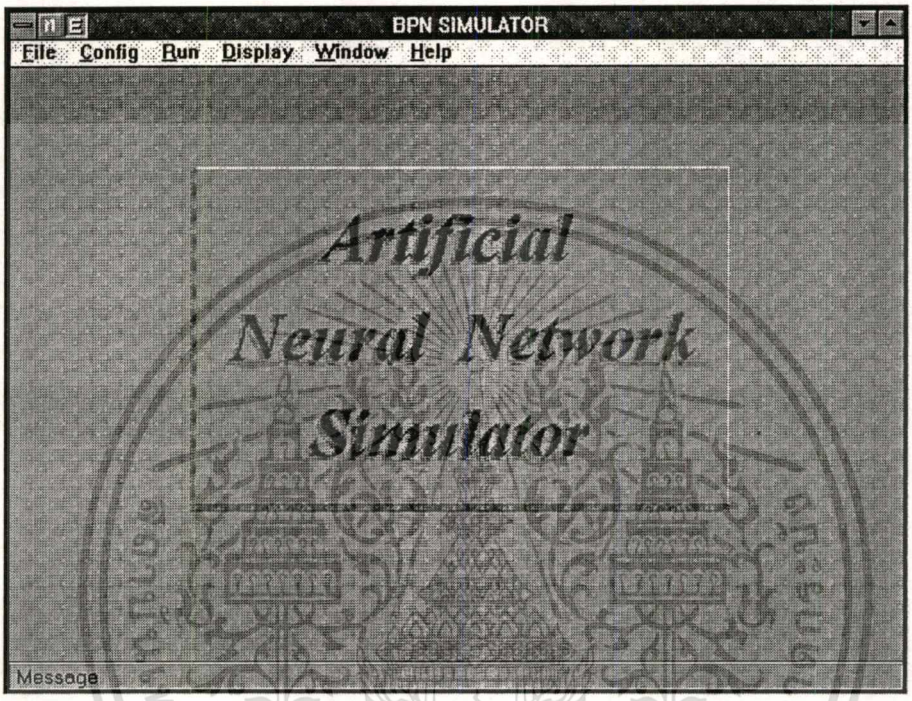
การเตรียมแฟ้มข้อมูลโครงสร้างนั้นสามารถที่จะนำแฟ้มข้อมูลโครงสร้างเดิมที่ได้สร้างไว้แล้วในขั้นตอนการฝึกหัดโครงข่ายประสาทเทียม จากหัวข้อที่ผ่านมาได้เคยสร้างแฟ้มข้อมูลโครงสร้างชื่อว่า XOR.NET ซึ่งจะใช้แฟ้มนี้ในการทดสอบโครงข่าย จากการฝึกหัดในหัวข้อที่แล้วได้ทำการเก็บค่าน้ำหนักไว้ที่แฟ้มข้อมูลน้ำหนักโดยใช้ชื่อว่า XOR.WGH ซึ่งก็จะนำแฟ้มข้อมูลน้ำหนักนี้มาใช้ร่วมด้วยเช่นกัน ส่วนแฟ้มข้อมูลสุดท้ายคือ แฟ้มข้อมูลทดสอบนั้นจะต้องสร้างขึ้นใหม่โดยให้แฟ้มมีนามสกุล .TES ภายในแฟ้มนี้จะมีข้อมูลที่กำหนดเฉพาะส่วนที่เป็นอินพุทไม่ต้องมีค่าเอาต์พุทหรือค่าเป้าหมายเพราะจะให้โครงข่ายทำการหาค่าเอาต์พุทให้ จะขอยกตัวอย่างแฟ้มข้อมูลทดสอบข้อมูลทดสอบชื่อว่า XOR.TES โดยกำหนดให้มีข้อมูลภายในแฟ้มดังนี้

```
0.1     0.1
0.1     0.9
0.9     0.1
0.9     0.9
↑_____↑__ อินพุทเวคเตอร์
```

ข้อมูลภายในแฟ้มได้ถูกกำหนดให้ทั้งคอลัมท์ที่ 1 และคอลัมท์ที่ 2 เป็นข้อมูลอินพุท ดังนั้นจึงมี 2 อินพุท และจำนวนที่ใช้ในการทดสอบคือ 4 เวคเตอร์ หรือ 4 แพทเทิร์น ข้อกำหนดจำนวนเวคเตอร์สูงสุดของโปรแกรมจำลองการทำงานโครงข่ายประสาทเทียมได้กำหนดให้มีจำนวนเวคเตอร์ได้สูงสุดคือ 100 เวคเตอร์

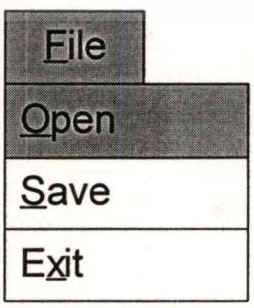
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อเตรียมแฟ้มต่างๆครบตามต้องการแล้วก็พร้อมที่จะใช้งานโปรแกรมจำลองการทำงานโครงข่ายประสาทเทียมโดยเรียกโปรแกรม BPNSIM.EXE ที่ไฟล์แมนเนจเจอร์ในระบบปฏิบัติการวินโดว์ เมื่อโปรแกรมเริ่มทำงานจะปรากฏภาพที่จอภาพดังนี้



ภาพที่ 62 การแสดงผลเริ่มต้นของโปรแกรม

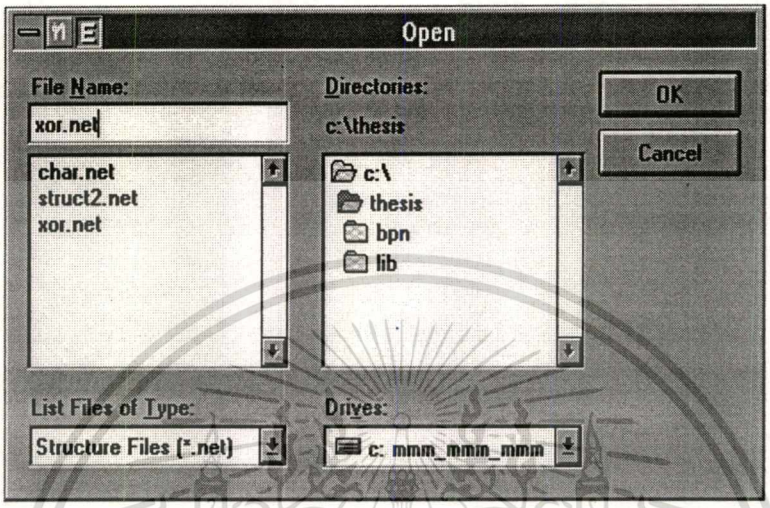
จากนั้นให้เลือกการทำงานที่หัวข้อ File ที่เมนูบาร์จะปรากฏพูลดาวน์เมนูให้เลือก Open เพื่อที่จะเปิดแฟ้มข้อมูลโครงสร้าง



ภาพที่ 63 แสดงการเปิดแฟ้มข้อมูลโครงสร้าง

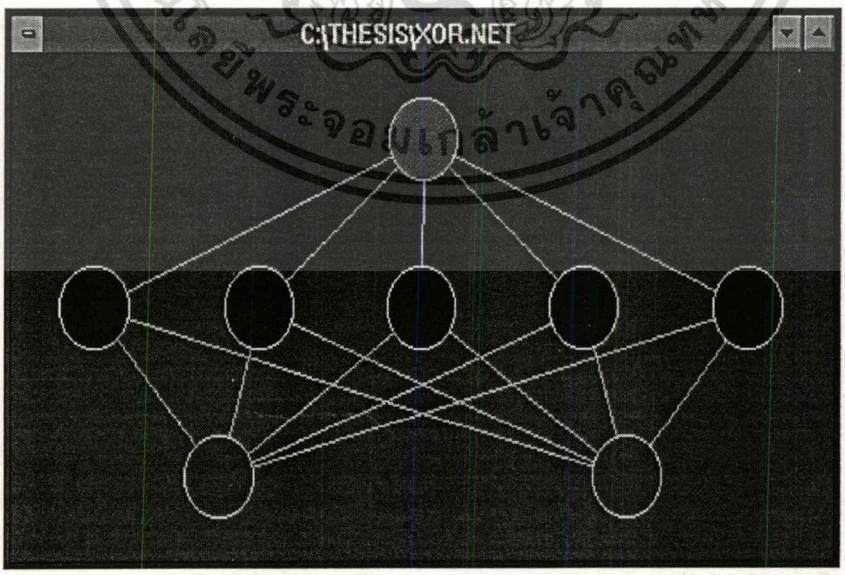
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อเลือก Open แล้วจะปรากฏไดอะล็อกบ๊อก สำหรับเปิดแฟ้มข้อมูลให้คีย์ชื่อแฟ้มข้อมูล XOR.NET จากนั้นกดปุ่ม OK.



ภาพที่ 64 แสดงไดอะล็อกบ๊อกเปิดแฟ้มข้อมูล

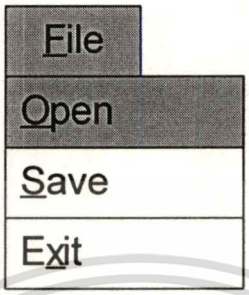
โปรแกรมจำลองการทำงานจะสร้างหน้าต่างลูกโดยภายในหน้าต่างแสดงให้เห็นถึงลักษณะโครงสร้างโครงข่ายที่ได้ทำการกำหนดไว้ในแฟ้มข้อมูลโครงสร้าง XOR.NET



ภาพที่ 65 หน้าต่างลูกแสดงโครงข่าย

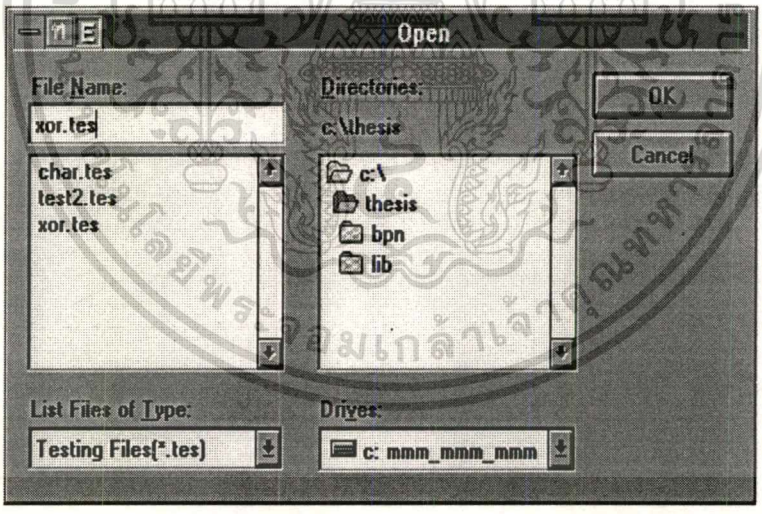
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นตอนต่อไปให้ทำการเปิดแฟ้มข้อมูลทดสอบเพื่อที่จะหาเอาต์พุทของโครงข่ายโดยการเลือกการทำงานที่หัวข้อ File ที่เมนูบาร์แล้วเลือก Open



ภาพที่ 66 แสดงการเปิดแฟ้มข้อมูลทดสอบ

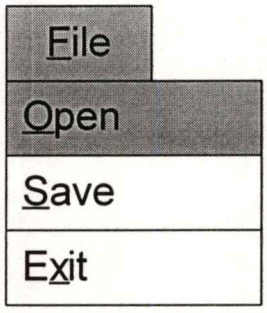
จากนั้นคีย์ชื่อแฟ้มข้อมูล XOR.TES ซึ่งเป็นแฟ้มข้อมูลทดสอบที่ได้เตรียมไว้ แล้วกดปุ่ม OK.



ภาพที่ 67 แสดงไดอะล็อกบ๊อกเปิดแฟ้มข้อมูล

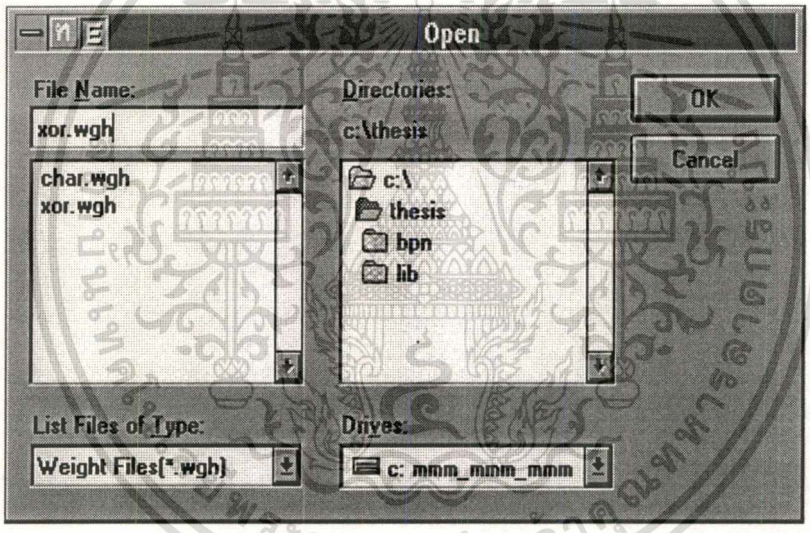
และต้องเปิดแฟ้มข้อมูลสุดท้ายคือแฟ้มข้อมูลน้ำหนักแฟ้มนี้ คือ XOR.WGH เพื่อที่จะโหลดค่าน้ำหนักที่เก็บอยู่ในแฟ้มให้แก่โครงข่ายประสาทเทียม เลือกการทำงานที่หัวข้อ File แล้วเลือก Open

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 68 แสดงการเปิดแฟ้มข้อมูลน้ำหนัก

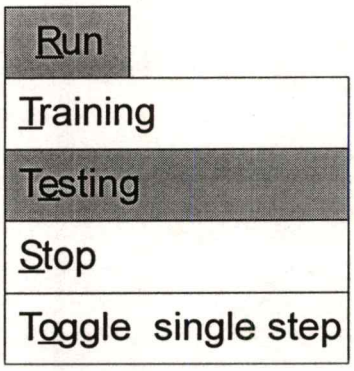
จากนั้นคีย์ชื่อแฟ้มข้อมูลน้ำหนัก XOR.WGH แล้วกดปุ่ม OK.



ภาพที่ 69 แสดงไดอะลือกบ๊อกเปิดแฟ้มข้อมูลน้ำหนัก

เมื่อต้องการที่จะทดสอบหาค่าเอาต์พุทของโครงข่ายสามารถทำได้โดยเลือกการทำงานที่หัวข้อ Run บนเมนูบาร์แล้วเลือก Testing

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



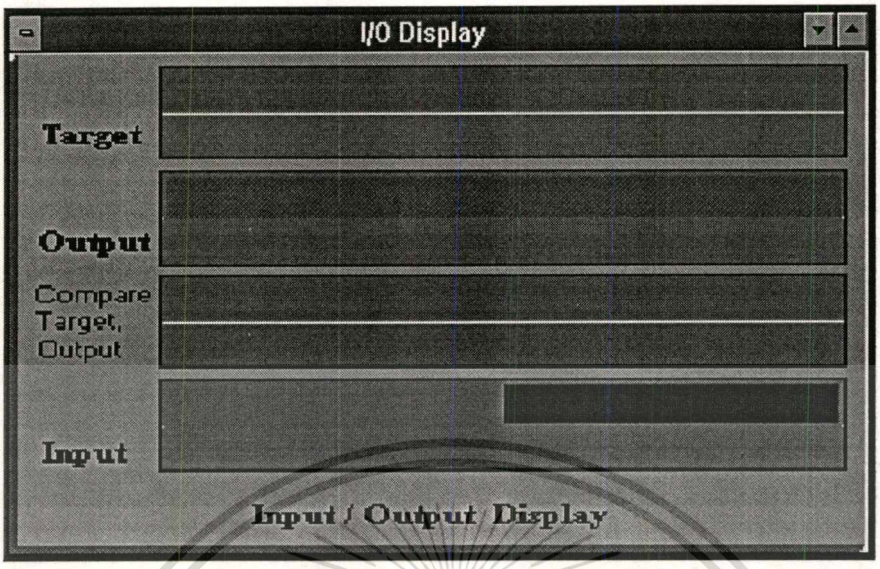
ภาพที่ 70 แสดงการเลือกทดสอบโครงข่าย

โปรแกรมจะเริ่มทำงานในการทดสอบโครงข่ายโดยจะสร้างหน้าต่างลูกขึ้นมาอีกสองหน้าต่างคือหน้าต่างแสดงลำดับข้อมูลทดสอบและหน้าต่างลูกแสดงค่าอินพุทและค่าเอาต์พุทของโครงข่ายประสาทเทียม



ภาพที่ 71 หน้าต่างลูกแสดงลำดับข้อมูลทดสอบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 72 หน้าต่างลูกแสดง I/O

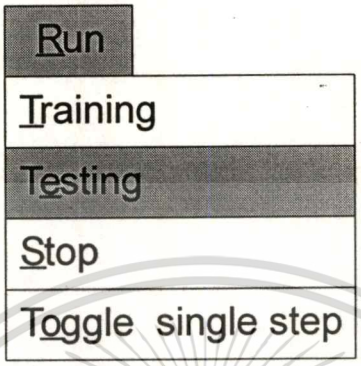
ที่หน้าต่างลูกแสดงอินพุทและเอาต์พุทนั้นจะใช้เฉพาะช่องแสดงผลเอาต์พุท (Output) และช่องแสดงผลอินพุท (Input) เท่านั้น เพื่อแสดงค่าอินพุทหรือข้อมูลทดสอบที่กำหนดอยู่ในแฟ้ม XOR.TES ส่วนที่ช่องเอาต์พุทจะแสดงค่าเอาต์พุทของโครงข่ายประสาทเทียม การทำงานของการทดสอบโครงข่ายนั้นสามารถที่จะหยุดการทำงานชั่วคราวได้โดยเลือกการทำงานที่ Run บนเมนูบาร์และเลือก Stop หรือจะเลือกการทำงานแบบซิงเกิลสเตปก็ได้โดยเลือกการทำงานที่ Run บนเมนูบาร์และเลือก Toggle single step



ภาพที่ 73 แสดงการเลือกการทำงานแบบซิงเกิลสเตป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่ออยู่ในการทำงานแบบซิงเกิลสเตปจะสามารถที่จะให้โปรแกรมแสดงค่าอินพุทและเอาต์พุทได้ตามลำดับข้อมูลทดสอบโดยการเลือกการทำงาน Run บนเมนูบาร์และเลือก Testing

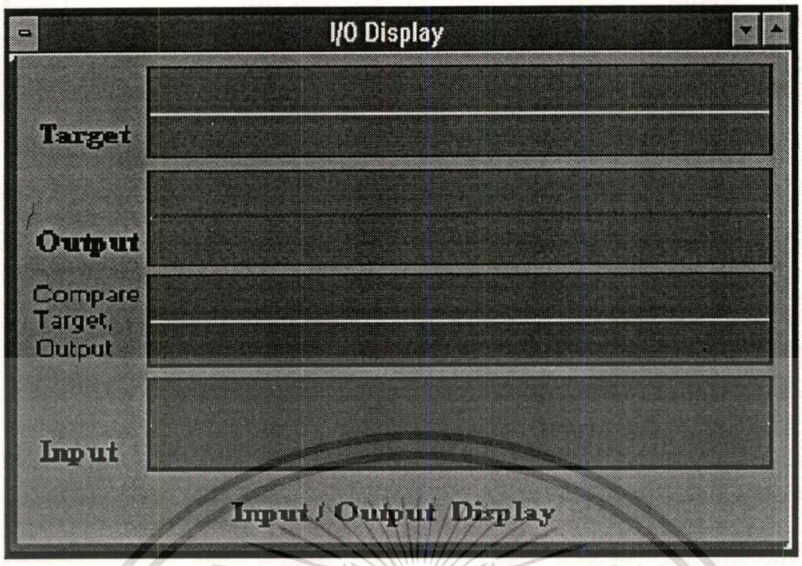


ภาพที่ 74 แสดงการเลือกลำดับข้อมูลทดสอบถัดไปเมื่ออยู่ในโหมดซิงเกิลสเตป



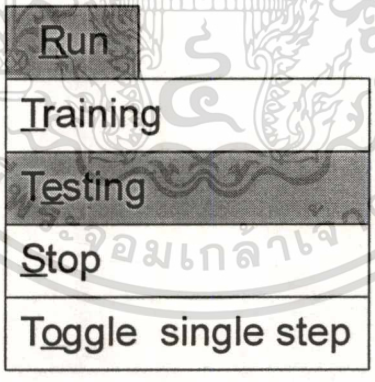
ภาพที่ 75 หน้าต่างลูกแสดงลำดับข้อมูลทดสอบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



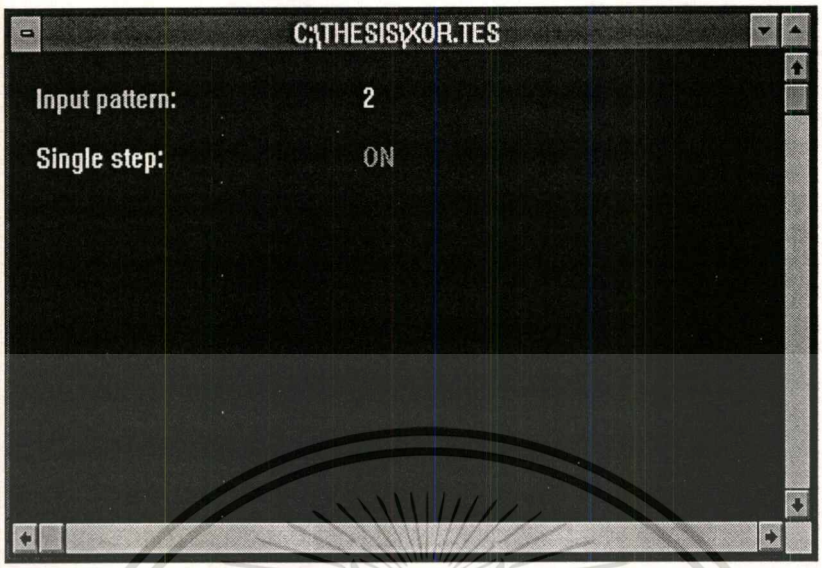
ภาพที่ 76 หน้าต่างลูกแสดง I/O ที่ลำดับข้อมูลทดสอบที่ 1

เลือกการทำงาน Runบนแมนูบาร์เลือก Testing เพื่อแสดงค่าเอาต์พุทของข้อมูลทดสอบที่ 2

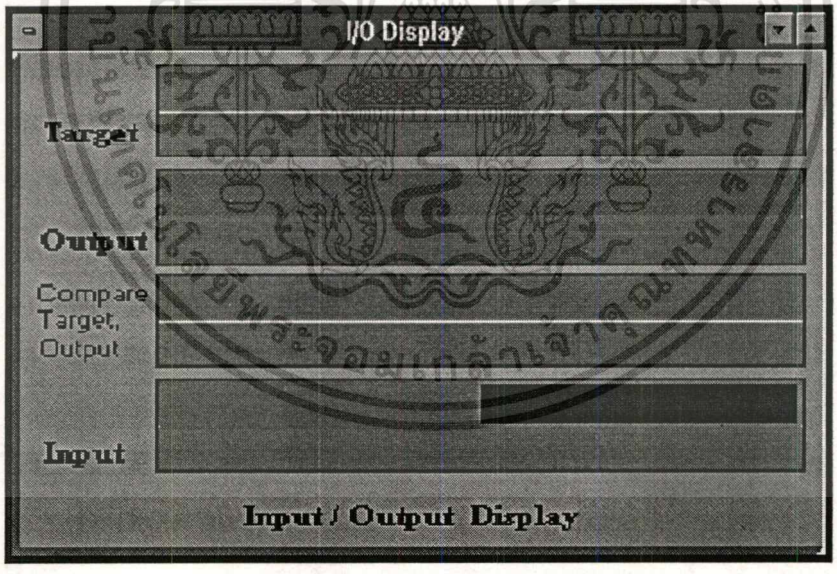


ภาพที่ 77 แสดงการเลือกลำดับข้อมูลทดสอบถัดไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 78 หน้าต่างลูกแสดงลำดับข้อมูลทดสอบ



ภาพที่ 79 หน้าต่างลูกแสดง I/O ที่ลำดับข้อมูลทดสอบที่ 2

การเลือกโหมดซิงเกิลสเตปทำให้สามารถตรวจสอบข้อมูลได้ที่ละลำดับข้อมูลทดสอบ ซึ่งผลลัพธ์จากการทดสอบนี้จะถูกเก็บเข้าสู่แฟ้มข้อมูลเอาต์พุทชื่อว่า OUTPUT.DAT สามารถเรียกดูโดยใช้เท็กต์อิดิเตอร์ใด ๆ ก็ได้ ในกรณีตัวอย่างจะได้ข้อมูลในแฟ้ม OUTPUT.DAT ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

for input vector :

0.10000 0.100000

output vector :

0.126892

---

for input vector :

0.100000 0.900000

output vector :

0.886272

---

for input vector :

0.900000 0.100000

output vector :

0.884644

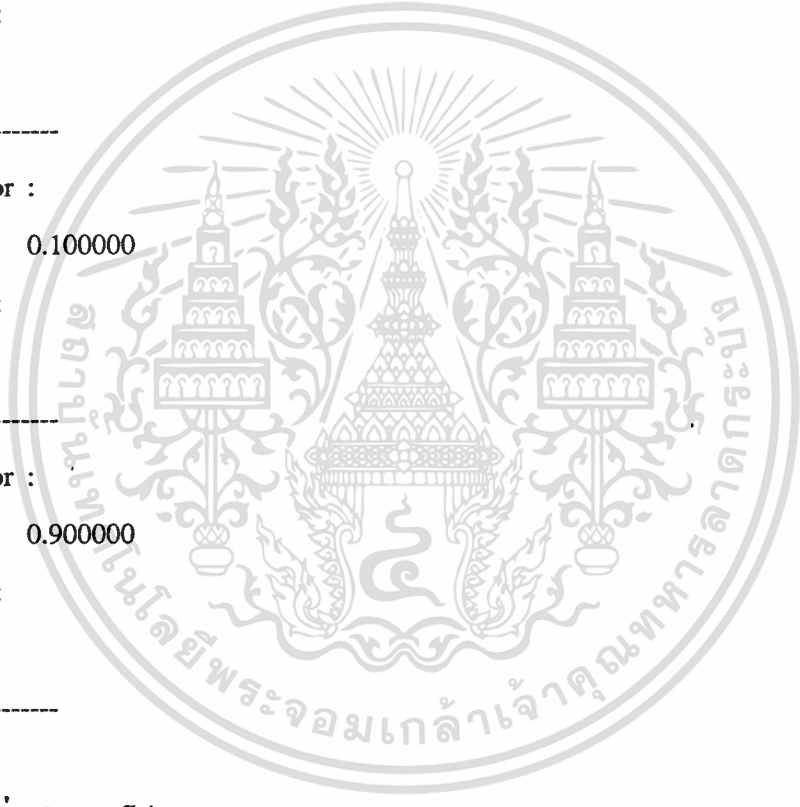
---

for input vector :

0.900000 0.900000

output vector :

0.117793



ส่วนเพิ่มเติมของโปรแกรม

โปรแกรมจำลองโครงข่ายประสาทเทียมโดยหลักการแพร่ย้อนกลับนี้ ยังมีส่วนเพิ่มเติมนอกเหนือจากที่ได้กล่าวมาแล้ว คือยังได้จัดเตรียมส่วนสนับสนุนในการเขียนโปรแกรมเพื่อใช้งานโดยอยู่ในรูปแบบของโปรแกรมไลบรารี่ (Library) ที่สามารถเรียกใช้ฟังก์ชั่นภายในที่กำหนดขึ้นซึ่งจะใช้ลิงค์(link)ร่วมกับการเขียนโปรแกรมด้วยภาษาซีพลัสพลัสของบอร์แลนด์ซีเวอร์ชั่น 4 โดยจะกล่าวรายละเอียดในบทการทดลอง ซึ่งจะแสดงการทดลองเกี่ยวกับการนำโครงข่ายประสาทเทียมไปใช้ในการรู้จำรูปแบบอักขระ โปรแกรมไลบรารี่นี้มีประโยชน์อย่างมากในการนำโครงข่ายประสาทเทียมไปใช้งานได้จริงในลักษณะของการเขียนโปรแกรมขึ้นใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทสรุป

ในบทนี้ได้อธิบายการใช้งานโปรแกรมจำลองอะดาไลน์และโปรแกรมจำลองโครงข่ายประสาทเทียมโดยหลักการแพร่ย้อนกลับอย่างเป็นขั้นตอน ในหัวข้อนี้จะสรุปข้อจำกัดการใช้งานทั้งสองโปรแกรมดังนี้

ข้อจำกัดของโปรแกรมจำลองอะดาไลน์

1. สามารถกำหนดจำนวนอินพุทได้สูงสุด 19 อินพุท
2. สามารถกำหนดจำนวนเวคเตอร์ที่ใช้ฝึกหัด หรือใช้ทดสอบได้ สูงสุด100 เวคเตอร์
3. สามารถกำหนดอัตราการเรียนรู้ได้ในช่วงระหว่าง 0 ถึง 1
4. สามารถกำหนดค่าเป้าหมายได้ 2 ค่าคือ 1 หรือ -1
5. สามารถกำหนดค่าของอินพุทเวคเตอร์เป็นค่าจำนวนเต็มเท่านั้นโดยมีค่าอยู่ในช่วง -2,147,483,647 ถึง 2,147,483,647

ข้อจำกัดของโปรแกรมจำลองโครงข่ายประสาทเทียมโดยหลักการแพร่ย้อนกลับ

1. สามารถกำหนดจำนวนเวคเตอร์ที่ใช้ฝึกหัด หรือใช้ทดสอบได้ สูงสุด100 เวคเตอร์
2. สามารถกำหนดจำนวนชั้นของโครงข่ายจำนวนต่ำสุดคือ 3 ชั้น สูงสุด 5 ชั้น
3. สามารถกำหนดอัตราการเรียนรู้ได้ในช่วงระหว่าง 0 ถึง 1
4. สามารถกำหนดค่าผิดพลาดต่ำสุดได้ในช่วง 0.001 ถึง 100
5. สามารถกำหนดเทอมโมเมนตัมได้โดยมีค่าตั้งแต่ 0 ถึง 1
6. สามารถกำหนดจำนวนรอบในการฝึกหัดตั้งแต่ 1 ถึง 1,000,000
7. สามารถกำหนดค่าเป้าหมายได้ในช่วงระหว่าง 0 ถึง 1
8. ควรกำหนดค่าอินพุทในช่วงระหว่าง 0 ถึง 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

การทดลอง

ในบทนี้จะกล่าวถึงการนำโครงข่ายประสาทเทียมมาประยุกต์ใช้กับการรู้จำรูปแบบอักขระภาษาอังกฤษจำนวน 26 อักขระคืออักขระ A ถึงอักขระ Z และ การรู้จำภาพทางเรขาคณิต ซึ่งจะกล่าวโดยลำดับต่อไป

<u>การประยุกต์ใช้กับการรู้จำรูปแบบอักขระ</u>

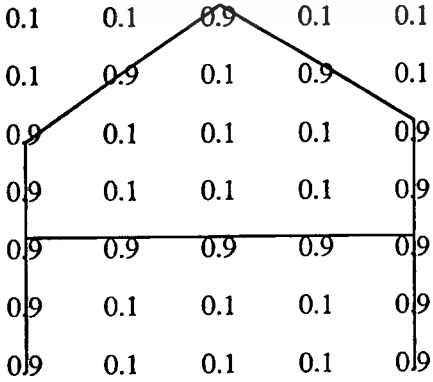
ได้จัดแบ่งลักษณะการทดลองเป็น 2 ลักษณะคือ

1. การฝึกหัดโครงข่ายประสาทเทียมโดยอาศัยโปรแกรมจำลองโครงข่ายประสาทเทียมโดยหลักการแพร่ย้อนกลับ

2.การทดสอบโครงข่ายประสาทเทียมโดยอาศัยการเขียนโปรแกรมเพื่อเรียกใช้ฟังก์ชั่นไลบรารี่ที่มีมาให้

<u>การฝึกหัดโครงข่ายประสาทเทียม</u>

ได้มีการจัดเตรียมแฟ้มข้อมูลฝึกหัดเพื่อที่จะใช้ฝึกหัดโครงข่ายแฟ้มนี้ให้ชื่อว่า CHAR.TRN ข้อมูลภายในแฟ้มข้อมูลฝึกหัดได้จัดแสดงไว้ในภาคผนวกที่ท้ายเล่ม ข้อมูลภายในแฟ้ม CHAR.TRN ได้ถูกกำหนดให้มีข้อมูลฝึกหัดจำนวน 26 เวคเตอร์ หรือ 26 แพทเทิร์น โดยแต่ละแพทเทิร์นประกอบด้วยลำดับ อินพุท-เอาต์พุท ข้อมูลฝึกหัดอินพุทหรืออินพุทเวคเตอร์ ได้มาจากค่าของพิกเซล (pixel) ของตัวอักขระซึ่งกำหนดไว้ให้มีขนาด 5x7 พิกเซล โดยขอยกตัวอย่างอินพุทเวคเตอร์แรกในแฟ้มข้อมูล CHAR.TRN คือ



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เป็นข้อมูลของอักขระ A จากขนาดที่กำหนดไว้ 5x7 พิกเซล ซึ่งทำให้เป็นข้อมูลอินพุทเวคเตอร์ขนาด 35 อินพุท ส่วนเอาต์พุทเวคเตอร์หรือค่าเป้าหมายนั้น ถูกกำหนดให้มีค่าอยู่ในช่วงระหว่าง 0.0 ถึง 1.0 เพราะว่าโปรแกรมจำลองโครงข่ายประสาทเทียมโดยหลักการแพร่กลับนี้ใช้ฟังก์ชั่นกระตุ้นไบนารี่ซิกมอยด์ แต่อย่างไรก็ตามค่าเป้าหมายนี้จะกำหนดไว้เพียงสองค่าคือ 0.1 และ 0.9 โดยจะแทนค่าเหล่านี้เป็นค่าไบนารี่ 0 และ 1 โดยให้ผ่านฟังก์ชั่นกำหนดเทรสโฮลด์ (threshold) คือ

if target = 0.1 then target = 0

if target = 0.9 then target = 1

ตัวอย่างค่าเอาต์พุทเวคเตอร์ของตัวอักขระ A คือ 0.1 0.9 0.1 0.1 0.1 0.1 0.1 0.9 เมื่อผ่านฟังก์ชั่นเทรสโฮลด์แล้วจะได้ค่าเลขไบนารี คือ 01000001 เมื่อแปลงเป็นค่าเลขฐานสิบหกแล้วจะได้ค่า 41H ซึ่งก็คือค่าของรหัสแอสกี้ (ASCII) ของอักขระ A นั่นเอง ดังนั้นที่เอาต์พุทเวคเตอร์จึงมีขนาด 8 เอาต์พุท จากที่กล่าวมาแล้วมาจำนวนข้อมูลฝึกหัดภายในแฟ้ม CHAR.TRN นั้นมีจำนวน 26 แพทเทิร์น ซึ่งก็ได้มีการจัดเรียงลำดับไว้ ตั้งแต่อักขระ A ถึงอักขระ Z

อีกแฟ้มหนึ่งที่ต้องจัดเตรียมในการฝึกหัดโครงข่ายนี้ก็คือแฟ้มข้อมูลโครงสร้างโครงข่ายประสาทเทียม โดยได้มีการสร้างแฟ้ม CHAR.NET ซึ่งมีข้อมูลกำหนดโครงสร้างดังนี้

LAYER: 1

NODE : 35

LAYER: 2

NODE : 10

LAYER: 3

NODE : 10

LAYER: 4

NODE : 8

MOMENTUM : 0.0

LEARN-TOLERANCE : 0.01

MAX-CYCLES : 10000

ข้อมูลโครงสร้างได้กำหนดให้โครงข่ายมีอินพุท 35 อินพุท ที่ชั้นซ่อนที่ 1 มี 10 นิวรอน ชั้นซ่อนที่ 2 มี 10 นิวรอน และที่ชั้นเอาต์พุท มี 8 นิวรอน กำหนดเทอมโมเมนต์ตัมเท่ากับศูนย์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กำหนดอัตราการเรียนรู้เท่ากับ 0.5 กำหนดค่าผิดพลาดต่ำสุดที่สามารถยอมรับได้ไว้ที่ 0.01 และกำหนดจำนวนรอบสูงสุดเพื่อใช้ในการเรียนรู้ไว้ที่ 10000 รอบ

หลังจากได้จัดเตรียมแฟ้มต่าง ๆ ที่จำเป็นต้องใช้เรียบร้อยแล้ว กระบวนการถัดไปคือ การใช้งานโปรแกรมจำลองโครงข่ายประสาทเทียมเพื่อการฝึกหัด โดยได้ทำการฝึกหัดโครงข่ายตามขั้นตอนที่ได้กล่าวผ่านมาแล้วในเรื่องของการใช้โปรแกรม สิ่งที่ได้จากการฝึกหัดโครงข่ายนี้ก็คือแฟ้มข้อมูลน้ำหนักซึ่งให้ตั้งชื่อแฟ้มข้อมูลน้ำหนักนี้ไว้ชื่อว่า CHAR.WGH โดยจะนำแฟ้มข้อมูลน้ำหนักนี้ไปใช้ในการเขียนโปรแกรมทดสอบโครงข่ายประสาทเทียมโดยจะกล่าวรายละเอียดในหัวข้อถัดไป

<u>การทดสอบโครงข่ายประสาทเทียม</u>

ได้มีการจัดเตรียมแฟ้มข้อมูลทดสอบเพื่อที่จะใช้ทดสอบโครงข่ายแฟ้มนี้ให้ชื่อว่า CHAR.TES ข้อมูลภายในแฟ้มข้อมูลทดสอบได้เพิ่มส่วนของสัญญาณรบกวนเข้าไปในข้อมูลด้วย ซึ่งข้อมูลทั้งหมดภายในแฟ้มนี้จัดแสดงไว้ที่ภาคผนวกท้ายเล่ม จากข้อมูลภายในแฟ้ม CHAR.TES ได้ถูกกำหนดให้มีข้อมูลทดสอบ 26 เวคเตอร์ หรือ 26 แพทเทิร์นเพื่อนำข้อมูลเหล่านี้ไปทดสอบหาเอาต์พุทของโครงข่าย ส่วนอีกสองแฟ้มที่จำเป็นสำหรับการทดสอบโครงข่ายคือ แฟ้มข้อมูลโครงสร้างCHAR.NET ที่ได้เตรียมไว้ในขั้นตอนการฝึกหัดโครงข่ายและแฟ้มข้อมูล CHAR.WGH ที่ได้ทำการจัดเก็บไว้ในขั้นตอนฝึกหัดเช่นกัน หลังจากที่ได้จัดเตรียมแฟ้มต่างๆได้ครบถ้วนแล้วก็สามารถที่จะทำการทดสอบหาเอาต์พุทของโครงข่ายได้ วิธีการทดสอบนั้นสามารถจะกระทำได้สองวิธีคือ

1. ใช้โปรแกรมจำลองโครงข่ายประสาทเทียมโดยวิธีการแพร่ย้อนกลับ ซึ่งวิธีนี้ได้กล่าวถึงการใช้งานแล้วในเรื่องของการใช้งานโปรแกรมเพื่อการทดสอบ

2. สร้างโปรแกรมทดสอบโดยใช้ภาษาซีพลัสพลัสและบอร์แลนด์ซีคอมไพเลอร์เวอร์ชั่น 4 ร่วมกับฟังก์ชั่นไลบรารี่ที่มีมาให้กับโปรแกรม

ในการทดลองนี้จะใช้วิธีที่สองคือสร้างโปรแกรมทดสอบขึ้นโดยเชื่อมโยงกับฟังก์ชั่นไลบรารี่ซึ่งมีประโยชน์มากสำหรับการนำไปประยุกต์ใช้กับงานด้านอื่นๆเพราะผู้ใช้สามารถสร้างโครงข่ายประสาทเทียม เพื่อนำค่าเอาต์พุทจากการทดสอบไปใช้งานตามความต้องการได้

<u>ฟังก์ชั่นไลบรารี่</u>

ในชุดของโปรแกรมที่ได้สร้างขึ้นนี้นอกจากจะสร้างโปรแกรมจำลองการทำงานแล้วยังได้สร้างส่วนที่เป็นฟังก์ชั่นไลบรารี่เพื่อให้ผู้ใช้สามารถสร้างโปรแกรมเรียกใช้ฟังก์ชั่นเหล่านี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เพื่อประยุกต์ใช้กับงานด้านต่างๆตามความต้องการได้โดยแฟ้มข้อมูลไลบรารี่นี้มีชื่อว่า BPNLIB.LIB และใช้ร่วมกับแฟ้มอินคลูด (include) ชื่อว่า BPNLIB.H ลำดับต่อไปจะกล่าวถึงหน้าที่ของฟังก์ชั่นต่าง ๆ ในแฟ้ม BPNLIB.LIB ที่ได้จัดเตรียมไว้ให้ดังนี้

class BackpNet

หน้าที่ ใช้กำหนดออปเจคท์ (object) เพื่อให้สามารถเรียกใช้ฟังก์ชั่นสมาชิกภายในได้

void SetLayer (int num)

หน้าที่ เป็นฟังก์ชั่นสมาชิกภายในคลาส BackpNet ใช้กำหนดจำนวนชั้น (layer) ของโครงข่ายประสาทเทียม โดยที่ตัวแปร num คือจำนวนชั้นที่ต้องการกำหนด

void SetInputNodes (int num)

หน้าที่ เป็นฟังก์ชั่นสมาชิกภายในคลาส BackpNet ใช้กำหนดจำนวนอินพุท ที่ชั้นอินพุทของโครงข่ายประสาทเทียม โดยที่ตัวแปร num คือจำนวนอินพุทที่ต้องการกำหนด

void SetHidden1Nodes (int num)

หน้าที่ เป็นฟังก์ชั่นสมาชิกภายในคลาส BackpNet ใช้กำหนดจำนวนนิวรอน ที่ชั้นซ่อนที่ 1 ของโครงข่ายประสาทเทียม โดยที่ตัวแปร num คือจำนวนนิวรอนที่ต้องการกำหนด

void SetHidden2Nodes (int num)

หน้าที่ เป็นฟังก์ชั่นสมาชิกภายในคลาส BackpNet ใช้กำหนดจำนวนนิวรอน ที่ชั้นซ่อนที่ 2 ของโครงข่ายประสาทเทียม โดยที่ตัวแปร num คือจำนวนนิวรอนที่ต้องการกำหนด

void SetHidden3Nodes (int num)

หน้าที่ เป็นฟังก์ชั่นสมาชิกภายในคลาส BackpNet ใช้กำหนดจำนวนนิวรอน ที่ชั้นซ่อนที่ 3 ของโครงข่ายประสาทเทียม โดยที่ตัวแปร num คือจำนวนนิวรอนที่ต้องการกำหนด

void SetOutputNodes (int num)

หน้าที่ เป็นฟังก์ชั่นสมาชิกภายในคลาส BackpNet ใช้กำหนดจำนวนนิวรอนที่ชั้นเอาต์พุท ของโครงข่ายประสาทเทียม โดยที่ตัวแปร num คือจำนวนนิวรอนที่ต้องการกำหนด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

void SetupNetwork ( )

หน้าที่ เป็นฟังก์ชั่นสมาชิกภายในคลาส BackpNet ใช้ในการสร้างโครงข่ายประสาทเทียม หลังจากที่ได้มีการกำหนดรายละเอียดต่าง ๆ แล้ว

void ReadWeights (FILE *fp)

หน้าที่ เป็นฟังก์ชั่นสมาชิกภายในคลาส BackpNet ใช้อ่านค่าน้ำหนักในแฟ้มข้อมูลน้ำหนัก โดยที่ตัวแปรไฟล์พอยท์เตอร์ fp ชี้ไปยังแฟ้มข้อมูลน้ำหนักที่ต้องการอ่านค่าให้แก่โครงข่าย

int FillIOBuffer (FILE *fp)

หน้าที่ เป็นฟังก์ชั่นสมาชิกภายในคลาส BackpNet ใช้อ่านข้อมูลทดสอบในแฟ้มข้อมูลทดสอบลงสู่บัฟเฟอร์ โดยที่ตัวแปรไฟล์พอยท์เตอร์ fp ชี้ไปยังแฟ้มข้อมูลทดสอบที่ต้องการ ฟังก์ชั่นจะรีเทิร์นจำนวนเวคเตอร์หรือแพทเทิร์นให้ถ้าการทำงานสมบูรณ์หรือรีเทิร์นค่า -1 ถ้าฟังก์ชั่นไม่สามารถทำงานได้

void SetupPattern (int p)

หน้าที่ เป็นฟังก์ชั่นสมาชิกภายในคลาส BackpNet ใช้กำหนดข้อมูลทดสอบในบัฟเฟอร์ให้แก่ชั้นอินพุทของโครงข่ายประสาทเทียม โดยที่ตัวแปร p คือลำดับเวคเตอร์ที่ต้องการกำหนดให้แก่โครงข่าย

void ForwardProp ( )

หน้าที่ เป็นฟังก์ชั่นสมาชิกภายในคลาส BackpNet ทำหน้าที่แพร่สัญญาณไปข้างหน้า เพื่อกำเนิดค่าเอาต์พุทที่ชั้นเอาต์พุท

float GetInputsValue (int i)

หน้าที่ เป็นฟังก์ชั่นสมาชิกภายในคลาส BackpNet ทำหน้าที่รีเทิร์นค่าอินพุทที่ชั้นอินพุท โดยที่ตัวแปร i คือลำดับอินพุทที่ต้องการทราบค่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

float GetOutputsValue (int o)

หน้าที่ เป็นฟังก์ชั่นสมาชิกภายในคลาส BackpNet ทำหน้าที่รีเทิร์นค่าเอาต์พุทโดยที่ตัวแปร o คือลำดับนิวรอนที่ต้องการทราบค่า

ข้อสังเกตุในการใช้ฟังก์ชั่นไลบรารี่นี้คือการกำหนดโครงสร้างโครงข่ายประสาทเทียมจะใช้ฟังก์ชั่นไลบรารี่เป็นตัวกำหนดจึงไม่จำเป็นต้องอาศัยแฟ้มข้อมูลโครงสร้าง และข้อจำกัดของการสร้างโครงข่ายคือ จะมีจำนวนชั้นได้มากที่สุด 5 ชั้น และข้อมูลทดสอบภายในแฟ้มข้อมูลทดสอบสามารถมีจำนวนเวคเตอร์หรือแพทเทิร์นได้มากที่สุด 100 เวคเตอร์

ลำดับถัดไปจะแสดงตัวอย่างโปรแกรมทดสอบโครงข่ายประสาทเทียมโดยให้ชื่อว่า SAMPLE.CPP ซึ่งมีรายละเอียดของโปรแกรมดังนี้

```
/*****************************************/
/*** Back-propagation testing program  demo ***/
/*****************************************/
#include<owl\applicat.h>
#include<owl\framewin.h>
#include<owl\dc.h>
#include"bpnlib.h"
#define TEST_FILE "char.tes"
#define WEIGHTS_FILE "char.wgh"
int timer;
BackpNet   *backp;
FILE  *test_file_ptr,  *weights_file_ptr;
int  vectors_in_buffer;
int  run,count;

class TApp : public TApplication
{
  public :
    TApp ( ) : TApplication ( ) { }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    void InitMainWindow ( ) ;
};

class TWndw : public TFrameWindow
{
  public :
    TWndw(TWindow  *parent, const char far  *title);
  protected :
    void SetupWindow ( );
    void CleanupWindow ( );
    void EvTimer (UINT);
    void CmRunNext ( );
    void Paint (TDC &, BOOL, TRect &);
  DECLARE_RESPONSE_TABLE(TWndw);
};

DEFINE_RESPONSE_TABLE1(TWndw,TFrameWindow)
  EV_WM_TIMER,
  EV_COMMAND(101, CmRunNext),
END_RESPONSE_TABLE;

Twndw :: TWndw (TWindow *parent,  const char far *title);
            TFrameWindow (parent, title)
{
  Attr.X = 20;
  Attr.Y = 20;
  Attr.W = 500;
  Attr.H = 400;
  AssingnMenu(100);
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
void TWndw :: SetupWindow ( )
{
  TFramWindow :: SetupWindow ( );
  backp = new BackpNet;   // define object of network
  // set number layer of network
  backp -> SetLayer (4);
  // set input nodes(layer1) to 35
  backp -> SetInputNodes(35):
  // set hidden neuron(layer2) to 10;
  backp -> SetHidden1Nodes(10);
  // set hidden neuron(layer3) to 10
  backp -> SetHidden2Nodes(10);
  // set output neuron(layer4) to 8
  backp -> SetOutputNodes (8);
  // construct the network
  backp -> SetupNetwork ( );
  // open test file
  test_file_ptr = fopen(TEST_FILE, "r");
  // open weight file
  weights_file_ptr = fopen(WEIGHTS_FILE, "r");
  // load weights to network.
  backp -> ReadWeights(weights_file_ptr);
  // fill data test to buffer.
  vectors_in_buffer = back -> FillIOBuffer(test_file_ptr);
  run = 1;
  count = 0;
  timer = SetTimer(1, 10, 0);
  if(timer == 0)
    MessageBox("No timers!","Error",MB_OK);
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
void TWndw :: CleanupWindow( )
{
  fclose(test_file_ptr);
  fclose(weights_file_ptr);
  delete backp;
  KillTimer(timer);
}

void TWndw :: CmRunNext ( )
{
  run=1;
}

void TWndw :: EvTimer(UINT)
{
  if(run){
    if(count < vectors_in_buffer){
      // set pattern to input layer
      backp -> SetupPattern(count);
      // forward propagate to output layer
      backp -> Forward Prop ( );
      Invalidate( );
      count++;
    }
  }
  run=0;
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
void TWndw :: Paint (TDC  &dc, BOOL, TRect  &)
{
  char str[100];
  int i, j, x, y, color;
  float data;
  sprintf(str, "INPUT  PATTERN");
  dc.SetBkMode(TRANSPARENT);
  dc.SetTextColor(TColor(0, 0, 255));
  dc.TextOut(50, 10, str);
  sprintf(str, "OUTPUT  VALUES");
  dc.SetTextColor(Tcolor(0, 0, 255));
  dc.TextOut(200, 280, str);
  x = 0;
  y = 30;
  for(i =0, j=0; i < 35; i++){
    //get input pattern value.
    data = backp -> GetInputsValue(i);
    if( j == 5){
      j = 1;
      x = 30;
      y = y + 30;
    }
    else{
      x = x+30;
      j++;
    }
    color = (int) (data * 150) + 100;
    Tbrush  *Brush = new TBrush(RGB(color, 0, 0));
    dc.SelectObject( *Brush);
    dc.Rectangle(x, y, x+30, y+30);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    delete Brush;
  }
  for(i=0; i < 8; i++){
    // get output value
    data = backp -> GetOutputsValue(i);
    sprintf(str, "%0.3f", data);
    dc.SetTextColor(TColor(255, 0, 0));
    dc.TextOut((60*i)+20, 300, str);
  }
}
void TApp :: InitMainWidow ( )
{
  TFrameWindow  *wndw = new TWndw(0, "BPN DEMO");
  SetMainWindow(wndw);
}
int OwlMain(int, char*[ ])
{
  TApp App;
  return App.Run( );
}
```

และยังต้องสร้างแฟ้มทรัพยากรเพื่อคอมไพล์ร่วมกับโปรแกรม SAMPLE.CPP โดยแฟ้มทรัพยากรนี้ต้องมีนามสกุล .RC ซึ่งตัวอย่างนี้ให้ชื่อว่า SAMPLE.RC มีรายละเอียดดังนี้

```
100 MENU
{
  POPUP "& Run"
  {
    MENUITEM "&Next   input", 101
  }
}
```

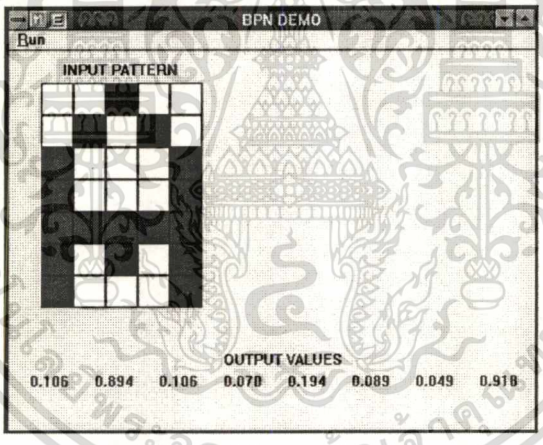
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำความเข้าใจในการทำงานของโปรแกรมนั้น ผู้ศึกษาต้องมีความรู้พื้นฐานด้านภาษาซีพลัสพลัส และมีประสบการณ์การเขียนโปรแกรมบนระบบปฏิบัติการวินโดว์โดยใช้คอมไพเลอร์ของบอร์แลนด์ซีพลัสพลัสเวอร์ชั่น 4 บ้างพอสมควร การคอมไพล์โปรแกรมเพื่อให้ทำงานภายใต้ระบบปฏิบัติการวินโดว์ต้องสร้างโปรเจคท์ไฟล์ซึ่งแฟ้มนี้จะอินคลูด(include)แฟ้มโปรแกรมต่างๆ ที่จำเป็นต้องคอมไพล์ร่วมกัน แฟ้มเหล่านี้คือ

1. แฟ้ม SAMPLE.CPP
2. แฟ้ม SAMPLE.RC
3. แฟ้ม BPNLIB.LIB

และต้องกำหนดเมมโมรี่โมเดลให้เป็นลาร์จ (Large memory model) เมื่อคอมไพล์แล้วก็จะได้โปรแกรมทดสอบโครงข่ายประสาทเทียมที่สามารถทำงานได้บนระบบปฏิบัติการวินโดว์โดยจะอธิบายการทำงานต่อไป

เมื่อโปรแกรม SAMPLE.EXE เริ่มต้นทำงานจะปรากฏภาพที่จอภาพดังนี้



ภาพที่ 80 แสดงการทำงานของโปรแกรมตัวอย่าง

ตรงตำแหน่งอินพุทแพทเทิร์นจะปรากฏภาพอักขระของอินพุทแพทเทิร์นข้อมูลทดสอบลำดับแรก คืออักขระ A ที่มีสัญญาณรบกวนและที่ตำแหน่งเอาต์พุทแวลูแสดงค่าของเอาต์พุทที่ได้จากโครงข่ายประสาทเทียมความหมายของค่าเอาต์พุทนี้กำหนดให้มีการตั้งเทรสโฮลด์คือ

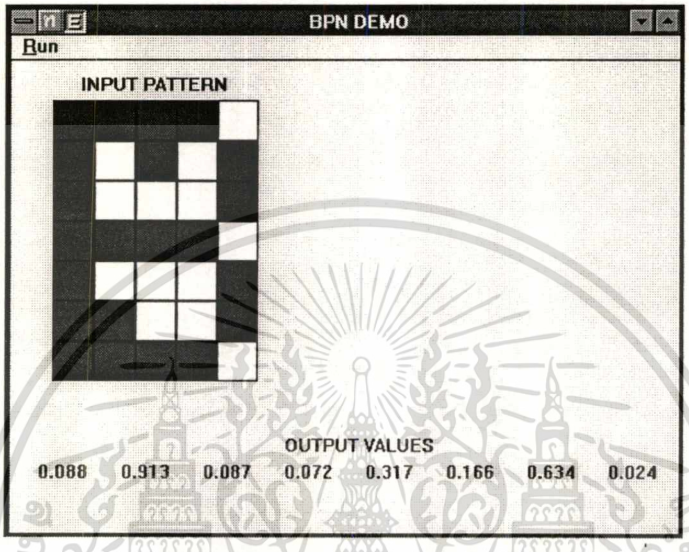
if Output(n) >= 0.5 then Output(n) = 1

else Output(n) = 0

จากภาพที่ 80 จะได้ค่าที่เอาต์พุทของโครงข่ายคือ 0.106 0.894 0.106 0.070 0.194 0.089 0.049 0.918 เมื่อผ่านเทรสโฮลด์แล้วจะได้ 01000001 เป็นค่าไบนารีฐานสองเมื่อแปลงเป็นฐานสิบหกจะได้ 41h ตรงกับรหัส ASCII ของอักขระ A แสดงว่าโครงข่ายกำเนิดค่าเอาต์พุทของอักขระ A ในกรณีนี้ได้ถูกต้องถึงแม้จะมีสัญญาณรบกวนอยู่ก็ตาม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

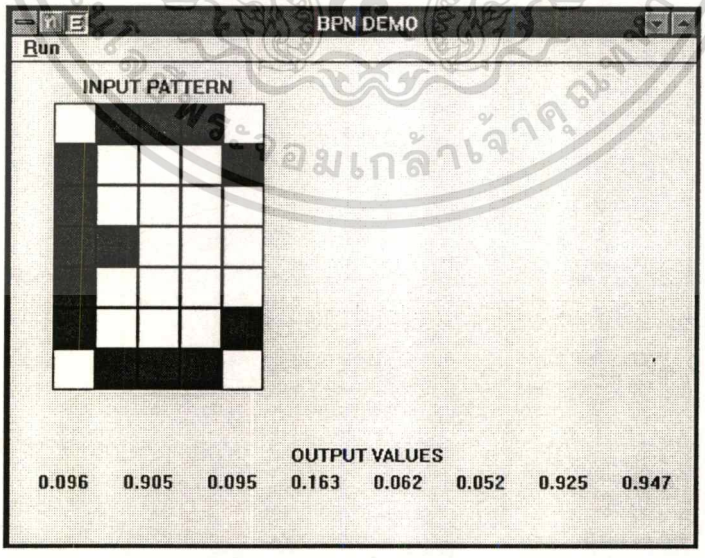
เราสามารถดูแพทเทิร์นถัดไปได้โดยการเลือกการทำงานที่ Run บนเมนูบาร์ แล้วเลือก Next input ก็จะปรากฏอักขระ B ขึ้นแทนอักขระ A แล้วก็จะได้ค่าเอาต์พุทของอักขระ B ที่ตำแหน่งเอาต์พุทแวลู



ภาพที่ 81 แสดงผลลัพธ์ของอักขระ B

ค่าเอาต์พุทคือ 42h

ค่าที่ถูกต้องคือ 42h

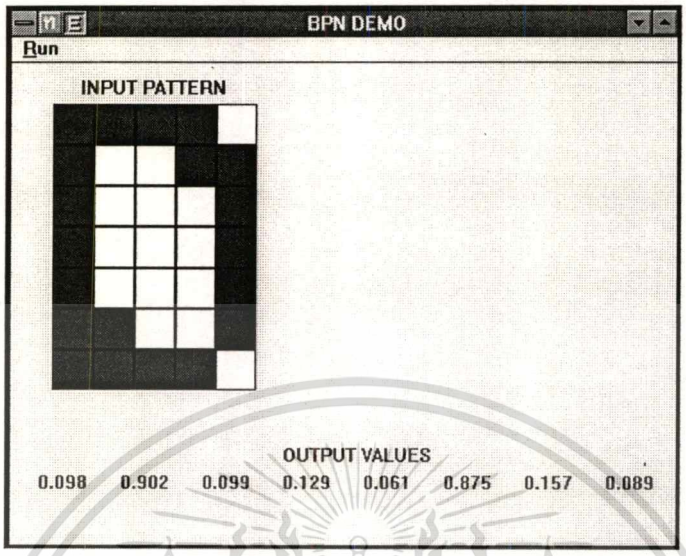


ภาพที่ 82 แสดงผลลัพธ์ของอักขระ C

ค่าเอาต์พุทคือ 43h

ค่าที่ถูกต้องคือ 43h

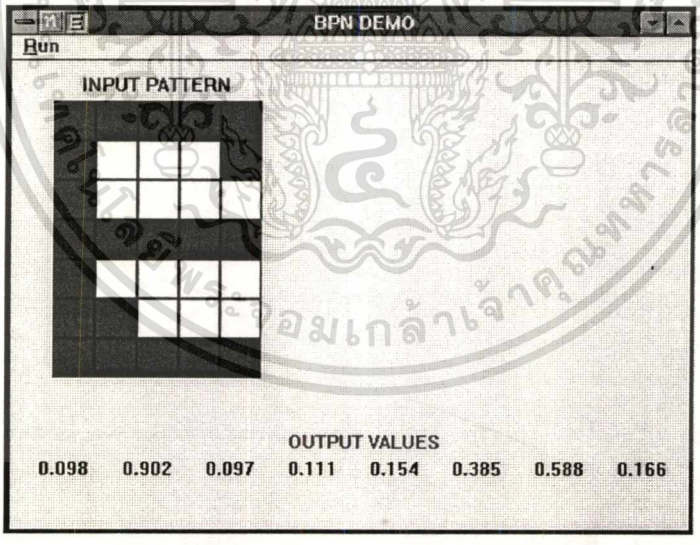
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 83 แสดงผลลัพธ์ของอักขระ D

ค่าเอาต์พุทคือ 44h

ค่าที่ถูกต้องคือ 44h

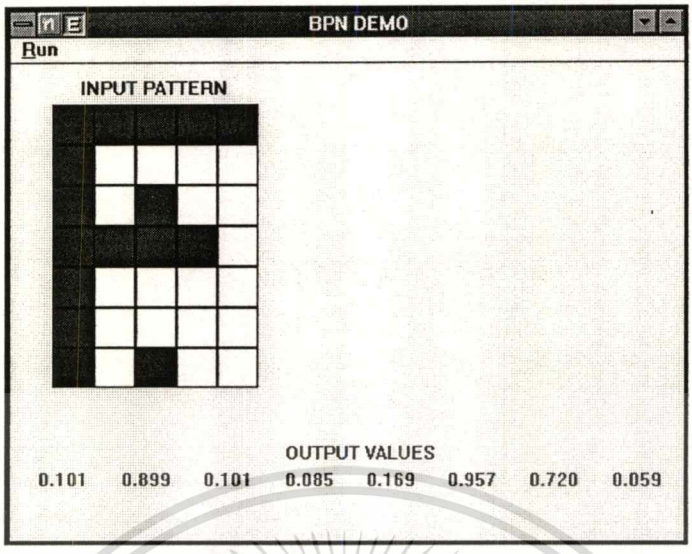


ภาพที่ 84 แสดงผลลัพธ์ของอักขระ E

ค่าเอาต์พุทคือ 42h

ค่าที่ถูกต้องคือ 45h

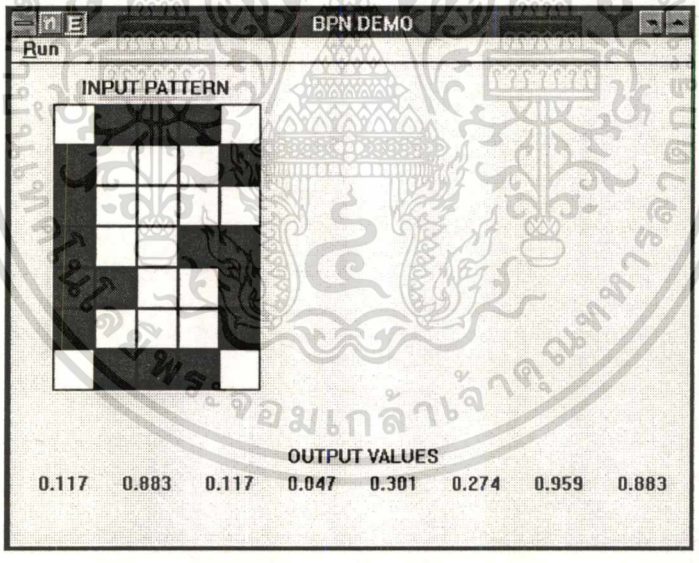
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 85 แสดงผลลัพธ์ของอักขระ F

ค่าเอาต์พุทคือ 46h

ค่าที่ถูกต้องคือ 46h

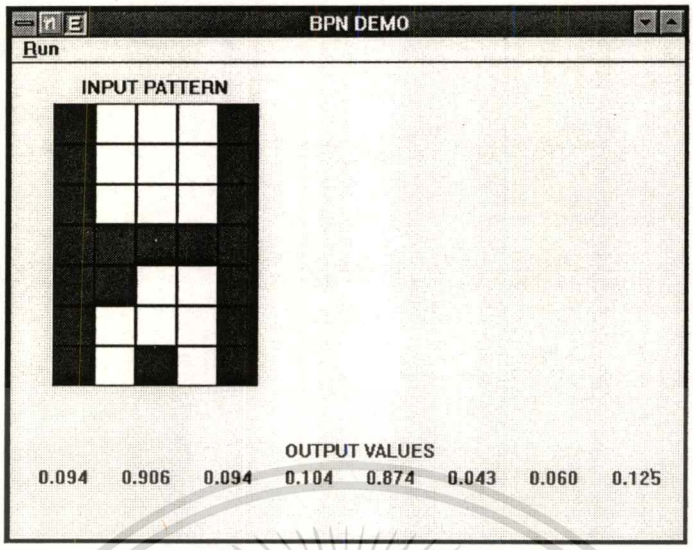


ภาพที่ 86 แสดงผลลัพธ์ของอักขระ G

ค่าเอาต์พุทคือ 43h

ค่าที่ถูกต้องคือ 47h

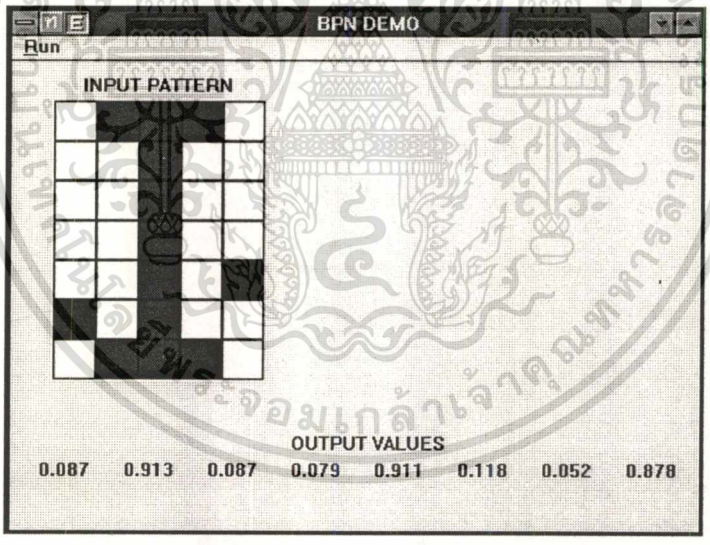
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 87 แสดงผลลัพธ์ของอักขระ H

ค่าเอาต์พุทคือ 48h

ค่าที่ถูกต้องคือ 48h

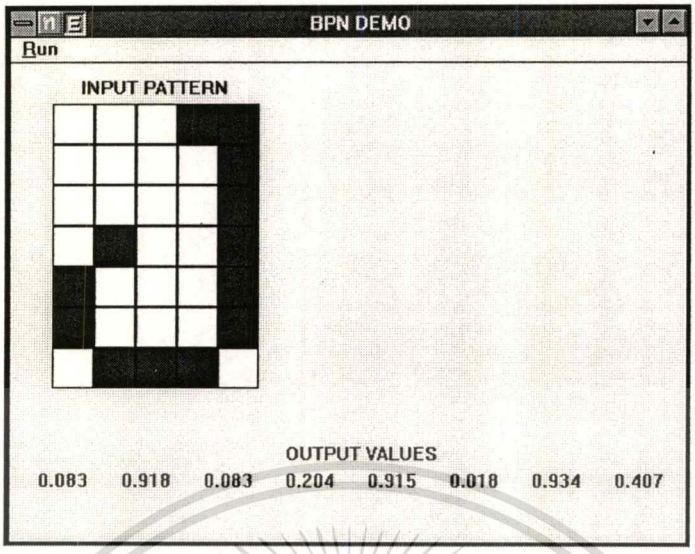


ภาพที่ 88 แสดงผลลัพธ์ของอักขระ I

ค่าเอาต์พุทคือ 49h

ค่าที่ถูกต้องคือ 49h

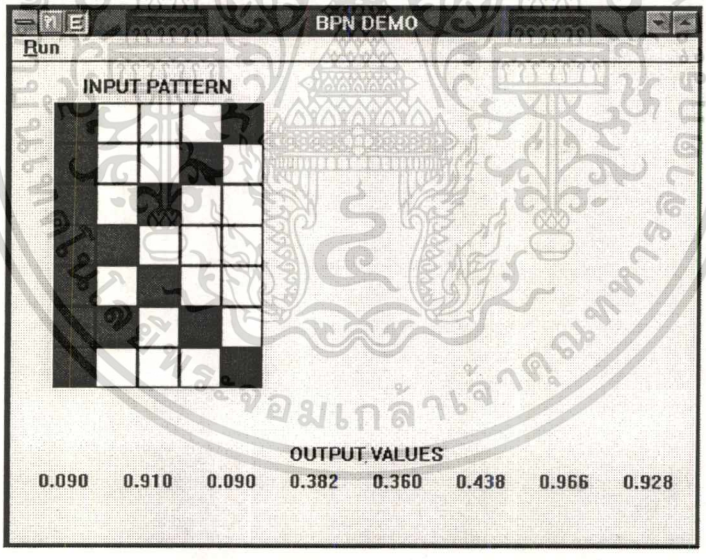
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 89 แสดงผลลัพธ์ของอักขระ J

ค่าเอาต์พุทคือ 4Ah

ค่าที่ถูกต้องคือ 4Ah

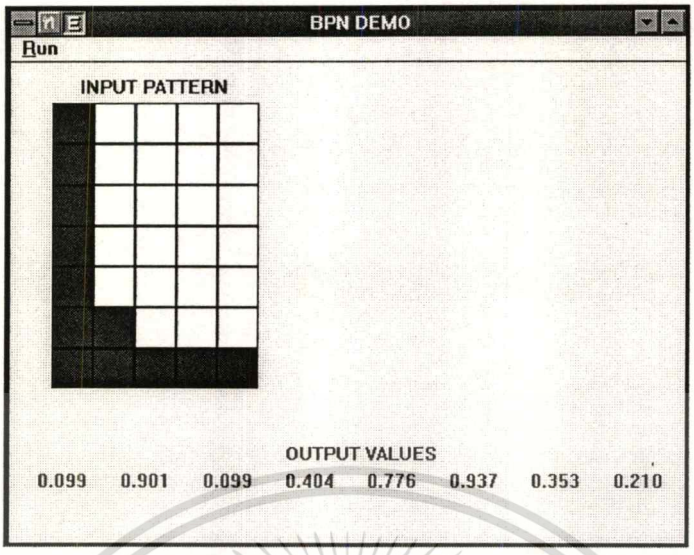


ภาพที่ 90 แสดงผลลัพธ์ของอักขระ K

ค่าเอาต์พุทคือ 43h

ค่าที่ถูกต้องคือ 4Bh

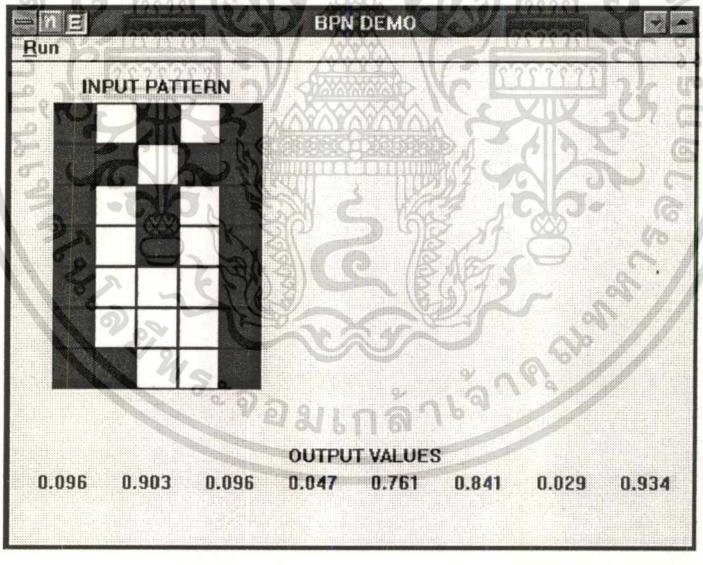
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 91 แสดงผลลัพธ์ของอักขระ L

ค่าเอาต์พุทคือ 4Ch

ค่าที่ถูกต้องคือ 4Ch

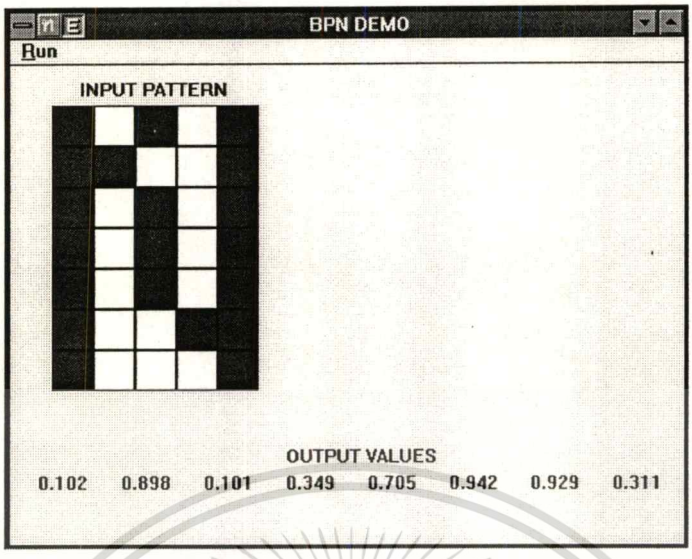


ภาพที่ 92 แสดงผลลัพธ์ของอักขระ M

ค่าเอาต์พุทคือ 4Dh

ค่าที่ถูกต้องคือ 4Dh

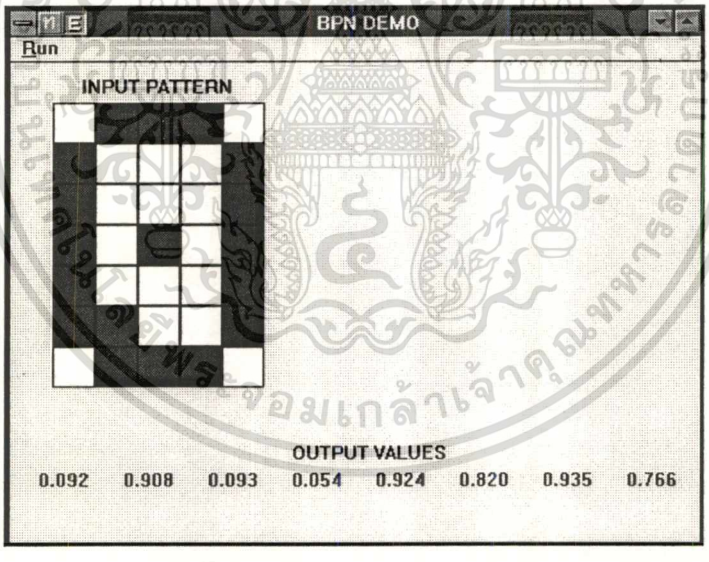
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 93 แสดงผลลัพธ์ของอักขระ N

ค่าเอาต์พุทคือ 4Eh

ค่าที่ถูกต้องคือ 4Eh

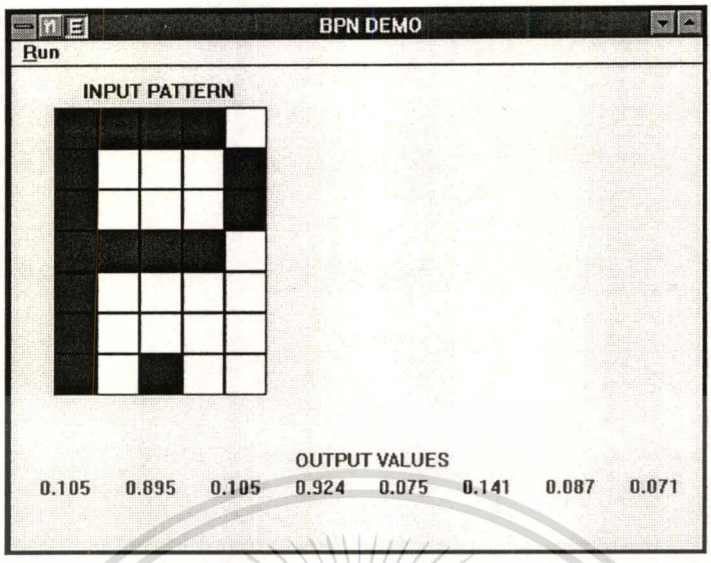


ภาพที่ 94 แสดงผลลัพธ์ของอักขระ O

ค่าเอาต์พุทคือ 4Fh

ค่าที่ถูกต้องคือ 4Fh

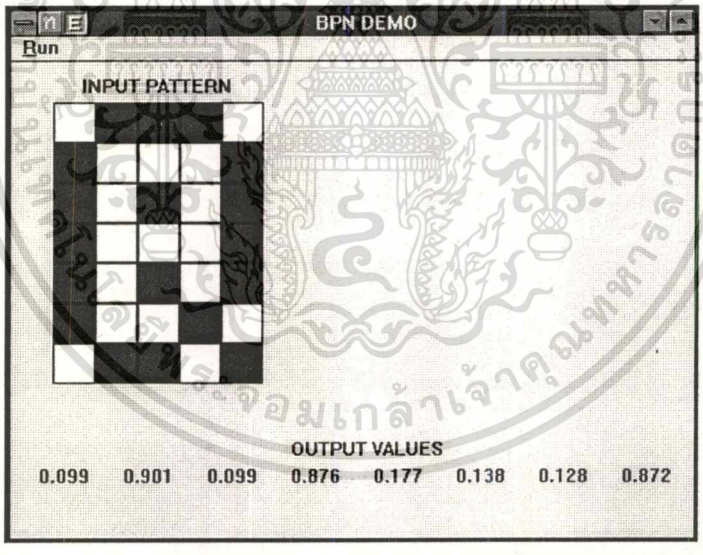
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 95 แสดงผลลัพธ์ของอักขระ P

ค่าเอาต์พุทคือ 50h

ค่าที่ถูกต้องคือ 50h

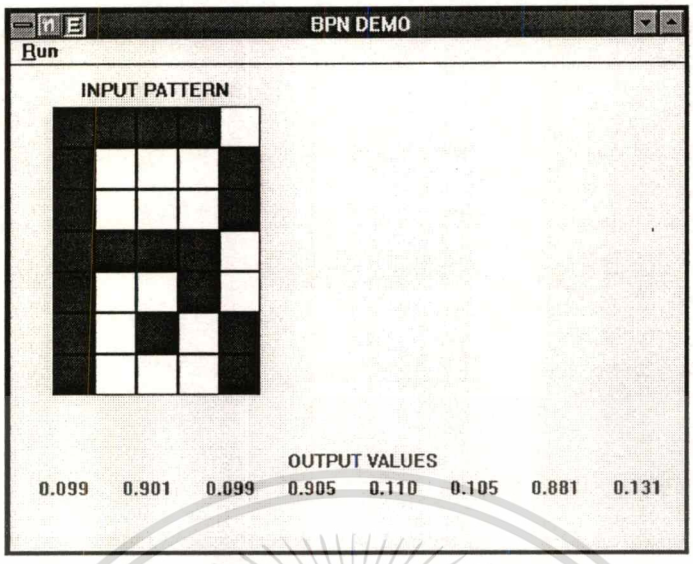


ภาพที่ 96 แสดงผลลัพธ์ของอักขระ Q

ค่าเอาต์พุทคือ 51h

ค่าที่ถูกต้องคือ 51h

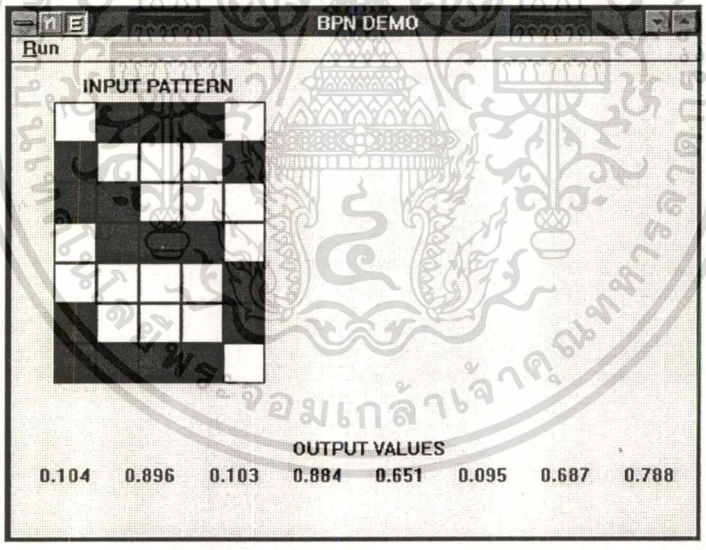
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 97 แสดงผลลัพธ์ของอักขระ R

ค่าเอาต์พุทคือ 52h

ค่าที่ถูกต้องคือ 52h

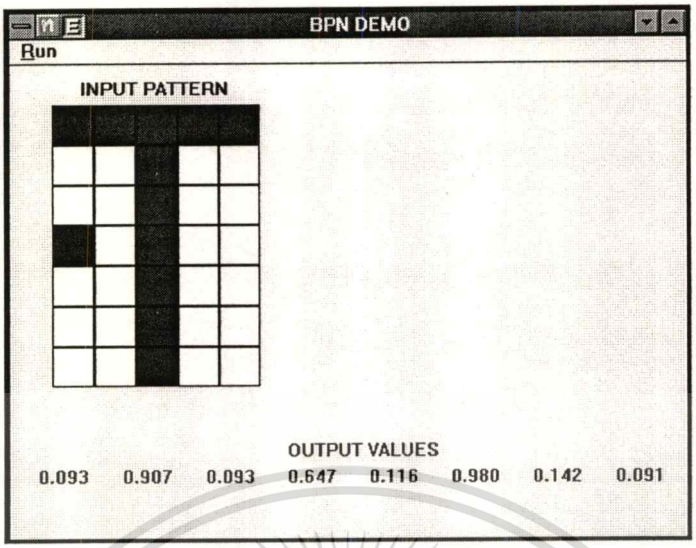


ภาพที่ 98 แสดงผลลัพธ์ของอักขระ S

ค่าเอาต์พุทคือ 5Bh

ค่าที่ถูกต้องคือ 53h

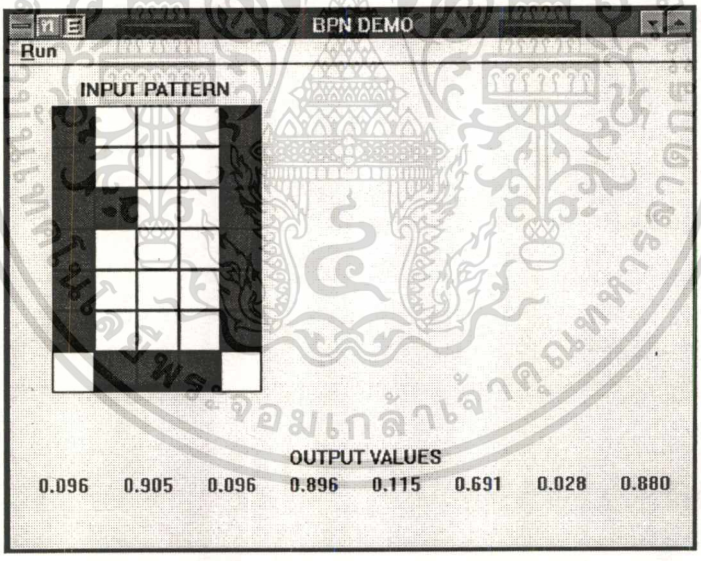
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 99 แสดงผลลัพธ์ของอักขระ T

ค่าเอาต์พุทคือ 54h

ค่าที่ถูกต้องคือ 54h

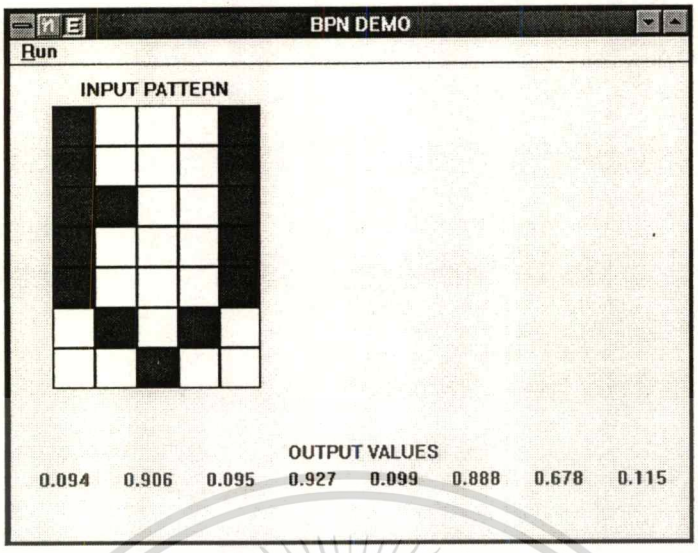


ภาพที่ 100 แสดงผลลัพธ์ของอักขระ U

ค่าเอาต์พุทคือ 55h

ค่าที่ถูกต้องคือ 55h

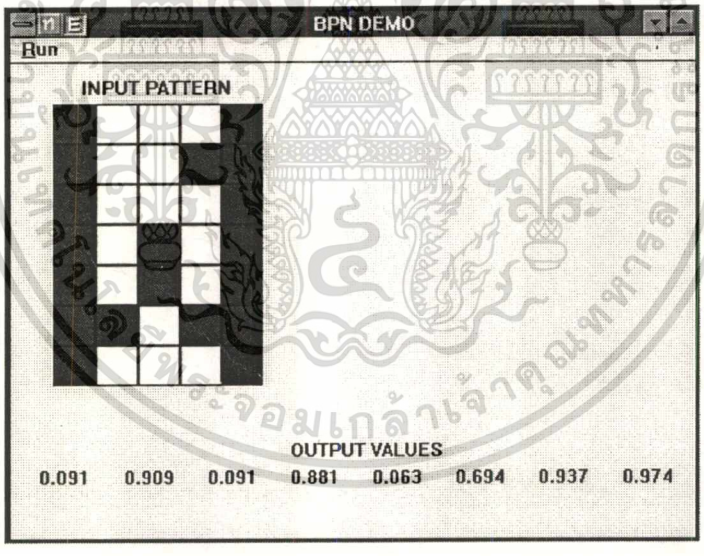
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 101 แสดงผลลัพธ์ของอักขระ V

ค่าเอาต์พุทคือ 56h

ค่าที่ถูกต้องคือ 56h

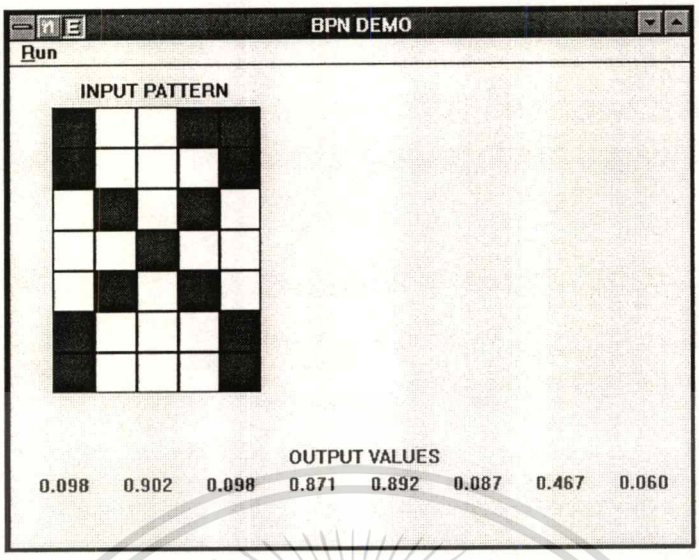


ภาพที่ 102 แสดงผลลัพธ์ของอักขระ W

ค่าเอาต์พุทคือ 57h

ค่าที่ถูกต้องคือ 57h

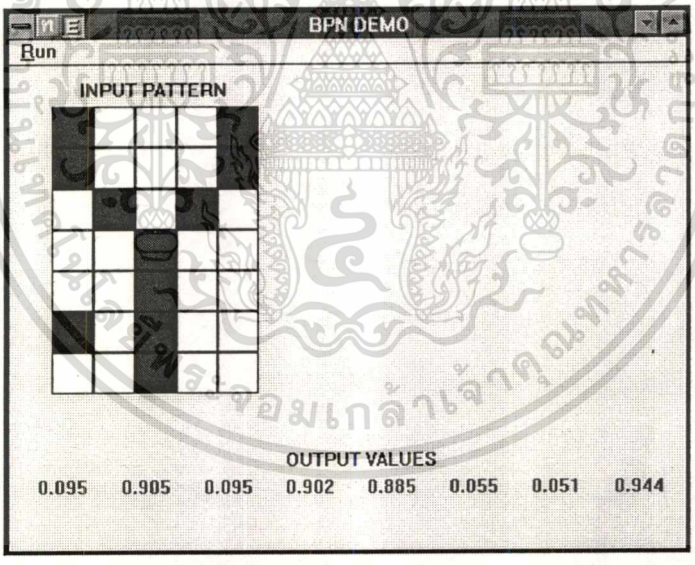
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 103 แสดงผลลัพธ์ของอักขระ X

ค่าเอาต์พุทคือ 58h

ค่าที่ถูกต้องคือ 58h

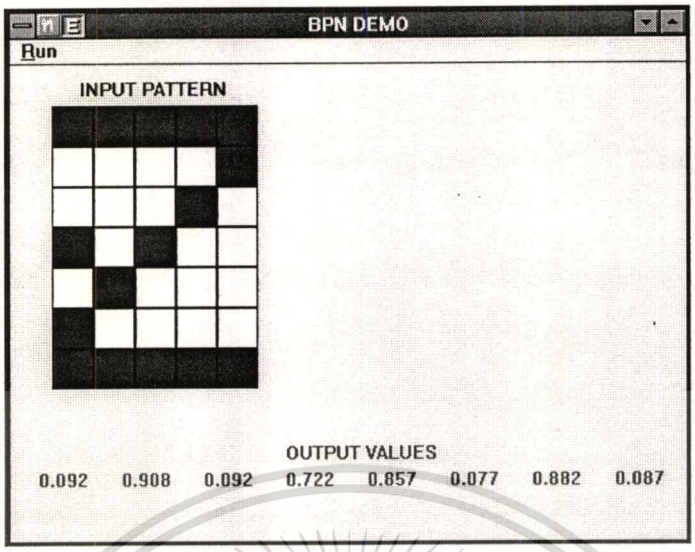


ภาพที่ 104 แสดงผลลัพธ์ของอักขระ Y

ค่าเอาต์พุทคือ 59h

ค่าที่ถูกต้องคือ 59h

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 105 แสดงผลลัพธ์ของอักขระ Z

ค่าเอาต์พุทคือ 5Ah

ค่าที่ถูกต้องคือ 5Ah

จากการทดลองจะเห็นว่าแม้จะเพิ่มสัญญาณรบกวนให้กับข้อมูลทดสอบความสามารถในการตัดสินใจก็ยังคงถูกต้องพอสมควร การเพิ่มประสิทธิภาพในการตัดสินใจนั้นขึ้นอยู่กับปริมาณของข้อมูลฝึกหัดด้วยยิ่งปริมาณของข้อมูลฝึกหัดมากขึ้นเท่าไรความถูกต้องในการตัดสินใจก็มากขึ้นตามไปด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การประยุกต์ใช้กับการรู้จำรูปทรงเรขาคณิต

ได้จัดแบ่งลักษณะการทดลองเป็น 2 ลักษณะคือ

1. การฝึกหัดโครงข่ายประสาทเทียมโดยอาศัยโปรแกรมจำลองโครงข่ายประสาทเทียมโดยหลักการแพร่ย้อนกลับ

2.การทดสอบโครงข่ายประสาทเทียมโดยอาศัยการเขียนโปรแกรมเพื่อเรียกใช้ฟังก์ชั่นไลบรารี่ที่มีมาให้

การฝึกหัดโครงข่ายประสาทเทียม

ได้มีการจัดเตรียมแฟ้มข้อมูลฝึกหัดเพื่อที่จะใช้ฝึกหัดโครงข่ายแฟ้มนี้ให้ชื่อว่า GEOME.TRN ข้อมูลภายในแฟ้มข้อมูลฝึกหัดได้จัดแสดงไว้ในภาคผนวกที่ท้ายเล่ม ข้อมูลภายในแฟ้ม GEOME.TRN ได้ถูกกำหนดให้มีข้อมูลฝึกหัดจำนวน 6 เวคเตอร์ หรือ 6 แพทเทิร์น โดยแต่ละแพทเทิร์นประกอบด้วยลำดับอินพุท-เอาต์พุท ข้อมูลฝึกหัดอินพุทหรืออินพุทเวคเตอร์ได้มาจากค่าของพิกเซล (pixel) ซึ่งกำหนดไว้ให้มีขนาด 10x10 พิกเซล โดยขอยกตัวอย่างอินพุทเวคเตอร์แรกในแฟ้มข้อมูล GEOME.TRN คือ

0.9 0.9 0.9 0.9 0.9 0.9 0.9 0.9 0.9 0.9
0.9 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.9
0.9 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.9
0.9 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.9
0.9 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.9
0.9 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.9
0.9 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.9
0.9 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.9
0.9 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.9
0.9 0.9 0.9 0.9 0.9 0.9 0.9 0.9 0.9 0.9

เป็นข้อมูลของรูปทรงสี่เหลี่ยม จากขนาดที่กำหนดไว้ 10x10 พิกเซล ซึ่งทำให้เป็นข้อมูลอินพุทเวคเตอร์ขนาด 100 อินพุท ส่วนเอาต์พุทเวคเตอร์หรือค่าเป้าหมายนั้น ถูกกำหนดให้มีค่าอยู่ในช่วงระหว่าง 0.0 ถึง 1.0 เพราะว่าโปรแกรมจำลองโครงข่ายประสาทเทียมโดยหลักการแพร่ย้อนกลับนี้ใช้ฟังก์ชั่นกระตุ้นไบนารี่ซิกมอยด์ แต่อย่างไรก็ตามค่าเป้าหมายนี้จะกำหนดไว้เพียงสองค่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คือ 0.1 และ 0.9 โดยจะแทนค่าเหล่านี้เป็นค่าไบนารี่ 0 และ 1 โดยให้ผ่านฟังก์ชั่นกำหนดเทรสโฮลด์ (threshold) คือ

if target = 0.1 then target = 0

if target = 0.9 then target = 1

ตัวอย่างค่าเอาต์พุทเวคเตอร์ของรูปทรงสี่เหลี่ยมคือ 0.1 0.9 เมื่อผ่านฟังก์ชั่นเทรสโฮลด์แล้วจะได้ค่าเลขไบนารี่ คือ 01 ดังนั้นที่เอาต์พุทเวคเตอร์จึงมีขนาด 2 เอาต์พุท ส่วนรูปทรงอื่นๆที่กำหนดไว้คือรูปทรงกลมและรูปทรงสามเหลี่มโดยกำหนดค่าเอาต์พุทแทนรูปทรงทั้งสองคือ 0.9 0.1 และ 0.9 0.9 ตามลำดับ

อีกแฟ้มหนึ่งที่ต้องจัดเตรียมในการฝึกหัดโครงข่ายนี้ก็คือแฟ้มข้อมูลโครงสร้างโครงข่ายประสาทเทียม โดยได้มีการสร้างแฟ้ม GEOME.NET ซึ่งมีข้อมูลกำหนดโครงสร้างดังนี้

LAYER: 1
NODE : 100
LAYER: 2
NODE : 10
LAYER: 3
NODE : 10
LAYER: 4
NODE : 2
MOMENTUM : 0.0
LEARN-TOLERANCE : 0.01
MAX-CYCLES : 10000

ข้อมูลโครงสร้างได้กำหนดให้โครงข่ายมีอินพุท 100 อินพุท ที่ชั้นซ่อนที่ 1 มี 10 นิวรอน ชั้นซ่อนที่ 2 มี 10 นิวรอน และที่ชั้นเอาต์พุท มี 2 นิวรอน กำหนดเทอมโมเมนต์ตัมเท่ากับศูนย์ กำหนดอัตราการเรียนรู้เท่ากับ 0.5 กำหนดค่าผิดพลาดต่ำสุดที่สามารถยอมรับได้ไว้ที่ 0.01 และกำหนดจำนวนรอบสูงสุดเพื่อใช้ในการเรียนรู้ไว้ที่ 10000 รอบ

หลังจากได้จัดเตรียมแฟ้มต่าง ๆ ที่จำเป็นต้องใช้เรียบร้อยแล้ว กระบวนการถัดไปคือ การใช้งานโปรแกรมจำลองโครงข่ายประสาทเทียมเพื่อการฝึกหัด โดยได้ทำการฝึกหัดโครงข่ายตามขั้นตอนที่ได้กล่าวผ่านมาแล้วในเรื่องของการใช้โปรแกรม สิ่งที่ได้จากการฝึกหัดโครงข่ายนี้ก็คือแฟ้ม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อมูลน้ำหนักซึ่งให้ตั้งชื่อแฟ้มข้อมูลน้ำหนักนี้ไว้ชื่อว่า GEOME.WGH โดยจะนำแฟ้มข้อมูลน้ำหนักนี้ไปใช้ในการเขียนโปรแกรมทดสอบโครงข่ายประสาทเทียมโดยจะกล่าวรายละเอียดในหัวข้อถัดไป

<u>การทดสอบโครงข่ายประสาทเทียม</u>

ได้มีการจัดเตรียมแฟ้มข้อมูลทดสอบเพื่อที่จะใช้ทดสอบโครงข่ายแฟ้มนี้ให้ชื่อว่า GEOME.TES ซึ่งข้อมูลทั้งหมดภายในแฟ้มนี้จัดแสดงไว้ที่ภาคผนวกท้ายเล่ม จากข้อมูลภายในแฟ้ม GEOME.TES ได้ถูกกำหนดให้มีข้อมูลทดสอบ 14 เวคเตอร์ หรือ 14 แพทเทิร์นเพื่อนำข้อมูลเหล่านี้ไปทดสอบหาเอาต์พุทของโครงข่าย ส่วนอีกสองแฟ้มที่จำเป็นสำหรับการทดสอบโครงข่ายคือแฟ้มข้อมูลโครงสร้าง GEOME.NET ที่ได้เตรียมไว้ในขั้นตอนการฝึกหัดโครงข่ายและแฟ้มข้อมูล GEOME.WGH ที่ได้ทำการจัดเก็บไว้ในขั้นตอนฝึกหัดเช่นกัน หลังจากที่ได้จัดเตรียมแฟ้มต่างๆ ได้ครบถ้วนแล้วก็สามารถที่จะทำการทดสอบหาเอาต์พุทของโครงข่ายได้ วิธีการทดสอบนั้นสามารถจะกระทำได้สองวิธีคือ

1. ใช้โปรแกรมจำลองโครงข่ายประสาทเทียมโดยวิธีการแพร่ย้อนกลับ ซึ่งวิธีนี้ได้กล่าวถึงการใช้งานแล้วในเรื่องของการใช้งานโปรแกรมเพื่อการทดสอบ
2. สร้างโปรแกรมทดสอบโดยใช้ภาษาซีพลัสพลัสและบอร์แลนด์ซีคอมไพเลอร์เวอร์ชั่น 4 ร่วมกับฟังก์ชั่นไลบรารี่ที่มีมาให้กับโปรแกรม

ในการทดลองนี้จะใช้วิธีที่สองคือสร้างโปรแกรมทดสอบขึ้นโดยเชื่อมโยงกับฟังก์ชั่นไลบรารี่ ซึ่งมีประโยชน์มากสำหรับการนำไปประยุกต์ใช้กับงานด้านอื่นๆเพราะผู้ใช้สามารถสร้างโครงข่ายประสาทเทียม เพื่อนำค่าเอาต์พุทจากการทดสอบไปใช้งานตามความต้องการได้

ลำดับถัดไปจะแสดงตัวอย่างโปรแกรมทดสอบโครงข่ายประสาทเทียมโดยให้ชื่อว่า SAMPLE2.CPP ซึ่งมีรายละเอียดของโปรแกรมดังนี้

```
/*****************************************/
/*** Back-propagation demo testing program ***/
/*****************************************/
#include <owl\applicat.h>
#include <owl\framewin.h>
#include <owl\dc.h>
#include "bpnlib.h"
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
#define  TEST_FILE     "geome.tes"
#define  WEIGHTS_FILE  "geome.wgh"

int timer;

BackpNet *backp;
FILE *test_file_ptr, *weights_file_ptr;
int vectors_in_buffer;
int run, count;

class TApp : public TApplication
{
  public:
    TApp() : TApplication() {}
    void InitMainWindow();
};

class TWndw : public TFrameWindow
{
  public:
    TWndw(TWindow *parent, const char far *title);
  protected:
    void SetupWindow();
    void CleanupWindow();
    void EvTimer(UINT);
    void CmRunNext();
    void Paint(TDC&, BOOL, TRect&);
    DECLARE_RESPONSE_TABLE(TWndw);
};
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
DEFINE_RESPONSE_TABLE1(TWndw, TFrameWindow)
  EV_WM_TIMER,
  EV_COMMAND(101, CmRunNext),
END_RESPONSE_TABLE;

TWndw::TWndw(TWindow *parent, const char far *title):
                TFrameWindow(parent, title)
{
  Attr.X = 20;
  Attr.Y = 20;
  Attr.W = 500;
  Attr.H = 400;
  AssignMenu(100);
}

void TWndw::SetupWindow()
{
  TFrameWindow::SetupWindow();
  backp = new BackpNet;          // define object of network
  backp->SetLayer(4);            // set number layer of network
  backp->SetInputNodes(100);     // set input nodes(layer 1) to 100
  backp->SetHidden1Nodes(10);  // set hidden nodes(layer 2) to 10
  backp->SetHidden2Nodes(10);  // set hidden nodes(layer 3) to 10
  backp->SetOutputNodes(2);      // set output nodes(layer 4) to 2
  backp->SetupNetwork();         // construct the network
  test_file_ptr = fopen(TEST_FILE, "r");       // open test file
  weights_file_ptr = fopen(WEIGHTS_FILE, "r"); // open weight file
  backp->ReadWeights(weights_file_ptr);        // load weights to network
  // fill data test to buffer
  vectors_in_buffer = backp->FillIOBuffer(test_file_ptr);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  run = 1;
  count = 0;
  timer = SetTimer(1, 10, 0);
  if(timer == 0)
    MessageBox("No timers!", "Error", MB_OK);
}

void TWndw::CleanupWindow()
{
  fclose(test_file_ptr);
  fclose(weights_file_ptr);
  delete backp;
  KillTimer(timer);
}

void TWndw::CmRunNext()
{
  run = 1;
}

void TWndw::EvTimer(UINT)
{
  if(run){
    if(count<vectors_in_buffer){
      backp->SetupPattern(count);  // set pattern to input layer
      backp->ForwardProp();        // forward propagate to output layer
      Invalidate();
      count++;
    }
  }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  run = 0;
}

void TWndw::Paint(TDC& dc, BOOL, TRect&)
{
  char str[100];
  int i, j, x, y, color;
  float data, d1, d2;
  sprintf(str,"INPUT PATTERN");
  dc.SetBkMode(TRANSPARENT);
  dc.SetTextColor(TColor(0,0,255));
  dc.TextOut(45, 10, str);
  sprintf(str,"OUTPUT VALUES");
  dc.SetTextColor(TColor(0,0,255));
  dc.TextOut(20, 280, str);
  sprintf(str,"DECIDE PICTURE TO:");
  dc.SetTextColor(TColor(0,0,255));
  dc.TextOut(200, 300, str);
  x = 0;
  y = 30;
  for(i=0, j=0; i<100; i++){
    data = backp->GetInputsValue(i); // get input pattern value
    if(j==10){
      j = 1;
      x = 20;
      y = y+20;
    }
    else{
      x= x+20;
      j++;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  }
  color = (int)(data * 150)+100;
  TBrush *Brush = new TBrush(RGB(color,0,0));
  dc.SelectObject(*Brush);
  dc.Rectangle(x,y,x+20,y+20);
  delete Brush;
}
for(i=0; i<2; i++){
  data = backp->GetOutputsValue(i); // get output value
  if(i==0) d1=data;
  else d2=data;
  sprintf(str,"%0.3f",data);
  dc.SetTextColor(TColor(255,0,0));
  dc.TextOut((60*i)+20, 300, str);
}
sprintf(str,"                    ");
dc.SetTextColor(TColor(255,0,0));
dc.TextOut(350, 300, str);
if(d1<0.5 && d2>=0.5){
  sprintf(str,"QUADRILATERAL");
  dc.SetTextColor(TColor(255,0,0));
  dc.TextOut(350, 300, str);
}
if(d2<0.5 && d1>=0.5){
  sprintf(str,"CYCLE");
  dc.SetTextColor(TColor(255,0,0));
  dc.TextOut(350, 300, str);
}
if(d1>=0.5 && d2>=0.5){
  sprintf(str,"TRIANGLE");
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    dc.SetTextColor(TColor(255,0,0));
    dc.TextOut(350, 300, str);
  }
}

void TApp::InitMainWindow()
{
  TFrameWindow *wndw = new TWndw(0, "BPN DEMO");
  SetMainWindow(wndw);
}

int OwlMain(int, char*[])
{
  TApp App;
  return App.Run();
}
```

และยังต้องสร้างแฟ้มทรัพยากรเพื่อคอมไพล์ร่วมกับโปรแกรม SAMPLE2.CPP โดยแฟ้มทรัพยากรนี้ต้องมีนามสกุล .RC ซึ่งตัวอย่างนี้ให้ชื่อว่า SAMPLE2.RC มีรายละเอียดดังนี้

```
100 MENU
{
  POPUP "& Run"
  {
    MENUITEM "&Next   input", 101
  }
```

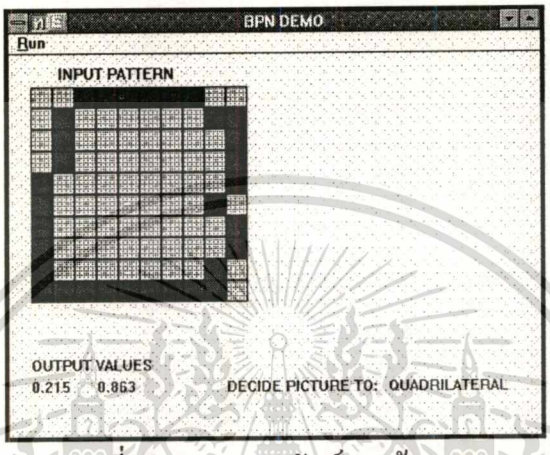
การคอมไพล์โปรแกรมเพื่อให้ทำงานภายใต้ระบบปฏิบัติการวินโดว์ต้องสร้างโปรเจคท์ไฟล์ ซึ่งแฟ้มนี้จะอินคลูด (include) แฟ้มโปรแกรมต่าง ๆ ที่จำเป็นต้องคอมไพล์ร่วมกัน แฟ้มเหล่านี้คือ

1. แฟ้ม SAMPLE2.CPP
2. แฟ้ม SAMPLE2.RC
3. แฟ้ม BPNLIB.LIB

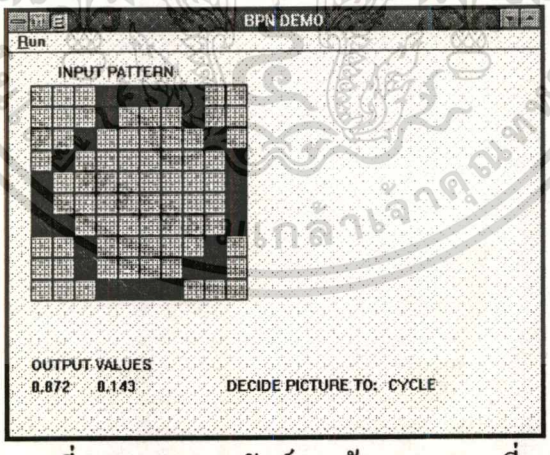
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และต้องกำหนดเมมโมรี่โมเดลให้เป็นลาร์จ (Large memory model) เมื่อคอมไพล์แล้วก็จะได้โปรแกรมทดสอบโครงข่ายประสาทเทียมที่สามารถทำงานได้บนระบบปฏิบัติการวินโดว์โดยจะอธิบายการทำงานต่อไป

เมื่อโปรแกรม SAMPLE2.EXE เริ่มต้นทำงานจะปรากฏภาพที่จอภาพดังนี้

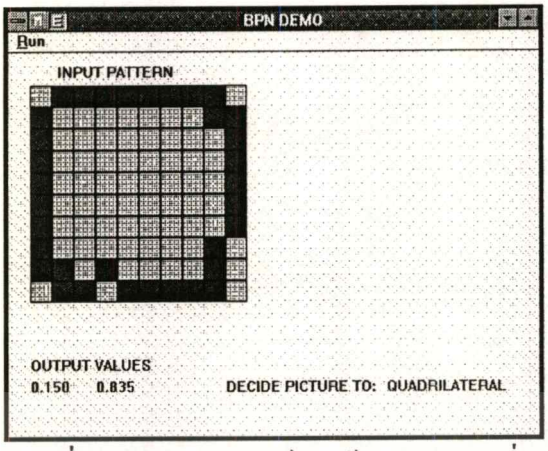


ภาพที่ 106 แสดงผลลัพธ์ของข้อมูลทดสอบที่ 1

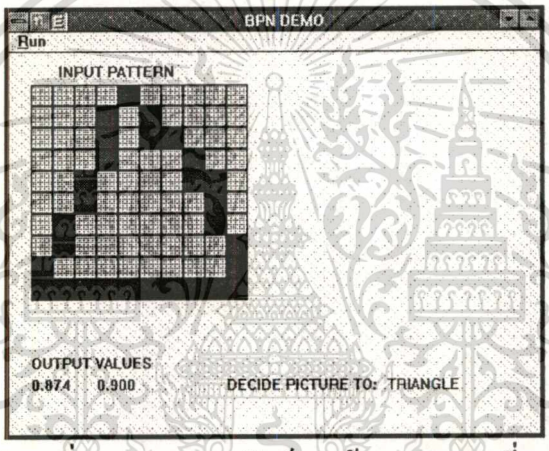


ภาพที่ 107 แสดงผลลัพธ์ของข้อมูลทดสอบที่ 2

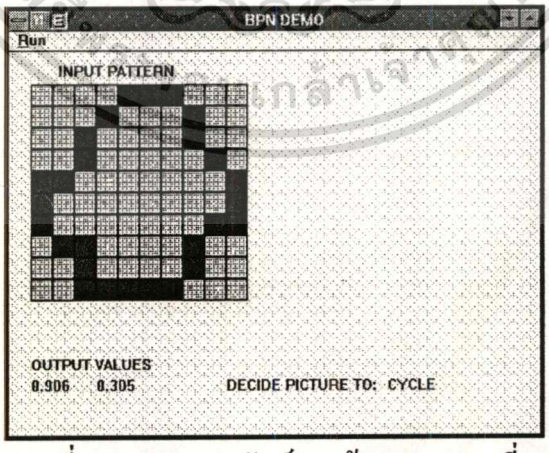
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 108 แสดงผลลัพธ์ของข้อมูลทดสอบที่ 3

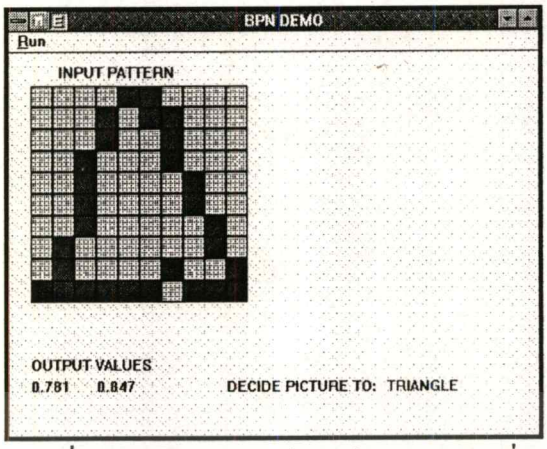


ภาพที่ 109 แสดงผลลัพธ์ของข้อมูลทดสอบที่ 4

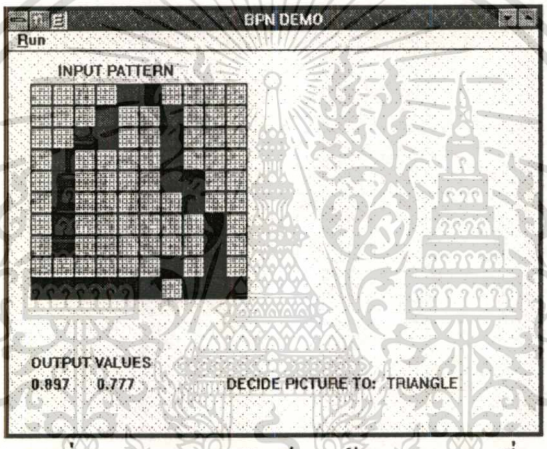


ภาพที่ 110 แสดงผลลัพธ์ของข้อมูลทดสอบที่ 5

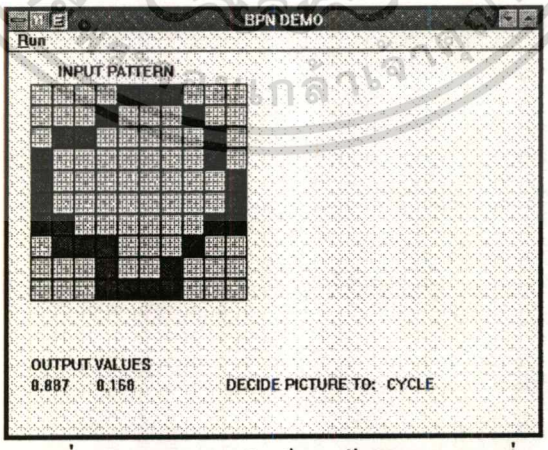
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 111 แสดงผลลัพธ์ของข้อมูลทดสอบที่ 6

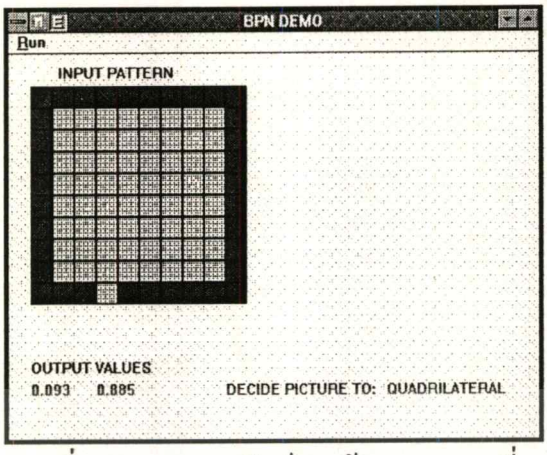


ภาพที่ 112 แสดงผลลัพธ์ของข้อมูลทดสอบที่ 7

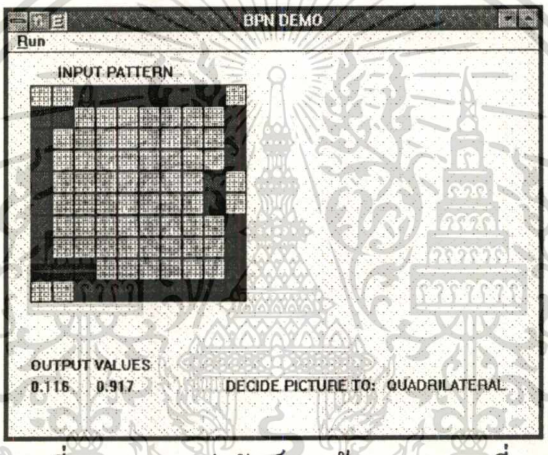


ภาพที่ 113 แสดงผลลัพธ์ของข้อมูลทดสอบที่ 8

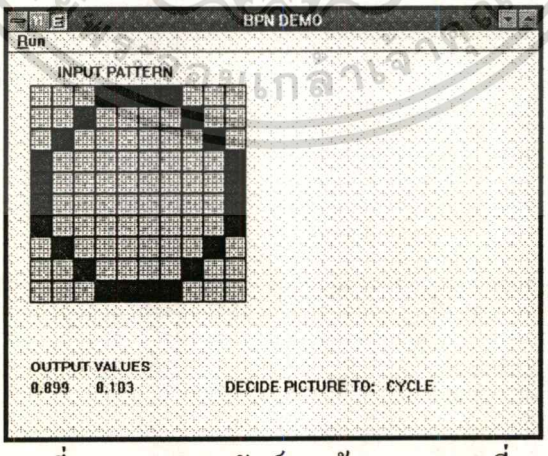
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 114 แสดงผลลัพธ์ของข้อมูลทดสอบที่ 9

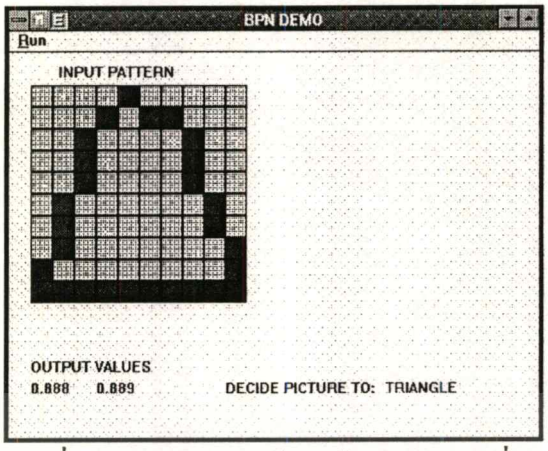


ภาพที่ 115 แสดงผลลัพธ์ของข้อมูลทดสอบที่ 10

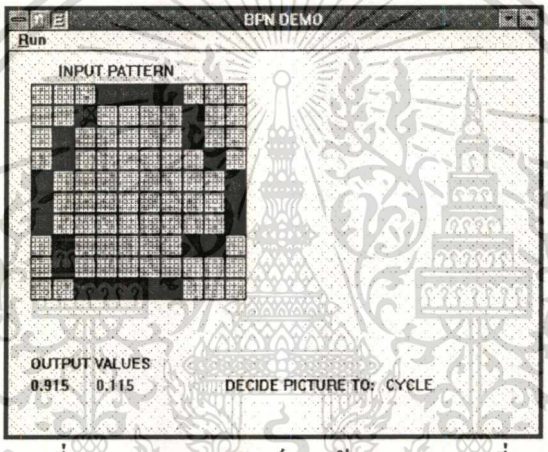


ภาพที่ 116 แสดงผลลัพธ์ของข้อมูลทดสอบที่ 11

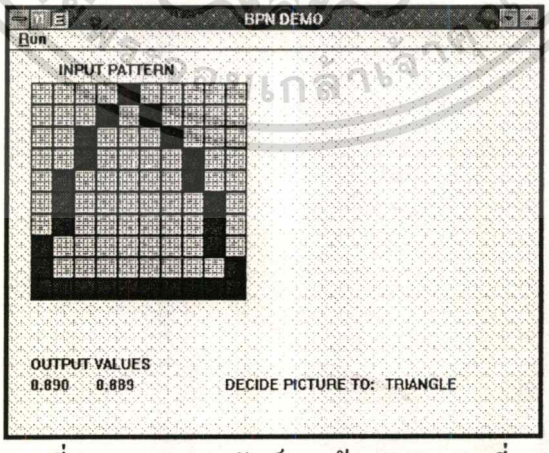
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 117 แสดงผลลัพธ์ของข้อมูลทดสอบที่ 12



ภาพที่ 118 แสดงผลลัพธ์ของข้อมูลทดสอบที่ 13



ภาพที่ 119 แสดงผลลัพธ์ของข้อมูลทดสอบที่ 14

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทสรุป

จากการทดลองที่ผ่านมาได้ทำการฝึกหัดโครงข่ายประสาทเทียมให้ทำการเรียนรู้รูปแบบอักขระ A ถึง Z และได้ทำการทดสอบโดยใช้กลุ่มข้อมูลทดสอบ A ถึง Z ที่ได้ผสมกับสัญญาณรบกวน ผลการทดลองคือโครงข่ายสามารถจำแนกข้อมูลได้ถูกต้องเป็นส่วนมากโดยให้ค่าเอาต์พุทเป็นรหัส ASCII การทดลองถัดมาคือนำโครงข่ายประสาทเทียมมาใช้กับการจดจำรูปทรงทางเรขาคณิตซึ่งมีรูปทรงสี่เหลี่ยม รูปทรงกลม และรูปทรงสามเหลี่ยม โดยการทดลองในบทนี้เป็นแค่เพียงตัวอย่างที่จะประยุกต์การใช้งานโปรแกรมจำลองโครงข่ายประสาทเทียมด้วยหลักการแพร่ย้อนกลับร่วมกับการเขียนโปรแกรมเรียกใช้ฟังก์ชั่นไลบรารี่ซึ่งเป็นตัวอย่างในการประยุกต์ใช้งานร่วมกับโครงงานอื่น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 6
สรุปผลการวิจัยและข้อเสนอแนะ

การวิจัยการจำลองโครงข่ายประสาทเทียมที่ได้ดำเนินการผ่านมานั้นปรากฏผลงานวิจัยในลักษณะของโปรแกรมจำลองโครงข่ายประสาทเทียมที่สามารถทำงานได้โดยใช้ไมโครคอมพิวเตอร์ โปรแกรมที่ได้สร้างขึ้นมานี้แบ่งได้เป็น 2 ประเภทคือ

1. โปรแกรมจำลองอะดาไลน์ ใช้จำลองการทำงานของนิวรอนเดี่ยว เมื่อผู้ใช้เรียกใช้โปรแกรมนี้จะทำให้ทราบถึงลักษณะการเรียนรู้ของนิวรอน ซึ่งโปรแกรมจะสร้างภาพกราฟิกติดต่อกับผู้ใช้ สิ่งที่แสดงทางจอภาพคือองค์ประกอบต่างๆของนิวรอน เช่น สัญญาณที่อินพุท น้ำหนักที่อินพุทที่กั้นสัญญาณอินพุทกับหน่วยประมวลผล และหน่วยประมวลผลที่ทำหน้าที่กำเนิดสัญญาณเอาต์พุทผู้ใช้จะเกิดความเข้าใจในกระบวนการเรียนรู้อย่างง่ายของศาสตร์โครงข่ายประสาทเทียมในระดับนิวรอน การนำอะดาไลน์ไปประยุกต์ใช้กับโครงงานอื่นๆนั้นมีข้อจำกัดมาก เพราะจะใช้ได้กับปัญหาเชิงเส้นเท่านั้น เนื่องจากจำนวนชั้นน้ำหนักเพียงชั้นเดียว

2. โปรแกรมจำลองโครงข่ายประสาทเทียมโดยอาศัยหลักการแพร่ย้อนกลับ ใช้จำลองการทำงานโครงข่ายประสาทเทียมแบบหลายชั้น สามารถนำไปประยุกต์ใช้กับโครงงานได้อย่างหลากหลายทำให้หลักการแพร่ย้อนกลับได้รับความนิยมมากในปัจจุบัน โปรแกรมที่ได้สร้างขึ้นมานี้นอกจากจะสร้างโปรแกรมจำลองโครงข่ายประสาทเทียมแล้วยังได้จัดสร้างโปรแกรมไลบรารี่บรรจุฟังก์ชั่นที่ผู้ใช้สามารถเขียนโปรแกรมเรียกฟังก์ชั่นเหล่านี้ได้ ทำให้ผู้ใช้กำหนดโครงสร้างตามลักษณะโครงงานได้ตามต้องการ ก่อให้เกิดประโยชน์ต่อผู้ใช้งานโปรแกรมจำลองโครงข่ายประสาทเทียมนี้พอสมควร

<u>ข้อเสนอแนะ</u>

การพัฒนาโปรแกรมจำลองโครงข่ายประสาทเทียม เพื่อเพิ่มประสิทธิภาพการทำงานนั้น ได้มีนักวิจัยระดับแนวหน้าในศาสตร์แขนงนี้หลายท่านได้เสนอแนะและแจกแจงไปเป็นข้อปลีกย่อยอีกหลายประการ ข้อเสนอแนะบางข้อนำมาใช้เพื่อลดช่วงเวลาในการเรียนรู้แต่ก็ขึ้นอยู่กับลักษณะของปัญหาหรือโครงงานที่จะนำโครงข่ายประสาทเทียมไปประยุกต์ใช้งานร่วมด้วย นั่นหมายความว่าหลักการดังกล่าวสามารถใช้งานได้ดีกับปัญหาบางประเภทเท่านั้น ไม่ใช่ครอบคลุมทั้งหมด เช่นกฎเกณฑ์ที่ว่าด้วยการใช้ NF (Noise Factor) ในกระบวนการปรีโปรเซสของกลุ่มสัญญาณทดสอบ อาจจะทำให้กระบวนการเรียนรู้ดีขึ้นกว่าเดิม หรือทฤษฎีที่กล่าวถึงการ

เอกสารนี้มีลิขสิทธิ์ ห้ามนำไปใช้ในการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เริ่มต้นค่าน้ำหนักให้แก่โครงข่ายที่สามารถทำให้โครงข่ายสามารถเรียนรู้บางปัญหาได้รวดเร็วยิ่งขึ้น และอื่นๆ ทฤษฎีเหล่านี้ไม่ใช่มาตรฐานในหลักการของโครงข่ายประสาทเทียม แต่เป็นเทคนิคที่นำมาใช้เพื่อเพิ่มประสิทธิภาพในหลักการที่เป็นมาตรฐาน แนวทางที่จะดำเนินงานวิจัยให้ต่อเนื่องจากงานวิจัยนี้อาจเป็นไปในรูปแบบของการประยุกต์ใช้โครงข่ายประสาทเทียมกับโครงงานอื่นๆ และสอดแทรกเทคนิคหรือคิดค้นเทคนิคใหม่ๆเพื่อให้การทำงานของโครงข่ายประสาทเทียมมีประสิทธิภาพดียิ่งขึ้น



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

เอกสารอ้างอิง

1 Richard K. Miller . Neural Networks : SEAI Technical Publications, 1987

2 Marilyn McCord Nelson & W.T. Illingwort . A Practical Guide to Neural Nets : Addison Wesley, 1990

3 Clidchanok Lursinsap . Artificial Neural Network : Semiconductor Electronic, October 1992

4 Bart Kosko . Neural Networks and Fuzzy System : Prentice-Hall International , 1992

5 Jame A. Freeman & David M. Skapura . Neural Networks, Algorithms, Applications, and Programming Techniques : Addison Wesley, 1991

6 Robert J. Schalkoff . Pattern Recognition | Statistical, Structural and Neural Approaches : John Wiley & Sons, 1992

7 Judith E. Dayhoff . Neural Network Architectures An Introduction : Van Nostrand Reinhold, 1990

8 Laurene Fausett . Fundamentals of Neural Networks Architectures, Algorithms, and Applications : Prentice Hall International, 1994

9 Clayton Walnum . Object-Oriented Programming with Borland C++ 4 : Que Corporation, 1994

10 Nabajyoti Bakakati . Borland C++ 4 Developer' s Guide : SAMS Publishing, 1994

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ภาคผนวก

แฟ้มข้อมูล CHAR.TRN

ภายในแฟ้มข้อมูลฝึกหัดมีข้อมูลดังนี้

0.1 0.1 0.9 0.1 0.1
0.1 0.9 0.1 0.9 0.1
0.9 0.1 0.1 0.1 0.9
0.9 0.1 0.1 0.1 0.9
0.9 0.9 0.9 0.9 0.9
0.9 0.1 0.1 0.1 0.9
0.9 0.1 0.1 0.1 0.9
0.1 0.9 0.1 0.1 0.1 0.1 0.1 0.9

0.9 0.9 0.9 0.9 0.1
0.9 0.1 0.1 0.1 0.9
0.9 0.1 0.1 0.1 0.9
0.9 0.9 0.9 0.9 0.1
0.9 0.1 0.1 0.1 0.9
0.9 0.1 0.1 0.1 0.9
0.9 0.9 0.9 0.9 0.1
0.1 0.9 0.1 0.1 0.1 0.1 0.9 0.1

0.1 0.9 0.9 0.9 0.1
0.9 0.1 0.1 0.1 0.9
0.9 0.1 0.1 0.1 0.1
0.9 0.1 0.1 0.1 0.1
0.9 0.1 0.1 0.1 0.1
0.9 0.1 0.1 0.1 0.9
0.1 0.9 0.9 0.9 0.1
0.1 0.9 0.1 0.1 0.1 0.1 0.9 0.9

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

0.9	0.9	0.9	0.9	0.1			
0.9	0.1	0.1	0.1	0.9			
0.9	0.1	0.1	0.1	0.9			
0.9	0.1	0.1	0.1	0.9			
0.9	0.1	0.1	0.1	0.9			
0.9	0.1	0.1	0.1	0.9			
0.9	0.9	0.9	0.9	0.1			
0.1	0.9	0.1	0.1	0.1	0.9	0.1	0.1
0.9	0.9	0.9	0.9	0.9			
0.9	0.1	0.1	0.1	0.1			
0.9	0.1	0.1	0.1	0.1			
0.9	0.9	0.9	0.9	0.9			
0.9	0.1	0.1	0.1	0.1			
0.9	0.1	0.1	0.1	0.1			
0.9	0.9	0.9	0.9	0.9			
0.1	0.9	0.1	0.1	0.1	0.9	0.1	0.9
0.9	0.9	0.9	0.9	0.9			
0.9	0.1	0.1	0.1	0.1			
0.9	0.1	0.1	0.1	0.1			
0.9	0.9	0.9	0.9	0.1			
0.9	0.1	0.1	0.1	0.1			
0.9	0.1	0.1	0.1	0.1			
0.9	0.1	0.1	0.1	0.1			
0.1	0.9	0.1	0.1	0.1	0.9	0.9	0.1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

0.1	0.9	0.9	0.9	0.1			
0.9	0.1	0.1	0.1	0.9			
0.9	0.1	0.1	0.1	0.1			
0.9	0.1	0.1	0.9	0.9			
0.9	0.1	0.1	0.1	0.9			
0.9	0.1	0.1	0.1	0.9			
0.1	0.9	0.9	0.9	0.1			
0.1	0.9	0.1	0.1	0.1	0.9	0.9	0.9
0.9	0.1	0.1	0.1	0.9			
0.9	0.1	0.1	0.1	0.9			
0.9	0.1	0.1	0.1	0.9			
0.9	0.9	0.9	0.9	0.9			
0.9	0.1	0.1	0.1	0.9			
0.9	0.1	0.1	0.1	0.9			
0.9	0.1	0.1	0.1	0.9			
0.1	0.9	0.1	0.1	0.9	0.1	0.1	0.1
0.1	0.9	0.9	0.9	0.1			
0.1	0.1	0.9	0.1	0.1			
0.1	0.1	0.9	0.1	0.1			
0.1	0.1	0.9	0.1	0.1			
0.1	0.1	0.9	0.1	0.1			
0.1	0.1	0.9	0.1	0.1			
0.1	0.9	0.9	0.9	0.1			
0.1	0.9	0.1	0.1	0.9	0.1	0.1	0.9

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

0.1	0.1	0.1	0.9	0.9			
0.1	0.1	0.1	0.1	0.9			
0.1	0.1	0.1	0.1	0.9			
0.1	0.1	0.1	0.1	0.9			
0.9	0.1	0.1	0.1	0.9			
0.9	0.1	0.1	0.1	0.9			
0.1	0.9	0.9	0.9	0.1			
0.1	0.9	0.1	0.1	0.9	0.1	0.9	0.1
0.9	0.1	0.1	0.1	0.9			
0.9	0.1	0.1	0.9	0.1			
0.9	0.1	0.9	0.1	0.1			
0.9	0.9	0.1	0.1	0.1			
0.9	0.1	0.9	0.1	0.1			
0.9	0.1	0.1	0.9	0.1			
0.9	0.1	0.1	0.1	0.9			
0.1	0.9	0.1	0.1	0.9	0.1	0.9	0.9
0.9	0.1	0.1	0.1	0.1			
0.9	0.1	0.1	0.1	0.1			
0.9	0.1	0.1	0.1	0.1			
0.9	0.1	0.1	0.1	0.1			
0.9	0.1	0.1	0.1	0.1			
0.9	0.1	0.1	0.1	0.1			
0.9	0.9	0.9	0.9	0.9			
0.1	0.9	0.1	0.1	0.9	0.9	0.1	0.1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

0.9	0.1	0.1	0.1	0.9			
0.9	0.9	0.1	0.9	0.9			
0.9	0.1	0.9	0.1	0.9			
0.9	0.1	0.9	0.1	0.9			
0.9	0.1	0.1	0.1	0.9			
0.9	0.1	0.1	0.1	0.9			
0.9	0.1	0.1	0.1	0.9			
0.1	0.9	0.1	0.1	0.9	0.9	0.1	0.9
0.9	0.1	0.1	0.1	0.9			
0.9	0.9	0.1	0.1	0.9			
0.9	0.1	0.9	0.1	0.9			
0.9	0.1	0.9	0.1	0.9			
0.9	0.1	0.9	0.1	0.9			
0.9	0.1	0.1	0.9	0.9			
0.9	0.1	0.1	0.1	0.9			
0.1	0.9	0.1	0.1	0.9	0.9	0.9	0.1
0.1	0.9	0.9	0.9	0.1			
0.9	0.1	0.1	0.1	0.9			
0.9	0.1	0.1	0.1	0.9			
0.9	0.1	0.1	0.1	0.9			
0.9	0.1	0.1	0.1	0.9			
0.9	0.1	0.1	0.1	0.9			
0.1	0.9	0.9	0.9	0.1			
0.1	0.9	0.1	0.1	0.9	0.9	0.9	0.9

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

0.9	0.9	0.9	0.9	0.1			
0.9	0.1	0.1	0.1	0.9			
0.9	0.1	0.1	0.1	0.9			
0.9	0.9	0.9	0.9	0.1			
0.9	0.1	0.1	0.1	0.1			
0.9	0.1	0.1	0.1	0.1			
0.9	0.1	0.1	0.1	0.1			
0.1	0.9	0.1	0.9	0.1	0.1	0.1	0.1
0.1	0.9	0.9	0.9	0.1			
0.9	0.1	0.1	0.1	0.9			
0.9	0.1	0.1	0.1	0.9			
0.9	0.1	0.1	0.1	0.9			
0.9	0.1	0.9	0.1	0.9			
0.9	0.1	0.1	0.9	0.1			
0.1	0.9	0.9	0.1	0.9			
0.1	0.9	0.1	0.9	0.1	0.1	0.1	0.9
0.9	0.9	0.9	0.9	0.1			
0.9	0.1	0.1	0.1	0.9			
0.9	0.1	0.1	0.1	0.9			
0.9	0.9	0.9	0.9	0.1			
0.9	0.1	0.1	0.9	0.1			
0.9	0.1	0.1	0.1	0.9			
0.9	0.1	0.1	0.1	0.9			
0.1	0.9	0.1	0.9	0.1	0.1	0.9	0.1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

0.1	0.9	0.9	0.9	0.1			
0.9	0.1	0.1	0.1	0.9			
0.9	0.1	0.1	0.1	0.1			
0.1	0.9	0.9	0.9	0.1			
0.1	0.1	0.1	0.1	0.9			
0.9	0.1	0.1	0.1	0.9			
0.1	0.9	0.9	0.9	0.1			
0.1	0.9	0.1	0.9	0.1	0.1	0.9	0.9
0.9	0.9	0.9	0.9	0.9			
0.1	0.1	0.9	0.1	0.1			
0.1	0.1	0.9	0.1	0.1			
0.1	0.1	0.9	0.1	0.1			
0.1	0.1	0.9	0.1	0.1			
0.1	0.1	0.9	0.1	0.1			
0.1	0.1	0.9	0.1	0.1			
0.1	0.9	0.1	0.9	0.1	0.9	0.1	0.1
0.9	0.1	0.1	0.1	0.9			
0.9	0.1	0.1	0.1	0.9			
0.9	0.1	0.1	0.1	0.9			
0.9	0.1	0.1	0.1	0.9			
0.9	0.1	0.1	0.1	0.9			
0.9	0.1	0.1	0.1	0.9			
0.1	0.9	0.9	0.9	0.1			
0.1	0.9	0.1	0.9	0.1	0.9	0.1	0.9

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

0.9 0.1 0.1 0.1 0.9

0.9 0.1 0.1 0.1 0.9

0.9 0.1 0.1 0.1 0.9

0.9 0.1 0.1 0.1 0.9

0.9 0.1 0.1 0.1 0.9

0.1 0.9 0.1 0.9 0.1

0.1 0.1 0.9 0.1 0.1

0.1 0.9 0.1 0.9 0.1 0.9 0.9 0.1

0.9 0.1 0.1 0.1 0.9

0.9 0.1 0.1 0.1 0.9

0.9 0.1 0.1 0.1 0.9

0.9 0.1 0.9 0.1 0.9

0.9 0.1 0.9 0.1 0.9

0.9 0.9 0.1 0.9 0.9

0.9 0.1 0.1 0.1 0.9

0.1 0.9 0.1 0.9 0.1 0.9 0.9 0.9

0.9 0.1 0.1 0.1 0.9

0.9 0.1 0.1 0.1 0.9

0.1 0.9 0.1 0.9 0.1

0.1 0.1 0.9 0.1 0.1

0.1 0.9 0.1 0.9 0.1

0.9 0.1 0.1 0.1 0.9

0.9 0.1 0.1 0.1 0.9

0.1 0.9 0.1 0.9 0.9 0.1 0.1 0.1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

0.9	0.1	0.1	0.1	0.9			
0.9	0.1	0.1	0.1	0.9			
0.1	0.9	0.1	0.9	0.1			
0.1	0.1	0.9	0.1	0.1			
0.1	0.1	0.9	0.1	0.1			
0.1	0.1	0.9	0.1	0.1			
0.1	0.1	0.9	0.1	0.1			
0.1	0.9	0.1	0.9	0.9	0.1	0.1	0.9

0.9	0.9	0.9	0.9	0.9			
0.1	0.1	0.1	0.1	0.9			
0.1	0.1	0.1	0.9	0.1			
0.1	0.1	0.9	0.1	0.1			
0.1	0.9	0.1	0.1	0.1			
0.9	0.1	0.1	0.1	0.1			
0.9	0.9	0.9	0.9	0.9			
0.1	0.9	0.1	0.9	0.9	0.1	0.9	0.1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แฟ้มข้อมูล CHAR.TES

ภายในแฟ้มข้อมูลทดสอบมีข้อมูลดังนี้

0.1	0.1	0.9	0.1	0.1
0.1	0.9	0.1	0.9	0.1
0.9	0.1	0.1	0.1	0.9
0.9	0.1	0.1	0.1	0.9
0.9	0.9	0.9	0.9	0.9
0.9	0.1	0.9	0.1	0.9
0.9	0.1	0.1	0.1	0.9
0.9	0.9	0.9	0.9	0.1
0.9	0.1	0.9	0.1	0.9
0.9	0.1	0.1	0.1	0.9
0.9	0.9	0.9	0.9	0.1
0.9	0.1	0.1	0.1	0.9
0.9	0.9	0.1	0.1	0.9
0.9	0.9	0.9	0.9	0.1
0.1	0.9	0.9	0.9	0.1
0.9	0.1	0.1	0.1	0.9
0.9	0.1	0.1	0.1	0.1
0.9	0.9	0.1	0.1	0.1
0.9	0.1	0.1	0.1	0.1
0.9	0.1	0.1	0.1	0.9
0.1	0.9	0.9	0.9	0.1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

0.9	0.9	0.9	0.9	0.1
0.9	0.1	0.1	0.9	0.9
0.9	0.1	0.1	0.1	0.9
0.9	0.1	0.1	0.1	0.9
0.9	0.1	0.1	0.1	0.9
0.9	0.9	0.1	0.1	0.9
0.9	0.9	0.9	0.9	0.1
0.9	0.9	0.9	0.9	0.9
0.9	0.1	0.1	0.1	0.9
0.9	0.1	0.1	0.1	0.1
0.9	0.9	0.9	0.9	0.9
0.9	0.1	0.1	0.1	0.1
0.9	0.9	0.1	0.1	0.1
0.9	0.9	0.9	0.9	0.9
0.9	0.9	0.9	0.9	0.9
0.9	0.1	0.1	0.1	0.1
0.9	0.1	0.9	0.1	0.1
0.9	0.9	0.9	0.9	0.1
0.9	0.1	0.1	0.1	0.1
0.9	0.1	0.1	0.1	0.1
0.9	0.1	0.9	0.1	0.1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

0.1	0.9	0.9	0.9	0.1
0.9	0.1	0.1	0.1	0.9
0.9	0.1	0.1	0.1	0.1
0.9	0.1	0.1	0.9	0.9
0.9	0.9	0.1	0.1	0.9
0.9	0.1	0.1	0.1	0.9
0.1	0.9	0.9	0.9	0.1

0.9	0.1	0.1	0.1	0.9
0.9	0.1	0.1	0.1	0.9
0.9	0.1	0.1	0.1	0.9
0.9	0.9	0.9	0.9	0.9
0.9	0.9	0.1	0.1	0.9
0.9	0.1	0.1	0.1	0.9
0.9	0.1	0.9	0.1	0.9

0.1	0.9	0.9	0.9	0.1
0.1	0.1	0.9	0.1	0.1
0.1	0.1	0.9	0.1	0.1
0.1	0.1	0.9	0.1	0.1
0.1	0.1	0.9	0.1	0.9
0.9	0.1	0.9	0.1	0.1
0.1	0.9	0.9	0.9	0.1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

0.1	0.1	0.1	0.9	0.9
0.1	0.1	0.1	0.1	0.9
0.1	0.1	0.1	0.1	0.9
0.1	0.9	0.1	0.1	0.9
0.9	0.1	0.1	0.1	0.9
0.9	0.1	0.1	0.1	0.9
0.1	0.9	0.9	0.9	0.1
0.9	0.1	0.1	0.1	0.9
0.9	0.1	0.1	0.9	0.1
0.9	0.1	0.9	0.1	0.1
0.9	0.9	0.1	0.1	0.1
0.9	0.1	0.9	0.1	0.1
0.9	0.9	0.1	0.9	0.1
0.9	0.1	0.1	0.1	0.9
0.9	0.1	0.1	0.1	0.1
0.9	0.1	0.1	0.1	0.1
0.9	0.1	0.1	0.1	0.1
0.9	0.1	0.1	0.1	0.1
0.9	0.1	0.1	0.1	0.1
0.9	0.9	0.1	0.1	0.1
0.9	0.9	0.9	0.9	0.9

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

0.9	0.1	0.9	0.1	0.9
0.9	0.9	0.1	0.9	0.9
0.9	0.1	0.9	0.1	0.9
0.9	0.1	0.9	0.1	0.9
0.9	0.1	0.1	0.1	0.9
0.9	0.1	0.1	0.1	0.9
0.9	0.9	0.1	0.1	0.9
0.9	0.1	0.9	0.1	0.9
0.9	0.9	0.1	0.1	0.9
0.9	0.1	0.9	0.1	0.9
0.9	0.1	0.9	0.1	0.9
0.9	0.1	0.9	0.1	0.9
0.9	0.1	0.1	0.9	0.9
0.9	0.1	0.1	0.1	0.9
0.1	0.9	0.9	0.9	0.1
0.9	0.1	0.1	0.1	0.9
0.9	0.1	0.1	0.1	0.9
0.9	0.1	0.9	0.1	0.9
0.9	0.1	0.1	0.1	0.9
0.9	0.9	0.1	0.1	0.9
0.1	0.9	0.9	0.9	0.1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

0.9	0.9	0.9	0.9	0.1
0.9	0.1	0.1	0.1	0.9
0.9	0.1	0.1	0.1	0.9
0.9	0.9	0.9	0.9	0.1
0.9	0.1	0.1	0.1	0.1
0.9	0.1	0.1	0.1	0.1
0.9	0.1	0.9	0.1	0.1
0.1	0.9	0.9	0.9	0.1
0.9	0.1	0.1	0.1	0.9
0.9	0.1	0.9	0.1	0.9
0.9	0.1	0.1	0.1	0.9
0.9	0.1	0.9	0.1	0.9
0.9	0.1	0.1	0.9	0.1
0.1	0.9	0.9	0.1	0.9
0.9	0.9	0.9	0.9	0.1
0.9	0.1	0.1	0.1	0.9
0.9	0.1	0.1	0.1	0.9
0.9	0.9	0.9	0.9	0.1
0.9	0.1	0.1	0.9	0.1
0.9	0.1	0.9	0.1	0.9
0.9	0.1	0.1	0.1	0.9

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

0.1	0.9	0.9	0.9	0.1
0.9	0.1	0.1	0.1	0.9
0.9	0.9	0.1	0.1	0.1
0.1	0.9	0.9	0.9	0.1
0.1	0.1	0.1	0.1	0.9
0.9	0.1	0.1	0.1	0.9
0.9	0.9	0.9	0.9	0.1

0.9	0.9	0.9	0.9	0.9
0.1	0.1	0.9	0.1	0.1
0.1	0.1	0.9	0.1	0.1
0.9	0.1	0.9	0.1	0.1
0.1	0.1	0.9	0.1	0.1
0.1	0.1	0.9	0.1	0.1
0.1	0.1	0.9	0.1	0.1

0.9	0.1	0.1	0.1	0.9
0.9	0.1	0.1	0.1	0.9
0.9	0.9	0.1	0.1	0.9
0.9	0.1	0.1	0.1	0.9
0.9	0.1	0.1	0.1	0.9
0.9	0.1	0.1	0.1	0.9
0.1	0.9	0.9	0.9	0.1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

0.9	0.1	0.1	0.1	0.9
0.9	0.1	0.1	0.1	0.9
0.9	0.9	0.1	0.1	0.9
0.9	0.1	0.1	0.1	0.9
0.9	0.1	0.1	0.1	0.9
0.1	0.9	0.1	0.9	0.1
0.1	0.1	0.9	0.1	0.1
0.9	0.1	0.1	0.1	0.9
0.9	0.1	0.1	0.9	0.9
0.9	0.1	0.1	0.1	0.9
0.9	0.1	0.9	0.1	0.9
0.9	0.1	0.9	0.1	0.9
0.9	0.9	0.1	0.9	0.9
0.9	0.1	0.1	0.1	0.9
0.9	0.1	0.1	0.9	0.9
0.9	0.1	0.1	0.1	0.9
0.1	0.9	0.1	0.9	0.1
0.1	0.1	0.9	0.1	0.1
0.1	0.9	0.1	0.9	0.1
0.9	0.1	0.1	0.1	0.9
0.9	0.1	0.1	0.1	0.9

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

0.9	0.1	0.1	0.1	0.9
0.9	0.1	0.1	0.1	0.9
0.1	0.9	0.1	0.9	0.1
0.1	0.1	0.9	0.1	0.1
0.1	0.1	0.9	0.1	0.1
0.9	0.1	0.9	0.1	0.1
0.1	0.1	0.9	0.1	0.1

0.9	0.9	0.9	0.9	0.9
0.1	0.1	0.1	0.1	0.9
0.1	0.1	0.1	0.9	0.1
0.9	0.1	0.9	0.1	0.1
0.1	0.9	0.1	0.1	0.1
0.9	0.1	0.1	0.1	0.1
0.9	0.9	0.9	0.9	0.9

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แฟ้มข้อมูล GEOME.TRN

ภายในแฟ้มข้อมูลฝึกหัดมีข้อมูลดังนี้

.9 .9 .9 .9 .9 .9 .9 .9 .9 .9

.9 .1 .1 .1 .1 .1 .1 .1 .1 .9

.9 .1 .1 .1 .1 .1 .1 .1 .1 .9

.9 .1 .1 .1 .1 .1 .1 .1 .1 .9

.9 .1 .1 .1 .1 .1 .1 .1 .1 .9

.9 .1 .1 .1 .1 .1 .1 .1 .1 .9

.9 .1 .1 .1 .1 .1 .1 .1 .1 .9

.9 .1 .1 .1 .1 .1 .1 .1 .1 .9

.9 .1 .1 .1 .1 .1 .1 .1 .1 .9

.9 .9 .9 .9 .9 .9 .9 .9 .9 .9

.1 .9

.1 .1 .9 .9 .9 .9 .9 .9 .9 .1

.9 .9 .1 .1 .1 .1 .1 .1 .1 .9

.9 .1 .1 .1 .1 .1 .1 .1 .1 .9

.9 .1 .1 .1 .1 .1 .1 .1 .1 .9

.9 .1 .1 .1 .1 .1 .1 .1 .9 .1

.9 .1 .1 .1 .1 .1 .1 .1 .9 .1

.9 .1 .1 .1 .1 .1 .1 .1 .1 .9

.9 .1 .1 .1 .1 .1 .1 .1 .1 .9

.9 .9 .9 .1 .1 .1 .1 .1 .1 .9

.1 .1 .9 .9 .9 .9 .9 .9 .9 .9

.1 .9

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

.1 .1 .1 .9 .9 .9 .9 .1 .1 .1

.1 .1 .9 .1 .1 .1 .1 .9 .1 .1

.1 .9 .1 .1 .1 .1 .1 .1 .9 .1

.9 .1 .1 .1 .1 .1 .1 .1 .1 .9

.9 .1 .1 .1 .1 .1 .1 .1 .1 .9

.9 .1 .1 .1 .1 .1 .1 .1 .1 .9

.9 .1 .1 .1 .1 .1 .1 .1 .1 .9

.1 .9 .1 .1 .1 .1 .1 .1 .9 .1

.1 .1 .9 .1 .1 .1 .1 .9 .1 .1

.1 .1 .1 .9 .9 .9 .9 .1 .1 .1

.9 .1

.1 .1 .1 .9 .9 .9 .9 .1 .1 .1

.1 .1 .9 .1 .1 .1 .1 .9 .1 .1

.1 .9 .1 .1 .1 .1 .1 .9 .1 .1

.1 .9 .1 .1 .1 .1 .1 .1 .9 .1

.9 .1 .1 .1 .1 .1 .1 .1 .1 .9

.9 .1 .1 .1 .1 .1 .1 .1 .1 .9

.9 .1 .1 .1 .1 .1 .1 .1 .1 .9

.1 .9 .1 .1 .1 .1 .1 .9 .9 .1

.1 .9 .1 .1 .1 .1 .1 .9 .1 .1

.1 .1 .9 .9 .9 .9 .9 .1 .1 .1

.9 .1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

.1 .1 .1 .1 .9 .1 .1 .1 .1 .1

.1 .1 .1 .9 .1 .9 .1 .1 .1 .1

.1 .1 .9 .1 .1 .1 .9 .1 .1 .1

.1 .1 .9 .1 .1 .1 .1 .9 .1 .1

.1 .9 .1 .1 .1 .1 .1 .9 .1 .1

.1 .9 .1 .1 .1 .1 .1 .1 .9 .1

.1 .9 .1 .1 .1 .1 .1 .1 .9 .1

.9 .1 .1 .1 .1 .1 .1 .1 .9 .1

.9 .1 .1 .1 .1 .1 .1 .1 .1 .9

.9 .9 .9 .9 .9 .9 .9 .9 .9 .9

.9 .9

.1 .1 .1 .1 .9 .1 .1 .1 .1 .1

.1 .1 .1 .9 .1 .9 .9 .1 .1 .1

.1 .1 .9 .1 .1 .1 .1 .9 .1 .1

.1 .1 .9 .1 .1 .1 .1 .9 .1 .1

.1 .1 .9 .1 .1 .1 .1 .9 .1 .1

.1 .9 .1 .1 .1 .1 .1 .1 .9 .1

.1 .9 .1 .1 .1 .1 .1 .1 .9 .1

.1 .9 .1 .1 .1 .1 .1 .1 .1 .9

.9 .1 .1 .1 .1 .1 .1 .1 .1 .9

.9 .9 .9 .9 .9 .9 .9 .9 .9 .9

.9 .9

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แฟ้มข้อมูล GEOME.TES

ภายในแฟ้มข้อมูลทดสอบมีข้อมูลดังนี้

.1 .1 .9 .9 .9 .9 .9 .9 .1 .1

.1 .9 .1 .1 .1 .1 .1 .1 .9 .9

.1 .9 .1 .1 .1 .1 .1 .1 .1 .9

.1 .9 .1 .1 .1 .1 .1 .1 .1 .9

.9 .1 .1 .1 .1 .1 .1 .1 .1 .9

.9 .1 .1 .1 .1 .1 .1 .1 .9 .1

.9 .1 .1 .1 .1 .1 .1 .1 .1 .9

.9 .1 .1 .1 .1 .1 .1 .1 .1 .9

.9 .1 .1 .1 .1 .1 .1 .1 .9 .1

.9 .9 .9 .9 .9 .9 .9 .9 .9 .1

.1 .1 .1 .9 .9 .9 .9 .9 .1 .1

.1 .1 .1 .9 .1 .1 .1 .9 .1 .1

.1 .1 .9 .1 .1 .1 .1 .1 .9 .1

.1 .9 .1 .1 .1 .1 .1 .1 .1 .9

.9 .1 .1 .1 .1 .1 .1 .1 .1 .9

.9 .1 .1 .1 .1 .1 .1 .1 .1 .9

.9 .9 .1 .1 .1 .1 .1 .1 .1 .9

.1 .1 .9 .1 .1 .1 .1 .1 .9 .1

.1 .1 .9 .1 .1 .1 .1 .9 .9 .1

.1 .1 .1 .9 .9 .9 .9 .1 .1 .1

.1 .9 .9 .9 .9 .9 .9 .9 .9 .1

.9 .1 .1 .1 .1 .1 .1 .1 .9 .9

.9 .1 .1 .1 .1 .1 .1 .1 .1 .9

.9 .1 .1 .1 .1 .1 .1 .1 .1 .9

.9 .1 .1 .1 .1 .1 .1 .1 .1 .9

.9 .1 .1 .1 .1 .1 .1 .1 .1 .9

.9 .1 .1 .1 .1 .1 .1 .1 .1 .9

.9 .1 .1 .1 .1 .1 .1 .1 .1 .9

.9 .1 .1 .1 .1 .1 .1 .1 .9 .1

.9 .9 .1 .9 .1 .1 .1 .1 .9 .1

.1 .9 .9 .1 .9 .9 .9 .9 .9 .1

เอกสารนี้มีลิขสิทธิ์ สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

.1 .1 .1 .1 .9 .1 .1 .1 .1 .1

.1 .1 .1 .9 .1 .9 .1 .1 .1 .1

.1 .1 .1 .9 .1 .9 .9 .1 .1 .1

.1 .1 .1 .9 .1 .1 .1 .9 .1 .1

.1 .1 .9 .1 .1 .1 .1 .9 .9 .1

.1 .1 .9 .1 .1 .1 .1 .1 .9 .1

.1 .9 .1 .1 .1 .1 .1 .1 .9 .1

.1 .9 .1 .1 .1 .1 .1 .1 .1 .9

.9 .1 .1 .1 .1 .1 .1 .1 .1 .9

.9 .9 .9 .9 .9 .9 .9 .9 .9 .9

.1 .1 .1 .1 .9 .9 .9 .1 .1 .1

.1 .1 .1 .9 .1 .1 .1 .9 .1 .1

.1 .1 .9 .1 .1 .1 .1 .9 .1 .1

.1 .1 .9 .1 .1 .1 .1 .1 .9 .1

.9 .9 .1 .1 .1 .1 .1 .1 .1 .9

.9 .1 .1 .1 .1 .1 .1 .1 .1 .9

.9 .1 .1 .1 .1 .1 .1 .1 .9 .9

.1 .9 .9 .1 .1 .1 .1 .9 .1 .1

.1 .1 .9 .1 .1 .1 .1 .9 .1 .1

.1 .1 .9 .9 .9 .9 .9 .1 .1 .1

.1 .1 .1 .1 .9 .9 .1 .1 .1 .1

.1 .1 .1 .9 .1 .9 .9 .1 .1 .1

.1 .1 .1 .9 .1 .1 .9 .1 .1 .1

.1 .1 .9 .1 .1 .1 .9 .1 .1 .1

.1 .1 .9 .1 .1 .1 .1 .9 .1 .1

.1 .1 .9 .1 .1 .1 .1 .9 .1 .1

.1 .1 .9 .1 .1 .1 .1 .1 .9 .1

.1 .9 .1 .1 .1 .1 .1 .1 .9 .1

.1 .9 .1 .1 .1 .1 .9 .1 .1 .9

.9 .9 .9 .9 .9 .9 .1 .9 .9 .9

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

.1 .1 .1 .1 .9 .9 .1 .1 .1 .1

.1 .1 .1 .9 .1 .1 .9 .1 .1 .1

.1 .1 .9 .9 .1 .1 .9 .1 .1 .1

.1 .9 .1 .1 .1 .1 .9 .1 .1 .1

.1 .9 .1 .1 .1 .1 .9 .9 .1 .1

.1 .9 .1 .1 .1 .1 .1 .9 .1 .1

.1 .9 .1 .1 .1 .1 .1 .1 .9 .1

.1 .9 .1 .1 .1 .1 .1 .1 .9 .1

.1 .1 .1 .1 .1 .1 .9 .1 .9 .1

.9 .9 .9 .9 .9 .9 .1 .9 .9 .9

.1 .1 .1 .1 .9 .9 .9 .1 .1 .1

.1 .1 .1 .9 .1 .1 .1 .9 .1 .1

.1 .9 .9 .1 .1 .1 .1 .1 .9 .1

.9 .1 .1 .1 .1 .1 .1 .1 .9 .1

.9 .1 .1 .1 .1 .1 .1 .1 .1 .9

.9 .1 .1 .1 .1 .1 .1 .1 .1 .9

.9 .9 .1 .1 .1 .1 .1 .1 .9 .9

.1 .9 .9 .9 .1 .1 .1 .9 .1 .1

.1 .1 .1 .9 .1 .1 .9 .1 .1 .1

.1 .1 .1 .9 .9 .9 .9 .1 .1 .1

.9 .9 .9 .9 .9 .9 .9 .9 .9 .9

.9 .1 .1 .1 .1 .1 .1 .1 .1 .9

.9 .1 .1 .1 .1 .1 .1 .1 .1 .9

.9 .1 .1 .1 .1 .1 .1 .1 .1 .9

.9 .1 .1 .1 .1 .1 .1 .1 .1 .9

.9 .1 .1 .1 .1 .1 .1 .1 .1 .9

.9 .1 .1 .1 .1 .1 .1 .1 .1 .9

.9 .1 .1 .1 .1 .1 .1 .1 .1 .9

.9 .1 .1 .1 .1 .1 .1 .1 .1 .9

.9 .9 .9 .1 .9 .9 .9 .9 .9 .9

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

.1 .1 .9 .9 .9 .9 .9 .9 .9 .1

.9 .9 .1 .1 .1 .1 .1 .1 .1 .9

.9 .1 .1 .1 .1 .1 .1 .1 .1 .9

.9 .1 .1 .1 .1 .1 .1 .1 .1 .9

.9 .1 .1 .1 .1 .1 .1 .1 .9 .1

.9 .1 .1 .1 .1 .1 .1 .1 .9 .1

.9 .1 .1 .1 .1 .1 .1 .1 .1 .9

.9 .1 .1 .1 .1 .1 .1 .1 .1 .9

.9 .9 .9 .1 .1 .1 .1 .1 .1 .9

.1 .1 .9 .9 .9 .9 .9 .9 .9 .9

.1 .1 .1 .9 .9 .9 .9 .1 .1 .1

.1 .1 .9 .1 .1 .1 .1 .9 .1 .1

.1 .9 .1 .1 .1 .1 .1 .1 .9 .1

.9 .1 .1 .1 .1 .1 .1 .1 .1 .9

.9 .1 .1 .1 .1 .1 .1 .1 .1 .9

.9 .1 .1 .1 .1 .1 .1 .1 .1 .9

.9 .1 .1 .1 .1 .1 .1 .1 .1 .9

.1 .9 .1 .1 .1 .1 .1 .1 .9 .1

.1 .1 .9 .1 .1 .1 .1 .9 .1 .1

.1 .1 .1 .9 .9 .9 .9 .1 .1 .1

.1 .1 .1 .1 .9 .1 .1 .1 .1 .1

.1 .1 .1 .9 .1 .9 .9 .1 .1 .1

.1 .1 .9 .1 .1 .1 .1 .9 .1 .1

.1 .1 .9 .1 .1 .1 .1 .9 .1 .1

.1 .1 .9 .1 .1 .1 .1 .9 .1 .1

.1 .9 .1 .1 .1 .1 .1 .1 .9 .1

.1 .9 .1 .1 .1 .1 .1 .1 .9 .1

.1 .9 .1 .1 .1 .1 .1 .1 .1 .9

.9 .1 .1 .1 .1 .1 .1 .1 .1 .9

.9 .9 .9 .9 .9 .9 .9 .9 .9 .9

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

.1 .1 .1 .9 .9 .9 .9 .1 .1 .1

.1 .1 .9 .1 .1 .1 .1 .9 .1 .1

.1 .9 .1 .1 .1 .1 .1 .9 .1 .1

.1 .9 .1 .1 .1 .1 .1 .1 .9 .1

.9 .1 .1 .1 .1 .1 .1 .1 .1 .9

.9 .1 .1 .1 .1 .1 .1 .1 .1 .9

.9 .1 .1 .1 .1 .1 .1 .1 .1 .9

.1 .9 .1 .1 .1 .1 .1 .9 .9 .1

.1 .9 .1 .1 .1 .1 .1 .1 .1 .1

.1 .1 .9 .9 .9 .9 .9 .1 .1 .1

.1 .1 .1 .1 .9 .1 .1 .1 .1 .1

.1 .1 .1 .9 .1 .9 .1 .1 .1 .1

.1 .1 .9 .1 .1 .1 .9 .1 .1 .1

.1 .1 .9 .1 .1 .1 .1 .9 .1 .1

.1 .9 .1 .1 .1 .1 .1 .9 .1 .1

.1 .9 .1 .1 .1 .1 .1 .1 .9 .1

.1 .9 .1 .1 .1 .1 .1 .1 .9 .1

.9 .1 .1 .1 .1 .1 .1 .1 .9 .1

.9 .1 .1 .1 .1 .1 .1 .1 .1 .9

.9 .9 .9 .9 .9 .9 .9 .9 .9 .9

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การจำลองการทำงานโครงข่ายประสาทเทียมแบบแพร่ย้อนกลับ

Neural Network Back-Propagation Learning Simulation

สุรสิทธิ์ วรรณไกรโรจน์ * สุทธิชัย มณีรัตนรุ่งโรจน์ **

ปัจจุบันโครงข่ายประสาทเทียมเริ่มได้รับความสนใจที่จะนำไปใช้ในโครงงานด้านต่างๆ เช่นนำไปประยุกต์ใช้กับการจดจำรูปแบบ การประมวลผลสัญญาณภาพ ควบคุมการเคลื่อนไหวของหุ่นยนต์ และอื่นๆ ซึ่งทำให้การพัฒนาศาสตร์ทางด้านโครงข่ายประสาทเทียมนี้มีความสำคัญยิ่งเพราะเป็นแนวทางหนึ่งของความก้าวหน้าทางเทคโนโลยี ในบทความที่นำเสนอนี้กล่าวถึงการจำลองการทำงานโครงข่ายประสาทเทียมโดยวิธีการแพร่ย้อนกลับซึ่งเป็นวิธีที่สำคัญและนิยมใช้กันมากประมาณ 90 เปอร์เซ็นของโครงงานที่ใช้โครงข่ายประสาทเทียมประยุกต์ร่วมด้วย โดยจะจำลองการทำงานด้วยโปรแกรมคอมพิวเตอร์และแสดงผลการทำงานด้วยภาพกราฟิก

So far, it is well recognized that artificial neural network can be applied and used in many fields of works. For example, it can be applied to use in the pattern recognition, image processing, adaptive robot movement control, and so on. Therefore, the development of the field is considered to be extreamly important issue. In this report describes the architecture and simulation of a feedforward network. The most widely used and easily recognized neural network is the so-call backpropagation network. By some estimates, it accounts for 90 percent of all neural network applications. There is also the simulation of working according to theory made by computer program that show working effect with graphic.

1. บทนำ

หลักการแพร่ย้อนกลับได้รับการพัฒนาโดยรูเมลฮาร์ท (Rumelhart) และถูกนำเสนอในช่วงปี 1986 โดยหลักการแพร่ย้อนกลับนี้สามารถประยุกต์เข้ากับโครงงานต่างๆได้อย่างหลากหลายทำให้ได้รับความนิยมที่จะนำไปใช้มากที่สุดในปัจจุบัน หลักการนี้มีพื้นฐานมาจากกฎเดลต้า (delta rule) ที่พัฒนาขึ้นมาโดยวิดโรว์ (Widrow) และ ฮอฟฟ์ (Hoff) ในช่วงปี 1960-1962 ซึ่งกระบวนการของกฎเดลต้าจะลดค่าผิดพลาดที่เอาต์พุทของโครงข่ายเปรียบเทียบกับค่าเป้าหมาย โดยค่าผิดพลาดที่ลดลงนี้เกิดจากการปรับแต่งค่าน้ำหนัก (weight) ที่อินพุทของแต่ละโหนด บทความที่นำเสนอนี้จะกล่าวถึงทฤษฎีแพร่ย้อนกลับและจำลองการทำงานด้วยโปรแกรมคอมพิวเตอร์โดยแสดงผลการทำงานด้วยภาพกราฟิกซึ่งมีความยืดหยุ่นและสะดวกต่อการใช้งาน

2. หลักการแพร่ย้อนกลับ

จากรูปที่ 1 แสดงให้เห็นถึงลักษณะของโครงข่ายบีพีเอ็น (BPN, Black-Propagation Neural Network) ซึ่งเป็นฟีดฟอร์เวิร์ดเน็ทเวิร์ค (feedforward network) โดยมีการเชื่อมโยงในแต่ละชั้น(layer) แบบฟูลลี่ (fully) ซึ่งลักษณะเช่นนี้จะไม่มีการป้อนกลับ จากรูปแสดงให้เห็นโครงข่ายที่มี 3 ชั้น คือชั้นอินพุท (input layer) ชั้นซ่อน(hidden layer)และชั้นเอาต์พุท (output layer) ในทางปฏิบัติที่ชั้นซ่อนสามารถที่จะมีได้มากกว่า 1 ชั้น

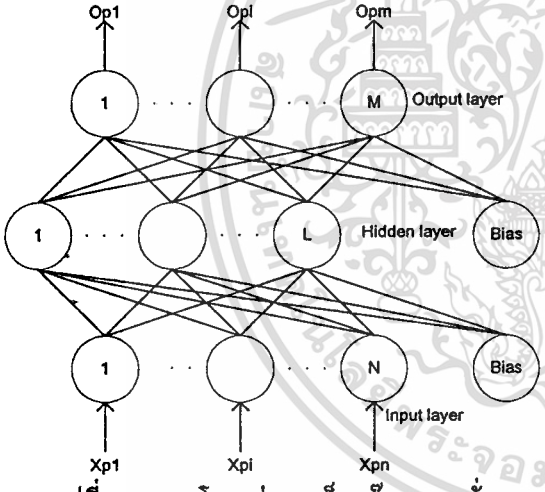
* อาจารย์คณะเทคโนโลยีสารสนเทศ สจล.

** นักศึกษาปริญญาโท คณะเทคโนโลยีสารสนเทศ สจล.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โครงข่ายประสาทเทียมเรียกได้ว่าเป็นแมพปิ้งเน็ทเวิร์ค(mapping network) หากโครงข่ายสามารถคำนวณฟังก์ชั่นในบางลักษณะที่กำเนิดความสำพันธ์ระหว่างอินพุทและเอาต์พุทในโครงข่าย ตัวอย่างเช่น ถ้ามีอินพุทไปยังโครงข่ายซึ่งมีค่าเป็นมุม (angle) และเอาต์พุทคือค่าโคซายน์(cosine) ของมุมที่อินพุท โครงข่ายจะมีการแมพปิ้งคือ $\theta \longrightarrow \cos(\theta)$

สมมุติให้มีเซต P เป็นเซตของคู่ลำดับอินพุท-เอาต์พุท (vector-pairs) ซึ่งมีสมาชิก (X1,Y1), (X2,Y2),...,(Xt,Yt) ตัวอย่างของฟังก์ชั่นแมพปิ้งคือ $y = \varnothing(x)$, $x \in RN$, $y \in RM$ และต้องการฝึกหัดโครงข่ายให้เรียนรู้ในการเข้าหาค่าประมาณที่ใกล้เคียงคือ $O=y' = \varnothing'(x)$ การฝึกหัดโครงข่ายนั้นก็คือวิธีการที่จะหาค่าหนักที่เหมาะสม



รูปที่ 1 แสดงโครงข่ายแบล็คพร๊อบพะเกชั่น

เริ่มต้นด้วยการพิจารณารูปที่ 1 ซึ่งเป็นโครงข่ายขนาด 3 ชั้น ที่มีอินพุทเวกเตอร์ $x_p=(x_{p1},x_{p2},...,x_{pn})t$ ที่เป็นอินพุทป้อนให้กับโหนดอินพุทที่ชั้นอินพุทของโครงข่าย และโหนดอินพุทจะกระจายค่าไปยังโหนดต่างๆ ในชั้นซ่อนซึ่งที่ชั้นซ่อนนี้เองจะทำการคำนวณค่าเน็ท (net) สมการหาค่าเน็ทที่โหนด j ในชั้นซ่อนคือ

$$net^h_{pj} = \sum_{i=1}^{N} w^h_{ji} x_{pi} + \theta^h_j$$

เมื่อ w^h_{ji} คือน้ำหนักที่ชั้นซ่อนตรงจุดต่อจากโหนดอินพุท i ไปยังโหนด j ในชั้นซ่อน และ θ^h_j คือค่าเทอมไบแอส (bias) เมื่อได้ค่าเน็ทแล้วขั้นถัดไปคือคำนวณค่าเอาต์พุทของแต่ละโหนดในชั้นซ่อนโดยคำนวณได้ดังนี้

$$i_{pj} = f^h_j(net^h_{pj})$$

จากนั้นจึงทำการคำนวณค่าเน็ทและค่าเอาต์พุทที่ชั้นเอาต์พุทโดยใช้สมการ

$$net^o_{pk} = \sum_{j=1}^{L} w^o_{kj} i_{pj} + \theta^o_k$$

$$o_{pk} = f^o_k(net^o_{pk})$$

*** หมายเหตุ การกำหนดค่าน้ำหนักเริ่มต้นนั้นได้มาจากการสุ่มค่า(random) โดยทั่วๆไปค่านี้จะอยู่ระหว่าง -0.5 และ 0.5 หรือ -1 และ +1

2.1 การหาค่าผิดพลาดและการปรับน้ำหนักที่ชั้นเอาต์พุท

สมการปรับค่าน้ำหนักที่ชั้นเอาต์พุทแสดงได้ดังนี้

$$w^o_{kj}(t+1) = w^o_{kj}(t) + \Delta_p w^o_{kj}(t)$$

$$\Delta_p w^o_{kj} = \eta (y_{pk} - o_{pk}) f^{o}_{k}{}' (net^o_{pk}) i_{pj}$$

$$i_{pj} = f^h_j(net^h_{pj})$$

องค์ประกอบη คืออัตราการเรียนรู้ (learning rate)โดยปกติค่านี้จะมีค่าเป็นบวกและน้อยกว่า1 ลองกลับไปดูฟังก์ชั่น f^o_k และ f^h_j ซึ่งมี 2 รูปแบบที่น่าสนใจคือ

$$f^o_k(net^o_{jk}) = net^o_{jk}$$

$$f^o_k(net^o_{jk}) = (1 + e^{-netojk})^{-1}$$

ฟังก์ชั่นรูปแบบแรกเป็นการกำหนดเอาต์พุทเป็นแบบลิเนียร์ ส่วนฟังก์ชั่นรูปแบบหลังเรียกว่าซิกมอยด์ (sigmoid) ซึ่งรูปที่ 2 แสดงกราฟของฟังก์ชั่นนี้ การเลือกใช้ฟังก์ชั่นนั้นขึ้นอยู่กับว่าต้องการเลือกลักษณะข้อมูลที่เอาต์พุทอย่างไร ถ้าเลือกฟังก์ชั่นรูป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แบบแรก $f^{o}_{k}' = 1$ ถ้าเลือกรูปแบบที่ 2 ฟังก์ชั่น $f^{o}_{k}' = f^{o}_{k}(1 - f^{o}_{k}) = o_{pk}(1 - o_{pk})$ ดังนั้นการปรับน้ำหนักในกรณีของลิเนียร์จะกำหนดได้ดังนี้

$$w^{o}_{kj}(t+1) = w^{o}_{kj}(t) + \eta(y_{pk} - o_{pk})\, i_{pj}$$

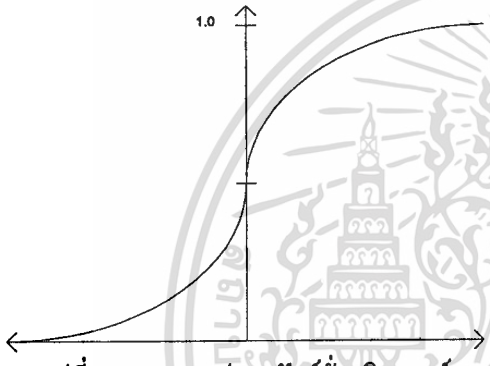
ส่วนฟังก์ชั่นซิกมอยด์กำหนดได้ดังนี้

$$w^{o}_{kj}(t+1) = w^{o}_{kj}(t) + \eta(y_{pk} - o_{pk})\, o_{pk}\,(1-o_{pk})\, i_{pj}$$

เขียนสมการให้ดูง่ายขึ้น

$$\sigma^{o}_{pk} = (y_{pk} - o_{pk})\, f^{o}_{k}'(net^{o}_{pk})$$

$$w^{o}_{kj}(t+1) = w^{o}_{kj}(t) + \eta\, \sigma^{o}_{pk}\, i_{pj}$$



รูปที่ 2 แสดงกราฟของฟังก์ชั่น ซิกมอยด์

2.2 การปรับน้ำหนักที่ชั้นซ่อน

การปรับแต่งค่าน้ำหนักที่ชั้นซ่อนจะต้องทำการคำนวณหาค่าผิดพลาดที่ชั้นซ่อน(hidden-layer error term)ก่อนโดยใช้สมการ

$$\sigma^{h}_{pj} = f^{h}_{j}'(net^{h}_{pj}) \sum_{k} \sigma^{o}_{pk}\, w^{o}_{kj}$$

จากนั้นจึงสามารถปรับแต่งค่าน้ำหนักโดยสมการที่คล้ายคลึงกับสมการปรับแต่งค่าน้ำหนักที่ชั้นเอาต์พุทคือ

$$w^{h}_{ji}(t+1) = w^{h}_{ji}(t) + \eta\sigma^{h}_{pj}\, x_{i}$$

2.3 สรุปขั้นตอนการฝึกหัดโครงข่ายประสาทเทียมโดยวิธีการแพร่ย้อนกลับ

1. กำหนดอินพุทเวคเตอร์ $x_n = (x_{p1}, x_{p2}, \ldots, x_{pn})$ แก่โหนดอินพุท

2. คำนวณหาค่าเน็ทอินพุทในโหนดของชั้นซ่อนโดยสมการ

$$net^{h}_{pj} = \sum_{i=1}^{N} w^{h}_{ji}\, x_{pi} + \theta^{h}_{j}$$

3. คำนวณค่าเอาต์พุทของโหนดในชั้นซ่อน

$$i_{pj} = f^{h}_{j}(net^{h}_{pj})$$

4. คำนวณค่าเน็ทของโหนดในชั้นเอาต์พุท

$$net^{o}_{pk} = \sum_{j=1}^{L} w^{o}_{kj}\, i_{pj} + \theta^{o}_{k}$$

5. คำนวณค่าเอาต์พุท

$$o_{pk} = f^{o}_{k}(net^{o}_{pk})$$

6. คำนวณเทอมค่าผิดพลาดสำหรับโหนดในชั้นเอาต์พุท

$$\sigma^{o}_{pk} = (y_{pk} - o_{pk})\, f^{o}_{k}'(net^{o}_{pk})$$

7. คำนวณเทอมค่าผิดพลาดสำหรับโหนดในชั้นซ่อน

$$\sigma^{h}_{pj} = f^{h}_{j}'(net^{h}_{pj}) \sum_{k} \sigma^{o}_{pk}\, w^{o}_{kj}$$

*** หมายเหตุ ค่าเทอมผิดพลาดของโหนดในชั้นซ่อนต้องคำนวณก่อนที่จะมีการปรับแต่งค่าน้ำหนักที่ชั้นเอาต์พุท

8. ปรับแต่งค่าน้ำหนักของชั้นเอาต์พุท

$$w^{o}_{kj}(t+1) = w^{o}_{kj}(t) + \eta\sigma^{o}_{pk}\, i_{pj}$$

9. ปรับแต่งค่าน้ำหนักของชั้นซ่อน

$$w^{h}_{ji}(t+1) = w^{h}_{ji}(t) + \eta\sigma^{h}_{pj}\, x_{i}$$

และต้องคำนวณเทอมค่าผิดพลาด

$$Ep = \left(\sum_{p=1}^{M} \left(\sum_{k=1} (y_{pk} - o_{pk})^{2}\right)^{1/2}\right)/p$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซึ่งเป็นปริมาณการวัดว่าโครงข่ายเรียนรู้ได้ดีเพียงไร เมื่อค่าผิดพลาดมีค่าต่ำลงจนสามารถยอมรับได้สำหรับทุก ๆ คู่อันดับ อินพุท-เอาต์พุท ของกลุ่มข้อมูลทดสอบ การฝึกหัดนี้ก็ยุติลงได้

3. การออกแบบโปรแกรมจำลองการทำงานโครงข่ายประสาทเทียม

จากทฤษฎีของวิธีการแบบแพร่ย้อนกลับดังที่ได้กล่าวมาแล้ว ได้ถูกนำมาสร้างเป็นโปรแกรมเพื่อใช้งานจริงโดยได้นำหลักทฤษฎีทางคณิตศาสตร์ข้างต้นมาดัดแปลงให้อยู่ในรูปของอัลกอริธึม เพื่อลำดับขั้นตอนในการสร้างโปรแกรม ในหัวข้อนี้จะกล่าวถึงกระบวนการออกแบบและวิธีการที่นำมาใช้ในการทำงานของโปรแกรม

3.1 โครงสร้างหลักของโปรแกรม

โปรแกรมที่สร้างขึ้นมานี้ถูกออกแบบในลักษณะของโปรแกรมเชิงวัตถุ(Oject-Oriented Programming) โดยจัดแบ่งหน้าที่การทำงานในส่วนต่างๆเป็นคลาส(class) ซึ่งสามารถจำแนกคลาสต่างๆ ได้ดังนี้

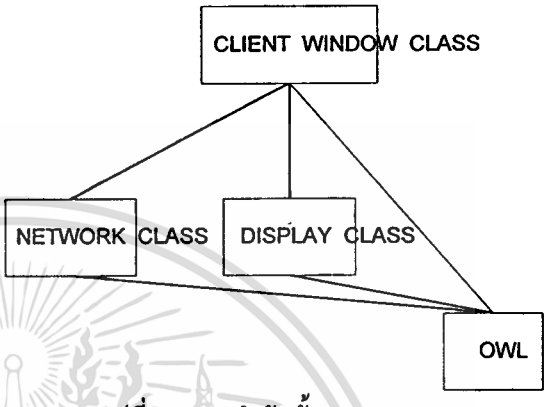
1. คลาสไคลเอนท์วินโดว์ (client window class) เป็นคลาสหลักของการทำงาน ภายในคลาสประกอบด้วยตัวแปรและกลุ่มฟังก์ชั่นสมาชิกที่ตอบสนองคำสั่งจากเมนู การทำงานของฟังก์ชั่นบางฟังก์ชั่นภายในคลาสยังเรียกใช้คลาสย่อยอื่นๆที่กำหนดไว้ด้วย

2. คลาสโครงข่าย (network class) ภายในคลาสประกอบด้วยตัวแปรและฟังก์ชั่นสมาชิกที่ใช้ในการทำงานของการจำลองโครงข่ายประสาทเทียม

3. คลาสแสดงผล(display class) ภายในคลาสประกอบด้วยตัวแปรและฟังก์ชั่นสมาชิกที่ใช้ในการแสดงผลการทำงานทางจอภาพ

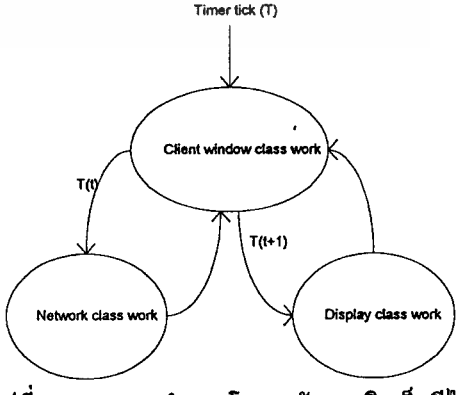
4.ออปเจคท์วินโดว์ไลบรารี่ (Object Window Library, OWL) บรรจุคลาสต่างๆ ของบอร์แลนด์ซีคอมไพเลอร์อันอำนวยความสะดวกแก่การสร้างโปรแกรม

เนื่องจากโปรแกรมทำงานภายใต้สภาวะแวดล้อมของไมโครซอฟท์วินโดว์(Microsoft Window3.1)จึงมีเทคนิคการสร้างโปรแกรมเรียกว่ามัลติเปิลดอ



รูปที่ 3 แสดงลำดับชั้นของคลาส

คูเมนท์อินเตอร์เฟส(Multiple Document Interface) หรือ เอ็มดีไอ(MDI) ซึ่งทำให้การทำงานของตัวโปรแกรมสามารถที่จะทำงานได้หลายๆ งานพร้อม ๆ กัน โดยอาศัยฐานเวลาเป็นตัวกระตุ้นการทำงาน ที่ระดับการทำงานในชั้นของคลาสไคลเอนท์วินโดว์จะได้การกระตุ้นของฐานเวลาทุกๆ 10 มิลลิเซค (ค่าฐานเวลานี้สามารถกำหนดได้ตามความต้องการ) จากนั้นที่ไคลเอนท์ก็จะส่งต่อการทำงานให้กับคลาสอื่นๆ ที่ได้กำหนดไว้คือคลาสโครงข่ายและคลาสแสดงผลตามลำดับ ดังแสดงในรูปที่ 4



รูปที่ 4 แสดงการทำงานโดยอาศัยเทคนิคเอ็มดีไอ

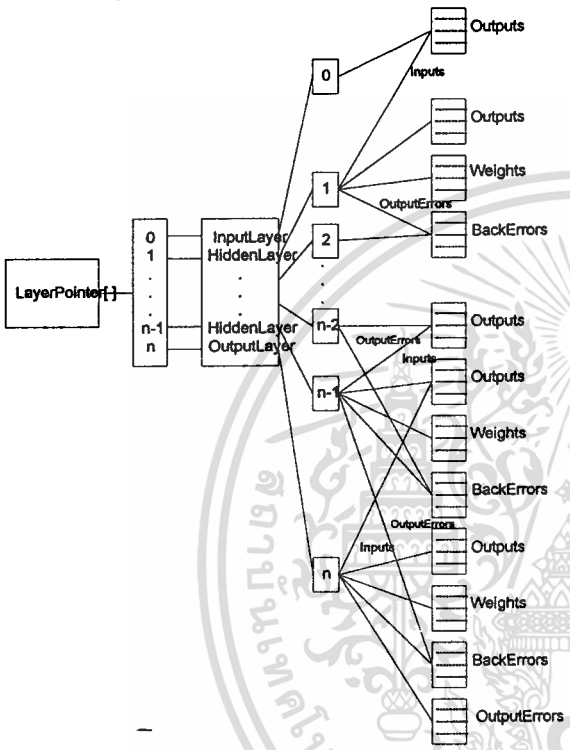
ในหัวข้อถัดไปจะกล่าวถึงอัลกอริธึมที่ใช้ในการจำลองการทำงานโครงข่ายประสาทเทียมแบบแพร่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ย้อนกลับซึ่งอัลกอริธึมเหล่านี้ถูกนำมาสร้างเป็นฟังก์ชั่นสมาชิกในคลาสโครงข่าย

3.2 โครงสร้างแบบจำลองโครงข่ายประสาทเทียม

โครงสร้างแบบจำลองโครงข่ายประสาทเทียมนี้ถูกกำหนดไว้ภายในคลาสโครงข่ายโดยมีลักษณะดังรูปที่ 5



รูปที่ 5 แสดงโครงสร้างแบบลองโครงข่ายประสาทเทียม

3.3อัลกอริธึมในกวรสร้างส่วนจำลองการทำงานโครงข่ายประสาทเทียม

สามารถแบ่งหน้าที่เกี่ยวกับการจำลองได้เป็นสองส่วนหลักๆ คือ การทำงานในการฝึกหัดโครงข่ายและการทำงานในการทดสอบหาผลลัพธ์ของโครงข่าย

วิธีการทำงานในการฝึกหัดโครงข่ายมีลำดับขั้นตอนดังนี้

```
begin
  random weight
  while network error is larger than some predefine limit
  do
    set input to apply a taining input
    forward propagate to generate an outputs
    backward propagate to update error values
    update weights
  end do
  save weights to file
end
```

วิธีการทดสอบโครงข่ายมีลำดับขั้นตอนดังนี้

```
begin
  set weights value from file
  set input to stimulate the network with an input
  forward propagate to ganerate an output
  save output to file
end
```

ต่อไปจะแสดงอัลกอริธึมที่สำคัญสามส่วน คือ forward propagate, backward propagate และ update weight และเนื่องจากโปรแกรมจำลองการทำงานโครงข่ายประสาทเทียมนี้ถูกพัฒนาขึ้นมาจากภาษาซีพลัสพลัส(C++) ดังนั้นจึงแสดงการทำงานในรูปแบบของภาษาที่ใช้อยู่นี้

3.3.1 อัลกอริธึม forward propagate

ฟังก์ชั่น forward propagate ทำหน้าที่คำนวณค่าเอาต์พุทที่ชั้นซ่อนและชั้นเอาต์พุท

```
void Network :: ForwardPropagate()
{
 int i;
 for(i=1; i<NumberOfLayers; i++)
   LayerPointer[i]->CalculateOutput();
}
void CalculateOutput()
{
 int i, j, k;
 float Net;
 for(j=0; j<NumOutputs; j++){
   for(i=0; i<NumInputs; i++){
     k = i*NumOutputs;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    Outputs[j] = Weights[k+j] * Inputs[i];
    Net += Outputs[j]
   }
   Outputs[j] = (float)(1/(1+exp(-Net));
  }
}
```

3.3.2 อัลกอริธึม backward propagate

มีหน้าที่ในการคำนวณค่าผิดพลาดที่ชั้นเอาต์พุทและชั้นซ่อน

```
void Network :: BackwardPropagate()
{
 int i;
 for(i=NumberOfLayers; i>0; i--)
  LayerPointer[i]->CalculateError();
}
```

ฟังก์ชั่นย่อยที่เรียกใช้คือ CalculateError ซึ่งเป็นฟังก์ชั่นโพลิมอฟิค(polymorphic) ที่ชั้นเอาต์พุทและชั้นซ่อนดังแสดงได้ดังนี้

```
void OutputLayer :: CalculateError()
{
 int i, j, k;
 float Summation = 0;
 for(j=0; j<NumOutputs; j++)
  OutputErrors[j] = (TargetValue[j] - Outputs[j])*
                    (Outputs[j]*(1-Outputs[j]));
 for(i=0; i<NumInputs; i++)
  k = i*NumOutputs;
  for(j=0; j<NumOutputs; j++){
   BackErrors[i] = Weights[k+j] * OutputErrors[j];
   Summation += BackErrors[i];
  }
  BackErrors[i] = Summation;
  Summation = 0;
  BackErrors[i] *= Inputs[i] * (1-Inputs[i]);
 }
}
```

```
void HiddenLayer :: CalculateError()
{
 int i, j, k;
 float Summation = 0;
 for(i=0; i<NumInputs; i++)
  k = i*NumOutputs;
  for(j=0; j<NumOutputs; j++){
   BackErrors[i] =Weights[k+j] * OutputErrors[j];
   Summation += BackErrors[i];
  }
  BackErrors[i] = Summation;
  Summation = 0;
  BackErrors[i] *= Inputs[i] * (1-Inputs[i]);
 }
}
```

3.3.3 อัลกอริธึม update weights

มีหน้าที่ในการปรับแต่งค่าน้ำหนักที่ชั้นซ่อนและชั้นเอาต์พุท

```
void Network :: UpdateWeights()
{
 int i;
 for(i=1; i<NumberOfLayers; i++)
  LayerPointer[i]->Update(Eta);
}
void Update(float Eta)
{
 int i, j, k;
 for(i=0; i<NumInputs; i++){
  k = i * NumOutputs;
  for(j=0; j<NumOutputs; j++)
   Weights[k+j] += Eta * OutputErrors *Inputs[i];
 }
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. การใช้งานโปรแกรมจำลองการทำงานโครงข่ายประสาทเทียม

ลำดับต่อไปจะกล่าวถึงการใช้โปรแกรมจำลองการทำงานซึ่งโปรแกรมที่สร้างขึ้นนี้มีการทำงานเป็น 2 ลักษณะคือใช้เพื่อการฝึกหัดโครงข่ายประสาทเทียมและใช้เพื่อการทดสอบหาผลลัพธ์ทางเอาต์พุทของโครงข่ายประสาทเทียมโดยจะกล่าวรายละเอียดถึงวิธีการใช้ในหัวข้อถัดไป

4.1 การฝึกหัดโครงข่ายประสาทเทียม

การใช้งานโปรแกรมจำลองการทำงานโครงข่ายประสาทเทียมในการฝึกหัดนั้นก็คือ การนำกลุ่มข้อมูลฝึกหัดซึ่งประกอบด้วยคู่ลำดับ อินพุท-เอาต์พุท ป้อนให้กับโครงข่ายเพื่อให้โครงข่ายทำการเรียนรู้ ซึ่งก่อนที่จะใช้งานโปรแกรมเพื่อการฝึกหัดนั้นจะต้องมีการเตรียมการ ล่วงหน้าก่อน สิ่งที่ต้องตระเตรียมคือ

- กำหนดแฟ้มข้อมูลที่มีนามสกุล .NET เพื่อกำหนดโครงสร้างของโครงข่ายประสาทเทียมที่ต้องการโดยขอยกตัวอย่างแฟ้มให้ชื่อว่า STRUCT.NET มีข้อมูลในแฟ้มดังนี้

LAYER : 1
NODE : 2
LAYER : 2
NODE : 5
LAYER : 3
NODE : 1
LEARN-RATE : 0.5
ERROR-TOLERANCE : 0.01
MAX-CYCLE : 1000

ข้อมูลกำหนดโครงสร้างมีความหมายดังนี้

LAYER : x ใช้กำหนดลำดับชั้นของโครงข่าย

NODE : n ใช้กำหนดจำนวนของโหนดในชั้นที่ได้ระบุก่อนหน้านี้โดยที่ n คือจำนวนโหนดมีค่าได้ตั้งแต่ 1 ขึ้นไป

LEARN-RATE : f ใช้กำหนดอัตราการเรียนรู้โดยที่ f มีค่าได้ตั้งแต่ 0.01 ถึง 1.0

ERROR-TOLERANCE : e ใช้กำหนดค่าผิดพลาดที่ยอมรับได้ในการฝึกหัดโครงข่ายโดยที่ e สามารถกำหนดได้ในช่วง 0.001 ถึง 100.0

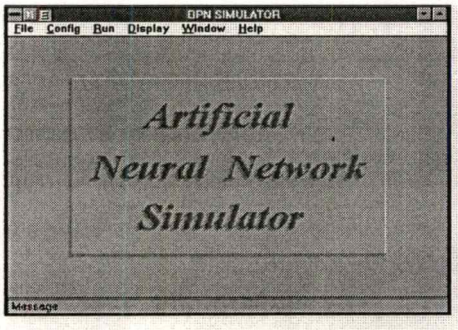
MAX-CYCLE : m ใช้กำหนดจำนวนรอบสูงสุดในการเรียนรู้โดยที่ m สามารถกำหนดได้ตั้งแต่ 1ถึง 1000000

- กำหนดแฟ้มข้อมูลที่มีนามสกุล .TRN ซึ่งเป็นแฟ้มข้อมูลฝึกหัดภายในแฟ้มนี้ประกอบด้วยข้อมูลอินพุท-เอาต์พุท ที่ใช้ในการฝึกหัดโครงข่าย จะขอยกตัวอย่างของการฝึกหัดโครงข่ายให้เรียนรู้ถึงตรรก "XOR" ซึ่งเป็นปัญหาไม่เชิงเส้นมาตรฐานของวงการโครงข่ายประสาทเทียม ข้อมูลภายในแฟ้ม XOR.TRN มีดังนี้

0.1 0.1 0.1
0.1 0.9 0.9
0.9 0.1 0.9
0.9 0.9 0.1

คอลัมท์ที่ 1 และ 2 คือข้อมูลอินพุท คอลัมท์ที่ 3 คือข้อมูลเอาต์พุทที่เป็นค่าเป้าหมาย

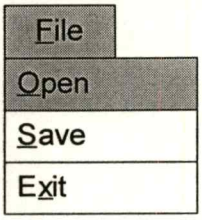
หลังจากที่ได้สร้างแฟ้มข้อมูล STRUCT.NET และ XOR.TRN เป็นที่เรียบร้อยแล้วก็เป็นอันว่าพร้อมที่จะใช้งานโปรแกรมจำลองการทำงานโครงข่ายประสาทเทียม เมื่อเรียกโปรแกรมมาทำงานก็จะเห็นการแสดงผลเริ่มต้นดังรูปที่ 6



รูปที่ 6 การแสดงผลเริ่มต้นของโปรแกรม

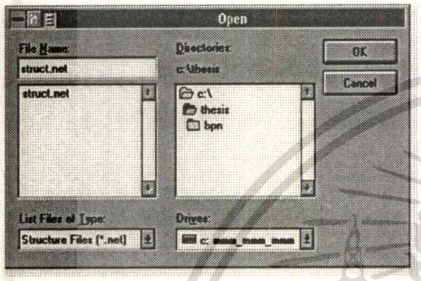
ให้เลือกการทำงานที่หัวข้อ File ที่เมนูบาร์ จากนั้นจึงเลือก Open

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



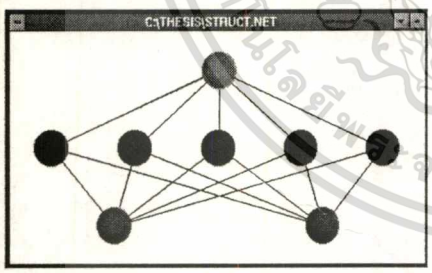
รูปที่ 7 แสดงการเปิดแฟ้มข้อมูล

เมื่อเลือก Open แล้วจะปรากฏไดอะล็อกสำหรับเปิดแฟ้มข้อมูลให้คีย์ชื่อแฟ้มข้อมูล STRUCT.NET ที่ได้สร้างไว้ก่อน แล้วกดปุ่ม OK



รูปที่ 8 แสดงไดอะล็อกเปิดแฟ้มข้อมูล

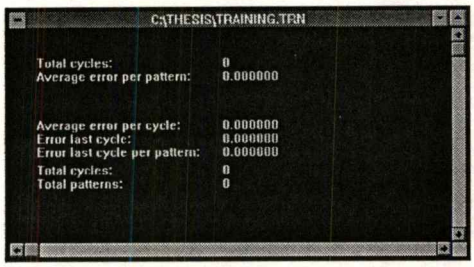
โปรแกรมจำลองการทำงานจะสร้างหน้าต่างลูก (child window) โดยภายในหน้าต่างแสดงให้เห็นถึงลักษณะโครงข่ายที่ได้ทำการกำหนดไว้ในแฟ้มข้อมูล STRUCT.NET



รูปที่ 9 หน้าต่างลูกแสดงโครงข่าย

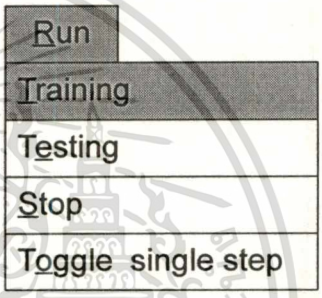
จากข้อมูลโครงสร้างที่กำหนดไว้จะมีชั้นอินพุท ชั้นซ่อน และชั้นเอาต์พุท ชั้นอินพุทแสดงด้วยโหนดสีแดง ชั้นซ่อนแสดงด้วยโหนดสีน้ำเงินและชั้นเอาต์พุทแสดงด้วยโหนดสีเขียว

ขั้นตอนต่อไปให้ทำการเปิดแฟ้มข้อมูลอีกครั้งหนึ่งโดยครั้งนี้ให้คีย์ชื่อ XOR.TRN ซึ่งเป็นแฟ้มข้อมูลฝึกหัด กดปุ่ม OK โปรแกรมจำลองการทำงานจะสร้างหน้าต่างลูกขึ้นมาอีกหนึ่งหน้าต่างโดยหน้าต่างที่สร้างขึ้นใหม่นี้จะแสดงข้อมูลรายละเอียดในระหว่างทำการฝึกหัดโครงข่าย



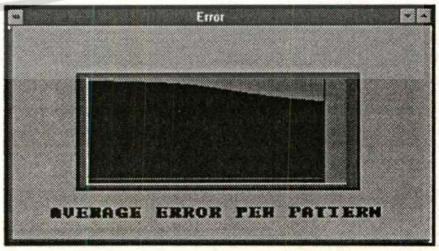
รูปที่10หน้าต่างลูกแสดงรายละเอียดในการทำงาน

เลือกการทำงานที่หัวข้อ Run บนเมนูบาร์ แล้วเลือก Training



รูปที่ 11 แสดงการเลือกฝึกหัดโครงข่าย

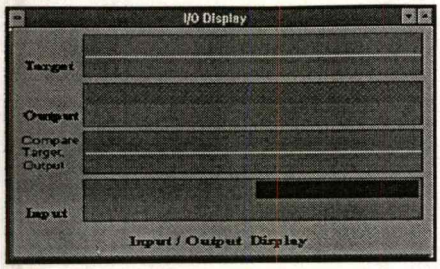
โปรแกรมจะเริ่มทำงานในการเรียนรู้ตรรก "XOR" ที่กำหนดไว้ในแฟ้มข้อมูลฝึกหัด ในระหว่างที่ทำการฝึกหัดโครงข่ายนี้สามารถที่จะเปิดหน้าต่างแสดงค่าผิดพลาดได้ ด้วยการเลือกหัวข้อ Display บนเมนูบาร์แล้วเลือก Display Error โปรแกรมจะสร้างหน้าต่างลูกเพื่อแสดงค่าผิดพลาดอีกหนึ่งหน้าต่าง



รูปที่ 12 หน้าต่างลูกแสดงค่าผิดพลาด

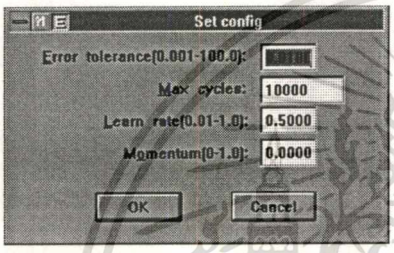
เลือก Display I/O โปรแกรมจะสร้างหน้าต่างลูกเพื่อแสดงระดับสัญญาณอินพุท เอาต์พุท ค่าเป้าหมายและค่าผิดพลาดอีกหนึ่งหน้าต่าง

เอกสารนี้มีไว้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



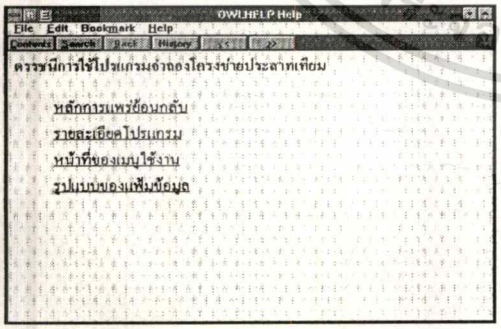
รูปที่ 13 หน้าต่างลูกแสดง I/O

ถ้าต้องการเปลี่ยนแปลงค่าพารามิเตอร์ที่ใช้ในการฝึกหัดก็สามารถทำได้โดยการเลือกหัวข้อ Config แล้วเลือก Set config เพื่อทำการแก้ไขค่าพารามิเตอร์ใหม่



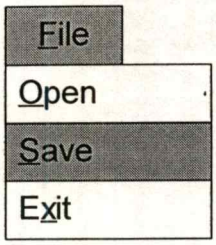
รูปที่ 14 ไดอะล็อกสำหรับเปลี่ยนแปลงค่าพารามิเตอร์

โปรแกรมยังมีส่วนช่วยเหลือ(Help) กรณีที่ผู้ใช้ต้องการทราบรายละเอียดเกี่ยวกับการใช้งานโปรแกรมและส่วนช่วยเหลือจะแสดงผลเป็นภาษาไทย โดยผู้ใช้สามารถใช้ส่วนช่วยเหลือนี้ได้โดยเลือกเมนูบาร์ Help แล้วเลือก Index help



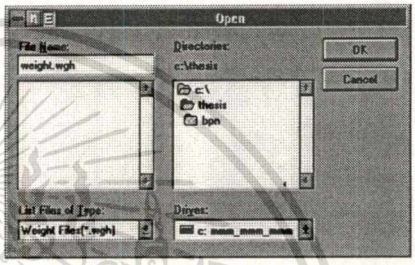
รูปที่ 15 แสดงส่วนช่วยเหลือของโปรแกรม

เมื่อการฝึกหัดเสร็จสมบูรณ์ก็ทำการเก็บค่าน้ำหนักไว้ในแฟ้มข้อมูลน้ำหนักที่มีนามสกุล .WGH การเก็บค่าน้ำหนักนี้ทำได้โดยเลือกการทำงานที่หัวข้อ File ที่เมนูบาร์จากนั้นจึงเลือก Save



รูป 16 แสดงการจัดเก็บค่าน้ำหนัก

ใส่ชื่อแฟ้มข้อมูลที่จะทำการเก็บค่าน้ำหนัก โดยในตัวอย่างนี้ขอตั้งชื่อแฟ้มว่า WEIGHT.WGH แล้วจึงกดปุ่ม OK



รูป 17 แสดงไดอะล็อกการจัดเก็บค่าน้ำหนัก

4.2การทดสอบหาผลลัพธ์ของโครงข่ายประสาทเทียม

การใช้งานโปรแกรมจำลองการทำงานโครงข่ายประสาทเทียมในการทดสอบหาผลลัพธ์นั้นก็คือการนำโครงข่ายประสาทเทียมที่ได้มีการฝึกหัดแล้วมาใช้เป็นตัวตัดสินในการหาผลลัพธ์ของกลุ่มข้อมูลทดสอบ ก่อนที่จะใช้งานโปรแกรมจะต้องมีการจัดเตรียมแฟ้มข้อมูลที่จำเป็นคือ

- แฟ้มข้อมูลโครงสร้าง มีนามสกุล .NET
- แฟ้มข้อมูลน้ำหนัก ได้จากการฝึกหัด มีนามสกุล .WGH
- แฟ้มข้อมูลทดสอบ มีนามสกุล .TES

แฟ้มข้อมูลสองแฟ้มแรกนั้นได้กล่าวผ่านมาแล้ว จะมีแฟ้มใหม่ที่ต้องสร้างขึ้นมาคือแฟ้มข้อมูลทดสอบโดยขอยกตัวอย่างข้อมูลในแฟ้มข้อมูล TEST.TES ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

0.8 0.8

0.9 0.2

0.15 0.8

0.3 0.2

จะเรียกข้อมูลในแต่ละแถวว่าแพทเทิร์น (pattern) โดยที่หนึ่งแพทเทิร์นประกอบด้วยข้อมูลอินพุท ในแฟ้ม TEST.TES นี้จะมีสองอินพุท

ขั้นตอนการทดสอบหาผลลัพธ์ของโครงข่ายประสาทเทียมทำได้ดังนี้ ขั้นแรกให้เลือกการทำงานที่หัวข้อ File บนเมนูบาร์จากนั้นจึงเลือก Open เพื่อทำการเปิดแฟ้มข้อมูลโครงสร้าง STRUCT.NET เมื่อเปิดแฟ้มแล้วโปรแกรมจำลองการทำงานจะสร้างหน้าต่างลูกเพื่อแสดงโครงสร้างของโครงข่ายที่ได้กำหนดไว้ ดังแสดงในรูปที่ 9 จากนั้นจึงเปิดแฟ้มข้อมูลน้ำหนัก WEIGHT.WGH เพื่อโหลดค่าน้ำหนักให้แก่โครงข่าย สุดท้ายจึงทำการเปิดแฟ้มข้อมูลทดสอบ TEST.TES เพื่อใช้เป็นอินพุทแก่โครงข่าย ขั้นตอนถัดไปให้เลือกการทำงานที่หัวข้อ Run บนเมนูบาร์ แล้วเลือก Testing โปรแกรมจำลองการทำงานจะสร้างหน้าต่างลูกแสดงให้เห็นข้อมูลแพทเทิร์นทางอินพุทและผลลัพธ์ทางเอาต์พุท เมื่อการทำงานสิ้นสุดลงโปรแกรมจะสร้างแฟ้มข้อมูล OUTPUT.DAT เพื่อเก็บผลลัพธ์ทางเอาต์พุทของแต่ละแพทเทิร์นข้อมูลในแฟ้ม OUTPUT.DAT ในกรณีตัวอย่างจะให้ผลดังนี้

input vector :

0.800000 0.800000

output vector:

0.136662

---

input vector :

0.900000 0.200000

output vector:

0.844238

---

input vector :

0.150000 0.800000

output vector:

0.813372

---

input vector :

0.300000 0.200000

output vector:

0.323352

5.บทสรุป

หลักการของโครงข่ายประสาทเทียมนับว่าเป็นศาสตร์ที่มีการพัฒนากันมานานพอสมควรแต่เพิ่งเริ่มที่จะนำมาใช้กันอย่างจริงจัง ทั้งนี้เพราะในปัจจุบันกฎการเรียนรู้ของโครงข่ายประสาทได้รับการพัฒนาจนพอที่จะประยุกต์ใช้กับโครงงานต่างๆได้อย่างหลากหลายและมีประสิทธิภาพพอเพียง แต่การพัฒนารูปแบบหรือวิธีการก็ยังคงกระทำกันต่อไป โดยมีเป้าประสงค์ที่จะก่อกำเนิดรูปแบบการเรียนรู้ที่คล้ายคลึงกับมนุษย์ ซึ่งดูเลื่อนลางเพราะนั่นหมายถึงว่าเราอาจจะต้องศึกษาและเข้าใจกลไกการทำงานของสมองหรือระบบประสาทกันอย่างถ่องแท้ก่อน จากนั้นจึงแปลงกระบวนการเหล่านี้ให้อยู่ในเชิงคณิตศาสตร์ที่สามารถอธิบายได้ซึ่งเป็นเรื่องที่ท้าทายที่จะทำการวิจัยกันต่อไป

และจากโปรแกรมจำลองโครงข่ายประสาทเทียมที่ได้นำเสนอไปนั้นได้สร้างขึ้นมาเพื่ออำนวยความสะดวกแก่ผู้ใช้ได้ในระดับหนึ่งทำให้สามารถนำไปประยุกต์ใช้กับโครงงานอื่นๆได้ตามความต้องการ การวิจัยลำดับถัดไปจะทำการจัดสร้างโปรแกรมไลบรารี่ซึ่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ แสวงหาผลกำไร หรือการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รวบรวมฟังก์ชั่นที่จำเป็นต่างๆโดยมีวัตถุประสงค์เพื่อให้ผู้ใช้สามารถเขียนโปรแกรมและเรียกใช้ฟังก์ชั่นเหล่านี้ในการประยุกต์โครงข่ายประสาทเทียมกับโครงงานได้

5.เอกสารอ้างอิง

1. Richard K.Miller, Neural Network, SEAI Technical Publications, 1987
2. James A. Freeman and David M. Skapura, Neural networks - Algorithms - Applications and Programming Techniques, Addison Wesley, 1991
3. I. Guyon, P. Albrecht, Y. Le Cun, J. Denker, and W. Hubbard, Design of a Neural Network Character Recognizer for Touch Terminal, Pattern Recognition, Vol. 24. No2 , PP 105-119, 1991
4. Clayton Walnum, Object-Oriented Programming with Borland C++ 4, Que Corporation, 1994
5. Nabajyoti Barkakati, Borland C++ 4 Developer's Guide, SAMS Publishing, 1994

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประวัติผู้เขียน

ชื่อผู้เขียน	นายสุทธิชัย มณีรัตนรุ่งโรจน์
วันเดือนปีเกิด	วันที่ 21 มกราคม พ.ศ.2511
สถานที่เกิด	จังหวัดชัยภูมิ
วุฒิการศึกษาระดับปริญญาตรี	วิทยาศาสตรบัณฑิต สาขาวิทยาการคอมพิวเตอร์
สถานที่สำเร็จการศึกษา	มหาวิทยาลัยมหิดล
ปีที่สำเร็จการศึกษา	ปีการศึกษา 2535
ผลงานทางวิชาการที่ได้รับการตีพิมพ์	เรื่องการจำลองการทำงานโครงข่ายประสาทเทียมแบบแพร่ย้อนกลับ
ประสพการณ์การทำงาน	เจ้าหน้าที่ฝ่ายวิจัยและพัฒนา บริษัทเพาเวอร์ซอฟท์ เมื่อปี พ.ศ. 2531 ถึง พ.ศ. 2534 เจ้าหน้าที่ฝ่ายวิจัยและพัฒนา บริษัทโมบิคอม เมื่อปี พ.ศ. 2537 ถึง พ.ศ. 2538
อาชีพปัจจุบัน	นักศึกษาสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้