

การปรับเปลี่ยนรูปแบบการเชื่อมต่อของโครงข่ายประสาทเทียม  
ด้วยอัลกอริทึม

ADJUSTING ARTIFICIAL NEURAL NETWORKS TOPOLOGY  
USING ANT ALGORITHM

นิกร โภคอุดม  
NEKORN POKUDOM

วิทยานิพนธ์นี้เป็น ส่วนหนึ่งของเอกสารศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต  
สาขาวิชาวิศวกรรมคอมพิวเตอร์  
คณะวิศวกรรมศาสตร์  
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ. 2552

KMITL-2009-EN-IT-070-182

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

การปรับเปลี่ยนรูปแบบการเชื่อมต่อของโครงข่ายประสาทเทียม  
ด้วยอัลกอริธึมมด

ADJUSTING ARTIFICIAL NEURAL NETWORKS TOPOLOGY  
USING ANT ALGORITHM



นิกร โภคอุดม  
NIKORN POKUDOM

เลขหมู่.....  
เลขทะเบียน.....105512  
วัน,เดือน,ปี..... 24 พ.ย. 2552



วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต  
สาขาวิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ. 2552

KMITL-2009-EN-M-070-152

**ADJUSTING ARTIFICIAL NEURAL NETWORKS TOPOLOGY  
USING ANT ALGORITHM**

**NIKORN POKUDOM**

**A THESIS SUBMITTED IN PARTIAL FULFILLMENT  
OF THE REQUIREMENT FOR THE DEGREE OF  
MASTER OF ENGINEERING IN COMPUTER ENGINEERING  
FACULTY OF ENGINEERING  
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG  
2009  
KMITL-2009-EN-M-070-152**

**COPYRIGHT 2009**

**FACULTY OF ENGINEERING**

**KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG**

คณะวิศวกรรมศาสตร์  
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง  
ใบรับรองวิทยานิพนธ์

หัวข้อวิทยานิพนธ์ การปรับเปลี่ยนรูปแบบการเชื่อมต่อของโครงข่ายประสาทเทียมด้วยอัลกอริทึมมด  
Thesis Title Adjusting Artificial Neural Networks Topology Using Ant Algorithm  
นักศึกษา นายนิกร โภคอุดม  
รหัสประจำตัว 48060707  
ปริญญา วิศวกรรมศาสตรมหาบัณฑิต  
สาขาวิชา วิศวกรรมคอมพิวเตอร์  
อาจารย์ที่ปรึกษาวิทยานิพนธ์ รศ.ดร.บุญวัฒน์ อัดชู  
อาจารย์ที่ปรึกษาวิทยานิพนธ์ร่วม รศ.ดร.เอื้อน ปิ่นเงิน  
หมายเลขวิทยานิพนธ์ KMITL-2009-EN-M-070-152

คณะกรรมการสอบวิทยานิพนธ์		ลายมือชื่อ
ดร.อรัญญา	วลัยรัชต์	
ผศ.ดร.สมศักดิ์	วลัยรัชต์	
รศ.ดร.ชม	กัมปาน	
ผศ.ดร.เกียรติกุล	เจียรนัยชนะกิจ	
รศ.ดร.บุญวัฒน์	อัดชู	

วัน/เดือน/ปี ที่สอบ วันจันทร์ที่ 13 กรกฎาคม พ.ศ. 2552 เวลา 13.00-15.00 น.

สถานที่สอบ ณ อาคาร A ชั้น 3 ห้องประชุม 2

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

คณะวิศวกรรมศาสตร์ รับรองแล้ว



(รองศาสตราจารย์ ดร.กอบชัย เดชหาญ)

คณบดี คณะวิศวกรรมศาสตร์

วันที่ 13 กรกฎาคม พ.ศ. 2552

หัวข้อวิทยานิพนธ์	การปรับเปลี่ยนรูปแบบการเชื่อมต่อของโครงข่ายประสาทเทียมด้วยอัลกอริธึมมด
นักศึกษา	นายนิกร โภคอุดม
รหัสนักศึกษา	48060707
ปริญญา	วิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชา	วิศวกรรมคอมพิวเตอร์
พ.ศ.	2552
อาจารย์ที่ปรึกษาวิทยานิพนธ์	รศ.ดร.บุญวัฒน์ อัดชู
อาจารย์ที่ปรึกษาวิทยานิพนธ์ร่วม	รศ.ดร.เอื้อน ปิ่นเงิน

### บทคัดย่อ

วิทยานิพนธ์นี้นำเสนอการใช้อัลกอริธึมมด (Ant algorithm) ทำงานร่วมกับการเรียนรู้แบบ Back-Propagation (BP) มาออกแบบหาโครงข่ายประสาทเทียมที่เหมาะสมทั้งรูปแบบการเชื่อมต่อระหว่างนิวรอนและค่าน้ำหนักการเชื่อมภายในไปพร้อมๆ กัน โดยอัลกอริธึมมดจะใช้สำหรับเลือกโครงข่ายที่เหมาะสม ส่วน BP จะใช้ฝึกสอนแบบบางส่วนกับโครงข่ายที่ได้มาจากมดแต่ละตัวด้วยจำนวนรอบน้อยๆ เพื่อปรับค่าน้ำหนักและวัดค่าความเหมาะสมจากค่าความผิดพลาดของโครงข่ายนั้น เพื่อนำไปใช้ในการปรับปรุงค่าพารามิเตอร์ต่างๆ และค่าพีโรโมนในกระบวนการของอัลกอริธึมมดรอบถัดไป วิธีที่นำเสนอนี้ถูกพัฒนาขึ้นเพื่อหลีกเลี่ยงปัญหาซึ่งเกิดจากการฝึกสอนด้วย BP เพียงอย่างเดียวได้แก่การติดกับดักปัญหาจุดต่ำสุดเฉพาะบริเวณ (local minimum) และปัญหาการเข้ากันกับชุดข้อมูลฝึกสอนมากเกินไป (overfitting) สำหรับงานวิจัยนี้ได้เลือกใช้ชุดข้อมูลเกี่ยวกับปัญหาการจำแนกข้อมูล (classification) จำนวน 3 ชุดข้อมูล คือ Cancer1, Diabetes1 และ Glass1 ซึ่งอยู่ในชุดข้อมูลมาตรฐาน PROBEN1 [9] โดยวิธีที่นำเสนอนี้จะถูกนำไปเปรียบเทียบกับวิธีโครงข่ายแบบมาตรฐานที่ฝึกสอนด้วยวิธี BP, วิธีโครงข่ายมาตรฐานฝึกสอนด้วย BP ร่วมกับเทคนิค Early stopping และวิธี ACO-BP ของงานวิจัย [8] ผลการทดลองพบว่าวิธีที่นำเสนอมีผลลัพธ์ที่ดีกว่าในทุกชุดข้อมูล

<b>Thesis Title</b>	Adjusting Artificial Neural Networks Topology Using Ant Algorithm
<b>Student</b>	Mr. Nikorn Pokudom
<b>Student ID.</b>	48060707
<b>Degree</b>	Master of Engineering
<b>Program</b>	Computer Engineering
<b>Year</b>	2009
<b>Thesis Advisor</b>	Assoc. Prof. Dr. Boonwat Attachoo
<b>Thesis Co-Advisor</b>	Assoc. Prof. Dr. Ouen Pinngern

### **ABSTRACT**

This thesis proposed a new method using an Ant algorithm to evolve the feed-forward ANN's structures both connection between neurons and connection weights and partially train the network of ant using BP to adjust parameters and pheromone in an Ant algorithm process. Due to training difficulties in general NN structure, this method could cope with trapping, local minimum and overfitting problems. To evaluate our algorithm, we applied the proposed method to classification problems from PROBEN1 benchmark data set [9] by choosing Cancer1, Diabetes1 and Glass1 for experimentations. The proposed method then was compared with the standard BP neural network, the standard BP neural network with Early stopping techniques and ACO-BP in [8]. The results shown that both precision and efficiency of the proposed method was superior.

## กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงได้อย่างดีเนื่องด้วยได้รับคำแนะนำและคำปรึกษาจาก รศ.ดร.บุญวัฒน์ อัครฐ ซึ่งเป็นอาจารย์ผู้ควบคุมวิทยานิพนธ์ และ รศ.ดร.เอื้อน ปิ่นเงิน อาจารย์ผู้ควบคุมวิทยานิพนธ์ร่วม ข้าพเจ้ารู้สึกทราบบ้างในความกรุณาของอาจารย์ทั้งสองท่านและขอกราบขอบพระคุณเป็นอย่างสูง

ขอกราบขอบพระคุณคณาจารย์ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบังทุกๆ ท่าน ที่ได้ประสิทธิ์ประสาทวิชาให้กับข้าพเจ้า

ขอขอบพระคุณมหาวิทยาลัยอีสเทิร์นเอเซียที่ได้มอบ โอกาสและทุนการศึกษาให้ข้าพเจ้าได้เข้าศึกษาต่อและได้พัฒนาความรู้ความสามารถของข้าพเจ้า จนสำเร็จการศึกษาระดับบัณฑิตศึกษา

ขอขอบคุณที่นักศึกษาระดับ ป.โทและ ป.เอก จากสำนักวิจัยการสื่อสารและเทคโนโลยีสารสนเทศ (ReCCIT) ที่ได้ให้คำแนะนำดีๆ ในการทำงานวิจัย

สุดท้ายนี้ข้าพเจ้าขอกราบขอบพระคุณบิดา มารดา ครอบครัวและขอบคุณภรรยาของข้าพเจ้าที่คอยเป็นกำลังใจและให้การสนับสนุนทางด้านต่างๆ จนทำให้ข้าพเจ้าสามารถทำวิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงด้วยดี

นิกร โภคอุดม

# สารบัญ

	หน้า
บทคัดย่อภาษาไทย.....	I
บทคัดย่อภาษาอังกฤษ.....	II
กิตติกรรมประกาศ.....	III
สารบัญ.....	IV
สารบัญตาราง.....	VII
สารบัญรูป.....	VIII
<b>บทที่ 1 บทนำ.....</b>	<b>1</b>
1.1 ความเป็นมาและความสำคัญของปัญหา.....	1
1.2 ความมุ่งหมายและวัตถุประสงค์ของการศึกษา.....	2
1.3 สมมุติฐานของการศึกษา.....	2
1.4 ทฤษฎีหรือแนวความคิดที่ใช้ในการวิจัย.....	2
1.5 ขอบเขตการวิจัย.....	3
1.6 ขั้นตอนการศึกษา.....	3
1.7 รายละเอียดในแต่ละบท.....	4
<b>บทที่ 2 ทฤษฎีและแนวความคิดที่ใช้ในการวิจัย.....</b>	<b>5</b>
2.1 การพัฒนาค่าถ่วงน้ำหนักของโครงข่ายที่เหมาะสม.....	5
2.2 การพัฒนาหาโครงสร้างของโครงข่ายที่เหมาะสม.....	6
2.3 การพัฒนาโครงสร้างและค่าถ่วงน้ำหนักของโครงข่ายที่เหมาะสมพร้อมกัน.....	7
2.4 การพัฒนาโครงข่ายแบบ Hybrid.....	9
<b>บทที่ 3 โครงข่ายประสาทเทียม (Artificial Neural Networks: ANN).....</b>	<b>13</b>
3.1 หน่วยการคำนวณนิวรอน.....	14
3.2 โครงข่ายประสาทเทียมที่คำนวณแบบเคลื่อนไปข้างหน้า.....	16
3.2.1 โครงข่ายประสาทเทียมแบบชั้นเดียว.....	16
3.2.2 โครงข่ายประสาทเทียมแบบหลายชั้น.....	16
3.3 การเรียนรู้แบบแพร่ย้อนกลับ.....	17
3.3.1 สรุปขั้นตอนการฝึกสอนโครงข่ายด้วยการเรียนรู้แบบแพร่ย้อนกลับ.....	20
3.3.2 การวัดหาค่าความคลาดเคลื่อนของโครงข่าย.....	22

## สารบัญ (ต่อ)

	หน้า
3.3.3 ตัวอย่างของการทำงานของโครงข่าย.....	23
3.4 จุดอ่อนของโครงข่ายประสาทเทียม.....	26
3.4.1 ปัญหาการติดกับดักจุดต่ำสุดเฉพาะบริเวณ (local minimum).....	26
3.4.2 ปัญหาการเข้ากันกับชุดข้อมูลฝึกสอนมากเกินไป (overfitting).....	27
<b>บทที่ 4 อัลกอริทึมมด (Ant Algorithm) .....</b>	<b>32</b>
4.1 ความฉลาดแบบกลุ่ม .....	32
4.2 ปัญหาเกี่ยวกับการหาเส้นทางเดินของพนักงานขาย.....	33
4.3 ระบบมด (Ant System: AS) .....	34
4.4 ระบบอาณานิคมมด (Ant Colony System: ACS) .....	36
4.4.1 การปรับปรุงค่าฟีโรโมนในระดับเฉพาะบริเวณ.....	36
4.4.2 การปรับปรุงค่าฟีโรโมนในระดับครอบคลุมทั้งระบบ.....	36
4.4.3 กระบวนการตัดสินใจเลือกตำแหน่งถัดไป.....	37
4.4.4 รหัสเทียมขั้นตอนการทำงานระบบอาณานิคมมด.....	38
<b>บทที่ 5 การปรับเปลี่ยนรูปแบบการเชื่อมต่อของโครงข่ายประสาทเทียมด้วยอัลกอริทึมมด.....</b>	<b>39</b>
5.1 การออกแบบโครงข่ายประสาทเทียม .....	39
5.1.1 การลบนิวรอนออกจากโครงข่าย.....	40
5.1.2 การคำนวณของโครงข่ายที่ออกแบบ.....	41
5.2 ตารางพารามิเตอร์.....	43
5.3 อัลกอริทึมการปรับเปลี่ยนรูปแบบการเชื่อมต่อของโครงข่ายประสาทเทียมด้วย อัลกอริทึมมด.....	45
<b>บทที่ 6 ผลการทดลอง.....</b>	<b>49</b>
6.1 ชุดข้อมูลที่ใช้ในการทดลอง.....	49
6.2 ข้อกำหนดการทดลอง.....	50
6.3 ผลการทดลอง.....	51
6.3.1 กราฟแสดงค่า MSE ของโครงข่ายในระหว่างการทำงาน.....	51
6.3.2 ผลการทดลองประสิทธิภาพของโครงข่ายประสาทเทียม.....	56

## สารบัญ (ต่อ)

	หน้า
บทที่ 7 สรุปผลการวิจัยและข้อเสนอแนะ.....	60
7.1 สรุปผลการวิจัย.....	60
7.2 ข้อเสนอแนะ.....	61
บรรณานุกรม.....	62
ประวัติผู้เขียน.....	64

# สารบัญตาราง

ตารางที่	หน้า
2.1 ตารางพารามิเตอร์ค่าน้ำหนัก ( $w_i$ ) .....	11
3.1 ความสัมพันธ์ระหว่างแบบจำลองของโครงข่ายประสาททางชีววิทยากับ ANN.....	14
3.2 ชุดข้อมูลฝึกสอนสำหรับปัญหา Exclusive-OR.....	23
5.1 ตัวอย่างค่าพารามิเตอร์ ( $parameter_n$ ) ใน $PTB_n$ .....	44
6.1 ผลการทดลองชุดกับข้อมูล Cancer1.....	57
6.2 ผลการทดลองชุดกับข้อมูล Diabetes1.....	58
6.3 ผลการทดลองชุดกับข้อมูล Glass1 .....	59

# สารบัญรูป

รูปที่	หน้า
2.1 การเข้ารหัสโครโมโซมแบบบิต (ก) ค่าถ่วงน้ำหนักของโครงข่าย (ข) ค่าภายในโครโมโซม .....	5
2.2 การเข้ารหัสโครโมโซมแบบจำนวนจริง (ก) ค่าถ่วงน้ำหนักของโครงข่ายประสาทเทียม (ข) ค่าภายในโครโมโซม .....	6
2.3 โครงข่ายประสาทเทียม (ก) การเชื่อมต่อภายใน (ข) ตารางการเชื่อมต่อระหว่างนิวรอน (ค) ค่าภายในโครโมโซม .....	6
2.4 ตัวอย่างการเข้ารหัสโครโมโซมที่ 1 ของงานวิจัย V. Maniezzo (ก) โครงข่ายนิวรอนการ เชื่อมต่อ (ข) กลุ่มบิตสตริง (ค) ค่าภายในโครโมโซม .....	7
2.5 ตัวอย่างการเข้ารหัสโครโมโซมที่ 2 ของงานวิจัย V. Maniezzo (ก) โครงข่ายนิวรอนการ เชื่อมต่อ (ข) กลุ่มบิตสตริง (ค) ค่าภายในโครโมโซม .....	7
2.6 กลุ่มบิตของ SGTNN (ก) Control genes (ข) โครงข่ายที่ได้จาก control genes (ค) Connection genes .....	8
2.7 โครงสร้างโครโมโซม SGTNN .....	9
2.8 โครงข่ายของ Evolutionary Neural Networks .....	9
2.9 การถอดรหัสโครโมโซมเป็นโครงข่ายของวิธี EA (ก) Genotype (ข) Phenotype .....	10
2.10 การดำเนินการ crossover ของวิธี EA (ก) ก่อน crossover (ข) หลัง crossover .....	10
2.11 การดำเนินการ mutation ของวิธี EA (ก) ก่อน mutation (ข) หลัง mutation .....	10
2.12 ขั้นตอนการทำงานของวิธี ACO-BP.....	12
3.1 โครงข่ายระบบประสาททางชีววิทยา.....	13
3.2 โครงข่ายประสาทเทียม (ANN) .....	14
3.3 หน่วยคำนวณนิวรอน (neuron) .....	15
3.4 ฟังก์ชันการกระตุ้น (Activation Functions) รูปแบบต่างๆ.....	15
3.5 โครงข่ายประสาทเทียมแบบชั้นเดียว (Perceptron) .....	16
3.6 โครงข่ายประสาทเทียมแบบหลายชั้นมี 1 ชั้นซ่อน.....	17
3.7 โครงข่ายประสาทเทียมที่มีจำนวน 3 ชั้น สำหรับการเรียนรู้ปัญหา Exclusive-OR.....	23
3.8 แสดงลักษณะการติดกับดักปัญหา local minimum.....	26
3.9 แสดงลักษณะของ global minimum.....	27
3.10 ลักษณะการเกิด Overfitting ของโครงข่ายเมื่อใช้กับฟังก์ชัน $y = \sin(x/3) + v$ .....	28

# สารบัญรูป (ต่อ)

รูปที่	หน้า
3.11 แสดงค่าความผิดพลาด (Error) ของ Training set และ Test set เมื่อเกิดปัญหา overfitting ของโครงข่ายระหว่างการฝึกสอน.....	29
3.12 จุดการหยุดฝึกสอน โครงข่ายเมื่อใช้เทคนิค Early stopping.....	30
3.13 ปัญหาการแกว่งของค่า validation error เมื่อใช้เทคนิค Early stopping.....	30
4.1 ธรรมชาติการเดินทางของมดโดยใช้เส้นทางที่สั้น.....	32
4.2 ตัวอย่างปัญหา TSP (Eil50) (ก) เส้นทางภายใน (ข) เส้นทางการเดินทางที่สั้นที่สุด.....	33
4.3 กราฟของปัญหา TSP จำนวน 6 เมือง (ก) เส้นทางระหว่างเมืองทั้งหมด (ข) เส้นทางที่มดสามารถเลือกเดินได้ขณะอยู่ที่เมือง $d$ .....	35
4.4 ขั้นตอนการทำงานของระบบอาณานิคมมด .....	38
5.1 ลักษณะ โครงข่ายที่ออกแบบ (ก) นิเวรอนหมายเลข 3 ที่ไม่มีอินพุตเข้ามา (ข) ลบนิเวรอนหมายเลข 3 ออกไป .....	39
5.2 การลบนิเวรอนที่ไม่มีเชื่อมต่อเข้ามา (ก) นิเวรอนหมายเลข 4 ที่ไม่มีเอาต์พุตออกไป (ข) ลบนิเวรอนหมายเลข 4 ออกไป .....	40
5.3 การลบนิเวรอนที่ไม่มีเชื่อมต่อออกไป (ก) นิเวรอนหมายเลข 4 ที่ไม่มีเอาต์พุตออกไป (ข) ลบนิเวรอนหมายเลข 4 ออกไป .....	40
5.4 เส้นทางภายในโครงข่ายประสาทเทียม .....	43
5.5 ตารางพารามิเตอร์ ( $PTB_n$ ) ในแต่ละเส้นทาง .....	44
5.6 อัลกอริธึมการพัฒนาโครงข่ายประสาทเทียมด้วยวิธี ACS .....	45
5.7 การสร้างโครงข่ายประสาทเทียมของมดตัวที่ $k$ (ก) พารามิเตอร์ที่เลือกโดยมดตัวที่ $k$ (ข) โครงข่ายของมดตัวมดตัวที่ $k$ .....	47
6.1 กราฟแสดงค่า MSE ของโครงข่าย ทดสอบกับชุดข้อมูล Cancer1.....	52
6.2 กราฟแสดงค่า MSE ของโครงข่าย ทดสอบกับชุดข้อมูล Diabetes1.....	52
6.3 กราฟแสดงค่า MSE ของโครงข่าย ทดสอบกับชุดข้อมูล Glass1.....	53
6.4 กราฟแสดงค่า MSE ของโครงข่ายที่ได้มาจากวิธีที่นำเสนอทดสอบกับชุดข้อมูล Cancer1.....	54
6.5 กราฟแสดงค่า MSE ของโครงข่ายที่ได้มาจากวิธีที่นำเสนอทดสอบกับชุดข้อมูล Diabetes1.....	54
6.6 กราฟแสดงค่า MSE ของโครงข่ายที่ได้มาจากวิธีที่นำเสนอทดสอบกับชุดข้อมูล Glass1.....	55

# บทที่ 1

## บทนำ

### 1.1 ความเป็นมาและความสำคัญของปัญหา

โครงข่ายประสาทเทียมหรือ Artificial Neural Networks (ANN) เป็นเทคนิคที่ทำให้คอมพิวเตอร์สามารถที่จะเรียนรู้ได้ โดยการเลียนแบบการทำงานของระบบประสาทมนุษย์ซึ่งได้นำลักษณะการทำงานทางชีววิทยาของระบบประสาทมาจำลองเป็นรูปแบบการคำนวณทางคณิตศาสตร์ ในการนำ ANN มาใช้งานก็จะต้องทำการฝึกสอน (Training) ให้กับ ANN เสียก่อนจึงจะสามารถนำไปใช้งานได้ วิธีการหนึ่งที่นิยมนำมาใช้ฝึกสอน ANN คือการเรียนรู้แบบแพร่ย้อนกลับ (Back-Propagation learning: BP)

การฝึกสอนโครงข่ายประสาทเทียมแบบแพร่ย้อนกลับเป็นการปรับปรุงค่าน้ำหนักการเชื่อมต่อระหว่างนิวรอนด้วยวิธี Gradient descent ที่มีเป้าหมายเพื่อลดความผิดพลาดของโครงข่ายในแต่ละรอบการทำงานนั้น พบว่าบ่อยครั้งค่าน้ำหนักที่ได้จากการปรับจะตกอยู่ในช่วงของจุดต่ำสุดเฉพาะบริเวณหรือ local minimum [1,2,3,4] อันเป็นสาเหตุทำให้ไม่สามารถทำการฝึกสอนโครงข่ายได้สำเร็จ ปัญหาดังกล่าวอาจเกิดจากการกำหนดรูปแบบการเชื่อมต่อภายในและค่าน้ำหนักเริ่มต้นที่ไม่เหมาะสมให้โครงข่าย ซึ่งผู้ฝึกสอนมักแก้ไขโดยกำหนดโครงข่ายเริ่มต้นใหม่อีกครั้ง (trial-and-error) และปัญหาอีกอันหนึ่งก็คือ overfitting หมายถึงปัญหาของการเข้ากันกับชุดข้อมูลฝึกสอนมากเกินไป [5,6,7] เป็นปัญหาที่เกิดจากการที่โครงข่ายประสาทเทียมนั้นมีประสิทธิภาพในการเรียนรู้กับชุดข้อมูลฝึกสอนดีมาก ทำให้โครงข่ายที่ได้มานั้นมีความสามารถในการระบุกลุ่มของข้อมูลฝึกสอนได้สูงแต่กลับมีความถูกต้องในการระบุกลุ่มข้อมูลทดสอบต่ำ

งานวิจัยนี้ได้นำเสนอการนำอัลกอริทึมมด (Ant Algorithm) โดยเลือกใช้เทคนิควิถีระบบอาณานิคมมด (Ant Colony System : ACS) ซึ่งเป็นวิธีหนึ่งของอัลกอริทึมมดมาใช้ในการพัฒนาโครงข่ายประสาทเทียม ซึ่งวิธี ACS เป็นขั้นตอนวิธีที่มีลักษณะในการค้นหาคำตอบที่เหมาะสมอีกแบบหนึ่ง โดยมีพื้นฐานมาจากการร่วมมือกันค้นหาแหล่งอาหารของมดภายในฝูง และในการค้นหาเส้นทางจากรังไปยังแหล่งอาหารนั้นจะพยายามหาเส้นทางที่มีระยะทางที่สั้นที่สุดโดยใช้สารเคมีที่เรียกว่า ฟีโรโมน (Pheromone) เป็นข้อมูลในการค้นหา โดยได้ออกแบบให้ ACS ทำหน้าที่ค้นหารูปแบบการเชื่อมต่อภายในและค่าน้ำหนักที่เหมาะสมให้กับโครงข่ายประสาทเทียมไปพร้อมๆ กัน และโครงข่ายที่ออกแบบนั้นนิวรอนภายในสามารถเชื่อมต่อกันข้ามชั้นได้ โดยใช้มดเป็นตัวค้นหาจากโครงข่ายที่ได้กำหนดจำนวนของชั้นซ่อนและจำนวนนิวรอนในแต่ละชั้นซ่อนไว้ และจะใช้การฝึกสอนแบบ BP ทำงานร่วมกันในกระบวนการของ ACS ด้วย สำหรับ BP นั้นจะนำมาช่วยในการสังเกตแนวโน้มความเหมาะสมของโครงข่ายที่กำลังพิจารณาโดยทำการ

ฝึกสอนเบื้องต้นด้วยจำนวนรอบเพียงน้อยๆ แนวโน้มความเหมาะสมของโครงข่ายที่ได้มาจากการฝึกสอนจะถูกนำไปใช้ในกระบวนการปรับค่าพีโรโมนและค่าน้ำหนักใหม่ของ ACS

## 1.2 จุดมุ่งหมายและวัตถุประสงค์ของการศึกษา

วัตถุประสงค์ของวิทยานิพนธ์เพื่อศึกษาการทำงานของ ANN และข้อจำกัดของ ANN ที่ทำการฝึกสอนด้วยการเรียนรู้แบบ BP ซึ่งมีปัญหาเกี่ยวกับการติดกับดักของปัญหา local minimum ทำให้มีการเรียนรู้ที่ไม่สมบูรณ์และปัญหา overfitting ทำให้โครงข่ายที่ได้มีความสามารถในการระบุกลุ่มของข้อมูลฝึกสอนได้สูงทำให้ข้อมูลอื่นๆ ที่ไม่ใช่ข้อมูลฝึกสอนถูกโครงข่ายตีความว่าเป็นข้อมูลรบกวน (noise) จึงทำให้มีความถูกต้องในการระบุกลุ่มข้อมูลทดสอบต่ำ ดังนั้นผู้วิจัยจึงมีแนวคิดที่จะนำอัลกอริธึมมาพัฒนาเพื่อให้ได้โครงข่ายที่มีประสิทธิภาพและสามารถแก้ไขปัญหาดังกล่าวได้

## 1.3 สมมติฐานของการศึกษา

ในปัจจุบันข้อมูลต่างๆ มีความซับซ้อนมากขึ้น เช่นจำนวนคุณลักษณะหรือมิติของข้อมูล จำนวนประเภทของกลุ่มและมีจำนวนข้อมูลในการฝึกสอนมาก ซึ่งบางข้อมูลเป็นคนละกลุ่มกัน แต่กลับมีคุณลักษณะที่ใกล้เคียงกัน ในการจำแนกกลุ่มด้วย ANN นั้นอาจพบปัญหาว่าทำได้ไม่ดี ซึ่งอาจเกิดจากการกำหนดลักษณะของโครงข่ายเริ่มต้นที่ไม่เหมาะสม เช่น รูปแบบการเชื่อมต่อของนิวรอนในโครงข่ายและค่าน้ำหนักการเชื่อมต่อ ดังนั้นการนำวิธี ACS ซึ่งเป็นวิธีที่มีความสามารถในการแก้ปัญหาการหาค่าเหมาะสมจะสามารถพัฒนาหารูปแบบของโครงข่ายที่ดีได้

## 1.4 ทฤษฎีหรือแนวความคิดที่ใช้ในการวิจัย

วิทยานิพนธ์นี้นำเสนอการพัฒนาโครงข่ายประสาทเทียม โดยใช้วิธี ACS ทำหน้าที่ค้นหา รูปแบบการเชื่อมต่อภายในและค่าน้ำหนักที่เหมาะสมให้กับโครงข่ายประสาทเทียมและจะใช้การฝึกสอนแบบ BP ทำงานร่วมกัน สำหรับ BP นั้นจะนำมาช่วยในการสังเกตแนวโน้มความเหมาะสมของโครงข่ายที่กำลังพิจารณาและจะถูกนำไปใช้ในกระบวนการปรับค่าพีโรโมนและค่าน้ำหนักใหม่ในกระบวนการของ ACS ซึ่งวิทยานิพนธ์นี้ได้ศึกษาแนวคิดจากงานวิจัยที่ผ่านมา ดังนี้

- V. Maniezzo [1] ได้ใช้ GA พัฒนาหาโครงข่ายที่เหมาะสมทั้งการเชื่อมต่อระหว่างนิวรอนและค่าน้ำหนักไปพร้อมกันซึ่งนิวรอนนั้นสามารถเชื่อมต่อกันข้ามชั้นได้ การเข้ารหัสปัญหาเป็นค่าบิต 0 และ 1

- K.S. Tang [2] ใช้วิธีการทางพันธุศาสตร์ (genetic algorithms : GA) มาฝึกสอนหาค่าน้ำหนักของโครงข่ายประสาทเทียมที่เหมาะสมแทนการเรียนรู้แบบ BP
- S.J. Han [4] นำเสนอการพัฒนาโครงข่ายประสาทเทียมสำหรับปัญหา anomaly detection โดยออกแบบให้โครงข่ายไม่มีชั้นซ่อนภายในแต่จะกำหนดเป็นกลุ่มของนิวรอนหรือ โหนดซ่อน (hidden nodes) ไว้แทนและโหนดภายในนั้นจะเรียงต่อกันไปโดยโหนดภายในมีเชื่อมต่อแบบไปข้างหน้าและสามารถเชื่อมต่อข้ามโหนดได้แต่จะไม่สามารถเชื่อมต่อย้อนกลับมา โหนดข้างหลังได้ และใช้ GA พัฒนาค้นหารูปแบบการเชื่อมต่อระหว่างโหนดซ่อนภายในและค่าน้ำหนักที่เหมาะสมไปพร้อมๆ กันได้ การเข้ารหัสปัญหาเป็นแบบเมทริกซ์
- Y.P. Liu [8] นำเสนอวิธี ACO-BP โดยเป็นการนำวิธีการหาค่าตอบที่เหมาะสมด้วยอาณานิคมมด (Ant Colony Optimization: ACO) [14,15,16,17] มาค้นหาค่าน้ำหนักการเชื่อมต่อเริ่มต้นที่เหมาะสมของโครงข่ายประสาทเทียมแบบมาตรฐาน (ANN) จากนั้นจะนำโครงข่ายที่ได้ไปฝึกสอนด้วย BP อีกครั้ง

## 1.5 ขอบเขตการวิจัย

ศึกษาและทดสอบประสิทธิภาพของโครงข่ายระหว่างวิธีที่นำเสนอโดยเปรียบเทียบกับวิธี ANN แบบมาตรฐานที่ฝึกสอนด้วยวิธี BP, วิธี ANN แบบมาตรฐานที่ฝึกสอนด้วยวิธี BP ร่วมกับเทคนิค Early stopping และวิธี ACO-BP [8] โดยวัดจากชุดข้อมูลฝึกสอน (training set), ชุดข้อมูลประเมินประสิทธิภาพ (validation set) และชุดข้อมูลทดสอบ (test set) และอัตราความผิดพลาดในการระบุประเภทข้อมูล (Classification Error rate) กับชุดข้อมูลทดสอบ สำหรับงานวิจัยนี้ได้เลือกใช้ชุดข้อมูลเกี่ยวกับปัญหาการจำแนกข้อมูล (classification) จำนวน 3 ชุดข้อมูล คือ Cancer1, Diabetes1 และ Glass1 ซึ่งอยู่ในชุดข้อมูลมาตรฐาน PROBEN1 [9] โดยนำข้อมูลมาจาก <ftp://ftp.ira.uka.de/pub/neuron/proben1.tar.gz>

## 1.6 ขั้นตอนของการศึกษา

วิทยานิพนธ์ฉบับนี้มีขั้นตอนในการศึกษาดังนี้

- 1.6.1 ศึกษากระบวนการทำงานและจุดอ่อนของ ANN
- 1.6.2 ศึกษางานวิจัยที่เกี่ยวข้องกับการปรับปรุงประสิทธิภาพของ ANN
- 1.6.3 ศึกษาการทำงานของอัลกอริทึมมด
- 1.6.4 ศึกษาและออกแบบการนำอัลกอริทึมมดมาใช้พัฒนา ANN
- 1.6.5 ทดลองผลการทำงานของวิธีที่นำมาเปรียบเทียบกับวิธีที่นำเสนอ
- 1.6.6 สรุปผลการทดลองและจัดทำวิทยานิพนธ์

## 1.7 รายละเอียดในแต่ละบท

วิทยานิพนธ์ฉบับนี้ แบ่งเนื้อหาออกเป็น 6 บทคือ

บทที่ 1 กล่าวถึงความเป็นมาของงานวิจัย ความมุ่งหมายและวัตถุประสงค์ สมมติฐาน ทฤษฎีที่ใช้ ขอบเขตของการวิจัยและขั้นตอนการศึกษา

บทที่ 2 กล่าวถึงทฤษฎีพื้นฐานของโครงข่ายประสาทเทียม (Artificial Neural Networks) และปัญหาของโครงข่ายประสาทเทียม

บทที่ 3 กล่าวถึงทฤษฎีของอัลกอริธึมมด (Ant Algorithm)

บทที่ 4 กล่าวถึงหลักการงานของการปรับเปลี่ยนรูปแบบการเชื่อมต่อของโครงข่ายประสาทเทียมด้วยอัลกอริธึมมด

บทที่ 5 กล่าวถึงผลการทดลองกับชุดข้อมูลต่างๆ

บทที่ 6 บทสรุปผลการวิจัยและข้อเสนอแนะ

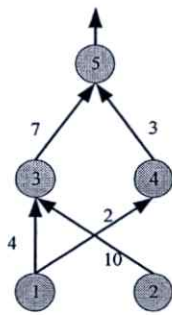
## บทที่ 2

# ทฤษฎีและแนวความคิดที่ใช้ในการวิจัย

วิทยานิพนธ์นี้นำเสนอการพัฒนาโครงข่ายประสาทเทียม โดยใช้วิธี ACS ทำหน้าที่ค้นหา รูปแบบการเชื่อมต่อภายในและค่าถ่วงน้ำหนักที่เหมาะสมให้กับโครงข่ายประสาทเทียมและจะใช้ การเรียนรู้แบบ BP ทำงานร่วมกัน สำหรับ BP นั้นจะนำมาช่วยในการสังเกตแนวโน้มความ เหมาะสมของโครงข่ายที่กำลังพิจารณาและจะถูกนำไปใช้ในกระบวนการปรับค่าพีโร โมนและค่า ถ่วงน้ำหนักใหม่ในกระบวนการของ ACS ซึ่งวิทยานิพนธ์นี้ได้ศึกษาแนวคิดจากงานวิจัยที่ผ่านมา ที่ได้ นำเสนอการแก้ไขปัญหาคัดกับดักปัญหา local minimum และปัญหา overfitting ดังต่อไปนี้

### 2.1 การพัฒนาหาค่าถ่วงน้ำหนักของโครงข่ายที่เหมาะสม

งานวิจัยของ M. Srinivas [10] นำเสนอการใช้วิธีการทางพันธุศาสตร์ (Genetic Algorithm : GA) มาฝึกสอนโครงข่ายประสาทเทียมเพื่อหาค่าถ่วงน้ำหนักของโครงข่ายที่เหมาะสมแทนการใช้ BP จากรูปที่ 2.1 (ก) แสดงถึงลักษณะโครงข่ายที่มี 2 อินพุต มี 1 ชั้นซ่อน 2 นิวรอน และมี 1 เอาต์พุตและรูปที่ 2.1 (ข) แสดงถึงวิธีการเข้ารหัสของปัญหา (encode) ให้อยู่ในรูปแบบของ โครโมโซมค่าของอินจะแสดงเป็นค่าบิต 0 และ 1 ค่าถ่วงน้ำหนักการเชื่อมต่อที่เชื่อมต่อกันในแต่ละ นิวรอนจะแสดงเป็นเลขฐานสองจำนวน 4 บิต ในการถอดรหัสโครโมโซม (decode) จะทำให้อยู่ ในรูปของค่าถ่วงน้ำหนักของโครงข่ายเป็นเลขฐานสิบ เช่น 0100 มีค่าถ่วงน้ำหนักเท่ากับ 4 และ 0000 มีค่าถ่วงน้ำหนักเท่ากับ 0 แสดงว่าไม่มีการเชื่อมต่อระหว่างนิวรอน แต่การเข้ารหัสของวิธีนี้ จะได้ถ่วงน้ำหนักเป็นเลขจำนวนเต็มฐานสิบไม่มีจุดทศนิยม จากนั้นจะวัดค่าผิดพลาดของ โครงข่ายกับชุดข้อมูลฝึกสอนเพื่อใช้กำหนดเป็นค่าความเหมาะสมของโครโมโซมต่อไป



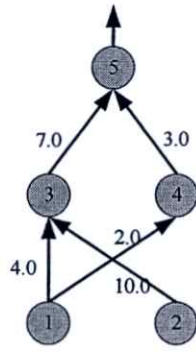
(ก)

0100 1010 0010 0000 0111 0011

(ข)

รูปที่ 2.1 การเข้ารหัสโครโมโซมแบบบิต (ก) ค่าถ่วงน้ำหนักของโครงข่าย (ข) ค่าภายในโครโมโซม (นำมาจาก : [10])

งานวิจัยของ D. Montana และ L. Davis [11] นำเสนอการใช้ genetic algorithm มาฝึกสอนโครงข่ายประสาทเทียมเพื่อหาค่าถ่วงน้ำหนักของโครงข่ายที่เหมาะสมแทนการใช้ BP โดยการเข้ารหัสโครโมโซมนั้นค่าของยีนจะเป็นแทนด้วยค่าจำนวนจริง (real number) ตัวอย่างการเข้ารหัสโครโมโซมแสดงดังรูปที่ 2.2



(ก)

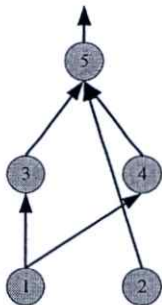
4.0, 10.0, 2.0, 0.0, 7.0, 3.0

(ข)

รูปที่ 2.2 การเข้ารหัสโครโมโซมแบบจำนวนจริง (ก) ค่าถ่วงน้ำหนักของโครงข่ายประสาทเทียม (ข) ค่าภายในโครโมโซม (นำมาจาก : [11])

## 2.2 การพัฒนาหาโครงสร้างของโครงข่ายที่เหมาะสม

โครงข่ายแบบมาตรฐานที่มีการเชื่อมต่อแบบ Fully connection นั้นการเชื่อมต่อของนิวรอนภายในโครงข่ายจะเชื่อมต่อไปยังทุกๆ นิวรอนในชั้นถัดไปเท่านั้นไม่สามารถเชื่อมต่อข้ามชั้นกันได้ งานวิจัยของ D. Whitley [14] จึงได้นำเสนอการใช้ genetic algorithm มาพัฒนาค้นหารูปแบบการเชื่อมต่อที่เหมาะสมของโครงข่าย โดยได้ออกแบบให้นิวรอนสามารถเชื่อมกันข้ามชั้นได้และไม่จำเป็นต้องเชื่อมต่อไปยังทุกๆ นิวรอนในชั้นถัดไป เมื่อได้โครงข่ายที่เหมาะสมแล้วจึงจะนำไปฝึกสอนด้วย BP ต่อไป การแสดงปัญหาและการเข้ารหัสโครโมโซมแสดงดังรูปที่ 2.3



(ก)

	To				
	1	2	3	4	5
From 1	0	0	1	1	0
From 2	0	0	1	0	1
From 3	0	0	0	0	1
From 4	0	0	0	0	1
From 5	0	0	0	0	0

(ข)

0110 101 01 1

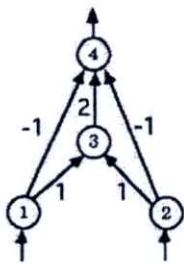
(ค)

รูปที่ 2.3 โครงข่ายประสาทเทียม (ก) การเชื่อมต่อภายใน (ข) ตารางการเชื่อมต่อระหว่างนิวรอน (ค) ค่าภายในโครโมโซม (นำมาจาก : [14])

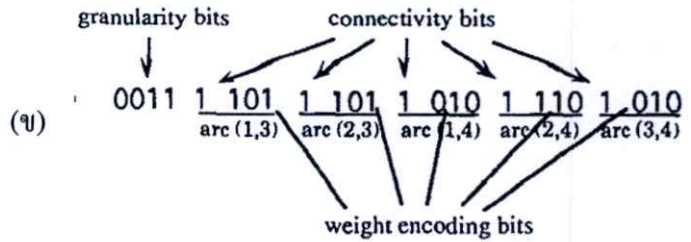
### 2.3 การพัฒนาโครงสร้างและค่าถ่วงน้ำหนักของโครงข่ายที่เหมาะสมพร้อมกัน

งานวิจัยของ V. Maniezzo [1] ได้ใช้ genetic algorithm มาออกแบบโครงข่ายประสาทเทียม โดยทำการพัฒนาค้นหาโครงข่ายที่มีรูปแบบการเชื่อมต่อและค่าถ่วงน้ำหนักที่เหมาะสมไปพร้อมๆ กัน และนิเวรอนภายในโครงข่ายนั้นสามารถเชื่อมต่อข้ามชั้นได้ การเข้ารหัสโครโมโซมเป็นแบบบิตฐานสองภายในโครโมโซมประกอบด้วยบิต 3 ประเภท

ประเภทแรกคือ granularity bit ทำหน้าที่ควบคุมจำนวนบิตของ weight encoding bit เช่น 0011 แปลงเป็นเลขฐานสิบได้ค่า 3 หมายถึงกำหนดให้ weight encoding bit ซึ่งมีจำนวนบิต 3 บิต โดย weight encoding bit คือบิตประเภทที่สองเป็นกลุ่มของบิตที่ทำหน้าที่แสดงค่าถ่วงน้ำหนักการเชื่อมต่อระหว่างนิเวรอนจะทำให้ค่าถ่วงน้ำหนักสามารถกำหนดให้มีค่าที่แตกต่างกันได้ 8 ค่า เช่น มีค่าตั้งแต่ 3 (000) ถึง 4 (111) เป็น ประเภทสุดท้ายคือ connectivity bit แสดงการเชื่อมต่อระหว่างนิเวรอน ( 0 แทนการไม่เชื่อมต่อและ 1 แทนการเชื่อมต่อ) ตัวอย่างการเข้ารหัสปัญหาแสดงดังรูปที่ 2.4 และ 2.5



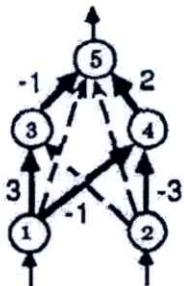
(ก)



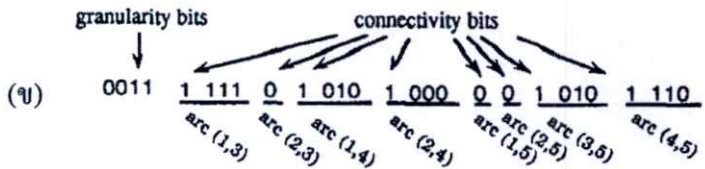
(ค) 001111011101101011101010

รูปที่ 2.4 ตัวอย่างการเข้ารหัสโครโมโซมที่ 1 ของงานวิจัย V. Maniezzo

(ก) โครงข่ายนิเวรอนการเชื่อมต่อ (ข) กลุ่มบิตสตริง (ค) ค่าภายในโครโมโซม (นำมาจาก : [1])



(ก)



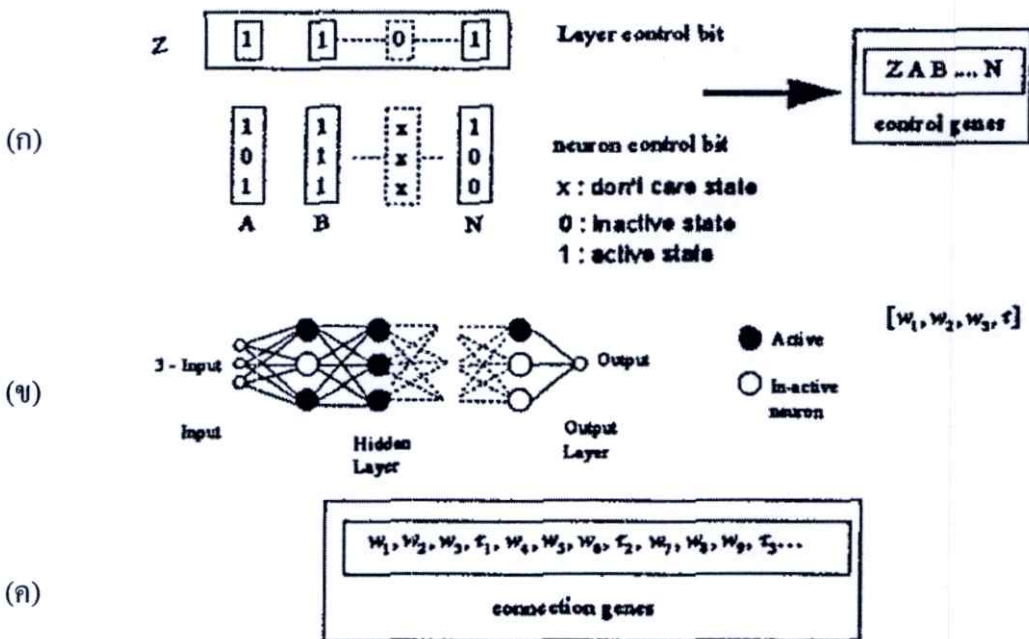
(ค) 001111110101010000010101110

รูปที่ 2.5 ตัวอย่างการเข้ารหัสโครโมโซมที่ 2 ของงานวิจัย V. Maniezzo

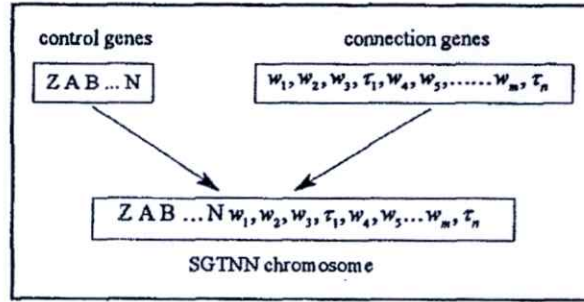
(ก) โครงข่ายนิเวรอนการเชื่อมต่อ (ข) กลุ่มบิตสตริง (ค) ค่าภายในโครโมโซม (นำมาจาก : [1])

งานวิจัยของ K.S. Tang [2] นำเสนอการฝึกสอนโครงข่ายประสาทเทียมแทนการเรียนรู้แบบ BP โดยใช้ Structural Genetic Algorithm (SGA) และเรียกวิธีที่นำเสนอว่า Structural Genetic Trained Neural Network (SGTNN) ซึ่งก่อนการทำงานจะต้องกำหนดจำนวนของชั้นซ่อนและจำนวนของนิวรอนในชั้นซ่อนสูงสุดที่ต้องการไว้ก่อน จากนั้นจึงทำการพัฒนาหาหาลักษณะของโครงสร้างและค่าถ่วงน้ำหนักไปพร้อมๆ กัน งานวิจัยนี้มีกรอบแบบการเข้ารหัสและถอดรหัสโครโมโซมแตกต่างไปจากวิธีข้างต้น ภายในโครโมโซมนั้นประกอบด้วยยีน 2 ชนิด คือ control genes และ connection genes

โดย control genes ประกอบด้วยกลุ่มบิตที่ทำหน้าที่ควบคุมการใช้ชั้นซ่อนแทนด้วย Z และกลุ่มบิตที่ทำหน้าที่ควบคุมการเลือกใช้นิวรอนในแต่ละชั้นซ่อนแทนด้วย A, B, ..., N ค่าของยีนแทนด้วยบิต x (don't care) หมายถึงเป็นนิวรอนของชั้นซ่อนที่ไม่ได้นำมาใช้งาน, 0 (inactive) หมายถึงนิวรอนนี้จะไม่ถูกกระตุ้นให้ทำงานและ 1 (active) หมายถึงนิวรอนนี้จะถูกกระตุ้นให้ทำงานได้ ส่วน connection genes ทำหน้าที่แสดงค่าถ่วงน้ำหนักและค่าไบแอสหรือเทรชโลด (threshold) ของนิวรอนเป็นเลขจำนวนจริง รายละเอียดแสดงดังรูปที่ 2.6 เมื่อนำกลุ่มบิตทั้ง 3 กลุ่มมารวมกันจะได้เป็นโครงสร้างของโครโมโซม SGTNN แสดงดังรูปที่ 2.7



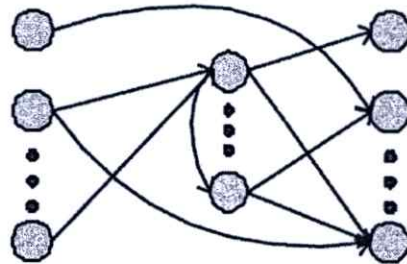
รูปที่ 2.6 กลุ่มบิตของ SGTNN (ก) Control genes (ข) โครงข่ายที่ได้จาก control genes (ค) Connection genes (นำมาจาก : [2])



รูปที่ 2.7 โครงสร้างโครโมโซม SGTNN (นำมาจาก : [2])

## 2.4 การพัฒนาโครงข่ายแบบ Hybrid

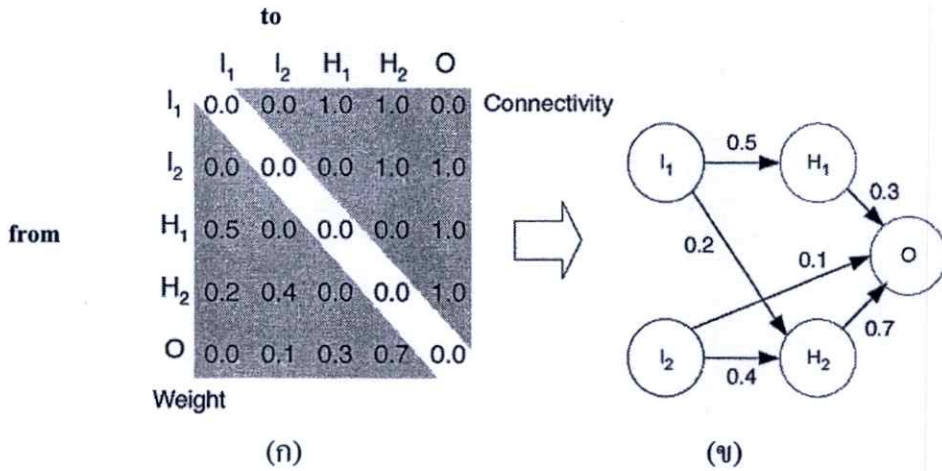
งานวิจัยของ S.J. Han และ S.B. Cho [4] นำเสนอการพัฒนาโครงข่ายประสาทเทียมสำหรับปัญหาการตรวจจับผู้บุกรุกทางเครือข่าย (Intrusion Detection System: IDS) เกี่ยวกับพฤติกรรมการใช้งานที่ผิดปกติ (anomaly detection) โดยใช้ Evolutionary Algorithms (EA) หรือ Genetic Algorithms ทำงานร่วมกับการเรียนรู้ด้วย BP แบบบางส่วน (partial learning) วิธีนี้ออกแบบให้โครงข่ายไม่มีชั้นซ่อนภายในแต่จะกำหนดเป็นกลุ่มของนิวรอนหรือโหนดซ่อน (hidden nodes) ไว้ กลุ่มของนิวรอนในชั้นซ่อนจะเรียงต่อกันไปการเชื่อมต่อของนิวรอนเป็นแบบเชื่อมต่อไปข้างหน้า (forward link) โดยสามารถเชื่อมต่อกันข้ามนิวรอนและสามารถเชื่อมต่อไปยังนิวรอนในชั้นเอาต์พุตได้โดยตรงแต่ไม่สามารถเชื่อมย้อนกลับมาถึงนิวรอนก่อนหน้าได้ ตัวอย่างของโครงข่ายแสดงดังรูปที่ 2.8



Input layer      Hidden layer      Output layer

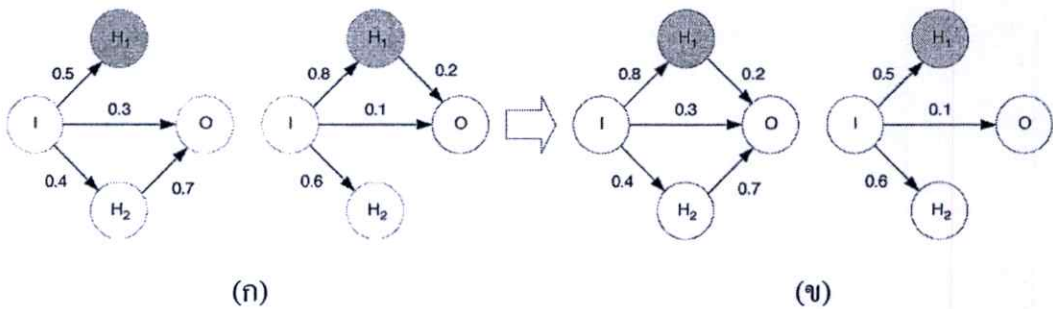
รูปที่ 2.8 โครงข่ายของ Evolutionary Neural Networks (นำมาจาก : [4])

การทำงานจะใช้ EA พัฒนาค้นหารูปแบบการเชื่อมต่อระหว่างนิวรอนซ่อนภายในและค่าถ่วงน้ำหนักที่เหมาะสมไปพร้อมๆ กัน จากนั้นจะนำโครงข่ายที่ได้มาผ่านการเรียนรู้ด้วย BP แบบบางส่วนด้วยจำนวนรอบน้อยๆ เพื่อค้นหาความเหมาะสมของโครงข่ายที่ได้มาโดยวัดจากค่าความผิดพลาดของโครงข่ายและกำหนดเป็นค่าความเหมาะสมของโครโมโซมต่อไป การแสดงปัญหาและการเข้ารหัสโครโมโซมของวิธีนี้จะออกแบบให้ลักษณะของโครโมโซมต่างจากวิธีอื่นๆ ที่ได้กล่าวมาก่อนหน้านี้คือ โครโมโซมจะเป็นแบบเมทริกซ์ (matrix) แสดงดังรูปที่ 2.9

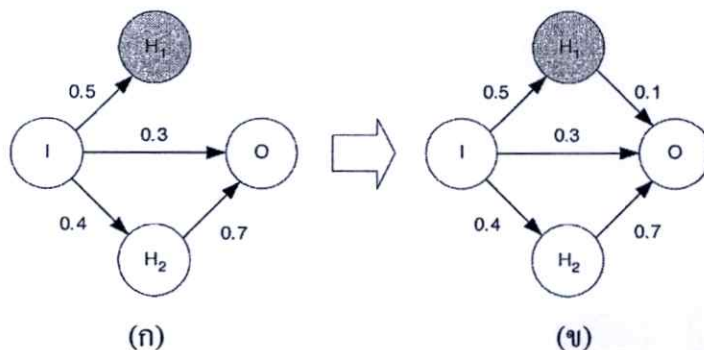


รูปที่ 2.9 การถอดรหัสโครโมโซมเป็นโครงข่ายของวิธี EA (ก) Genotype (ข) Phenotype (นำมาจาก : [4])

รูปที่ 2.9 (ก) เป็นค่าภายในโครโมโซม (genotype) ภายในประกอบด้วยสามเหลี่ยมด้านขวาบนของโครโมโซมทำหน้าที่ระบุเกี่ยวกับการเชื่อมต่อกัน (Connectivity) ระหว่างนิวรอนแบบเชื่อมต่อไปข้างหน้า ค่า “0” คือไม่มีการเชื่อมต่อ ค่า “1” คือมีการเชื่อมต่อ และสามเหลี่ยมด้านซ้ายล่างของโครโมโซมเป็นค่าถ่วงน้ำหนัก (Weight) ระหว่างนิวรอนหากมีการเชื่อมระหว่างกัน ส่วนการดำเนินการ crossover และ mutation ของโครโมโซมแสดงดังรูปที่ 2.10 และ 2.11



รูปที่ 2.10 การดำเนินการ crossover ของวิธี EA (ก) ก่อน crossover (ข) หลัง crossover (นำมาจาก : [4])



รูปที่ 2.11 การดำเนินการ mutation ของวิธี EA (ก) ก่อน mutation (ข) หลัง mutation (นำมาจาก : [4])

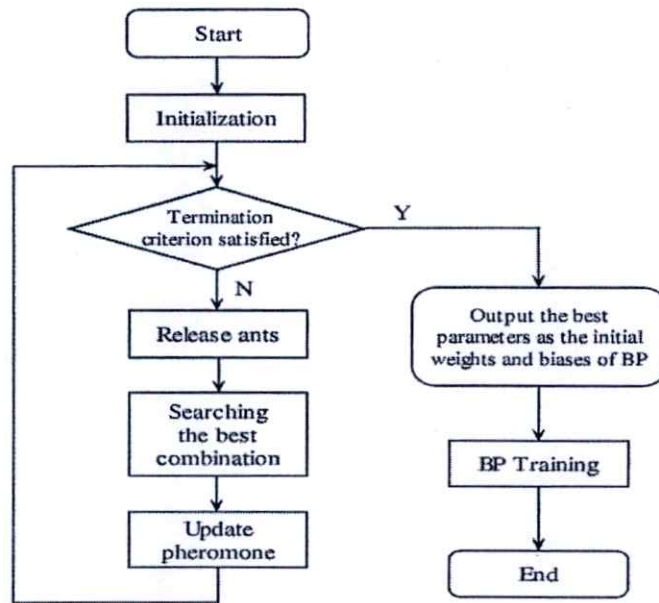
แต่ W. Gao [3] ได้กล่าวไว้ว่าการพัฒนาโครงข่ายประสาทเทียมด้วย GA นั้นที่มีจุดอ่อนที่สำคัญเกี่ยวกับกระบวนการคัดเลือกโครโมโซมพ่อแม่และแม่โครโมโซมใดที่ไม่ได้ถูกคัดเลือกจะต้องถูกคัดทิ้งไปซึ่งบางครั้งโครโมโซมที่ถูกคัดทิ้งอาจจะมีคำตอบที่สามารถนำมาสร้างเป็นโครงข่ายที่เหมาะสมได้ และในรอบการทำงานหลายๆ โครโมโซมส่วนใหญ่จะมีลักษณะของคำตอบหรือค่าของยีนที่คล้ายกันทำให้ GA มักจะไม่สามารถพัฒนาหาคำตอบใหม่ๆ ที่ดีกว่าเดิมได้ และจุดอ่อนเกี่ยวกับการดำเนินการของ GA คือในกระบวนการดำเนินการ crossover เป็นการแลกเปลี่ยนโครงข่ายแบบทางขวาและทางซ้าย (right-and-left) ระหว่าง 2 โครโมโซม ดังนั้นการเปลี่ยนแปลงของโครงข่ายนั้นมีค่อนข้างมากทำให้เกิดความผันผวนต่อประสิทธิภาพของโครงข่ายได้ง่าย

งานวิจัยของ Y.P. Liu [8] ได้นำเสนอวิธี ACO-BP ซึ่งเป็นการนำวิธีการหาคำตอบที่เหมาะสมด้วยอาณานิคมมด (Ant Colony Optimization: ACO) มาพัฒนาค้นหาเฉพาะค่าถ่วงน้ำหนักการเชื่อมต่อเริ่มต้นที่เหมาะสมของโครงข่ายประสาทเทียมแบบมาตรฐาน จากนั้นจะนำโครงข่ายที่ได้ไปฝึกสอนด้วย BP อีกครั้ง วิธีนี้แบ่งให้การเชื่อมต่อระหว่างนิวรอนแต่ละอันมีชุดของค่าถ่วงน้ำหนักที่มีค่าแตกต่างกันอยู่จำนวนหนึ่งแสดงดังตารางที่ 2.1 ตัวแปร  $w_i$  เป็นค่าน้ำหนักของเส้นทางเชื่อมต่อระหว่างนิวรอนในตำแหน่งที่  $i$  โดยจะมีตัวเลือกค่าน้ำหนักอยู่ในช่วง  $[W_{min}, W_{max}]$ ,  $m$  คือค่าแบ่งส่วนจำนวนตัวเลือก เช่น  $m = 59$  แสดงว่ามีจะตัวเลือกพารามิเตอร์ค่าน้ำหนักให้เลือกได้จำนวน 60 ตัวเลือก ( $m+1$ ) ส่วน  $a_i$  แสดงเป็นตัวบอกตำแหน่งที่แบ่งส่วน และ  $\tau_{(i)}$  เป็นค่าฟีโรโมนในตำแหน่ง  $a_i$  ที่แบ่งส่วน

ตารางที่ 2.1 ตารางพารามิเตอร์ค่าน้ำหนัก ( $w_i$ )

	$w_i$			
Tag	1	2	...	$m+1$
Split calibration	$a_1$	$a_2$	...	$a_{m+1}$
Pheromone intensity	$\tau(1)$	$\tau(2)$	...	$\tau(m+1)$

การทำงานจะใช้มดในระบบทำหน้าที่พิจารณาเลือกค่าถ่วงน้ำหนักจากค่าฟีโรโมนในแต่ละการเชื่อมต่อระหว่างนิวรอน จากนั้นจะนำค่าถ่วงน้ำหนักที่ได้มากำหนดให้กับโครงข่ายและนำไปวัดค่าความผิดพลาดกับชุดข้อมูลฝึกสอนเพื่อให้ในกระบวนการปรับค่าฟีโรโมนต่อไป ขั้นตอนการทำงานแสดงดังรูปที่ 2.12



รูปที่ 2.12 ขั้นตอนการทำงานของวิธี ACO-BP (นำมาจาก : [8])

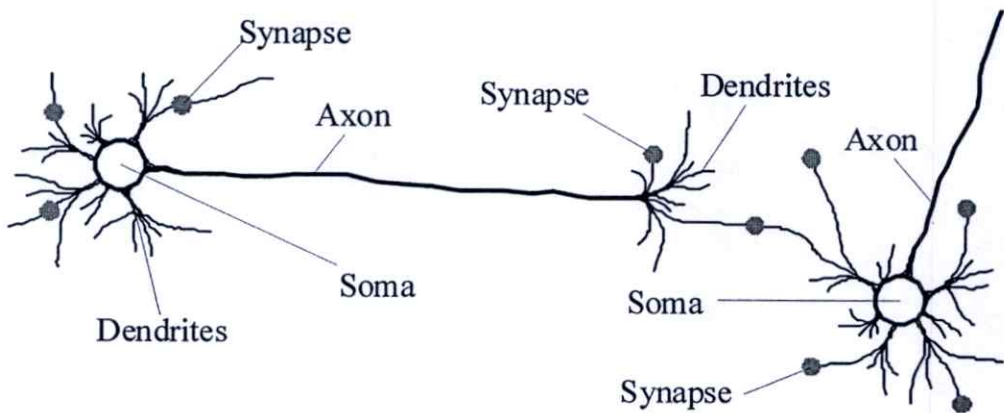
ซึ่งจุดเด่นของ ACO ที่แตกต่างจาก GA นั่นคือ ACO จะไม่มีการตัดคำตอบที่ไม่ดีทิ้งไป ดังนั้นในรอบการทำงานหลายๆ จึงยังคงมีตัวเลือกของคำตอบให้เลือกเท่าเดิม และในรอบถัดไปของวิธี ACO มดแต่ละตัวจะเลือกเส้นทางหรือคำตอบโดยพิจารณาเลือกจากค่าฟีโรโมนของแต่ละเส้นทางซึ่งโดยมากก็จะได้เส้นทางหรือคำตอบที่ไม่ต่างจากเดิมมากนักดังนั้นคำตอบใหม่ๆ ที่ได้มาจึงมีทิศทางที่มุ่งเข้าหาคำตอบได้รวดเร็วกว่าวิธีการ crossover ของ GA

ในบทนี้ได้นำเสนองานวิจัยที่มีผู้เสนอวิธีการพัฒนาให้ได้โครงข่ายประสาทเทียมที่มีประสิทธิภาพที่ดีขึ้นด้วยวิธีการต่างๆ และวิทยานิพนธ์นี้ได้นำวิธีต่างๆ เหล่านี้มาใช้เป็นแนวทางในการพัฒนาโครงข่าย โดยนำวิธีระบบอานาคิมมด (ACS) ซึ่งเป็นวิธีหนึ่งของ ACO มาใช้พัฒนาโครงข่ายดังวิธีของ Y.P. Liu [8] และนำวิธีของ S.J. Han [4] มาใช้ในการออกแบบให้นิวรอนในโครงข่ายสามารถเชื่อมต่อกันข้ามชั้นได้และพัฒนาหาลักษณะการเชื่อมต่อและค่าน้ำหนักไปพร้อมๆ กัน แต่จะออกแบบให้โครงข่ายนั้นมีชั้นซ่อนภายในดังวิธีของ K.S. Tang [2] และนำการเรียนรู้ด้วย BP แบบบางส่วน (partial learning) ของวิธี S.J. Han [4] มาใช้ในส่วนของการพัฒนาโครงข่ายและการปรับพารามิเตอร์ในกระบวนการของวิธีระบบอานาคิมมด (ACS) ด้วย

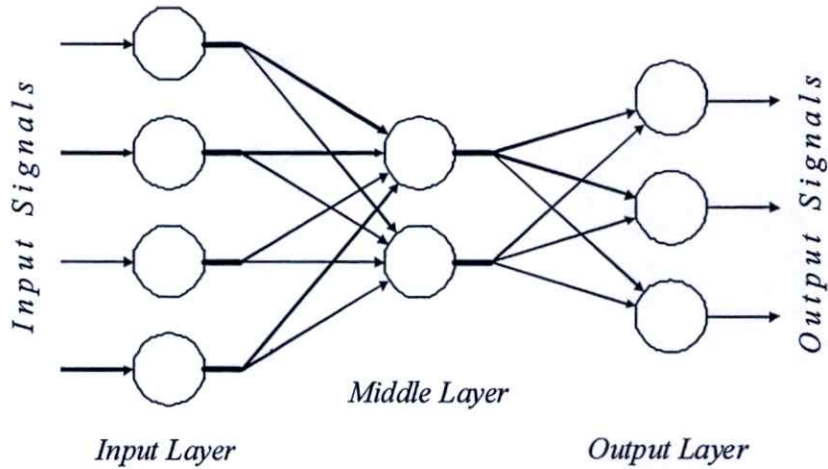
## โครงข่ายประสาทเทียม (Artificial Neural Networks)

โครงข่ายประสาทเทียม (Artificial Neural Networks: ANN) [14] เป็นแนวคิดที่จำลองลักษณะการทำงานของระบบสมองทางชีววิทยาของมนุษย์มาพัฒนาให้อยู่ในรูปแบบการคำนวณทางคณิตศาสตร์ ซึ่ง ANN ได้รับความนิมอย่างมากในงานทางด้านปัญญาประดิษฐ์ (Artificial Intelligent: AI) ซึ่งมีความสามารถในการเรียนรู้จากชุดข้อมูลฝึกสอนได้ จึงนิยมนำ ANN ประยุกต์ใช้กับงานในด้านต่างๆ เช่น การเรียนรู้การจดจำรูปแบบ (Pattern Recognition), การอุปมาความรู้ (Knowledge Deduction), การวิเคราะห์วินิจัยโรค (Diagnostics), การทำนายเหตุการณ์ (Forecasting) และการจำแนกกลุ่มข้อมูล (Classification) เป็นต้น

โครงสร้างของ ANN จะประกอบด้วยหน่วยประมวลผล (the processing elements) ที่เรียกว่า “นิวรอน” (Neuron) ทำหน้าที่เหมือนกับตัวเซลล์ (Soma) ในระบบประสาทของมนุษย์ โดยแต่ละนิวรอนจะเชื่อมต่ออยู่กับนิวรอนอื่นๆ โดยนิวรอนจะมีส่วนปลายรับกระแสประสาทที่เรียกว่า “เดนไดรต์” (Dendrite) ทำหน้าที่รับสัญญาณนำเข้าหรืออินพุต (input) ซึ่งส่งมาจากนิวรอนอื่นเพื่อนำเข้ามาประมวลผลภายในและเมื่อประมวลผลเสร็จสิ้นก็จะส่งสัญญาณผลลัพธ์หรือเอาต์พุต (output) ออกมาจากส่วนปลายส่งกระแสประสาทที่เรียกว่า “แอกซอน” (Axon) แล้วส่งเอาต์พุตนั้นผ่านส่วนที่เรียกว่า “ไซแนปส์” (Synapses) ซึ่งเป็นจุดประสานระหว่างนิวรอนและทำหน้าที่เป็นตัวเก็บความรู้ของสมอง ซึ่งนิวรอนอื่นที่เชื่อมต่ออยู่ด้วยจะรับค่านั้นเป็นอินพุตผ่านทางเดนไดรต์ต่อไป ซึ่งรูปที่ 3.1 แสดงถึงโครงข่ายทางชีววิทยาและรูปที่ 3.2 แสดงถึงโครงข่ายประสาทเทียมที่จำลองขึ้นมา



รูปที่ 3.1 โครงข่ายระบบประสาททางชีววิทยา (นำมาจาก : [14])



รูปที่ 3.2 โครงข่ายประสาทเทียม (ANN) (นำมาจาก : [14])

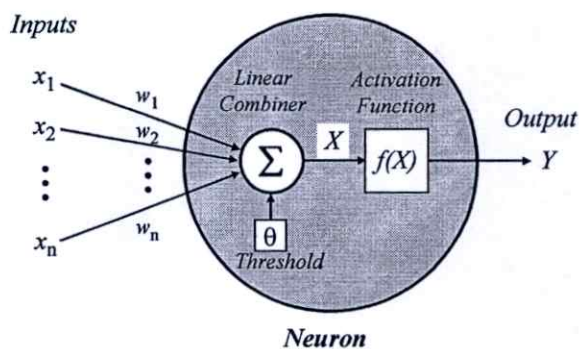
โดยความสัมพันธ์ระหว่างโครงข่ายประสาททางชีววิทยากับโครงข่ายประสาทเทียม (ANN) ที่จำลองขึ้นมาสามารถแสดงได้ดังตารางที่ 3.1

ตารางที่ 3.1 ความสัมพันธ์ระหว่างแบบจำลองของโครงข่ายประสาททางชีววิทยากับ ANN

<i>Biological Neural Network</i>	<i>Artificial Neural Network</i>
โซมา (Soma)	นิวรอน (Neuron)
เดนไดรต์ (Dendrite)	ตัวแปรนำเข้า (Input)
แอกซอน (Axon)	ตัวแปรส่งออก (Output)
ไซแนปส์ (Synapse)	ค่าถ่วงน้ำหนัก (Weight)

### 3.1 หน่วยการคำนวณนิวรอน (neuron)

โครงข่ายประสาทจะประกอบไปด้วยหน่วยการคำนวณทางคณิตศาสตร์ซึ่งเรียกว่า นิวรอน (neuron) หรือ โหนด (node) แสดงดังรูปที่ 3.3 โดยนิวรอนแต่ละตัวนั้นจะรับสัญญาณหรือข้อมูลอินพุตเข้ามา ซึ่งสามารถรับมาได้ทั้งจากภายนอกโครงข่ายหรือรับมาจากค่าเอาต์พุตของนิวรอนอื่นที่ได้ส่งผ่านมาทางไซแนปส์หรือค่าถ่วงน้ำหนัก (weight) จากนั้นนิวรอนจะคำนวณค่า “สัญญาณอินพุตสุทธิ” (net weight) ที่ส่งมายังนิวรอน ซึ่งคำนวณจากผลรวมของผลคูณระหว่างอินพุตแต่ละอันกับค่าถ่วงน้ำหนักที่เชื่อมต่อมายังนิวรอนและเปรียบเทียบกับค่า “เทรชโฮลด์” (threshold:  $\theta$ ) ของแต่ละนิวรอน แสดงดังสมการที่ (3.1) จากนั้นนำค่าสัญญาณอินพุตสุทธิไปผ่าน “ฟังก์ชันการกระตุ้น” (Activation function) เพื่อแปลงเป็นค่าเอาต์พุตของนิวรอน แสดงดังสมการที่ (3.2) และตัวอย่างของฟังก์ชันกระตุ้นที่นิยมใช้แสดงดังรูปที่ 3.4



รูปที่ 3.3 หน่วยคำนวณนิวรอน (neuron)

$$X = \sum_{i=1}^n x_i w_i - \theta \quad (3.1)$$

$$Y = f(X) \quad (3.2)$$

โดยที่  $X$  หมายถึงค่าสัญญาณอินพุตสุทธิ (net weight) ที่เข้ามายังนิวรอน  
 $x_i$  หมายถึงค่าอินพุตลำดับที่  $i$   
 $w_i$  หมายถึงค่าถ่วงน้ำหนักลำดับที่  $i$   
 $n$  หมายถึงจำนวนของนิวรอนอื่นที่เชื่อมต่อเข้ามา  
 $Y$  หมายถึงค่าเอาต์พุตของนิวรอน

Step function	Sign function	Sigmoid function	Linear function
$Y^{step} = \begin{cases} 1, & \text{if } X \geq 0 \\ 0, & \text{if } X < 0 \end{cases}$	$Y^{sign} = \begin{cases} +1, & \text{if } X \geq 0 \\ -1, & \text{if } X < 0 \end{cases}$	$Y^{sigmoid} = \frac{1}{1 + e^{-X}}$	$Y^{linear} = X$

รูปที่ 3.4 ฟังก์ชันการกระตุ้น (Activation Functions) รูปแบบต่างๆ (นำมาจาก : [14])

ซึ่งฟังก์ชัน step และ sign เป็นฟังก์ชันแบบ hard limit มักจะถูกใช้ในงานเกี่ยวกับการตัดสินใจในงานจำแนกกลุ่มข้อมูล (classification) และงานรู้จำรูปแบบ (pattern recognition)

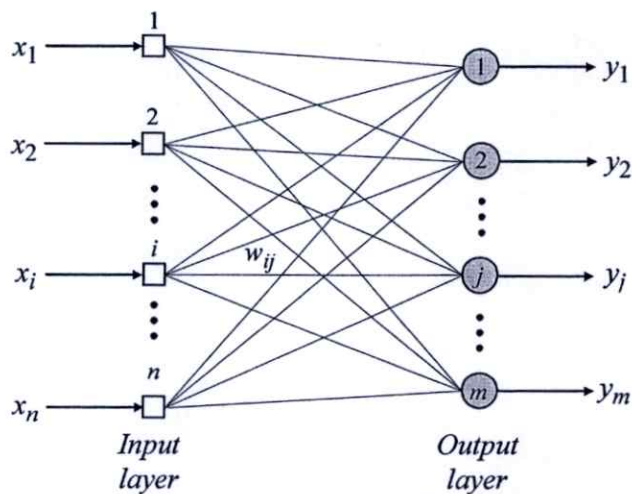
ฟังก์ชันซิกมอยด์ (sigmoid) เป็นฟังก์ชันที่นิยมใช้มากที่สุดสำหรับโครงข่ายที่มีการเรียนรู้แบบแพร่ย้อนกลับ (backpropagation learning) ค่าเอาต์พุตของนิวรอนมีค่าอยู่ระหว่าง 0 และ 1 ฟังก์ชัน linear ค่าเอาต์พุตของฟังก์ชันนี้จะมีค่าเดียวกับค่าผลรวมอินพุตสุทธิที่เข้ามา

### 3.2 โครงข่ายประสาทเทียมที่คำนวณแบบเคลื่อนไปข้างหน้า (Feed-forward Neural Network)

โครงข่ายประสาทเทียมที่มีการคำนวณแบบเคลื่อนไปข้างหน้าหรือ Feed-forward Neural Networks เป็นโครงข่ายที่มีการทำงานโดยมีการคำนวณไปที่ละชั้นเริ่มต้นแต่นิวรอนในชั้นอินพุตเคลื่อนต่อไปยังชั้นถัดไปจนถึงชั้นเอาต์พุต ตัวอย่างของโครงข่ายแบบนี้ได้แก่ โครงข่ายประสาทเทียมแบบชั้นเดียว (Single-Layer Neural Network) หรือเรียกว่าเพอร์เซพตรอน (Perceptron) และโครงข่ายประสาทเทียมแบบหลายชั้น (Multilayer Neural Network)

#### 3.2.1 โครงข่ายประสาทเทียมแบบชั้นเดียว (Perceptron)

เป็นโครงข่ายที่ประกอบไปด้วยนิวรอนที่เชื่อมต่อกันอยู่โดยเรียงตัวเป็นชั้น ประกอบด้วย 2 ชั้นคือชั้นข้อมูลนำเข้า (input layer) และชั้นข้อมูลออก (output layer) แสดงดังรูปที่ 3.5



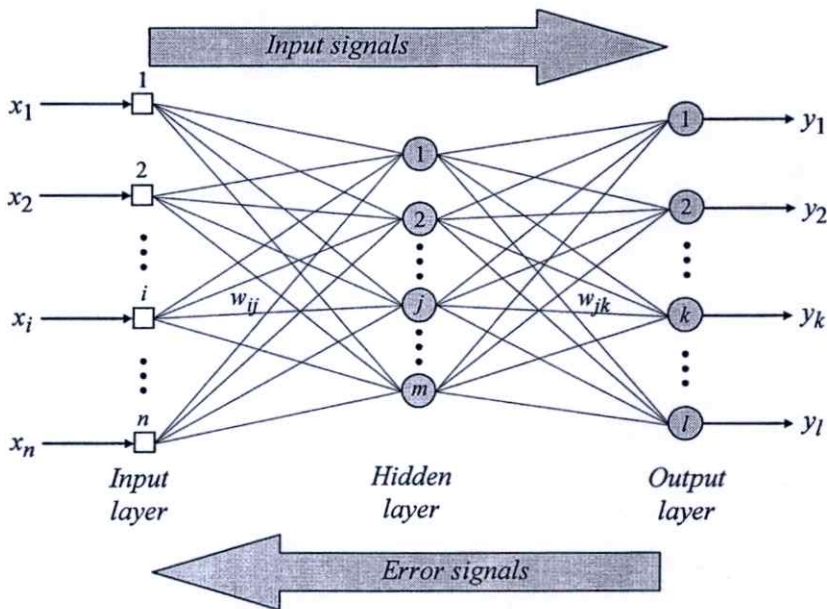
รูปที่ 3.5 โครงข่ายประสาทเทียมแบบชั้นเดียว (Perceptron)

#### 3.2.2 โครงข่ายประสาทเทียมแบบหลายชั้น (Multi-Layer Perceptron neural networks : MLP)

เป็นโครงข่ายที่ประกอบไปด้วยนิวรอนที่เชื่อมต่อกันอยู่โดยเรียงตัวเป็นชั้นในลักษณะเคลื่อนไปข้างหน้า (Feedforward) ประกอบด้วยชั้นข้อมูลนำเข้าหรือชั้นอินพุต (input layer), ชั้นซ่อน (hidden layer) และชั้นข้อมูลออกหรือชั้นเอาต์พุต (output layer) ซึ่งชั้นซ่อนนั้นสามารถมี

ได้มากกว่า 1 ชั้น การเชื่อมต่อกันของนิวรอนระหว่างชั้นเป็นการเชื่อมต่อกันอย่างทั่วถึง (fully connected) นั่นคือนิวรอนทุกอันในชั้นก่อนหน้าจะเชื่อมต่อไปยังทุกนิวรอนที่อยู่ในชั้นถัดไปและจะไม่มีการเชื่อมต่อแบบข้ามชั้น โดยข้อมูลอินพุตจะถูกส่งมาให้กับนิวรอนในชั้นอินพุตจากนั้นจะคำนวณค่าเอาต์พุตและส่งออกไปยังนิวรอนในชั้นซ่อนและจะมีการส่งต่อไปยังนิวรอนในชั้นถัดๆ ไปจนกระทั่งถึงชั้นเอาต์พุต แผนภาพของโครงข่ายแสดงดังรูปที่ 3.6

โครงข่าย MLP ได้รับความนิยมมากเนื่องจากสามารถเรียนรู้กับข้อมูลที่มีมิติสูงหรือมีความซับซ้อนได้ดี โดยกระบวนการเรียนรู้ของโครงข่ายคือการปรับค่าถ่วงน้ำหนักที่มีการเชื่อมต่อกันระหว่างนิวรอน วิธีหนึ่งที่ได้รับนิยมนิยมเป็นอย่างมากคือ “การเรียนรู้แบบแพร่ย้อนกลับ” (Back-propagation learning)



รูปที่ 3.6 โครงข่ายประสาทเทียมแบบหลายชั้นมี 1 ชั้นซ่อน (นำมาจาก : [14])

### 3.3 การเรียนรู้แบบแพร่ย้อนกลับ (Back-propagation learning)

วิธีการเรียนรู้แบบแพร่ย้อนกลับ (Back-propagation หรือ BP) เป็นการเรียนรู้แบบมีผู้สอน (supervised learning) มีวิธีการปรับค่าถ่วงน้ำหนักที่เหมาะสมที่เรียกว่า Gradient descent ซึ่งจะทำการปรับค่าถ่วงน้ำหนักเพื่อทำให้ค่าความผิดพลาดของโครงข่ายเมื่อวัดกับชุดข้อมูลที่ใช้ฝึกสอนมีค่าลดลงไปในแต่ละรอบของการคำนวณ โดยสัญญาณอินพุต (input signals) หรือข้อมูลอินพุตของข้อมูลฝึกสอนจะถูกส่งต่อแบบเคลื่อนไปข้างหน้าตามลำดับชั้นจนถึงชั้นเอาต์พุต จากนั้นค่าเอาต์พุตที่ได้มาจากการคำนวณ (actual output) จะถูกนำมาเปรียบเทียบกับค่าเอาต์พุตเป้าหมาย (desired output) เพื่อหาค่าความคลาดเคลื่อนที่เกิดขึ้น ค่าความคลาดเคลื่อนที่ได้มาจะนำไปใช้สำหรับการปรับค่าถ่วงน้ำหนัก โดยแพร่ค่าในทิศทางย้อนกลับไปยังชั้นซ่อนจนไปถึงชั้นอินพุต

โครงข่าย MLP รูปที่ 3.5 ที่มี 1 ชั้นซ่อน สามารถนำมาฝึกสอนได้โดยกำหนดให้  $i, j$  และ  $k$  เป็นลำดับของนิวรอนในชั้นอินพุต, ชั้นซ่อนและชั้นเอาต์พุตตามลำดับ สัญญาณอินพุต  $x_1, x_2, \dots, x_n$  จะถูกนำเข้าไปในโครงข่ายโดยแพร่จากทางซ้ายไปทางขวา ส่วนค่าสัญญาณผิดพลาด (error signal)  $e_1, e_2, \dots, e_l$  ของนิวรอนในชั้นเอาต์พุตจะถูกแพร่ย้อนกลับเข้ามาในโครงข่ายจากทางขวาไปทางซ้าย,  $w_{ij}$  คือค่าถ่วงน้ำหนักของการเชื่อมต่อระหว่างนิวรอน  $i$  ในชั้นอินพุตกับนิวรอน  $j$  และ  $w_{jk}$  คือค่าถ่วงน้ำหนักของการเชื่อมต่อระหว่างนิวรอน  $j$  ในชั้นซ่อนกับนิวรอน  $k$  ในชั้นเอาต์พุต

ในการทำงานของ BP นั้นจะนำค่าสัญญาณผิดพลาดของชั้นเอาต์พุตทำงานแพร่ย้อนกลับเข้ามายังชั้นซ่อน การคำนวณค่าสัญญาณผิดพลาดของนิวรอน  $k$  ที่อยู่ในชั้นเอาต์พุตในรอบการทำงาน (epoch) ที่  $p$  คำนวณได้จากสมการที่ (3.3)

$$e_k(p) = y_{d,k}(p) - y_k(p) \quad (3.3)$$

โดยที่  $e_k(p)$  หมายถึงค่าสัญญาณผิดพลาดของนิวรอน  $k$  รอบการทำงานที่  $p$

$y_{d,k}(p)$  หมายถึงค่าเอาต์พุตเป้าหมาย

$y_k(p)$  หมายถึงค่าเอาต์พุตที่ได้จากการคำนวณของนิวรอน  $k$

การปรับปรุงค่าถ่วงน้ำหนักระหว่างการเชื่อมต่อของนิวรอน  $j$  ที่ได้เชื่อมต่อมายังนิวรอน  $k$  ที่อยู่ในชั้นเอาต์พุตสำหรับรอบการทำงานถัดไปซึ่งคำนวณได้จากสมการที่ (3.4)

$$w_{jk}(p+1) = w_{jk}(p) + \Delta w_{jk}(p) \quad (3.4)$$

โดย  $\Delta w_{jk}(p)$  หมายถึงค่าความเปลี่ยนแปลงของค่าถ่วงน้ำหนัก (weight correction) ของนิวรอน  $j$  ที่ได้เชื่อมต่อมายังนิวรอน  $k$  คำนวณได้จากสมการที่ (3.5)

$$\Delta w_{jk}(p) = \alpha \times y_j(p) \times \delta_k(p) \quad (3.5)$$

โดย  $\alpha$  หมายถึงอัตราการเรียนรู้ (learning rate) ของโครงข่าย

$\delta_k(p)$  หมายถึงค่าความผิดพลาดเกรเดียนต์ (error gradient) ของนิวรอน  $k$  ในรอบการทำงานที่  $p$  ซึ่ง  $\delta_k(p)$  คำนวณได้จากสมการที่ (3.6)

$$\delta_k(p) = y_k(p) \times [1 - y_k(p)] \times e_k(p) \quad (3.6)$$

โดย  $y_k(p)$  หมายถึงค่าเอาต์พุตที่ได้จากการคำนวณของนิวรอน  $k$  ในรอบการทำงานที่  $p$  หากกำหนดให้ใช้ฟังก์ชันการกระตุ้นเป็นแบบฟังก์ชัน sigmoid จะคำนวณดังสมการที่ (3.7)

$$y_k(p) = \frac{1}{1 + e^{-X_k(p)}} \quad (3.7)$$

ซึ่ง  $X_k(p)$  หมายถึงค่าสัญญาณอินพุตสุทธิที่เข้ามายังนิวรอน  $k$  คำนวณได้จากสมการที่ (3.8)

$$X_k(p) = \sum_{j=1}^m x_j(p) \times w_{jk}(p) - \theta_k \quad (3.8)$$

การปรับปรุงค่า threshold ของนิวรอน  $k$  ที่อยู่ในชั้นเอาต์พุตสำหรับรอบการทำงานถัดไป ซึ่งคำนวณได้จากสมการที่ (3.9)

$$\theta_k(p+1) = \theta_k(p) + \Delta\theta_k(p) \quad (3.9)$$

โดย  $\Delta\theta_k(p)$  หมายถึงค่าความเปลี่ยนแปลงของค่าเทรลโฮลด์ของนิวรอน  $k$  คำนวณได้จากสมการที่ (3.10)

$$\Delta\theta_k(p) = \alpha \times (-1) \times \delta_k(p) \quad (3.10)$$

โดย -1 เป็นค่าอินพุตไบแอส (bias input) ที่ผ่าน  $\theta_k(p)$  มายังนิวรอน  $k$  ในชั้นเอาต์พุต

การปรับปรุงค่าถ่วงน้ำหนักระหว่างการเชื่อมต่อของนิวรอน  $i$  ในชั้นอินพุตที่ได้เชื่อมต่อกับมายังนิวรอน  $j$  ที่อยู่ในชั้นซ่อนสำหรับรอบการทำงานถัดไปซึ่งคำนวณได้จากสมการที่ (3.11)

$$w_{ij}(p+1) = w_{ij}(p) + \Delta w_{ij}(p) \quad (3.11)$$

โดย  $\Delta w_{ij}(p)$  หมายถึงค่าความเปลี่ยนแปลงของค่าถ่วงน้ำหนัก (weight correction) ของนิวรอน  $i$  ที่ได้เชื่อมต่อกับมายังนิวรอน  $j$  คำนวณได้จากสมการที่ (3.12)

$$\Delta w_{ij}(p) = \alpha \times x_i(p) \times \delta_j(p) \quad (3.12)$$

โดย  $\delta_j(p)$  หมายถึงค่าความผิดพลาดเกรเดียนต์ (error gradient) ของนิวรอน  $j$  ในรอบการทำงานที่  $p$  ซึ่งคำนวณได้จากสมการที่ (3.13)

$$\delta_j(p) = y_j(p) \times [1 - y_j(p)] \times \sum_{k=1}^l \delta_k(p) w_{jk}(p) \quad (3.13)$$

โดย  $y_j(p)$  หมายถึงค่าเอาต์พุตที่ได้จากการคำนวณของนิวรอน  $j$  ในชั้นซ่อน ในรอบการทำงานที่  $p$  หากกำหนดให้ใช้ฟังก์ชันการกระตุ้นเป็นแบบฟังก์ชัน sigmoid จะคำนวณดังสมการที่ (3.14)

$$y_j(p) = \frac{1}{1 + e^{-X_j(p)}} \quad (3.14)$$

ซึ่ง  $X_j(p)$  หมายถึงค่าสัญญาณอินพุตสุทธิที่เข้ามายังนิวรอน  $j$  คำนวณได้จากสมการที่ (3.15)

$$X_j(p) = \sum_{i=1}^n x_i(p) \times w_{ij}(p) - \theta_j \quad (3.15)$$

โดย  $n$  หมายถึงจำนวนของนิวรอนที่อยู่ในชั้นอินพุต

การปรับปรุงค่า threshold ของนิวรอน  $j$  ที่อยู่ในชั้นซ่อนสำหรับรอบการทำงานถัดไป คำนวณได้จากสมการที่ (3.16)

$$\theta_j(p+1) = \theta_j(p) + \Delta\theta_j(p) \quad (3.16)$$

โดย  $\Delta\theta_j(p)$  หมายถึงค่าความเปลี่ยนแปลงของค่า threshold ของนิวรอน  $j$  คำนวณได้จากสมการที่ (3.17)

$$\Delta\theta_j(p) = \alpha \times (-1) \times \delta_j(p) \quad (3.17)$$

โดย  $-1$  เป็นค่าอินพุตไบแอส (bias input) ที่ผ่าน  $\theta_j(p)$  มายังนิวรอน  $j$  ในชั้นอินพุต

### 3.3.1 สรุปขั้นตอนการฝึกสอนโครงข่ายด้วยการเรียนรู้แบบแพร่ย้อนกลับ

จากวิธีการเรียนรู้แบบแพร่ย้อนกลับที่ใช้ฝึกสอนโครงข่ายที่ได้กล่าวมานั้น สามารถสรุปเป็นขั้นตอนการทำงานได้ดังนี้

**ขั้นตอนที่ 1 :** การกำหนดค่าเริ่มต้นของโครงข่าย

สุ่มค่าถ่วงน้ำหนักเริ่มต้นและค่า threshold ให้กับโครงข่าย โดยเป็นค่าน้อยๆ

**ขั้นตอนที่ 2 :** การกระตุ้น (Activation)

ในการกระตุ้นโครงข่ายแบบ BP นั้นจะมีการนำของค่าสัญญาณอินพุต  $x_1(p), x_2(p), \dots, x_n(p)$  และค่าเอาต์พุตเป้าหมาย  $y_{d,1}(p), y_{d,2}(p), \dots, y_{d,l}(p)$  ของแต่ละนิวรอนในชั้นเอาต์พุตมาใช้ และต้องกำหนดประเภทของฟังก์ชันกระตุ้นที่จะนำมาใช้กับนิวรอนในโครงข่าย เช่นฟังก์ชัน sigmoid

(ก) คำนวณค่าเอาต์พุตของนิวรอนภายในชั้นซ่อน

$$y_j(p) = \text{sigmoid} \left[ \sum_{i=1}^n x_i(p) \times w_{ij}(p) - \theta_j \right]$$

โดย  $n$  คือจำนวนของอินพุตที่เข้ามายังนิวรอน  $j$  ในชั้นซ่อนและ *sigmoid* คือประเภทของฟังก์ชันกระตุ้นที่เลือกใช้กับนิวรอนในชั้นซ่อน

(ข) คำนวณค่าเอาต์พุตของนิวรอนภายในชั้นเอาต์พุต

$$y_k(p) = \text{sigmoid} \left[ \sum_{j=1}^m x_{jk}(p) \times w_{jk}(p) - \theta_k \right]$$

โดย  $m$  คือจำนวนของอินพุตที่เข้ามายังนิวรอน  $k$  ในชั้นเอาต์พุตและ  $\text{sigmoid}$  คือประเภทของฟังก์ชันกระตุ้นที่เลือกใช้กับนิวรอนในชั้นเอาต์พุต

**ขั้นตอนที่ 3 :** การปรับปรุ้ค่าน้ำหนักการเชื่อมต่อ (Weight training) และค่า threshold

สำหรับการปรับปรุ้ค่าน้ำหนักการเชื่อมต่อ (Weight training) และค่า threshold ของโครงข่ายแบบ BP นั้นจะทำการแพร่ค่าย้อนกลับโดยเริ่มจากนิวรอนในชั้นเอาต์พุตไปจนถึงนิวรอนในชั้นอินพุต

(ก) การปรับปรุ้ค่าถ่วงน้ำหนักและค่า threshold ในชั้นเอาต์พุต

- คำนวณค่า error gradient ของนิวรอน  $k$  ภายในชั้นซ่อนของรอบการทำงานที่  $p$

$$\delta_k(p) = y_k(p) \times [1 - y_k(p)] \times e_k(p)$$

$$\text{ซึ่ง } e_k(p) = y_{d,k}(p) - y_k(p)$$

- คำนวณค่าความเปลี่ยนแปลงของค่าถ่วงน้ำหนัก (weight correction) ของนิวรอน  $j$  ที่ได้เชื่อมต่อมายังนิวรอน  $k$

$$\Delta w_{jk}(p) = \alpha \times y_j(p) \times \delta_k(p)$$

- ปรับปรุ้ค่าถ่วงน้ำหนักของนิวรอนในชั้นเอาต์พุต

$$w_{jk}(p+1) = w_{jk}(p) + \Delta w_{jk}(p)$$

- คำนวณค่าความเปลี่ยนแปลงของค่า threshold ของนิวรอน  $k$

$$\Delta \theta_k(p) = \alpha \times (-1) \times \delta_k(p)$$

- ปรับปรุ้ค่า threshold ของนิวรอน  $k$  ในชั้นเอาต์พุต

$$\theta_k(p+1) = \theta_k(p) + \Delta \theta_k(p)$$

(ข) การปรับค่าถ่วงน้ำหนักและค่า threshold ในชั้นซ่อน

- คำนวณค่า error gradient ของนิวรอน  $j$  ภายในชั้นซ่อนของรอบการทำงานที่  $p$

$$\delta_j(p) = y_j(p) \times [1 - y_j(p)] \times \sum_{k=1}^l \delta_k(p) w_{jk}(p)$$

- คำนวณค่าความเปลี่ยนแปลงของค่าถ่วงน้ำหนัก (weight correction) ของนิวรอน  $i$  ที่ได้เชื่อมต่อกับนิวรอน  $j$

$$\Delta w_{ij}(p) = \alpha \times x_i(p) \times \delta_j(p)$$

- ปรับปรุงค่าถ่วงน้ำหนักของนิวรอนในชั้นซ่อน

$$w_{ij}(p+1) = w_{ij}(p) + \Delta w_{ij}(p)$$

- คำนวณค่าความเปลี่ยนแปลงของค่า threshold ของนิวรอน  $j$

$$\Delta \theta_j(p) = \alpha \times (-1) \times \delta_j(p)$$

- ปรับปรุงค่า threshold ของนิวรอน  $j$  ในชั้นซ่อน

$$\theta_j(p+1) = \theta_j(p) + \Delta \theta_j(p)$$

ขั้นตอนที่ 4 : ทำซ้ำขั้นตอนที่ 2 จนกระทั่งครบรอบการฝึกสอนที่กำหนดไว้หรือโครงข่ายมีความคลาดเคลื่อนอยู่ในเกณฑ์ที่ยอมรับได้

### 3.3.2 การวัดค่าความคลาดเคลื่อนของโครงข่าย

เนื่องจากกระบวนการเรียนรู้ของโครงข่ายแบบ BP มีวัตถุประสงค์เพื่อลดความคลาดเคลื่อนของโครงข่ายกับชุดข้อมูลทดสอบให้มีค่าน้อยลงในทุกรอบของการทำงาน โดยการวัดค่าความคลาดเคลื่อนของโครงข่ายที่นิยมใช้นั้น ได้แก่ การวัดค่าความคลาดเคลื่อนยกกำลังสอง (Sum of Square Error: SSE), การวัดค่าความคลาดเคลื่อนเฉลี่ยยกกำลังสอง (Mean of Square Error: MSE) และการวัดค่ารากที่สองของค่าความคลาดเคลื่อนเฉลี่ยยกกำลังสอง (Root Mean of Square Error: RMSE) เป็นต้น ซึ่งแต่ละวิธีสามารถคำนวณหาความคลาดเคลื่อนได้ดังนี้

$$SSE = \frac{1}{2} \sum_{t=1}^T \sum_{k=1}^l (y_{d,k(t)}(p) - y_{k(t)}(p))^2 \quad (3.18)$$

$$MSE = \frac{1}{T} \sum_{t=1}^T \sum_{k=1}^l (y_{d,k(t)}(p) - y_{k(t)}(p))^2 \quad (3.19)$$

$$RMSE = \sqrt{\frac{1}{T} \sum_{t=1}^T \sum_{k=1}^l (y_{d,k(t)}(p) - y_{k(t)}(p))^2} \quad (3.20)$$

ซึ่งค่า  $\frac{1}{2}$  เป็นค่าคงที่ใส่ลงไปเพื่อความเหมาะสมทางด้านคณิตศาสตร์ของการคำนวณ SSE

$l$  หมายถึงจำนวนของนิวรอนในชั้นเอาต์พุต

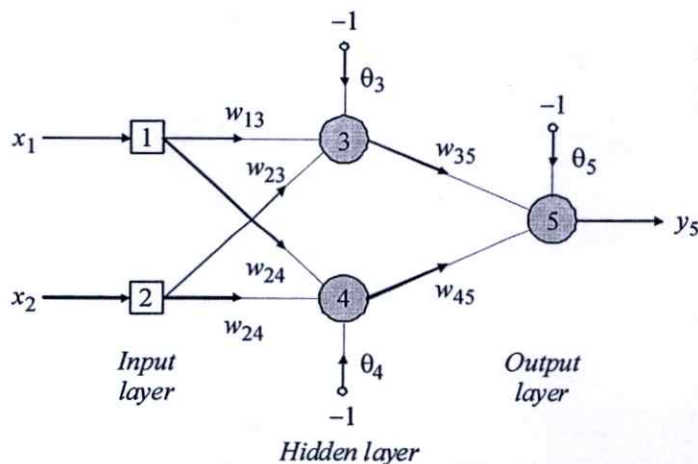
$T$  หมายถึงจำนวนของรูปแบบข้อมูลที่ใช้ในการฝึกสอน (training patterns)

### 3.3.3 ตัวอย่างของการทำงาน

ในหัวข้อนี้เป็นตัวอย่างของการนำโครงข่ายประสาทเทียมมาทำการเรียนรู้กับปัญหา Exclusive-OR ซึ่งข้อมูลที่ใช้ฝึกสอนโครงข่ายสำหรับปัญหา Exclusive-OR แสดงดังตารางที่ 3.2 และได้ออกแบบให้โครงข่ายมี 1 ชั้นซ่อน มีจำนวนนิวรอนในชั้นอินพุต, ชั้นซ่อนและชั้นเอาต์พุต จำนวน 2, 2 และ 1 นิวรอนตามลำดับ ภาพโครงข่ายที่ออกแบบแสดงดังรูปที่ 3.7

ตารางที่ 3.2 ชุดข้อมูลฝึกสอนสำหรับปัญหา Exclusive-OR

รูปแบบที่	ข้อมูลอินพุต		ค่าเอาต์พุตเป้าหมาย
	$x_1$	$x_2$	
1	1	1	0
2	0	1	1
3	1	0	1
4	0	0	0



รูปที่ 3.7 โครงข่ายประสาทเทียมที่มีจำนวน 3 ชั้น สำหรับการเรียนรู้ปัญหา Exclusive-OR (นำมาจาก : [14])

โดยนิวรอนหมายเลข 1 และ 2 เป็นนิวรอนในชั้นอินพุตซึ่งจะรับค่าสัญญาณอินพุต  $x_1$  และ  $x_2$  ตามลำดับ ( $x_1$  และ  $x_2$  เป็นค่าสัญญาณอินพุตจากชุดข้อมูลฝึกสอน) และค่าเอาต์พุตที่ออกมาจากนิวรอนหมายเลข 1 และ 2 นั้นจะเป็นค่าสัญญาณอินพุต  $x_3$  และ  $x_4$  เดิมที่รับเข้ามา (ไม่มีการนำไปคำนวณด้วยฟังก์ชันการกระตุ้น) และจะส่งต่อไปเป็นสัญญาณอินพุตสำหรับนิวรอนหมายเลข 3 และ 4 ต่อไป กำหนดค่า threshold ( $\theta$ ) ให้กับนิวรอนเฉพาะที่อยู่ในชั้นซ่อนและชั้นเอาต์พุตเท่านั้น โดยเป็นเหมือนกับค่าถ่วงน้ำหนัก (weight) อันหนึ่งที่เชื่อมต่อมาขงนิวรอนและให้ มีค่าอินพุตไบแอสคือ -1

จากตัวอย่างสมมติให้ค่าถ่วงน้ำหนักและค่า threshold ที่สุ่มค่าได้มีค่าดังนี้

$$w_{13} = 0.5, w_{14} = 0.9, w_{23} = 0.4, w_{24} = 1.0, w_{35} = -1.2 \text{ และ } w_{45} = 1.1$$

$$\theta_3 = 0.8, \theta_4 = -0.1 \text{ และ } \theta_5 = 0.3$$

และมีอัตราการเรียนรู้ ( $\alpha$ ) เท่ากับ 0.1

สำหรับตัวอย่างจะแสดงให้เห็นถึงลักษณะการทำงานของกรเรียนรู้แบบแพร่ย้อนกลับ ซึ่งทำงานกับข้อมูลฝึกสอนรูปแบบที่ 1 ซึ่งมีค่าสัญญาณอินพุต  $x_1$  และ  $x_2$  มีค่าเป็น 1 เท่ากันและมีค่าเอาต์พุตเป้าหมาย  $y_{d,5}$  มีค่าเป็น 0 ซึ่งค่าเอาต์พุตของนิวรอนหมายเลข 3 และ 4 คำนวณได้ดังนี้

$$y_3 = \text{sigmoid}(x_1 w_{13} + x_2 w_{23} - \theta_3) = 1 / [1 + e^{-(1 \cdot 0.5 + 1 \cdot 0.4 - 0.8)}] = 0.5250$$

$$y_4 = \text{sigmoid}(x_1 w_{14} + x_2 w_{24} - \theta_4) = 1 / [1 + e^{-(1 \cdot 0.9 + 1 \cdot 1.0 + 0.1)}] = 0.8808$$

จากนั้นคำนวณค่าเอาต์พุตของนิวรอนหมายเลข 5 ในชั้นเอาต์พุตดังนี้

$$y_5 = \text{sigmoid}(y_3 w_{35} + y_4 w_{45} - \theta_5) = 1 / [1 + e^{-(0.5250 \cdot 1.2 + 0.8808 \cdot 1.1 - 0.3)}] = 0.5097$$

ซึ่งได้ค่าความคลาดเคลื่อนของโครงข่ายกับรูปแบบของชุดข้อมูลฝึกสอนดังนี้

$$e = y_{d,5} - y_5 = 0 - 0.5097 = -0.5097$$

ขั้นตอนต่อไปเป็นการปรับปรุงค่าถ่วงน้ำหนักและค่า threshold โดยเริ่มจากการคำนวณค่า error gradient ของนิวรอนหมายเลข 5 ในชั้นเอาต์พุต

$$\delta_5 = y_5 (1 - y_5) e = 0.5097 \cdot (1 - 0.5097) \cdot (-0.5097) = -0.1274$$

จากนั้นคำนวณค่าความเปลี่ยนแปลงของค่าถ่วงน้ำหนักและค่า threshold ของนิวรอนหมายเลข 5 ได้ดังนี้

$$\Delta w_{35} = \alpha \cdot y_3 \cdot \delta_5 = 0.1 \cdot 0.5250 \cdot (-0.1274) = -0.0067$$

$$\Delta w_{45} = \alpha \cdot y_4 \cdot \delta_5 = 0.1 \cdot 0.8808 \cdot (-0.1274) = -0.0112$$

$$\Delta \theta_5 = \alpha \cdot (-1) \cdot \delta_5 = 0.1 \cdot (-1) \cdot (-0.1274) = -0.0127$$

ต่อไปคำนวณค่า error gradient ของนิวรอนหมายเลข 3 และ 4 ในชั้นซ่อน

$$\delta_3 = y_3(1 - y_3) \cdot \delta_5 \cdot w_{35} = 0.5250 \cdot (1 - 0.5250) \cdot (-0.1274) \cdot (-1.2) = 0.0381$$

$$\delta_4 = y_4(1 - y_4) \cdot \delta_5 \cdot w_{45} = 0.8808 \cdot (1 - 0.8808) \cdot (-0.1274) \cdot 1.1 = -0.0147$$

จากนั้นคำนวณค่าความเปลี่ยนแปลงของค่าถ่วงน้ำหนักและค่า threshold ของนิวรอนหมายเลข 3 และ 4 ได้ดังนี้

$$\Delta w_{13} = \alpha \cdot x_1 \cdot \delta_3 = 0.1 \cdot 1 \cdot 0.0381 = 0.0038$$

$$\Delta w_{23} = \alpha \cdot x_2 \cdot \delta_3 = 0.1 \cdot 1 \cdot 0.0381 = 0.0038$$

$$\Delta \theta_3 = \alpha \cdot (-1) \cdot \delta_3 = 0.1 \cdot (-1) \cdot 0.0381 = -0.0038$$

$$\Delta w_{14} = \alpha \cdot x_1 \cdot \delta_4 = 0.1 \cdot 1 \cdot (-0.0147) = -0.0015$$

$$\Delta w_{24} = \alpha \cdot x_2 \cdot \delta_4 = 0.1 \cdot 1 \cdot (-0.0147) = -0.0015$$

$$\Delta \theta_4 = \alpha \cdot (-1) \cdot \delta_4 = 0.1 \cdot (-1) \cdot (-0.0147) = 0.0015$$

และทำการปรับปรุงค่าถ่วงน้ำหนักและค่า threshold

$$w_{13} = w_{13} + \Delta w_{13} = 0.5 + 0.0038 = 0.5038$$

$$w_{14} = w_{14} + \Delta w_{14} = 0.9 - 0.0015 = 0.8985$$

$$w_{23} = w_{23} + \Delta w_{23} = 0.4 + 0.0038 = 0.4038$$

$$w_{24} = w_{24} + \Delta w_{24} = 1.0 - 0.0015 = 0.9985$$

$$w_{35} = w_{35} + \Delta w_{35} = -1.2 - 0.0067 = -1.2067$$

$$w_{45} = w_{45} + \Delta w_{45} = 1.1 - 0.0112 = 1.0888$$

$$\theta_3 = \theta_3 + \Delta \theta_3 = 0.8 - 0.0038 = 0.7962$$

$$\theta_4 = \theta_4 + \Delta \theta_4 = -0.1 + 0.0015 = -0.0985$$

$$\theta_5 = \theta_5 + \Delta \theta_5 = 0.3 + 0.0127 = 0.3127$$

ซึ่งโครงข่ายจะต้องนำรูปแบบของข้อมูลฝึกสอนอันถัดไปมาฝึกสอนต่อจนครบทุกรูปแบบ ข้อมูลจึงถือว่าทำงานครบ 1 รอบจากนั้นวัดค่าความคลาดเคลื่อนของโครงข่าย หากมีค่าน้อยไม่ถึงเกณฑ์ที่กำหนดไว้จะต้องทำงานฝึกสอนต่ออีกครั้ง เช่นกำหนดให้ทำการฝึกสอนโครงข่ายต่อจนกว่าค่า SSE ของโครงข่ายจะมีค่าน้อยกว่า 0.001 เป็นต้น หรือหยุดทำการฝึกสอนโครงข่ายเมื่อครบจำนวนรอบการฝึกสอนตามที่ได้กำหนดไว้

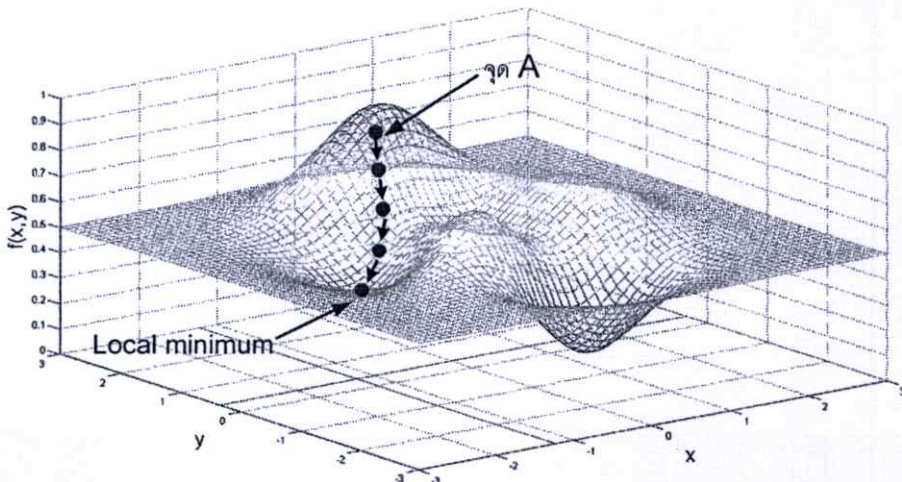
### 3.4 จุดอ่อนของโครงข่ายประสาทเทียม

ปัญหาเกี่ยวกับโครงข่ายประสาทเทียมที่มีการฝึกสอนด้วยการเรียนรู้แบบแพร่ย้อนกลับ (Back-Propagation: BP) ที่สำคัญได้แก่ ปัญหาการติดกับดักจุดต่ำสุดเฉพาะบริเวณ (local minimum) และปัญหาการเข้ากันกับชุดข้อมูลฝึกสอนมากเกินไป (overfitting) ซึ่งมีรายละเอียดดังนี้

#### 3.4.1 ปัญหาการติดกับดักจุดต่ำสุดเฉพาะบริเวณ (local minimum)

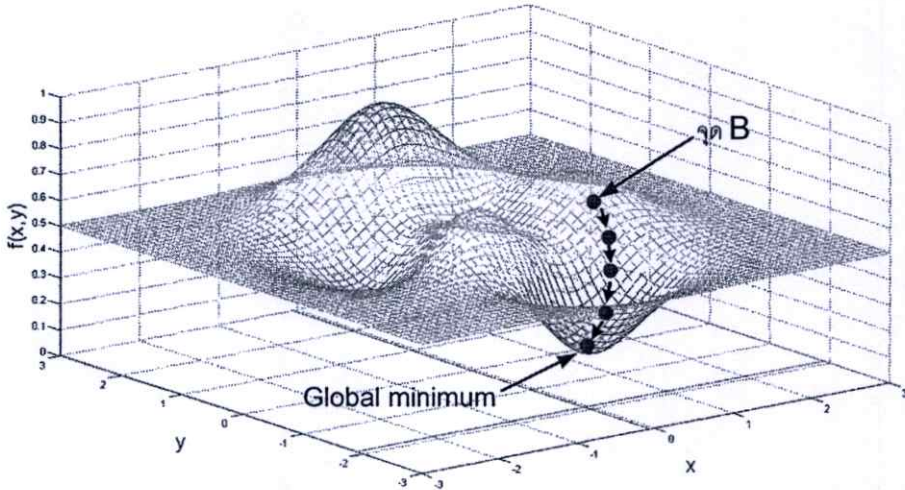
ในการฝึกสอนโครงข่ายด้วยวิธี BP ซึ่งใช้หลักการของ gradient descent ซึ่งเป็นวิธีที่มีวัตถุประสงค์เพื่อปรับค่าตัวเลขน้ำหนักเพื่อทำให้ความชันบนพื้นผิวค่าความผิดพลาด (Error surface) เคลื่อนที่ในทิศทางที่ลดลง แต่ในกรณีที่ปัญหาหรือโครงข่ายมีความซับซ้อนมาก เมื่อนำโครงข่ายมาฝึกสอนด้วย BP บางครั้งอาจจะเกิดความล้มเหลวได้เนื่องจากพื้นผิวค่าความผิดพลาดของโครงข่ายอาจมีความซับซ้อนได้เช่น มีการบิดเบี้ยว มีเนินความชันและจุดต่ำของพื้นผิวได้หลายจุด บางครั้งหากจุดต่ำที่พบอาจจะจุดต่ำสุดเฉพาะบริเวณ (local minimum) ไม่สามารถปรับค่าน้ำหนักให้เคลื่อนที่ข้ามเนินความชันของค่าความผิดพลาดหรือหลุดออกจากบริเวณดังกล่าวไปยังจุดต่ำที่สุดของระบบ (global minimum) ได้

ตัวอย่างในรูปที่ 3.8 และรูปที่ 3.9 เป็นการแสดงทิศทางการเคลื่อนที่เพื่อหาค่าต่ำสุดของฟังก์ชัน  $f(x, y) = 3(1-x)^2 e^{-x^2-(y+1)^2} - 10(x/5 - x^3 - y^5) e^{-x^2-y^2} - 1/3 e^{-(x+1)^2-y^2}$  โดยใช้วิธี gradient descent ซึ่งค่าเอาต์พุตจะถูกปรับให้มีค่าอยู่ในช่วง  $[0,1]$  ในรูปที่ 3.8 แสดงให้เห็นว่ากรณีที่ค่าน้ำหนักเริ่มต้นสุ่มได้จุด A จะพบว่าทิศทางการปรับค่าน้ำหนัก  $x$  และ  $y$  นั้นจะพยายามปรับให้ค่าของฟังก์ชันมีค่าลดลงไปเรื่อยๆ ทูกรอบการคำนวณแต่ละจะเห็นได้ว่าการปรับนั้นจุดที่ปรับจะไปตกอยู่ในจุดที่เป็น local minimum ซึ่งเป็นจุดที่ไม่ใช่จุดที่ทำให้ค่าของฟังก์ชันมีค่าต่ำสุดของระบบ



รูปที่ 3.8 แสดงลักษณะการติดกับดักปัญหา local minimum

ส่วนรูปที่ 3.9 นั้นเป็นตัวอย่างของการกำหนดค่าน้ำหนักเริ่มต้นสุ่มได้จุด B ซึ่งแตกต่างจากรูปที่ 3.8 จากรูปจะพบว่าทิศทางการปรับค่าน้ำหนัก  $x$  และ  $y$  นั้น จะพยายามปรับให้ค่าของฟังก์ชันมีค่าลดลงให้ไปเรื่อยๆ ทุกรอบการคำนวณและสามารถปรับไปตกอยู่ในจุดที่เป็น global minimum ซึ่งเป็นจุดที่ทำให้ค่าของฟังก์ชันมีค่าต่ำสุดของระบบได้



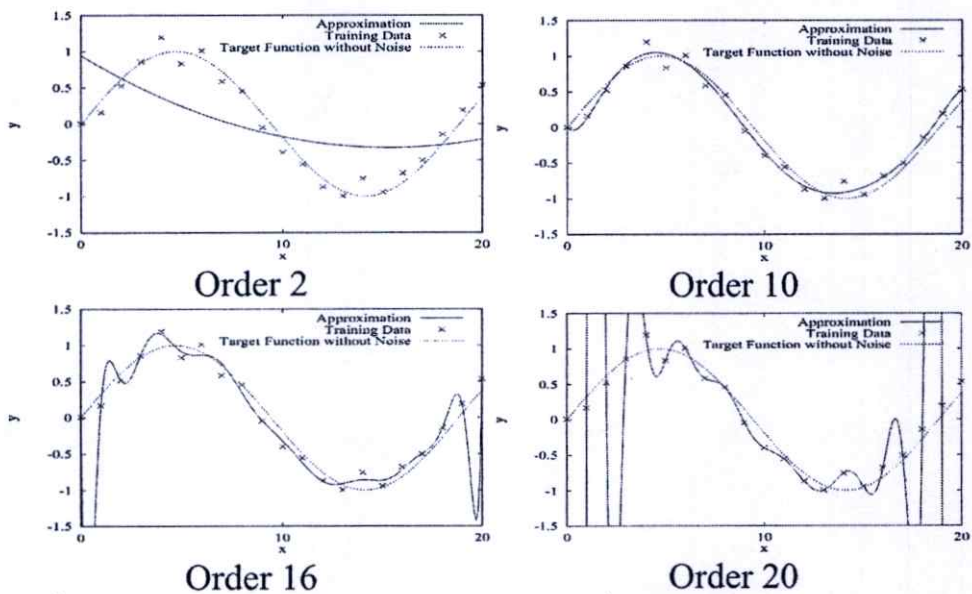
รูปที่ 3.9 แสดงลักษณะของ global minimum

จากตัวอย่างที่ผ่านมาจะเห็นได้ว่าการสุ่มค่าน้ำหนักเริ่มต้นของค่า  $x$  และ  $y$  จึงมีความสำคัญมากต่อความสำเร็จในการหาค่าต่ำสุด ดังนั้นปัญหาการติดกับดัก local minimum นี้ อาจเกิดจากการกำหนดรูปแบบเริ่มต้นของโครงข่ายที่ไม่เหมาะสม เช่นรูปแบบการเชื่อมต่อภายในและค่าน้ำหนักการเชื่อมต่อของโครงข่าย จึงเป็นสาเหตุทำให้ไม่สามารถทำการฝึกสอนโครงข่ายได้สำเร็จ ซึ่งผู้ฝึกสอนมักแก้ไขโดยกำหนดลักษณะโครงข่ายเริ่มต้นใหม่อีกครั้ง (trial-and-error)

#### 3.4.2 ปัญหาการเข้ากันกับชุดข้อมูลฝึกสอนมากเกินไป (overfitting)

คุณสมบัติของโครงข่ายประสาทเทียมที่สำคัญอันหนึ่งก็คือ Generalization หมายถึงความสามารถในการใช้งานทั่วไป ซึ่งโครงข่ายจะมีกระบวนการเรียนรู้จากข้อมูลตัวอย่างซึ่งเป็นส่วนหนึ่งของข้อมูลทั้งหมดและทำให้สามารถระบุหรือหาข้อสรุปกับข้อมูลที่ไม่เคยได้รับการฝึกสอนมาก่อน ได้ถูกต้อง แต่ในกระบวนการเรียนรู้ของโครงข่ายบางครั้งอาจเกิดปัญหาได้หากได้รับการฝึกสอนกับชุดข้อมูลฝึกสอน (Training set) มากจนเกินไป (Overtraining) เช่น จำนวนชุดข้อมูลฝึกสอนมากหรือฝึกสอนด้วยจำนวนรอบที่มากเกินไป จึงทำให้โครงข่ายประสาทเทียมนั้นมีประสิทธิภาพในการเรียนรู้กับชุดข้อมูลฝึกสอนดีมากเกินไป โครงข่ายที่ได้จึงมีความสามารถในการระบุกลุ่มของข้อมูลฝึกสอนได้สูงแต่กลับมีความถูกต้องในการระบุกลุ่มข้อมูลทดสอบต่ำ ซึ่งปัญหานี้เรียกว่า “Overfitting”

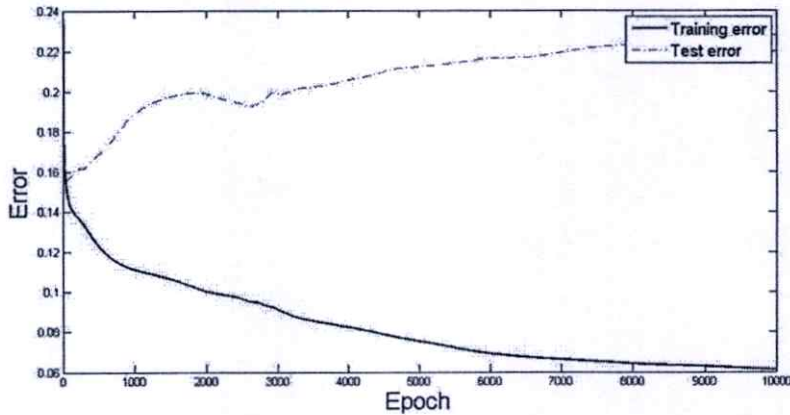
รูปที่ 3.10 [7] ได้แสดงตัวอย่างเกี่ยวกับการเกิดปัญหา Overfitting ขึ้น ซึ่งเป็นตัวอย่างของ โครงข่ายที่นำมาใช้กับการประมาณค่าแบบ polynomial คือสมการ  $y = \sin(x/3) + v$  โดย  $v$  เป็น ค่ารบกวน (noise) ที่ได้จากการสุ่มซึ่งกระจายอยู่ในช่วง  $-0.25$  ถึง  $0.25$  และชุดข้อมูลที่ใช้สำหรับการฝึกสอนจะมีทั้งหมด 21 จุด (ค่าในแนวแกน  $x$ ) ตั้งแต่  $0, 1, 2, \dots, 20$  ซึ่งข้อมูลนี้จะถูกนำไป ปรับโครงข่ายให้เหมาะสมกับโมเดล polynomial ดังกล่าวอย่างเป็นทางการลำดับตั้งแต่ 2 ถึง 20 จากรูปที่ 3.10 เส้นที่ approximation จะแสดงถึงความสามารถในการประมาณค่าเอาต์พุตสมการของ โครงข่ายที่ทำการฝึกสอน, เครื่องหมาย  $\times$  เป็นค่าเอาต์พุตของชุดข้อมูลที่นำมาฝึกสอน (Training Data) โครงข่ายและเส้นปะคือค่าเอาต์พุตจริงของสมการกับค่า  $x$  ใดๆโดยปราศจากค่า  $v$  มา เกี่ยวข้อง



รูปที่ 3.10 ลักษณะการเกิด Overfitting ของโครงข่ายเมื่อใช้กับฟังก์ชัน  $y = \sin(x/3) + v$  (นำมาจาก : [7])

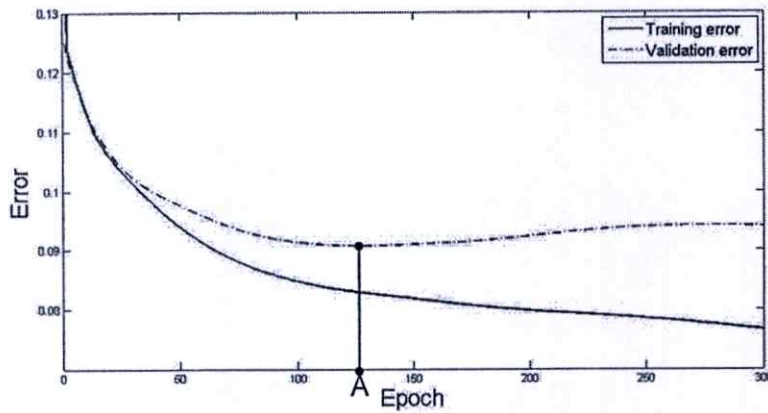
ในรอบการทำงานที่ 2 พบว่าการประมาณค่าเอาต์พุตของ โครงข่ายนั้นมีประสิทธิภาพต่ำ สำหรับรอบการทำงานที่ 10 การประมาณค่าเอาต์พุตของ โครงข่ายนั้นมีประสิทธิภาพที่ดีเพราะมี ค่าที่ได้มีความใกล้เคียงกับค่าเอาต์พุตสมการของ Training Data และค่าเอาต์พุตจริงของสมการ แต่เมื่อรอบของการฝึกสอนเริ่มมากขึ้นก็จะเริ่มเกิดปัญหา Overfitting ขึ้น ในรอบการทำงานที่ 16 การประมาณค่าเอาต์พุตของ โครงข่ายเริ่มจะปรับให้มีความสามารถให้ใกล้เคียงกับกับค่าเอาต์พุต สมการมากขึ้นแต่ในทางกลับกันกับเริ่มมีความคลาดเคลื่อนมากขึ้นเมื่อเปรียบเทียบกับค่าเอาต์พุต จริงของสมการ และในรอบการทำงานที่ 20 จะพบว่าโครงข่ายนั้นมีความสามารถในการเรียนรู้กับ ชุด Training Data ได้อย่างดีแต่กลับไม่มีประสิทธิภาพกลับในการประมาณค่าเอาต์พุตของชุด ข้อมูลอื่นๆ ได้

รูปที่ 3.11 แสดงให้เห็นว่าค่าความผิดพลาดของโครงข่ายเมื่อจำนวนการทำงานเพิ่มขึ้น โดยเมื่อวัดกับชุดข้อมูลฝึกสอน (train set) พบว่าค่าความผิดพลาดจะมีค่าลดลงเรื่อยๆ แต่สำหรับชุดข้อมูลทดสอบ (test set) พบว่าค่าความผิดพลาดกลับมีค่าเพิ่มขึ้นซึ่งตรงข้ามกับชุดข้อมูลฝึกสอน



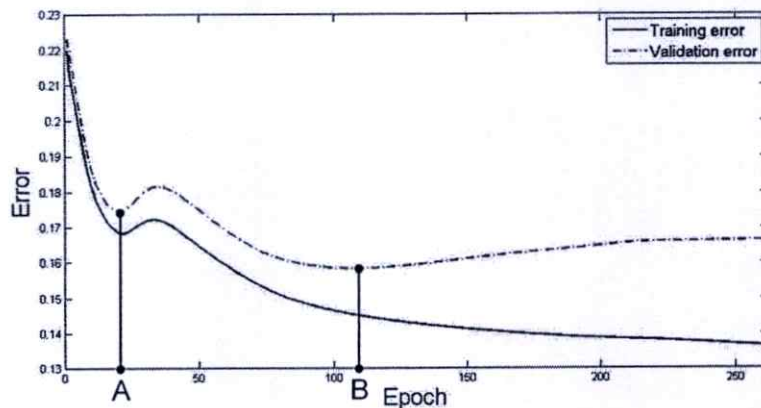
รูปที่ 3.11 แสดงค่าความผิดพลาด (Error) ของ Training set และ Test set เมื่อเกิดปัญหา overfitting ของโครงข่ายระหว่างการฝึกสอน

Early stopping เป็นเทคนิคหนึ่งที่ยอมรับมาใช้แก้ไขการเกิดปัญหา Overfitting ซึ่งวิธีนี้จะแบ่งข้อมูลฝึกสอนออกเป็นสองส่วนคือ ชุดข้อมูลฝึกสอน (training set) และชุดข้อมูลวัดประสิทธิภาพ (validation set) การทำงานจะใช้ชุดข้อมูล training set ทำการฝึกสอนปรับค่าน้ำหนักการเชื่อมต่อ (weight) และค่าไบแอส (bias) หรือค่าเทรชโฮลด์ (threshold) ของโครงข่าย ส่วนชุดข้อมูล validation set จะใช้สำหรับวัดประสิทธิภาพการ generalization ของโครงข่ายในระหว่างทำการฝึกสอนแต่จะไม่ใช้ปรับค่าน้ำหนักการเชื่อมต่อ (weight) และค่าไบแอส (bias) หรือค่าเทรชโฮลด์ (threshold) และจะทำการฝึกสอนโครงข่ายไปจนกระทั่งค่าความผิดพลาดของโครงข่ายเมื่อวัดกับชุดข้อมูล validation set (validation error) เริ่มมีค่าเพิ่มขึ้นก็จะหยุดทำการฝึกสอนทันทีเนื่องจากกำลังจะเกิดปัญหา Overfitting จากรูปที่ 3.12 หากโครงข่ายได้นำเทคนิค Early stopping มาใช้ในการแก้ปัญหาโครงข่ายจะหยุดการฝึกสอนในจุด A เนื่องจากเป็นจุดที่มีค่า validation error ที่น้อยที่สุด จากจุด A เป็นต้นไปค่า validation error จะเริ่มมีค่าความผิดพลาดเพิ่มมากขึ้น



รูปที่ 3.12 จุดการหยุดฝึกสอนโครงข่ายเมื่อใช้เทคนิค Early stopping

อย่างไรก็ตามจากเงื่อนไขการหยุดฝึกสอน (stopping criteria) ดังกล่าวอาจจะทำให้ไม่สามารถพบจุดการหยุดฝึกสอนที่เหมาะสมกว่านี้ได้ เพราะในช่วงแรกๆ ของการฝึกสอนโครงข่าย ค่า validation error อาจจะมีการแกว่งอยู่หากทำการฝึกสอนต่อไปอาจจะพบจุดที่ค่าความผิดพลาดเมื่อวัดกับชุดข้อมูล validation set ที่น้อยกว่านี้ก็เป็นไปได้ จากรูปที่ 3.13 จะเห็นว่าหากใช้เทคนิค Early stopping โครงข่ายก็จะหยุดการฝึกสอนที่จุด A แต่จากกราฟจะพบว่าหากทำการฝึกสอนโครงข่ายต่อไปอาจจะพบกับจุด B ซึ่งมีค่า validation error ที่เหมาะสมกับการหยุดกับการฝึกสอนมากกว่า



รูปที่ 3.13 ปัญหาการแกว่งของค่า validation error เมื่อใช้เทคนิค Early stopping

L. Prechelt [9] ได้นำเสนอเงื่อนไขการหยุด  $GL_{\alpha}$  เพื่อแก้ไขปัญหาดังกล่าวโดยกำหนดให้  $E_{opt}(t)$  เป็นค่า validation set error ที่น้อยที่สุดที่ได้มาภายในรอบการฝึกสอน  $t$ , ให้  $E_{val}(t)$  เป็นค่า validation error ณ รอบการฝึกสอนที่  $t$  และให้ค่า generalization loss ณ รอบการฝึกสอน  $t$  แทนด้วย  $GL(t)$  เป็นการวัดค่าที่เกี่ยวข้องกับการสูญเสียความสามารถในการทำงานได้ทั่วไป โดยเทียบกันของค่า validation error ที่เพิ่มขึ้น กับค่า  $E_{opt}(t)$  ซึ่งค่า  $GL(t)$  ที่ได้จะเป็นเปอร์เซ็นต์โดยคำนวณจากสมการ 3.21

$$GL(t) = 100 \times \left( \frac{E_{val}(t)}{E_{opt}(t)} - 1 \right) \quad (3.21)$$

โดย  $GL_\alpha$  เป็นเงื่อนไขการหยุดฝึกสอนโครงข่ายเมื่อ  $GL(t) > \alpha$  หมายถึงค่า validation error ณ รอบการฝึกสอน  $t$  มีค่ามากกว่า validation error ที่น้อยที่สุดที่ได้มาภายในรอบการฝึกสอน  $t$  อยู่  $\alpha$  % เช่น  $GL_5$  หมายความว่า จะหยุดฝึกสอนโครงข่ายเมื่อ  $GL(t) > 5$

เกณฑ์การหยุดที่สองคือ *training progress criterion* เป็นเงื่อนไขการหยุดโดยพิจารณาจากความก้าวหน้าของการฝึกสอน โดยกำหนดให้ความก้าวหน้าของการฝึกสอนแทนด้วย  $P_k(t)$  ค่าที่วัดจะมีค่าเป็นอัตราส่วนต่อหนึ่งพัน ซึ่งจะทำการคำนวณโดยวัดค่าที่ละช่วงต่อเนื่องกันเป็นจำนวน  $k$  รอบการฝึกสอน ดังนั้นในการวัดค่าจะทำในช่วงของรอบการฝึกสอนที่  $n+1 \dots n+k$  ขณะที่  $n$  เป็นค่าใดๆ ที่  $k$  สามารถหารได้ลงตัว หาก  $P_k(t) < (0.1 \text{ ส่วนหนึ่งพัน})$  ก็จะหยุดทำการฝึกสอนโครงข่าย ซึ่งค่า  $P_k(t)$  ที่วัดจะเป็นค่าเฉลี่ยของค่าความผิดพลาดที่ได้จากชุดข้อมูลฝึกสอน (average training error) ในแต่ละช่วงจำนวน  $k$  รอบของการฝึกสอนแสดงได้ดังสมการ 3.22

$$P_k(t) = 1000 \times \left( \frac{\sum_{t'=t-k+1 \dots t} E_{tr}(t')}{k \min_{t'=t-k+1 \dots t} E_{tr}(t')} - 1 \right) \quad (3.22)$$

โดย  $E_{tr}(t')$  คือค่าความผิดพลาดของโครงข่ายเมื่อวัดกับชุดข้อมูล training set ณ รอบการฝึกสอน  $t'$  ในระหว่างฝึกสอนโครงข่ายเมื่อค่า training error เริ่มมีค่าคงที่ก็จะทำการหยุดฝึกสอนเนื่องจากค่า validation error ก็จะเริ่มคงที่เช่นเดียวกัน ซึ่ง L. Prechelt ได้นำไปใช้เป็นเกณฑ์ในหยุดฝึกสอนโครงข่ายร่วมกับ  $GL_5$

ในบทนี้ได้นำเสนอเกี่ยวกับแนวคิดและวิธีการทำงานของโครงข่ายประสาทเทียม ซึ่งจำลองลักษณะระบบสมองทางชีววิทยาของมนุษย์มาพัฒนาให้อยู่ในรูปแบบการคำนวณทางคณิตศาสตร์และโครงข่ายประสาทเทียมนั้นได้ถูกนำไปใช้งานอย่างแพร่หลาย แต่จากปัญหาของโครงข่ายประสาทเทียมตามที่ได้กล่าวมานั้น วิทยานิพนธ์นี้จึงได้นำเสนอวิธีการแก้ปัญหาดังกล่าว โดยได้นำเสนอในบทที่ 5 ต่อไป

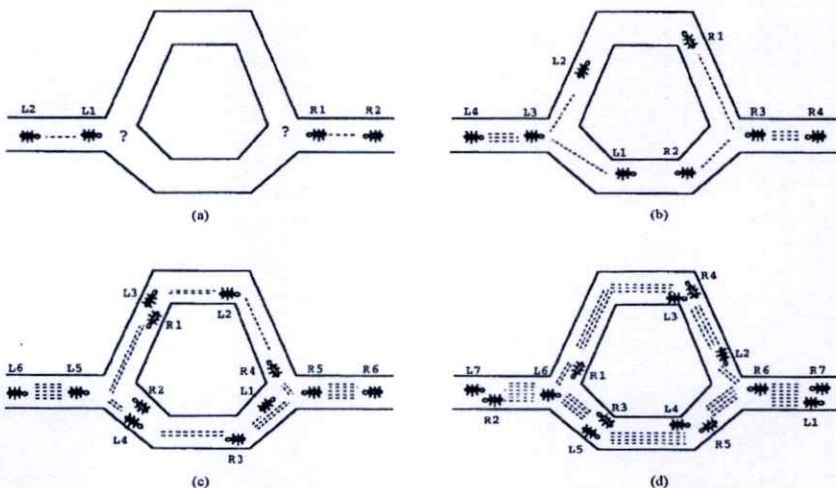
## บทที่ 4

### อัลกอริทึมมด (Ant Algorithm)

#### 4.1 ความฉลาดแบบกลุ่ม

ความฉลาดแบบกลุ่มหรือฝูง (Swarm Intelligence) [14] เป็นการจำลองพฤติกรรมของสัตว์ในธรรมชาติที่อยู่รวมกันเป็นกลุ่มหรือฝูง โดยเป็นศาสตร์แขนงหนึ่งที่ได้ถูกนำไปรวมไว้ในปัญญาประดิษฐ์ (Artificial Intelligence) เป็นเทคนิคหนึ่งในสาขาเกี่ยวกับการคำนวณที่ได้นำหลักการทางด้านชีววิทยา (Biologically-inspired computing) เช่นเดียวกับกับเทคนิคการเลียนแบบการทำงานของระบบสมองของมนุษย์ของโครงข่ายประสาทเทียม (Artificial Neural Networks) และกระบวนการวิวัฒนาการทางพันธุกรรมของสิ่งมีชีวิต (Genetic Algorithm) เป็นต้น วิธีการทำงานของความฉลาดแบบกลุ่มถูกนิยมนำมาใช้ในการแก้ปัญหาเกี่ยวกับการคำนวณหรือการหาคำตอบที่เหมาะสม (Optimization) โดยวิธีที่นิยมใช้มี 2 วิธีคืออัลกอริทึมมด (Ant Algorithm) ซึ่งเป็นการหาคำตอบที่เหมาะสมด้วยอาณานิคมมด (Ant Colony Optimization: ACO) ซึ่งเป็นการจำลองพฤติกรรมการหาเส้นทางที่สั้นสำหรับการอาหารของฝูงมด และการค่าที่เหมาะสมแบบกลุ่มอนุภาค (Particle Swarm Optimization) ซึ่งเป็นการจำลองพฤติกรรมการหาอาหารของฝูงนกหรือฝูงปลา เป็นต้น ในวิธีการเหล่านี้จะประกอบไปด้วยสมาชิกภายในฝูงที่ร่วมมือกันทำงานและมีวิธีการสื่อสารกันภายในกลุ่มเพื่อที่จะทำให้สมาชิกภายในฝูงทราบถึงเส้นทางของแหล่งอาหารได้

สำหรับ ACO นั้นได้แนวคิดมาจากพื้นฐานของการค้นหาแหล่งอาหารของฝูงมดในธรรมชาติ [15, 16, 17] โดยมดทุกตัวในฝูงจะร่วมมือกันค้นหาแหล่งอาหารและพยายามค้นหาเส้นทางจากรังไปยังแหล่งอาหารให้ได้ระยะทางที่สั้นที่สุดตัวอย่างแสดงดังรูปที่ 4.1



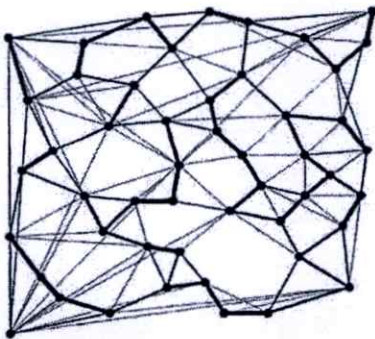
รูปที่ 4.1 ธรรมชาติการเดินของมดโดยใช้เส้นทางที่สั้น (นำมาจาก : [17])

การค้นหาเส้นทางนั้นมดใช้สารเคมีที่เรียกว่า ฟีโรโมน (Pheromone) เป็นข้อมูลในการหาเส้นทางทั้งขาไปและขากลับรัง โดยมดจะพ่นฟีโรโมนลงบนเส้นทางที่ผ่านเพื่อเป็นข้อมูลให้มดตัวอื่นๆ เดินตามและมดที่เดินตามก็จะวางฟีโรโมนทับลงไปอีก ฟีโรโมนในเส้นทางใดมีค่ามากก็จะมีจำนวนมดผ่านมากและเส้นทางจะมีความสั้นกว่าเส้นทางที่มีปริมาณฟีโรโมนน้อย จนกระทั่งสุดท้ายเส้นทางที่มีมดเลือกเดินมากที่สุดคือเส้นทางที่มีระยะทางที่สั้นที่สุด

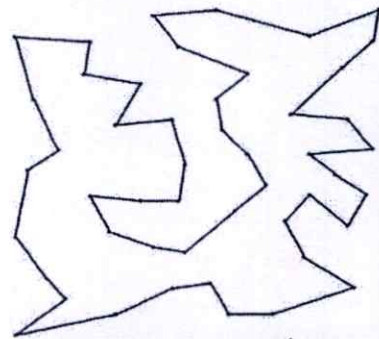
ทำให้ฟีโรโมนเป็นข้อมูลสำคัญต่อการใช้ค้นหาแหล่งอาหารของมดและการเดินทางกลับรัง เรียกว่า เป็นกลไกที่เกี่ยวกับผลสะท้อนกลับในทางบวก (Positive Feedback) โดยสนับสนุนวิธีการแก้ปัญหาที่ดีซึ่งจะทำให้ได้คำตอบที่ดียิ่งขึ้นซึ่งเก็บอยู่ในความทรงจำของมด โดยที่มดจะสามารถนำออกมาใช้เพื่อสร้างคำตอบที่ดีขึ้นได้ แต่จะต้องมีวิธีที่จะต้องป้องกันการลู่เข้าสู่คำตอบเร็วกว่าที่ควรจะเป็น โดยการละทิ้งคำตอบที่ดีบางคำตอบเนื่องจากการสนับสนุนมากเกินไป จึงอาศัยการนำผลสะท้อนกลับในทางลบ (Negative Feedback) มาใช้ โดยจำลองให้ฟีโรโมนนั้นมีการระเหย (Evaporation) หรือลดลงไปตามเวลาที่เพิ่มขึ้น ขั้นตอนวิธีระบบอาณานิคมมดจึงถูกใช้แก้ปัญหาการหาค่าเหมาะสมที่สุดเชิงการจัด (Combinatorial optimization) ในบทนี้จะได้กล่าวถึงวิธีการปรับปรุงประสิทธิภาพขั้นตอนวิธีอาณานิคมมด 2 วิธี ได้แก่ ระบบมด (Ant System) และระบบอาณานิคมมด (Ant Colony System) โดยอธิบายกระบวนการทำงานสำหรับการแก้ปัญหาเกี่ยวกับการหาเส้นทางเดินของพนักงานขาย (Traveling Salesman Problem: TSP)

#### 4.2 ปัญหาเกี่ยวกับการหาเส้นทางเดินของพนักงานขาย

ปัญหาเกี่ยวกับการหาเส้นทางเดินของพนักงานขายนั้นเป็นปัญหาเกี่ยวกับการหาค่าเหมาะสมที่สุดเชิงการจัด มีวัตถุประสงค์เพื่อหาเส้นทางการเดินทางไปยังเมืองต่างๆ ของพนักงานขาย เพื่อให้ได้ระยะทางการเดินที่สั้นที่สุด โดยจะต้องเดินให้ครบทุกเมืองและแต่ละเมืองจะถูกเดินผ่านได้เพียงหนึ่งครั้งเท่านั้น เมื่อพนักงานขายเดินมาจนครบทุกเมืองแล้วจะต้องเดินกลับมายังเมืองแรกที่เริ่มเดิน (จุดเริ่มต้น) เพื่อให้ครบรอบการเดินทาง การนำเสนอปัญหาที่นิยมใช้กันก็จะแสดงให้อยู่ในรูปของกราฟแสดงระยะห่างระหว่างเมืองตัวอย่างดังรูปที่ 4.2



(ก) เส้นทางภายใน



(ข) เส้นทางการเดินทางที่สั้นที่สุด

รูปที่ 4.2 ตัวอย่างปัญหา TSP (Eil50) (นำมาจาก : [14])

### 4.3 ระบบมด (Ant System: AS)

วิธีระบบมดนี้เป็นอัลกอริทึมแรกๆ ที่นำมาใช้ใน ACO ซึ่งถูกนำเสนอโดย M. Dorigo, V. Maniezzo และ A. Colomi [16] เริ่มแรกของวิธีการนี้จะถูกนำมาแก้ไขปัญหา TSP ซึ่ง AS นั้นมีประสิทธิภาพในการแก้ปัญหาเกี่ยวกับการกำหนดเส้นทางที่สั้นที่สุดที่ได้ดีหากพื้นที่ของคำตอบ (Solution space) หรือมีจำนวนของเมืองไม่มากนัก แต่ไม่เหมาะสำหรับปัญหาที่จำนวนเมืองมากๆ หรือเป็นปัญหาที่มีพื้นที่ของคำตอบมาก

คุณลักษณะที่สำคัญในแต่ละรอบของการทำงานของ AS ก็คือมดทุกตัวในระบบหรือฝูงเมื่อเลือกเส้นทางในการเดินจากตำแหน่ง  $i$  ไปตำแหน่ง  $j$  แล้วต่อไปต้องทำการปรับปรุงค่าฟีโรโมน  $\tau_{ij}$  บนเส้นทางทุกๆ เส้นทางด้วยสมการที่ (4.1)

$$\tau_{ij} \leftarrow (1 - \rho) \cdot \tau_{ij} + \sum_{k=1}^m \Delta \tau_{ij}^k \quad (4.1)$$

ซึ่ง  $\rho$  หมายถึงสัมประสิทธิ์ของอัตราการระเหยของฟีโรโมน

$m$  หมายถึงจำนวนของมดทั้งหมด

หลังจากที่มดตัวที่  $k$  เดินทางครบทุกตำแหน่งแล้ว ก็จะวางฟีโรโมนเป็น  $\Delta \tau_{ij}^k(t)$  บนด้าน  $(i, j)$  ซึ่งมดตัวที่  $k$  ได้เดินผ่าน โดย  $\Delta \tau_{ij}^k$  เป็นปริมาณของฟีโรโมนของมดตัวที่  $k$  ที่วางไว้บนด้าน  $(i, j)$  ซึ่งได้เดินผ่านคำนวณได้จากสมการที่ (4.2)

$$\Delta \tau_{ij}^k = \begin{cases} Q / L^k & \text{if ant } k \text{ used edge}(i, j) \text{ in its tour,} \\ 0 & \text{otherwise} \end{cases} \quad (4.2)$$

ซึ่ง  $Q$  เป็นค่าคงที่

$L^k$  เป็นระยะทางของการเดินทางครบรอบของมดตัวที่  $k$

จากสมการที่ (4.1) ในเทอมแรกคือ  $(1 - \rho) \cdot \tau_{ij}$  เป็นกระบวนการของการปรับลดฟีโรโมนเดิมบนเส้นทาง  $(i, j)$  ลงเนื่องจากการระเหยของฟีโรโมน ส่วนในเทอมที่สองคือ  $\sum_{k=1}^m \Delta \tau_{ij}^k$  จะเป็นการเพิ่มฟีโรโมนทับลงไปบนเส้นทางจากมดทุกตัวที่ได้เดินผ่าน ฉะนั้นการกำหนดจำนวนมดทั้งหมด  $m$  ตัวสำหรับการทำงานในระบบจึงมีความสำคัญมากเนื่องจากหากมีจำนวนมดมากเกินไปจะทำให้ค่าของฟีโรโมนบนเส้นทางมีค่าที่เพิ่มขึ้นอย่างรวดเร็วทำให้มีการลู่เข้าหาคำตอบที่เร็วและอาจจะได้คำตอบที่ไม่ดี แต่หากมีจำนวนของมดน้อยเกินไปผลการทำงานนั้นจะไม่ได้ผลลัพธ์ที่ดีเนื่องจากการระเหยของฟีโรโมนจะมีมากกว่าการเพิ่มฟีโรโมนลงไป จะสังเกตเห็นว่าเส้นทาง  $(i, j)$  ใดมีจำนวนมดผ่านมากจะมีค่าของฟีโรโมนเพิ่มขึ้นมาก แต่ถ้าเส้นทาง  $(i, j)$  ใดมีจำนวนมดผ่านน้อยก็จะมีค่าของฟีโรโมนเพิ่มขึ้นน้อยกว่า

การสร้างคำตอบหรือการจัดลำดับของเส้นทางการเดินทางนั้น ในขณะที่มดตัวที่  $k$  อยู่ในตำแหน่งเมืองปัจจุบันที่  $i$  ในการเลือกไปยังตำแหน่งเมือง  $j$  ถัดไปนั้น จะพิจารณาจากเซต  $s^P$  ซึ่งเป็นกลุ่มของเมืองที่มดตัวที่  $k$  ยังไม่ได้เลือกเดินไป โดยการเลือกเดินนั้นจะอาศัยค่าความน่าจะเป็นของแต่ละเมืองมาเป็นข้อมูลในการเลือกและค่าความน่าจะเป็นที่แต่ละเมืองจะถูกเลือกนั้นจะคำนวณจากสมการการส่งผ่านหรือค่าความน่าจะเป็น ( $p_{ij}^k$ ) ดังสมการที่ (4.3)

$$p_{ij}^k = \begin{cases} \frac{\tau_{ij}^\alpha \cdot \eta_{ij}^\beta}{\sum_{c_{il} \in N(s^P)} \tau_{il}^\alpha \cdot \eta_{il}^\beta} & \text{if } c_{ij} \in N(s^P), \\ 0 & \text{otherwise} \end{cases} \quad (4.3)$$

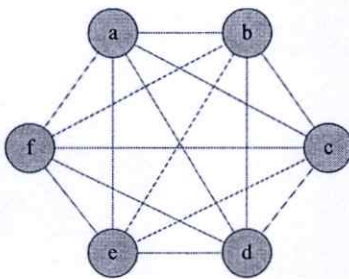
ซึ่ง  $N(s^P)$  คือเซตของเส้นทางจากเมือง  $i$  ไปยังเมืองอื่นซึ่งมดตัวที่  $k$  ยังไม่ได้เลือกเดินไป

$c_{il}$  หมายถึงเส้นทางใดๆ ที่มดตัวที่  $k$  สามารถเลือกเดินได้

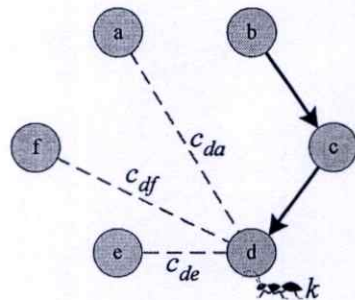
$l$  หมายถึงเมืองใดๆ ที่มดตัวที่  $k$  ยังไม่ได้เลือกเดินไป

โดยที่  $\alpha$  และ  $\beta$  เป็นค่าคงที่ค่าหนึ่งใช้สำหรับควบคุมความสำคัญของความสัมพันธ์กันระหว่างค่าฟีโรโมน  $\tau_{ij}$  กับข้อมูลของปัญหาที่สนใจ  $\eta_{ij}$  (Heuristic information) ว่าในการทำงานนั้นจะให้ความสำคัญกับสิ่งใดมากกว่ากัน เช่นถ้าให้  $\alpha = 0$  แสดงว่าตำแหน่งที่ใกล้ที่สุดก็จะมีค่าความน่าจะเป็นที่จะถูกเลือกมากกว่า ในทางตรงข้ามถ้า  $\beta = 0$  แสดงว่าจะมีเพียงค่าของฟีโรโมนเท่านั้นที่มีผลต่อการเลือก โดยปัญหา TSP นั้นค่า  $\eta_{ij} = 1/d_{ij}$  ซึ่ง  $d_{ij}$  คือระยะทางระหว่างเมือง  $i$  และ  $j$

ตัวอย่างการใช้สมการในหัวข้อนี้ยกตัวอย่างปัญหา TSP ดังรูปที่ 4.3



(ก) เส้นทางระหว่างเมืองทั้งหมด



(ข) เส้นทางที่มดสามารถเลือกเดินได้ขณะอยู่ที่เมือง  $d$

รูปที่ 4.3 กราฟของปัญหา TSP จำนวน 6 เมือง

สำหรับปัญหา TSP รูปที่ 4.3 แสดงว่ามีจำนวนเมืองทั้งหมด 6 เมืองคือ  $\{a, b, c, d, e, f\}$  รูปที่ 4.3 (ก) แสดงถึงเส้นทางระหว่างเมืองทั้งหมดในระบบและรูปที่ 4.3 (ข) แสดงว่ามดตัวที่  $k$  ได้เลือกเดินไปยังเมือง  $b, c$  และ  $d$  แล้วตามลำดับและเมืองปัจจุบัน ( $i$ ) คือเมือง  $d$  ดังนั้นในการ

พิจารณาว่ามดตัวที่  $k$  จะเลือกเดินไปยังเมืองถัดไป ( $j$ ) จะได้ว่าเซตของเมืองที่ยังไม่ได้เดินผ่านคือ  $s^p = \{a, e, f\}$  และเซตเส้นทางจากเมือง  $d$  ไปเมืองที่ยังไม่ได้เดินผ่านคือ  $N(s^p) = \{c_{da}, c_{dp}, c_{de}\}$

#### 4.4 ระบบอาณานิคมมด (Ant Colony System: ACS)

ระบบอาณานิคมมด (Ant Colony System: ACS) ถูกนำเสนอโดย M. Dorigo และ L. M. Gambardella [15,17] ได้รับการพัฒนาขึ้นเพื่อเพิ่มประสิทธิภาพจากระบบมดเพื่อทำให้สามารถแก้ไขปัญหามีขนาดใหญ่ได้ โดยได้เพิ่มการทำงานดังนี้

##### 4.4.1 การปรับปรุงค่าฟีโรโมนในระดับเฉพาะบริเวณ (local pheromone update)

การปรับปรุงค่าฟีโรโมนในระดับเฉพาะบริเวณจะกระทำโดยมดทุกตัวในระบบ โดยจะปรับปรุงในทุกเส้นทางที่มดแต่ละตัวได้เลือกตำแหน่งเมือง  $j$  ถัดไปในขณะที่ปัจจุบันมดอยู่ที่ตำแหน่ง  $i$  และปริมาณฟีโรโมนบนด้านที่เชื่อมต่อระหว่างเมือง ( $i, j$ ) จะถูกปรับปรุงด้วยสมการที่ (4.4)

$$\tau_{ij} = (1 - \varphi) \cdot \tau_{ij} + \varphi \cdot \tau_0 \quad (4.4)$$

ซึ่ง  $\tau_0$  คือค่าเริ่มต้นของฟีโรโมน

$\varphi$  เป็นสัมประสิทธิ์ของการเสื่อมสลายของฟีโรโมนในระดับเฉพาะบริเวณซึ่ง  $\varphi \in (0,1]$

หลักสำคัญของการปรับปรุงฟีโรโมนในระดับเฉพาะที่ คือเป็นการป้องกันไม่ให้เส้นทางที่มีจำนวนมดผ่านมากมีค่าของฟีโรโมนในเส้นทางนั้นมากเกินไป (ค่าความน่าจะเป็นของเส้นทางนั้นมีค่ามาก) จนทำให้เส้นทางอื่นมีโอกาสดูถูกเลือกน้อย ดังนั้นเมื่อมีมดผ่านมายังด้านระหว่างสองตำแหน่งใดๆ การปรับปรุงในระดับพื้นที่นี้จะทำให้ระดับของฟีโรโมนบนด้านนั้นมีค่าความน่าสนใจลดน้อยลง ดังนั้นจึงทำให้มดตัวอื่นในฝูงมีโอกาสดูที่สำรวจเส้นทางใหม่ๆ หรือค้นพบเส้นทางที่แตกต่างออกไปได้และก็มีค่าความน่าจะเป็นที่มดตัวใดตัวหนึ่งจะพบคำตอบซึ่งถูกปรับปรุงให้ดีขึ้นมากกว่าที่มีอยู่ก็เป็นได้

##### 4.4.2 การปรับปรุงค่าฟีโรโมนในระดับครอบคลุมทั้งระบบ (global pheromone update)

การปรับปรุงค่าฟีโรโมนในระดับครอบคลุมทั้งระบบ เมื่อครบรอบของการทำงานในแต่ละรอบแล้ว จะพิจารณาปรับปรุงค่าฟีโรโมนเฉพาะเส้นทางของมดตัวที่มีคำตอบที่ดีที่สุดเท่านั้น มาเป็นค่าในการปรับปรุงค่าฟีโรโมนในระดับครอบคลุมทั้งระบบดังสมการที่ (4.5)

$$\tau_{ij} \leftarrow \begin{cases} (1 - \rho) \cdot \tau_{ij} + \rho \cdot \Delta \tau_{ij} & \text{if } (i, j) \text{ belongs to best tour,} \\ \tau_{ij} & \text{otherwise} \end{cases} \quad (4.5)$$

ซึ่ง  $\Delta\tau_{ij}(t) = 1/L_{best}$  และ  $L_{best}$  คือรอบการเดินทางที่สั้นที่สุดจากการเดินของมดทั้งหมดทุกตัว ดังนั้นสำหรับปัญหา TSP กระบวนการนี้จะปรับปรุงค่าฟีโรโมนบนเส้นทาง  $(i, j)$  ซึ่งเป็นเส้นทางที่มีระยะทางของรอบการเดินทางที่สั้นที่สุดเท่านั้น

#### 4.4.3 กระบวนการตัดสินใจเลือกตำแหน่งถัดไป

ข้อแตกต่างที่สำคัญของ ACS และ AS อีกอันหนึ่งก็คือกระบวนการตัดสินใจเลือกตำแหน่งเมือง  $j$  ถัดไปขณะที่ปัจจุบันอยู่ตำแหน่งเมือง  $i$  ในการเลือกตำแหน่งเมืองถัดไปนั้นวิธี ACS จะพิจารณาจากสมการที่ (4.6)

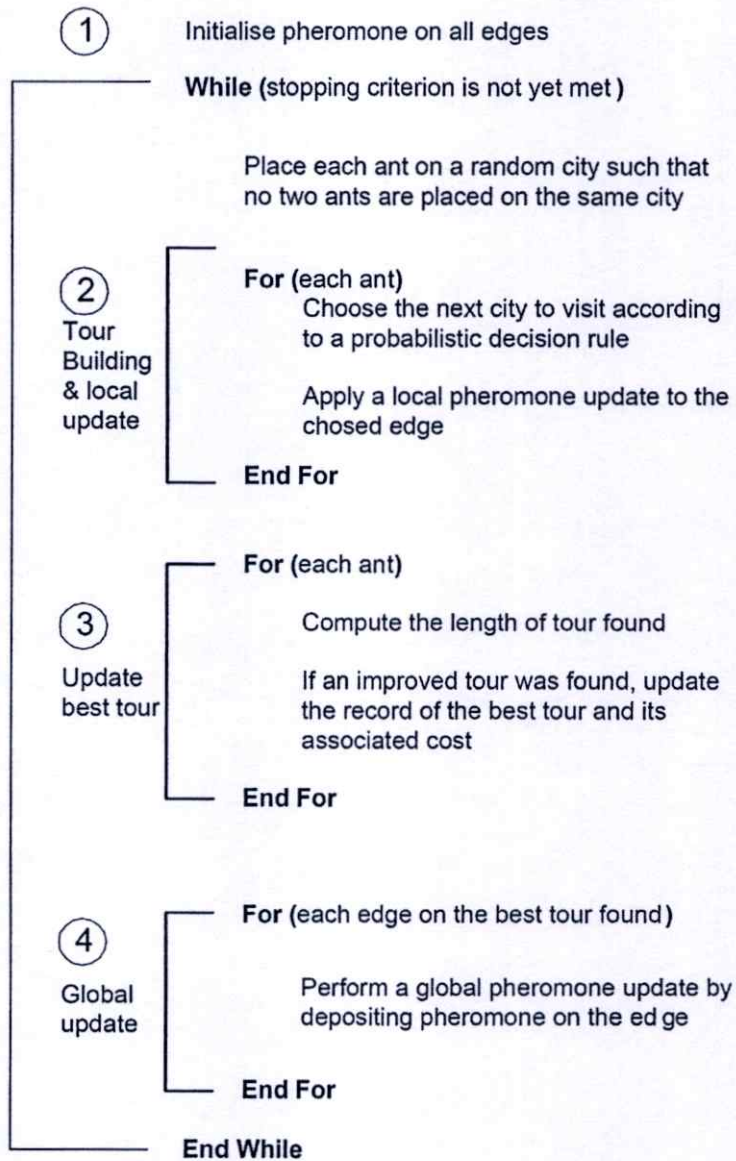
$$j = \begin{cases} \operatorname{argmax}_{c_{ij} \in N(s^p)} \{ \tau_{ij}^\alpha \cdot \eta_{ij}^\beta \} & \text{if } q \leq q_0, \\ J & \text{otherwise} \end{cases} \quad (4.6)$$

ค่า  $q_0$  เป็นค่าคงที่มีค่าอยู่ระหว่าง  $0 \leq q_0 \leq 1$  ส่วนตัวแปร  $q$  คือค่าจากการสุ่มมีค่าอยู่ระหว่าง  $[0, 1]$  มดแต่ละตัวจะต้องสุ่มกำหนดค่า  $q$  ใหม่ทุกครั้งเพื่อใช้สำหรับการเลือกเดินไปยังตำแหน่งเมืองถัดไป

หาก  $q \leq q_0$  จะเลือกเส้นทางที่มีค่าฟีโรโมนที่มากหรือดีที่สุด แสดงให้เห็นว่าวิธี ACS นั้นจะพยายามรักษาเส้นทางเดิมที่ดีที่สุดซึ่งระบบการทำงานในขณะนั้นสามารถหาได้ จึงแสดงว่าเส้นทางนั้นมีแนวโน้มว่าจะเป็นคำตอบที่ดีได้ แต่ถ้าหาก  $q > q_0$  มดก็จะทำการเลือกตำแหน่งเมืองถัดไปคือเมือง  $J$  โดยใช้การเลือกมาจากค่าความน่าจะเป็นของแต่ละเส้นทางที่เป็นไปได้ด้วยสมการที่ (4.3) ในกรณีนี้แสดงว่ามดจะได้ทำการสำรวจเส้นทางใหม่ๆ ซึ่งอาจได้คำตอบที่ดีกว่าเดิมก็เป็นไปได้

ฉะนั้นการกำหนดค่าของ  $q_0$  จึงต้องกำหนดให้มีค่าเข้าใกล้ 1 นิยมกำหนดให้เท่า 0.9 [16] เพื่อให้วิธี ACS ได้พยายามรักษาคำตอบที่ดีในขณะนั้นไว้และก็มีโอกาสที่จะค้นหาคำตอบใหม่ๆ ได้ แต่ถ้ากำหนดให้ค่าของ  $q_0$  มีค่าเข้าใกล้ 0 แล้ว วิธี ACS จึงมีโอกาสน้อยมากที่จะรักษาคำตอบที่ดีไว้ได้แต่จะไปอาศัยความน่าจะเป็นของแต่ละเมืองแทนทำให้คำตอบที่ได้มานั้นมีทิศทางที่จะไม่ลู่ไปในทิศทางเดียวกัน

#### 4.4.4 รหัสเทียมแสดงขั้นตอนการทำงานของระบบอาณานิคมมด



รูปที่ 4.4 ขั้นตอนการทำงานของระบบอาณานิคมมด (นำมาจาก : [18])

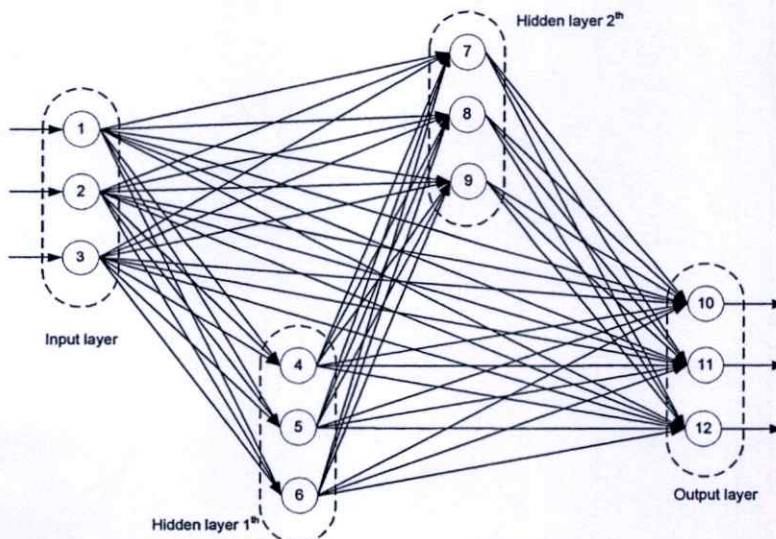
## บทที่ 5

# การปรับเปลี่ยนรูปแบบการเชื่อมต่อของโครงข่ายประสาทเทียม ด้วยอัลกอริธึมมด

ในบทนี้จะกล่าวถึงรายละเอียดเกี่ยวกับการพัฒนาโครงข่ายประสาทเทียม โดยวิธีที่นำเสนอในวิทยานิพนธ์นี้ได้นำแนวคิดเกี่ยวกับการนำเทคนิควิธีระบบอาณานิคมมด (Ant Colony System: ACS) มาทำการพัฒนาเพื่อค้นหาลักษณะของโครงข่ายที่เหมาะสมทั้งในส่วนของ การเชื่อมต่อภายในและค่าถ่วงน้ำหนักของโครงข่ายไปพร้อมๆ กัน และจะทำงานร่วมกับการเรียนรู้ด้วย BP แบบบางส่วน (partial learning) โดย BP นั้นจะนำมาใช้ฝึกสอนโครงข่ายแบบด้วยจำนวนรอบน้อยๆ และดูแนวโน้มความเหมาะสมของโครงข่ายที่ได้มาจากการค้นหาของมดแต่ละตัวในวิธี ACS โดยวัดจากค่าความผิดพลาดของโครงข่ายและนำค่าความผิดพลาดนั้นไปใช้ในการปรับปรุงค่าพารามิเตอร์ที่เป็นตัวเลือกของวิธี ACS ในรอบถัดไป ซึ่งรายละเอียดการออกแบบและขั้นตอนการทำงานของวิธีที่นำเสนอมีดังต่อไปนี้

### 5.1 การออกแบบโครงข่ายประสาทเทียม

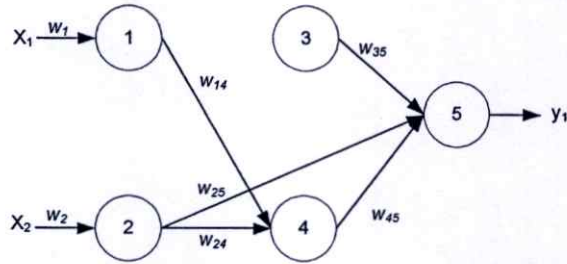
วิธีที่นำเสนอจะใช้วิธีระบบอาณานิคมมด (ACS) ในการค้นหาโครงสร้างของโครงข่ายประสาทเทียมที่เหมาะสมทั้งในส่วนของ การเชื่อมต่อภายในและค่าน้ำหนักจากโครงข่ายที่ได้ กำหนดจำนวนชั้นซ่อน รวมถึงจำนวนนิวรอนในชั้นอินพุต ชั้นซ่อน และชั้นเอาต์พุตไว้แล้ว ซึ่งนิวรอนภายในโครงข่ายสามารถเชื่อมต่อกันข้ามชั้นได้แต่ต้องไม่เชื่อมต่อย้อนกลับไปยังนิวรอนในชั้นก่อนหน้า และนิวรอนที่อยู่ในชั้นติดกันก็ไม่จำเป็นต้องเชื่อมต่อถึงกันก็ได้ ตัวอย่างโครงข่ายที่ออกแบบแสดงดังรูปที่ 5.1



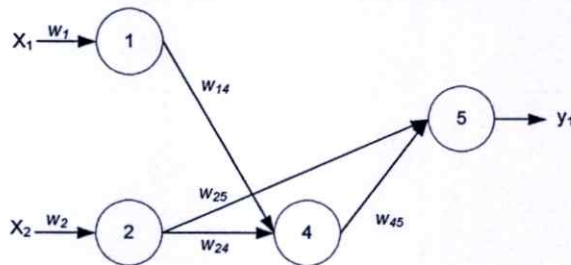
รูปที่ 5.1 ลักษณะโครงข่ายที่ออกแบบ

### 5.1.1 การลบนิวรอนออกจากโครงข่าย

ภายในกระบวนการทำงานของ ACS นั้นสามารถลบนิวรอนที่ไม่ได้นำมาใช้งานออกจากโครงข่ายได้ ซึ่งนิวรอนที่จะถูกลบได้นั้นจะเป็นนิวรอนที่ไม่มีการเชื่อมต่อเข้ามาจากนิวรอนอื่น (ไม่มีอินพุตเข้ามา) แสดงดังรูปที่ 5.2 หรือเป็นนิวรอนที่ไม่มีเชื่อมต่อไปออกไปยังนิวรอนอื่นๆ (ไม่มีเอาต์พุตออกจากนิวรอน) แสดงดังรูปที่ 5.3

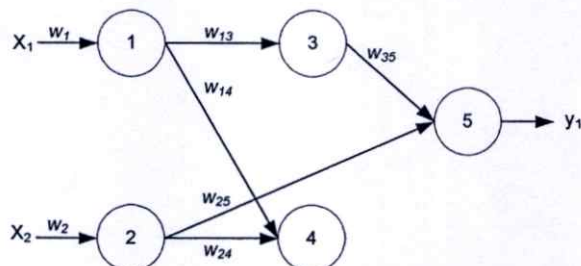


(ก) นิวรอนหมายเลข 3 ที่ไม่มีอินพุตเข้ามา

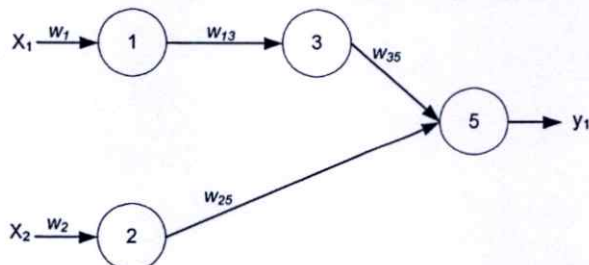


(ข) ลบนิวรอนหมายเลข 3 ออกไป

รูปที่ 5.2 การลบนิวรอนที่ไม่มีการเชื่อมต่อเข้ามา



(ก) นิวรอนหมายเลข 4 ที่ไม่มีเอาต์พุตออกไป



(ข) ลบนิวรอนหมายเลข 4 ออกไป

รูปที่ 5.3 การลบนิวรอนที่ไม่มีการเชื่อมต่อออกไป

### 5.1.2 การคำนวณของโครงข่ายที่ออกแบบ

เนื่องจากวิธีการนี้ได้ออกแบบให้นิวรอนภายในสามารถเชื่อมต่อกับนิวรอนอื่นๆ แบบข้ามชั้นถัดไปได้ ดังนั้นจึงต้องมีการปรับปรุงสมการการคำนวณให้สอดคล้องกับรูปแบบของโครงข่ายประสาทเทียมที่ออกแบบไว้ดังนี้

#### 1) การคำนวณค่าเอาต์พุตของนิวรอนที่อยู่ในชั้นซ่อนและชั้นเอาต์พุต

การคำนวณค่าเอาต์พุตในชั้นซ่อนและชั้นเอาต์พุต สามารถคำนวณได้ดังสมการที่ (5.1) และสมการที่ (5.2) ตามลำดับ

$$y_h = f\left(\sum_l x_l w_{lh} - \theta_h\right) \quad (5.1)$$

$$y_g = f\left(\sum_l x_l w_{lg} - \theta_g\right) \quad (5.2)$$

โดย  $y_h$  คือค่าเอาต์พุตของนิวรอน  $h$  ซึ่งเป็นนิวรอนที่อยู่ในชั้นซ่อน ส่วน  $y_g$  คือค่าเอาต์พุตของนิวรอน  $g$  ซึ่งเป็นนิวรอนที่อยู่ในชั้นเอาต์พุต,  $f$  เป็นฟังก์ชันการกระตุ้น,  $x_l$  คือค่าอินพุตที่ได้มาจากนิวรอน  $l$  ซึ่ง  $l$  คือหมายเลขนิวรอนใดๆ ในชั้นก่อนหน้า (รวมชั้นอินพุต) ที่ได้เชื่อมต้อมายังนิวรอน  $h$  หรือ  $g$ , ส่วน  $w_{lh}$  คือค่าน้ำหนักการเชื่อมต่อระหว่างนิวรอน  $l$  กับนิวรอน  $h$ , ส่วน  $w_{lg}$  คือค่าน้ำหนักการเชื่อมต่อระหว่างนิวรอน  $l$  กับนิวรอน  $g$  และ  $\theta_h$  กับ  $\theta_g$  คือค่า threshold ของนิวรอน  $h$  และ  $g$  ตามลำดับ

#### 2) การปรับปรุงค่าน้ำหนักการเชื่อมต่อของนิวรอนในชั้นเอาต์พุตและนิวรอนชั้นซ่อน

การปรับปรุงค่าน้ำหนักการเชื่อมต่อของนิวรอนในชั้นเอาต์พุตและนิวรอนชั้นซ่อน กำหนดให้มีการคำนวณดังนี้

- ค่า error gradient สำหรับนิวรอน  $g$  ในชั้นเอาต์พุต ( $\delta_g(p)$ ) สามารถคำนวณได้จากสมการที่ (5.3)

$$\delta_g(p) = y_g(p) \cdot [1 - y_g(p)] e_g(p) \quad (5.3)$$

$$e_g(p) = y_{d,g} - y_g(p) \quad (5.4)$$

ในสมการที่ (5.4) ตัวแปร  $e_g(p)$  คือค่าความผิดพลาดของนิวรอน  $g$  ในรอบที่  $p$  ระหว่างค่าเอาต์พุตเป้าหมาย  $y_{d,g}$  ที่กำหนดไว้กับค่าเอาต์พุตจริง  $y_g(p)$  ที่ได้จากการคำนวณและ  $p$  คือรอบการทำงานของโครงข่าย

- การคำนวณค่า weight corrections ( $\Delta w_{lg}(p)$ ) เพื่อใช้ในการปรับค่าน้ำหนักการเชื่อมต่อระหว่างนิวรอน  $l$  กับ  $g$  ดังสมการที่ (5.5)

$$\Delta w_{lg}(p) = \alpha \cdot y_l(p) \cdot \delta_g(p) \quad (5.5)$$

ซึ่ง  $\alpha$  คืออัตราการเรียนรู้ (learning rate) ของโครงข่ายประสาทเทียม

- การปรับค่าน้ำหนักการเชื่อมต่อระหว่างนิวรอน  $l$  กับ  $g$  ดังสมการที่ (5.6)

$$w_{lg}(p+1) = w_{lg}(p) + \Delta w_{lg}(p) \quad (5.6)$$

- การคำนวณค่า error gradient สำหรับนิวรอน  $g$  ในชั้นซ่อน ( $\delta_g(p)$ ) ดังสมการที่ (5.7)

$$\delta_h(p) = y_h(p) \cdot [1 - y_h(p)] \sum_r \delta_r(p) \cdot w_{hr}(p) \quad (5.7)$$

ซึ่ง  $r$  คือหมายเลขนิวรอนใดๆ ชั้นถัดไปที่นิวรอน  $h$  ได้เชื่อมต่ออยู่ด้วย

- การคำนวณค่า weight corrections ( $\Delta w_{hr}(p)$ ) เพื่อใช้ในการปรับค่าน้ำหนักการเชื่อมต่อระหว่างนิวรอน  $h$  กับ  $r$  ดังสมการที่ (5.8)

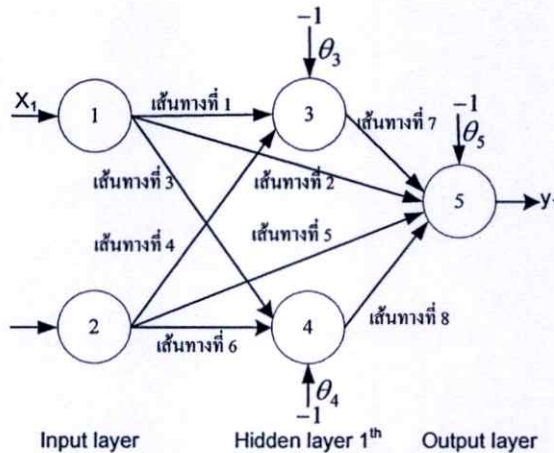
$$\Delta w_{hr}(p) = \alpha \cdot x_h(p) \cdot \delta_r(p) \quad (5.8)$$

- การปรับค่าน้ำหนักการเชื่อมต่อระหว่างนิวรอน  $h$  กับ  $r$  ดังสมการที่ (5.9)

$$w_{hr}(p+1) = w_{hr}(p) + \Delta w_{hr}(p) \quad (5.9)$$

## 5.2 ตารางพารามิเตอร์

ตารางพารามิเตอร์หมายถึงตารางที่เก็บรวบรวมข้อมูลเกี่ยวกับค่าพีโรโมน การเชื่อมต่อระหว่างนิวรอนและค่าน้ำหนักการเชื่อมต่อที่มีค่าแตกต่างกันไว้เพื่อใช้เป็นตัวเลือกสำหรับมดแต่ละตัวในระบบสำหรับนำมาสร้างเป็นโครงข่าย ในการพัฒนาหาโครงข่ายที่เหมาะสมนั้นจะต้องกำหนดลักษณะของโครงข่ายเริ่มต้นไว้ก่อน ได้แก่ จำนวนนิวรอนในชั้นอินพุต จำนวนชั้นซ่อน นิวรอนในชั้นซ่อน และนิวรอนในชั้นเอาต์พุต ซึ่งภายในโครงข่ายนั้นจะมีเส้นทาง (route) การเชื่อมต่อระหว่างนิวรอนเกิดขึ้นตัวอย่างแสดงดังรูปที่ 5.4



รูปที่ 5.4 เส้นทางภายในโครงข่ายประสาทเทียม

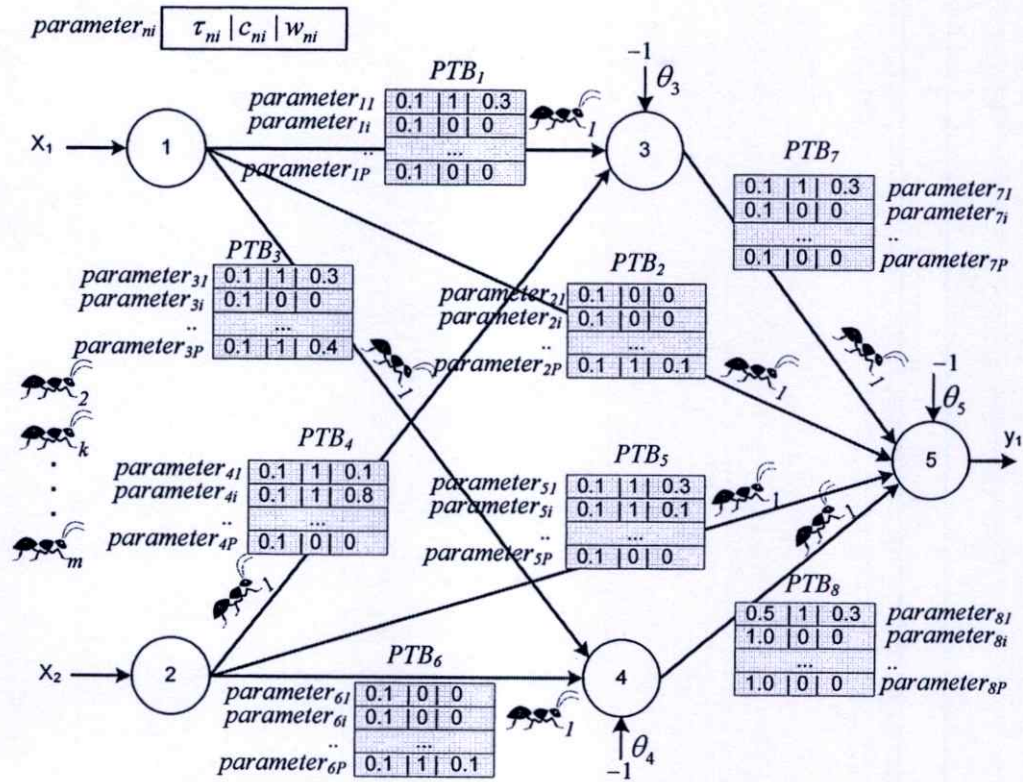
ในแต่ละตารางพารามิเตอร์เส้นทางที่  $n$  แทนด้วย  $PTB_n$

ซึ่ง  $PTB_n = \{ parameter_{ni}; i = 1, 2, \dots, P \}$

$parameter_{ni} = \langle \tau_{ni}, c_{ni}, w_{ni} \rangle$

เมื่อ	$parameter_{ni}$	หมายถึงพารามิเตอร์ซึ่งเป็นสมาชิกของ $PTB_n$
	$P$	หมายถึงจำนวนพารามิเตอร์ในตารางพารามิเตอร์
	$\tau_{ni}$	แทนค่าพีโรโมน
	$c_{ni}$	แทนการระบุการเชื่อมต่อระหว่างนิวรอน โดย $c_{ni} \in \{0,1\}$ ซึ่ง 0 หมายถึงการกำหนดให้ไม่มีการเชื่อมต่อระหว่างนิวรอนและ 1 หมายถึงกำหนดให้มีการเชื่อมต่อระหว่างนิวรอน
	$w_{ni}$	แทนค่าน้ำหนักการเชื่อมต่อ

ในรูปที่ 5.5 แสดงภาพรวมของ  $PTB_n$  ที่อยู่บนแต่ละเส้นทางและตารางที่ 5.1 แสดงตัวอย่างของค่า  $parameter_{ni}$  ที่เป็นสมาชิกของ  $PTB_n$



รูปที่ 5.5 ตารางพารามิเตอร์ ( $PTB_n$ ) ในแต่ละเส้นทาง

ตารางที่ 5.1 ตัวอย่างค่าพารามิเตอร์ ( $parameter_{ni}$ ) ใน  $PTB_n$

ลำดับพารามิเตอร์	ค่าฟีโรโมน ( $\tau_{ni}$ )	การเชื่อมต่อ ( $c_{ni}$ )	ค่าน้ำหนักการเชื่อมต่อ ( $w_{ni}$ )
$parameter_{n1}$	0.1	1	0.3
$parameter_{n2}$	0.1	1	-1.2
$parameter_{ni}$	0.1	0	0
...	...	...	...
$parameter_{np}$	0.1	1	0.4

จากรูปที่ 5.5 ในการทำงานนั้นจะให้มดแต่ละตัวพิจารณาเลือก  $parameter_{ni}$  ที่อยู่ใน  $PTB_n$  จากค่า  $\tau_{ni}$  มาจำนวน 1 พารามิเตอร์จนครบทุกเส้นทาง เพื่อนำมากำหนดลักษณะของโครงข่ายประสาทเทียม มดแต่ละตัวในระบบ ACS แทนด้วย  $k$  ซึ่ง  $k = 1, \dots, m$  โดย  $m$  แทนจำนวนมดทั้งหมดในระบบ

### 5.3 อัลกอริธึมการปรับเปลี่ยนรูปแบบการเชื่อมต่อของโครงข่ายประสาทเทียมด้วย อัลกอริธึมมด

ภาพรวมแสดงขั้นตอนการทำงานของวิธีที่นำเสนอแสดงได้ดังรูปที่ 5.6

```

T = จำนวนรอบการทำงานของ ACS      m = จำนวนของมดในระบบ
R = จำนวนเส้นทางในโครงข่าย        q0 คือค่าคงที่ (0 ≤ q0 ≤ 1) นิยม 0.9

กำหนดค่าเริ่มต้นของโครงข่าย และ PTBn
For (int t = 1; t <= T; t++)
  For (int k = 1; k <= m ; k++)
    If(t==1) Then // รอบการทำงานแรก
      For(int n = 1; n <= R; n++)
        มด kth เลือก parameternj ใน PTBn แบบสุ่ม;
      End For
    Else
      For(int n = 1; n <= R; n++)
        มด kth เลือก parameternj ใน PTBn จากระดับค่าฟีโรโมน โดยได้มาจาก
          q = ค่าสุ่มในช่วง (0,1);
          If(q ≤ q0) Then
            j = ลำดับ parameternj ใน PTBn ที่มีค่าฟีโรโมน τnj มากที่สุด;
          Else
            j = ลำดับ parameternj ใน PTBn ที่ได้จากสมการการส่งผ่านหรือ Transition rule ;
          End If
        End For
      Local pheromone update(ตำแหน่งพารามิเตอร์ใน PTB ทุกเส้นทาง ที่มด kth ได้เลือกมา) ;
    End If

    มด kth สร้างโครงข่ายจากค่าพารามิเตอร์ที่เลือกและฝึกสอนโครงข่ายแบบบางส่วน(training set, epochs) ;
    มด kth วัดค่าความผิดพลาดของโครงข่าย(validation set) ;

  End For

มด antBest = ค้นหาตัวที่มีค่าความผิดพลาดน้อยที่สุด(มดทุกตัวในระบบ);
Global pheromone update (ค่าความผิดพลาดของมด antBest, ทุกตำแหน่งพารามิเตอร์ที่มด antBest เลือกมา) ;
ปรับปรุงค่าถ่วงน้ำหนักในตารางพารามิเตอร์(โครงข่ายของ antBest, ตำแหน่งพารามิเตอร์ที่ antBest เลือกมา);
ปรับปรุงค่า threshold (θ) ในโครงข่าย(ค่า threshold ของโครงข่าย antBest )
End For

```

รูปที่ 5.6 อัลกอริธึมการปรับเปลี่ยนรูปแบบการเชื่อมต่อของโครงข่ายประสาทเทียมด้วยอัลกอริธึมมด

รายละเอียดขั้นตอนการทำงานและสมการคำนวณของ ACS สำหรับการพัฒนาโครงข่ายประสาทเทียมแสดงดังต่อไปนี้

**ขั้นตอนที่ 1** กำหนดค่าเริ่มต้นให้กับ  $parameter_{ni}$  ใน  $PTB_n$  ทุกๆ เส้นทางดังนี้

- สุ่มค่าการเชื่อมต่อ ( $c_{ni}$ ) ซึ่ง  $c_{ni} \in \{0,1\}$

ถ้าสุ่มได้  $c_{ni} = 1$  หมายถึงมีการเชื่อมต่อระหว่างนิวรอน จะกำหนดค่าน้ำหนัก ( $w_{ni}$ ) โดยสุ่มให้มีค่าอยู่ในช่วง  $[W_{min}, W_{max}]$

แต่ถ้าสุ่มได้  $c_{ni} = 0$  หมายถึงไม่มีการเชื่อมต่อระหว่างนิวรอน จะกำหนดให้ค่าน้ำหนักการเชื่อมต่อ ( $w_{ni}$ ) = 0

- ค่าพีโรโมน ( $\tau_{ni}$ ) มีค่าเท่ากับค่าพีโรโมนเริ่มต้น ( $\tau_0$ )

โดยค่า  $\tau_0 = 1/MSE_{first}$  ซึ่ง  $MSE_{first}$  เป็นความผิดพลาดแบบ MSE ของโครงข่ายเมื่อวัดกับชุดข้อมูล training set และชุดข้อมูล validation set รวมกัน โดยโครงข่ายนี้ได้มาจากการสุ่มเลือก  $parameter_{nj}$  จาก  $PTB_n$  มาสร้างเป็นโครงข่าย 1 โครงข่าย

**ขั้นตอนที่ 2** ในเส้นทางที่  $n$  มดตัวที่  $k$  พิจารณาเลือก  $parameter_{nj}$  จาก  $PTB_n$

ถ้าเป็นการทำงานรอบแรกของ ACS ให้เลือก  $parameter_{nj}$  แบบสุ่ม แต่ถ้าไม่ใช่ให้เลือก  $parameter_{nj}$  โดยใช้สมการที่ (5.10) และทำการเลือกพารามิเตอร์จนครบทุกเส้นทาง

$$j = \begin{cases} \arg \max_{1 \leq i \leq P} \{\tau_{ni}\} & \text{if } q \leq q_0; \\ J & \text{if } q > q_0; \end{cases} \quad (5.10)$$

โดย  $j$  คือลำดับของพารามิเตอร์ใน  $PTB_n$  ที่มดตัวที่  $k$  เลือกและ  $q_0$  คือค่าคงที่ ( $0 \leq q_0 \leq 1$ ) ส่วน  $q$  คือค่าสุ่มที่อยู่ในช่วง  $[0,1]$  (แต่ละเส้นทางจะสุ่มค่า  $q$  ใหม่ทุกครั้ง)

- ถ้า  $q \leq q_0$  มดตัวที่  $k$  จะเลือกพารามิเตอร์ลำดับที่  $j$  ซึ่งมีค่าพีโรโมนมากที่สุดใน  $PTB_n$
- แต่ถ้า  $q > q_0$  จะเลือกพารามิเตอร์ลำดับที่  $J$  ซึ่ง  $J$  คือลำดับพารามิเตอร์ที่ได้จากการสุ่มโดยอาศัยค่าความน่าจะเป็นของการถูกเลือกของแต่ละพารามิเตอร์แทนด้วย  $p_{ni}$  ซึ่งค่า  $p_{ni}$  ของแต่ละพารามิเตอร์ใน  $PTB_n$  คำนวณได้จากสมการที่ (5.11)

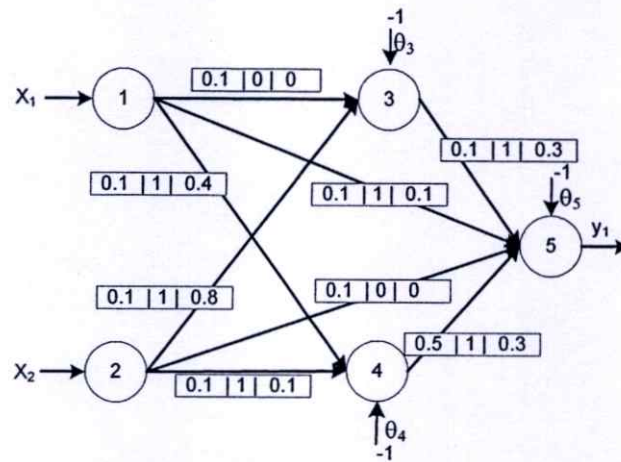
$$p_{ni} = \frac{\tau_{ni}}{\sum_{j=1}^P \tau_{nj}} \quad (5.11)$$

**ขั้นตอนที่ 3** มดตัวที่  $k$  ปรับปรุงค่าฟีโรโมนแบบเฉพาะบริเวณ (local pheromone update) เฉพาะพารามิเตอร์ลำดับที่  $j$  ที่ได้เลือกมาด้วยสมการที่ (5.12) ใน  $PTB_n$  ทุกๆ เส้นทาง

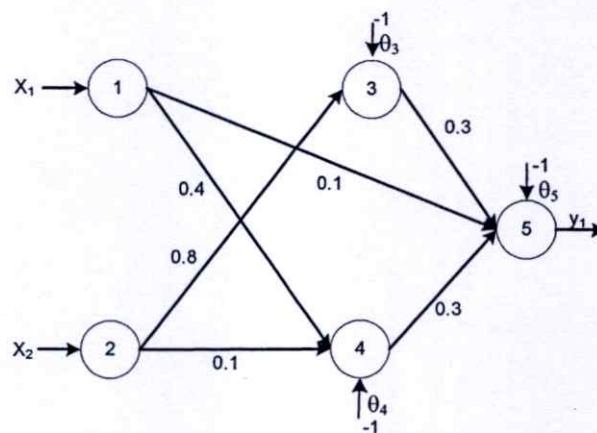
$$\tau_{nj} = (1 - \varphi) \cdot \tau_{nj} + \varphi \cdot \tau_0 \quad (5.12)$$

โดย  $\varphi \in (0,1]$  คือสัมประสิทธิ์การเสื่อมสลายของฟีโรโมนแบบเฉพาะบริเวณ

**ขั้นตอนที่ 4** มดตัวที่  $k$  นำการเชื่อมต่อ  $c_{nj}$ , ค่าน้ำหนัก  $w_{nj}$  จากพารามิเตอร์ที่เลือกมาและค่า threshold ( $\theta$ ) มาประกอบกันสร้างเป็นโครงข่ายประสาทเทียมดังรูปที่ 5.7



(ก) พารามิเตอร์ที่เลือกโดยมดตัวที่  $k$



(ข) โครงข่ายของมดตัวที่  $k$

รูปที่ 5.7 การสร้างโครงข่ายประสาทเทียมของมดตัวที่  $k$

นำโครงข่ายที่ได้ไปทำการฝึกสอนด้วย BP แบบบางส่วนด้วยจำนวนรอบค่าหนึ่งกับชุดข้อมูลฝึกสอน (training set) จากนั้นวัดค่าความผิดพลาด  $MSE_k$  (Mean of Square Error) จากสมการที่ 5.13

$$MSE_k = MSE_{training\ set} + MSE_{validation\ set} \quad (5.13)$$

โดย  $MSE_{training\ set}$  คือค่า MSE ของโครงข่ายเมื่อวัดกับชุดข้อมูล training set (ป้องกันการเกิดปัญหา local minimum) และ  $MSE_{validation\ set}$  คือค่า MSE ของโครงข่ายเมื่อวัดกับชุดข้อมูล validation set (ป้องกันการเกิดปัญหา overfitting)

**ขั้นตอนที่ 5** ทำซ้ำขั้นตอนที่ 2 – 4 จนครบมดทุกตัว (จำนวน  $m$  ตัว)

**ขั้นตอนที่ 6** มดตัวที่มีค่า  $MSE$  ที่น้อยที่สุด ( $MSE_{best}$ ) และทำการปรับปรุงค่าฟีโรโมนแบบครอบคลุมทั้งระบบ (global pheromone update) ในพารามิเตอร์ลำดับที่  $j$  ในทุกเส้นทางที่  $n$  ที่มดตัวนั้นได้เลือกมาโดยใช้สมการที่ 5.14

$$\tau_{nj} \leftarrow \begin{cases} (1 - \rho) \cdot \tau_{nj} + \rho \cdot \Delta\tau_{nj} & \text{กรณี } j \text{ คือลำดับของพารามิเตอร์ที่ถูกเลือก} \\ & \text{โดยมดตัวที่มีค่า } MSE_{best}, \\ \tau_{nj} & \text{กรณีอื่นๆ} \end{cases} \quad (5.14)$$

โดย  $\rho \in (0,1]$  คือค่าสัมประสิทธิ์การเสื่อมสลายของฟีโรโมนแบบครอบคลุมทั้งระบบ และ  $\Delta\tau_{nj} = 1 / MSE_{best}$

และมดตัวที่มีค่าความผิดพลาดที่ดีที่สุดจะนำค่าน้ำหนักที่ผ่านการฝึกสอนด้วย BP มาแทนที่ค่าเดิมในพารามิเตอร์ลำดับที่  $j$  ที่ได้เลือกมาและค่า threshold ของมดนั้นก็จะเป็นค่า threshold ใหม่ของโครงข่ายสำหรับรอบการทำงานถัดไป

**ขั้นตอนที่ 7** ทำซ้ำขั้นตอนที่ 2 – 6 จนครบตามจำนวนรอบการทำงาน ( $T$ ) ของ ACS

**ขั้นตอนที่ 8** เลือกใช้ค่าพารามิเตอร์และค่า threshold ของมดตัวที่มีค่าความผิดพลาดที่ดีที่สุด ( $MSE_{best}$ ) ในรอบสุดท้ายของวิธี ACS มาสร้างเป็นโครงข่ายและนำไปทดสอบประสิทธิภาพกับชุดข้อมูลทดสอบ

สรุปในบทนี้ได้อธิบายถึงรายละเอียดเกี่ยวกับวิธีการพัฒนาโครงข่ายประสาทเทียมที่วิทยานิพนธ์นี้ได้นำเสนอ ซึ่งเป็นการนำเทคนิควิธีระบบอาณานิคมมด (Ant Colony System: ACS) มาทำการพัฒนาเพื่อค้นหาลักษณะของโครงข่ายที่เหมาะสมทั้งในส่วนของการเชื่อมต่อภายในและค่าถ่วงน้ำหนักของโครงข่ายไปพร้อมๆ กัน และจะทำงานร่วมกับการเรียนรู้ด้วย BP แบบบางส่วน (partial learning) และในบทถัดไปจะกล่าวถึงการทดสอบประสิทธิภาพของโครงข่ายประสาทเทียมได้จากวิธีที่วิทยานิพนธ์นี้ได้เสนอซึ่งจะเปรียบเทียบกับโครงข่ายที่ได้จากวิธี ACO-BP จากงานวิจัยของ Y.P. Liu [8] และ ANN มาตรฐานที่ฝึกสอนด้วย BP

## บทที่ 6

### ผลการทดลอง

ในบทนี้จะกล่าวถึงการทดลองเพื่อวัดประสิทธิภาพของโครงข่ายประสาทเทียมระหว่างวิธีที่งานวิจัยนี้นำเสนอ โดยเปรียบเทียบกับอีก 3 วิธีคือ วิธี ANN มาตรฐานที่ฝึกสอนด้วย BP, วิธี ANN มาตรฐานที่ฝึกสอนด้วย BP ร่วมกับเทคนิค Early stopping และวิธี ACO-BP (รายละเอียดการทำงานอยู่ในบทที่ 2) ในการทดลองเปรียบเทียบกันระหว่างแต่ละวิธีนั้นจะทำการวัดประสิทธิภาพของโครงข่ายจาก 2 หลักเกณฑ์คือ ค่าความคลาดเคลื่อนเฉลี่ยยกกำลังสอง (Mean of Square Error: MSE) ซึ่งวัดจากชุดข้อมูลฝึกสอน (training set), ชุดข้อมูลประเมินประสิทธิภาพ (validation set) และชุดข้อมูลทดสอบ (test set) และอัตราความผิดพลาดในการระบุประเภทข้อมูล (Classification Error rate) กับชุดข้อมูลทดสอบ ผู้วิจัยได้นำวิธีการดังกล่าวมาพัฒนาขึ้นด้วยภาษา C++ และทำการทดสอบกับชุดข้อมูลมาตรฐานเกี่ยวกับการจำแนกประเภทข้อมูล (Classification problem)

#### 6.1 ชุดข้อมูลที่ใช้ในการทดลอง

ในการทดลองของงานวิจัยนี้ได้เลือกใช้ชุดข้อมูลมาตรฐาน PROBEN1 โดยนำข้อมูลมาจาก <ftp://ftp.ira.uka.de/pub/neuron/proben1.tar.gz> โดยได้เลือกใช้ชุดข้อมูลจำนวน 3 ชุดข้อมูล ได้แก่ Cancer1, Diabetes1 และ Glass1 ซึ่งเป็นชุดข้อมูลเกี่ยวกับการจำแนกประเภทข้อมูล (Classification problem) โดยทุกชุดข้อมูลมาตรฐาน PROBEN1 แต่ละปัญหาจะถูกแบ่งออกเป็นสามส่วนคือ ส่วนแรกเป็นชุดข้อมูล training set จะใช้ข้อมูลจำนวน 50% แรกกำหนดเป็น ข้อมูลสำหรับฝึกสอนโครงข่าย ส่วนที่สองเป็นข้อมูล validation set จะใช้ข้อมูลจำนวน 25% ต่อมา กำหนดใช้วัดประสิทธิภาพของโครงข่ายระหว่างทำการฝึกสอนและส่วนที่สามเป็นชุดข้อมูล test set จะใช้ข้อมูลจำนวน 25% สุดท้ายกำหนดเป็นข้อมูลที่ใช้สำหรับทดสอบโครงข่าย

ซึ่งแต่ละชุดข้อมูลมีรายละเอียดดังต่อไปนี้

1) ชุดข้อมูล Cancer1 เป็นข้อมูลเกี่ยวกับปัญหาการวินิจฉัยการเกิดโรคมะเร็ง ลักษณะของข้อมูล Cancer1 ประกอบด้วย 9 อินพุตและ 2 เอาท์พุต มีข้อมูลทั้งหมด 699 รูปแบบ แบ่งออกเป็น

- ข้อมูล training set ใช้ส่วนของข้อมูล 350 รูปแบบแรก
- ข้อมูล validation set ใช้ส่วนของข้อมูล 175 รูปแบบถัดมา
- ข้อมูล test set ใช้ส่วนของข้อมูล 174 รูปแบบสุดท้าย

2) ชุดข้อมูล Diabetes1 เป็นข้อมูลเกี่ยวกับปัญหาการวินิจฉัยการเกิดโรคเบาหวาน ลักษณะของข้อมูล Diabetes1 ประกอบด้วย 8 อินพุตและ 2 เอาท์พุต มีข้อมูลทั้งหมด 768 รูปแบบ แบ่งออกเป็น

- ข้อมูล training set ใช้ส่วนของข้อมูล 384 รูปแบบแรก
- ข้อมูล validation set ใช้ส่วนของข้อมูล 192 รูปแบบถัดมา
- ข้อมูล test set ใช้ส่วนของข้อมูล 192 รูปแบบสุดท้าย

3) ชุดข้อมูล Glass1 เป็นข้อมูลเกี่ยวกับปัญหาการวิเคราะห์ห้องกระจกประกอบทางเคมีชิ้นส่วนแก้ว ลักษณะของข้อมูล Glass1 ประกอบด้วย 9 อินพุตและ 6 เอาท์พุต มีข้อมูลทั้งหมด 214 รูปแบบ แบ่งออกเป็น

- ข้อมูล training set ใช้ส่วนของข้อมูล 107 รูปแบบแรก
- ข้อมูล validation set ใช้ส่วนของข้อมูล 54 รูปแบบถัดมา
- ข้อมูล test set ใช้ส่วนของข้อมูล 53 รูปแบบสุดท้าย

## 6.2 ข้อกำหนดการทดลอง

สำหรับการทดลองจะทดลองแต่ละวิธีเป็นจำนวน 10 ครั้งต่อชุดข้อมูล โดยทดสอบกับโครงข่ายแบบ 1 ชั้นซ่อนและ 2 ชั้นซ่อนและแต่ละชั้นซ่อนมีจำนวน 10 นิวรอน ซึ่งใช้ฟังก์ชันกระตุ้นแบบ *sigmoid*, อัตราการเรียนรู้ ( $\alpha$ ) เท่ากับ 0.01 และเกณฑ์การหยุดฝึกสอนโครงข่ายด้วยเทคนิค Early stopping จะใช้  $GL_s, P_k(t) < 0.1 \times 10^{-3}$  และมีจำนวนรอบการฝึกสอนด้วย BP เท่ากับ 3,000 รอบ ซึ่งใช้ตามงานวิจัย L. Prechelt [9]

1) วิธี ANN มาตรฐานที่ฝึกสอนด้วย BP การทดลองโดยวิธีนี้มีวัตถุประสงค์เพื่อวัดประสิทธิภาพของโครงข่ายที่ใช้การฝึกสอนด้วย BP เพียงอย่างเดียว โดยมีจำนวนรอบการฝึกสอนด้วย BP เท่ากับ 3,000 รอบ

2) วิธี ANN มาตรฐานที่ฝึกสอนด้วย BP ร่วมกับเทคนิค Early stopping โดยใช้เกณฑ์การหยุดฝึกสอนโครงข่ายตามงานวิจัย L. Prechelt [9]

3) วิธี ACO-BP ของงานวิจัย Y.P. Liu [8] ได้กำหนดให้ค่าพารามิเตอร์ของวิธี ACO-BP ดังนี้ ค่าแบ่งส่วนจำนวนตัวเลือก ( $m$ ) = 59 จะได้จำนวนพารามิเตอร์ที่ใช้เป็นตัวเลือก = 60, จำนวนมด = 30, ช่วงค่าน้ำหนักการเชื่อมต่อในพารามิเตอร์  $[W_{min}, W_{max}] = [-2, 2]$ , จำนวนรอบการทำงานของวิธี ACO ( $T$ ) = 100 โครงข่ายที่เหมาะสมที่ได้จากวิธี ACO จะนำไปฝึกสอนด้วย BP ร่วมกับเทคนิค Early stopping อีกครั้งโดยใช้เกณฑ์การหยุดฝึกสอนโครงข่ายตามงานวิจัย L. Prechelt [9] เช่นเดียวกัน

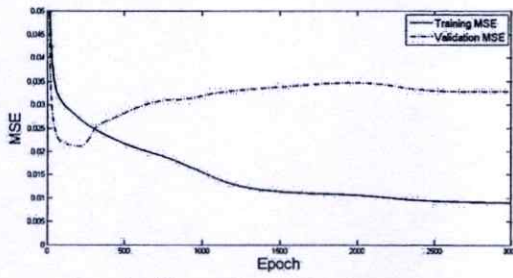
4) วิธีที่งานวิจัยนี้นำเสนอจะกำหนดให้ใช้ค่าพารามิเตอร์ต่างๆ เช่นเดียวกับวิธี ACO-BP และให้มีจำนวนพารามิเตอร์ ( $P$ ) = 60 ตามวิธี ACO-BP ของงานวิจัย Y.P. Liu [8] และใช้จำนวนรอบสำหรับฝึกสอนโครงข่ายที่ได้มาจากมดแต่ละตัวแบบบางส่วนด้วย BP จำนวนรอบ 10 รอบ สำหรับค่าพารามิเตอร์ของอัลกอริทึมมดในวิธี ACO-BP และ ACS ของวิธีที่นำเสนอ จะใช้ตามงานวิจัย M. Dorigo [17] ซึ่งกำหนดให้ค่าสัมประสิทธิ์ของการเสื่อมสลายของฟีโรโมนในระดับเฉพาะบริเวณ ( $\rho$ ) และครอบคลุมทั้งระบบ ( $\rho$ ) มีค่าเท่ากับ 0.1 ทั้งสองค่าและ  $q_0$  กำหนดให้เท่ากับ 0.9 (รายละเอียดการกำหนดค่า  $q_0$  อยู่ในบทที่ 4) โดยการทำงานนั้นจะใช้จำนวนมด  $m$  ตัว ซึ่งจะกำหนดให้มีค่าคงที่ตลอดการคำนวณ ซึ่ง E. Bonabeau [14] ได้ระบุไว้ว่าสำหรับปัญหา TSP นั้นจำนวนมดที่เหมาะสมควรมีจำนวนเท่ากับจำนวนเมืองหรือจำนวนตัวเลือกของปัญหา หากมีจำนวนมดมากเกินไปจะทำให้ใช้เวลาในการคำนวณมาก ดังนั้นวิธีที่นำเสนอจะทำการทดลองโดยใช้จำนวนมด ( $m$ ) ไม่เกินจำนวนพารามิเตอร์ ( $P$ ) ทำการทดลองโดยใช้จำนวนมดที่แตกต่างกันคือ 15, 30 และ 60 ตัว ในการทำงานแต่ละรอบ

### 6.3 ผลการทดลอง

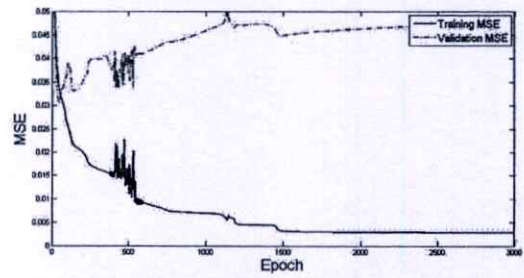
#### 6.3.1 กราฟแสดงค่า MSE ของโครงข่ายในระหว่างการทำงาน

ผลการฝึกสอนโครงข่ายจาก 3 วิธีคือ วิธี ANN มาตรฐานที่ฝึกสอนด้วย BP, วิธี ANN มาตรฐานที่ฝึกสอนด้วย BP ร่วมกับเทคนิค Early stopping และวิธี ACO-BP จากกราฟในรูปที่ 6.1, 6.2 และ 6.3 แสดงค่า MSE ของโครงข่ายเมื่อทดสอบกับชุดข้อมูล Cancer1, Diabetes1 และ Glass1 ตามลำดับ พบว่าโครงข่ายที่ทำการฝึกสอนด้วย BP เพียงอย่างเดียวมีค่า Training MSE (ค่าความคลาดเคลื่อนเฉลี่ยยกกำลังสองวัดกับชุดข้อมูล training set) ที่ได้นั้นลักษณะกราฟมีค่าลดลงไปเรื่อยๆ โดยมีค่าที่น้อยกว่าอีก 2 วิธีอย่างเห็นได้ชัด แต่เมื่อพิจารณาค่า Validation MSE (ค่าความคลาดเคลื่อนเฉลี่ยยกกำลังสองวัดกับชุดข้อมูล validation set) ลักษณะกราฟกลับมีแนวโน้มที่ค่าเพิ่มมากขึ้นซึ่งสวนทางกันกับค่า Training MSE ซึ่งแสดงให้เห็นว่าโครงข่ายที่ทำการฝึกสอนด้วย BP เพียงอย่างเดียวจะเกิดปัญหา Overfitting ขึ้นมาได้

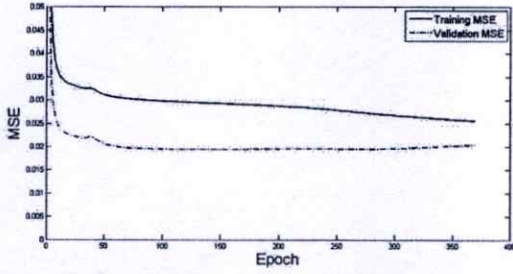
ส่วนวิธี ANN มาตรฐานที่ฝึกสอนด้วย BP ร่วมกับเทคนิค Early stopping และวิธี ACO-BP ผลการทดลองพบว่ากราฟค่า Training MSE และค่า Validation MSE ก็จะมีลักษณะเดียวกันกับวิธี ANN มาตรฐานที่ฝึกสอนด้วย BP เพียงอย่างเดียวแต่ว่าโครงข่ายที่นำมาใช้งานนั้นเป็นโครงข่ายที่ถูกบันทึกไว้ ณ ตำแหน่งที่ค่า Validation MSE มีค่าต่ำสุดในระหว่างการฝึกสอนซึ่งแตกต่างจากวิธี ANN มาตรฐานที่ฝึกสอนด้วย BP โดยในวิธี ACO-BP ค่า MSE ของโครงข่ายที่เหมาะสมที่ได้จากวิธี ACO นั้นเมื่อนำมาฝึกสอนด้วย BP จะพบว่าค่า MSE ในรอบแรกของการฝึกสอนจะมีค่าที่ต่ำกว่าวิธี ANN มาตรฐานที่ฝึกสอนด้วย BP และ วิธี ANN มาตรฐานที่ฝึกสอนด้วย BP ร่วมกับเทคนิค Early stopping



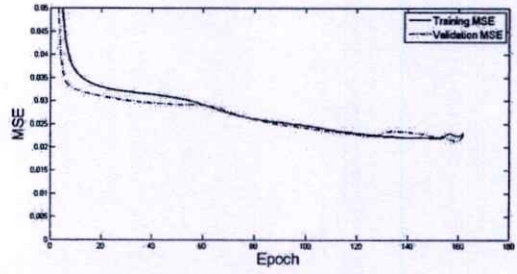
(ก) โครงข่ายที่ฝึกสอนด้วย BP โครงข่าย 9-10-2



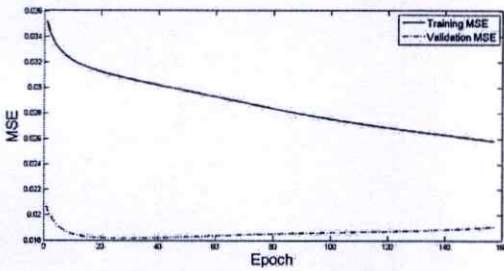
(ข) โครงข่ายที่ฝึกสอนด้วย BP โครงข่าย 9-10-10-2



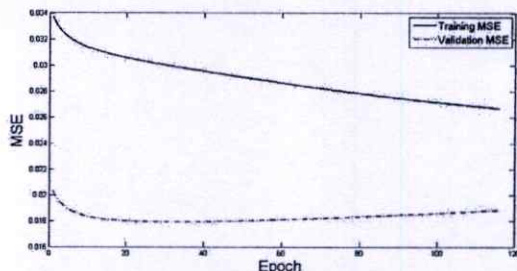
(ค) โครงข่ายที่ฝึกสอนด้วย BP ร่วมกับเทคนิค Early Stopping  
โครงข่าย 9-10-2



(ง) โครงข่ายที่ฝึกสอนด้วย BP ร่วมกับเทคนิค Early Stopping  
โครงข่าย 9-10-10-2

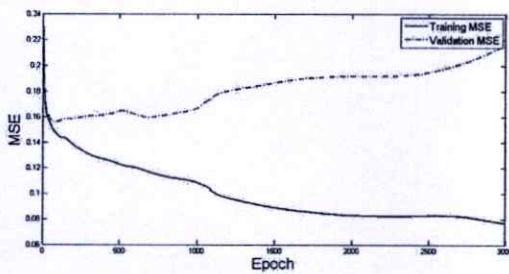


(จ) โครงข่ายจากวิธี ACO-BP โครงข่าย 9-10-2

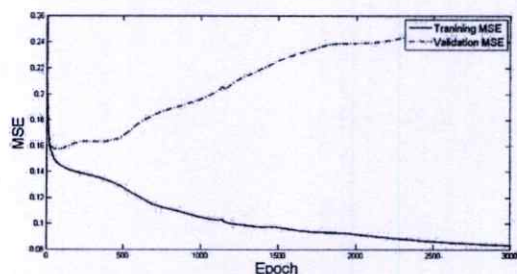


(ฉ) โครงข่ายจากวิธี ACO-BP โครงข่าย 9-10-10-2

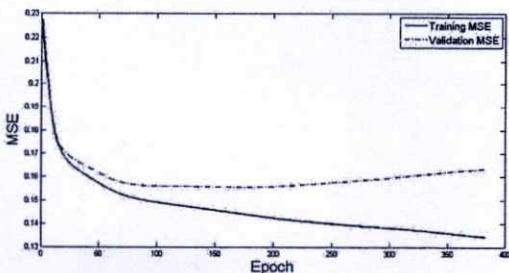
รูปที่ 6.1 กราฟแสดงค่า MSE ของโครงข่าย ทดสอบกับชุดข้อมูล Cancer1



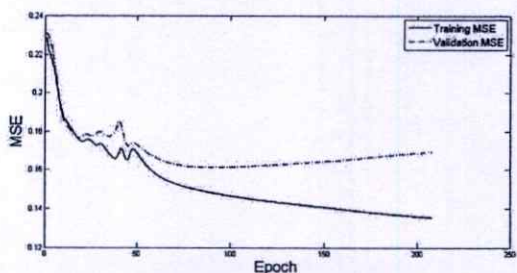
(ก) โครงข่ายที่ฝึกสอนด้วย BP โครงข่าย 9-10-2



(ข) โครงข่ายที่ฝึกสอนด้วย BP โครงข่าย 9-10-10-2

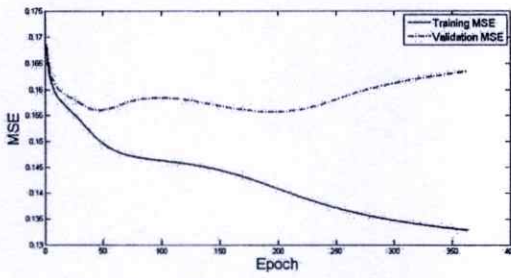


(ค) โครงข่ายที่ฝึกสอนด้วย BP ร่วมกับเทคนิค Early Stopping  
โครงข่าย 9-10-2

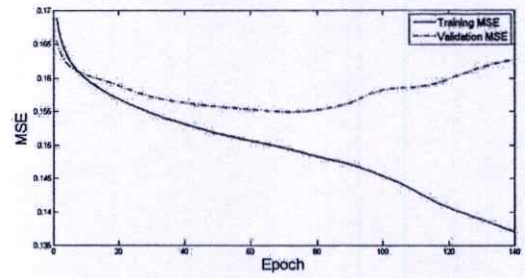


(ง) โครงข่ายที่ฝึกสอนด้วย BP ร่วมกับเทคนิค Early Stopping  
โครงข่าย 9-10-10-2

รูปที่ 6.2 กราฟแสดงค่า MSE ของโครงข่าย ทดสอบกับชุดข้อมูล Diabetes1

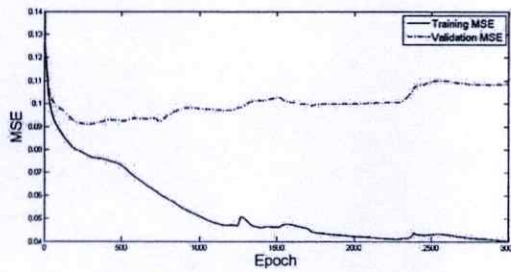


(จ) โครงข่ายจากวิธี ACO-BP โครงข่าย 9-10-2

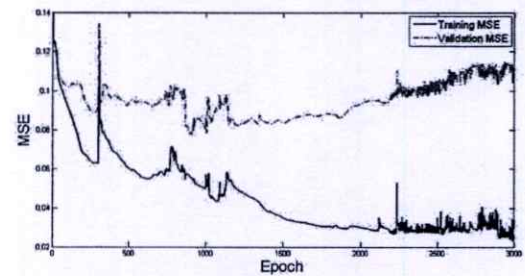


(ค) โครงข่ายจากวิธี ACO-BP โครงข่าย 9-10-10-2

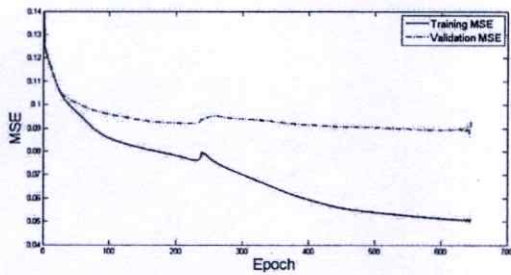
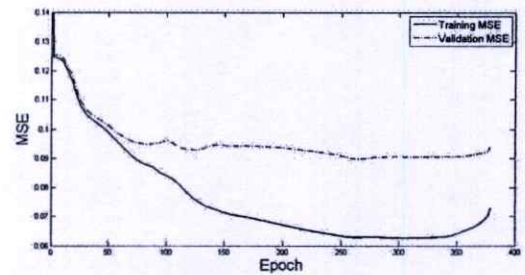
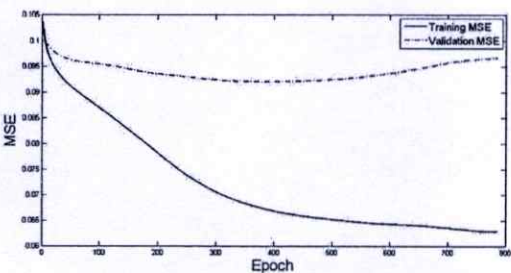
รูปที่ 6.2 กราฟแสดงค่า MSE ของโครงข่าย ทดสอบกับชุดข้อมูล Diabetes1 (ต่อ)



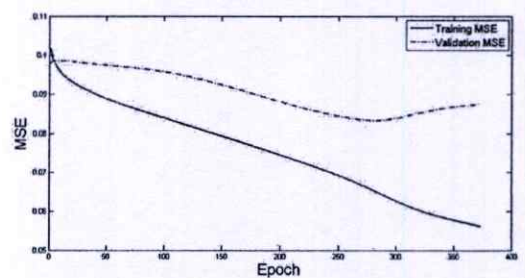
(ก) โครงข่ายที่ฝึกสอนด้วย BP โครงข่าย 9-10-2



(ข) โครงข่ายที่ฝึกสอนด้วย BP โครงข่าย 9-10-10-2

(ค) โครงข่ายที่ฝึกสอนด้วย BP ร่วมกับเทคนิค Early Stopping  
โครงข่าย 9-10-2(ง) โครงข่ายที่ฝึกสอนด้วย BP ร่วมกับเทคนิค Early Stopping  
โครงข่าย 9-10-10-2

(จ) โครงข่ายจากวิธี ACO-BP โครงข่าย 9-10-2

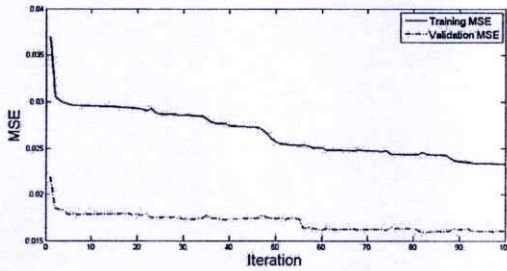
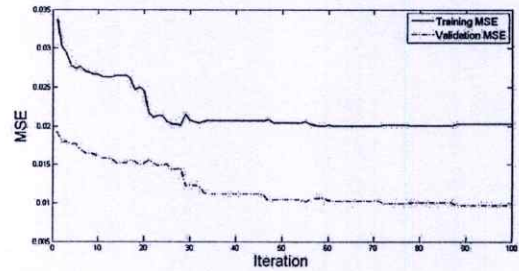
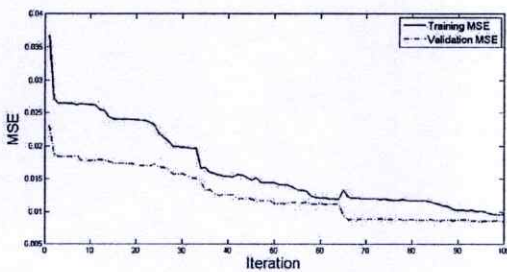
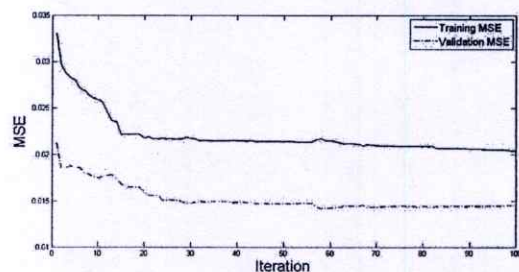
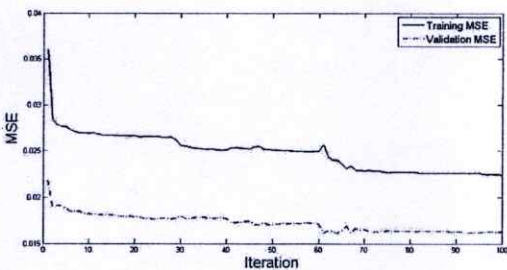
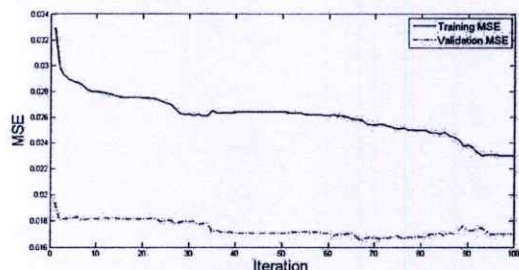


(ค) โครงข่ายจากวิธี ACO-BP โครงข่าย 9-10-10-2

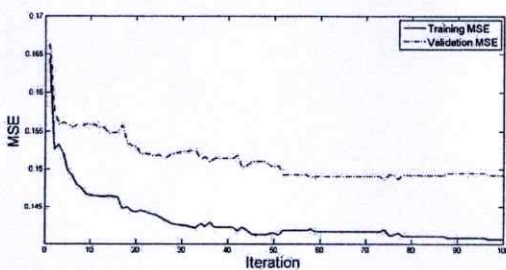
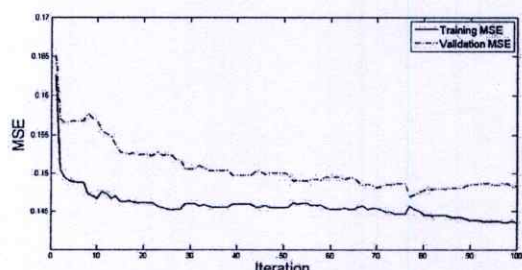
รูปที่ 6.3 กราฟแสดงค่า MSE ของโครงข่าย ทดสอบกับชุดข้อมูล Glass1

สำหรับวิธีที่งานวิจัยนี้นำเสนอ นั้นจะแสดงกราฟค่า MSE ของโครงข่ายที่ดีที่สุดที่ได้มาจากการทำงานในส่วนของ ACS ในแต่ละรอบ ซึ่งแตกต่างจากอีก 3 วิธีข้างต้นที่ค่า MSE ของโครงข่ายในแต่ละรอบการฝึกสอนด้วย BP โดยแสดงดังรูปที่ 6.4 - 6.6 จากผลการทดลองแสดงได้ว่าโครงข่ายประสาทเทียมที่ได้จากวิธีการที่นำเสนอ นั้นโครงข่ายที่ดีที่สุดที่ได้จาก ACS ในแต่ละ

รอบนั้นพบว่าโครงข่ายที่ได้มานั้นมีค่า Training MSE และ Validation MSE ที่ลดลงเรื่อยๆ เนื่องจากในการเลือกโครงข่ายนั้นจะพิจารณาเลือกโครงข่ายที่มีผลการฝึกสอนด้วยชุดข้อมูล training set และมีผลการประเมินด้วยชุดข้อมูล validation set ที่ดีทั้งสองชุดข้อมูล

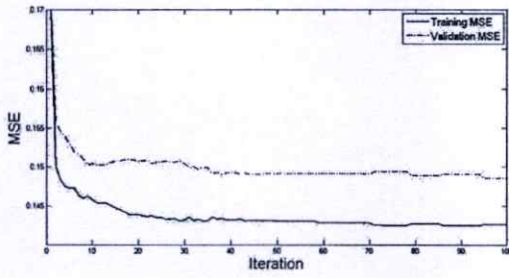
(ก)  $m = 15$ , โครงข่าย 9-10-2(ข)  $m = 15$ , โครงข่าย 9-10-10-2(ค)  $m = 30$ , โครงข่าย 9-10-2(ง)  $m = 30$ , โครงข่าย 9-10-10-2(จ)  $m = 60$ , โครงข่าย 9-10-2(ฉ)  $m = 60$ , โครงข่าย 9-10-10-2

รูปที่ 6.4 กราฟแสดงค่า MSE ของโครงข่ายที่ได้มาจากวิธีที่นำเสนอ ทดสอบกับชุดข้อมูล Cancer1

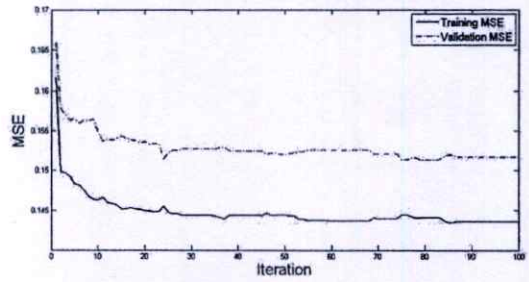
(ก)  $m = 15$ , โครงข่าย 9-10-2(ข)  $m = 15$ , โครงข่าย 9-10-10-2

รูปที่ 6.5 กราฟแสดงค่า MSE ของโครงข่ายที่ได้มาจากวิธีที่นำเสนอ ทดสอบกับชุดข้อมูล

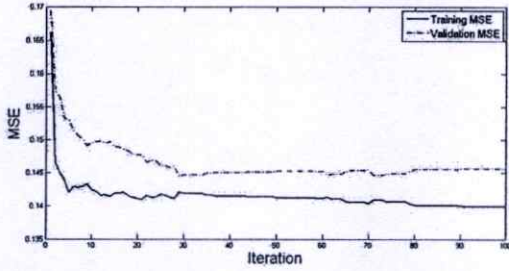
Diabetes1



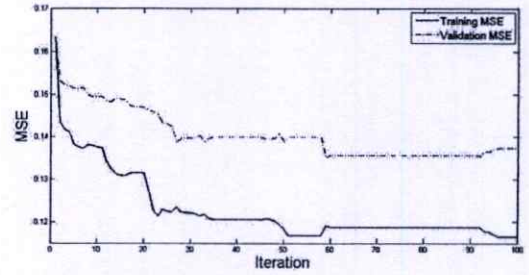
(ค)  $m = 30$ , โครงข่าย 9-10-2



(ง)  $m = 30$ , โครงข่าย 9-10-10-2

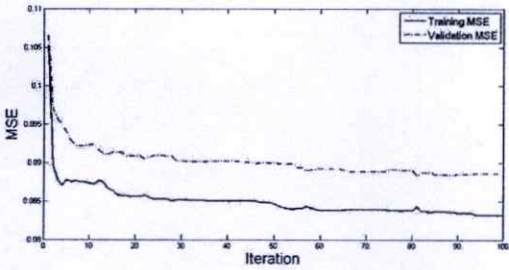


(จ)  $m = 60$ , โครงข่าย 9-10-2

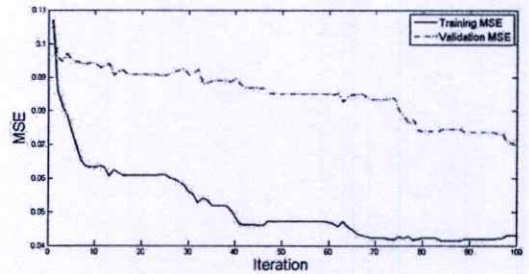


(ฉ)  $m = 60$ , โครงข่าย 9-10-10-2

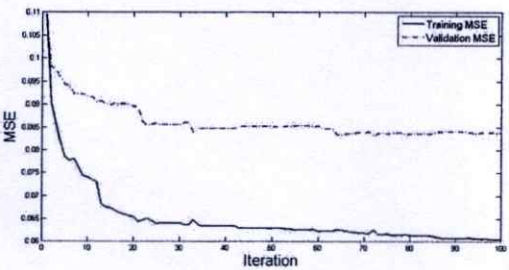
รูปที่ 6.5 กราฟแสดงค่า MSE ของโครงข่ายที่ได้มาจากวิธีที่นำเสนอ ทดสอบกับชุดข้อมูล Diabetes1 (ต่อ)



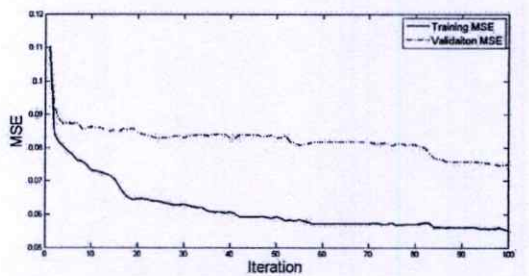
(ก)  $m = 15$ , โครงข่าย 9-10-2



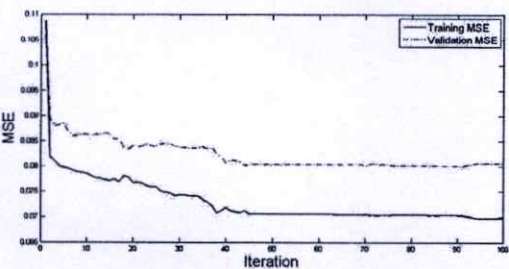
(ข)  $m = 15$ , โครงข่าย 9-10-10-2



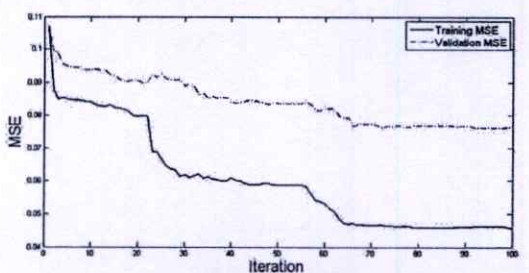
(ค)  $m = 30$ , โครงข่าย 9-10-2



(ง)  $m = 30$ , โครงข่าย 9-10-10-2



(จ)  $m = 60$ , โครงข่าย 9-10-2



(ฉ)  $m = 60$ , โครงข่าย 9-10-10-2

รูปที่ 6.6 กราฟแสดงค่า MSE ของโครงข่ายที่ได้มาจากวิธีที่นำเสนอ ทดสอบกับชุดข้อมูล Glass1

### 6.3.2 ผลการทดลองประสิทธิภาพของโครงข่ายประสาทเทียม

ในส่วนนี้เป็นการแสดงผลของการวัดประสิทธิภาพของโครงข่ายเปรียบเทียบกับกัน โดยวัดจากค่าความคลาดเคลื่อนเฉลี่ยยกกำลังสอง (Mean of Square Error: MSE) ซึ่งวัดจากชุดข้อมูลฝึกสอน (training set), ชุดข้อมูลประเมินประสิทธิภาพ (validation set) และชุดข้อมูลทดสอบ (test set) และอัตราความผิดพลาดในการระบุประเภทข้อมูล (Classification Error rate) กับชุดข้อมูลฝึกสอน ผลการทดลองกับชุดข้อมูล Cancer1, Diabetes1 และ Glass1 แสดงได้ดังตารางที่ 6.1, 6.2 และ 6.3 ตามลำดับ

จากผลการทดลองแสดงพบว่าวิธี ANN มาตรฐานที่ฝึกสอนด้วย BP มีค่าเฉลี่ยของ training MSE ที่น้อยที่สุดเมื่อเทียบกับวิธีการอื่นๆ ในทุกชุดข้อมูลทดลอง แต่วิธีนี้ก็กลับมีค่าเฉลี่ยของ validation MSE และ test MSE และมีค่าเฉลี่ยอัตราความผิดพลาดในการระบุประเภทข้อมูลที่มากที่สุด จากผลดังกล่าวแสดงว่าชุดข้อมูลมาตรฐานเกี่ยวกับการจำแนกประเภทข้อมูลเมื่อนำมาใช้งานกับโครงข่ายและฝึกสอนโครงข่ายด้วย BP เพียงอย่างเดียวมันจะพบกับปัญหา overfitting แม้ว่าโครงข่ายที่ได้จะมีค่า training MSE ที่ต่ำก็ตาม ทำให้โครงข่ายที่ได้จะมีประสิทธิภาพต่ำเมื่อทดสอบกับข้อมูลฝึกสอน

ผลการทดลองของวิธี ANN มาตรฐานที่ฝึกสอนด้วย BP ร่วมกับเทคนิค Early stopping นั้น ผลการฝึกสอนโครงข่ายที่ได้มีค่าเฉลี่ยของ training MSE ที่สูงกว่าการฝึกสอนโครงข่ายด้วย BP เพียงอย่างเดียว แต่ในทางกลับกันพบว่าวิธีนี้สามารถแก้ไขปัญหา overfitting ได้ โดยแสดงได้จากเฉลี่ยของ validation MSE และ test MSE และมีค่าเฉลี่ยอัตราความผิดพลาดในการระบุประเภทข้อมูลที่มีค่าน้อยกว่าในทุกชุดข้อมูลทดลอง

ผลการทดลองของวิธี ACO-BP ซึ่งวิธีนี้ได้ใช้ ACO ทำการค้นหาโครงข่ายที่เหมาะสมก่อนที่จะนำไปฝึกสอนด้วย BP ร่วมกับเทคนิค Early stopping อีกครั้ง ผลที่ได้มันพบว่ามีค่าเฉลี่ยของ validation MSE และ test MSE และมีค่าเฉลี่ยอัตราความผิดพลาดในการระบุประเภทข้อมูลที่น้อยกว่าอีกสองวิธีข้างต้น ส่วนค่าเฉลี่ยของ training MSE จะมีค่าที่สูงกว่าการฝึกสอนโครงข่ายด้วย BP เพียงอย่างเดียวแต่จะมีค่าที่น้อยกว่าวิธี ANN มาตรฐานที่ฝึกสอนด้วย BP ร่วมกับเทคนิค Early stopping ซึ่งแสดงให้เห็นว่าการฝึกสอนโครงข่ายด้วย BP นั้นการเลือกโครงข่ายเริ่มต้นที่ดีจะมีผลต่อประสิทธิภาพที่ดีของโครงข่ายเมื่อฝึกสอนเสร็จ

ส่วนในการพัฒนาหาโครงข่ายด้วยวิธีที่งานวิจัยนี้นำเสนอ ผลที่ได้พบว่าในทุกชุดข้อมูลทดลองสามารถได้โครงข่ายที่เหมาะสมซึ่งมีค่าเฉลี่ยของ validation MSE และ test MSE และมีค่าเฉลี่ยอัตราความผิดพลาดในการระบุประเภทข้อมูลที่น้อยกว่าอีกสามวิธีข้างต้น และมีค่าเฉลี่ยของ training MSE ที่น้อยกว่าวิธี ANN มาตรฐานที่ฝึกสอนด้วย BP ร่วมกับเทคนิค Early stopping และวิธี ACO-BP แต่ค่าเฉลี่ยของ training MSE จะมีค่าที่สูงกว่าวิธี ANN มาตรฐานที่ฝึกสอนด้วย BP เพียงอย่างเดียว เพราะให้การเลือกโครงข่ายวิธีที่นำเสนอจะพิจารณาเลือกโครงข่ายที่ค่า MSE ที่เหมาะสมทั้ง training MSE และ validation MSE

ตารางที่ 6.1 ผลการทดลองชุดกับข้อมูล Cancer1

โครงข่าย	วิธี	training MSE				validation MSE				test MSE				classification error (%)			
		max	min	mean	stdv	max	min	mean	stdv	max	min	mean	stdv	max	min	mean	stdv
9-10-2	1. BP	0.0185	0.0034	<b>0.0071</b>	0.00428	0.0441	0.0249	0.0352	0.00684	0.0503	0.0240	0.0329	0.00765	5.7471	2.8736	4.1954	0.90057
	2. BP +early stopping	0.0294	0.0228	0.0261	0.00173	0.0224	0.0196	0.0208	0.00111	0.0163	0.0113	0.0141	0.00173	2.8736	1.1494	1.7241	0.60583
	3. ACO-BP	0.0271	0.0228	0.0254	0.00121	0.0196	0.0173	0.0187	0.00069	0.0135	0.0105	0.0119	0.00076	1.7241	0.5747	1.2643	0.45333
	4. วิธีที่นำเสนอ																
	m = 15	0.0269	0.0146	0.0239	0.00348	0.0183	0.0147	<b>0.0170</b>	0.00106	0.0122	0.0084	<b>0.0103</b>	0.00110	1.7241	0.5747	<b>0.9770</b>	0.38789
	m = 30	0.0279	0.0185	0.0248	0.00282	0.0187	0.0140	0.0171	0.00152	0.0114	0.0098	0.0106	0.00053	1.7241	0.5747	1.2643	0.36347
	m = 60	0.0264	0.0213	0.0239	0.00144	0.0188	0.0172	0.0180	0.00052	0.0116	0.0086	0.0104	0.00089	1.7241	0.5747	1.0919	0.32622
	1. BP	0.0185	0.0029	<b>0.0089</b>	0.00565	0.0476	0.0189	0.0343	0.01111	0.0462	0.0293	0.0374	0.00586	4.5977	3.4483	4.3678	0.40183
	2. BP +early stopping	0.0319	0.0175	0.0271	0.00420	0.0263	0.0197	0.0228	0.00207	0.0243	0.0114	0.0176	0.00431	4.0230	1.7241	2.7012	0.71936
	3. ACO-BP	0.0284	0.0217	0.0262	0.00219	0.0195	0.0179	0.0188	0.00052	0.0128	0.0106	0.0117	0.00065	1.7241	0.5747	1.1494	0.38313
	4. วิธีที่นำเสนอ																
	m = 15	0.0272	0.0202	0.0250	0.00204	0.0184	0.0096	<b>0.0169</b>	0.00264	0.0142	0.0087	0.0107	0.00175	1.7241	0.5747	1.0345	0.45332
m = 30	0.0266	0.0195	0.0233	0.00259	0.0195	0.0145	0.0170	0.00167	0.0143	0.0078	<b>0.0101</b>	0.00184	1.1494	0.5747	<b>0.9195</b>	0.29677	
m = 60	0.0260	0.0205	0.0238	0.00183	0.0187	0.0159	0.0176	0.00075	0.0163	0.0084	0.0106	0.00216	1.7241	0.5747	1.0919	0.50320	

ตารางที่ 6.2 ผลการทดลองชุดกับข้อมูล Diabetes1

โครงข่าย	วิธี	training MSE			validation MSE			test MSE			classification error (%)						
		max	min	mean	stdv	max	min	mean	stdv	max	min	mean	stdv				
9-10-2	1. BP	0.0964	0.0769	<b>0.0881</b>	0.00560	0.2245	0.1948	0.2093	0.00998	0.2344	0.2023	0.2132	0.01370	27.6040	22.9170	25.0521	1.81141
	2. BP +early stopping	0.1502	0.1421	0.1460	0.00309	0.1584	0.1553	0.1566	0.00104	0.1711	0.1635	0.1674	0.00265	24.4790	20.3130	22.5002	1.03575
	3. ACO-BP	0.1532	0.1412	0.1473	0.00372	0.1579	0.1522	0.1557	0.00167	0.1684	0.1579	0.1627	0.00420	23.9580	21.8750	22.6043	0.74475
	4. วิธีที่นำเสนอ	0.1465	0.1370	0.1427	0.00297	0.1534	0.1456	0.1500	0.00260	0.1642	0.1579	0.1620	0.00210	23.4380	20.3120	<b>21.9792</b>	0.91237
9-10-2	1. BP	0.2084	0.0953	<b>0.1262</b>	0.03191	0.2303	0.1936	0.2141	0.01009	0.2512	0.1781	0.2148	0.02056	35.9380	23.4380	28.8542	4.29625
	2. BP +early stopping	0.1611	0.1320	0.1505	0.00876	0.1692	0.1584	0.1614	0.00320	0.1886	0.1625	0.1710	0.00685	27.6040	20.3120	23.2291	1.93636
	3. ACO-BP	0.1517	0.1432	0.1492	0.00241	0.1576	0.1535	0.1561	0.00134	0.1615	0.1563	0.1585	0.00164	23.9580	21.3540	22.5001	0.80693
	4. วิธีที่นำเสนอ	0.1480	0.1366	0.1438	0.00377	0.1506	0.1439	<b>0.1486</b>	0.00234	0.1675	0.1533	<b>0.1576</b>	0.00424	23.4380	20.3120	<b>22.1876</b>	0.92550
9-10-2	1. BP	0.1473	0.1239	0.1354	0.00777	0.1576	0.1427	0.1503	0.00434	0.1688	0.1529	0.1612	0.00496	23.4380	20.8330	22.3960	1.04200
	2. BP +early stopping	0.1379	0.1300	0.1358	0.00327	0.1570	0.1484	0.1531	0.00339	0.1635	0.1542	0.1582	0.00399	23.9580	20.8330	22.2916	1.18771
	3. ACO-BP	0.1473	0.1239	0.1354	0.00777	0.1576	0.1427	0.1503	0.00434	0.1688	0.1529	0.1612	0.00496	23.4380	20.8330	22.3960	1.04200
	4. วิธีที่นำเสนอ	0.1379	0.1300	0.1358	0.00327	0.1570	0.1484	0.1531	0.00339	0.1635	0.1542	0.1582	0.00399	23.9580	20.8330	22.2916	1.18771

ตารางที่ 6.3 ผลการทดลองชุดกับข้อมูล Glass1

โครงการ	วิธี	training MSE				validation MSE				test MSE				classification error (%)				
		max	min	mean	stdv	max	min	mean	stdv	max	min	mean	stdv	max	min	mean	stdv	
9-10-6	1. BP	0.0444	0.0323	<b>0.0396</b>	0.00460	0.1124	0.0818	0.1022	0.01207	0.0990	0.0753	0.0910	0.00894	41.5090	28.3020	33.5847	4.77308	
	2. BP +early stopping	0.0782	0.0483	0.0619	0.01086	0.0917	0.0777	0.0865	0.00437	0.0942	0.0755	0.0873	0.00568	35.8490	26.4150	32.4526	2.92294	
	3. ACO-BP	0.0697	0.0496	0.0600	0.00779	0.0923	0.0839	0.0885	0.00292	0.0984	0.0800	0.0872	0.00576	37.7360	28.3020	32.4527	2.63844	
	4. วิธีที่นำเสนอ																	
	m = 15	0.0832	0.0417	0.0669	0.01186	0.0886	0.0805	0.0844	0.00319	0.0914	0.0844	0.0870	0.00227	35.8490	26.4150	<b>30.7546</b>	2.95651	
	m = 30	0.0834	0.0467	0.0631	0.01111	0.0907	0.0720	<b>0.0823</b>	0.00567	0.0940	0.0796	<b>0.0842</b>	0.00466	35.8490	28.3020	32.2641	2.58536	
	m = 60	0.0775	0.0523	0.0637	0.00794	0.0888	0.0763	0.0827	0.00371	0.0926	0.0814	0.0869	0.00377	35.8490	28.3020	32.0753	2.81253	
	1. BP	0.0667	0.0324	<b>0.0512</b>	0.01254	0.1146	0.0942	0.1066	0.00602	0.1213	0.0868	0.0971	0.01047	47.1700	33.9620	37.9456	3.70759	
	2. BP +early stopping	0.0878	0.0466	0.0713	0.01525	0.0974	0.0864	0.0926	0.00416	0.1027	0.0789	0.0908	0.00753	41.5090	28.3020	34.9056	4.10009	
	3. ACO-BP	0.0725	0.0353	0.0551	0.01182	0.0958	0.0801	0.0873	0.00456	0.0896	0.0799	0.0854	0.00370	39.6230	28.3020	32.9140	3.78669	
	4. วิธีที่นำเสนอ																	
	m = 15	0.0763	0.0411	0.0579	0.01211	0.0876	0.0705	0.0796	0.00609	0.0935	0.0738	0.0846	0.00656	37.7360	28.3020	32.2642	2.87510	
m = 30	0.0665	0.0481	0.0540	0.00575	0.0828	0.0698	0.0783	0.00386	0.0896	0.0737	<b>0.0817</b>	0.00468	35.8490	28.3020	32.2641	2.87509		
m = 60	0.0598	0.0386	0.0512	0.00738	0.0816	0.0681	<b>0.0761</b>	0.00555	0.0897	0.0741	0.0827	0.00496	33.9620	26.4150	<b>31.3206</b>	2.54681		

## บทที่ 7

# สรุปผลงานวิจัยและข้อเสนอแนะ

### 7.1 สรุปผลการวิจัย

โครงข่ายประสาทเทียม (Artificial Neural Networks: ANN) นั้นได้รับความนิยมอย่างมากในงานทางด้านปัญญาประดิษฐ์เนื่องจากมีความสามารถในการเรียนรู้จากความรู้ต่างๆ ที่นำไปฝึกสอนให้ได้ แต่การฝึกสอนโครงข่ายด้วยการเรียนรู้แบบแพร่ย้อนกลับ (Backpropagation learning: BP) พบว่าบ่อยครั้งเกิดปัญหาการติดกับดักจุดต่ำสุดเฉพาะบริเวณ (local minimum) อันเป็นสาเหตุทำให้ไม่สามารถทำการฝึกสอนโครงข่ายได้สำเร็จ ทั้งนี้อาจเกิดจากการกำหนดรูปแบบการเชื่อมต่อภายในและค่าน้ำหนักเริ่มต้นที่ไม่เหมาะสมให้โครงข่ายจึงต้องทำการกำหนดโครงข่ายเริ่มต้นใหม่อีกครั้ง (trial-and-error) และปัญหาของการเข้ากันกับชุดข้อมูลฝึกสอนมากเกินไป (overfitting) เป็นปัญหาที่โครงข่ายนั้นมีประสิทธิภาพในการเรียนรู้ที่ดีมากทำให้สามารถในการระบุกลุ่มของข้อมูลฝึกสอนได้สูงแต่กลับมีความถูกต้องในการระบุกลุ่มข้อมูลทดสอบต่ำ เนื่องจากพิจารณาว่าเป็นสัญญาณรบกวน ซึ่งวิทยานิพนธ์นี้ได้เสนอแนวทางการนำอัลกอริทึมมด (Ant Algorithm) โดยเลือกใช้เทคนิควิธีระบบอาณานิคมมด (Ant Colony System : ACS) ซึ่งเป็นวิธีหนึ่งของอัลกอริทึมมดรวมกับการฝึกสอนโครงข่ายแบบบางส่วนด้วย BP มาใช้ในการพัฒนาหาโครงข่ายประสาทเทียมที่เหมาะสมทั้งการเชื่อมต่อและค่าน้ำหนักของนิวรอนระหว่างชั้น โดยใช้วิธี ACS ทำการค้นหาลักษณะของโครงข่ายที่เหมาะสม ซึ่งจุดเด่นของวิธีที่นำเสนอนี้คือ ACS สามารถปรับเปลี่ยนการเชื่อมต่อภายในและค่าน้ำหนักของโครงข่ายได้อย่างอัตโนมัติและได้ออกแบบให้นิวรอนนั้นสามารถเชื่อมต่อกันข้ามชั้นได้ สำหรับวิธีที่นำเสนอได้พิจารณาการเลือกโครงข่ายที่เหมาะสมในสองส่วน โดยมีผลในการฝึกสอนโครงข่ายที่ดีคือมีค่า training MSE ที่ต่ำเพื่อป้องกันปัญหาการติดกับดัก local minimum และต้องเป็นโครงข่ายมีผลในการประเมินผลประเมินประสิทธิภาพที่ดีคือมีค่า validation MSE ที่ต่ำเพื่อป้องกันการเกิดปัญหา overfitting

จากผลการทดลองกับชุดข้อมูลทดลอง Cancer1, Diabetes1 และ Glass1 ซึ่งเป็นชุดข้อมูลเกี่ยวกับการจำแนกประเภทข้อมูล (Classification problem) พบว่าวิธี ANN มาตรฐานที่ฝึกสอนด้วย BP นั้นพบกับปัญหา overfitting เกิดขึ้นคือมีค่าเฉลี่ยอัตราความผิดพลาดในการระบุประเภทข้อมูลที่มากเมื่อเปรียบเทียบกับวิธีการอื่นแม้ว่าจะสามารถฝึกสอนโครงข่ายให้มีค่าเฉลี่ยของ training MSE ที่น้อยที่สุดก็ตาม ส่วนวิธี ANN มาตรฐานที่ฝึกสอนด้วย BP ร่วมกับเทคนิค Early stopping และวิธี ACO-BP ก็สามารถแก้ไขการเกิดปัญหา overfitting ได้โดยดูได้จากค่าเฉลี่ยของ validation MSE และ test MSE และมีค่าเฉลี่ยอัตราความผิดพลาดในการระบุประเภทข้อมูลที่ต่ำกว่าทำให้ได้โครงข่ายที่มีประสิทธิภาพที่ดีกว่า แม้ค่าเฉลี่ยของ training MSE จะมีค่าที่มากกว่าวิธี

ANN มาตรฐานที่ฝึกสอนด้วย BP ก็ตาม เนื่องจากในระหว่างการฝึกสอน โครงข่ายเมื่อโครงเริ่มเกิดปัญหา overfitting ขึ้นก็ต้องการหยุดฝึกสอนโครงข่าย

ส่วนในการพัฒนาหาโครงข่ายด้วยวิธีที่งานวิจัยนี้นำเสนอ ผลที่ได้พบว่าในทุกการทดลองในใช้จำนวนมดที่แตกต่างกันกับทุกชุดข้อมูลทดลองสามารถได้โครงข่ายที่เหมาะสม โดยสามารถแก้ไขการเกิดปัญหา local minimum ในการฝึกสอนโครงข่ายจากวิธี ANN มาตรฐานที่ฝึกสอนด้วย BP ร่วมกับเทคนิค Early stopping และวิธี ACO-BP ได้ ซึ่งวัดค่าค่าเฉลี่ยของ training MSE ที่มีค่าน้อยกว่า ถึงแม้ผลการทดลองจะพบว่าค่าเฉลี่ยของ training MSE จากการฝึกสอนโครงข่ายจะยังมีค่ามากกว่าวิธี ANN มาตรฐานที่ฝึกสอนด้วย BP เพียงอย่างเดียว เนื่องจากวิธีที่นำเสนอจะต้องพิจารณาโครงข่ายที่มีความเหมาะสมทั้งการฝึกสอน โครงข่ายที่ดีคือมีค่า training MSE ที่ต่ำเพื่อป้องกันปัญหาการติดกับดัก local minimum และต้องเป็นโครงข่ายมีผลในการประเมินผลประเมินประสิทธิภาพที่ดีคือมีค่า validation MSE ที่ต่ำเพื่อป้องกันการเกิดปัญหา overfitting ไปพร้อมๆ กัน ซึ่งจากผลของวิธี ANN มาตรฐานที่ฝึกสอนด้วย BP ที่ได้ในช่วงต้นนั้นแสดงให้เห็นว่าแม้จะได้โครงข่ายที่มีการฝึกสอนคือมีค่า training MSE ที่ต่ำแต่กลับมีประสิทธิภาพในการระบุกลุ่มของข้อมูลทดสอบที่ต่ำมากกว่า และวิธีที่นำเสนอนั้นยังสามารถแก้ไขการเกิดปัญหา overfitting ได้เมื่อเปรียบเทียบกับอีกสามวิธีข้างต้น โดยวัดได้จากค่าเฉลี่ยของ validation MSE และ test MSE และมีค่าเฉลี่ยอัตราความผิดพลาดในการระบุประเภทข้อมูลที่ต่ำกว่าในทุกๆ การทดลอง

## 7.2 ข้อเสนอแนะ

การพัฒนาโครงข่ายประสาทเทียมเพื่อให้ได้ลักษณะการเชื่อมต่อระหว่างนิวรอนและค่าน้ำหนักเชื่อมต่อโดยการใช้วิธีอัลกอริธึมมดและทำงานร่วมกับการเรียนรู้แบบการแพร่ย้อนกลับ (BP) นั้นมีความจำเป็น เพราะยังไม่มีหลักเกณฑ์ใดที่จะกำหนดได้ว่าการเชื่อมต่อและค่าน้ำหนักการเชื่อมต่อระหว่างนิวรอนภายในโครงข่ายควรเป็นอย่างไร และยังสามารถลดการเกิดปัญหาการติดกับจุดต่ำสุดเฉพาะบริเวณและปัญหาการเข้ากันกับชุดข้อมูลฝึกสอนมากเกินไปได้ วิธีที่นำเสนอนี้เป็นการพัฒนาค้นหาและปรับเปลี่ยนเพียงรูปแบบการเชื่อมต่อและค่าน้ำหนักภายในของโครงข่ายเท่านั้น ซึ่งผู้เขียนจะได้ทำการศึกษาและวิจัยเพื่อทำให้สามารถพัฒนาหารูปแบบของจำนวนชั้นซ่อน นิวรอนในแต่ละชั้นซ่อนที่เหมาะสมเพิ่มเติมในโอกาสต่อไป

## บรรณานุกรม

- [1] V. Maniezzo, **“Genetic Evolution of the Topology and Weight Distribution of Neural Networks,”** IEEE Trans. on NN, pp. 39-53, 1994.
- [2] K.S. Tang, C.Y. Chan, K.F. Man and S.Kwong **“Genetic Structure for NN Topology and Weights Optimization”** Genetic Algorithms in Engineering Systems: Innovations and Applications, Conference Publication No. 414, September 1995.
- [3] W. Gao, **“Study on new evolutionary neural network,”** Proc. of ICLMC2003, IEEE, New York, pp. 1287-1293, 2003.
- [4] S.J. Han and S.B. Cho, **“Evolutionary Neural Networks for Anomaly Detection Based on the Behavior of a Program,”** IEEE Trans. On System, Man, And Cybernetics—Part B: Cybernetis, vol. 36, No. 3, pp 559-570, June 2006.
- [5] A. K. Jain, J. Mao, and K. M. Mohiuddin. **“Artificial neural networks: A tutorial,”** IEEE Computer, 29(3) :31-44, 1996.
- [6] P. L. Rosin and F. Fierens, **“Improving Neural Network Generalisation,”** In Proceedings of IGARSS’95, Firenze, Italy, July 1995.
- [7] S. Lawrence , C. L. Giles, and A. C. Tsoi, **“Lessons in Neural Network Training: Overfitting May be Harder than Expected,”** Proceedings of the Fourteenth National Conference on Artificial Intelligence, AAAI-97, AAAI Press, Menlo Park, California, pp. 540–545, 1997.
- [8] Y.P. Liu, M.G. Wu, and J.X. Qian, **“Evolving Neural Networks Using the Hybrid of Ant Colony Optimization and BP Algorithms,”** Advances in Neural Networks, vol. 3971, pp 714-722, 2006.
- [9] L. Prechelt, **“Proben1 – a set of neural network benchmark problems and benchmarking rules,”** Technical Report 21, Fakultät für Informatik, Universität Karlsruhe, Karlsruhe, German, 1994.
- [10] M. Srinivas and L. M. Patnaik, **“Learning neural network weights using genetic algorithms—Improving performance by search-space reduction,”** in Proc. 1991 IEEE Int. Joint Conf. Neural Networks (IJCNN’91 Singapore), vol. 3, pp. 2331–2336.
- [11] D. Montana and L. Davis, **“Training feedforward neural networks using genetic algorithms,”** in Proc. 11th Int. Joint Conf. Artificial Intelligence. San Mateo, CA: Morgan Kaufmann, pp. 762–767, 1989.

- [12] D. Whitley, T. Starkweather, and C. Bogart, "**Genetic algorithms and neural networks: Optimizing connections and connectivity,**" *Parallel Comput.*, vol. 14, no. 3, pp. 347–361, 1990.
- [13] Negnevitsky, Michael. **Artificial Intelligence: A Guide to Intelligent Systems.** Pearson, 2002.
- [14] E. Bonabeau, M. Dorigo, and G. Theraulaz, **Swarm Intelligence: From Natural to Artificial Systems** *Swarm Intelligence.* Oxford University Press, 1999.
- [15] M. Dorigo, M. Birattari, and T. Stützle, "**Ant Colony Optimization,**" *IEEE Computational Intelligence Magazine*, Nov. 2006.
- [16] M. Dorigo, V. Maniezzo, and A. Colomi, "**The ant system: optimization by a colony of cooperating agents,**" *IEEE Transactions on Systems, Man, and Cybernetics–Part B*, vol. 26, No. 2, pp. 29–41, 1996.
- [17] M. Dorigo and L.M. Gambardella, "**Ant Colony System: A cooperative learning approach to the traveling salesman problem,**" *IEEE Trans. Evolutionary Computation.*, vol. 1, no. 1, pp. 53–66, 1997.
- [18] E. Ridge, D. Kudenko, and D. Kazakov, "**A Study of Concurrency in the Ant Colony System Algorithm,**" *IEEE Congress on Evolutionary Computation Sheraton Vancouver Wall Centre Hotel, Vancouver, BC, Canada, July 16-21, 2006.*

## ประวัติผู้เขียน

นายนิกร โภคอุดม เกิดเมื่อวันที่ 14 มีนาคม พ.ศ. 2524 ที่จังหวัดมหาสารคาม สำเร็จการศึกษาชั้นประถมจากโรงเรียนบ้านหนองสาหร่าย อ.ปากช่อง จ.นครราชสีมา เมื่อปี พ.ศ. 2535 การศึกษาชั้นมัธยมศึกษาตอนต้นและตอนปลาย จากโรงเรียนปากช่อง อ.ปากช่อง จ.นครราชสีมา เมื่อปี พ.ศ. 2538 และ 2541 ตามลำดับ และการศึกษาในระดับปริญญาตรีวิศวกรรมศาสตรบัณฑิต (วิศวกรรมคอมพิวเตอร์) จากคณะวิศวกรรมศาสตร์ สาขาวิชาวิศวกรรมคอมพิวเตอร์ มหาวิทยาลัยอีสเทิร์นเอเซีย ในปีการศึกษา 2545 และเข้าศึกษาต่อในระดับปริญญาโท หลักสูตรวิศวกรรมศาสตรมหาบัณฑิต สาขาวิศวกรรมคอมพิวเตอร์ ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ในปีการศึกษา 2548

โดยได้เข้าทำงานในตำแหน่งอาจารย์ประจำ คณะวิศวกรรมศาสตร์ สาขาวิชาวิศวกรรมคอมพิวเตอร์ มหาวิทยาลัยอีสเทิร์นเอเซีย ในปี พ.ศ. 2546 และได้รับทุนศึกษาต่อระดับปริญญาโท สาขาวิศวกรรมคอมพิวเตอร์ ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง จากมหาวิทยาลัยอีสเทิร์นเอเซีย