

การทำนายโหลดระยะสั้นโดยใช้โครงข่ายประสาทเทียม  
(Short-Term Load Forecasting Using Neural Networks)



นายวีระศักดิ์ อิงคปกรณ์  
นายวุฒิชัย วิชาอนุมาศ

เลขที่.....  
เลขทะเบียน... 45970  
วัน, เดือน, ปี 2.6 ก.พ. 2546

b.....
i.....

ปฏิญญาพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรบัณฑิต  
สาขาวิศวกรรมไฟฟ้า  
สถาบันเทคโนโลยีพระจอมเกล้า เจ้าคุณทหารลาดกระบัง  
ปีการศึกษา 2544


ปริญญาานิพนธ์ปีการศึกษา 2544

ภาควิชาวิศวกรรมไฟฟ้า

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง การทำนายโหลดกระยะสั้นโดยใช้โครงข่ายประสาทเทียม

ผู้จัดทำ

- 
1. นายวีระศักดิ์ อิงคปกรณ
  2. นายวุฒิชัย วิทยาอนุมาส

Somchat Jirivithakorn อาจารย์ที่ปรึกษา

(ดร. สมชาติ จีรวิภากร)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## การทำนายโหลดระยะสั้นโดยใช้โครงข่ายประสาทเทียม

วีระศักดิ์ อิงคปกรณ์

วุฒิชัย วิทยาอนุมาส

ดร.สมชาติ จิรวิภากร อาจารย์ที่ปรึกษา

ปีการศึกษา 2544

### บทคัดย่อ

ปริญญานิพนธ์ฉบับนี้ เสนอถึงวิธีการประยุกต์ใช้โครงข่ายประสาทเทียมในการทำนายความต้องการกำลังไฟฟ้าของนิคมอุตสาหกรรมอมตะนครในจังหวัดชลบุรี ภายในระยะเวลา 24 ชั่วโมงล่วงหน้า โดยการทำนายความต้องการกำลังไฟฟ้าช่วงระยะเวลาสั้นกำลังเป็นที่ได้รับความสนใจจากผู้ผลิตกระแสไฟฟ้า เนื่องจากจะช่วยให้การผลิตกำลังไฟฟ้าได้ใกล้เคียงกับความต้องการที่แท้จริงเพื่อช่วยในการประหยัดค่าใช้จ่ายที่จะผลิตกระแสไฟฟ้า โดยโครงงานนี้ได้เชื่อมต่อโปรแกรมทั้งหมดเข้ากับฐานข้อมูลไมโครซอฟต์แอคเซส เพื่อสะดวกในการรับส่งข้อมูลที่จะเรียกในการใช้งาน ซึ่งพบว่าผลจากการทำนายมีค่าความผิดพลาดเฉลี่ยประมาณ 2%

## Short-Term Load Forecasting Using Neural Networks

Verasak Ingkapakorn

Wuttichai Wittayaanumas

Dr. Somchat Jiriwibhakorn Advisor

2001

### Abstract

This thesis presents the approach to apply the artificial neural networks to forecast the electric demand of Amata-Nakorn industry community in Chonburi within 24 hours ahead. Short term load forecasting is more interesting from many utilities. Because it can help utilities to generate electric power close to real demand and bring to saving operation costs. This project is linked all programs using to database is the convenient for users to transfer data from database. The results of neural networks have mean absolute percentage errors about 2 percent compared with the real outputs.

## สารบัญ

	หน้า
บทคัดย่อ	I
Abstract	II
สารบัญ	III
สารบัญรูป	VI
สารบัญตาราง	XI
บทที่ 1. บทนำ	1
1.1 วัตถุประสงค์ของโครงการปริญญาโท	3
1.2 ประโยชน์ที่ได้รับจากโครงการปริญญาโท	3
1.3 ขอบเขตของโครงการปริญญาโท	3
บทที่ 2. โครงข่ายประสาทเทียม	4
2.1 แนวความคิดเริ่มต้นของโครงข่ายประสาทเทียม	4
2.2 การเรียนรู้ของโครงข่ายประสาทเทียม	4
2.2.1 การเรียนรู้โดยมีผู้สอน	5
2.2.2 การเรียนรู้ในลักษณะที่ไม่มีผู้สอน	5
2.3 คุณลักษณะของโครงข่ายประสาทเทียม	5
2.4 รูปแบบและลักษณะของโครงข่ายประสาทเทียม	6
2.5 โครงข่ายประสาทเทียมแบบฟีดฟอร์เวิร์ด	6
2.6 แบบจำลองของโครงข่ายประสาทเทียม	7
2.7 สถาปัตยกรรมของโครงข่าย	8
2.8 รูปแบบของฟังก์ชันกระตุ้น	9
2.9 การเรียนรู้แบบแพร่ย้อนกลับ	11
2.10 การนอร์มัลไลเซชันของชุดข้อมูลที่ให้เรียนรู้	15
บทที่ 3. การทำนายโหลด	20
3.1 ประเภทของการทำนายโหลด	20
3.2 ปัจจัยที่มีอิทธิพลต่อการทำนายโหลด	21
3.3 ประโยชน์ของการทำนายโหลด	22

3.4 ตัวอย่างการศึกษาการพยากรณ์และการวางแผนระบบ ไฟฟ้ากำลัง ที่ทำผ่านมาของการไฟฟ้าส่วนภูมิภาค	22
บทที่ 4. การทำนายการใช้กำลัง ไฟฟ้าโดยใช้โครงข่ายประสาทเทียม	27
4.1 การจัดเตรียมโครงข่ายประสาทเทียม	27
4.2 การจัดเตรียมข้อมูลที่ใช้ในการเรียนรู้ของโครงข่ายประสาทเทียม	28
4.3 การเรียนรู้ของ โครงข่ายประสาทเทียม	28
บทที่ 5. ผลการวิเคราะห์โครงข่ายประสาทเทียม	30
5.1 ผลการเรียนรู้โครงข่ายประสาทเทียม	30
5.2 ผลการวิเคราะห์โครงข่ายประสาทเทียม	58
บทที่ 6. บทวิจารณ์และสรุป	60
บทที่ 7. การสร้างและติดต่อบริการข้อมูล	61
7.1 ความรู้เบื้องต้นเกี่ยวกับ เพิ่มข้อมูล และ ฐานข้อมูล	61
7.2 การสร้างฐานข้อมูลใน Microsoft Access	61
7.3 การติดต่อบริการข้อมูล	70
ภาคผนวก Source Code	75
กิตติกรรมประกาศ	90
บรรณานุกรม	91

## สารบัญรูป

	หน้า
รูปที่ 2-1 แสดงโครงข่ายประสาทเทียมที่โหนดมีอินพุตเดียว	7
รูปที่ 2-2 แสดงโครงข่ายประสาทเทียมที่โหนดมี R อินพุต	7
รูปที่ 2-3 แสดงโครงข่ายประสาทเทียมชั้นเดียว ประกอบด้วย R อินพุต	8
รูปที่ 2-4 แสดงโครงข่ายประสาทเทียมหลายชั้น ประกอบด้วย R อินพุต	9
รูปที่ 2-5 แสดงทรานเฟอร์ฟังก์ชันแบบลิเนียร์	9
รูปที่ 2-6 แสดงทรานเฟอร์ฟังก์ชันแบบ Log-sigmoid	10
รูปที่ 2-7 แสดงทรานเฟอร์ฟังก์ชันแบบ Tan-sigmoid	10
รูปที่ 2-8 แสดงโครงข่ายประสาทเทียมแบบฟีดฟอร์เวิร์ด 3 ชั้น ประกอบด้วย R อินพุต	14
รูปที่ 2-9 แสดงกราฟการกระจายของข้อมูลจากการนอร์มัลไลเซชัน	16
รูปที่ 4-1 แสดง Flow chart การทำงานของโครงข่ายประสาทเทียม	29
รูปที่ 5-1 แสดงกราฟโหลทรายชั่วโมงของการทดสอบวันที่ 1 ระหว่างโหลดจริงเทียบกับ โหลดจากการทำนายโดยใช้โครงข่ายประสาทเทียม	53
รูปที่ 5-2 แสดงกราฟโหลทรายชั่วโมงของการทดสอบวันที่ 2 ระหว่างโหลดจริงเทียบกับ โหลดจากการทำนายโดยใช้โครงข่ายประสาทเทียม	53
รูปที่ 5-3 แสดงกราฟโหลทรายชั่วโมงของการทดสอบวันที่ 3 ระหว่างโหลดจริงเทียบกับ โหลดจากการทำนายโดยใช้โครงข่ายประสาทเทียม	54
รูปที่ 5-4 แสดงกราฟโหลทรายชั่วโมงของการทดสอบวันที่ 4 ระหว่างโหลดจริงเทียบกับ โหลดจากการทำนายโดยใช้โครงข่ายประสาทเทียม	54
รูปที่ 5-5 แสดงกราฟโหลทรายชั่วโมงของการทดสอบวันที่ 5 ระหว่างโหลดจริงเทียบกับ โหลดจากการทำนายโดยใช้โครงข่ายประสาทเทียม	55
รูปที่ 5-6 แสดงกราฟโหลทรายชั่วโมงของการทดสอบวันที่ 6 ระหว่างโหลดจริงเทียบกับ โหลดจากการทำนายโดยใช้โครงข่ายประสาทเทียม	55
รูปที่ 5-7 แสดงกราฟโหลทรายชั่วโมงของการทดสอบวันที่ 7 ระหว่างโหลดจริงเทียบกับ โหลดจากการทำนายโดยใช้โครงข่ายประสาทเทียม	56
รูปที่ 5-8 แสดงกราฟโหลทรายชั่วโมงของการทดสอบวันที่ 8 ระหว่างโหลดจริงเทียบกับ โหลดจากการทำนายโดยใช้โครงข่ายประสาทเทียม	56

รูปที่ 5-9 แสดงกราฟของโหลคจริงและโหลคที่ได้จากการทำนายโดยโครงข่ายประสาทเทียม ตั้งแต่วันที่ 1 มีนาคม – 31 พฤษภาคม 2544	57
รูปที่ 7-1 แสดงการเรียกโปรแกรม Microsoft Access	61
รูปที่ 7-2 แสดงการสร้างฐานข้อมูลใหม่	62
รูปที่ 7-3 แสดงการตั้งชื่อไฟล์ฐานข้อมูล	63
รูปที่ 7-4 แสดงหน้าจอหลักของ Microsoft Access	64
รูปที่ 7-5 แสดงการเลือกรูปแบบของตาราง	65
รูปที่ 7-6 แสดงการกำหนดรายละเอียดของฟิลด์ข้อมูล	66
รูปที่ 7-7 แสดงการเซ็ทค่าคีย์หลักของตารางข้อมูล Electron	66
รูปที่ 7-8 แสดงการใส่ชื่อตารางข้อมูล Electron	67
รูปที่ 7-9 แสดงรายละเอียดของข้อมูลตาราง Result สำหรับเก็บผลการทดสอบการทำนายโหลค	68
รูปที่ 7-10 แสดงไฟล์ฐานข้อมูลชื่อ Project1.mdb ซึ่งประกอบด้วยตารางข้อมูล 2 ชุด	68
รูปที่ 7-11 แสดงการเก็บข้อมูลโหลคทั้งหมดในตาราง Electron เพื่อใช้ในการทำนายโหลค	69
รูปที่ 7-12 แสดงการเก็บข้อมูลจากผลการทำนายโหลคโดยใช้โครงข่ายประสาทเทียม	69
รูปที่ 7-13 แสดงการเรียกใช้งานฐานข้อมูลที่สร้างจาก Microsoft Access Driver(*.MDB) ได้ โดย การเรียกใช้งานผ่านทาง ODBC Driver	70
รูปที่ 7-14 แสดงการเรียก Control panel จาก taskbar	71
รูปที่ 7-15 แสดงวิธีการเข้าสู่การติดตั้ง ODBC	71
รูปที่ 7-16 แสดงการติดตั้ง System Data Source ของฐานข้อมูล	72
รูปที่ 7-17 แสดงการเรียกไดร์ฟเวอร์ของฐานข้อมูล	72
รูปที่ 7-18 แสดงการติดตั้ง ODBC สำหรับ Microsoft Access	73
รูปที่ 7-19 แสดงการตั้งชื่อ DSN สำหรับติดต่อฐานข้อมูล Microsoft Access	73
รูปที่ 7-20 แสดงวิธีการเลือกติดต่อฐานข้อมูล Microsoft Access ชื่อ Project1.mdb	74
รูปที่ 7-21 แสดง DSN แลไดร์ฟเวอร์ที่ติดตั้ง ODBC สำหรับ Microsoft Access	74

## สารบัญตาราง

	หน้า
ตารางที่ 5-1 แสดงผลการทำนายโหลดของโครงข่ายประสาทเทียม 2 ชั้น โดยใช้โหนดในชั้นซ่อน 4 โหนด และชั้นเอาต์พุต 24 โหนด	31
ตารางที่ 5-2 แสดงผลการทำนายโหลดของโครงข่ายประสาทเทียม 2 ชั้น โดยใช้โหนดในชั้นซ่อน 8 โหนด และชั้นเอาต์พุต 24 โหนด	32
ตารางที่ 5-3 แสดงผลการทำนายโหลดของโครงข่ายประสาทเทียม 2 ชั้น โดยใช้โหนดในชั้นซ่อน 16 โหนด และชั้นเอาต์พุต 24 โหนด	33
ตารางที่ 5-4 แสดงผลการทำนายโหลดของโครงข่ายประสาทเทียม 2 ชั้น โดยใช้โหนดในชั้นซ่อน 30 โหนด และชั้นเอาต์พุต 24 โหนด	34
ตารางที่ 5-5 แสดงผลการทำนายโหลดของโครงข่ายประสาทเทียม 2 ชั้น โดยใช้โหนดในชั้นซ่อน 35 โหนด และชั้นเอาต์พุต 24 โหนด	35
ตารางที่ 5-6 แสดงผลการทำนายโหลดของโครงข่ายประสาทเทียม 3 ชั้น โดยใช้โหนด ในชั้นซ่อนแรก 4 โหนด, ชั้นซ่อนที่สอง 4 โหนด และชั้นเอาต์พุต 24 โหนด	36
ตารางที่ 5-7(ต่อ) แสดงผลการทำนายโหลดของโครงข่ายประสาทเทียม 3 ชั้น โดยใช้โหนด ในชั้นซ่อนแรก 4 โหนด, ชั้นซ่อนที่สอง 4 โหนด และชั้นเอาต์พุต 24 โหนด	37
ตารางที่ 5-8 แสดงผลการทำนายโหลดของโครงข่ายประสาทเทียม 3 ชั้น โดยใช้โหนด ในชั้นซ่อนแรก 8 โหนด, ชั้นซ่อนที่สอง 3 โหนด และชั้นเอาต์พุต 24 โหนด	38
ตารางที่ 5-9(ต่อ) แสดงผลการทำนายโหลดของโครงข่ายประสาทเทียม 3 ชั้น โดยใช้โหนด ในชั้นซ่อนแรก 8 โหนด, ชั้นซ่อนที่สอง 3 โหนด และชั้นเอาต์พุต 24 โหนด	39
ตารางที่ 5-10 แสดงผลการทำนายโหลดของโครงข่ายประสาทเทียม 3 ชั้น โดยใช้โหนด ในชั้นซ่อนแรก 15 โหนด, ชั้นซ่อนที่สอง 8 โหนด และชั้นเอาต์พุต 24 โหนด	40
ตารางที่ 5-11(ต่อ) แสดงผลการทำนายโหลดของโครงข่ายประสาทเทียม 3 ชั้น โดยใช้โหนด ในชั้นซ่อนแรก 15 โหนด, ชั้นซ่อนที่สอง 8 โหนด และชั้นเอาต์พุต 24 โหนด	41
ตารางที่ 5-12 แสดงผลการทำนายโหลดของโครงข่ายประสาทเทียม 3 ชั้น โดยใช้โหนด ในชั้นซ่อนแรก 32 โหนด, ชั้นซ่อนที่สอง 28 โหนด และชั้นเอาต์พุต 24 โหนด	42
ตารางที่ 5-13(ต่อ) แสดงผลการทำนายโหลดของโครงข่ายประสาทเทียม 3 ชั้น โดยใช้โหนด ในชั้นซ่อนแรก 32 โหนด, ชั้นซ่อนที่สอง 28 โหนด และชั้นเอาต์พุต 24 โหนด	43

ตารางที่ 5-14 แสดงผลการทำนายโหลดในวันที่ 1 ของชุดข้อมูลทดสอบ โดยมีค่า ความผิดพลาดเฉลี่ย 4.7581 % และความผิดพลาดสูงสุด 12.1625 %	45
ตารางที่ 5-15 แสดงผลการทำนายโหลดในวันที่ 2 ของชุดข้อมูลทดสอบ โดยมีค่า ความผิดพลาดเฉลี่ย 0.9237 % และความผิดพลาดสูงสุด 3.9197 %	46
ตารางที่ 5-16 แสดงผลการทำนายโหลดในวันที่ 3 ของชุดข้อมูลทดสอบ โดยมีค่า ความผิดพลาดเฉลี่ย 1.4716 % และความผิดพลาดสูงสุด 3.8036 %	47
ตารางที่ 5-17 แสดงผลการทำนายโหลดในวันที่ 4 ของชุดข้อมูลทดสอบ โดยมีค่า ความผิดพลาดเฉลี่ย 3.0789 % และความผิดพลาดสูงสุด 10.5975 %	48
ตารางที่ 5-18 แสดงผลการทำนายโหลดในวันที่ 5 ของชุดข้อมูลทดสอบ โดยมีค่า ความผิดพลาดเฉลี่ย 2.7220 % และความผิดพลาดสูงสุด 8.4155 %	49
ตารางที่ 5-19 แสดงผลการทำนายโหลดในวันที่ 6 ของชุดข้อมูลทดสอบ โดยมีค่า ความผิดพลาดเฉลี่ย 0.6875 % และความผิดพลาดสูงสุด 1.6363 %	50
ตารางที่ 5-20 แสดงผลการทำนายโหลดในวันที่ 7 ของชุดข้อมูลทดสอบ โดยมีค่า ความผิดพลาดเฉลี่ย 0.7898 % และความผิดพลาดสูงสุด 1.8916 %	51
ตารางที่ 5-21 แสดงผลการทำนายโหลด ในวันที่ 8 ของชุดข้อมูลทดสอบ โดยมีค่า ความผิดพลาดเฉลี่ย 5.8305 % และความผิดพลาดสูงสุด 29.3482 %	52

## บทที่ 1

### บทนำ

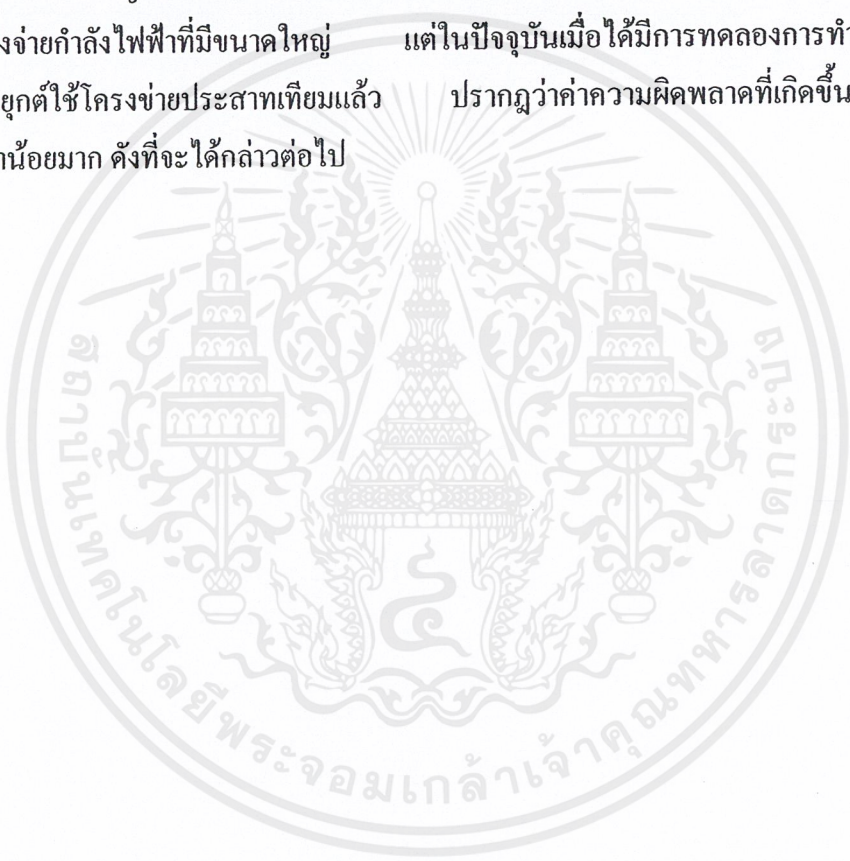
ในปัจจุบันได้มีความเจริญก้าวหน้าทางเทคโนโลยีในด้านต่าง ๆ เป็นจำนวนมาก ซึ่งทำให้เกิดระบบที่มีขีดความสามารถในการตัดสินใจหาตัวเองในขอบเขตระดับหนึ่ง และรวมถึงการพัฒนาของเทคโนโลยีทางด้านคอมพิวเตอร์ ทำให้ได้มีการนำเทคโนโลยีทั้งสองเข้ามาผสมผสานรวมกัน เพื่อประยุกต์ใช้ในการแก้ปัญหาหรือการวิเคราะห์ในระบบไฟฟ้ากำลัง

จากการประสบความสำเร็จในการลอกเลียนแบบการทำงานของเซลล์ประสาทของสิ่งมีชีวิตได้ก่อให้เกิดระบบประสาทเทียมขึ้น ซึ่งมีความสามารถในการตัดสินใจคล้ายคลึงกับมนุษย์เป็นอย่างมากและได้มีการนำระบบโครงข่ายประสาทเทียมนี้มาประยุกต์ใช้ในระบบไฟฟ้ากำลัง อาทิเช่น การวิเคราะห์การวางแผนในระบบไฟฟ้ากำลัง, การประมวลสัญญาณบอกเหตุ, การทำนายโหลด ( Load Forecasting) เป็นต้น โดยในปริญญาณิพนธ์ฉบับนี้จะกล่าวถึงการนำโครงข่ายประสาทเทียมมาประยุกต์ใช้ในการทำนายโหลด หรือใช้ในการทำนายการใช้กำลังไฟฟ้าโดยอาศัยข้อมูลของโหลดชุดเก่ากับข้อมูลทางสถานะอากาศและช่วงวันหยุดมาเป็นหลักในการทำนาย โดยในที่นี้จะไม่ได้ทำการวิเคราะห์ถึงผลของการเปลี่ยนแปลงของโหลดตามสถานะทางเศรษฐกิจ

ในระบบไฟฟ้ากำลังนั้น โรงจักรไฟฟ้าจะทำหน้าที่ในการผลิตกระแสไฟฟ้าและทำการส่งจ่ายกำลังไฟฟ้าไปยังโหลดโดยผ่านทางระบบสายส่ง (Transmission Line) ซึ่งในการผลิตกระแสไฟฟ้าโรงจักรไฟฟ้าแต่ละโรงจักรจำเป็นต้องทำการผลิตกำลังไฟฟ้าให้เพียงพอต่อความต้องการของโหลด ทั้งนี้ก็เนื่องมาจากหากโรงจักรไฟฟ้าไม่สามารถผลิตกำลังไฟฟ้าได้เพียงพอต่อความต้องการของโหลดแล้วย่อมทำให้เกิดผลเสียหาย อาทิเช่น เจนเนอเรเตอร์ (Generator) เกิดความเสียหาย, ทำให้อุปกรณ์เครื่องใช้ได้รับความเสียหาย, เกิดความเสียหายทางเศรษฐกิจและอุตสาหกรรม เป็นต้น ในทางกลับกันหากโรงจักรไฟฟ้าทำการผลิตกระแสไฟฟ้ามากเกินไปเกินความต้องการของโหลดแล้ว ย่อมก่อให้เกิดปัญหาในด้านการใช้ทรัพยากรธรรมชาติอย่างไม่มีประสิทธิภาพอันจะก่อให้เกิดปัญหาการขาดแคลนทรัพยากรธรรมชาติตามมาในภายหลัง อีกทั้งยังก่อเกิดต้นทุนการผลิตที่สูงขึ้นอันเนื่องมาจากการผลิตกระแสไฟฟ้าเกินความต้องการ เป็นต้น

ในอดีตนั้นการทำนายโหลคมักมีด้วยกันหลายวิธี ได้แก่ การทำนายโหลคโดยอาศัยการสำรวจความต้องการกำลังไฟฟ้าในแต่ละพื้นที่, การทำนายโหลคโดยอาศัยข้อมูลทางสถิติ, การทำนายโหลคโดยอาศัยสมการทางคณิตศาสตร์ และวิธีสุดท้ายคือ การทำนายโหลคโดยอาศัยสมการทางคณิตศาสตร์ ร่วมกับการพิจารณาถึงสภาวะทางเศรษฐกิจ

ในการทำนายโหลคด้วยวิธีการดังที่ได้กล่าวมาแล้วในข้างต้น โดยส่วนใหญ่แล้วจะให้ผลการทำนายที่ให้ค่าความผิดพลาดค่อนข้างสูง ซึ่งค่าความผิดพลาดที่เกิดขึ้นดังกล่าวนี้หากว่าเกิดขึ้นกับระบบไฟฟ้าที่มีขนาดใหญ่แล้วก็จะยิ่งทวีความผิดพลาดมากยิ่งขึ้นไปตามลำดับ จึงไม่เหมาะสมที่จะนำมาใช้ในระบบส่งจ่ายกำลังไฟฟ้าที่มีขนาดใหญ่ แต่ในปัจจุบันเมื่อได้มีการทดลองการทำนายโหลคโดยอาศัยการประยุกต์ใช้โครงข่ายประสาทเทียมแล้ว ปรากฏว่าค่าความผิดพลาดที่เกิดขึ้นจากการทำนายโหลคนั้นมีค่าน้อยมาก ดังที่จะได้กล่าวต่อไป



## 1.1 วัตถุประสงค์ของโครงการปริญญานิพนธ์

1.1.1 เพื่อทำการศึกษาคูณลักษณะของโครงข่ายประสาทเทียม

- 1) ลักษณะการทำงานของโครงข่ายประสาทเทียม
- 2) ลักษณะการเรียนรู้ของโครงข่ายประสาทเทียม

1.1.2 เพื่อศึกษาความสัมพันธ์ของค่าพารามิเตอร์ที่มีผลต่อการเรียนรู้ของโครงข่ายประสาทเทียม ได้แก่

- 1) ค่าน้ำหนักเริ่มต้น
- 2) ค่าไบอัสเริ่มต้น
- 3) ชนิดของ Transfer Function
- 4) ขนาดโครงสร้างของโครงข่ายประสาทเทียม

1.1.3 เพื่อศึกษาระบบฐานข้อมูลที่ใช้ในการเชื่อมโยงกับโปรแกรมสร้างโครงข่ายประสาทเทียม

## 1.2 ประโยชน์ที่ได้รับจากโครงการปริญญานิพนธ์

1.2.1 ได้เรียนรู้ถึงหลักการทำงานของโครงข่ายประสาทเทียมและสามารถเขียนโปรแกรมโครงข่ายประสาทเทียมได้

1.2.2 ได้เรียนรู้ถึงวิธีการต่าง ๆ ที่ใช้ในการทำนายการใช้กำลังไฟฟ้า

1.2.3 ได้เรียนรู้เรื่องระบบฐานข้อมูลเพื่อที่จะนำไปใช้ในการออกแบบฐานข้อมูลได้

1.2.4 สามารถนำงานปริญญานิพนธ์ไปประยุกต์ใช้งานในระบบไฟฟ้ากำลังได้

## 1.3 ขอบเขตของโครงการปริญญานิพนธ์

1.3.1 เขียนโปรแกรมโครงข่ายประสาทเทียมแบบฟีดฟอร์เวิร์ด (Feed Forward Neural Network) โดยใช้ในการเรียนรู้แบบแพร่ย้อนกลับ (Back-Propagation) ด้วยโปรแกรม MATLAB เพื่อใช้ในการทำนายการใช้ความต้องการพลังงานไฟฟ้า

1.3.2 ทำการวิเคราะห์ผลของการทำนายโหลดโดยอาศัยการเขียนโปรแกรม MATLAB

1.3.3 ทำระบบฐานข้อมูล (Database System) โดยใช้ Microsoft Access เพื่อใช้ในการติดต่อรับส่งข้อมูลไปยังโปรแกรม MATLAB เพื่อใช้ในการทำนายโหลดด้วยภาษาเอสคิวแอล (SQL-Language)

## บทที่ 2

### โครงข่ายประสาทเทียม

#### โครงข่ายประสาทเทียม (Artificial Neural Networks)

##### 2.1 แนวความคิดเริ่มต้นของโครงข่ายประสาทเทียม

ในการลอกเลียนแบบลักษณะการทำงานของเซลล์ประสาทภายในสมองมนุษย์ เราจะอธิบายด้วยหลักการพื้นฐานและพฤติกรรมของเซลล์ประสาทในทางคณิตศาสตร์ ซึ่งโดยรวมแล้วมีหลักการมาจากโครงข่ายของเซลล์ประสาทในทางชีววิทยาอย่างใกล้เคียงที่สุด

สถาปัตยกรรมของโครงข่ายประสาทเทียม เราสามารถนิยามได้ถึงลักษณะทางโครงสร้างของโครงข่ายประสาทเทียม ซึ่งหมายถึง จำนวนของเซลล์ประสาทภายในโครงข่ายและลักษณะของการเชื่อมต่อกันระหว่างเซลล์ประสาท

โครงข่ายประสาทจะมีรูปแบบการประมวลผลข้อมูลอยู่บนพื้นฐานของการแยกส่วนแบบขนาน (Parallel Decomposition) แทนที่การประมวลผลแบบอัลกอริทึม (Algorithm) ตามลำดับขั้นตอน โดยโครงข่ายประสาทจะประมวลผลโดยอาศัยการแยกส่วนข้อมูลที่มีความซับซ้อนออก และส่งไปยังหน่วยพื้นฐานหรือหน่วยประมวลผลย่อย ตัวอย่างเช่น สีที่ถูกผสมกันแล้วจะไม่สามารถที่จะจำแนกออกเป็นสีพื้นฐานได้ แต่ในทางทฤษฎีของโครงข่ายประสาทเทียมแล้ว สีอาจจะถูกจำลองขึ้นมาใหม่โดยยังคงไว้ซึ่งลักษณะของสีเดิม และที่หน่วยประมวลผลย่อยก็จะทำการจดจำความสัมพันธ์ของรูปแบบของข้อมูลที่ป้อนเข้าสู่โครงข่ายต่อไป เช่น เมื่อเรามองวัตถุ สมองจะไม่เก็บข้อมูลของวัตถุที่เรามองเห็นในรูปของหน่วยความจำแบบเมตริกซ์ของแต่ละพิกเซล (Pixel) แต่สมองจะทำการเก็บบันทึกคุณลักษณะพื้นฐานของวัตถุที่มองเห็น เช่น ขนาด, รูปร่าง, สี เป็นต้น [5]

##### 2.2 การเรียนรู้ของโครงข่ายประสาทเทียม

การเรียนรู้ของโครงข่ายประสาทเทียมจะไม่มีข้อสรุปการประมวลผลที่ตายตัว นั่นคือ การประมวลผลที่แตกต่างกัน จะมีความเหมาะสมในประเภทของแบบจำลองโครงข่ายที่แตกต่างกัน ดังนั้น การประมวลผลการเรียนรู้จึงมีประสิทธิภาพที่แตกต่างกัน

ขบวนการเรียนรู้ของโครงข่ายประสาทเทียม คือ ช่วงที่โครงข่ายจะปรับตัวเอง (การปรับค่าน้ำหนักและไบอัส) จากการกระตุ้นอินพุตเพื่อสร้างผลตอบสนองเอาที่พู่ทของโครงข่ายให้ได้ตามที่ต้องการ

และในที่สุดโครงข่ายประสาทเทียม ก็จะสามารถสร้างผลตอบสนองเอาที่พู่ที่ต้องการได้ในทุก ๆ รูปแบบของอินพุตนั้น โครงข่ายก็จะสิ้นสุดการเรียนรู้อย่างสมบูรณ์ ลักษณะการเรียนรู้ของโครงข่ายประสาทเทียมที่สำคัญ สามารถแบ่งได้ 2 ประเภท คือ การเรียนรู้แบบมีผู้สอน(Supervised Learning) และการเรียนรู้โดยไม่มีผู้สอน(Unsupervised Learning) [1]

### 2.2.1 การเรียนรู้ในลักษณะมีผู้สอน (Supervised Learning)

ในระหว่างช่วงของการเรียนรู้ของโครงข่ายประสาทเทียม โครงข่ายจะสร้างผลตอบสนองขึ้นที่เอาที่พู่ โดยผลตอบสนองเอาที่พู่ของโครงข่ายจะมีความแตกต่างจากผลตอบสนองเอาที่พู่เป้าหมายที่ต้องการ ซึ่งจะมีค่าผิดพลาดเกิดขึ้น โดยค่าผิดพลาดที่เกิดขึ้นจะถูกนำมาใช้ในการคำนวณปรับปรุงค่าน้ำหนักและไบอัสทุกอย่างของโครงข่ายในแต่ละรอบของการเรียนรู้จนกว่าโครงข่ายจะสามารถสร้างผลตอบสนองได้เหมือนกับเอาที่พู่เป้าหมาย นั่นก็คือ กระบวนการเรียนจะรู้สิ้นสุดได้ก็ต่อเมื่อ ค่าผิดพลาดที่เกิดขึ้นมีค่าเท่ากับศูนย์หรือมีค่าอยู่ในเกณฑ์ที่ยอมรับได้

ในกระบวนการปรับค่าผิดพลาดให้น้อยที่สุดนั้น เริ่มแรกต้องมึวจรในการสร้างค่าผิดพลาดขึ้น เรียกว่า ผู้สอน (Teacher) หรือผู้ควบคุม (Supervised) ซึ่งก็คือ ค่าเอาที่พู่เป้าหมายที่ต้องการในแต่ละรูปแบบของอินพุต ดังนั้นเราจึงเรียการเรียนรู้ของโครงข่ายในลักษณะนี้ว่า การเรียนรู้ลักษณะมีผู้สอน (Supervised Learning)

### 2.2.2 การเรียนรู้ในลักษณะไม่มีผู้สอน (Unsupervised Learning)

การเรียนรู้ลักษณะนี้จะแตกต่างกับการเรียนรู้แบบมีผู้สอนตรงที่ ในแต่ละรูปแบบของอินพุต จะไม่มีค่าเอาที่พู่เป้าหมายในช่วงการเรียนรู้ของโครงข่ายประสาทเทียม ในลักษณะการเรียนรู้ชนิดนี้ โครงข่ายจะไม่ต้องการผู้สอนแต่ต้องการเพียงสิ่งชี้แนะ(Guide Line) ในการพิจารณาเริ่มแรกว่า โครงข่ายจะสร้างกลุ่มข้อมูลได้อย่างไรตามคุณลักษณะต่างๆของอินพุต ดังนั้นถ้าไม่มีตัวชี้แนะโครงข่ายจะไม่ทราบว่าจะใช้คุณลักษณะชนิดใดในการจัดกลุ่มข้อมูล

## 2.3 คุณลักษณะของโครงข่ายประสาทเทียม

โครงข่ายประสาทเทียม จะมีการเชื่อมต่อกันของหน่วยประมวลผลย่อย (Processing Element) ที่จำลองมาจากการทำงานของเซลล์ประสาทตัวหนึ่ง โครงข่ายมีความสามารถในการปรับตัวเองในช่วงของการเรียนรู้ได้ ในทางคณิตศาสตร์ โครงข่ายประสาทเทียมสามารถจำลองได้ด้วยกลุ่มของสมการดิฟเฟอเรนเชียลของผลตอบสนองของหน่วยประมวลผลย่อยที่เชื่อมต่อเข้าด้วยกัน ซึ่งคล้ายกับทฤษฎีการควบคุมแบบป้อนกลับ

## 2.4 รูปแบบและลักษณะของโครงข่ายประสาทเทียม (Artificial Neural Network Topology)

โครงข่ายประสาทเทียมจะประกอบด้วยเซลล์ประสาทจำนวนมากที่เชื่อมโยงเข้าด้วยกันในเส้นทางที่แน่นอน โดยปกติการกระตุ้นอินพุตจะถูกป้อนเข้าสู่ชั้นอินพุตและผลตอบสนองเอาต์พุตแต่ละหน่วยประมวลผลย่อยจะถูกส่งไปเป็นอินพุตในชั้นถัดไปของโครงข่ายประสาทเทียม

ชั้นของโหนดที่อยู่ระหว่างอินพุตและเอาต์พุต เรียกว่า ชั้นซ่อน (Hidden Layer) ซึ่งอาจมีจำนวน 1 ชั้นหรือมากกว่าก็ได้ การมีจำนวนของชั้นซ่อนมากเท่าใดก็ทำให้โครงข่ายประสาทเทียมมีแนวโน้มในความสามารถการแก้ปัญหาที่มีความซับซ้อนที่ดียิ่งขึ้น

ในการทดลองนี้ เราจะใช้โครงข่ายประสาทเทียมแบบฟีดฟอร์เวิร์ด (Feed-Forward Neural Networks) ซึ่งประกอบด้วยชั้นซ่อน(Hidden Layer) 1 ชั้นและ 2 ชั้น โดยใช้วิธีเรียนรู้แบบแพร่ย้อนกลับ (Back-Propagation Learning Algorithm) ลักษณะของการเรียนรู้แบบนี้จะเป็นการเรียนรู้แบบมีผู้สอน (Supervised Learning)

## 2.5 โครงข่ายประสาทเทียมแบบฟีดฟอร์เวิร์ด (Feed-Forward Neural Networks)

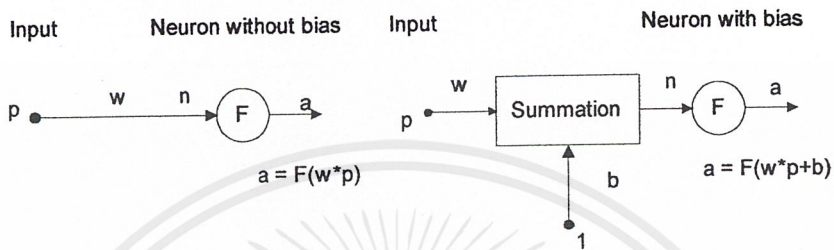
โครงข่ายประสาทเทียมแบบฟีดฟอร์เวิร์ดจะประกอบด้วยชั้นของโหนด หรือ ชั้นของเซลล์ประสาท ซึ่งมีการเชื่อมต่อของเซลล์ประสาทในระหว่างชั้นเซลล์ของโครงข่าย โดยแต่ละโครงข่ายจะประกอบไปด้วยชั้นต่างๆดังต่อไปนี้ คือ

- ชั้นซ่อน (Hidden Layer) คือ ชั้นของโหนดที่มีการเชื่อมต่อตั้งแต่อินพุตจนถึงชั้นสุดท้ายของโครงข่ายประสาทเทียม ซึ่งโครงข่ายหนึ่งๆจะมี 1 ชั้นซ่อน หรือ หลายชั้นซ่อนก็ได้
- ชั้นเอาต์พุต (Output Layer) คือ ชั้นของโหนดที่อยู่สุดท้ายซึ่งมีการเชื่อมต่อกับเอาต์พุตของโครงข่ายประสาทเทียม ซึ่งโครงข่ายหนึ่งๆจะมีชั้นเอาต์พุตเพียงชั้นเดียว

ในการทำงานของโครงข่ายประสาทเทียมนั้น โครงข่ายในแต่ละชั้นจะทำการประมวลผลข้อมูลหรืออินพุตที่ได้รับและทำการสร้างผลตอบสนอง หรือเอาต์พุตเพื่อส่งต่อไปเป็นข้อมูลอินพุตของเซลล์ประสาทในชั้นถัดไป โดยผ่านค่าน้ำหนักของการไซแนปส์และไบอัสของการเชื่อมต่อระหว่างเซลล์นั้น

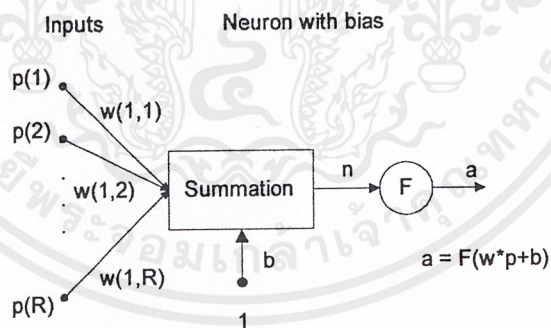
## 2.6 แบบจำลองของโครงข่ายประสาทเทียม (Neural Networks Model)

- โครงข่ายประสาทเทียมที่โหนดมีอินพุตเดียว (Single Input Neuron Networks)



รูปที่ 2-1 แสดงโครงข่ายประสาทเทียมที่โหนดมีอินพุตเดียว

- โครงข่ายประสาทเทียมที่โหนดมี R อินพุต (Single Neuron Networks with R inputs)



รูปที่ 2-2 แสดงโครงข่ายประสาทเทียมที่โหนดมี R อินพุต

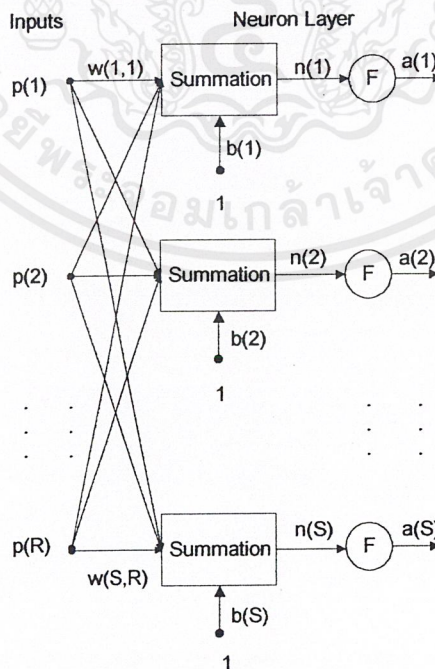
รูปที่ 2-1 เป็นลักษณะของโครงข่ายประสาทเทียมที่โหนดมีอินพุตเดียว โครงข่ายทางด้านซ้ายจะไม่มีค่าถ่วงหรือตัวคด (Bias) ส่วนโครงข่ายทางด้านขวาที่โหนดจะมีค่าถ่วงหรือตัวคด (Bias) ในแต่ละโหนด (neural) จะมีค่าทรานสเฟอร์ฟังก์ชัน (F) ที่จะคำนวณเปลี่ยนค่าจาก net input (n) ให้เป็นค่าเอาต์พุต (a) ซึ่งค่าน้ำหนัก (w) และค่า ไบอัส (b) จะสามารถทำการปรับค่าได้

รูปที่ 2-2 เป็นลักษณะของโครงข่ายประสาทเทียมที่โหนดมี R อินพุตซึ่งเรียกกลุ่มข้อมูลอินพุตว่า เวกเตอร์อินพุต ที่แต่ละตำแหน่งของอินพุต จะมีค่าน้ำหนักที่เป็นกลุ่มของเวกเตอร์ประกอบอยู่ด้วย จะเห็นว่าในกรณีที่โหนดมีหลายอินพุต (Multiple Input Neuron) ค่า net input (n) สามารถคำนวณได้จาก ผลรวมของผลคูณของค่าของอินพุต ( $p(1), p(2), \dots, p(R)$ ) กับค่าน้ำหนัก ( $w(1,1), w(1,2), \dots, w(1,R)$ ) ตามลำดับ บวกด้วยค่าไบอัส ดังแสดงได้ดังสมการที่ 2.1

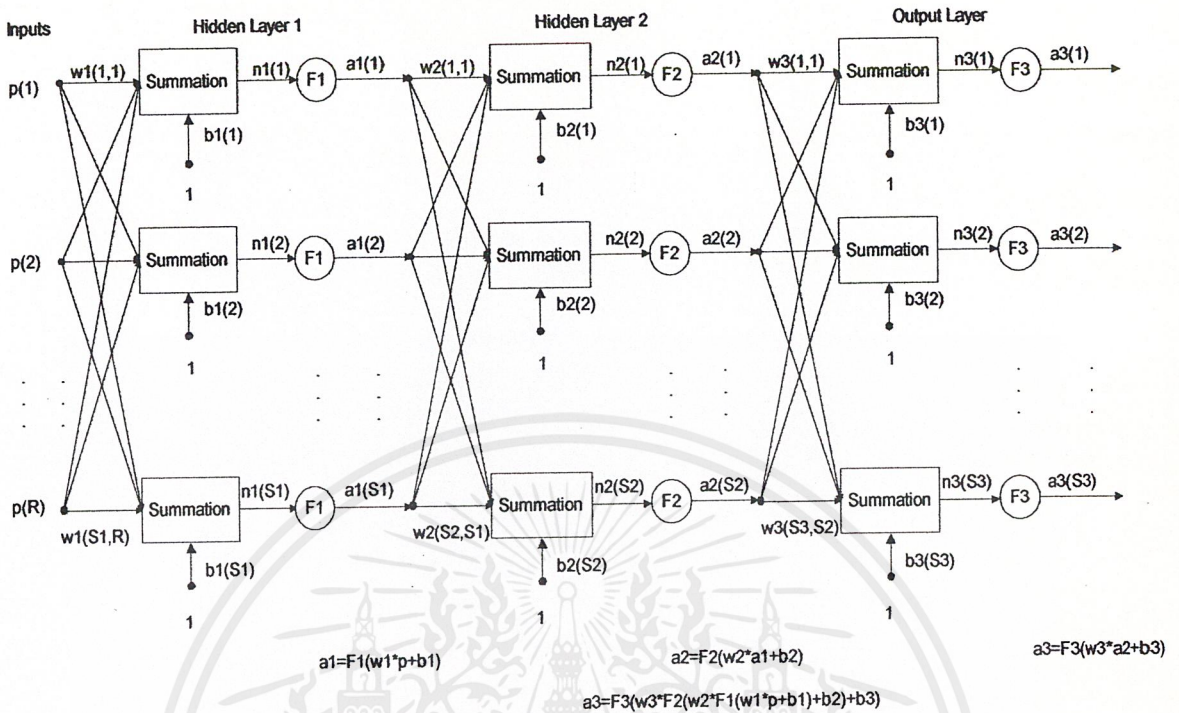
$$n = p(1).w(1,1) + p(2).w(1,2) + \dots + p(R).w(1,R) + b \quad (2.1)$$

## 2.7 สถาปัตยกรรมของโครงข่าย (Network Architectures)

จำนวนโหนดตั้งแต่ 2 โหนดขึ้นไปอาจประกอบอยู่ในแต่ละชั้นและโครงข่ายอาจจะมีจำนวนชั้นมากกว่า 1 ชั้นขึ้นไป ในตอนแรกเราจะพิจารณาโครงข่ายที่มีชั้นเดียว ซึ่งแสดงดังรูปที่ 2-3 ซึ่งมี R อินพุต และมีโหนด S โหนด อย่างไรก็ตาม ในแต่ละชั้นของโครงข่ายไม่จำเป็นต้องมีจำนวนอินพุต (R) เท่ากับจำนวนโหนด (S) ในชั้น นอกจากนี้โครงข่ายประสาทเทียมหนึ่งๆสามารถมีได้หลายชั้น ซึ่งแต่ละชั้นจะมีค่าน้ำหนัก (w), ค่าไบอัส (b) และค่าเอาต์พุต (a) แสดงได้ดังรูปที่ 2-4



เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ซึ่งห้ามเผยแพร่โดยไม่ได้รับอนุญาตจากเจ้าของลิขสิทธิ์ หากมีการนำเอกสารนี้ไปใช้โดยไม่ได้รับอนุญาต เจ้าของลิขสิทธิ์ขอสงวนสิทธิ์ในการดำเนินคดีตามกฎหมายที่เกี่ยวข้อง



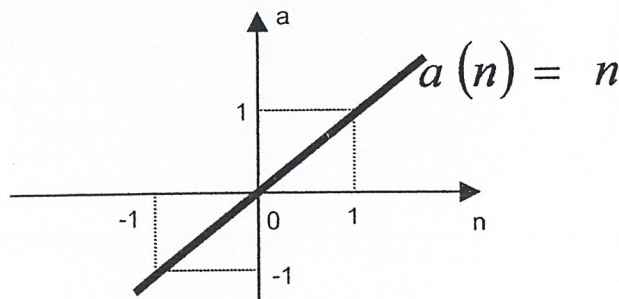
รูปที่ 2-4 แสดงโครงข่ายประสาทเทียมหลายชั้น ประกอบด้วย R อินพุต

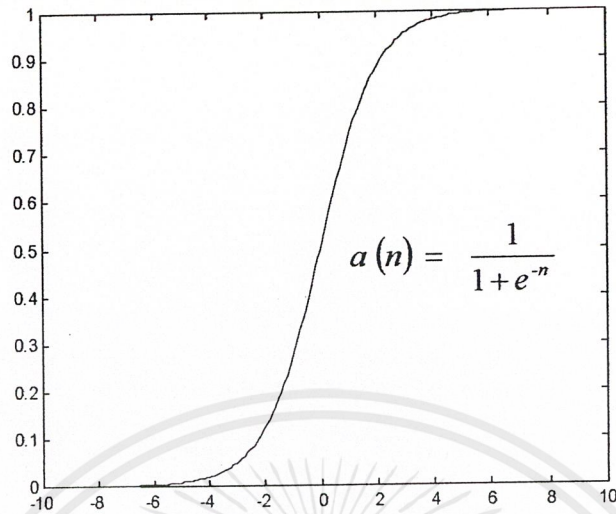
จากรูปที่ 2-4 ค่าผิดพลาด ( $e(i) ; i = 1, 2, \dots, S3$  (neurons)) ที่เกิดขึ้นที่ชั้นเอาต์พุตจะมีค่าเท่ากับความแตกต่างระหว่างค่าเอาต์พุตจริง ( $t(i) ; i = 1, 2, \dots, S3$ (neurons)) และ ค่าเอาต์พุตที่ได้จากนิรอล ( $a3(i) ; i = 1, 2, \dots, S3$  (neurons)) ซึ่งแสดงได้ดังสมการที่ 2.2

$$e(i) = t(i) - a3(i) ; i = 1, 2, \dots, S3 \tag{2.2}$$

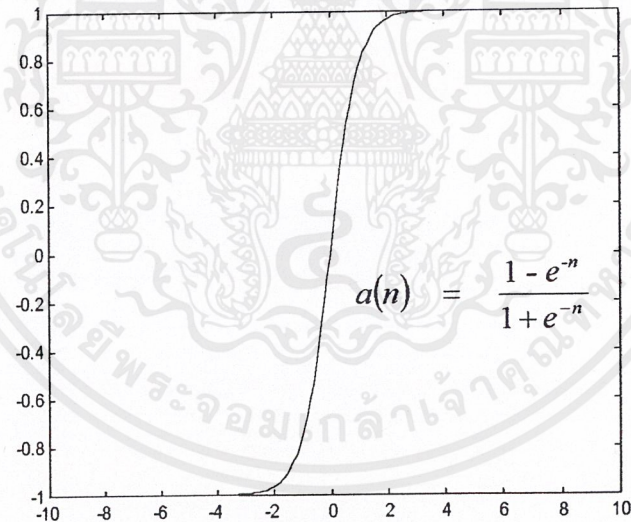
### 2.8 รูปแบบของฟังก์ชันกระตุ้น (Transfer Function)

ฟังก์ชันกระตุ้น (Transfer Function) ที่ใช้ในโครงข่ายประสาทเทียมมี 3 รูปแบบ แสดงดังรูป





รูปที่ 2-6 ทรานเฟอร์ฟังก์ชันแบบ Log-sigmoid



รูปที่ 2-7 ทรานเฟอร์ฟังก์ชันแบบ Tan-sigmoid

จากรูปที่ 2-5 ถึง 2-7 แสดงกราฟและสมการของทรานเฟอร์ฟังก์ชัน โดยที่  $a$  เป็นค่าเอาต์พุตของทรานเฟอร์ฟังก์ชัน และ  $n$  เป็นค่าอินพุตรวม โครงข่ายประสาทเทียมที่ใช้การเรียนรู้แบบแพร่ย้อนกลับ นิยมใช้ทรานเฟอร์ฟังก์ชันแบบ Log-sigmoid และ Tan-sigmoid ที่ใช้ทรานส์เฟอร์ฟังก์ชันแบบนี้ เพราะสามารถทำการดิฟเฟอเรนเชียลได้ โหนดที่ใช้ทรานเฟอร์ฟังก์ชันแบบนี้จะใช้กับการประมวลค่าแบบไม่เป็นเชิงเส้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.9 การเรียนรู้แบบแพร่ย้อนกลับ (Back-Propagation Learning Algorithm)

การเรียนรู้แบบแพร่ย้อนกลับ คือ กระบวนการทางคณิตศาสตร์ที่ใช้ในการปรับค่าน้ำหนักของโครงข่ายประสาทเทียมระหว่างเซลล์ประสาทภายในโครงข่าย โดยใช้ค่าผิดพลาดที่เกิดจากการตอบสนองของโครงข่ายต่อรูปแบบข้อมูลอินพุตและเอาต์พุตที่ต้องการมาเป็นค่าเริ่มต้น โดยมีเป้าหมายคือ การทำให้ค่าผิดพลาดที่ได้จากโครงข่ายประสาทเทียมนั้นมีค่าลดลง การเรียนรู้แบบแพร่ย้อนกลับสามารถปรับค่าน้ำหนัก, ค่าไบอัส และผลรวมของค่าผิดพลาดยกกำลังสอง (Sum-Squared Error) ได้ดังสมการที่ 2.3-2.5

$$W_{new} = W_{old} + \Delta W \quad (2.3)$$

$$b_{new} = b_{old} + \Delta b \quad (2.4)$$

$$SSE = \sum_{i=1}^N (t_i - a_i)^2 \quad (2.5)$$

โดยที่  $N$  คือ จำนวนชุดข้อมูลอินพุตที่ใช้ในการเรียนรู้

กระบวนการปรับค่าน้ำหนักและไบอัสที่เชื่อมต่ออยู่ระหว่างนิวรอน จะทำให้ค่าผิดพลาดระหว่างค่าเอาต์พุตจริงและเอาต์พุตที่ได้จากโครงข่ายประสาทเทียมมีค่าน้อยลง ลักษณะการเรียนรู้โดยวิธีของ gradient descent หรือกฎเดลต้า ค่าน้ำหนักและค่าไบอัสจะถูกปรับเปลี่ยนโดย การปรับค่าของอัตราการเรียนรู้ ( $\eta$ ) และค่าสัมประสิทธิ์ของโมเมนตัม ( $m$ ) ตามสมการที่ 2.6 และ 2.7 ตามลำดับ

$$\Delta W_{iq,t} = (1-m) \cdot \eta \cdot \frac{\partial SSE}{\partial W_{iq,t}} + m \cdot \Delta W_{iq,(t-1)} \quad (2.6)$$

$$\Delta b_{j,t} = (1-m) \cdot \eta \cdot \frac{\partial SSE}{\partial b_{j,t}} + m \cdot \Delta b_{j,(t-1)} \quad (2.7)$$

เมื่อ  $SSE$  คือค่าผลรวมของค่าผิดพลาดยกกำลังสอง ซึ่งค่าผิดพลาดเป็นค่าแตกต่างระหว่างเอาต์พุตจริงและเอาต์พุตที่ได้จากโครงข่ายประสาทเทียม ค่าเริ่มต้นของน้ำหนัก, ค่าไบอัส, ค่าเทอมของโมเมนตัม และอัตราการเรียนรู้ จะมีผลต่อการประมวลผลของโครงข่ายประสาทเทียม

การเรียนรู้โดยวิธีของ Levenberg-Marquardt คัดแปลงมาจากวิธีของ Gauss-Newton non-linear least squares optimisation ซึ่งจะใช้การเรียนรู้แบบแพร่ย้อนกลับ (back-propagation learning) ซึ่งค่าน้ำหนัก “w” และค่าไบอัส “b” จะถูกปรับเปลี่ยนตามสมการที่ 2.8 และ 2.9 ตามลำดับ

$$w_{new} = w_{old} + \Delta w = w_{old} + [J^T(w) \cdot J(w) + \mu I]^{-1} \cdot J^T(w) \cdot e \quad (2.8)$$

โดยที่  $J(w)$  คือ จาโคเบียนเมทริกซ์ของอนุพันธ์ค่าผิดพลาดเทียบกับน้ำหนัก

$$J(w) = \begin{bmatrix} \frac{\partial e(1)}{\partial w_{3(1,1)}} & \frac{\partial e(1)}{\partial w_{3(1,2)}} & \dots & \frac{\partial e(1)}{\partial w_{3(1,S2)}} \\ \frac{\partial e(2)}{\partial w_{3(2,1)}} & \frac{\partial e(2)}{\partial w_{3(2,2)}} & \dots & \frac{\partial e(2)}{\partial w_{3(2,S2)}} \\ \vdots & \vdots & \dots & \vdots \\ \frac{\partial e(S3)}{\partial w_{3(S3,1)}} & \frac{\partial e(S3)}{\partial w_{3(S3,2)}} & \dots & \frac{\partial e(S3)}{\partial w_{3(S3,S2)}} \end{bmatrix}$$

$$b_{new} = b_{old} + \Delta b = b_{old} + [J^T(b) \cdot J(b) + \mu I]^{-1} \cdot J^T(b) \cdot e \quad (2.9)$$

โดยที่  $J(b)$  คือ จาโคเบียนเมทริกซ์ของอนุพันธ์ค่าผิดพลาดเทียบกับไบอัส

e คือ เมทริกซ์ของค่าผิดพลาดระหว่างเอาต์พุตจริงและเอาต์พุตที่ได้จากโครงข่ายประสาทเทียม

$\mu$  คือ ค่าคงที่ เช่น ค่าเริ่มต้นของ  $\mu$  มีค่าเท่ากับ 0.001

$$J(b) = \begin{bmatrix} \frac{\partial e(1)}{\partial b_3(1)} \\ \frac{\partial e(2)}{\partial b_3(2)} \\ \vdots \\ \frac{\partial e(S3)}{\partial b_3(S3)} \end{bmatrix}$$

ค่าคงที่ ( $\mu$ ) จะถูกเพิ่มขึ้นเป็นจำนวนเท่าเมื่อ  $SSE$  เพิ่มขึ้นจากระดับเดิมโดยตัวคูณ  $\alpha$  เช่น  $\alpha$  มีค่าเท่ากับ 10 และจะถูกลดลงเป็นจำนวนเท่าโดยตัวคูณเดียวกันเมื่อ  $SSE$  ลดลงจากระดับเดิม ถ้าค่าเฉลี่ยของ  $J(w).e$  หรือ  $J(b).e$  นั้นน้อยกว่าค่าน้อยที่สุดที่ได้กำหนดไว้ หรือค่า  $\mu$  นั้นมากกว่าค่ามากที่สุดที่ได้กำหนดไว้ การทำงานของโปรแกรมก็จะหยุดลงโดยอัตโนมัติ

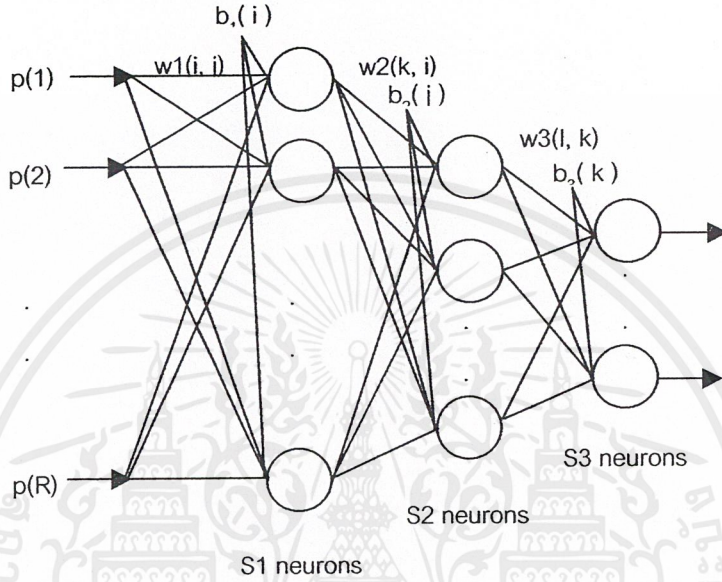
ค่าคงที่ ( $\mu$ ) จะนำมาใช้เริ่มต้นที่ทำการเรียนรู้และจะปรับเปลี่ยนในช่วงของกระบวนการเรียนรู้ เมื่อเริ่มต้นของกระบวนการเรียนรู้ โครงข่ายประสาทเทียมจะประมวลผลให้ค่าเอาต์พุตจากชุดข้อมูลอินพุตและเอาต์พุตจริงที่ได้ให้กับโครงข่าย เพื่อที่จะประมวลผลให้ตรงกับเอาต์พุตจริงนั้น ที่ทุกๆรอบของการเรียนรู้ ค่าผิดพลาด (error) จะถูกคำนวณโดยโครงข่ายและถูกนำไปใช้ในการปรับแก้ค่าน้ำหนักและค่าไบอัส ที่ไหนคเอาต์พุตหนึ่งในชั้นเอาต์พุต ผลรวมค่าความผิดพลาดยกกำลังสอง sum-squared error (SSE) ซึ่งให้ค่า  $e(1)$  สำหรับการเรียนรู้ข้อมูลอินพุต  $n$  ชุด สามารถจำกัดความได้ดังสมการที่ 2.10

$$SSE = \sum_{i=1}^n e_i(1)^2 \quad (2.10)$$

ความแตกต่างระหว่างการเรียนรู้โดยวิธีของ gradient descent และ Levenberg-Marquardt คือ สำหรับขั้นตอนวิธีของ Levenberg-Marquardt นั้น อินพุตทั้งหมดจะนำเข้าไปในโครงข่ายประสาทเทียมที่เวลาเดียวกันในแต่ละรอบของการเรียนรู้และค่าน้ำหนักและค่าไบอัสจะถูกปรับเปลี่ยนตามสมการที่ 2.8 และ 2.9 ตามลำดับ ในส่วนวิธีของ gradient descent ค่าน้ำหนักและค่าไบอัสจะปรับเปลี่ยนสำหรับแต่ละข้อมูลอินพุตในทุกๆรอบของการเรียนรู้ สำหรับในแต่ละรอบของการเรียนรู้ วิธีของ Levenberg-Marquardt ต้องใช้เวลาในการคำนวณและหน่วยความจำมากกว่าวิธีของ gradient descent อย่างไรก็ตามการปรับตัวเพื่อให้เข้ากับลักษณะเฉพาะของปัญหาโดยวิธีของ Levenberg-Marquardt จะเร็วกว่าและได้คำตอบในการแก้ปัญหาในจำนวนรอบที่น้อยกว่าวิธีของ gradient descent [1]

จากการทดสอบการเรียนรู้โดยวิธีเรียนรู้แบบแพร่ย้อนกลับ โดยเปรียบเทียบวิธีของ gradient descent และวิธีของ Levenberg-Marquardt เมื่อเปรียบเทียบกันในเรื่องของความถูกต้องของเอาต์พุตและเวลาในการเรียนรู้ สำหรับที่ค่าความถูกต้องที่เอาต์พุตเดียวกัน วิธีของ Levenberg-Marquardt จะใช้เวลาในการเรียนรู้น้อยกว่าวิธีของ gradient descent ดังนั้น ในการทดลองนี้ การปรับค่าน้ำหนักและไบอัสที่ใช้โครงข่ายประสาทเทียมแบบฟีดฟอร์เวิร์ด เราจึงใช้วิธีการเรียนรู้ของ Levenberg-Marquardt

จากรูปที่ 2-8 แสดงโครงข่ายประสาทเทียมแบบฟีดฟอร์เวิร์ด ซึ่งประกอบด้วย 3 ชั้น คือชั้นซ่อนที่ 1, ชั้นซ่อนที่ 2 และชั้นเอาต์พุต ซึ่งมีอินพุตทั้งหมด R อินพุต ในแต่ละชั้นจะมีจำนวนนิวรอน S1, S2 และ S3 นิวรอน ตามลำดับ



รูปที่ 2-8 แสดงโครงข่ายประสาทเทียมแบบฟีดฟอร์เวิร์ด 3 ชั้นประกอบด้วย R อินพุต

ค่าอินพุตรวมที่โหนดในชั้นซ่อนที่ 1 ซึ่งมี S1 นิวรอน จะมีค่าดังสมการที่ 2.11

$$n1(S1) = \sum_{j=1}^{j=R} w1(S1, j) \cdot p(j) + b_1(S1) \quad (2.11)$$

ซึ่งค่าเอาต์พุตที่ได้ในชั้นซ่อนที่ 1 มีค่าดังสมการที่ 2.12

$$a1(S1) = F1(n1(S1)) = F1\left(\sum_{j=1}^{j=R} w1(S1, j) \cdot p(j) + b_1(S1)\right) \quad (2.12)$$

ค่าอินพุตรวมที่โหนดในชั้นซ่อนที่ 2 ซึ่งมี S2 นิวรอน จะมีค่าดังสมการที่ 2.13

$$n2(S2) = \sum_{i=1}^{i=S1} w2(S2, i) \cdot a1(i) + b_2(S2) \quad (2.13)$$

ซึ่งค่าเอาต์พุตที่ได้ในชั้นซ่อนที่ 2 มีค่าดังสมการที่ 2.14

$$a_2(S_2) = F_2(n_2(S_2)) = F_2\left(\sum_{i=1}^{i=S_1} w_2(S_2, i) \cdot a_1(i) + b_2(S_2)\right) \quad (2.14)$$

ค่าอินพุตรวมที่โหนดในชั้นเอาต์พุตมีค่าดังสมการที่ 2.15

$$n_3(S_3) = \sum_{k=1}^{k=S_2} w_3(S_3, k) \cdot a_2(k) + b_3(S_3) \quad (2.15)$$

ซึ่งค่าเอาต์พุตที่ได้ในชั้นเอาต์พุตมีค่าดังสมการที่ 2.16

$$a_3(S_3) = F_3(n_3(S_3)) = F_3\left(\sum_{k=1}^{k=S_2} w_3(S_3, k) \cdot a_2(k) + b_3(S_3)\right) \quad (2.16)$$

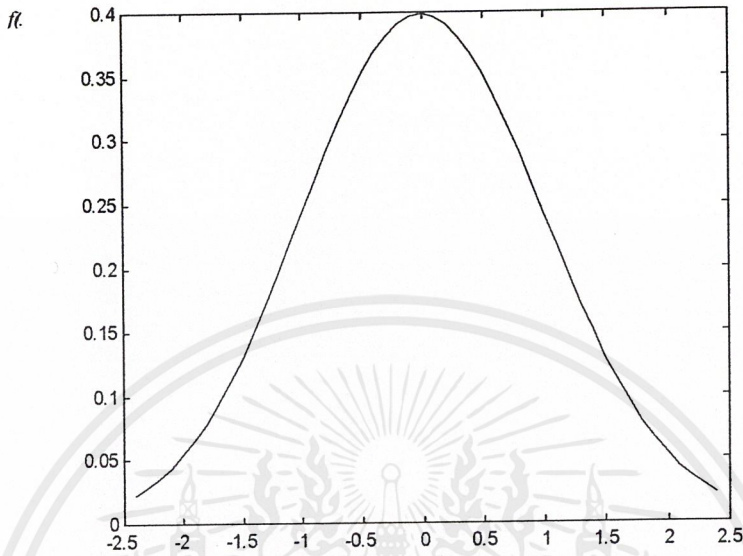
เมื่อให้ค่าเอาต์พุตที่ได้จากโครงข่ายประสาทเทียมในชั้นเอาต์พุตแต่ละโหนดมีค่า  $a_3(i)$  โดย  $i = 1, 2, \dots, S_3$ . และค่าเอาต์พุตจริงมีค่า  $t(i)$  โดย  $i = 1, 2, \dots, S_3$ . ค่าผิดพลาด  $e(i)$  สำหรับแต่ละโหนดในชั้นเอาต์พุตจะมีค่าดังสมการที่ 2.17

$$\begin{bmatrix} e(1) \\ e(2) \\ \vdots \\ e(S_3) \end{bmatrix} = \begin{bmatrix} t(1) - a_3(1) \\ t(2) - a_3(2) \\ \vdots \\ t(S_3) - a_3(S_3) \end{bmatrix} \quad (2.17)$$

## 2.10 การนอร์มัลไลเซชันของชุดข้อมูลที่ใช้เรียนรู้ (Normalisation of Training Data)

เป็นวิธีการจัดระเบียบข้อมูล การนอร์มัลไลเซชันชุดข้อมูลที่ใช้เรียนรู้ของโครงข่ายประสาทเทียมจะมีประโยชน์คือ ลดเวลาที่ใช้ในการเรียนรู้ และเพิ่มความถูกต้องของข้อมูลเอาต์พุตที่ได้จากโครงข่ายประสาทเทียม การนอร์มัลไลเซชันของชุดข้อมูลอินพุตและเอาต์พุตจะทำก่อนกระบวนการเรียนรู้ของโครงข่ายประสาทเทียม ซึ่งจะทำให้ค่าของข้อมูลอยู่บนพื้นฐานของ zero mean ( $\mu = 0$ ) และ unity variance ( $\sigma = 1$ )

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2-9 กราฟแสดงการกระจายค่าของข้อมูลจากการนอร์มัลไลเซชัน

สำหรับทุกจำนวนชุดข้อมูลอินพุตและเอาต์พุต ถ้าแต่ละอินพุตและเอาต์พุตนั้นประกอบด้วย “n” ชุดข้อมูล ( $X_1, X_2, X_3, \dots, X_n$ ) ดังนั้นสามารถนอร์มัลไลซ์ชุดข้อมูลโดยใช้สมการที่ 2.18 และ 2.19 ตามลำดับ

$$\text{If } X = \begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_n \end{bmatrix}, \mu_X = \frac{\sum_{i=1}^n X_i}{n} \text{ and If } Y = \begin{bmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_n \end{bmatrix} = \begin{bmatrix} X_1 - \mu_X \\ X_2 - \mu_X \\ \vdots \\ X_n - \mu_X \end{bmatrix}, \mu_Y = 0 \quad (2.18)$$

$$\sigma_Y = \sqrt{\frac{\sum_{i=1}^n (y_i)^2}{n}} \quad (2.19)$$

$$\text{If } Z = \begin{bmatrix} Z_1 \\ Z_2 \\ \vdots \\ Z_n \end{bmatrix} = Y \cdot \frac{1}{\sigma_Y} = \begin{bmatrix} (X_1 - \mu_X)/\sigma_Y \\ (X_2 - \mu_X)/\sigma_Y \\ \vdots \\ (X_n - \mu_X)/\sigma_Y \end{bmatrix}, \sigma_Z = 1 \text{ and } \mu_Z = 0$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หลังจากทุกชุดข้อมูลอินพุตและเอาต์พุตทำการนอร์มัลไลซ์ตามสมการที่ 2.18 และ 2.19 แล้ว ควรจะทำการสเกลหรือบาวด์ข้อมูล เพื่อจัดข้อมูลให้เป็นระเบียบอยู่ในช่วงที่เรากำหนด โดยใช้สมการที่ 2.20 ถึง 2.22 ซึ่งจะสร้างรูปแบบการกระจายของข้อมูลเป็นแบบลิเนียร์ ในการทดลองนี้จะทำการสเกลชุดข้อมูลอินพุตที่ใช้ในการเรียนรู้และใช้ทดสอบโครงข่ายประสาทเทียมให้มีค่าอยู่ระหว่าง  $[-0.9, 0.9]$  และชุดข้อมูลเอาต์พุตที่ใช้ในการเรียนรู้และใช้ทดสอบโครงข่ายประสาทเทียมให้มีค่าระหว่าง  $[0.1, 0.9]$

$$\text{จากเมตริกซ์ } [Z] = \begin{bmatrix} Z_1 \\ Z_2 \\ \vdots \\ Z_n \end{bmatrix}$$

กำหนดให้  $MX$  คือ ค่าสูงสุดของเมตริกซ์  $[Z]$

$MV$  คือ ค่าต่ำสุดของเมตริกซ์  $[Z]$

สามารถหาค่าคงที่  $aw$  และ  $bw$  ได้จากสมการที่ 2.20 และ 2.21

$$aw = \frac{(b-a)}{mx - mn} \quad (2.20)$$

$$bw = b - aw \cdot mx \quad (2.21)$$

โดยที่  $a$  และ  $b$  คือช่วงของการสเกลคือ  $[a, b]$

ดังนั้นแต่ละสมาชิกในเมตริกซ์  $[Z]$  ทำการสเกลโดยสมการที่ 2.22

$$\text{เอาต์พุตจากการสเกล} = aw \cdot Z_i + bw \quad \text{โดยที่ } i = 1, 2, \dots, n \quad (2.22)$$

ในระหว่างกระบวนการเรียนรู้ ค่าน้ำหนัก “ $w$ ” และค่าไบอัส “ $b$ ” จะปรับเปลี่ยนในทุก ๆ รอบของการเรียนรู้ เพื่อที่จะหาค่าที่เหมาะสมที่สุด ค่าความผิดพลาดกำลังสองเฉลี่ย (SSE) จะถูกนำมาใช้เพื่อหาค่าเอาต์พุต ถ้าเมตริกซ์  $[O]$  แสดงการนอร์มัลไลซ์และสเกลของค่าเอาต์พุตจากโครงข่ายประสาทเทียมในช่วงการเรียนรู้ แต่ละค่าของเอาต์พุตนั้นจะสามารถหาค่าจริงได้จากการทำ Unscaled และ Unnormalise ตามสมการที่ 2.23 และ 2.34 ตามลำดับ

$$[L] = \begin{bmatrix} O_1 / aw - bw / aw \\ O_2 / aw - bw / aw \\ \vdots \\ O_n / aw - bw / aw \end{bmatrix} \quad (2.23)$$

$$[Q] = \begin{bmatrix} L_1 \cdot \sigma_Y + \mu_X \\ L_2 \cdot \sigma_Y + \mu_X \\ \vdots \\ L_n \cdot \sigma_Y + \mu_X \end{bmatrix} \quad (2.24)$$

ค่าคงที่  $\mu_X$  และ  $\sigma_Y$  ในสมการที่ 2.24 มาจากสมการที่ 2.18 และ 2.19 ตามลำดับ โดยเมทริกซ์  $[Q]$  จะแสดง “m” ชุดข้อมูลเอาต์พุตที่ได้จากโครงข่ายประสาทเทียม ซึ่งสามารถนำไปเปรียบกับชุดข้อมูลเอาต์พุตจริง หลังจากการเรียนรู้เสร็จสิ้น ค่าน้ำหนัก “w” และ ไบอัส “b” จะถูกเก็บไว้ และสามารถนำไปใช้งานได้ ชุดข้อมูลทดสอบโครงข่ายประสาทเทียมนั้น จะต้องเป็นชุดข้อมูลที่โครงข่ายไม่เคยเรียนรู้มาก่อน ถ้ากำหนดให้นำมาทดสอบทั้งหมด “m” ชุดข้อมูล ชุดข้อมูลทดสอบนี้จะต้องทำการนอร์มัลไลซ์ ซึ่งใช้หลักของ zero mean ( $\mu = 0$ ) และ unity variance ( $\sigma = 1$ ) จากนั้นก็ทำการสเกลข้อมูลให้มีค่าอยู่ระหว่าง  $[-0.9, 0.9]$  ตามสมการข้างบนที่ได้กล่าวไว้แล้ว ชุดข้อมูลทดสอบที่ได้ก็จะนำไปใช้ทดสอบกับค่าน้ำหนัก “w” และ ไบอัส “b” เดียวกันกับที่เก็บไว้ ถ้าเมทริกซ์  $[T]$  แสดงการนอร์มัลไลซ์และสเกลของชุดข้อมูลเอาต์พุตทดสอบที่ได้จากโครงข่ายประสาทเทียม ดังนั้นแต่ละค่าของเอาต์พุตที่ได้จากโครงข่ายประสาทเทียมนั้นจะสามารถหาค่าจริงได้จากการทำ Unscaled และ Unnormalise ตามสมการที่ 2.25 และ 2.26 ตามลำดับ

$$[R] = \begin{bmatrix} T_1 / aw - bw / aw \\ T_2 / aw - bw / aw \\ \vdots \\ T_m / aw - bw / aw \end{bmatrix} \quad (2.25)$$

$$[H] = \begin{bmatrix} R_1 \cdot \sigma_Y + \mu_X \\ R_2 \cdot \sigma_Y + \mu_X \\ \vdots \\ R_m \cdot \sigma_Y + \mu_X \end{bmatrix} \quad (2.26)$$

เมทริกซ์  $[H]$  จะแสดง เอาท์พุทที่ได้จากโครงข่ายประสาทเทียม ของชุดข้อมูลทดสอบ “m” ชุด ซึ่งสามารถนำไปเปรียบเทียบกับเอาท์พุทจริงของชุดข้อมูลทดสอบ โดยค่าคงที่  $aw$ ,  $bw$ ,  $\mu_x$  และ  $\sigma_y$  ในสมการที่ 2.25 และ 2.26 จะเป็นค่าเดียวกันกับสมการที่ 2.18 ถึง 2.21



## บทที่ 3

### การทำนายโหลด

ปัจจุบันได้มีการศึกษานำการทำนายโหลดเพื่อนำมาใช้ในส่วนของความเสถียรภาพในระบบผลิตกำลังไฟฟ้า, ระบบส่งจ่ายกำลังงานไฟฟ้า, การวางแผนการทำงานของโรงจักรไฟฟ้าและตลอดจนการจัดการบริหารทางด้านพลังงาน เพื่อป้องกันอันตรายที่อาจจะเกิดขึ้นอันเนื่องมาจากการที่โรงจักรไฟฟ้าผลิตกำลังงานไฟฟ้าที่มีขนาดไม่เหมาะสมต่อความต้องการกำลังไฟฟ้าของโหลด อีกทั้งยังช่วยป้องกันการใช้ทรัพยากรอย่างสิ้นเปลือง และยังสามารถวางแผนการเพิ่มหรือขยายหน่วยผลิตเพื่อรองรับการขยายตัวของความต้องการกำลังไฟฟ้าที่นับวันจะมีความต้องการมากขึ้น

#### 3.1 ประเภทของการทำนายโหลด

การจำแนกประเภทของการทำนายโหลด สามารถจำแนกออกเป็นประเภทใหญ่ๆ ได้ตามช่วงระยะเวลาของการทำนายโหลดได้ 3 ประเภทดังต่อไปนี้ [2,3] คือ

##### 1) SHORT-TERM LOAD FORECASTING

หมายถึง การทำนายโหลดหรือการทำนายความต้องการกำลังไฟฟ้าที่จะเกิดขึ้นภายในระยะเวลาตั้งแต่ 1 ชั่วโมงล่วงหน้าจนถึง 168 ชั่วโมงหรือ 1 สัปดาห์ล่วงหน้าซึ่งโดยส่วนใหญ่แล้วผลของการทำนายโหลดประเภทนี้จะนำไปใช้ในส่วนของการวางแผนการทำงานของโรงจักรไฟฟ้า (Unit Commitment) หรือการวางแผนการผลิตของโรงจักรไฟฟ้า (Scheduling and Operative Reserve) เป็นต้น

##### 2) MEDIUM-TERM LOAD FORECASTING

หมายถึง การทำนายโหลดที่จะเกิดขึ้นตั้งแต่ระยะเวลา 1 สัปดาห์ล่วงหน้าจนถึง 3 ปีล่วงหน้า ซึ่งผลที่ได้จากการทำนายโหลดในช่วงเวลาดังกล่าวจะนำมาใช้ในส่วนของการวางแผนการบริหารเชื้อเพลิงการผลิต (Fuel Management) หรือการวางแผนการซ่อมบำรุง (Maintenance Planning) เป็นต้น

##### 3) LONG-TERM LOAD FORECASTING

หมายถึง การทำนายโหลดที่จะเกิดขึ้นในระยะเวลาตั้งแต่ 3 ปีขึ้นไปโดยส่วนมากแล้วผลที่ได้จากการทำนายในช่วงเวลาดังกล่าว จะนำมาใช้ในส่วนของการวางแผนการขยายหน่วยการผลิต (Capacity Expansion Planning) ในอนาคต

### 3.2 ปัจจัยที่มีอิทธิพลต่อการทำนายโหลด

ในการทำนายโหลดให้มีความถูกต้อง จำเป็นที่เราจะต้องคำนึงถึงปัจจัยต่างๆ ที่มีผลกระทบต่อ การเปลี่ยนแปลงของโหลดมาพิจารณาในการจำลองลักษณะของสมการทางคณิตศาสตร์ หรือจำลอง โมเดล (Model) ของระบบโครงข่ายประสาทเทียม (Artificial Neural Networks) ซึ่งปัจจัยต่างๆ ที่มีผล กระทบต่อการเปลี่ยนแปลงของโหลดนั้นประกอบด้วย [2,3]

- 1) โหลด (Load)
  - โหลดสูงสุดในแต่ละวัน (Daily peak load)
  - โหลด ณ เวลาใดๆ (Load at any time)
- 2) อุณหภูมิ (Temperature)
- 3) อุณหภูมิหยดน้ำค้าง (Dew-point temperature)
- 4) ความชื้น (Humidity)
- 5) ความเร็วลมและทิศทางของลม (Wind speed and direction)
- 6) ปริมาณเมฆ (Cloud)
- 7) ดัชนีแสง (Light index)
- 8) ช่วงระยะเวลาที่มีแสงแดด (Length of daylight)
- 9) ปริมาณน้ำฝน (Rain)
- 10) ฤดูกาล (Seasons)
- 11) วันในสัปดาห์ (Day of week)
- 12) วันหยุด (Holidays)
- 13) กิจกรรมทางเศรษฐกิจ (Economic Activities)
  - ผลผลิตทางอุตสาหกรรม (Industrial Production)
  - ผลผลิตมวลรวมของประเทศ (Gross Nation Product)
  - การบริโภคพลังงานดิบ (Raw Energy Consumption)
  - การบริโภควัตถุดิบ (Material Consumption)
- 14) ราคาเชื้อเพลิง (Cost of Energy)
- 15) การขยายตัวของประชากร (Population Distribution)
- 16) การขยายตัวของแรงงาน (Labor Distribution)
- 17) รายได้ต่อบุคคล (Income per person)
- 18) การบริโภคพลังงานไฟฟ้าต่อบุคคล (Electrical Energy Consumption per person)

### 3.3 ประโยชน์ของการทำนายโหลด

การทำนายโหลดนั้นมีความสำคัญและมีความจำเป็นควบคู่ไปกับการผลิตกำลังไฟฟ้า ซึ่งโดยทั่วไปแล้วสามารถจำแนกประโยชน์ของการทำนายได้ดังต่อไปนี้ [2] คือ

- 1) เพื่อใช้ในการวางแผนการทำงานของแต่ละโรงจักรไฟฟ้าให้สามารถทำการผลิตกำลังไฟฟ้าโดยที่ก่อให้เกิดค่าใช้จ่ายที่ต่ำ (Economic Load Dispatch)
- 2) เพื่อใช้ในการวางแผนการผลิตกำลังไฟฟ้าให้เพียงพอต่อความต้องการของโหลด
- 3) เพื่อใช้ในการวางแผนการขยายหรือสร้างโรงจักรไฟฟ้าหรือเพิ่มหน่วยการผลิตเพื่อรองรับการขยายตัวของความต้องการกำลังไฟฟ้าที่จะเพิ่มขึ้นในอนาคต
- 4) เพื่อใช้ในการวางแผนการแก้ปัญหาการขาดแคลนทรัพยากรหรือการทดแทนทรัพยากรในอนาคต
- 5) เพื่อใช้ในส่วนของการวางแผนการซ่อมบำรุงของโรงจักรไฟฟ้า
- 6) เพื่อใช้ในการวางแผนการสำรองเชื้อเพลิงการผลิตของโรงจักรไฟฟ้า
- 7) ทำให้ระบบส่งจ่ายกำลังไฟฟ้ามีความน่าเชื่อถือ
- 8) ก่อให้เกิดการใช้ทรัพยากรที่มีอยู่อย่างมีประสิทธิภาพ

### 3.4 ตัวอย่างการศึกษาการพยากรณ์และการวางแผนระบบไฟฟ้ากำลังที่ทำผ่านมาของการไฟฟ้าส่วนภูมิภาค

การพยากรณ์ที่ได้ทำผ่านมาของการไฟฟ้าส่วนภูมิภาค แบ่งเป็น 2 ประเภท ได้แก่

- การพยากรณ์ตามช่วงเวลา
- การพยากรณ์ตามวัตถุประสงค์

#### การพยากรณ์ตามช่วงเวลา

การไฟฟ้าส่วนภูมิภาค จะทำการพยากรณ์ตามช่วงเวลาปีละครั้ง โดยทำการพยากรณ์เมื่อมีข้อมูลการใช้ไฟฟ้าครบทั้งปีงบประมาณในแต่ละปี ซึ่งได้แบ่งการพยากรณ์ดังกล่าวเป็น 3 ระยะ ได้แก่

- การพยากรณ์ระยะสั้น

การพยากรณ์ในระยะ 1-3 ปี โดยนำผลการพยากรณ์ไปใช้ในการปฏิบัติการของศูนย์ควบคุมระบบไฟฟ้า การบำรุงรักษาและการจัดตั้งงบประมาณประจำปี

- การพยากรณ์ระยะปานกลาง

การพยากรณ์ในระยะ 5-10 ปี โดยนำผลการพยากรณ์ไปใช้ในการกำหนดแผนการขยายงานของระบบไฟฟ้า ที่ต้องพิจารณาตัดสินใจดำเนินการทันที

- การพยากรณ์ระยะยาว

การพยากรณ์ในระยะ 10-20 ปี โดยนำผลการพยากรณ์ไปใช้ในการกำหนดนโยบายต่างๆไป โดยเฉพาะนโยบายทางด้านพลังงานของประเทศ

ซึ่งการพยากรณ์ตามช่วงเวลาได้แบ่งการพยากรณ์ออกเป็น 2 ระดับ ได้แก่

- การพยากรณ์ระดับมหภาค (Macro Forecast)
- การพยากรณ์ระดับจุลภาค (Micro Forecast)

### การพยากรณ์ระดับมหภาค (Macro Forecast)

การพยากรณ์ระดับมหภาคเป็นการพยากรณ์ภาพรวมทั้งระบบของการไฟฟ้าส่วนภูมิภาค จำแนกเป็นภาคเหนือ ภาคตะวันออกเฉียงเหนือ ภาคกลาง และภาคใต้ โดยทำการพยากรณ์แยกตามประเภทผู้ใช้ไฟตามรายการที่การไฟฟ้าเก็บค่าไฟฟ้ามีทั้งหมด 9 ประเภท ได้แก่

1. บ้านอยู่อาศัย (Residential)
2. กิจการขนาดเล็ก (Small General Service)
3. กิจการขนาดกลาง (Medium General Service)
4. กิจการขนาดใหญ่และไฟสำรอง (Large General Service & Standby Rate)
5. ธุรกิจเฉพาะอย่าง (Specific Business)
6. ส่วนราชการและองค์กรที่ไม่แสวงหากำไร (Government Office & No Profit Organization)
7. สูบน้ำเพื่อการเกษตร (Agricultural Pumping)
8. ไฟฟ้าชั่วคราว (Temporary)
9. ไฟสาธารณะ หรือไฟอื่นที่ไม่ได้มีการเก็บค่าไฟฟ้า (Free of Charge)

### สิ่งที่พยากรณ์ในระดับมหภาค

การพยากรณ์ในระดับมหภาค (Macro Forecast) นั้นมีการพยากรณ์อยู่ 4 ส่วน ได้แก่

- การพยากรณ์จำนวนผู้ใช้ไฟ (Number of Customer Forecast)
- การพยากรณ์พลังงานไฟฟ้า (Energy Forecast)
- การพยากรณ์ความต้องการพยากรณ์ไฟฟ้าในส่วนที่รับจาก พพ. และในส่วนที่ กฟภ. ผลิตเอง (Power Purchase from DEDP Forecast and Power Generated by PEA Forecast)
- การพยากรณ์พลังงานไฟฟ้าสูงสุด (Peak Demand Forecast)

### วิธีการในการพยากรณ์ระดับมหภาค

ใช้วิธีการ Trend Curve

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ปัญหาและอุปสรรคที่พบในการพยากรณ์ระดับมหภาค (Macro Forecast)

ปัญหาที่สำคัญที่พบในการพยากรณ์ระดับมหภาค (Macro Forecast) ดังนี้

1. ข้อมูล Load Profile ที่สำรวจตั้งแต่ปี 2539 มีข้อมูลไม่ครบทุกประเภทของผู้ใช้ไฟฟ้า
2. การนำเอา Load Profile จากปี 2539 มาใช้ในการพยากรณ์ในปัจจุบันนี้อาจไม่ถูกต้องนัก สำหรับผู้ใช้ไฟบางประเภท เนื่องจาก มาตรการต่างๆ เพื่อจูงใจให้ผู้ใช้ไฟปรับเปลี่ยนพฤติกรรมการใช้ไฟฟ้า เช่น การเปลี่ยนอัตราค่าไฟฟ้า TOD เป็น TOU ซึ่งทำให้กิจการบางประเภท มีพฤติกรรมการใช้ไฟที่เปลี่ยนจากเดิมมาก
3. ความล่าช้าในการได้รับข้อมูลเพื่อใช้ในการพยากรณ์
4. รูปแบบของข้อมูลที่ได้รับส่วนใหญ่ไม่สามารถนำไปใช้ได้เลย ต้องทำการปรับเปลี่ยนรูปแบบ หรือต้องทำการป้อนข้อมูลใหม่ให้อยู่ในรูปแบบ Excel ก่อนจึงจะนำไปทำการพยากรณ์ได้

## การพยากรณ์ระดับจุลภาค (Micro Forecast)

การพยากรณ์ในระดับจุลภาค (Micro Forecast) ของ กฟภ. จะใช้วิธีการพยากรณ์แบบรวม โดยทำการพยากรณ์เป็นรายสถานี แล้วนำมารวมเป็นเขต ซึ่ง กฟภ. ได้แบ่งเขตการรับผิดชอบเป็น 4 ภาค คือ ภาคเหนือ ภาคตะวันออกเฉียงเหนือ ภาคกลาง ภาคใต้ และแต่ละภาคก็แบ่งเป็น 3 เขตย่อย ดังนั้นเขตการรับผิดชอบของการไฟฟ้าส่วนภูมิภาคจึงมี 12 เขตการรับผิดชอบ

ดังนั้นการพยากรณ์ความต้องการไฟฟ้าแบบรวมหรือการพยากรณ์ระดับจุลภาค (Micro) จะพยากรณ์แยกตามการไฟฟ้าเขตทั้ง 12 เขต โดยแยกตามสถานีไฟฟ้าของการไฟฟ้าเขตนั้นๆ ผลการพยากรณ์ที่ได้จะเป็น ความต้องการพลังงานไฟฟ้า (Energy – GWh) และความต้องการพลังงานไฟฟ้า (Demand - MW) วิธีการ (Methodology) ในการพยากรณ์ จะใช้วิธีการ Trend Curve และประสบการณ์ โดยการ Trend Curve จะใช้ข้อมูลในอดีต อัตราการเพิ่มของการพยากรณ์ระดับมหภาค (Macro) และการพยากรณ์ในอดีตที่ผ่านมาประกอบกัน การพยากรณ์นี้จะเป็นการพยากรณ์แบบระยะยาว (Long – Term) ซึ่งมีจำนวนปีในการพยากรณ์ 15 ปี ในการพยากรณ์นี้จะต้องพยากรณ์สถานีไฟฟ้าโครงการ (Planning Substation) ด้วย โดยแบ่งมาจากสถานีไฟฟ้าปัจจุบัน (Existing Substation) สถานีโครงการนี้จะอยู่ในโครงการที่เรียกว่า โครงการพัฒนาระบบสายส่งและสถานีย่อย (Transmission System And Substation Development Project) โดยมีชื่อย่อว่า คสพ. หรือ TSD. ซึ่งสถานีโครงการดังกล่าวจะแบ่งเป็น 7 ระยะ

## สิ่งที่พยากรณ์ในระดับจุลภาค

การพยากรณ์ในระดับจุลภาค (Micro Forecast) ได้แก่

- การพยากรณ์สถานีไฟฟ้าปัจจุบันของ กฟผ.(EGAT. Existing substation)
- การพยากรณ์สถานีไฟฟ้าปัจจุบันของ กฟภ.(PEA. Existing Substation)

เอกสารนี้เป็น • การพยากรณ์สถานีไฟฟ้าโครงการเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### วิธีการในการพยากรณ์ระดับจุลภาค

วิธีการในการพยากรณ์จะใช้การ Trend Curve และประสบการณ์ในการพยากรณ์ โดยจะพยากรณ์สถานีไฟฟ้าปัจจุบันก่อน (Existing Substation) แล้วจึงพยากรณ์สถานีไฟฟ้าโครงการ(Project Substation) ด้วยการแบ่งมาจากสถานีไฟฟ้าปัจจุบัน

### ปัญหาและอุปสรรคที่พบในการพยากรณ์ระดับจุลภาค (Micro Forecast)

ปัญหาส่วนใหญ่ที่พบในการพยากรณ์ระดับจุลภาค (Micro Forecast) เนื่องจากเรื่องข้อมูลที่ใช้ในการพยากรณ์ ซึ่งทำให้การพยากรณ์เกิดความล่าช้าและผิดพลาด ดังนี้

1. ข้อมูลที่ใช้ในการพยากรณ์ที่ต้องการจากหน่วยงานอื่น ไม่มีการปรับปรุงข้อมูลให้ทันกับปัจจุบัน (Up-date) และข้อมูลไม่ถูกต้องอันเนื่องมาจากผู้ป้อนข้อมูลหรือหน้างาน เช่น ข้อมูลสภาพจ่ายไฟ ข้อมูล Load01 เป็นต้น
2. สภาพจ่ายไฟของสถานีไฟฟ้าบางสถานีไฟฟ้า ทั้งทางด้าน 22 kV และ 115 kV ซึ่งมีการเปลี่ยนแปลงตลอด แต่ไม่มีการรายงานจากหน้างานก่อนการตัดจ่ายโหลด
3. ใช้เวลาค่อนข้างมากในการขอข้อมูลที่ต้องใช้ในการทำการพยากรณ์จากหน่วยงานอื่นๆ
4. รูปแบบของข้อมูลที่แผนกพยากรณ์ได้รับเพื่อจะใช้ทำการพยากรณ์ ส่วนใหญ่ต้องเสียเวลาในการป้อนข้อมูลใหม่หรือแปลงจากรูแบบเดิมให้อยู่ในรูปแบบ Excel เนื่องจากข้อมูลที่ได้รับเดิมบางอย่างอยู่ในระบบฐานข้อมูลที่สามารถดูได้อย่างเดียว หรืออยู่ในโปรแกรมไม่สามารถนำข้อมูลไปใช้ได้เลย หรือไม่อยู่ในรูปของ Electronic File
5. เนื่องจากสถานี 115 kV ของการไฟฟ้าส่วนภูมิภาค ไม่มีการติดตั้ง Energy Demand Meter จึงไม่ทราบปริมาณพลังงานไฟฟ้า (Energy Demand) ที่แท้จริง

### วิธีการในการพยากรณ์ระดับจุลภาค

วิธีการในการพยากรณ์จะใช้การ Trend Curve และประสบการณ์ในการพยากรณ์ โดยจะพยากรณ์สถานีไฟฟ้าปัจจุบันก่อน (Existing Substation) แล้วจึงพยากรณ์สถานีไฟฟ้าโครงการ(Project Substation) ด้วยการแบ่งมาจากสถานีไฟฟ้าปัจจุบัน

### การพยากรณ์ตามวัตถุประสงค์

การไฟฟ้าฯ จะทำการพยากรณ์ตามวัตถุประสงค์เมื่อมีโครงการขยายเขตระบบจำหน่าย และต้องการข้อมูลการพยากรณ์เพื่อประกอบการตัดสินใจในการดำเนินโครงการ โดยที่ข้อมูลดังกล่าวไม่มีอยู่ในรายงานผลการพยากรณ์ตามช่วงเวลา ซึ่งโครงการขยายเขตระบบจำหน่ายให้หมู่บ้านในชนบท และโครงการขยายเขตระบบจำหน่ายตามเกาะต่าง ๆ โดยการพยากรณ์ตามวัตถุประสงค์นั้นได้แบ่งออกเป็น 3

เอกสารแนบนี้ได้แก่สารที่ส่งจนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### การพยากรณ์พลังงานไฟฟ้า (Energy Forecast)

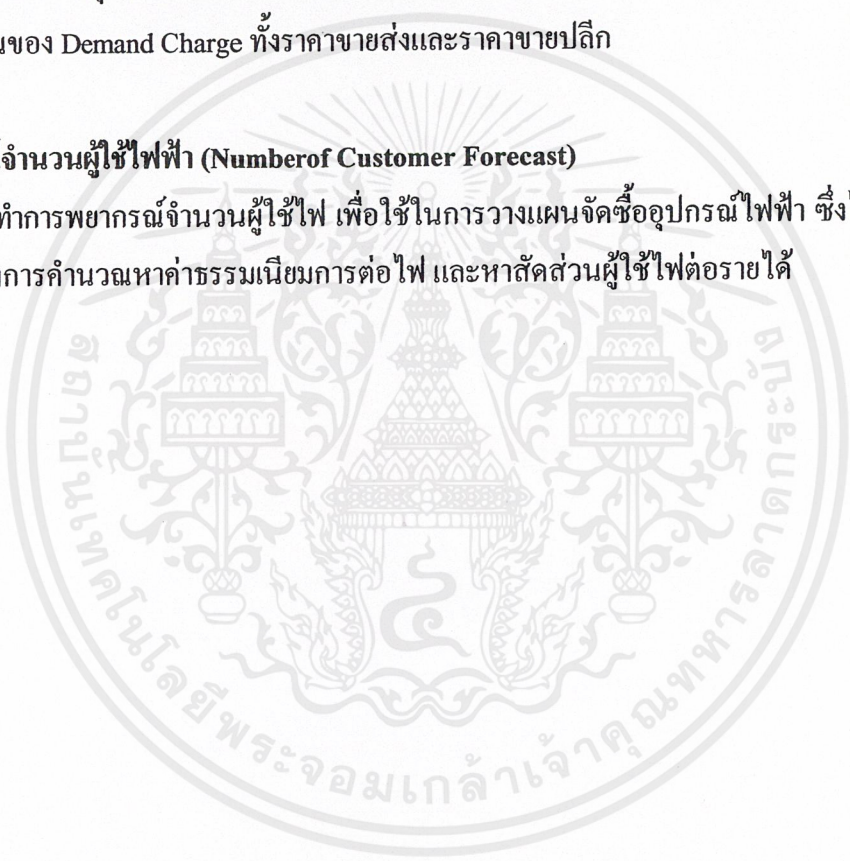
การทำการพยากรณ์พลังงานไฟฟ้าเพื่อกำหนดชนิดของโรงไฟฟ้า ชนิดของเชื้อเพลิงที่ใช้ในโรงไฟฟ้า การคำนวณรายได้และรายจ่าย การวิเคราะห์สถานะการเงิน การจัดทำโครงสร้างอัตราค่ากระแสไฟฟ้าในส่วนของ Energy Charge ทั้งราคาขายส่งและขายปลีก

### การพยากรณ์พลังงานไฟฟ้าสูงสุด (Peak Demand Forecast)

การทำการพยากรณ์พลังงานไฟฟ้าสูงสุดเพื่อกำหนดขนาดของสถานีไฟฟ้าหรือกำลังผลิตของโรงไฟฟ้า ขนาดของอุปกรณ์ไฟฟ้าในระบบสายส่ง และระบบจำหน่าย จัดทำโครงสร้างอัตราค่ากระแสไฟฟ้าในส่วนของ Demand Charge ทั้งราคาขายส่งและราคาขายปลีก

### การพยากรณ์จำนวนผู้ใช้ไฟฟ้า (Number of Customer Forecast)

การทำการพยากรณ์จำนวนผู้ใช้ไฟ เพื่อใช้ในการวางแผนจัดซื้ออุปกรณ์ไฟฟ้า ซึ่งได้แก่ มิเตอร์ ซีทีพีที รวมถึงการคำนวณหาค่าธรรมเนียมการต่อไฟ และหาสัดส่วนผู้ใช้ไฟต่อรายได้



## บทที่ 4

### การทำนายการใช้กำลังไฟฟ้าโดยใช้โครงข่ายประสาทเทียม

#### 4.1 การจัดเตรียมโครงข่ายประสาทเทียม

โครงข่ายประสาทเทียมที่ได้ใช้ในการทดลองครั้งนี้ ได้ใช้โครงข่ายประสาทเทียมที่มีลักษณะการทำงานแบบฟีดฟอร์เวิร์ด (Feed-Forward Neural Networks) และใช้การเรียนรู้แบบแพร่ย้อนกลับ (Back-Propagation Learning) ซึ่งฟังก์ชันกระตุ้นที่ใช้ในการทดลอง ได้ใช้ฟังก์ชันกระตุ้นแบบเป็นเชิงเส้น (Linear Active Function) ได้แก่ ฟังก์ชันลิเนียร์ (Linear Function) และฟังก์ชันกระตุ้นแบบไม่เป็นเชิงเส้น (Nonlinear Active Function) ได้แก่ ฟังก์ชันซิกมอยด์ (Sigmoid Function) ทั้งนี้เนื่องจากฟังก์ชันกระตุ้นที่ไม่เป็นเชิงเส้นนั้นมีความสามารถในการสร้างขอบเขตของการตัดสินใจที่มีลักษณะโค้งมนได้ จึงทำให้สามารถทดสอบกับรูปแบบของโครงข่ายประสาทเทียมที่ค่อนข้างซับซ้อนได้ ซึ่งจะทำให้มีความเหมาะสมกับการทำนายโหลด

ในการจัดเตรียมโครงข่ายประสาทเทียมเพื่อใช้ในการศึกษาการทำนายการใช้กำลังไฟฟ้าในครั้งนี้ ได้ทำการทดลองด้วยโครงข่ายประสาทเทียมที่มีชั้นซ่อน 1 ชั้นและ ชั้นซ่อน 2 ชั้น โดยที่โครงสร้างแต่ละโครงข่ายจะมีรูปแบบที่แตกต่างกันตามจำนวนนิวรอนและรูปแบบของฟังก์ชันกระตุ้นที่ใช้ในแต่ละชั้นของโครงข่ายประสาทเทียม

การเตรียมชุดข้อมูลที่ใช้ในการทดลองนี้ ได้ใช้ข้อมูลการใช้พลังงานไฟฟ้าจากสถานีไฟฟ้าย่อย นิคมอุตสาหกรรมอมตะนคร โดยนำข้อมูลที่เก็บบันทึกการใช้พลังงานไฟฟ้าเป็นรายชั่วโมงในแต่ละวัน ช่วงฤดูร้อน ตั้งแต่วันที่ 1 มีนาคม – 31 พฤษภาคม 2544 รวมทั้งหมด 92 วัน และใช้ข้อมูลอุณหภูมิจาก สถานีกรมอุตุฯ ในจังหวัดชลบุรี โดยนำข้อมูลที่เก็บบันทึกอุณหภูมิสูงสุดและต่ำสุดในแต่ละวัน ตั้งแต่วันที่ 1 มีนาคม – 31 พฤษภาคม 2544 เพื่อนำมาจัดเตรียมข้อมูลที่ใช้ในการเรียนรู้ของโครงข่ายประสาทเทียม โดยจะแบ่งชุดข้อมูลออกเป็น 2 ชุด คือ ชุดข้อมูลที่ใช้สำหรับการเรียนรู้ของโครงข่ายประสาทเทียมซึ่งจะใช้ทั้งหมด 84 วัน (ตั้งแต่วันที่ 1 มีนาคม – 23 พฤษภาคม) และชุดข้อมูลทดสอบโครงข่ายประสาทเทียม 8 วัน (ตั้งแต่วันที่ 24 พฤษภาคม – 31 พฤษภาคม)

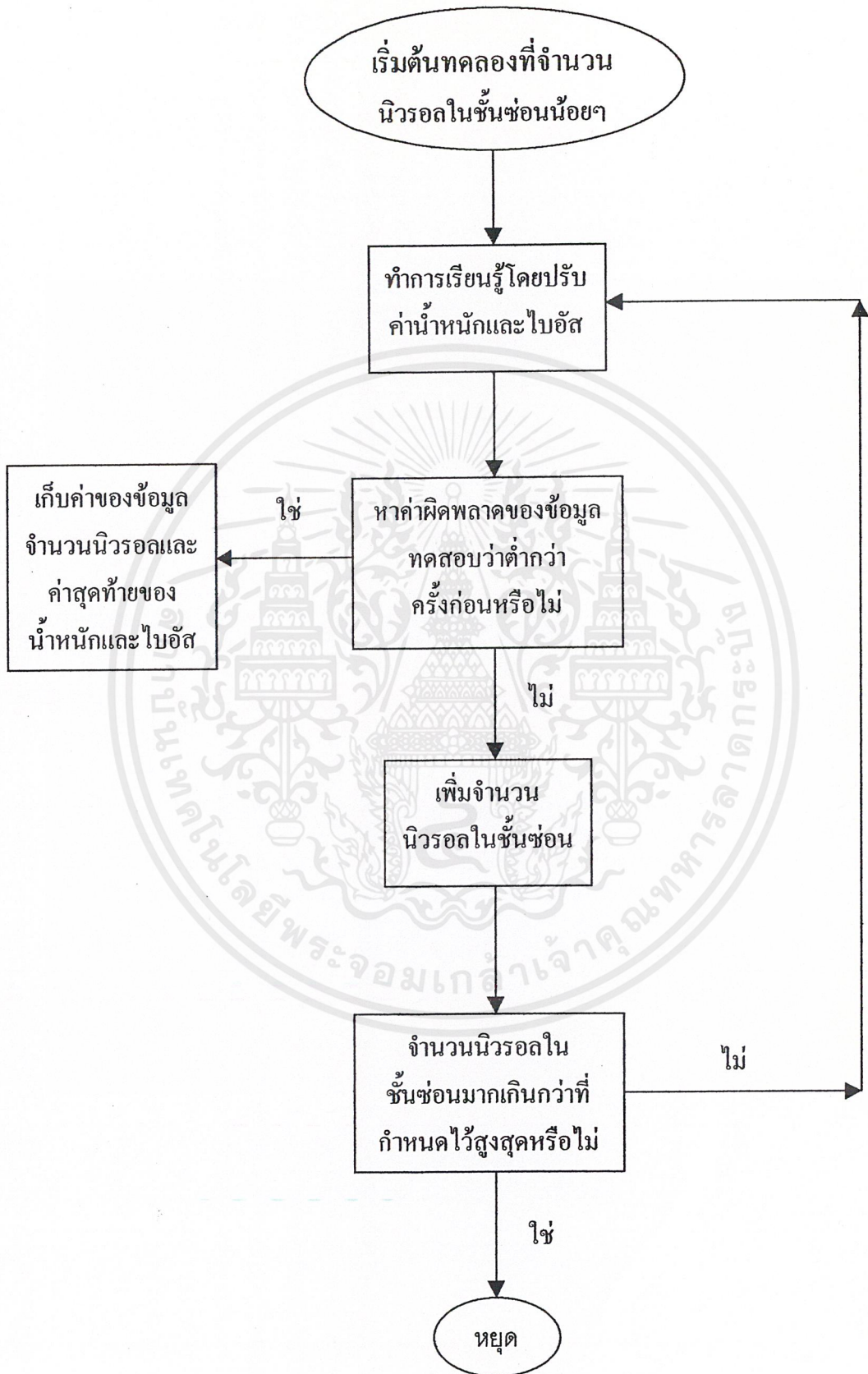
#### 4.2 การจัดเตรียมข้อมูลที่ใช้ในการเรียนรู้ของโครงข่ายประสาทเทียม

ในการทดลองนี้จะกำหนดให้ชุดข้อมูลอินพุตมีทั้งหมด 58 อินพุต ซึ่งประกอบด้วย

1. ข้อมูลของความต้องการพลังงานไฟฟ้ารายชั่วโมงก่อนวันทำนาย 2 วัน ประกอบด้วย 24 อินพุต
2. ข้อมูลของความต้องการพลังงานไฟฟ้ารายชั่วโมงก่อนวันทำนาย 1 วัน ประกอบด้วย 24 อินพุต
3. อุณหภูมิสูงสุด และต่ำสุดของวันก่อนการทำนาย 2 วัน ประกอบด้วย 2 อินพุต
4. อุณหภูมิสูงสุด และต่ำสุดของวันก่อนการทำนาย 1 วัน ประกอบด้วย 2 อินพุต
5. อุณหภูมิสูงสุด และต่ำสุดของวันที่ทำการทำนาย ประกอบด้วย 2 อินพุต
6. การกำหนดวัน (Day Code) ในรอบสัปดาห์ของการทำนาย โดยจะกำหนดค่าของวันที่ทำนาย โหลด (จันทร์ – อาทิตย์) ให้เป็น binary code 3 บิต (ตั้งแต่ 001 – 111) ซึ่งประกอบด้วย 3 อินพุต
7. การแสดงค่าวันหยุด (Holiday) ของวันที่ทำนาย โหลด โดยวันที่ทำนายโหลดเป็นวันหยุดการทำงานจะกำหนดให้เป็น 1 ถ้าไม่ได้เป็นวันหยุดการทำงานจะให้เป็น 0 ซึ่งประกอบด้วย 1 อินพุต

#### 4.3 การเรียนรู้ของโครงข่ายประสาทเทียม

โครงข่ายประสาทเทียมสามารถเรียนรู้ได้จาก ชุดของข้อมูลอินพุตและเอาต์พุตที่ให้กับโครงข่าย โดยการเรียนรู้เริ่มต้น โครงข่ายประสาทเทียมจะทำการสุ่มค่าน้ำหนักและไบอัส เพื่อนำไปใช้ในการประมวลผลให้ค่าเอาต์พุตจากโครงข่าย โดยการเรียนรู้แต่ละรอบนั้น โครงข่ายประสาทเทียมจะสร้างผลตอบสนองเอาต์พุต ซึ่งอาจมีค่าแตกต่างจากค่าเอาต์พุตจริง จะทำให้เกิดค่าผิดพลาดขึ้น ในการเรียนรู้รอบต่อไปนั้นจะใช้ค่าผิดพลาดที่เกิดขึ้นนำไปปรับค่าน้ำหนักและไบอัส เพื่อที่จะทำให้ค่าผิดพลาดที่เกิดขึ้นในรอบต่อไปมีค่าลดลง เป็นผลให้การเรียนรู้ของโครงข่ายประสาทเทียมนั้นดีขึ้น หลังจากการเรียนรู้เสร็จสิ้นก็จะทำการทดสอบโครงข่ายประสาทเทียม หากค่าผิดพลาดจากการทดสอบเช่นเดียวกับการเรียนรู้ โครงข่ายประสาทเทียมที่ให้ค่าผิดพลาดต่ำที่สุด จะสามารถนำไปใช้ในการทำนายโหลดได้ โดยทำการเก็บค่าน้ำหนักและค่าไบอัสของโครงสร้างที่ให้ค่าผิดพลาดจากการทดสอบต่ำที่สุด กระบวนการทำงานของโครงข่ายประสาทเทียม สามารถแสดงได้ดังรูปที่ 4-1 ซึ่งในการทดลองนี้เขียนโปรแกรมทั้งหมดโดยใช้ Neural Networks Toolbox ในโปรแกรม Matlab [4]



## บทที่ 5

### ผลการวิเคราะห์โครงข่ายประสาทเทียม

#### 5.1 ผลการเรียนรู้โครงข่ายประสาทเทียม

จากการทดลอง “การทำนายโพลคระยะสั้นโดยใช้โครงข่ายประสาทเทียม” ในครั้งนี้ ได้ทำการทดสอบการทำงานโดยใช้การเขียนโปรแกรมด้วยโปรแกรม ‘MATLAB’ Version 6.3 โดยปฏิบัติงานด้วยคอมพิวเตอร์ส่วนบุคคล ประสิทธิภาพ CPU Intel Celeron Processor 533 MHz หน่วยความจำ 256 MB บน OS (Operating System) Windows 98 SE และมีการใช้โปรแกรม Microsoft Access เพื่อเป็นฐานข้อมูลสำหรับเก็บข้อมูลและผลการทดลอง

จากการพิจารณาการเรียนรู้ของโครงข่ายประสาทเทียมนั้นพบว่า โครงข่ายจะมีการปรับปรุงค่าความผิดพลาดกำลังสองเฉลี่ย (Sum Squared Error) ให้มีค่าลดลงเข้าสู่ความผิดพลาดที่สามารถยอมรับได้ โดยที่ในช่วงเริ่มต้นของการเรียนรู้ค่าความผิดพลาดกำลังสองเฉลี่ยจะมีการลดลงอย่างรวดเร็ว และจะค่อยๆปรับเรียบลงในรอบการเรียนรู้ต่อมา

ในการเรียนรู้ของโครงข่ายประสาทเทียม ได้ใช้การปรับค่าพารามิเตอร์ที่สำคัญ คือ ค่าคงที่ ( $\mu$ ) และค่าจาโคเบียนเมตริกซ์ของอนุพันธ์ค่าผิดพลาดเทียบกับน้ำหนักและไบอัส ( $J(w)$ ,  $J(b)$ ) เพื่อช่วยเร่งกระบวนการเรียนรู้ของโครงข่ายประสาทเทียมให้มีความรวดเร็วมากขึ้น

ผลการทดสอบการเรียนรู้ของโครงข่ายประสาทเทียมที่ได้แสดงให้เห็นในบทนี้ ได้แสดงให้เห็นถึงความสามารถในการเรียนรู้ของโครงข่ายประสาทเทียม (ตั้งแต่วันที่ 1 มีนาคม – 23 พฤษภาคม) หลังจากโครงข่ายประสาทเทียมเรียนรู้เสร็จแล้ว ก็จะมีการทดสอบการทำนายโพลทั้งหมด 8 วัน (ตั้งแต่วันที่ 24 – 31 พฤษภาคม 2544) โดยที่ผลของการทดลองแต่ละโครงข่ายจะมีความแตกต่างกันตามจำนวนโหนดภายในและรูปแบบของฟังก์ชันกระตุ้นที่ใช้ในโครงข่ายประสาทเทียม ซึ่งผลการทดลองแสดงดังตารางที่ 5-1 ถึง 5-13 ในส่วนของตารางที่ 5-1 ถึง 5-5 จะแสดงผลของการทดลองโครงข่ายประสาทเทียมที่ประกอบด้วย 1 ชั้นซ่อน และตารางที่ 5-6 ถึง 5-7 แสดงถึงผลการทดลองของโครงข่ายที่ประกอบด้วย 2 ชั้นซ่อน ซึ่งค่าผิดพลาดสมบูรณ์ (Absolute percentage error) ที่คิดเป็นเปอร์เซ็นต์ในตารางที่ 5-1 ถึง 5-13 สามารถหาได้จาก

$$\text{Absolute percentage error (ape)} = \left| \frac{\text{ค่าโพลจริง} - \text{ค่าโพล การทำนาย}}{\text{ค่าโพลจริง}} \right| \times 100 (\%)$$

ตารางที่ 5-1 แสดงผลการทำนายโรคของโครงข่ายประสาทเทียม 2 ชั้น โดยใช้โหนดในชั้นซ่อน 4 โหนด และชั้นเอาต์พุต 24 โหนด

รูปแบบฟังก์ชันกระตุ้นในแต่ละชั้น		เวลาที่ใช้ เรียนรู้ (นาที)	ค่า ape จากการเรียนรู้ (%)			ค่า ape จากการทดสอบ (%)		
ชั้นซ่อน	ชั้นเอาต์พุต		ต่ำสุด	สูงสุด	เฉลี่ย	ต่ำสุด	สูงสุด	เฉลี่ย
purelin	purelin	1.8538	0.0051	131.8515	8.5399	0.0686	54.3453	7.7574
logsig	purelin	0.91	0.008	122.6273	5.4349	0.0087	58.0595	6.1377
tansig	purelin	0.8065	0.0092	133.2478	6.4447	0.0418	53.5068	6.431
logsig	logsig	0.6637	0.0122	301.0671	11.3761	0.0407	82.779	11.6902
purelin	logsig	0.7342	0.0175	128.8214	8.4625	0.0666	53.3431	7.494
tansig	logsig	0.8623	0.0054	85.9328	6.0904	0.0137	58.7276	9.0317
tansig	tansig	1.2175	0.0022	104.5621	6.0837	0.0432	57.1539	6.1443
purelin	tansig	1.787	0.0036	106.0781	8.0532	0.024	52.1544	8.2879
logsig	tansig	1.6387	5.20E-04	480.6818	13.2303	0.0758	155.5	9.1262

ตารางที่ 5-2 แสดงผลการทำนายโรคของโครงข่ายประสาทเทียม 2 ชั้น โดยใช้โหนดในชั้นซ่อน 8 โหนด และชั้นเอาต์พุต 24 โหนด

รูปแบบฟังก์ชันกระตุ้นในแต่ละชั้น		เวลาที่ใช้ เรียนรู้ (นาที)	ค่า ape จากการเรียนรู้ (%)			ค่า ape จากการทดสอบ (%)		
ชั้นซ่อน	ชั้นเอาต์พุต		ต่ำสุด	สูงสุด	เฉลี่ย	ต่ำสุด	สูงสุด	เฉลี่ย
purelin	purelin	1.7465	0.0287	130.0493	8.35	0.0393	31.6327	7.2106
logsig	purelin	3.1473	0.0089	64.7853	7.1241	0.06	30.9074	4.566
tansig	purelin	3.2452	0.0027	82.9611	7.4117	0.0173	31.6055	5.2437
logsig	logsig	5.2114	0.0024	98.2356	6.7852	0.0477	78.5241	6.9852
purelin	logsig	4.9872	0.0087	202.4787	10.8756	0.0589	179.5682	12.8745
tansig	logsig	5.8767	0.0044	78.6522	6.4754	0.0197	66.8543	6.7884
tansig	tansig	5.5703	0.0019	81.1738	5.1149	0.0249	57.7053	5.5385
purelin	tansig	5.4523	0.0045	67.7432	9.4478	0.0288	44.7325	7.3362
logsig	tansig	3.4503	0.0017	403.125	12.9872	0.0169	163.1	9.763

ตารางที่ 5-3 แสดงผลการทำนายโหนดของโครงข่ายประสาทเทียม 2 ชั้น โดยใช้โหนดในชั้นซ่อน 16 โหนด และชั้นเอาต์พุต 24 โหนด

รูปแบบฟังก์ชันกระตุ้นในแต่ละชั้น		เวลาที่ใช้ เรียนรู้ (นาที)	ค่า ape จากการเรียนรู้ (%)			ค่า ape จากการทดสอบ (%)		
ชั้นซ่อน	ชั้นเอาต์พุต		ต่ำสุด	สูงสุด	เฉลี่ย	ต่ำสุด	สูงสุด	เฉลี่ย
purelin	purelin	19.8754	0.02387	104.5682	10.0452	0.1475	35.4231	8.7966
logsig	purelin	20.4589	0.0154	98.6534	5.4236	0.1056	22.4922	4.3245
tansig	purelin	24.7245	0.0121	122.3537	7.5047	0.1674	20.223	6.2302
logsig	logsig	30.8754	0.01475	86.5321	8.5324	0.1247	31.542	5.6387
purelin	logsig	29.6534	0.04583	135.6897	11.4521	0.4532	57.6324	9.6534
tansig	logsig	24.3589	0.0166	86.5247	6.3245	0.0986	19.5486	4.9638
tansig	tansig	21.3568	0.0132	107.8944	8.6354	0.1356	32.8547	6.5874
purelin	tansig	27.5634	0.0178	106.6635	8.6354	0.1684	27.6352	7.5638
logsig	tansig	31.5482	0.0174	92.1358	8.4521	0.1654	24.5637	6.1325

ตารางที่ 5-4 แสดงผลการทำนายโหลดของโครงข่ายประสาทเทียม 2 ชั้น โดยใช้โหนดในชั้นซ่อน 30 โหนด และชั้นเอาต์พุท 24 โหนด

รูปแบบฟังก์ชันกระตุ้นในแต่ละชั้น		เวลาที่ใช้ เรียนรู้ (นาที)	ค่า ape จากการเรียนรู้ (%)			ค่า ape จากการทดสอบ (%)		
ชั้นซ่อน	ชั้นเอาต์พุท		ต่ำสุด	สูงสุด	เฉลี่ย	ต่ำสุด	สูงสุด	เฉลี่ย
purelin	purelin	132.4578	0.0704	26.3541	4.5968	0.0909	27.5634	5.6342
logsig	purelin	144.5362	0.0423	20.4571	3.5647	0.0746	22.3514	5.6387
tansig	purelin	187.6354	0.0214	26.3541	1.4625	0.0154	25.6389	4.9867
logsig	logsig	145.8693	0.0326	24.5871	3.9654	0.0563	27.5436	5.6387
purelin	logsig	123.5892	0.0045	16.3548	3.2145	0.0248	18.5347	4.6345
tansig	logsig	147.6352	0.0013	10.2478	2.6354	0.0094	16.3548	3.9864
tansig	tansig	195.6342	0.0026	16.5498	1.8754	0.0056	15.3897	3.7854
purelin	tansig	97.6548	0.0054	19.6173	3.2568	0.0425	24.3691	5.6342
logsig	tansig	163.2579	0.0097	18.5942	3.5741	0.0245	22.1564	4.8135

ตารางที่ 5-5 แสดงผลการทำนายโหนดของโครงข่ายประสาทเทียม 2 ชั้น โดยใช้โหนดในชั้นซ่อน 35 โหนด และชั้นเอาต์พุต 24 โหนด

รูปแบบฟังก์ชันกระตุ้นในแต่ละชั้น		เวลาที่ใช้ เรียนรู้ (นาที)	ค่า ape จากการเรียนรู้ (%)			ค่า ape จากการทดสอบ (%)		
ชั้นซ่อน	ชั้นเอาต์พุต		ต่ำสุด	สูงสุด	เฉลี่ย	ต่ำสุด	สูงสุด	เฉลี่ย
purelin	purelin	287.2578	0.0074	20.2548	3.1564	0.09464	23.5468	6.7894
logsig	purelin	225.3654	0.0056	19.3658	2.5498	0.08765	24.5687	5.6384
tansig	purelin	236.4981	0.0035	23.5684	3.5421	0.06321	19.6582	5.6234
logsig	logsig	244.3658	0.0029	16.5487	2.6354	0.04644	14.3256	4.3265
purelin	logsig	175.6384	0.0013	13.2457	1.1259	0.03254	16.3598	3.9648
tansig	logsig	246.8954	0.0062	14.5689	1.4587	0.01277	17.3652	4.3568
tansig	tansig	279.3658	0.0024	12.3654	2.3547	0.00986	14.2444	3.7865
purelin	tansig	254.7543	0.0098	14.5632	1.6348	0.0235	18.5247	4.9974
logsig	tansig	198.6354	0.0021	15.6321	1.5468	0.0253	18.6547	4.1268

ตารางที่ 5-6 แสดงผลการทำนายโหนดของโครงข่ายประสาทเทียม 3 ชั้น โดยใช้โหนดชั้นซ่อนแรก 4 โหนด, ชั้นซ่อนที่สอง 4 โหนด และชั้นเอาต์พุต 24 โหนด

รูปแบบฟังก์ชันกระตุ้นในแต่ละชั้น		เวลาที่ใช้ เรียนรู้ (นาที)	ค่า ape จากการเรียนรู้ (%)				ค่า ape จากการทดสอบ (%)		
ชั้นซ่อนแรก	ชั้นซ่อนที่สอง		ชั้นเอาต์พุต	ต่ำสุด	สูงสุด	เฉลี่ย	ต่ำสุด	สูงสุด	เฉลี่ย
purelin	purelin	purelin	purelin	0.0028	131.5395	8.5565	0.082	52.3096	7.7803
logsig	purelin	purelin	purelin	0.0097	112.5346	6.6419	0.0115	75.2174	6.7278
tansig	purelin	purelin	purelin	0.0019	187.0739	9.1187	0.0486	57.8641	8.5291
logsig	logsig	purelin	purelin	0.0072	79.1883	5.7745	0.0753	51.605	5.8149
logsig	tansig	purelin	purelin	0.0081	139.5267	7.1496	0.0803	60.1137	7.7751
tansig	tansig	purelin	purelin	6.21E-04	106.1338	5.476	0.0385	55.9545	5.9386
tansig	logsig	purelin	purelin	0.0018	118.9019	5.511	0.0396	52.6915	6.1622
logsig	logsig	logsig	logsig	7.88E-04	91.6869	5.8938	0.0182	58.4797	6.0113
purelin	logsig	logsig	logsig	0.0024	123.457	7.4693	0.0784	51.4983	8.4646
tansig	logsig	logsig	logsig	0.0026	109.8053	5.7624	0.0167	52.5705	7.0859
purelin	purelin	logsig	logsig	0.0038	147.6584	8.4858	0.079	50.4953	7.4478



ตารางที่ 5-8 แสดงผลการทำนายโหนดของโครงข่ายประสาทเทียม 3 ชั้น โดยใช้โหนดชั้นซ่อนแรก 8 โหนด, ชั้นซ่อนที่สอง 3 โหนด และชั้นเอาต์พุต 24 โหนด

รูปแบบฟังก์ชันกระตุ้นในแต่ละชั้น		เวลาที่ใช้		ค่า ape จากการเรียนรู้ (%)				ค่า ape จากการทดสอบ (%)				
ชั้นซ่อนแรก	ชั้นซ่อนที่ 2	ชั้นเอาต์พุต	เรียนรู้ (นาที)	ต่ำสุด	สูงสุด	เฉลี่ย	ต่ำสุด	สูงสุด	เฉลี่ย	ต่ำสุด	สูงสุด	เฉลี่ย
purelin	purelin	purelin	3.367	0.003	131.491	8.8252	0.0435	54.112	8.4943			
logsig	purelin	purelin	3.7867	0.004	142.9874	8.1254	0.0211	59.6374	8.4594			
tansig	purelin	purelin	3.4568	0.0074	131.1133	6.8352	0.03426	51.2341	7.6544			
logsig	logsig	purelin	3.2547	0.0077	145.9654	7.8477	0.0477	86.7621	7.9854			
logsig	tansig	purelin	2.9867	0.0023	108.6357	5.2314	0.0086	54.3967	5.6986			
tansig	tansig	purelin	2.8287	0.0016	117.3328	5.7904	0.0071	55.7726	5.9487			
tansig	logsig	purelin	3.6587	0.0032	125.3487	5.63247	0.0074	59.3548	5.8461			
logsig	logsig	logsig	3.6665	0.0089	136.5987	6.3598	0.0321	86.3547	8.5478			
purelin	logsig	logsig	3.2697	0.0054	117.7531	5.96385	0.0098	56.3254	6.3548			
tansig	logsig	logsig	3.6951	0.0021	142.3514	5.8964	0.0085	75.3247	5.6987			
purelin	purelin	logsig	3.6524	0.0047	126.3214	5.8634	0.0074	63.2458	6.3298			

ตารางที่ 5-9(ต่อ) แสดงผลการทำมาบโหนดของโครงข่ายประสาทเทียม 3 ชั้น โดยใช้โหนดชั้นซ่อนแรก 8 โหนด, ชั้นซ่อนที่สอง 3 โหนด และชั้นเอาต์พุท 24 โหนด

รูปแบบฟังก์ชันกระตุ้นในแต่ละชั้น		เวลาที่ใช้เรียนรู้ (นาที)		ค่า ape จากการเรียนรู้ (%)			ค่า ape จากการทดสอบ (%)		
ชั้นซ่อนแรก	ชั้นซ่อนที่สอง	ชั้นเอาต์พุท	เรียนรู้ (นาที)	ต่ำสุด	สูงสุด	เฉลี่ย	ต่ำสุด	สูงสุด	เฉลี่ย
purelin	tansig	logsig	3.2458	0.0037	96.5412	6.3241	0.0067	66.6744	6.8967
tansig	tansig	logsig	4.222	0.0022	88.039	5.2263	0.0051	58.6117	5.8615
tansig	purelin	logsig	3.6587	0.0034	74.635	5.4327	0.0047	66.354	5.6414
tansig	tansig	tansig	4.1144	0.0027	68.357	5.6328	0.0045	79.6345	5.8698
purelin	tansig	tansig	3.2698	0.0046	96.5247	5.9634	0.0052	92.3546	6.1254
logsig	tansig	tansig	4.5634	0.0026	86.3254	5.4328	0.0063	77.6344	5.8566
purelin	purelin	tansig	2.5687	0.0032	124.3565	6.3985	0.0058	132.5687	8.7416
purelin	logsig	tansig	3.4866	0.0049	56.3587	6.8574	0.0049	55.6358	6.3254
logsig	logsig	tansig	4.1156	0.0041	52.3698	5.6387	0.0032	59.3654	5.4326
logsig	purelin	tansig	3.6542	0.0038	57.7779	5.2746	0.0028	56.3247	5.6328

ตารางที่ 5-10 แสดงผลการทำนายโหนดของโครงข่ายประสาทเทียม 3 ชั้น โดยใช้โหนดชั้นซ่อนแรก 15 โหนด, ชั้นซ่อนที่สอง 8 โหนด และชั้นเอาต์พุต 24 โหนด

รูปแบบฟังก์ชันกระตุ้นในแต่ละชั้น		เวลาที่ใช้ เรียนรู้ (นาที)	ค่า ape จากการเรียนรู้ (%)			ค่า ape จากการทดสอบ (%)			
ชั้นซ่อนแรก	ชั้นซ่อนที่สอง		ชั้นเอาต์พุต	ต่ำสุด	เฉลี่ย	สูงสุด	ต่ำสุด	เฉลี่ย	สูงสุด
purelin	purelin	14.1456	0.0076	474.3254	10.5846	0.0397	128.8688	12.3546	
logsig	purelin	12.5684	8.64E-04	87.6532	5.6941	0.0076	36.5844	5.3242	
tansig	purelin	15.6847	0.0091	686.3547	17.6893	0.0476	253.4431	24.5624	
logsig	logsig	10.5824	9.35E-04	84.5632	6.5324	0.0057	64.3254	5.4632	
logsig	tansig	9.6587	0.0063	287.3654	7.6532	0.0357	117.7771	8.6547	
tansig	tansig	5.9147	8.65E-04	56.3574	6.9414	0.0389	48.9028	5.8911	
tansig	logsig	12.3254	0.0054	458.354	8.7958	0.0477	96.3254	6.3547	
logsig	logsig	15.3654	8.72E-04	86.3254	5.3247	0.0037	52.3256	6.3521	
purelin	logsig	10.2354	9.63E-04	87.6354	6.3254	0.0067	65.5566	5.6347	
tansig	logsig	17.8688	0.0064	365.2458	8.6354	0.325	68.3547	7.6686	
purelin	purelin	13.2644	0.0087	456.351	7.6354	0.0765	121.3547	6.3552	

ตารางที่ 5-11(ต่อ) แสดงผลการทำนายโหลดของโครงข่ายประสาทเทียม 3 ชั้น โดยใช้โหนดชั้นซ่อนแรก 15 โหนด, ชั้นซ่อนที่สอง 8 โหนด และชั้นเอาต์พุท 24 โหนด

รูปแบบฟังก์ชันกระตุ้นในแต่ละชั้น		เวลาที่ใช้ เรียนรู้ (นาที)	ค่า ape จากการเรียนรู้ (%)			ค่า ape จากการทดสอบ (%)				
ชั้นซ่อนแรก	ชั้นซ่อนที่สอง		ชั้นเอาต์พุท	ต่ำสุด	เฉลี่ย	สูงสุด	ต่ำสุด	เฉลี่ย	สูงสุด	
purelin	tansig	logsig	logsig	8.5461	0.0041	252.3674	7.6342	0.0243	98.6354	8.9654
tansig	tansig	logsig	logsig	9.6354	8.74E-04	58.7966	6.5443	0.0067	28.7614	5.3417
tansig	purelin	logsig	logsig	11.0638	9.21E-04	65.7762	4.0632	0.0078	20.2063	4.4355
tansig	tansig	tansig	tansig	12.6534	6.14E-04	54.3574	5.6897	0.0046	28.6144	5.8614
purelin	tansig	tansig	tansig	8.6354	0.0074	357.8664	7.6354	0.0088	58.9654	6.4474
logsig	tansig	tansig	tansig	8.5977	0.0221	748.8281	24.9202	0.0267	294.6828	25.485
purelin	purelin	tansig	tansig	13.6547	0.0324	565.3244	15.3844	0.0484	255.5566	17.6444
purelin	logsig	tansig	tansig	12.3473	0.0051	438.75	9.437	0.103	59.786	6.5101
logsig	logsig	tansig	tansig	10.5247	9.65E-04	87.9644	6.3547	0.0045	60.3604	5.6493
logsig	purelin	tansig	tansig	9.6345	0.0068	277.2774	7.3241	0.0374	112.3511	8.7898

ตารางที่ 5-12 แสดงผลการทำนายโหนดของโครงข่ายประสาทเทียม 3 ชั้น โดยใช้โหนดชั้นซ่อนแรก 32 โหนด, ชั้นซ่อนที่สอง 28 โหนด และชั้นเอาต์พุต 24 โหนด

รูปแบบฟังก์ชันกระตุ้นในแต่ละชั้น		เวลาที่ใช้ เรียนรู้ (นาที)	ค่า ape จากการเรียนรู้ (%)				ค่า ape จากการทดสอบ (%)			
ชั้นซ่อนแรก	ชั้นซ่อนที่สอง		ชั้นเอาต์พุต	ต่ำสุด	สูงสุด	เฉลี่ย	ต่ำสุด	สูงสุด	เฉลี่ย	
purelin	purelin	purelin	purelin	0.0098	56.3254	6.3144	0.0786	98.6354	8.7452	
logsig	purelin	purelin	purelin	0.0058	53.2147	5.6321	0.0474	76.3214	6.8414	
tansig	purelin	purelin	purelin	0.0019	59.6354	5.9632	0.0635	68.9657	6.3254	
logsig	logsig	purelin	purelin	0.0021	63.6354	6.1247	0.0646	98.3254	6.5877	
logsig	tansig	purelin	purelin	0.0032	52.3214	5.6324	0.0487	78.3254	6.8742	
tansig	tansig	purelin	purelin	0.0025	44.6844	3.1457	0.0316	52.2582	5.1472	
tansig	logsig	purelin	purelin	0.0027	47.8448	3.1563	0.0423	56.3254	5.3247	
logsig	logsig	logsig	logsig	0.0047	58.6985	5.6384	0.0784	86.3254	6.3524	
purelin	logsig	logsig	logsig	0.0028	84.3254	6.3254	0.0484	121.3547	6.8766	
tansig	logsig	logsig	logsig	0.0014	31.2547	3.6554	0.0211	48.6987	5.1257	
purelin	purelin	logsig	logsig	0.0035	57.6954	5.6801	0.0291	61.1684	6.8911	

ตารางที่ 5-13(ต่อ) แสดงผลการทำนายโหนดของโครงข่ายประสาทเทียม 3 ชั้น โดยใช้โหนดชั้นซ่อนแรก 32 โหนด, ชั้นซ่อนที่สอง 28 โหนด และชั้นเอาต์พุท 24 โหนด

รูปแบบฟังก์ชันกระตุ้นในแต่ละชั้น		เวลาที่ใช้เรียนรู้ (นาที)			ค่า ape จากการเรียนรู้ (%)			ค่า ape จากการทดสอบ (%)			
ชั้นซ่อนแรก	ชั้นซ่อนที่สอง	ชั้นเอาต์พุท	ต่ำสุด	สูงสุด	เฉลี่ย	ต่ำสุด	สูงสุด	เฉลี่ย	ต่ำสุด	สูงสุด	เฉลี่ย
purelin	tansig	logsig	0.0032	68.6547	3.147	0.0325	87.3547	5.6321			
tansig	tansig	logsig	0.0007	63.0796	2.0192	0.0003	29.0707	2.3434			
tansig	purelin	logsig	0.0028	52.3654	5.6354	0.0422	55.3331	6.3214			
tansig	tansig	tansig	0.0025	11.2241	1.9584	0.0328	25.5567	3.6587			
purelin	tansig	tansig	0.0024	59.3658	3.6547	0.0326	62.3214	5.3698			
logsig	tansig	tansig	0.0034	48.7447	5.2365	0.0424	51.2356	5.6352			
purelin	purelin	tansig	0.0058	52.6354	6.3547	0.0397	98.6358	8.4147			
purelin	logsig	tansig	0.0031	46.3254	5.3265	0.0563	52.6354	6.3254			
logsig	logsig	tansig	0.0051	37.6354	3.2654	0.0652	68.63254	5.3525			
logsig	purelin	tansig	0.0039	55.6553	5.6325	0.0352	77.6325	5.7866			

จากตารางที่ 5-1 ถึง 5-13 แสดงผลการเรียนรู้ของโครงข่ายประสาทเทียม โดยทำการวัดค่าความผิดพลาดสัมบูรณ์ (Absolute percentage error) จากการทดสอบหาค่าเอาต์พุตที่ได้จากโครงข่ายประสาทเทียมแล้วนำไปเปรียบเทียบกับเอาต์พุตจริง เพื่อหาค่าผิดพลาดและนำไปใช้ในการเลือกโครงสร้างที่เหมาะสมกับการทำนายโหนดของการทดลองนี้ การทดลองนี้ได้ปรับเปลี่ยนโครงสร้างที่มีความแตกต่างกันตั้งแต่โครงสร้างเล็กไปสู่โครงสร้างใหญ่ แล้วทำการบันทึกผลการทดลอง โดยตารางที่ 5-1 ถึง 5-13 จะนำผลการทดลองมาแสดงเพียงบางส่วน จากตารางพบว่ารูปแบบโครงสร้างที่เหมาะสมกับการทำนายโหนดในการทดลองนี้ คือ โครงสร้างที่ให้ค่า ape น้อยจากการทดสอบมีค่าต่ำสุด ซึ่งได้แก่รูปแบบโครงสร้างในตารางที่ 5-13(ต่อ) จะมีค่า ape น้อยของการทดสอบต่ำสุด 2.3434 %

จากการทดลองในตารางที่ 5-1 ถึง 5-13 สามารถหารูปแบบโครงสร้างที่เหมาะสมในการนำไปใช้ทำนายโหนดได้ ตารางที่ 5-14 ถึง 5-21 จะเป็นผลการทำนายโหนดรายชั่วโมงในแต่ละวัน โดยใช้รูปแบบโครงสร้างของโครงข่ายประสาทเทียมที่ได้จากผลการทดลองที่ผ่านมา คือ โครงข่ายประสาทเทียมที่มี 3 ชั้น โดยใช้โหนดชั้นซ่อนแรก 32 โหนด, ชั้นซ่อนที่สอง 28 โหนด และชั้นเอาต์พุต 24 โหนด ซึ่งค่า Transfer function ในชั้นซ่อนแรกเป็น tansig, ชั้นซ่อนที่สองเป็น tansig และชั้นเอาต์พุตเป็น logsig

จากตารางที่ 5-14 ถึง 5-21 เราสามารถนำผลการทำนายโหนดมาแสดงในลักษณะกราฟรายชั่วโมงต่อเนื่องกันในแต่ละวันได้ โดยรูปที่ 5-1 ถึง 5-8 จะแสดงกราฟรายชั่วโมงของการทดสอบการทำนายโหนดในแต่ละวัน ทั้งหมด 8 วัน ซึ่งสามารถสังเกตเห็นความแตกต่างระหว่างค่าโหนดจริงกับค่าโหนดจากการทำนายได้อย่างชัดเจน และจากตารางผลทดสอบโครงข่ายประสาทเทียมทั้งหมด 8 วัน เราสามารถแสดงกราฟรายชั่วโมงระหว่างค่าโหนดจริงเปรียบเทียบกับค่าโหนดจากการทำนายทั้งหมด ตั้งแต่การเรียนรู้จนถึงการทดสอบโครงข่ายประสาทเทียม คิดเป็นเวลาทั้งหมด 92 วัน โดยในช่วง 84 วันแรกจะเป็นการผลการทำนายโหนดจากการเรียนรู้ และ 8 วันสุดท้ายจะใช้ทดสอบการทำนายโหนดของโครงข่ายประสาทเทียมหลังจากการเรียนรู้เสร็จสิ้น ซึ่งแสดงได้ดังรูปที่ 5-9

ตารางที่ 5-14 แสดงผลการทำนายโหลดในวันที่ 1 ของชุดข้อมูลทดสอบ โดยมีค่า ape เฉลี่ย 4.7581 %  
และมีค่า ape สูงสุด 12.1625%

เวลา	โหลดจริง (MW)	โหลดทำนาย (MW)	Error (MW)	Absolute percentage error (%)
01.00	21.9000	21.4698	0.4302	1.9645
02.00	23.3000	22.7783	0.5217	2.2391
03.00	22.6000	21.1802	1.4198	6.2825
04.00	22.3000	22.2646	0.0354	0.1588
05.00	21.6000	21.8948	-0.2948	1.3649
06.00	22.6000	22.2060	0.3940	1.7435
07.00	22.5000	21.9877	0.5123	2.2769
08.00	29.8000	26.5030	3.2970	11.0638
09.00	39.1000	38.2887	0.8113	2.0750
10.00	41.2000	38.7062	2.4938	6.0529
11.00	38.9000	38.7073	0.1927	0.4955
12.00	31.6000	29.9309	1.6691	5.2821
13.00	37.4000	37.2332	0.1668	0.4461
14.00	41.7000	40.1137	1.5863	3.8040
15.00	41.0000	39.4300	1.5700	3.8292
16.00	41.2000	40.6034	0.5966	1.4479
17.00	31.8000	29.8990	1.9010	5.9778
18.00	29.5000	27.8203	1.6797	5.6938
19.00	29.4000	28.5600	0.8400	2.8572
20.00	29.0000	27.3658	1.6342	5.6353
21.00	24.0000	23.9985	0.0015	0.0064
22.00	26.6000	25.0469	1.5531	5.8387
23.00	24.4000	23.2835	1.1165	4.5758
24.00	25.3000	23.0019	2.2981	9.0832

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 5-15 แสดงผลการทำนายโหลดในวันที่ 2 ของชุดข้อมูลทดสอบ โดยมีค่า ape เฉลี่ย 0.9237% และค่า ape สูงสุด 3.9197%

เวลา	โหลดจริง (MW)	โหลดทำนาย (MW)	Error (MW)	Absolute percentage error (%)
01.00	25.4000	24.9841	0.4159	1.6375
02.00	25.2000	24.4300	0.7700	3.0557
03.00	22.4000	22.2890	0.1110	0.4954
04.00	23.1000	23.3422	-0.2422	1.0485
05.00	25.6000	24.7484	0.8516	3.3267
06.00	24.5000	24.8245	-0.3245	1.3246
07.00	23.9000	23.7807	0.1193	0.4993
08.00	25.2000	25.8290	-0.6290	2.4960
09.00	37.7000	37.5824	0.1176	0.3120
10.00	40.6000	40.1784	0.4216	1.0384
11.00	40.7000	41.0641	-0.3641	0.8945
12.00	32.7000	32.3912	0.3088	0.9443
13.00	37.9000	37.2721	0.6279	1.6566
14.00	41.0000	40.5705	0.4295	1.0476
15.00	38.7000	37.6675	1.0325	2.6679
16.00	37.4000	37.2931	0.1069	0.2858
17.00	30.0000	29.6587	0.3413	1.1376
18.00	29.8000	29.2832	0.5168	1.7341
19.00	26.5000	26.2114	0.2886	1.0889
20.00	24.6000	25.0697	-0.4697	1.9095
21.00	22.6000	22.2672	0.3328	1.4727
22.00	25.1000	24.7744	0.3256	1.2973
23.00	23.5000	23.2084	0.2916	1.2407
24.00	21.5000	21.3164	0.1836	0.8538

ตารางที่ 5-16 แสดงผลการทำนายโหลดในวันที่ 3 ของชุดข้อมูลทดสอบ โดยมีค่า ape เฉลี่ย 1.4716% และค่า ape สูงสุด 3.8036%

เวลา	โหลดจริง (MW)	โหลดทำนาย (MW)	Error (MW)	Absolute percentage error (%)
01.00	21.5000	21.3300	0.1700	0.7906
02.00	21.5000	20.8207	0.6793	3.1597
03.00	19.5000	19.5192	-0.0192	0.0985
04.00	19.5000	19.2378	0.2622	1.3444
05.00	19.5000	19.0944	0.4056	2.0802
06.00	18.5000	18.3523	0.1477	0.7985
07.00	18.5000	18.5271	-0.0271	0.1466
08.00	21.8000	22.0223	-0.2223	1.0198
09.00	32.5000	31.7872	0.7128	2.1933
10.00	30.0000	30.6381	-0.6381	2.1270
11.00	31.9000	32.2732	-0.3732	1.1698
12.00	25.0000	24.9986	0.0014	0.0058
13.00	31.0000	30.9722	0.0278	0.0897
14.00	32.3000	31.9645	0.3355	1.0388
15.00	31.7000	31.3142	0.3858	1.2169
16.00	30.9000	31.3193	-0.4193	1.3569
17.00	22.7000	22.8895	-0.1895	0.8346
18.00	23.1000	23.3515	-0.2515	1.0885
19.00	23.3000	23.3332	-0.0332	0.1424
20.00	21.5000	21.4982	0.0018	0.0085
21.00	19.5000	19.8563	-0.3563	1.8270
22.00	20.7000	20.5315	0.1685	0.8140
23.00	20.7000	20.4214	0.2786	1.3458
24.00	18.8000	19.2572	-0.4572	2.4320

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 5-17 แสดงผลการทำนายโหลดในวันที่ 4 ของชุดข้อมูลทดสอบ โดยมีค่า ape เฉลี่ย 3.0789% และมีค่า ape สูงสุด 10.5975%

เวลา	โหลดจริง (MW)	โหลดทำนาย (MW)	Error (MW)	Absolute percentage error (%)
01.00	18.2000	18.0334	0.1666	0.9154
02.00	18.2000	18.3012	-0.1012	0.5558
03.00	16.9000	17.4715	-0.5715	3.3817
04.00	16.9000	16.8351	0.0649	0.3837
05.00	16.9000	17.2133	-0.3133	1.8536
06.00	15.7000	16.2942	-0.5942	3.7844
07.00	15.7000	15.9323	-0.2323	1.4793
08.00	14.7000	15.2292	-0.5292	3.5999
09.00	18.8000	19.5420	-0.7420	3.9467
10.00	18.5000	18.8154	-0.3154	1.7051
11.00	19.5000	20.1570	-0.6570	3.3694
12.00	16.2000	16.0076	0.1924	1.1878
13.00	16.3000	16.7588	-0.4588	2.8148
14.00	18.2000	19.2379	-1.0379	5.7028
15.00	18.5000	18.5835	-0.0835	0.4512
16.00	19.2000	18.9814	0.2186	1.1387
17.00	14.9000	14.7441	0.1559	1.0463
18.00	14.7000	14.8786	-0.1786	1.2148
19.00	13.2000	13.5897	-0.3897	2.9519
20.00	12.5000	12.6375	-0.1375	1.1001
21.00	12.1000	12.7941	-0.6941	5.7366
22.00	13.6000	13.4650	0.1350	0.9926
23.00	12.0000	13.3197	-1.3197	10.9974
24.00	12.7000	13.4658	-0.7658	6.0300

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 5-18 แสดงผลการทำนายโหลดในวันที่ 5 ของชุดข้อมูลทดสอบ โดยมีค่า ape เฉลี่ย 2.7220% และมีค่า ape สูงสุด 8.4155%

เวลา	โหลดจริง (MW)	โหลดทำนาย (MW)	Error (MW)	Absolute percentage error (%)
01.00	12.5000	12.4610	0.0390	0.3117
02.00	12.5000	12.1422	0.3578	2.8623
03.00	12.5000	12.4819	0.0181	0.1448
04.00	12.5000	12.4809	0.0191	0.1524
05.00	12.6000	12.0547	0.5453	4.3278
06.00	13.4000	13.0964	0.3036	2.2657
07.00	15.6000	15.7299	-0.1299	0.8326
08.00	25.3000	25.2304	0.0696	0.2753
09.00	35.9000	35.7866	0.1134	0.3159
10.00	36.1000	34.9408	1.1592	3.2110
11.00	36.5000	36.7088	-0.2088	0.5721
12.00	30.5000	30.5417	-0.0417	0.1367
13.00	37.4000	37.1758	0.2242	0.5994
14.00	37.2000	37.1662	0.0338	0.0909
15.00	39.1000	38.4387	0.6613	1.6912
16.00	27.5000	28.2323	-0.7323	2.6628
17.00	34.1000	33.1958	0.9042	2.6517
18.00	29.7000	30.3743	-0.6743	2.2704
19.00	28.7000	27.4964	1.2036	4.1938
20.00	24.7000	24.8978	-0.1978	0.8009
21.00	23.4000	23.1101	0.2899	1.2391
22.00	25.7000	26.0705	-0.3705	1.4418
23.00	27.5000	27.2952	0.2048	0.7447
24.00	27.5000	26.6665	0.8335	3.0310

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 5-19 แสดงผลการทำนายโหลดในวันที่ 6 ของชุดข้อมูลทดสอบ โดยมีค่า ape เฉลี่ย 0.6875%  
และมีค่า ape สูงสุด 1.6363%

เวลา	โหลดจริง (MW)	โหลดทำนาย (MW)	Error (MW)	Absolute percentage error (%)
01.00	21.4000	21.2344	0.1656	0.7738
02.00	22.6000	22.4494	0.1506	0.6665
03.00	20.7000	20.5398	0.1602	0.7739
04.00	18.6000	18.7124	-0.1124	0.6040
05.00	22.2000	22.3314	-0.1314	0.5920
06.00	21.5000	21.4240	0.0760	0.3537
07.00	21.0000	20.7639	0.2361	1.1241
08.00	29.9000	30.8954	-0.9954	3.3291
09.00	40.2000	40.6070	-0.4070	1.0125
10.00	38.6000	38.8986	-0.2986	0.7735
11.00	41.5000	41.6570	-0.1570	0.3784
12.00	30.5000	30.6610	-0.1610	0.5278
13.00	35.7000	34.2743	1.4257	3.9937
14.00	39.8000	39.4594	0.3406	0.8557
15.00	37.1000	37.9326	-0.8326	2.2441
16.00	29.4000	29.5356	-0.1356	0.4611
17.00	31.8000	31.2982	0.5018	1.5779
18.00	27.8000	27.6866	0.1134	0.4079
19.00	29.2000	29.1427	0.0573	0.1963
20.00	23.8000	24.4677	-0.6677	2.8053
21.00	23.2000	23.0550	0.1450	0.6250
22.00	24.9000	24.6198	0.2802	1.1252
23.00	23.0000	23.7980	-0.7980	3.4696
24.00	33.0000	31.7308	1.2692	3.8461

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 5-20 แสดงผลการทำนายโหลดในวันที่ 7 ของชุดข้อมูลทดสอบ โดยมีค่า ape เฉลี่ย 0.7898% และมีค่า ape สูงสุด 1.8916%

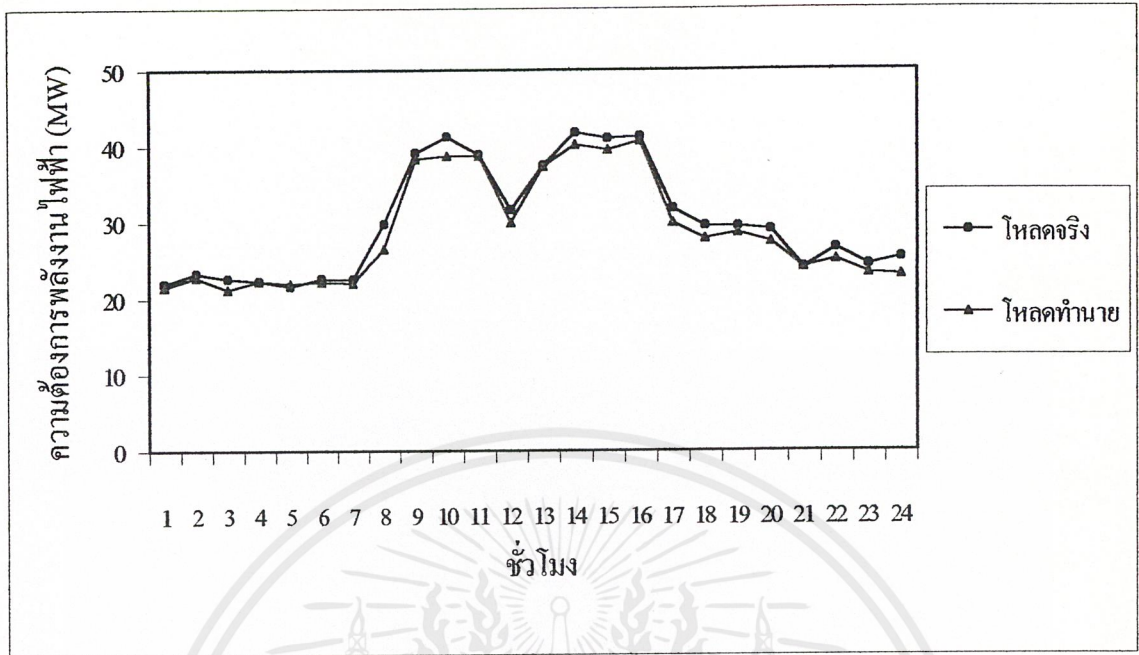
เวลา	โหลดจริง (MW)	โหลดทำนาย (MW)	Error (MW)	Absolute percentage error (%)
01.00	15.0000	15.2632	-0.2632	1.7549
02.00	30.0000	29.0398	0.9602	3.2007
03.00	30.0000	29.9895	0.0105	0.0351
04.00	22.5000	22.7205	-0.2205	0.9799
05.00	22.8000	22.4947	0.3053	1.3391
06.00	22.8000	23.5525	-0.7525	3.3004
07.00	25.0000	24.7828	0.2172	0.8688
08.00	27.4000	27.6186	-0.2186	0.7978
09.00	37.1000	37.0973	0.0027	0.0073
10.00	37.2000	37.3027	-0.1027	0.2762
11.00	39.5000	39.2201	0.2799	0.7087
12.00	32.1000	31.7948	0.3052	0.9508
13.00	34.7000	34.6765	0.0235	0.0677
14.00	38.5000	38.5309	-0.0309	0.0802
15.00	38.7000	38.6301	0.0699	0.1806
16.00	39.5000	39.9827	-0.4827	1.2220
17.00	33.7000	33.8247	-0.1247	0.3701
18.00	27.0000	26.7335	0.2665	0.9870
19.00	28.5000	28.6096	-0.1096	0.3846
20.00	24.0000	23.4841	0.5159	2.1495
21.00	24.7000	24.8669	-0.1669	0.6758
22.00	26.3000	25.8498	0.4502	1.7119
23.00	22.8000	23.1385	-0.3385	1.4848
24.00	22.8000	22.9011	-0.1011	0.4435

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

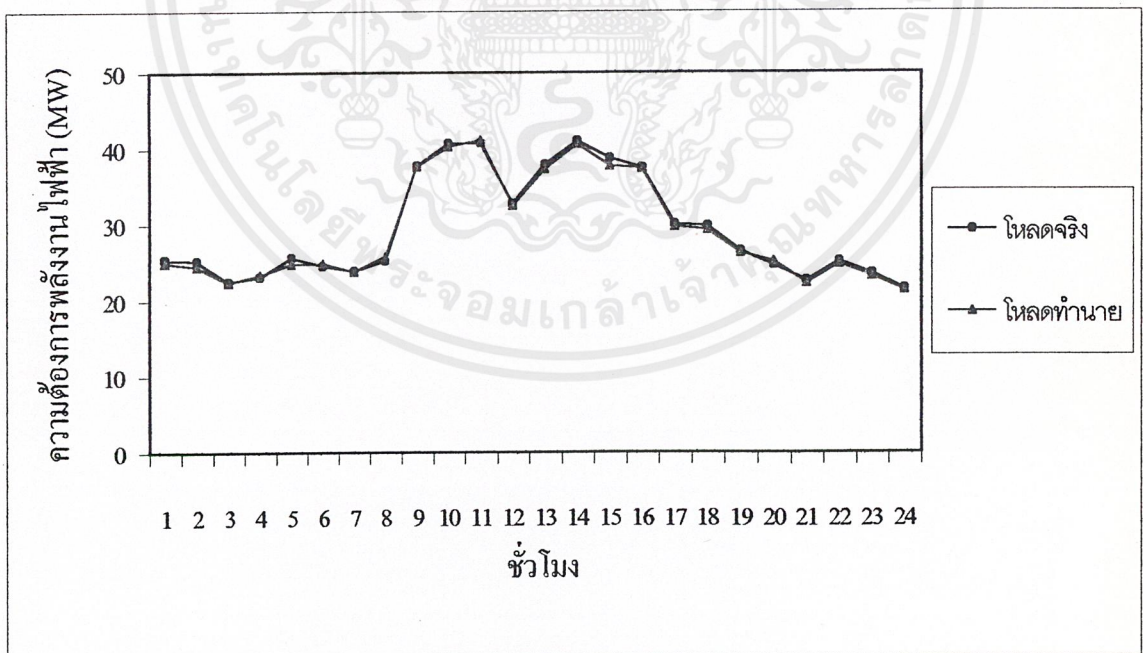
ตารางที่ 5-21 แสดงผลการทำนายโหลดในวันที่ 8 ของชุดข้อมูลทดสอบ โดยค่า ape เฉลี่ย 5.8305%  
และค่า ape สูงสุด 29.3482%

เวลา	โหลดจริง (MW)	โหลดทำนาย (MW)	Error (MW)	Absolute percentage error (%)
01.00	12.3000	12.0758	0.2242	1.8224
02.00	12.9000	14.8615	-1.9615	15.2051
03.00	12.9000	14.1548	-1.2548	9.7275
04.00	21.5000	21.4504	0.0496	0.2306
05.00	21.5000	22.1078	-0.6078	2.8271
06.00	21.5000	22.3100	-0.8100	3.7675
07.00	18.9000	21.4136	-2.5136	13.2995
08.00	28.5000	30.8708	-2.3708	8.3188
09.00	37.7000	38.5907	-0.8907	2.3625
10.00	35.3000	36.1773	-0.8773	2.4852
11.00	37.2000	39.9739	-2.7739	7.4567
12.00	31.6000	32.9320	-1.3320	4.2151
13.00	36.5000	37.1772	-0.6772	1.8552
14.00	38.8000	41.2300	-2.4300	6.2630
15.00	38.8000	40.8993	-2.0993	5.4105
16.00	30.5000	39.4512	-8.9512	29.3482
17.00	30.8000	33.6240	-2.8240	9.1690
18.00	26.1000	27.7279	-1.6279	6.2373
19.00	26.7000	27.5258	-0.8258	3.0930
20.00	26.3000	26.6647	-0.3647	1.3867
21.00	25.4000	24.8272	0.5728	2.2551
22.00	26.7000	27.0589	-0.3589	1.3443
23.00	25.8000	25.8802	-0.0802	0.3109
24.00	25.8000	26.1975	-0.3975	1.5406

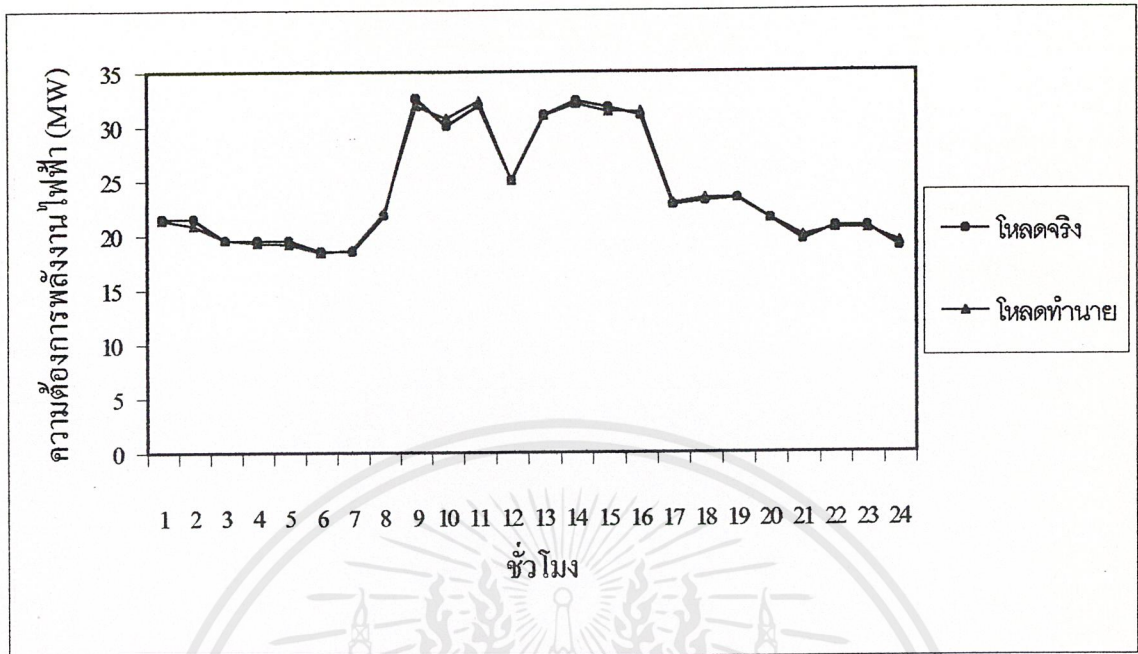
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



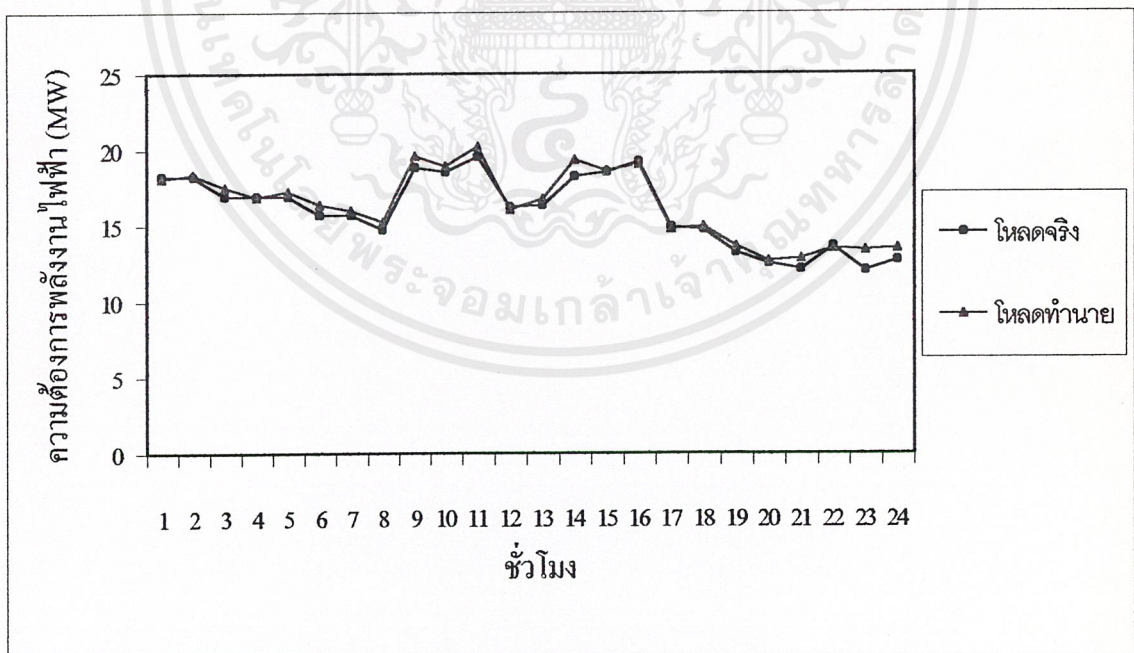
รูปที่ 5-1 แสดงกราฟโหลดรายชั่วโมงของการทดสอบวันที่ 1 ระหว่างโหลดจริงเทียบกับ โหลดจากการทำนายโดยใช้โครงข่ายประสาทเทียม



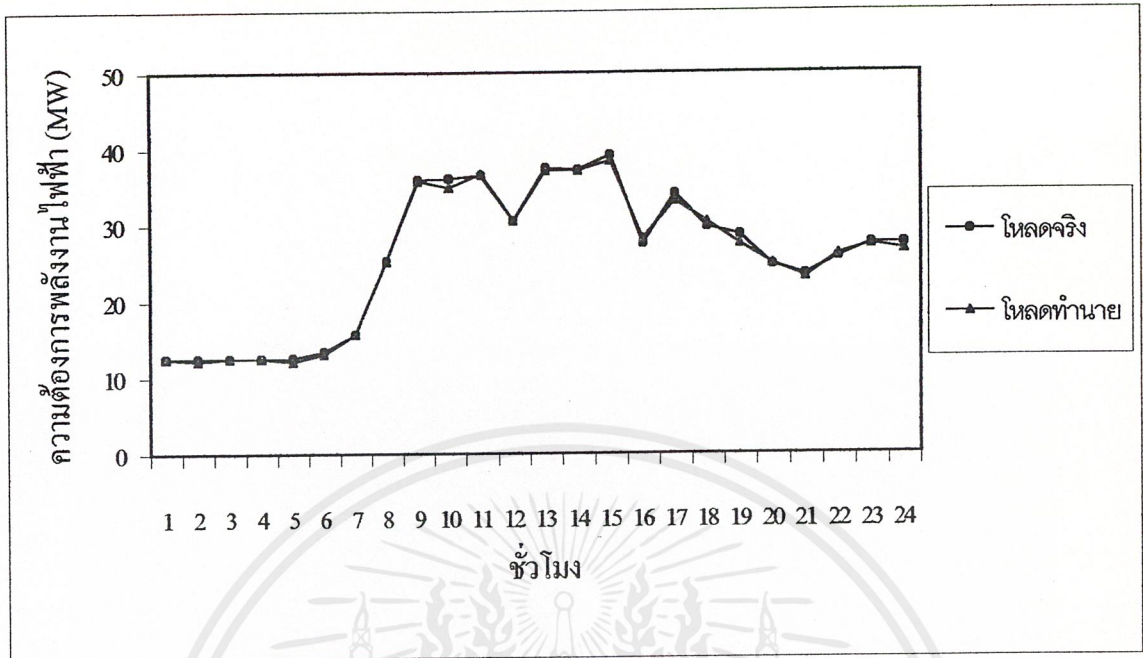
รูปที่ 5-2 แสดงกราฟโหลดรายชั่วโมงของการทดสอบวันที่ 2 ระหว่างโหลดจริงเทียบกับ โหลดจากการทำนายโดยใช้โครงข่ายประสาทเทียม



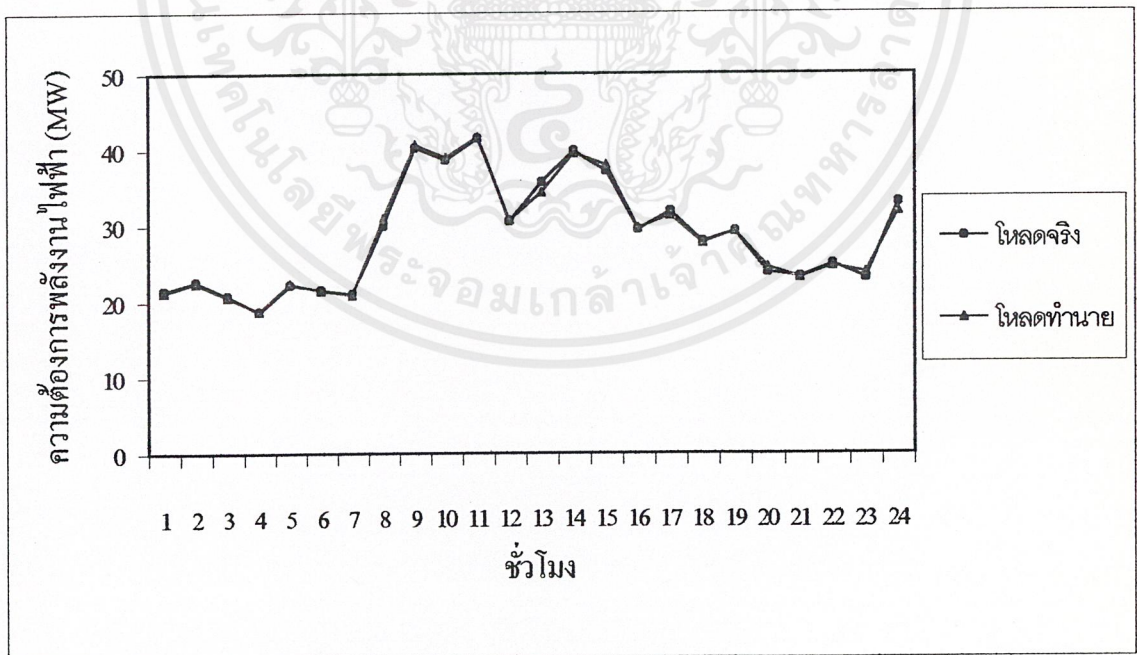
รูปที่ 5-3 แสดงกราฟโหลดรายชั่วโมงของการทดสอบวันที่ 3 ระหว่างโหลดจริงเทียบกับ โหลดจากการทำนายโดยใช้โครงข่ายประสาทเทียม



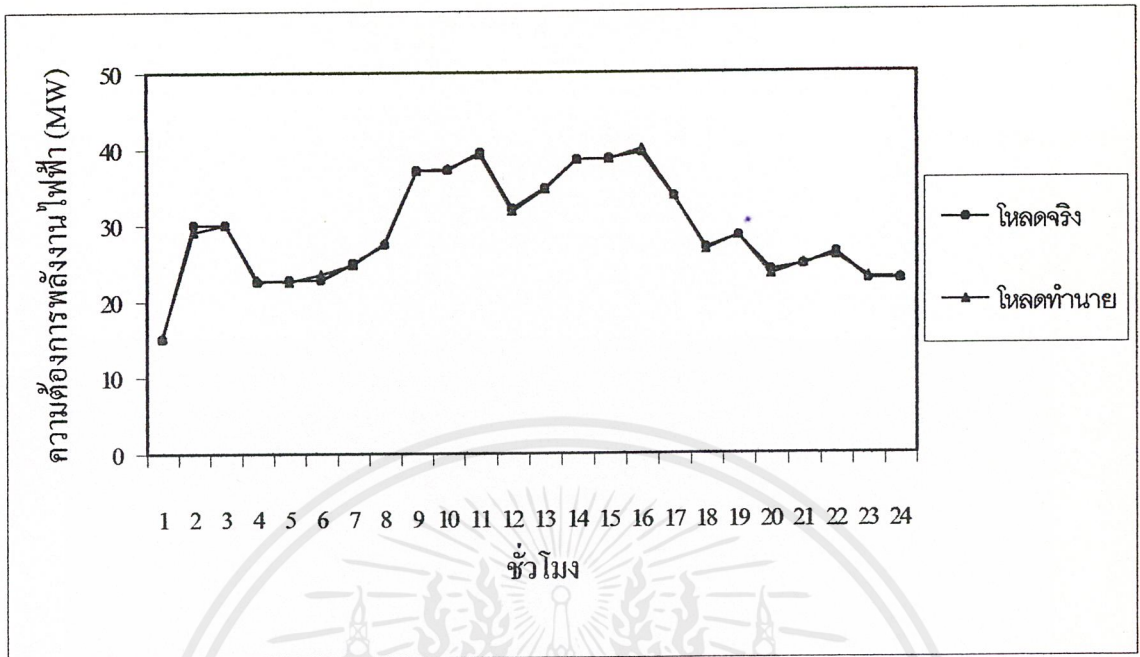
รูปที่ 5-4 แสดงกราฟโหลดรายชั่วโมงของการทดสอบวันที่ 4 ระหว่างโหลดจริงเทียบกับ โหลดจากการทำนายโดยใช้โครงข่ายประสาทเทียม



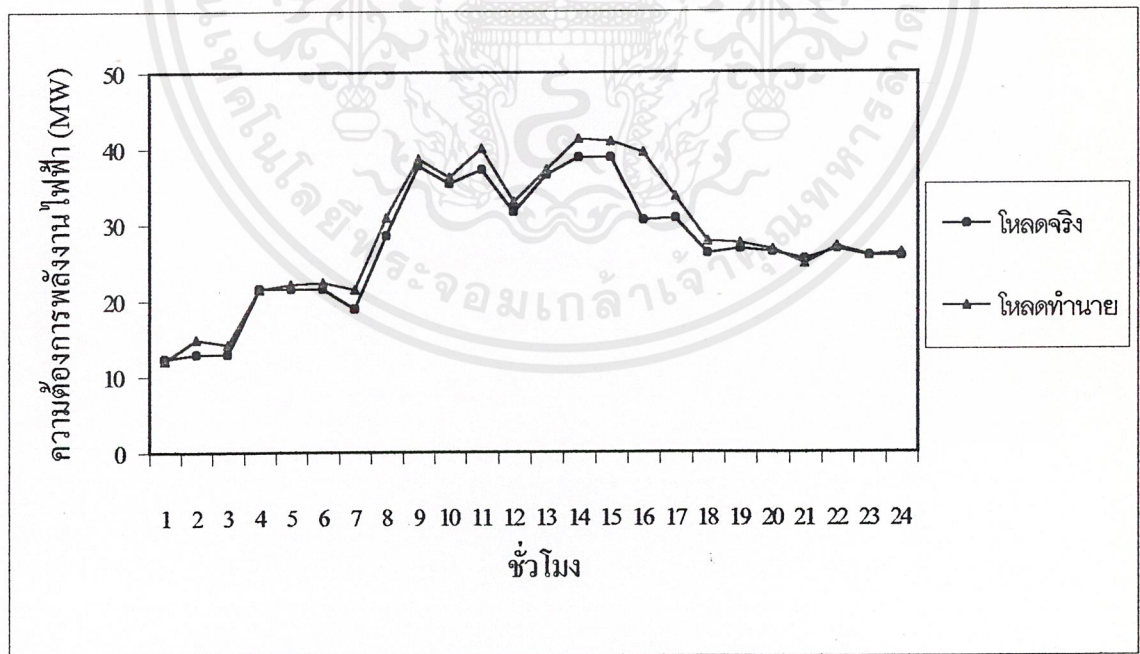
รูปที่ 5-5 แสดงกราฟโหลดรายชั่วโมงของการทดสอบวันที่ 5 ระหว่างโหลดจริงเทียบกับ โหลดจากการทำนายโดยใช้โครงข่ายประสาทเทียม



รูปที่ 5-6 แสดงกราฟโหลดรายชั่วโมงของการทดสอบวันที่ 6 ระหว่างโหลดจริงเทียบกับ โหลดจากการทำนายโดยใช้โครงข่ายประสาทเทียม

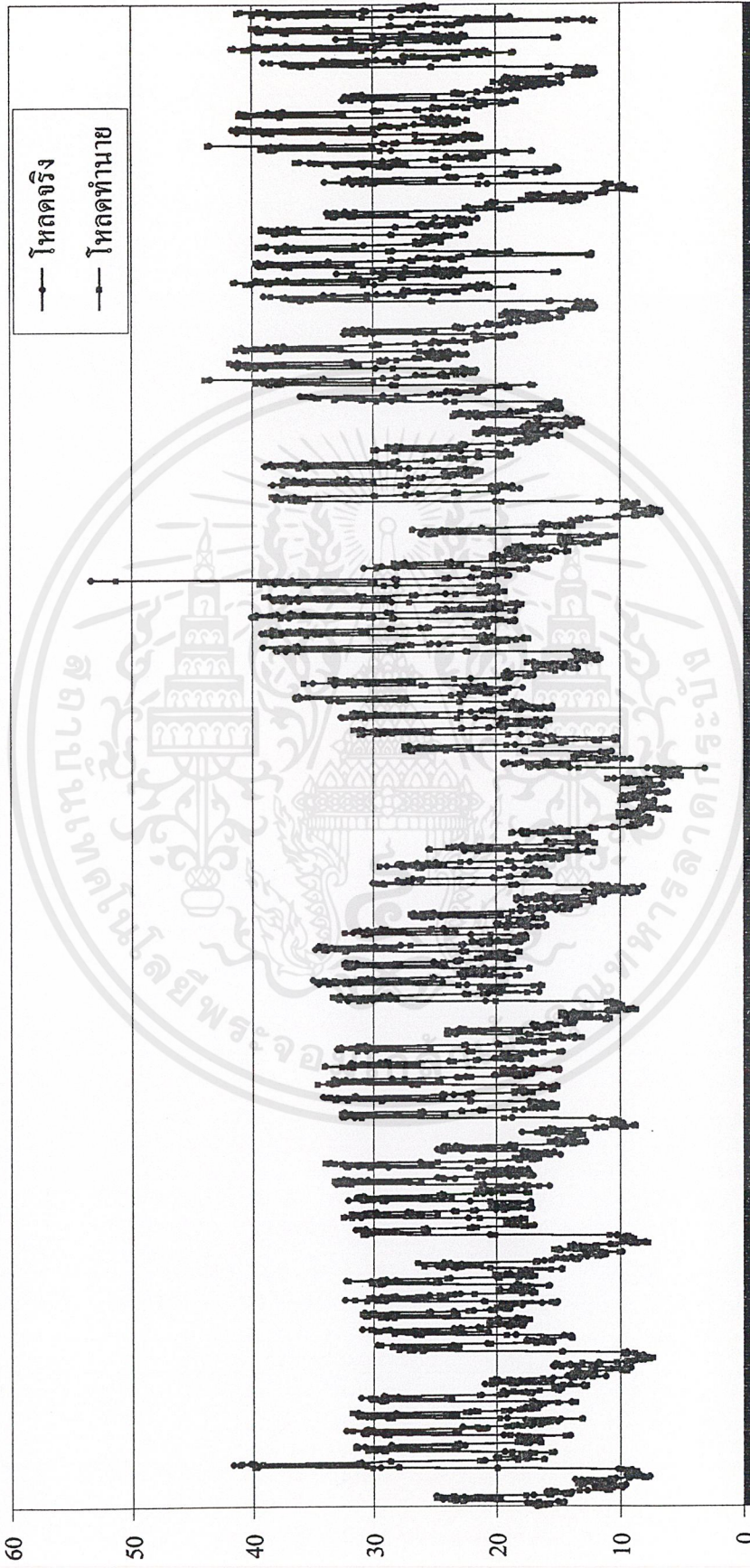


รูปที่ 5-7 แสดงกราฟโหลดรายชั่วโมงของการทดสอบวันที่ 7 ระหว่างโหลดจริงเทียบกับ โหลดจากการทำนายโดยใช้โครงข่ายประสาทเทียม



รูปที่ 5-8 แสดงกราฟโหลดรายชั่วโมงของการทดสอบวันที่ 8 ระหว่างโหลดจริงเทียบกับ โหลดจากการทำนายโดยใช้โครงข่ายประสาทเทียม

รูปที่ 5-9 แสดงกราฟของโหนดจริงและโหนดจากการทำนายโดยโครงข่ายประสาทเทียม ตั้งแต่วันที่ 1 มีนาคม - 31 พฤษภาคม 2544



1 81 161 241 321 401 481 561 641 721 801 881 961 1041 1121 1201 1281 1361 1441 1521 1601 1681 1761 1841 1921 2001 2081

ชั่วโมง

## 5.2 ผลการวิเคราะห์โครงข่ายประสาทดัดเทียม

จากการทดลองการทำนายโหนดด้วยโครงข่ายประสาทดัดเทียมที่โครงสร้างต่าง ๆ กันพบว่า โครงข่ายประสาทดัดเทียมที่ประกอบด้วย 3 ชั้น ซึ่งมีจำนวนโหนดในชั้นซ่อนที่ 1 เป็น 32 โหนด, ชั้นซ่อนที่ 2 เป็น 28 โหนด และชั้นเอาต์พุต 24 โหนด โดยอาศัยการปรับค่าความผิดพลาดที่ยอมรับได้ปรากฏว่า โครงข่ายประสาทดัดเทียมสามารถประมวลผลข้อมูลได้อย่างมีประสิทธิภาพที่สุด ซึ่งพิจารณาจากค่าความผิดพลาดจากการทดสอบมีค่าต่ำสุด คือ 2.3434%

จากผลของการทดลองในครั้งนี้ จะเห็นได้ว่าในการทำนายโหนดด้วยโครงข่ายประสาทดัดเทียมที่มีลักษณะโครงสร้างที่แตกต่างกัน ก็จะทำให้เกิดการเรียนรู้ของโครงข่ายประสาทดัดเทียมที่แตกต่างกัน ซึ่งเราสามารถจำแนกปัจจัยที่มีผลกระทบต่อการทำนายโหนดในการทดลองได้ดังต่อไปนี้ คือ

1) โหนด จากลักษณะของข้อมูลโหนดที่ใช้ในการทดลองนี้ สามารถจำแนกออกได้เป็น 2 ประเภทใหญ่ๆ ด้วยกัน คือ ข้อมูลโหนดที่เกิดขึ้นในขณะที่ทำนายโหนดและข้อมูลโหนดที่เกิดจริงในช่วงเวลาที่ต้องการทำนายโหนดในอดีต ซึ่งจากการทดลองจะเห็นได้ว่าหากข้อมูลโหนดมีการเปลี่ยนแปลงค่าไปผลของการทำนายโหนดโดยโครงข่ายประสาทดัดเทียมก็จะมีแนวโน้มเปลี่ยนแปลงไปในทิศทางเดียวกันกับการเปลี่ยนแปลงของข้อมูลโหนด ทั้งนี้และทั้งนั้น โครงข่ายประสาทดัดเทียมจะไม่ได้รับรู้ถึงความสำคัญมากน้อยของข้อมูลนั้นๆ แต่โครงข่ายประสาทดัดเทียมจะทำงาน โดยอาศัยการหาความสัมพันธ์ของคู่รูปแบบอินพุตและเอาต์พุต ดังนั้นข้อมูลโหนดจึงเป็นคล้ายสิ่งที่แสดงถึงระดับของโหนดอ้างอิงที่สภาวะแวดล้อมนั้นๆ

2) อุณหภูมิ อุณหภูมิเป็นปัจจัยที่สำคัญอีกประการหนึ่งที่ส่งผลกระทบต่ออัตราการใช้กำลังไฟฟ้า โดยผลกระทบที่เกิดจากอุณหภูมินั้นเราสามารถจำแนกออกได้เป็นประเภทดังต่อไปนี้ คือ

- ผลกระทบที่มีต่อระบบการส่งจ่ายกำลังไฟฟ้า กำลังการส่งจ่ายไฟฟ้าผ่านระบบสายส่ง (Transmission line) นั้นจะถูกจำกัดด้วยอุณหภูมิของสายตัวนำหรือที่เรียกกันว่า เทอร์มอลลิมิตของสายตัวนำและอุณหภูมิจะมีผลโดยตรงต่อค่าการสูญเสียที่เกิดขึ้นในระบบสายส่งกำลัง

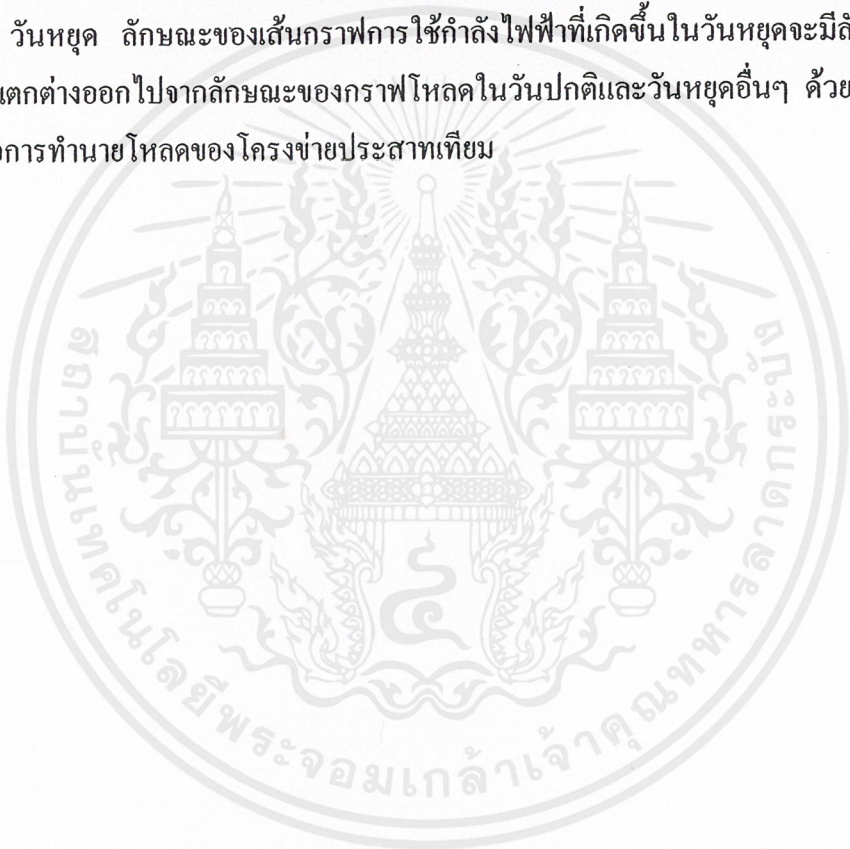
- ผลกระทบที่เกิดขึ้นต่อโหนดทางเศรษฐกิจ อุณหภูมิได้เป็นปัจจัยที่สำคัญต่อการใช้กำลังไฟฟ้าของโหนดประเภทนี้ ทั้งนี้เนื่องจากแหล่งนิคมอุตสาหกรรมล้าแล้วแต่ต้องการอำนวยความสะดวกสบายให้กับโรงงาน ดังจะเห็นได้จากการติดเครื่องปรับอากาศในห้องทำงาน และโดยเฉพาะในฤดูร้อนแล้วจะเห็นได้ว่า การใช้กำลังไฟฟ้าของโหนดประเภทนี้จะมีอัตราเพิ่มขึ้นเป็นอย่างมาก

- ผลกระทบที่เกิดขึ้นต่อโหนดอุตสาหกรรม จะเห็นได้โดยทั่วไปว่า การทำงานของเครื่องจักรต่างๆ ส่วนใหญ่แล้วต้องการการระบายความร้อนทั้งสิ้น โดยเฉพาะเครื่องจักรที่มีขนาดใหญ่หรือมีพิทคาลังสูงด้วยแล้ว การระบายความร้อนเป็นสิ่งที่หลีกเลี่ยงไม่ได้ ไม่ว่าจะเป็นการระบายความร้อนด้วยอากาศหรือการระบายความร้อนด้วยน้ำก็ตาม ทั้งนี้เนื่องจากอุณหภูมิที่สูงเกินไปจะเป็นอันตราย

ต่อการทำงานของเครื่องจักรและทำให้เครื่องจักรนั้นมีอายุการใช้งานสั้นลง ดังนั้น หากสภาวะแวดล้อมโดยรอบมีอุณหภูมิสูงขึ้นแล้วย่อมทำให้ระบบการระบายความร้อนของเครื่องจักรต้องทำงานมากขึ้น เพื่อระบายความร้อนออกจากตัวเครื่องจักรได้ในอัตราเท่าเดิม

3) เวลา จากลักษณะของเส้นกราฟโหลดของวันต่าง ๆ นั้นปรากฏว่า ลักษณะของเส้นกราฟโหลดจะมีการเปลี่ยนแปลงตามค่าเวลาเช่นกัน ดังนั้น ในการเรียนรู้ของโครงข่ายที่มีรูปแบบของอินพุทที่แสดงถึงเวลาจะทำให้โครงข่ายประสาทเทียมสามารถเรียนรู้ได้ถึงลักษณะของเส้นกราฟโหลดที่เกิดขึ้น ณ เวลาต่างๆ ได้ซึ่งสังเกตได้จากผลการทดลองโครงข่ายประสาทเทียมสามารถให้ผลการทำนายที่มีประสิทธิภาพ

4) วันหยุด ลักษณะของเส้นกราฟการใช้กำลังไฟฟ้าที่เกิดขึ้นในวันหยุดจะมีลักษณะเฉพาะตัวและมีความแตกต่างออกไปจากลักษณะของกราฟโหลดในวันปกติและวันหยุดอื่นๆ ด้วยกัน จึงทำให้วันหยุดมีผลต่อการทำนายโหลดของโครงข่ายประสาทเทียม



## บทที่ 6

### บทวิจารณ์และสรุป

เนื่องจากการทำนายการใช้กำลังไฟฟ้านั้นเราไม่สามารถแสดงได้ด้วยสมการทางคณิตศาสตร์ใดๆ ทั้งนี้เนื่องจากโหลดจะมีการเปลี่ยนแปลงอยู่ตลอดเวลาตามสภาวะแวดล้อมในขณะนั้น ไม่ว่าจะเป็นสภาวะอากาศหรือสภาวะทางเศรษฐกิจก็ตาม ดังนั้นการนำโครงข่ายประสาทเทียมเข้ามาประยุกต์ใช้ในการทำนายโหลดจึงเป็นอีกวิธีหนึ่งในการหาความสัมพันธ์ของการเปลี่ยนแปลงของโหลดกับสภาวะแวดล้อมในขณะนั้นๆ โดยอาศัยคุณลักษณะสำคัญของโครงข่ายประสาทเทียมที่มีความสามารถในการประมวลผลข้อมูลที่มีความซับซ้อน

จากผลการการทำนายการใช้กำลังไฟฟ้า ได้ผลการทำนายโหลดเป็นที่น่าพอใจ ทั้งนี้เนื่องจาก ค่าความผิดพลาดเฉลี่ยที่เกิดขึ้นจากการทดสอบมีค่าประมาณ 2 % และจากขอบเขตของการทำโครงการในครั้งนี้ ได้ใช้ปัจจัยจากสภาวะอากาศเป็นหลักในการทำนายโหลด คืออุณหภูมิต่ำสุดและสูงสุดภายในจังหวัดที่สถานีไฟฟ้าย่อยตั้งอยู่ โดยไม่คำนึงถึงผลจากสภาวะอื่น เช่น สภาวะทางเศรษฐกิจ ดังนั้นอาจเป็นสาเหตุที่ทำให้การทำนายโหลดในโครงการครั้งนี้ได้ผลที่มีค่าความผิดพลาดอยู่บ้าง

จากลักษณะของข้อมูลที่ใช้ในการเรียนรู้ของโครงข่ายประสาทเทียมและการทำนายกำลังไฟฟ้า จะเห็นได้ว่า เราต้องใช้ข้อมูลจำนวนมากสำหรับการเรียนรู้และการทดสอบโครงข่ายประสาทเทียม ในการทดลองนี้ได้ลดความยุ่งยากและลดข้อผิดพลาดในการจัดเตรียมชุดข้อมูลที่ใช้กับโครงข่ายประสาทเทียมโดยการนำข้อมูลทั้งหมดเก็บไว้ในฐานข้อมูล โดยสร้างระบบฐานข้อมูลในโปรแกรมไมโครซอฟต์ แอกเซส เพื่อจัดการเก็บข้อมูลทั้งหมด และทำการเชื่อมต่อโครงข่ายประสาทเทียมที่สร้างขึ้นด้วยโปรแกรม Matlab ทั้งหมดเข้ากับฐานข้อมูลไมโครซอฟต์ แอกเซส เพื่อใช้ในการรับส่งข้อมูลที่จะนำมาใช้งาน นอกจากนี้ผลจากการทำนายโหลดจากโปรแกรม Matlab ก็สามารถเก็บบันทึกลงสู่ฐานข้อมูลได้ ซึ่งสะดวกในการเรียกใช้งานและประหยัดเวลาในการจัดการกับข้อมูลที่จัดเก็บ ซึ่งระบบฐานข้อมูลดังกล่าวจะได้กล่าวในบทต่อไป

## บทที่ 7

### การสร้างและการติดต่อฐานข้อมูล

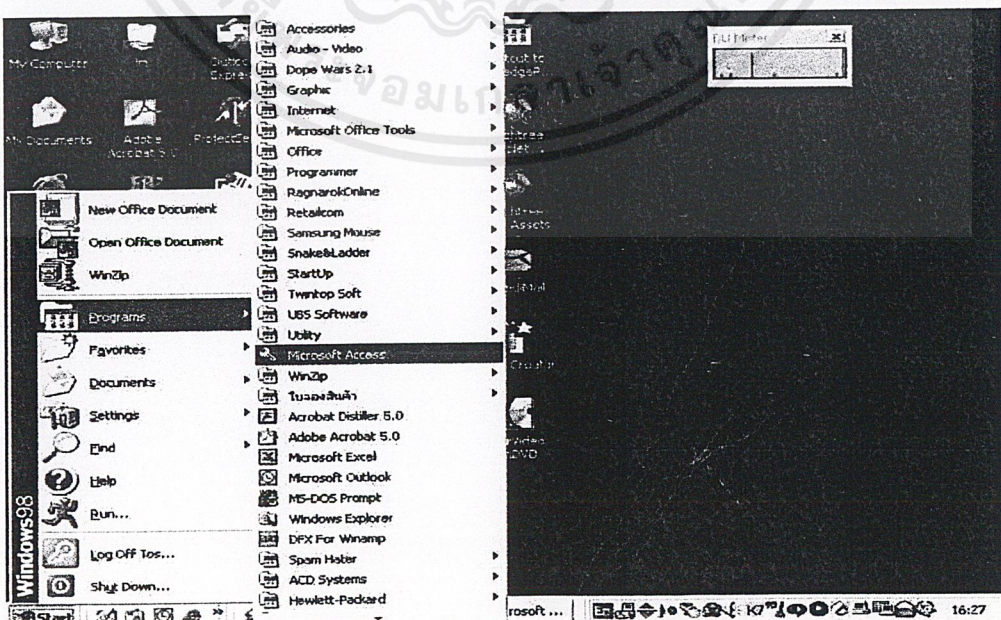
#### 7.1 ความรู้เบื้องต้นเกี่ยวกับ แฟ้มข้อมูล และ ฐานข้อมูล

ในอดีตการจัดเก็บข้อมูลในระบบคอมพิวเตอร์นั้น เรามักจะนำข้อมูลมาจัดเก็บลงใน แผ่นดิสก์ (Diskette) หรือ ฮาร์ดดิสก์(Harddisk) ในลักษณะของ แฟ้มข้อมูล(Data File) ต่างๆ โดยแต่ละแฟ้มที่ว่านี้ก็จะมีข้อมูลที่เกี่ยวข้องกับการทำงานในเรื่องใดเรื่องหนึ่งเท่านั้น ซึ่งนอกจากจะต้องเสียเวลาในการป้อนข้อมูลที่ซ้ำซ้อนกัน เข้าสู่ระบบคอมพิวเตอร์แล้ว ยังก่อให้เกิดปัญหาความไม่ถูกต้องตรงกันของข้อมูลด้วย ด้วยเหตุนี้ จึงมีการออกแบบวิธีการจัดเก็บข้อมูลในระบบคอมพิวเตอร์กันใหม่ โดยนำข้อมูลที่ต้องการใช้มาเก็บรวมกันไว้ที่เดียวและกำหนดกฎเกณฑ์การใช้ข้อมูลให้เป็นระบบ ทั้งนี้เพื่อลดการเก็บข้อมูลซ้ำซ้อนและป้องกันมิให้เกิดปัญหาความไม่ถูกต้องตรงกันของข้อมูล โดยเราเรียกที่รวมกันของข้อมูลทั้งหมดนี้ว่า “ฐานข้อมูล (Database)”

ปัจจุบัน มีโปรแกรมที่ถูกออกแบบมาเพื่อให้ทำงานในด้านการจัดการฐานข้อมูลหลายโปรแกรม ในจำนวนนี้มีโปรแกรมหนึ่งซึ่งกำลังเป็นที่นิยมใช้งานกันอยู่ทั่วไป เนื่องจากเป็นโปรแกรมที่มีราคาไม่สูงนัก และมีประสิทธิภาพในการทำงานดี คือ โปรแกรม Microsoft Access [6]

#### 7.2 การสร้างฐานข้อมูลใน Microsoft Access

การสร้างฐานข้อมูลใน Microsoft Access ที่ใช้ในการทดลองสามารถทำตามขั้นตอนได้ดังนี้  
เรียกโปรแกรม Microsoft Access จากเมนู Program โดยคลิกที่ปุ่ม Start ดังรูปที่ 7-1



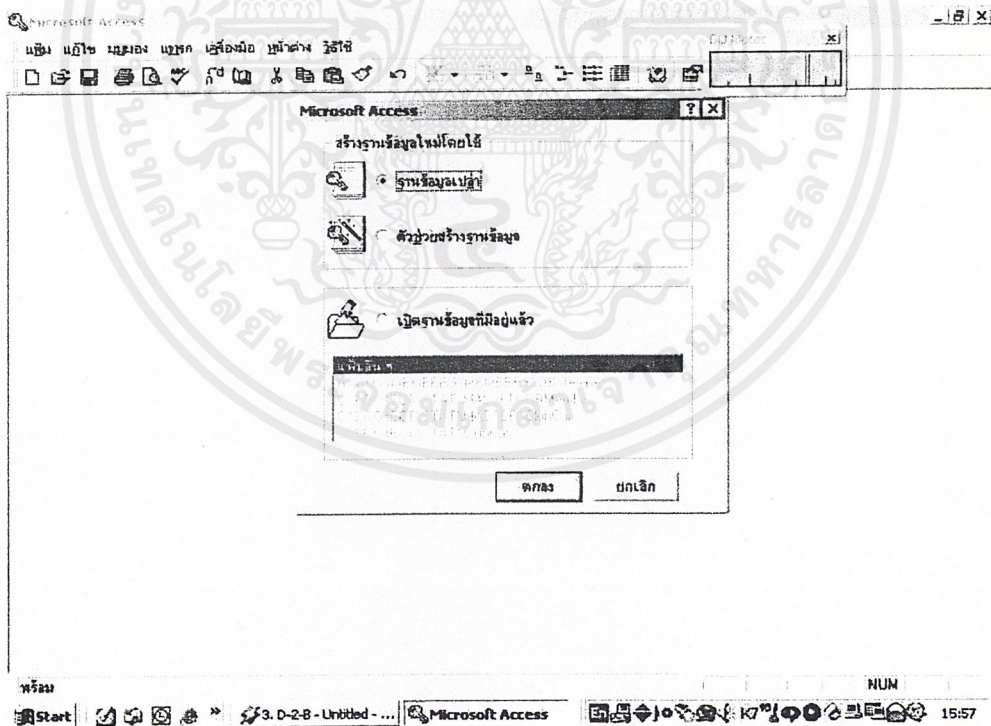
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
รูปที่ 7-1 แสดงการเรียกโปรแกรม Microsoft Access  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมีเหตุดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อเปิดโปรแกรม Microsoft Access ขึ้นมา จะมีกรอบข้อความคำสั่ง ให้เลือกใช้งาน 3 คำสั่ง คือ ฐานข้อมูลเปล่า, ตัวช่วยสร้างฐานข้อมูล และ เปิดฐานข้อมูลที่มีอยู่แล้ว โดยจะมีข้อแตกต่างกันคือ

ฐานข้อมูลเปล่า เป็นคำสั่งที่ใช้สร้างฐานข้อมูลในลักษณะที่เราจะต้องบอกรายละเอียดต่างๆ ของข้อมูลที่ต้องการจัดเก็บว่ามีฟิลด์(Field) อะไรบ้าง และแต่ละฟิลด์นั้นมีลักษณะอย่างไร เป็นข้อความหรือตัวเลข จำนวนกี่หลัก เป็นต้น

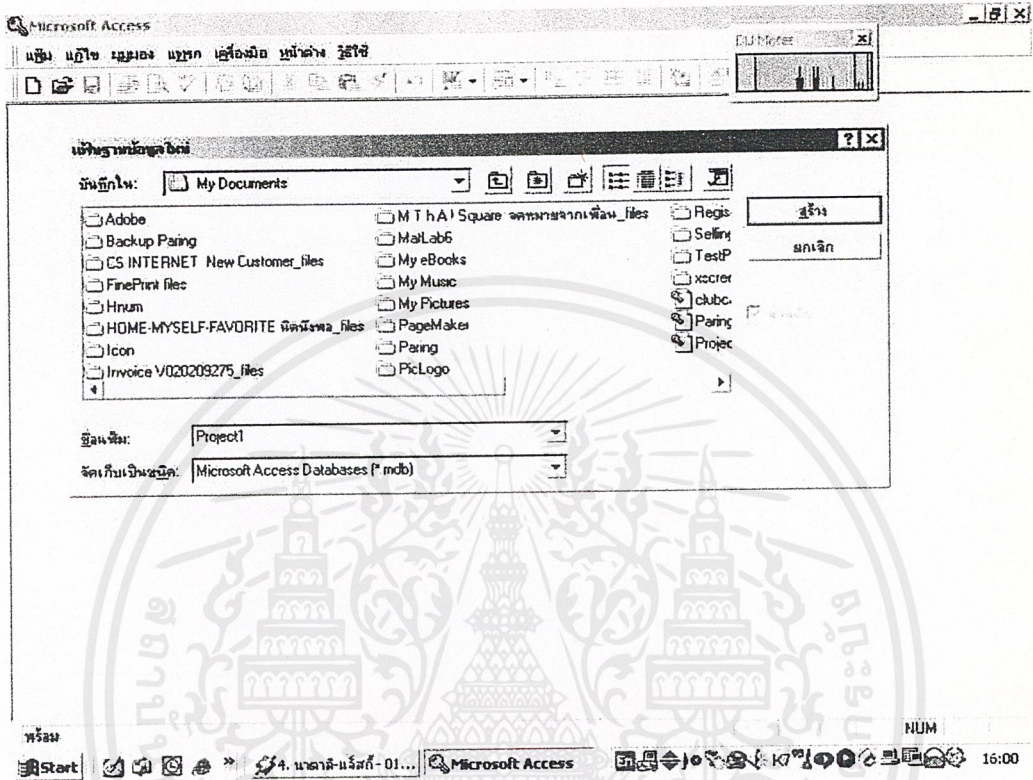
ตัวช่วยสร้างฐานข้อมูล เป็นคำสั่งที่ใช้สร้างฐานข้อมูลด้วยวิธีการง่ายๆ โดย Access จะคอยทำหน้าที่เป็นผู้ช่วย ให้คำแนะนำในการสร้างฐานข้อมูลไปทีละขั้นตอน โดยมีการเตรียม ฟิลด์(Field) ข้อมูลมาตรฐานในเรื่องต่างๆ

เปิดฐานข้อมูลที่มีอยู่แล้ว เป็นคำสั่งที่ใช้สำหรับอ่านฐานข้อมูลที่เราเคยสร้างไว้แล้ว ขึ้นมาใช้งาน แม้ว่า การสร้างฐานข้อมูล ด้วยคำสั่ง ตัวช่วยสร้างฐานข้อมูล นั้น จะสามารถทำได้ง่ายๆ แต่ก็มักพบว่า ข้อมูลหลายอย่างที่ Access เตรียมไว้ให้เราเลือกนั้น มีลักษณะไม่ตรงกับความต้องการใช้งานจริง โดยเฉพาะเมื่อต้องการกำหนดรายละเอียดของข้อมูล ดังนั้น จึงเราเลือกใช้คำสั่ง ฐานข้อมูลเปล่า แล้วคลิกปุ่ม ตกลง เพื่อสร้างฐานข้อมูลใหม่ ดังรูปที่ 7-2



รูปที่ 7-2 แสดงการสร้างฐานข้อมูลใหม่

จากนั้นจะเริ่มเข้าสู่ขั้นตอนการสร้างฐานข้อมูล โดยจะมีกรอบแสดงข้อความและรอให้ตั้งชื่อฐานข้อมูล ที่จะสร้างขึ้นดังรูปที่ 7-3 ทำการใส่ชื่อไฟล์ของฐานข้อมูล ในการทดลองนี้ใช้ชื่อไฟล์ว่า Project1 จากนั้นกดปุ่ม สร้าง



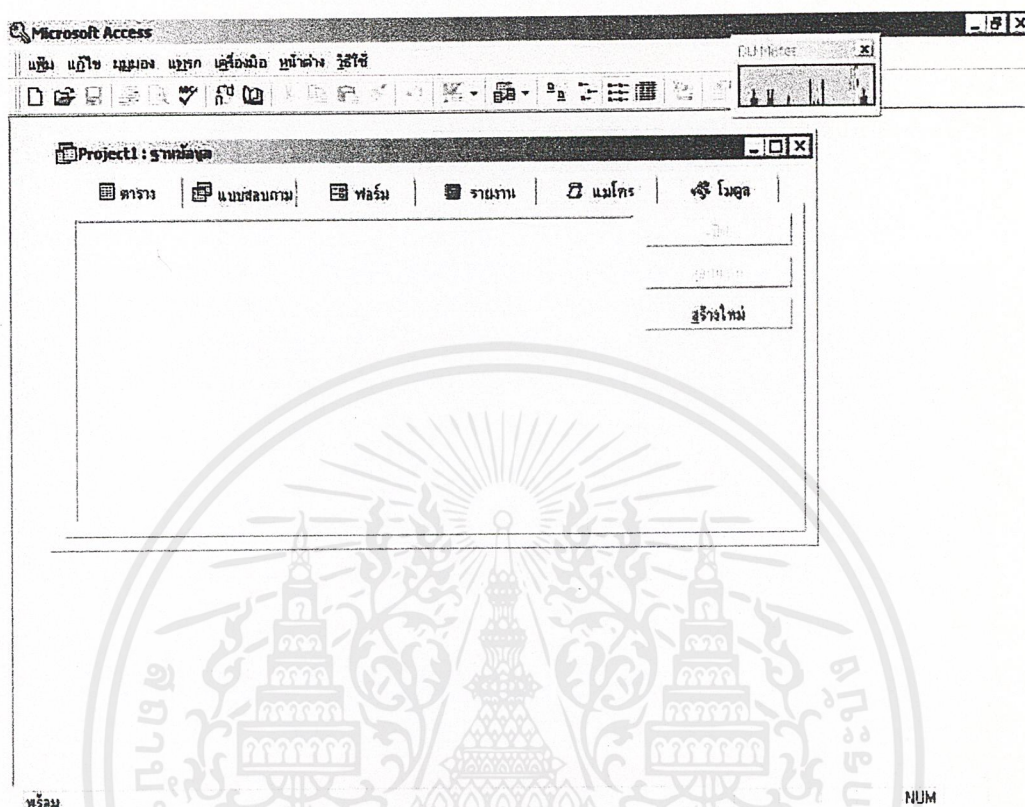
รูปที่ 7-3 แสดงการตั้งชื่อไฟล์ฐานข้อมูล

เมื่อดังชื่อไฟล์ฐานข้อมูลแล้วจะเข้าสู่เมนูหลักของ Microsoft Access ดังรูปที่ 7-4 จะแสดงกรอบหน้าต่างของ ฐานข้อมูล ที่มีชื่อว่า Project1 ซึ่งในกรอบหน้าต่างนี้ มีองค์ประกอบของฐานข้อมูลแสดงอยู่ 6 อย่าง คือ

1. ตาราง (Table) เป็นที่เก็บของข้อมูลต่างๆ ในฐานข้อมูล
2. แบบสอบถาม (Queries) เป็นตัวที่ใช้กำหนดเงื่อนไข เพื่อเลือกเอาเฉพาะข้อมูลที่ต้องการใช้จริงๆ ในแต่ละงาน
3. ฟอ้ม (Form) เป็นตัวที่ใช้กำหนดรูปแบบ ในการแสดงผลพร้อมหน้าจอภาพ
4. รายงาน (Report) เป็นตัวที่ใช้กำหนดรูปแบบ ในการแสดงผลพร้อมที่จะพิมพ์ออกทางกระดาษ
5. มาโคร (Macro) ใช้สำหรับสร้างคำสั่งพิเศษ ที่มีการทำงานซับซ้อนมากขึ้น
6. โมดูล (Module) ใช้สำหรับการสร้างโปรแกรม เพื่อให้ Access ทำงานหลายๆคำสั่งต่อเนื่องกัน โดยอัตโนมัติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในการทดลองนี้ เราต้องการสร้าง ตาราง (Table) เพื่อเป็นที่เก็บของข้อมูล ดังนั้น คลิกเลือกกรอบหน้าต่างที่มีคำว่า ตาราง แล้วคลิกปุ่ม สร้างใหม่ เพื่อทำการสร้างตารางใหม่



รูปที่ 7-4 แสดงหน้าจอหลักของ Microsoft Access

หลังจากกดปุ่ม สร้างใหม่ จอภาพก็จะแสดงกรอบหน้าต่างชื่อ ตารางใหม่ ดังรูปที่ 7-5 จะเห็นว่า มีคำสั่งให้เราเลือกว่า ต้องการสร้างฐานข้อมูลด้วยวิธีใด ได้แก่

Datasheet View หมายถึง สร้างโดยใช้ตารางข้อมูล

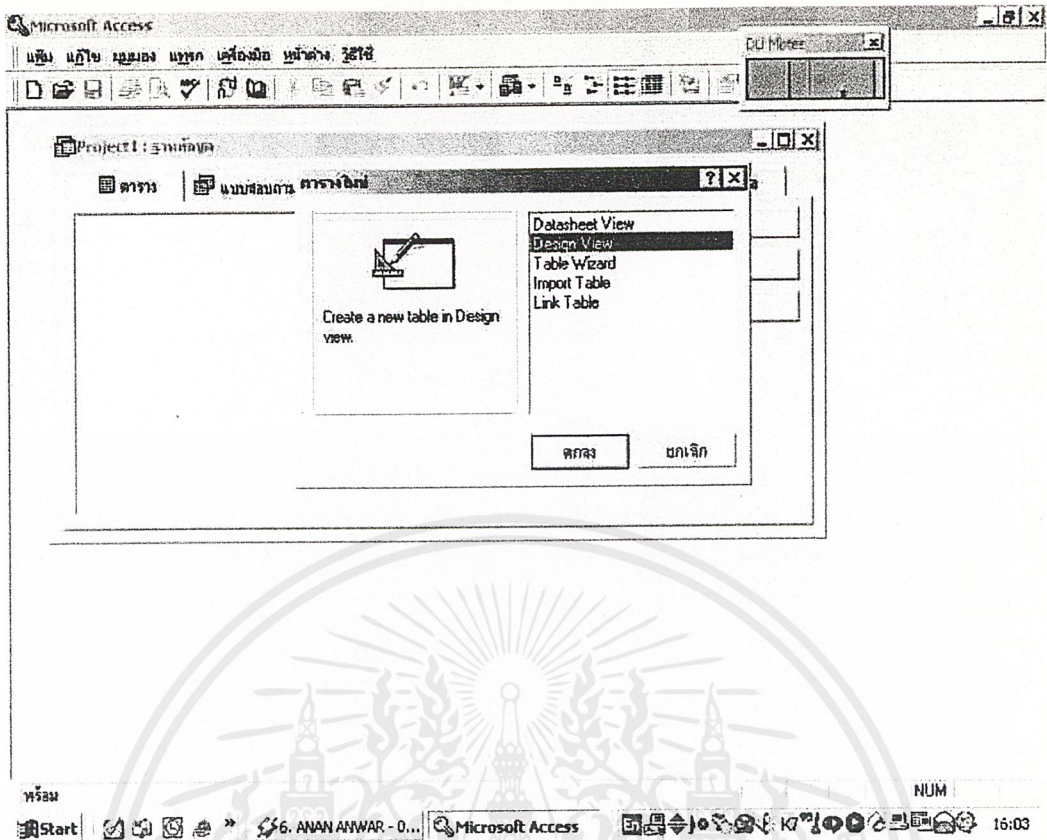
Design View หมายถึง สร้างโดยใช้ตารางระบุนรายละเอียดของข้อมูลที่ต้องการจัดเก็บ

Table Wizard หมายถึง สร้างโดยให้ Access คอยเป็นผู้ช่วย

Import Table หมายถึง สร้างโดยวิธีอ่านข้อมูลหรือคัดลอกข้อมูลจาก Table อื่นมาใช้

Link Table หมายถึง สร้างโดยวิธีการเชื่อมโยงไปหาข้อมูลใน Table อื่น

ในการทดลองนี้เลือกการสร้างฐานข้อมูลโดยวิธี Design View จากนั้นกดปุ่ม ตกลง



รูปที่ 7-5 แสดงการเลือกรูปแบบของตาราง

หลังจากเลือกวิธีการสร้างฐานข้อมูล ที่หน้าจอจะแสดงตารางเพื่อให้กรอกรายละเอียดของข้อมูลที่ต้องการจัดเก็บ ดังรูปที่ 7-6 จะมีช่องว่างให้เราป้อนรายละเอียด 3 ช่องคือ

ชื่อเขตข้อมูล (Field Name) เป็นช่องที่ใช้ป้อนฟิลด์ (Field) ข้อมูล  
ชนิดข้อมูล (Data type) เป็นช่องที่ใช้กำหนดว่าฟิลด์ (Field) นั้น มีลักษณะเป็นข้อมูลแบบใด  
(เป็นตัวเลขหรือข้อความ, ยาวจำนวนกี่หลัก ฯลฯ)

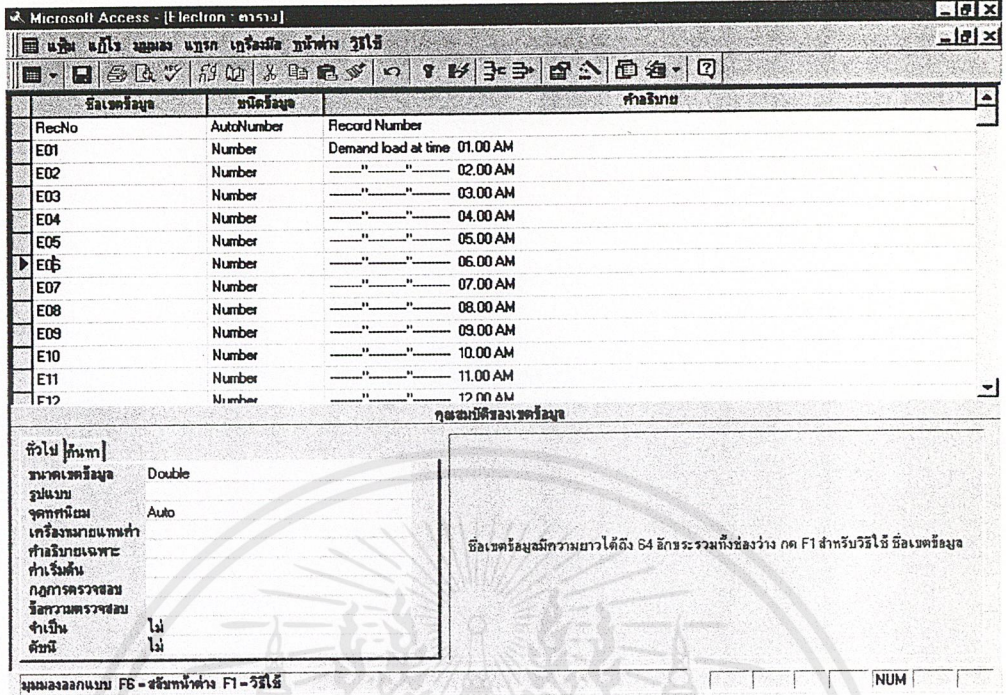
คำอธิบาย (Description) เป็นช่องที่ใช้บันทึกคำอธิบายเกี่ยวกับรายละเอียดของข้อมูลในฟิลด์นั้น

จากการทดลองทำการกรอกรายละเอียดของข้อมูลในตาราง Electron ดังต่อไปนี้

ชื่อเขตข้อมูล มีทั้งหมด 31 ชื่อ โดยกำหนดในแถวที่ 1 ให้เป็นชื่อ RecNo และตั้งแต่แถวที่ 2 ถึงแถวที่ 31 กำหนดให้เป็นชื่อ E01 ถึง E30 ตามลำดับ

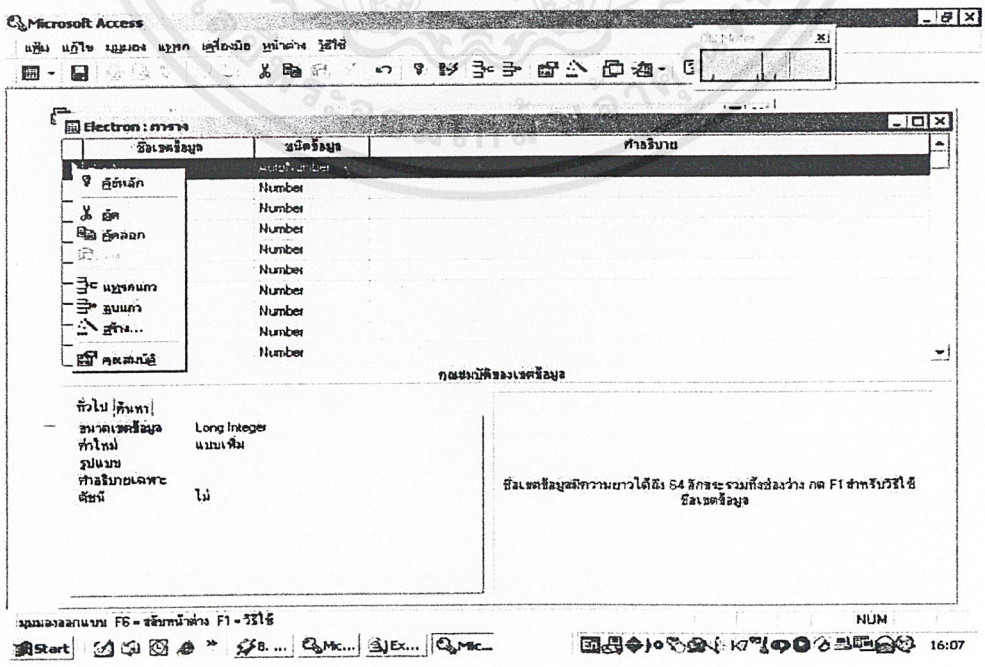
ชนิดข้อมูล กำหนดให้ RecNo เป็น AutoNumber และ E01 ถึง E30 เป็น Number

คำอธิบาย กำหนดให้ RecNo เป็นวันที่ทำการบันทึกค่าใช้พลังงานไฟฟ้าตั้งแต่ 1 มีนาคม – 31 พฤษภาคม 2544, E01 ถึง E24 บันทึกค่าการใช้ไฟฟ้าเป็นรายชั่วโมงในแต่ละวัน, E25 และ E26 บันทึกข้อมูลอุณหภูมิต่ำสุดและสูงสุดของแต่ละวันตามลำดับ, E27 ถึง E29 เก็บค่า Day code ของวันในรอบสัปดาห์ และ E30 เก็บค่า Holiday ของวันที่เป็นวันหยุด แสดงการกรอกรายละเอียดได้ดังรูปที่ 7-6



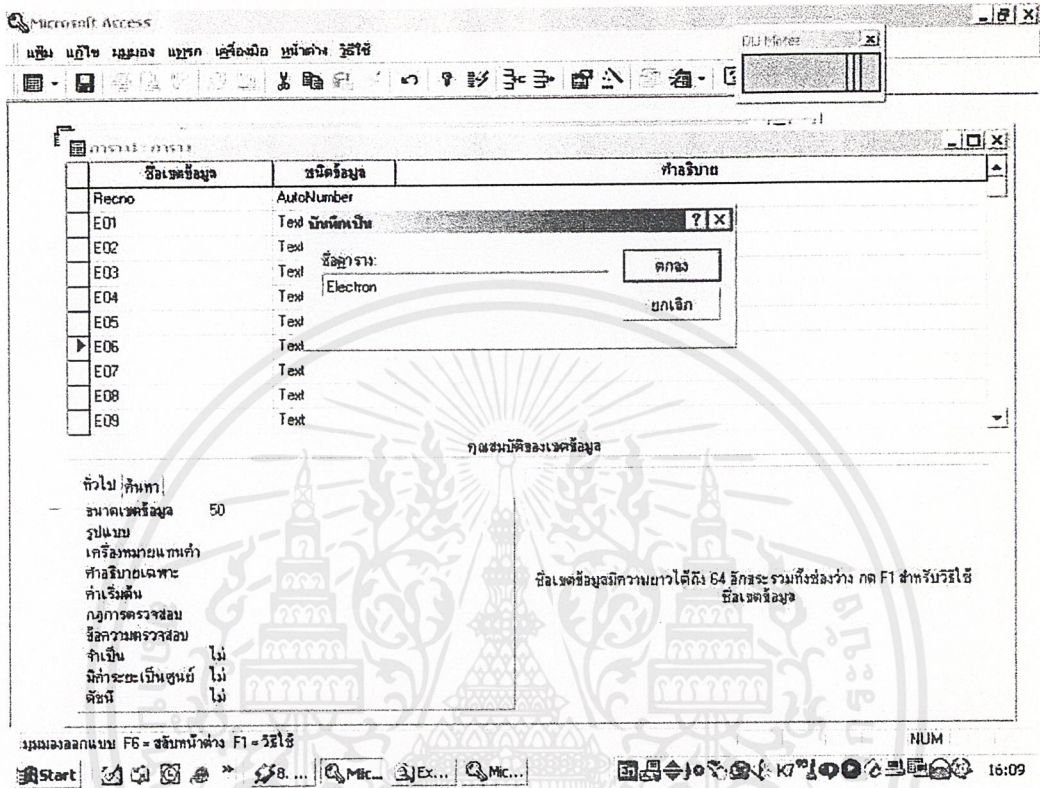
รูปที่ 7-6 แสดงการกำหนดรายละเอียดของฟิลด์ข้อมูล

หลังจากที่ได้กำหนดรายละเอียดของฟิลด์ต่างๆ เป็นที่เรียบร้อยแล้ว จะต้องทำการเซ็ทคีย์หลัก โดยในตารางนี้จะเซ็ทให้ฟิลด์ RecNo เป็นคีย์หลักโดยลากเมาส์ไปอยู่เหนือคำว่า RecNo แล้วกดปุ่มขวาบนเมาส์ ลากเมาส์ไปอยู่เหนือคำว่า คีย์หลักแล้วกดปุ่มซ้ายบนเมาส์ ดังรูปที่ 7-7



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 รูปที่ 7-7 แสดงการเซ็ทค่าคีย์หลักของตารางข้อมูล Electron  
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งที่ ผมไม่ได้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีมีการนำไปใช้

เมื่อทำการปิดหน้าต่างของตารางข้อมูลแล้วจะขึ้นหน้าต่างให้ใส่ชื่อของตาราง ในการทดลองนี้ใส่ชื่อว่า Electron เพื่อให้ตรงกับ Source Code ของโปรแกรม Matlab



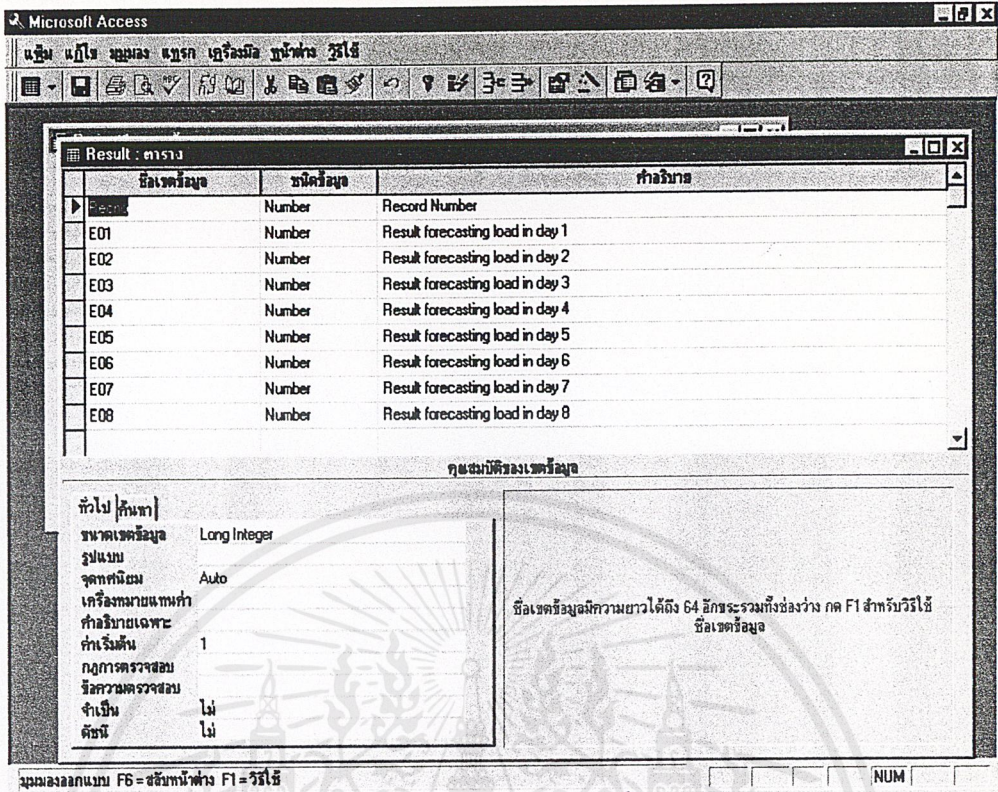
รูปที่ 7-8 แสดงการใส่ชื่อตารางข้อมูล Electron

ทำตารางข้อมูลเพิ่มอีก 1 ตาราง เพื่อทำการเก็บผลจากการทำนายโหลด โดยให้ชื่อตารางข้อมูลว่า Result เพื่อให้ตรงกับ Source Code ของโปรแกรม Matlab โดยเก็บตารางข้อมูลนี้ให้อยู่ในชื่อไฟล์ฐานข้อมูลเดียวกันกับตารางข้อมูลที่ชื่อ Electron (Project1.mdb) โดยวิธีการสร้างตารางทำเช่นเดียวกันกับที่ได้กล่าวมาตอนต้น โดยรายละเอียดของข้อมูลมีดังนี้

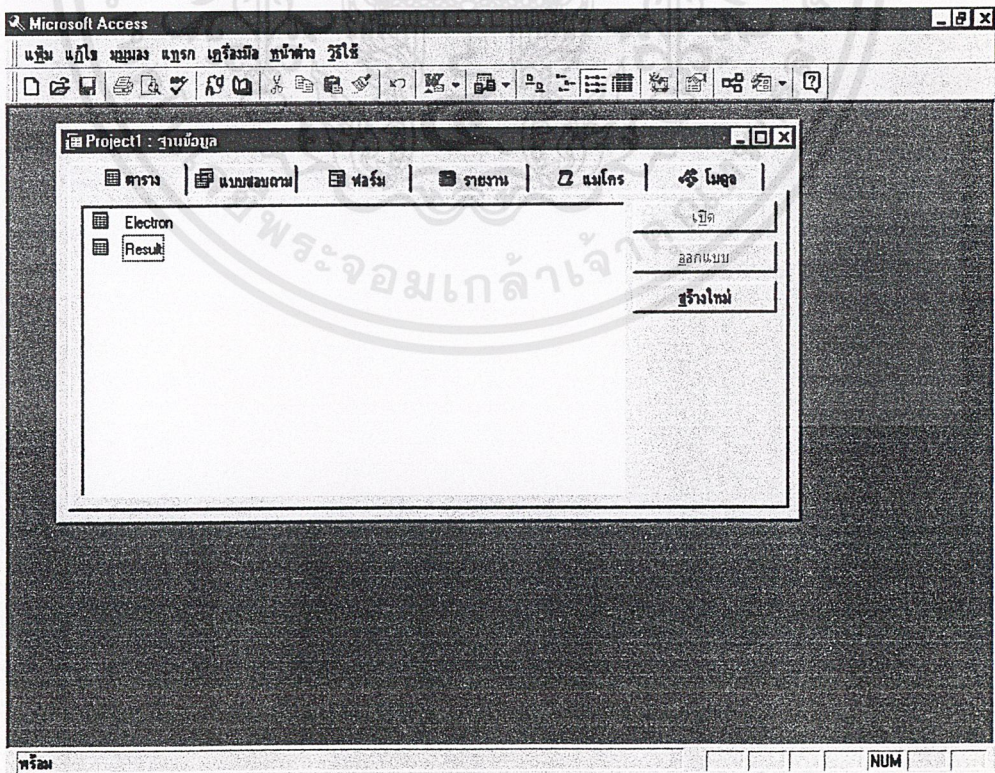
ชื่อเขตข้อมูล มีทั้งหมด 9 ชื่อ โดยกำหนดในแถวที่ 1 ให้เป็นชื่อ RecNo และตั้งแต่แถวที่ 2 ถึงแถวที่ 9 กำหนดให้เป็นชื่อ E01 ถึง E08 ตามลำดับ

ชนิดข้อมูล กำหนดให้ RecNo เป็น AutoNumber และ E01 ถึง E08 เป็น Number

คำอธิบาย กำหนดให้ RecNo เป็นผลการทดสอบโครงข่ายประสาทเทียมตั้งแต่ 24 พฤษภาคม – 31 พฤษภาคม 2544 และให้ E01 ถึง E08 เก็บค่าเอาต์พุตที่ได้จากการทำนายโหลดของโครงข่ายประสาทเทียม แสดงดังรูปที่ 7-9



รูปที่ 7-9 แสดงรายละเอียดข้อมูลของตาราง Result สำหรับเก็บผลการทดสอบการทำนายโหลด



เอกสารนี้เป็นรูปที่ 7-10 แสดงไฟล์ฐานข้อมูลชื่อ Project1.mdb ซึ่งประกอบด้วยตารางข้อมูล 2 ชุด ด้านการคำนวณค่าไม่พิกัดใดๆ ทั้งสิ้น อีกทั้งยังมีให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Microsoft Access - [Electron : ตาราง]

เมนู แอปพลิเคชัน ฐานข้อมูล ฐานข้อมูล เครื่องมือ ฐานข้อมูล ฐานข้อมูล

RowNo	E01	E02	E03	E04	E05	E06	E07
1	17.6	17.6	17.6	17.6	17.6	15.1	15.1
2	20.5	20.5	20.5	17.8	17.8	18.8	18.8
3	16.8	16.8	16.1	15.8	16.9	17.6	14.6
4	11.2	11.2	11	9.9	12.1	12.1	9.7
5	9	9	9	9.1	9.1	9.1	10.2
6	17.9	17.9	17.9	17.4	17.4	17.4	15.4
7	16.6	17.2	17.6	17.6	18.1	18.3	14.1
8	16.2	16.8	15.1	16.2	17.7	14.9	13.1
9	16.7	16.7	16.7	16.7	16.7	14	14
10	15	15	15	15	15	13	13
11	13.6	13.6	13.6	9.7	9.7	9.7	10.8
12	8.2	8.2	8.2	9.5	9.5	9.5	9.5
13	16.1	16.1	16.1	14	14	14	14.6
14	20.6	20.6	20.6	17.8	17.8	17.8	19.9
15	19.2	18	19.4	19.6	18	16.4	15.1
16	19.8	18.4	17.9	18.7	18.5	17.5	15.8

จำนวน: 16 จาก 92

Record Number NUM

รูปที่ 7-11 แสดงการเก็บข้อมูลโหลดทั้งหมดในตาราง Electron เพื่อใช้ในการทำนายโหลด

Microsoft Access - [Result : ตาราง]

เมนู แอปพลิเคชัน ฐานข้อมูล ฐานข้อมูล เครื่องมือ ฐานข้อมูล ฐานข้อมูล

RowNo	E01	E02	E03	E04	E05	E06	E07
1	19.2574116260	19.239835217	19.242607718	19.240112137	19.311200626	19.256605723	19.239891593
2	21.9849534574	21.877158959	21.894183282	21.878859712	22.312795180	21.980017857	21.877505205
3	22.1980724934	21.899524608	21.946812892	21.904251006	23.091509463	22.184449434	21.900486864
4	19.6893007245	19.455317837	19.492396647	19.459024109	20.388059192	19.678629407	19.45607241
5	19.2465368517	19.323122238	19.311064955	19.321918327	19.010033881	19.250056005	19.322877153
6	18.0894331319	17.860467904	17.896749550	17.864094478	18.773769574	18.078989711	17.86120625
7	18.6116369165	18.742325092	18.721770625	18.740273116	18.206293567	18.617648830	18.741907369
8	26.7928192974	27.027302051	26.990411345	27.023619019	26.066696764	26.803602109	27.026552289
9	29.9951302874	30.193355839	30.16218151	30.190243706	29.380004691	30.004249884	30.192722300
10	30.9533005998	30.711461363	30.749715818	30.715284030	31.682912797	30.942247461	30.712239613
11	31.1327928711	31.031905866	31.047834939	31.033497132	31.440082368	31.128172083	31.032229821
12	27.7578276029	27.703251626	27.711862823	27.704111761	27.924617122	27.755326009	27.703426733
13	29.3491918683	29.325402052	29.329154377	29.325776833	29.422022008	29.348100981	29.325478349
14	30.9885759310	30.82435058	30.850299835	30.826943157	31.486859135	30.981060687	30.824878389
15	30.6794415081	30.316126417	30.373692419	30.321880417	31.765789681	30.66286866	30.317297992
16	37.4019315573	37.258630399	37.282091832	37.261874116	37.835870588	37.395411786	37.286887179

จำนวน: 16 จาก 24

Record Number NUM

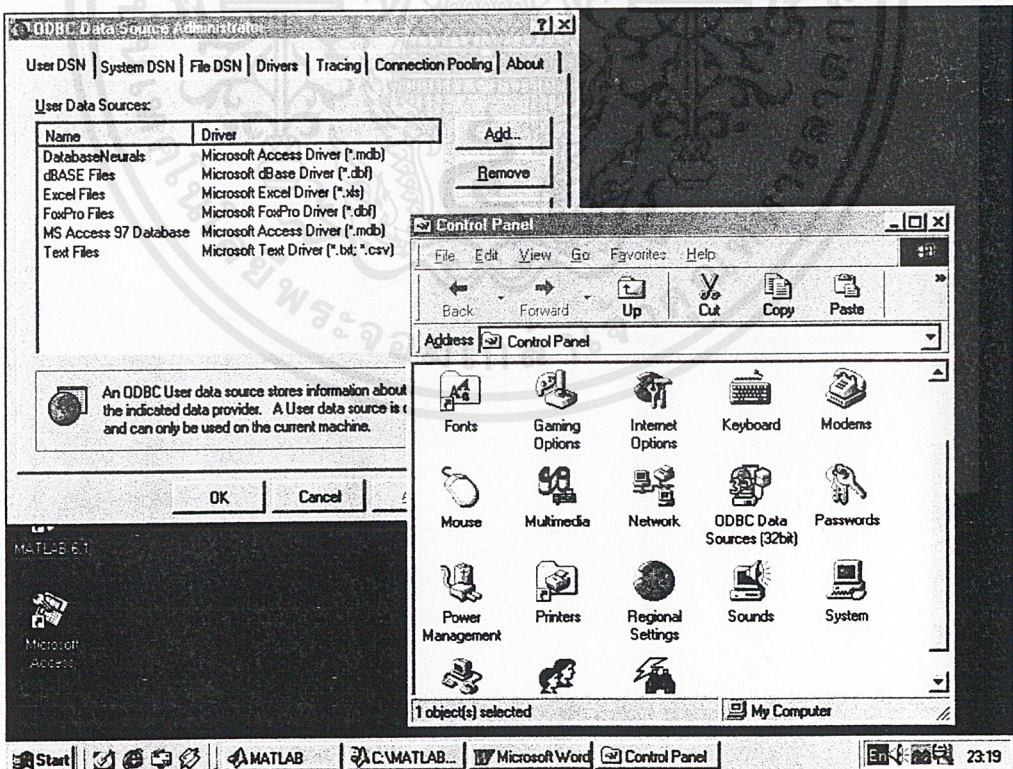
เอกสารนี้เป็นรูปที่ 7-12 แสดงการเก็บข้อมูลจากผลการทำนายโหลดโดยใช้โครงข่ายประสาทเทียม  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 7.3 การติดต่อฐานข้อมูล

#### Open Database Connectivity (ODBC)

เมื่อเราสร้างฐานข้อมูลขึ้นโดยใช้โปรแกรม Microsoft Access หลังจากนั้นหากต้องการเรียกใช้งานฐานข้อมูลดังกล่าว เราก็สามารถใช้โปรแกรม Microsoft Access เรียกใช้งานได้เลย แต่ถ้าต้องการใช้โปรแกรมแอปพลิเคชันอื่นที่ไม่ใช่ Microsoft Access อาทิเช่น Delphi หรือ ASP มาเรียกใช้ฐานข้อมูลที่สร้างจาก Microsoft Access โปรแกรมแอปพลิเคชันดังกล่าว ก็ต้องมีคุณสมบัติและความสามารถในการติดต่อกับฐานข้อมูลชนิดที่สร้างมาจากโปรแกรม Microsoft Access ซึ่งหมายถึงว่าต้องอ่านฐานข้อมูลที่สร้างและเก็บเป็นไฟล์ข้อมูลที่มีส่วนขยายเป็น .MDB ได้นั่นเอง

เพื่อเป็นการแก้ปัญหาในการเรียกใช้ฐานข้อมูลที่สร้างมาจากโปรแกรมแอปพลิเคชันต่างโปรแกรม เราจะอาศัยโปรแกรมที่ใช้เทคนิค ODBC หรือ Open Database Connectivity ซึ่งโปรแกรมดังกล่าวจะช่วยให้โปรแกรมแอปพลิเคชันที่แตกต่างกัน สามารถใช้ฐานข้อมูลร่วมกันได้โดยโปรแกรมที่ทำหน้าที่เป็น ODBC จะถูกเรียกว่า ODBC driver ซึ่งหากต้องการติดต่อกับฐานข้อมูลชนิดใด ก็มักจะตั้งชื่อให้ตรงกับชื่อชนิดของฐานข้อมูลนั้น



รูปที่ 7-13 แสดงการเรียกใช้งานฐานข้อมูลที่สร้างจาก Microsoft Access Driver (\*.MDB) ได้ โดย

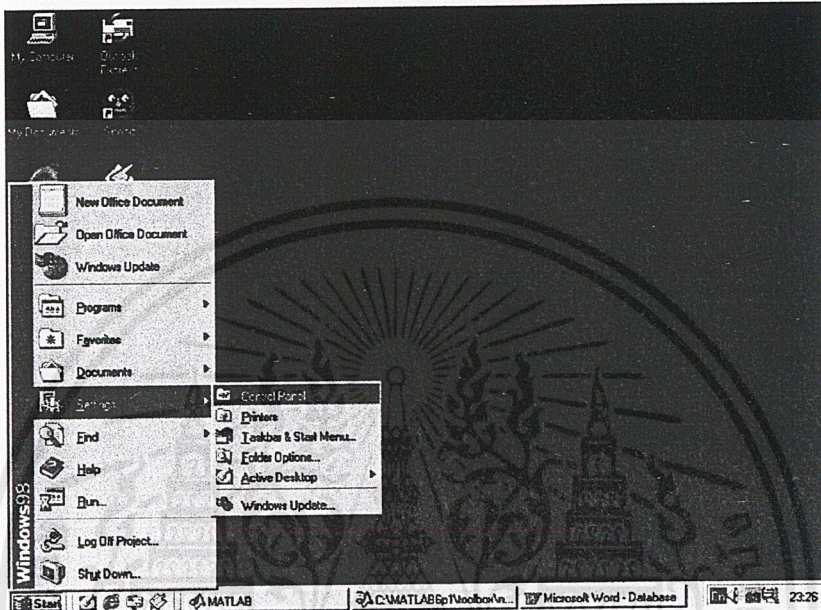
การเรียกใช้ผ่านทาง ODBC driver

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 7.4 การติดตั้ง ODBC

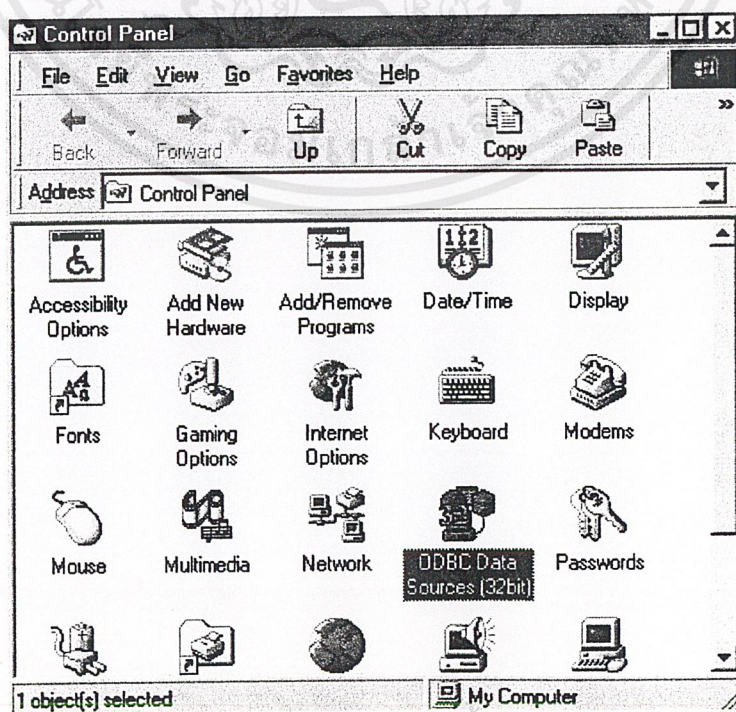
การติดตั้ง ODBC จะทำที่ Control Panel ซึ่งมีขั้นตอนในการติดตั้งดังนี้

1. เลือก Control Panel ที่ Setting ของเมนู Start ที่ Taskbar ดังรูปที่ 7-14



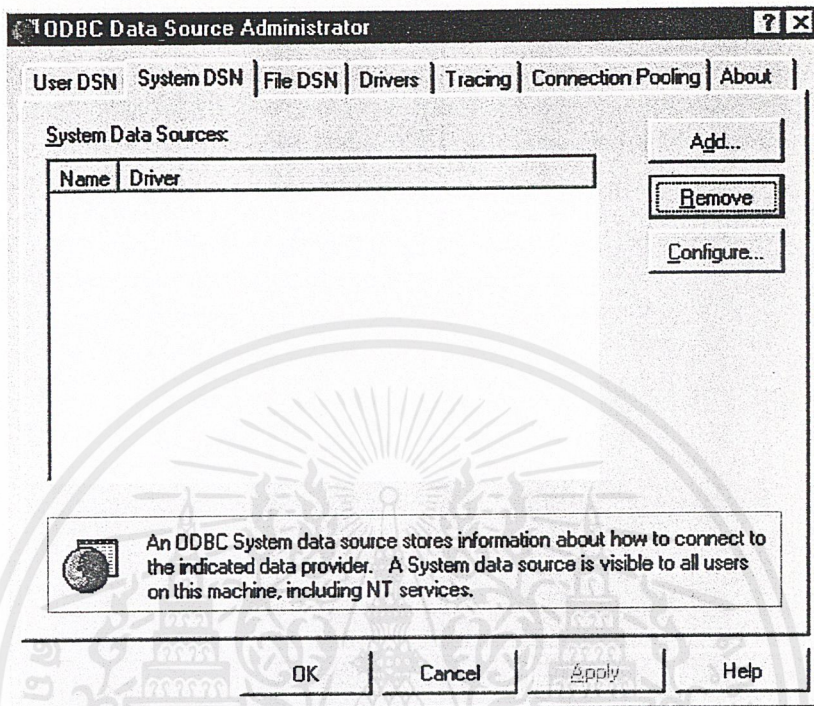
รูปที่ 7-14 แสดงการเรียก Control Panel จาก taskbar

2. แสดงไฟล์ต่าง ๆ ในโคเรกทอรี Control Panel เลือก ไอคอนของ ODBC ดังรูปที่ 7-15



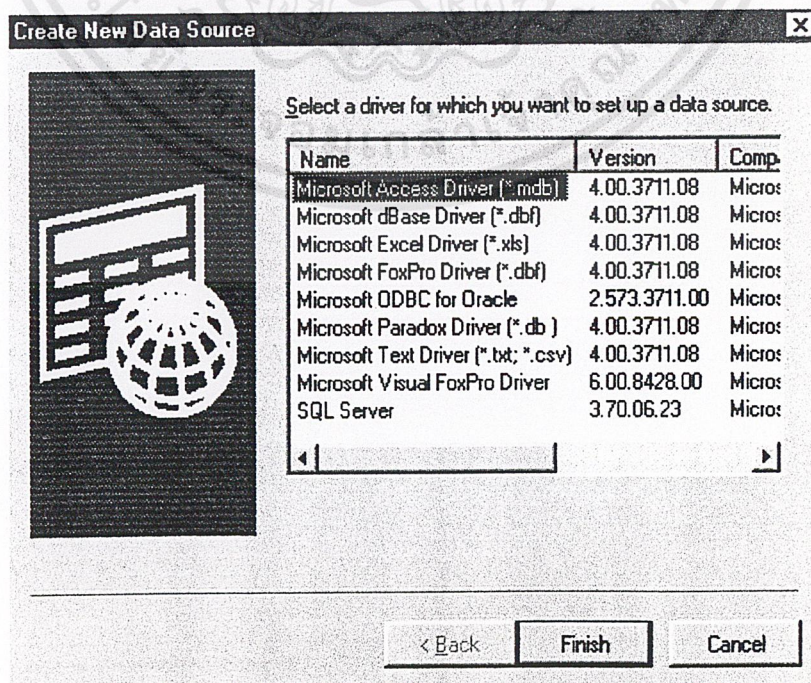
รูปที่ 7-15 แสดงวิธีการเข้าสู่การติดตั้ง ODBC

หลังจากเลือก ไอคอนของ ODBC แล้วจะแสดงองค์ประกอบของ ODBC เลือกแท็บ System DSN เพื่อทำการติดตั้ง System Data Source ของฐานข้อมูล แล้วคลิกปุ่ม Add... เพื่อทำการเพิ่ม System Data Source ดังรูปที่ 7-16



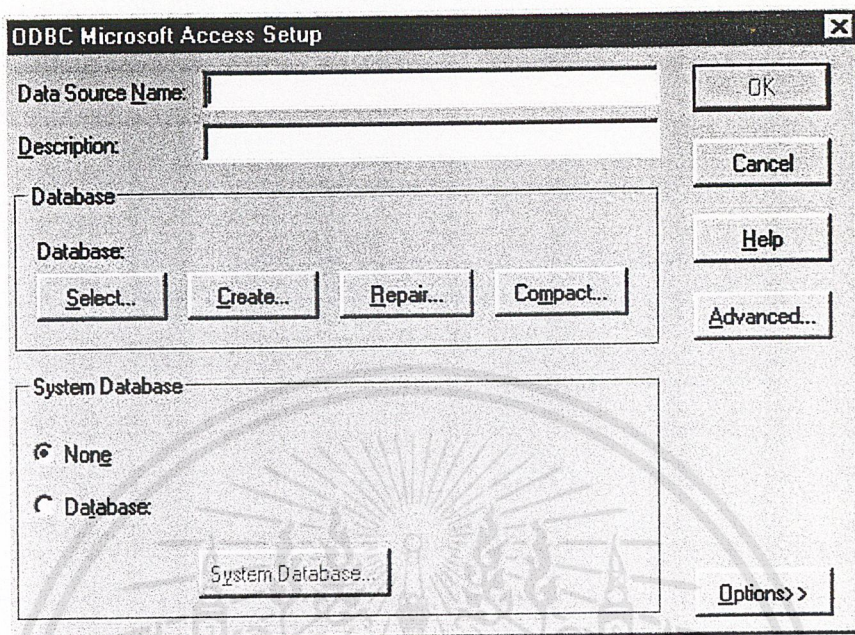
รูปที่ 7-16 แสดงการติดตั้ง System Data Source ของฐานข้อมูล

เลือกไดรฟ์เวอร์ของฐานข้อมูลที่ใช้ ในโปรแกรมนี้ใช้ Microsoft Access Driver แล้วคลิกปุ่ม Finish ดังรูปที่ 7-17



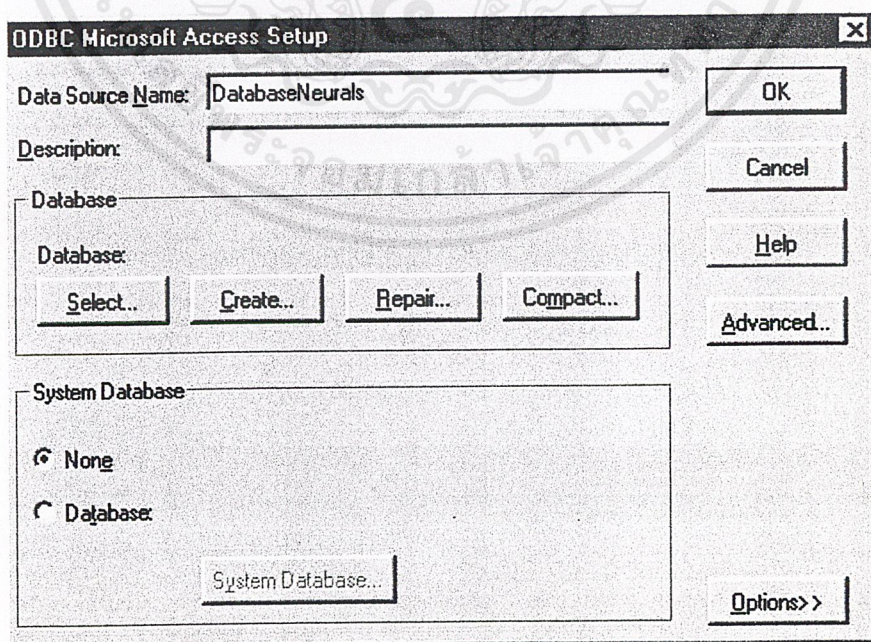
เอกสารนี้เป็นเอกสารที่สงวนไว้รูปที่ 7-17 แสดงการเลือกไดรฟ์เวอร์ของฐานข้อมูล ไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หลังจากกดปุ่ม Finish แล้วจะปรากฏฟอร์มเพื่อให้ติดตั้ง ODBC สำหรับ ACCESS ดังรูปที่ 7-18



รูปที่ 7-18 แสดงการติดตั้ง ODBC สำหรับ Microsoft Access

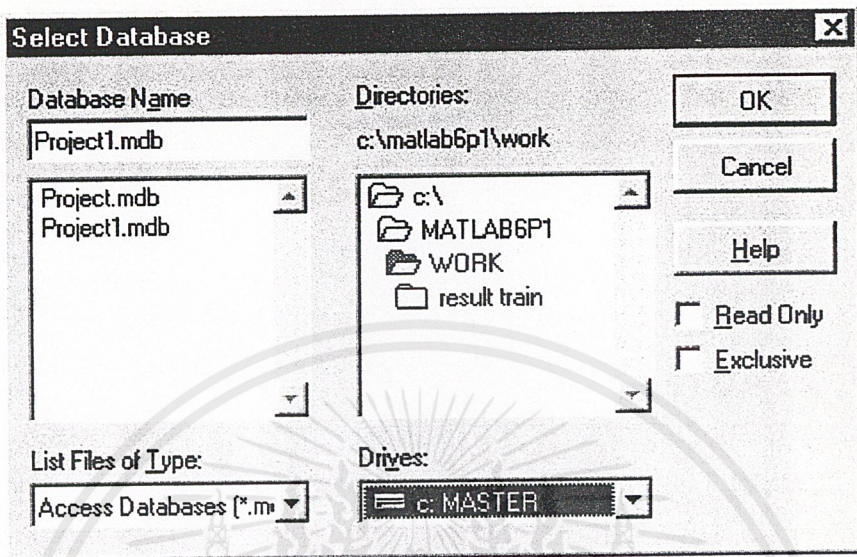
ทำการใส่ข้อมูล Data Source Name โดยชื่อที่ใช้ในการติดต่อฐานข้อมูลให้ชื่อว่า DatabaseNeurals เพื่อให้ตรงกับ Source Code ของโปรแกรม MatLab ดังรูปที่ 7-19



รูปที่ 7-19 แสดงการตั้งชื่อ DSN สำหรับติดต่อฐานข้อมูล Microsoft Access

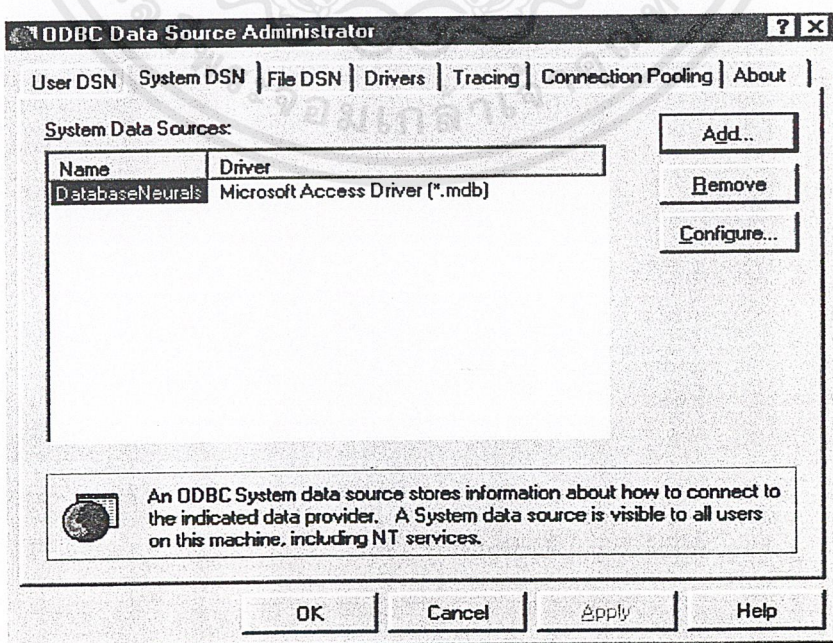
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หลังจากตั้งชื่อ DSN แล้ว กดปุ่ม Select แล้วเลือกไฟล์ฐานข้อมูลที่ต้องการติดต่อ ดังรูปที่ 7-20 โดยเลือกชื่อไฟล์ฐานข้อมูล Project1.mdb จากการสร้าง ในบทที่ได้กล่าวในตอนต้น จากนั้นกดปุ่ม OK



รูปที่ 7-20 แสดงวิธีการเลือกติดต่อฐานข้อมูล Microsoft Access ชื่อ Project1.mdb

หลังจากกดปุ่ม OK จะแสดงฟอร์มดังรูปที่ 7-21 เพื่อแสดง Data Source Name และ ไดรฟ์เวอร์ที่ติดตั้ง ODBC สำหรับ Microsoft Access



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
รูปที่ 7-21 แสดง DSN และ ไดรฟ์เวอร์ที่ติดตั้ง ODBC สำหรับ Microsoft Access  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามเผยแพร่ต่อสาธารณะและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ภาคผนวก

### Source Code

#### Neurals Networks โดยโปรแกรม MatLab

```

clear all

start = clock;

% Connect to a database.
connA=database('DatabaseNeurals','');

% Open cursor and execute a SQL statement.
cursorA=exec(connA,'select * from Electron order by recno');

cursorA=fetch(cursorA, 92);

inpu=cursorA.data;

% Number of data
endloop_Data = 31;
numberday = 92;
z =size(inpu);
x=z(1,1);
for j_loop = 1 : endloop_Data,
    for i_loop = 1 : x,
        inmatrix(i_loop,j_loop)=inpu{i_loop,j_loop};
    end
end
end

```

```

%zero mean and variance to 1 and is valid only for single dimension
for e_loop = 1 : endloop_Data,
    mean1(1,e_loop) = sum(inmatrix(:,e_loop))/length(inmatrix(:,e_loop));
end

for j_loop = 1 : endloop_Data,
    for i_loop = 1 : x,
        mn1(i_loop,j_loop) = inmatrix(i_loop,j_loop) - mean1(1,j_loop);
    end
end

for x_loop = 1 : endloop_Data,
    var(1,x_loop)=sqrt((sum(mn1(:,x_loop).^2))/length(inmatrix(:,x_loop)));
end

for j_loop = 1 : endloop_Data,
    for i_loop = 1 : x,
        zmv(i_loop,j_loop) = mn1(i_loop,j_loop) / var(1,j_loop);
    end
end

mx = max(zmv);
mn = min(zmv);

for j_loop = 1 : endloop_Data,
    awin(j_loop) = (( 0.9 - (-0.9) ) / ( mx(j_loop) - mn(j_loop) ) );
    bwin(j_loop) = 0.9 - ( awin(j_loop) * mx(j_loop) );
    awout(j_loop) = ( 0.9 - 0.1 ) / ( mx(j_loop) - mn(j_loop) );
    bwout(j_loop) = 0.9 - ( awout(j_loop) * mx(j_loop) );
end

```

```

for j_loop = 1 : endloop_Data,
    for i_loop = 1 : x,
        datainputbound(i_loop,j_loop) = ( zmv(i_loop,j_loop) * awin(j_loop) ) + bwin(j_loop);
        dataoutputbound(i_loop,j_loop) = ( zmv(i_loop,j_loop) * awout(j_loop) ) + bwout(j_loop);
    end
end

for m_loop = 1 : numberday,
    for n_loop = 1 : 24,
        datademandbound(m_loop,n_loop) = datainputbound(m_loop,n_loop+1);
    end
end

for o_loop = 1 : numberday-2,
    datatempbound(o_loop,1) = datainputbound(o_loop,26);
    datatempbound(o_loop,2) = datainputbound(o_loop,27);
    datatempbound(o_loop,3) = datainputbound(o_loop+1,26);
    datatempbound(o_loop,4) = datainputbound(o_loop+1,27);
    datatempbound(o_loop,5) = datainputbound(o_loop+2,26);
    datatempbound(o_loop,6) = datainputbound(o_loop+2,27);
end

for m_loop = 1 : numberday,
    daycodebound(m_loop,1) = datainputbound(m_loop,28);
    daycodebound(m_loop,2) = datainputbound(m_loop,29);
    daycodebound(m_loop,3) = datainputbound(m_loop,30);
end

```

```

%x = 1 start forecasting load in 3 March
for x_loop = 1 : numberday-2,
    for y_loop = 1 : 24,
        inputbound(x_loop,y_loop)=datademandbound(x_loop,y_loop);
    end

    for a_loop = 25 : 48,
        inputbound(x_loop,a_loop)=datademandbound(x_loop+1,a_loop-24);
    end

    for b_loop = 49 : 54,
        inputbound(x_loop,b_loop)=datatempbound(x_loop,b_loop-48);
    end

    for c_loop = 55 : 57;
        inputbound(x_loop,c_loop)=daycodebound(x_loop+2,c_loop-54);
    end
        inputbound(x_loop,58)=datainputbound(x_loop+2,31);
    end

    for j_loop = 1 : 58;
        for i_loop = 1 :82;
            input(i_loop,j_loop) = inputbound(i_loop,j_loop);
        end
    end

    inputboundtrain = input';

    for m_loop = 1 : 82,
        for n_loop = 1 : 24,
            outputtrainbound(m_loop,n_loop) = dataoutputbound(m_loop+2,n_loop+1);
        end
    end
end

```





```

-0.9 0.9
-0.9 0.9
-0.9 0.9
-0.9 0.9
-0.9 0.9
-0.9 0.9
-0.9 0.9
-0.9 0.9
-0.9 0.9
-0.9 0.9
-0.9 0.9
-0.9 0.9
-0.9 0.9
-0.9 0.9
-0.9 0.9
-0.9 0.9
-0.9 0.9],[u,w,24],{'tansig','tansig','logsig'},'trainlm');
net.trainParam.show=10;
net.trainParam.epochs=1000;
net.trainParam.max_fail=inf;
for q = 1: 3
net.layers{1}.initFcn = 'initwb';
net.inputWeights{1,1}.initFcn = 'rands';
net.biases{1}.initFcn = 'rands';
net.layers{2}.initFcn = 'initwb';
net.layerWeights{2,1}.initFcn = 'rands';
net.biases{2}.initFcn = 'rands';
net.layers{3}.initFcn = 'initwb';
net.layerWeights{3,2}.initFcn = 'rands';
net.biases{3}.initFcn = 'rands';

net = init(net);

```

```

[net,tr] = train(net,ptr,ptr,[ ],[ ],v);
outputtrain = sim(net,inputboundtrain);

for i_loop = 1 : 24 ,
    for j_loop = 1 : 82,
        R(q,1) = sqrt(sum((outputtrain(i_loop,j_loop)-outputboundtrain(i_loop,j_loop)).^2)/length
            (outputtrain(i_loop,j_loop))));
    end
end
if q = 1
    q11 = 1;
    w1 = outputtrain;
    x1 = net.iw{1,1};x2 = net.lw{2,1};x3 = net.lw{3,2};
    y1 = net.b{1};y2 = net.b{2};y3 = net.b{3};
    z1 = R(q,1)
else
    if R(q,1) < min(R(q-1,1))
        q11 = q;
        w1 = outputtrain;
        x1 = net.iw{1,1};x2 = net.lw{2,1};x3 = net.lw{3,2};
        y1 = net.b{1};y2 = net.b{2};y3 = net.b{3};
        z1 = R(q,1)
    end
end
end
q
w
end
w11 = w1;

```





```

-0.9 0.9
-0.9 0.9
-0.9 0.9],[u,w,24],{'tansig','tansig','logsig'},'trainlm');
net.iw{1,1} = x1; net.lw{2,1} = x2; net.lw{3,2} = x3;
net.b{1} = y1; net.b{2} = y2; net.b{3} = y3;
outputneuraltrain = sim(net,inputboundtrain);
outputneuraltest = sim(net,inputtestbound);
%unscale(output nueral training)
for i_loop = 1 : 24,
for j_loop = 1 : 82,
L(i_loop,j_loop) = [outputneuraltrain(i_loop,j_loop)/awout(i_loop+1) - bwout
(i_loop+1)/awout(i_loop+1)];
S(i_loop,j_loop) = [L(i_loop,j_loop)*var(i_loop+1) + mean1(i_loop+1)];
end
end
for j_loop = 1 : 24,
for i_loop = 1 : 82,
outputfromdata(i_loop,j_loop) = inmatrix(i_loop+2,j_loop+1);
end
end
outputrealtrain = outputfromdata';

for i_loop = 1 : 24,
for j_loop = 1 : 82 ,
errtrain(i_loop,j_loop) = 100 * abs( outputrealtrain(i_loop,j_loop) - S(i_loop,j_loop))/
outputrealtrain(i_loop,j_loop);
end
end

```

```

%unscale(output neural test)
for i_loop = 1 : 24,
    for j_loop = 1 : 8,
        L(i_loop,j_loop) = [outputneuraltest(i_loop,j_loop)/awout(i_loop+1) - bwout(i_loop+1)/awout
            (i_loop+1)];
        SS(i_loop,j_loop) = [L(i_loop,j_loop)*var(i_loop+1) + mean1(i_loop+1)];
    end
end
for m_loop = 1 : 8,
    for n_loop = 1 : 24,
        outputtest(m_loop,n_loop) = inmatrix(m_loop+84,n_loop+1);
    end
end
outputrealtest = outputtest';
for i_loop = 1 : 24,
    for j_loop = 1 : 8,
        errtest(i_loop,j_loop) = 100 * abs( outputrealtest(i_loop,j_loop) - SS(i_loop,j_loop))/
            outputrealtest(i_loop,j_loop);
    end
end
diff = [outputrealtest-SS]
SStest = [SS(:,1)', SS(:,2)', SS(:,3)',SS(:,4)', SS(:,5)', SS(:,6)',SS(:,7)', SS(:,8)'];
outputrealtestall = [outputrealtest(:,1)',outputrealtest(:,2)',outputrealtest(:,3)',outputrealtest
(:,4)',outputrealtest(:,5)',outputrealtest(:,6)',outputrealtest(:,7)',outputrealtest(:,8)'];
for i = 1 : length(outputrealtestall)
    percentdiff(i)=100 * abs( outputrealtestall(i) - SStest(i))/ outputrealtestall(i);
end
min1 = min(percentdiff)
max1 = max(percentdiff)
mean1 = mean(percentdiff)

```

```

x1_best = x1
x2_best = x2
x3_best = x3
y1_best = y1
y2_best = y2
y3_best = y3
RootMeanSquareError = z1
plot(percentdiff)

```

```

%storage data output testing from neural networks to database
colnames={'E01','E02','E03','E04','E05','E06','E07','E08'};
for ii = 1 : 24,
    for jj = 1 : 8,
        c{ii,jj} = SS(ii,jj);
    end
switch ii
case 1
    whereClause = 'where RecNo = 1';
case 2
    whereClause = 'where RecNo = 2';
case 3
    whereClause = 'where RecNo = 3';
case 4
    whereClause = 'where RecNo = 4';
case 5
    whereClause = 'where RecNo = 5';
case 6
    whereClause = 'where RecNo = 6';
case 7
    whereClause = 'where RecNo = 7';

```

case 8

whereClause = 'where RecNo = 8';

case 9

whereClause = 'where RecNo = 9';

case 10

whereClause = 'where RecNo = 10';

case 11

whereClause = 'where RecNo = 11';

case 12

whereClause = 'where RecNo = 12';

case 13

whereClause = 'where RecNo = 13';

case 14

whereClause = 'where RecNo = 14';

case 15

whereClause = 'where RecNo = 15';

case 16

whereClause = 'where RecNo = 16';

case 17

whereClause = 'where RecNo = 17';

case 18

whereClause = 'where RecNo = 18';

case 19

whereClause = 'where RecNo = 19';

case 20

whereClause = 'where RecNo = 20';

case 21

whereClause = 'where RecNo = 21';

case 22

whereClause = 'where RecNo = 22';

case 23

whereClause = 'where RecNo = 23';

case 24

whereClause = 'where RecNo = 24';

end

update(connA,'Result',colnames,c,whereClause)

end

%save valued trainingtime,weight and bias for load forecasting

save Trainingtime.txt Timetaken -ascii

save weight(1).txt x1 -ascii

save weight(2).txt x2 -ascii

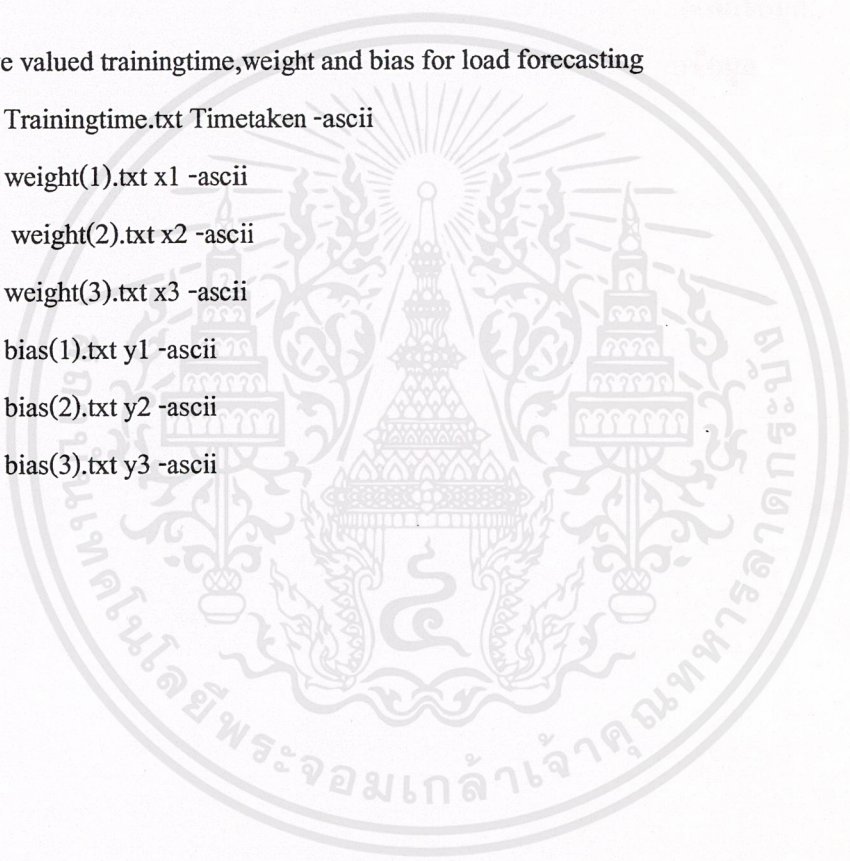
save weight(3).txt x3 -ascii

save bias(1).txt y1 -ascii

save bias(2).txt y2 -ascii

save bias(3).txt y3 -ascii

end



## กิตติกรรมประกาศ

ปริญญาบัตรฉบับนี้สามารถสำเร็จลุล่วงไปได้ด้วยดี ทั้งนี้เนื่องจาก  
 ดร. สมชาติ จิริวิภากร อาจารย์ที่ปรึกษาโครงการ ที่ได้ให้ความช่วยเหลือมาโดยตลอด  
 คุณพ่อและคุณแม่ ที่ได้ให้กำลังใจแก่ลูกเสมอมา  
 คุณกิติโชติ อุดมประเสริฐกุล ที่ได้แนะนำแนวทางในการทำโครงการและเอื้อเฟื้อข้อมูล  
 คุณทศพล ธนะไพบูลย์ ที่ได้ให้คำแนะนำและคำปรึกษาในเรื่องระบบฐานข้อมูล  
 เพื่อนๆ ที่คอยเป็นกำลังใจ  
 สถาบันกรมอุตุนิคมวิทยา จังหวัดชลบุรี ที่ได้เอื้อเฟื้อข้อมูล  
 สถานีไฟฟ้าช้อยนิคมอมตะนคร จังหวัดชลบุรี ที่ได้เอื้อเฟื้อข้อมูล  
 ซึ่งทางคณะผู้จัดทำขอขอบคุณมา ณ. ที่นี้

คณะผู้จัดทำ

## บรรณานุกรม

- [1] S. Jiriwibhakorn, "Transient stability Assessment of Power Systems Using Neural Networks," Ph.D. Thesis 2001, Dept. of Electrical and Electronic Engineering, Imperial College, University of London.
- [2] Kevin Warwick, Raj Aggarwal and Arthur Ekwue, "Artificial Intelligence Techniques", IEEE Power Engineering Series 22, London; Institution of Electrical Engineers, 1997
- [3] M.V. Deshpande, "Element of Electrical Power Station Design", Wheeler Publishing, 1986
- [4] H. Demuth and M. Beale, Neural Network Toolbox User's Guide, The Math Works, Inc., June 2001.
- [5] กิตติโชติ อุดมประเสริฐกุลและภุชงค์ อุปัญญา, " การทำนายนการใช้กำลังไฟฟ้าของประเทศไทยโดยการใช้โครงข่ายประสาทเทียม ", ปรินญานิพนธ์ คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง, 2541
- [6] เฉลิมพล ทัพซาย, "การใช้งาน Access 97", บริษัท 3495 บุกเซ็นเตอร์ จำกัด, 2541