ห้องสมุดคณะเทคโนโลยีสารสนเทศ สจล.

การจำแนกลักษณะลูกค้า
โดยเทคนิค นิวรอลเน็ตเวิร์ค

Client Classification by Neural Network Technique

โดย

ชัยวุฒิ โฆษิตจินดา

รหัส 42067069

H001751

อาจารย์ที่ปรึกษา

ดร. อาริต ธรรมโน

วัน เดือน ปี 10 ม.ค. 2550
เลขทะเบียน 01751
เลขเรียกหนังสือ วพ. ช 436ก 2543
"ห้องสมุดคณะเทคโนโลยีสารสนเทศ สจล."

รายงานนี้เป็นส่วนหนึ่งของวิชาโครงการพัฒนาระบบงาน
หลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ
ภาคเรียนที่ 2 ปีการศึกษา 2543
คณะเทคโนโลยีสารสนเทศ
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อ	การจำแนกลักษณะลูกค้า โดยใช้เทคนิคนิวรอลเน็ทเวิร์ค
นักศึกษา	นายชัยวุฒิ โฆษิตจินดา
อาจารย์ที่ปรึกษา	ดร.อาริต ธรรมโน
ระดับการศึกษา	วิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ
แขนงวิชา	วิทยาการสารสนเทศ
ปีการศึกษา	2543

บทคัดย่อ

ในปัจจุบันแบบจำลองนิวรอลเน็ทเวิร์คได้นำมาใช้พัฒนาการวิเคราะห์จำแนกเหตุการณ์ต่างๆ มากขึ้น ซึ่งแบบจำลองนิวรอลเน็ทเวิร์คนี้เป็นเทคโนโลยีที่มีความสามารถเรียนรู้ได้ด้วยตัวของระบบเอง ในบทความนี้กล่าวถึงการจำลองการทำงานโครงข่ายแบบจำลองนิวรอลเน็ทเวิร์คโดยวิธีการแพร่ย้อนกลับซึ่งเป็นวิธีที่สำคัญและนิยมใช้กันมากที่สุด เพื่อใช้ในธุรกิจประกันภัย

โครงการนี้จะประกอบไปด้วย การจำลองการทำงานของแบบจำลองนิวรอลเน็ทเวิร์ค ด้วยโปรแกรมคอมพิวเตอร์ และแสดงผลการทำงานที่ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Title	Client Classification by Neural Network Technique
Student	Mr. Chaivut Kositchinda
Advisor	Dr. Arit Thammano
Level of Study	Master of Science in Information Technology
Major	Information Science
Academic Year	2000

ABSTRACT

Today, the artificial neural network has been applied and used in many industries. Neural network model can learn to calculate output by itself. This article will focus on Back-Propagation Technique, the most popular technique, in applying to Insurance Industry.

This project will show the development of Neural Network Application that can classify a group of client, and report for the result.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ขอขอบพระคุณท่านอาจารย์ ดร.อาริต ธรรมโน ที่ได้กรุณาให้คำปรึกษา และให้แนวทางในการทำโครงงาน

ขอขอบคุณ คุณพ่อ-คุณแม่-ภรรยา และน้องสาว ที่คอยให้กำลังใจ และเป็นแรงผลักดันให้โครงงานนี้สำเร็จลงได้

ชัยวุฒิ โฆษิตจินดา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

หน้า

บทคัดย่อภาษาไทย I

บทคัดย่อภาษาอังกฤษ II

กิตติกรรมประกาศ III

สารบัญ IV

สารบัญตาราง VI

สารบัญภาพ VII

บทที่

1. บทนำ 1

    1.1 ความเป็นมา 1

    1.2 วัตถุประสงค์ของการศึกษา 2

    1.3 ขอบเขตของการศึกษา 2

    1.4 ประโยชน์ที่คาดว่าจะได้รับ 2

2. ทฤษฎีที่เกี่ยวข้อง 3

    2.1 การทำงานของเซลล์ประสาท 5

    2.2 สถาปัตยกรรมของโครงข่ายประสาทเทียม 6

    2.3 ทฤษฎีหลักการแพร่ย้อนกลับ 8

    2.4 สถาปัตยกรรมของโครงข่าย บี พี เอ็น 9

    2.5 การฝึกหัดโครงข่ายด้วยวิธีการแพร่ย้อนกลับ 10

    2.6 การกำหนดจำนวนชั้นซ่อน 16

    2.7 การประกันภัยรถยนต์ 16

3. วิธีดำเนินการศึกษา 18

    3.1 ส่วนของโครงสร้างแบบจำลอง 18

    3.2 ส่วนของอัลกอริธึม 19

    3.3 ส่วนของฐานข้อมูล 21

    3.4 ส่วนของโปรแกรม 22

4. สรุปผลการศึกษา และข้อเสนอแนะ 25

    4.1 สรุปผลการศึกษา 25

    4.2 ข้อเสนอแนะ 26

เอกสารนี้มีไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม 27
ภาคผนวก 28
ประวัติผู้เขียน 42

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

หน้า

ตารางที่ 2.1 แสดงแบบจำลองโครงข่ายประสาทเทียมในช่วงปี ค.ศ. 1950-1986 4
ตารางที่ 3.1 ตารางแสดงฟิลด์ของฐานข้อมูล 21

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ

หน้า

ภาพที่ 2.1 โครงสร้างและส่วนประกอบของเซลล์ประสาทโดยทั่วไป 5
ภาพที่ 2.2 สถาปัตยกรรมพื้นฐานของโครงข่ายประสาทเทียม 7
ภาพที่ 2.3 แสดงการปรับแต่งค่าน้ำหนักที่อินพุทของแต่ละนิวรอน 9
ภาพที่ 2.4 แสดงโครงข่ายบีพีเอ็น 10
ภาพที่ 2.5 แสดงโครงข่ายนิวรอน แบบแบคพรอพพาเกชั่น 11
ภาพที่ 3.1 แสดงโครงสร้างแบบจำลองโครงข่ายประสาทเทียม ที่ใช้ในโครงงานนี้ 19
ภาพที่ 3.2 แสดงหน้าจอหลัก 22
ภาพที่ 3.3 แสดงกราฟเปรียบเทียบ ค่าจริงกับค่าที่ได้จากการคำนวณ 24
ภาพที่ 4.1 ผลการคำนวณ 25

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ความเป็นมา

บริษัทประกันภัย เป็นองค์กรที่อยู่ภายใต้การควบคุมของกรมการประกันภัย กระทรวงพาณิชย์ แต่มีสภาพเป็นสถาบันการเงินประเภทหนึ่ง (ในส่วนประกันชีวิต ถือเป็นระบบเงินออมระยะยาว)

รายได้ขององค์กรมาจากธุรกิจประกันภัยด้านต่างๆ

- การประกันชีวิต
- การประกันภัยทางทะเล
- การประกันอัคคีภัย
- การประกันวินาศภัย
- การประกันภัยรถยนต์
- การประกันภัยเบ็ดเตล็ด

ในการศึกษานี้จะกล่าวถึง การตัดสินใจรับประกันภัยรถยนต์ที่ปัจจุบันยังไม่มีกฎเกณฑ์ที่แน่นอน เป็นผลให้เกิดความเสี่ยงในธุรกิจ จึงมีแนวความคิดที่จะพัฒนาระบบที่สามารถช่วยสนับสนุนการตัดสินใจโดยอาศัยข้อมูลจากฐานข้อมูลที่มีอยู่ (Data Mining) แล้วพิจารณาความเสี่ยงโดยอิงจากข้อมูลที่มีอยู่ เพื่อลดความเสี่ยงให้กับธุรกิจต่อไป

1.2 วัตถุประสงค์

1. เพื่อศึกษาและทำความเข้าใจ เกี่ยวกับการทำ Data Mining
2. เพื่อศึกษาและทำความเข้าใจ เกี่ยวกับการทำ Classification โดยใช้โครงข่ายประสาทเทียม (Neural Network)
3. เพื่อศึกษาและทำความเข้าใจ เกี่ยวกับทฤษฎีโครงข่ายประสาทเทียม แบบแบคพรอพเกชั่น (Back-Propagation Neural Network)
4. เพื่อศึกษาถึงแนวทางและความเป็นไปได้ ในการนำโครงข่ายประสาทเทียมมาประยุกต์ใช้กับ ธุรกิจประกันภัยรถยนต์
5. เพื่อเป็นแนวทางในการประยุกต์ใช้ ในการสนับสนุนการตัดสินใจรับประกันภัยรถยนต์
6. เพื่อเป็นแนวทางในการประยุกต์ใช้ ในการสนับสนุนการวางแผนทางธุรกิจต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.3 ขอบเขตของการศึกษา

การทำ Data Mining โดยวิธี Classification บนเทคนิค Neural Network ของโครงการนี้ ได้ทำการศึกษาโดยใช้ฐานข้อมูลประกันภัยจริงขององค์กร ซึ่งจัดเก็บอยู่บนระบบ IBM AS/400 แต่ในการศึกษานั้นไม่สะดวกที่จะใช้งานข้อมูล จึงได้คัดลอกข้อมูลเข้าสู่ Microsoft Access บนเครื่อง Personal Computer แทน โดยข้อมูลจะถูกแบ่งออกเป็น 2 ส่วน ข้อมูลส่วนแรกใช้สำหรับปรับค่าน้ำหนักในการฝึกสอน และข้อมูลอีกส่วนใช้ในการเปรียบเทียบค่าที่ได้จากระบบ เพื่อทดสอบดูความถูกต้องแม่นยำของโครงข่ายประสาทเทียม

1.4 ประโยชน์ที่คาดว่าจะได้รับจากโครงการ

1. เข้าใจในหลักการของ Data Mining แบบ Classification
2. เข้าใจการทำ Classification โดยใช้โครงข่ายประสาทเทียม (Neural Network)
3. เข้าใจในหลักการทฤษฎีโครงข่ายประสาทเทียม แบบแบคพรอพเกชั่น (Back-Propagation Neural Network)
4. สามารถช่วยลดความเสี่ยงในธุรกิจ
5. เพิ่มประสิทธิภาพในการทำงานขององค์กร
6. เป็นต้นแบบในการประยุกต์และพัฒนา การประมวลผลที่ครอบคลุมปัจจัยต่างๆ มากกว่านี้
7. เป็นข้อมูลให้ผู้บริหาร ในการวางแผนงานด้านธุรกิจต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎีที่เกี่ยวข้อง

ศาสตร์ทางด้านโครงข่ายประสาทเทียมได้เริ่มมีการคิดค้นวิจัย และพัฒนากันมาตั้งแต่ปี ค.ศ.1940 โดยจะขอลำดับเหตุการณ์ในช่วงเวลาที่ก่อให้เกิดสถาปัตยกรรมของโครงข่ายประสาทเทียมชนิดต่างๆ ดังนี้

ในปี ค.ศ. 1943 : วอร์เรน แมคคูลล็อค (Warren McCulloch) และวอลเตอร์ พิทท์ (Walter Pitts) ได้ออกแบบนิวรอนที่ถือเป็นต้นกำเนิด และเป็นพื้นฐานของโครงข่ายประสาทเทียมในปัจจุบัน โครงข่ายของแมคคูลล็อคและพิทท์ ประกอบด้วยชั้นของโครงข่ายจำนวน 2 ชั้น คือ ชั้นอินพุท และชั้นเอาต์พุท โครงข่ายจะมีการเชื่อมโยงจากชั้นอินพุทไปยังชั้นเอาต์พุท โดยผ่านค่าน้ำหนัก (Weight) ซึ่งค่าน้ำหนักจะถูกกำหนดค่าไว้ตายตัว การทำงานเบื้องต้นของโครงข่าย คือนำไปใช้จำลองฟังก์ชั่นทางลอจิกพื้นฐาน เช่น ฟังก์ชั่นลอจิก AND ฟังก์ชั่นลอจิก OR ฟังก์ชั่นลอจิก NOT และฟังก์ชั่นลอจิก XOR โดยนำเสนอการกำหนดค่าน้ำหนักเพื่อให้โครงข่ายทำงานได้กับการจำลองตามฟังก์ชั่นดังกล่าว

ในปี ค.ศ. 1949 : โดแนลด์ เฮบบ์ (Donald Hebb) ได้ออกแบบกฎการเรียนรู้สำหรับโครงข่ายประสาทเทียมเป็นคนแรก กฎการเรียนรู้นี้ก็คือ ขั้นตอนของกระบวนการปรับแต่งค่าน้ำหนักที่เชื่อมโยงจากชั้นอินพุทไปยังชั้นเอาต์พุท เพื่อทำให้เกิดค่าเอาต์พุทที่ต้องการ ซึ่งเป็นแนวทางในการพัฒนากฎการเรียนรู้รูปแบบอื่นๆ ตามมา

ในปี ค.ศ.1957 : แฟรงค์ โรเซนบลัทท์ (Frank Rosenblatt) ได้นำเสนอรูปแบบที่สำคัญรูปแบบหนึ่งของโครงข่ายประสาทเทียมนั่นก็คือ รูปแบบที่เรียกว่า เพอร์เซพตรอน เป็นสถาปัตยกรรมอันประกอบไปด้วย ชั้นอินพุทต่อผ่านน้ำหนักไปยังชั้นนิวรอนที่ชั้นเอาต์พุท และใช้กฎการเรียนรู้ของเพอร์เซพตรอนในการปรับแต่งค่าน้ำหนักในการเชื่อมโยง ที่มีประสิทธิภาพดีกว่ากฎการเรียนรู้ของแฮบบ์

ในปี ค.ศ. 1960 : เบอร์นาร์ด วิดโรว์ (Bernard Widrow) และมาเซียน ฮอฟฟ์ (Marcian Hoff) ได้พัฒนากฎการเรียนรู้ ซึ่งใกล้เคียงกับการเรียนรู้ของเพอร์เซพตรอน และนำกฎนี้ไปใช้กับระบบหรือรูปแบบที่เรียกว่า อะดาไลน์ (ADALINE – Adaptive Liner Neuron) และรูปแบบมาดาไลน์ (MADALINE – Multi Layer ADALINE) ซึ่งเป็นรูปแบบที่พัฒนาจาก อะดาไลน์ โดยวิดโรว์เช่นกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในปี ค.ศ. 1972 : ทูโว คาโคเนน (Teuvo Kohonen) ได้เริ่มพัฒนาการเรียนรู้แบบจัดตัวเอง (Self–Organized) สถาปัตยกรรมของโครงข่ายคาโคเนนนี้ประกอบด้วยจำนวนชั้น 2 ชั้น คือ ชั้นอินพุท และชั้นเอาต์พุท การเชื่อมโยงที่ชั้นอินพุทไปยังนิวรอนที่ชั้นเอาต์พุทนั้นจะต่อผ่านน้ำหนัก ข้อแตกต่างจากโครงข่ายอื่นๆ คือ กระบวนการเรียนรู้ที่สามารถจัดกลุ่มข้อมูลอินพุทได้ด้วยตัวของมันเอง และเรียกการเรียนรู้นี้ว่า การเรียนรู้แบบแข่งขัน (Competitive Learning)

หลังจากปี ค.ศ. 1980 : เป็นต้นมาได้มีการพัฒนารูปแบบของโครงข่ายประสาทเทียมชนิดต่างๆ เช่น รูปแบบโครงข่ายจัดตัวเองที่เรียกว่า ART คิดค้นโดย สตีเฟน กรอสเบิร์ก (Stephen Grossberg) และเกียล คาร์เพนเตอร์ (Gail Carpenter) รูปแบบโครงข่ายฮอปฟิลด์ คิดค้นโดย จอน ฮอปฟิลด์ (John Hopfield) รูปแบบโครงข่ายนีโอคอกนิตรอน คิดค้นโดย คูนิฮิโกะ ฟูกูชิมา (Kunihiko Fukushima) รูปแบบโครงข่ายโบลสแมนน์ แมชชีน (Boltzmann machine) คิดค้นโดยนักวิจัยหลายคน และรูปแบบที่นิยมใช้กันมากที่สุดคือ รูปแบบโครงข่ายแบบแพร่ย้อนกลับ (Back-Propagation) คิดค้นโดย เดวิด ปาร์คเคอร์ (David Parker) และเดวิด รูเมลฮาร์ท (David Rumelhard) ในบทนี้จะกล่าวถึงรูปแบบแพร่ย้อนกลับเท่านั้น

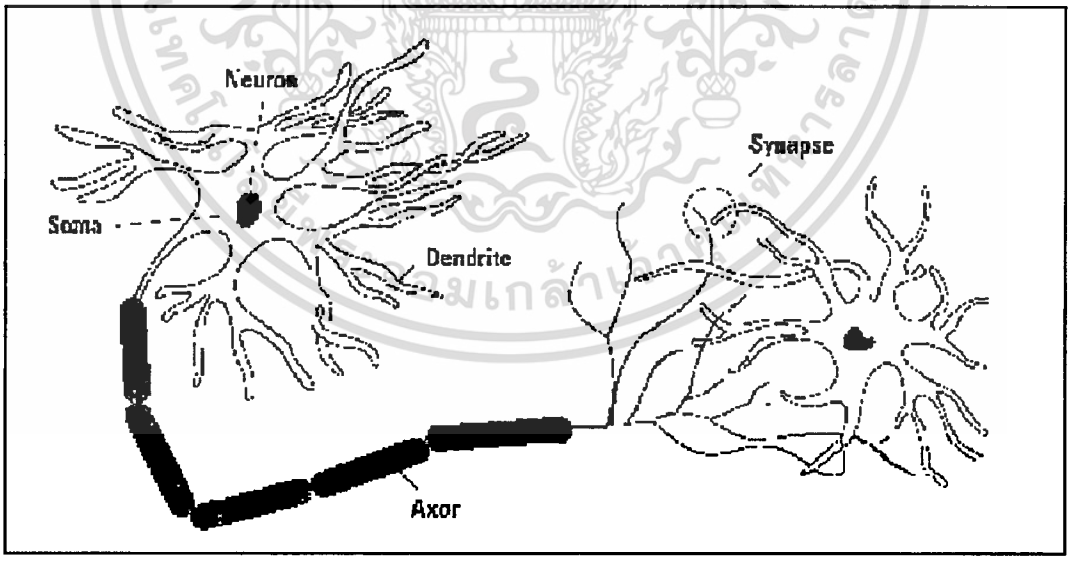
ตารางที่ 2.1 แสดงแบบจำลองโครงข่ายประสาทเทียมในช่วงปี ค.ศ. 1950-1986

ชื่อแบบจำลอง	ผู้คิดค้น	ปีที่พัฒนา	การใช้งาน
1. Perceptron	F. Rosenblatt	1957	อ่านตัวพิมพ์ดีด
2. ADALINE และ MADALINE	B. Widrow	1960-1962	Adaptive Equalizer
3. Avalanche	S. Grossberg	1967	เข้าใจคำพูด
4. Cerebellatron	D. Mar, J. Albus และ A. Pellionez	1969-1982	ควบคุมการเคลื่อนไหวของแขนหุ่นยนต์
5. Back Propagation	P. Werbos, D. Parker และ D. umelhart	1974-1985	เลียนแบบฟังก์ชั่นทางคณิตศาสตร์, อ่านตัวเขียน, คำพูด, ทำนายหุ้น
6. Brain-State-in-a-Box	J. Anderson	1977	สกัดข้อมูลบางอย่างจากฐานข้อมูล
7. Neocognitron	K. Fukushima	1978-1984	อ่านตัวเลข และตัวเขียน
8. Adaptive Resonance	G. Carpenter, S. Grossberg	1978-1986	Pattern Recognition

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

9. Self-Organizing Map	T. Kohonen	1980	จำลองลักษณะพื้นผิวของวัตถุ เช่น ปีกเครื่องบิน
10. Hopfield	J. Hopfield	1982	ทำนายส่วนของข้อมูลที่หายไป เช่น บางส่วนของหน้าคน, Optimization
11. Bidirectional Associative Memory	A. Kosko	1985	Content-Addressable Associative Memory
12. Boltzmann และ Cauchy Machine	J. Hinton, T. Sejnowsky และ H. Zu	1985-1986	Pattern Recognition
13. Counterpropagation	R. Hencht-Nielsen	1986	อัดข้อมูลของภาพให้น้อยลง, Table-Lookup

2.1 การทำงานของเซลล์ประสาท (Biological Neurons)



ภาพที่ 2.1 โครงสร้างและส่วนประกอบของเซลล์ประสาทโดยทั่วไป

ในธรรมชาติ เซลล์ประสาทของสิ่งมีชีวิตมีหลายประเภทแล้วแต่ตำแหน่งและหน้าที่ของเซลล์ เซลล์ประสาทในตัวของคนเราก็เช่นกัน มีอยู่หลายประเภทตามตำแหน่งและหน้าที่ของเซลล์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เช่นเซลล์ประสาทของกล้ามเนื้อ เซลล์ประสาทในสมอง เซลล์ประสาทที่ลิ้น และเซลล์ประสาทที่จมูก เป็นต้น โดยการส่งผ่านข้อมูลในระบบประสาทจะใช้เวลาประมาณ 10^{-3} วินาที ซึ่งนับว่าช้ามากเมื่อเทียบกับระบบคอมพิวเตอร์ที่ใช้เวลาเพียง 10^{-10} วินาที โครงสร้างและส่วนประกอบของเซลล์ประสาทโดยทั่วๆ ไป แสดงได้ดังภาพที่ 2.1

เซลล์ประสาทประกอบด้วยส่วนใหญ่ๆ 3 ส่วน คือ ตัวเซลล์ประสาท (โซมา หรือ นิวรอน) ซึ่งมีนิวเคลียสอยู่ตรงกลาง รอบๆ ตัวเซลล์ประสาทจะมีกิ่งเล็กๆ ที่ยื่นออกไป เพื่อใช้ปลายกิ่งเหล่านี้รับสัญญาณจากเซลล์ประสาทอื่นๆ เรียกว่า เดนไดรท์ (Dendrite) และจะมีส่วนที่ใช้ส่งสัญญาณกับเซลล์ประสาทอื่นๆ เรียกว่า แอคซอน (Axon) ช่องว่างระหว่างก้านของเซลล์ประสาทที่ต่างกันเรียกว่า ไซแนปส์ (Synapse) ซึ่งสามารถเปลี่ยนค่าความต้านทานได้ตามสัญญาณที่ส่งระหว่างกันของเซลล์ประสาท การส่งสัญญาณระหว่างเซลล์ประสาททำได้โดยการถ่ายเทสารประกอบโซเดียม โปรตัสเซียม และคลอไรด์

ฮอดกิ้น (Hodkin) และฮักลีย์ (Huxley) ซึ่งได้รับรางวัลโนเบลทางชีววิทยา ได้ค้นพบว่าการไหลของสารประกอบโซเดียมและโพแทสเซียมของเซลล์ประสาทของปลาหมึก ได้ทำให้เกิดความต่างศักย์ จะอยู่ระหว่าง 50 ถึง 70 มิลลิโวลต์ จากผลการศึกษาดังกล่าวทำให้เราสามารถจำลองการทำงานของเซลล์ประสาท โดยอาศัยวงจรอิเลคทรอนิกส์ได้

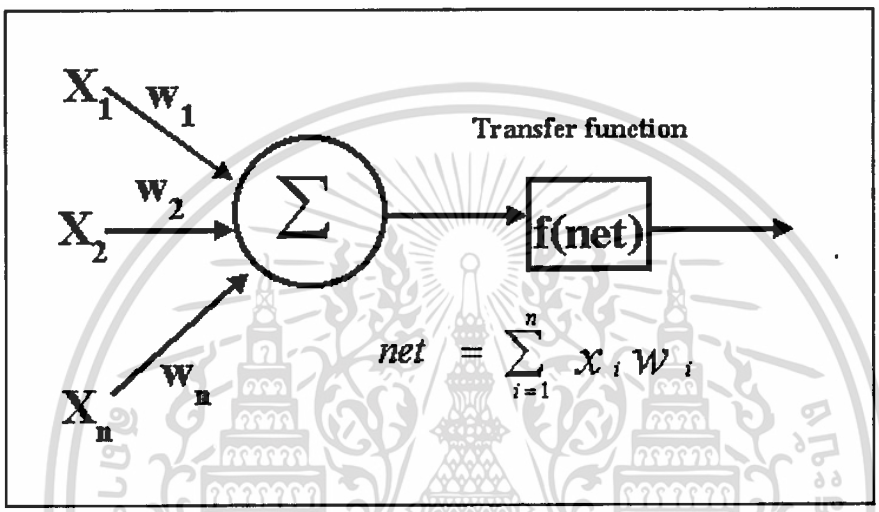
โครงสร้างของโครงข่ายประสาทเทียมมีลักษณะที่คล้ายคลึงกับสมอง และระบบประสาทซึ่งประกอบด้วยส่วนของการประมวลผลต่างๆ ที่เรียกว่า นิวรอน (Neuron) ทุกๆ นิวรอน สามารถมีอินพุทได้หลายอินพุท แต่มีเอาต์พุทเดียว และทุกๆ เอาต์พุทจะแยกไปยังอินพุทของนิวรอนอื่นๆ ภายในโครงข่าย การติดต่อกันภายในระหว่างนิวรอน ไม่ใช่ลักษณะของการต่อแบบธรรมดาทุกๆ อินพุทจะมีน้ำหนักเป็นตัวกำลังของการติดต่อภายใน และช่วยในการตัดสินใจ การทำงานของนิวรอนในบางโครงข่ายจะถูกกำหนดไว้ตายตัว แต่บางโครงข่ายสามารถที่จะปรับแต่งได้ ซึ่งอาจจะเป็นการปรับแต่งจากภายนอกโครงข่าย หรือนิวรอนสามารถปรับได้ด้วยตัวของมันเอง ในจุดนี้เองแสดงถึงความสามารถในการเรียนรู้ และจดจำของโครงข่ายประสาทเทียม

2.2 สถาปัตยกรรมพื้นฐานของโครงข่ายประสาทเทียม

จากลักษณะและการทำงานของเซลล์ประสาทหรือนิวรอนดังที่กล่าวมาข้างต้น ได้ถูกนำมาสร้างเป็นทฤษฎีทางคณิตศาสตร์และจำลองการทำงานในรูปแบบพื้นฐาน โดยใช้ชื่อว่า นิวรอนเทียม หรือ AN (Artificial Neuron) ซึ่งมีสถาปัตยกรรมพื้นฐานแสดงดังภาพที่ 2.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากภาพที่ 2.2 โครงข่ายประกอบด้วยชั้นของโครงข่ายจำนวน 2 ชั้น คือชั้นอินพุท และชั้นเอาต์พุท ที่ชั้นอินพุทจะมีหน่วยอินพุทจำนวน n หน่วยที่จะคอยรับสัญญาณอินพุท x_1 ถึง x_n ไปยังหน่วยเอาต์พุท ส่วนที่ชั้นเอาต์พุทจะมีหน่วยเอาต์พุท หรือหน่วยประมวลผลเพียงหน่วยเดียว



ภาพที่ 2.2 สถาปัตยกรรมพื้นฐานของโครงข่ายประสาทเทียม

ความสามารถในการจัดกลุ่ม

ความสามารถในการจัดกลุ่มของ AN จะจัดกลุ่มข้อมูลอินพุทในลักษณะเชิงเส้นเดี่ยวเท่านั้น ที่เป็นเช่นนี้ เพราะสมการที่ใช้คำนวณเป็นสมการเชิงเส้น พิจารณาภาพที่ 2.2 จะได้สมการคือ

$$x_0w_0 + x_1w_1 + x_2w_2 + x_3w_3 + \ldots + x_nw_n$$

จะเห็นว่าที่เทอม x_0w_0 เป็นเทอมที่นำมาใช้บอกถึงการตั้งค่าเทรสโฮลด์ของระบบ ซึ่งเรียกเทอมนี้ว่า เทอมไบแอส โดยกำหนดให้ x_0 มีค่าคงที่เท่ากับ 1 ดังนั้นจึงเขียนสมการใหม่ได้

$$w_0 + x_1w_1 + x_2w_2 + x_3w_3 + \ldots + x_nw_n$$

ลักษณะเช่นนี้ทำให้การคำนวณค่าเอาต์พุทของ AN ที่มีเทอมไบแอสได้ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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

เมื่อ

$$\text{Net} = \sum_{i=1}^{n} x_i w_i + w_o$$

ถ้าตัดเทอมไบแอสออกไปนั่นหมายถึง จะต้องมีการกำหนดค่าเทรสโฮลด์ (T) ภายนอกเพื่อใช้ตัดสินการคำนวณค่าทางเอาต์พุท แสดงได้โดยสมการ

$$x_1w_1 + x_2w_2 + x_3w_3 + \ldots + x_nw_n$$

การคำนวณค่าทางเอาต์พุทของ AN ที่ไม่มีเทอมไบแอสแสดงได้ดังนี้

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

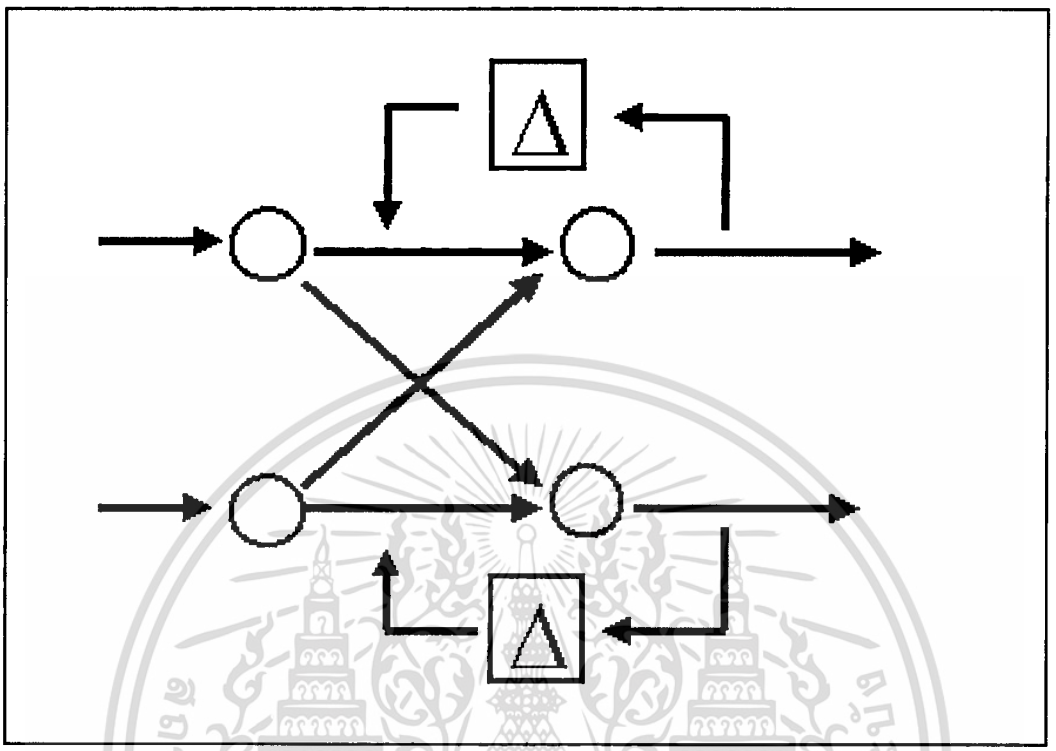
เมื่อ

$$\text{Net} = \sum_{i=1}^{n} x_i w_i$$

2.3 ทฤษฎีหลักการแพร่ย้อนกลับ

หลักการแพร่ย้อนกลับได้รับการพัฒนาโดย รูเมลฮาร์ท และถูกนำเสนอในช่วงปี ค.ศ. 1986 หลักการแพร่ย้อนกลับนี้สามารถนำไปใช้แก้ปัญหาในลักษณะเชิงเส้น (Linear) และปัญหาที่ไม่เชิงเส้นได้ (Non-Linear) ทำให้ประยุกต์เข้ากับโครงงานต่างๆ ได้อย่างหลากหลาย และได้รับความนิยมมากที่สุดในปัจจุบัน หลักการแพร่ย้อนกลับมีพื้นฐานมาจากกฎเดลต้า (Delta-Rule) ที่พัฒนาขึ้นโดยวิดโรว์ และฮอฟฟ์ ในช่วงปี ค.ศ. 1960-1962 ซึ่งกระบวนการของกฎเดลต้าจะลดค่าผิดพลาดที่เอาต์พุทของโครงข่ายเปรียบเทียบกับค่าเป้าหมาย โดยค่าผิดพลาดที่ลดลงนี้เกิดจากการปรับแต่งค่าน้ำหนักที่อินพุทของแต่ละนิวรอน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

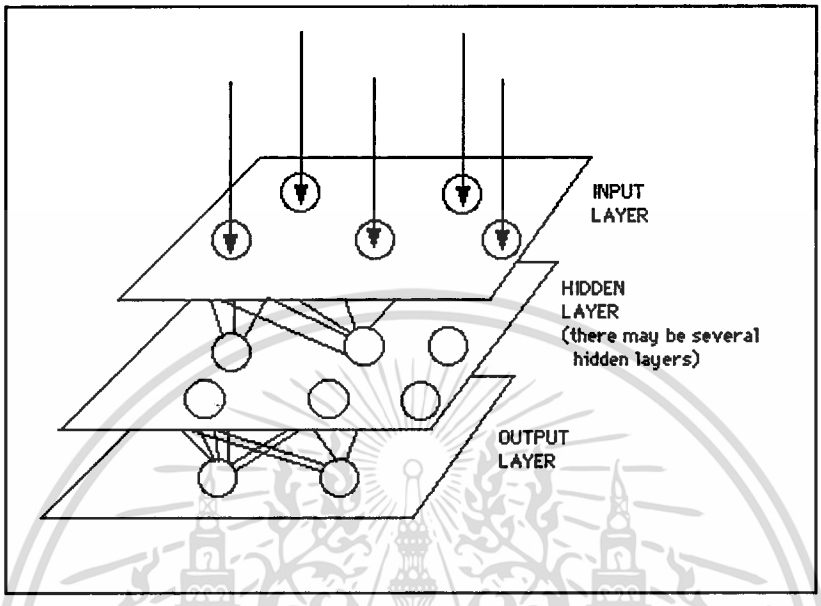


ภาพที่ 2.3 แสดงการปรับแต่งค่าน้ำหนักที่อินพุทของแต่ละนิวรอน

2.4 สถาปัตยกรรมของโครงข่าย บี พี เอ็น

โครงข่ายบีพีเอ็น (BPN, Back-Propagation Neural Network) เป็นโครงข่ายที่มีจำนวนชั้นมากกว่า 2 ชั้น คือมีได้ตั้งแต่ 3 ชั้นขึ้นไปคือ ชั้นอินพุท (Input layer) ชั้นซ่อน (Hidden layer) และชั้นเอาต์พุท (Output layer) จากภาพที่ 2.4 แสดงให้เห็นถึงลักษณะของโครงข่ายบีพีเอ็น ซึ่งเป็นฟีดฟอร์เวิร์ดเน็ทเวิร์ค (Feedforword Network) โดยมีการเชื่อมโยงในแต่ละชั้นแบบฟูลลี่ นั่นคือทุกๆ นิวรอนในชั้นอินพุทจะส่งสัญญาณไปยังทุกๆ นิวรอนในชั้นซ่อนชั้นแรก และทุกๆ นิวรอนในชั้นซ่อนชั้นแรกจะส่งสัญญาณไปยังทุกๆ นิวรอนในชั้นซ่อนถัดไปจนในที่สุดทุกๆ นิวรอนในชั้นซ่อนชั้นสุดท้ายจะส่งสัญญาณไปยังทุกๆ นิวรอนในชั้นเอาต์พุท จากภาพแสดงโครงข่ายที่มี 3 ชั้น คือ ชั้นอินพุท ชั้นซ่อน และชั้นเอาต์พุท ในทางปฏิบัติที่ชั้นซ่อนสามารถที่จะมีได้มากกว่า 1 ชั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 2.4 แสดงโครงข่ายบีพีเอ็น

โครงข่ายประสาทเทียมเรียกได้ว่าเป็นแมพปิ้งเน็ทเวิร์ค (Mapping Network) ถ้าหากโครงข่ายสามารถคำนวณฟังก์ชั่นในบางลักษณะ ที่กำเนิดความสัมพันธ์ระหว่างอินพุท และเอาต์พุทในโครงข่าย ตัวอย่างเช่น ถ้ามีอินพุทป้อนให้กับโครงข่ายซึ่งมีค่าเป็นมุม (Angle) และเอาต์พุทคือค่าโคซายน์ (COSINE) ของมุมที่อินพุท โครงข่ายจะมีการแมพปิ้งคือ 0 → COS (0) สิ่งสำคัญที่ทำให้การแมพปิ้งถูกต้องนั้นจำเป็นต้องมีการฝึกหัดโครงข่าย และการฝึกหัดโครงข่ายโดยอาศัยหลักการแพร่ย้อนกลับนั้น แบ่งออกได้เป็นสามขั้นตอนคือ

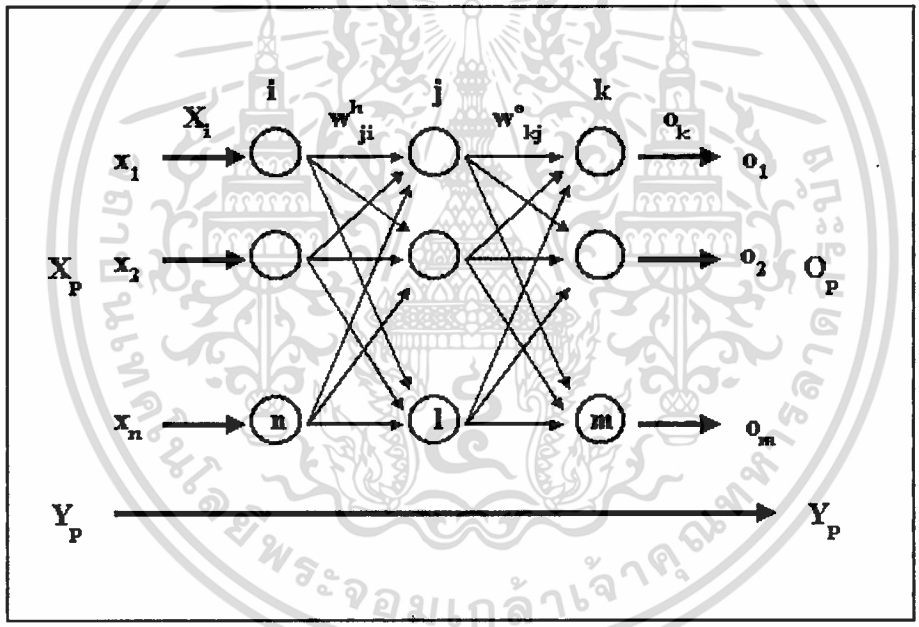
1. แพร่สัญญาณที่ชั้นอินพุทไปยังชั้นซ่อนแรก และแพร่สัญญาณไปยังชั้นซ่อนถัดไปจนถึงชั้นเอาต์พุท โดยผ่านฟังก์ชั่นกระตุ้น
2. คำนวณค่าผิดพลาดของค่าเอาต์พุทกับค่าเป้าหมาย และแพร่ย้อนกลับค่าผิดพลาดเหล่านี้ไปยังชั้นซ่อน
3. ปรับแต่งค่าน้ำหนักที่ชั้นเอาต์พุทและชั้นซ่อนเพื่อลดค่าผิดพลาด โดยทำให้ค่าเอาต์พุทมีค่าใกล้เคียง หรือเท่ากับค่าเป้าหมาย

2.5 การฝึกหัดโครงข่ายด้วยวิธีการแพร่ย้อนกลับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เริ่มต้นด้วยการพิจารณาภาพที่ 2.5 ซึ่งเป็นโครงข่ายขนาด 3 ชั้น ที่มีอินพุทเวกเตอร์ (Pattern) $X_p = (x_{p1}, x_{p2}, \ldots, x_{pn})$ อินพุท X_p นี้ถูกป้อนให้กับชั้นอินพุทของโครงข่าย และที่ชั้นอินพุทนี้จะกระจายค่าไปยังนิวรอนในชั้นซ่อน ที่ชั้นซ่อนนี้เองจะทำการคำนวณค่าเน็ท สมการหาค่าเน็ทที่นิวรอน j ในชั้นซ่อนคือ

$$Net^h_j = \sum_{i=1}^{N} x_i w^h_{ji} + \theta^h_j$$



ภาพที่ 2.5 แสดงโครงข่ายนิวรอน แบบแบคพรอพพาเกชั่น

เมื่อ w^h_{ji} คือน้ำหนักที่ชั้นซ่อนตรงจุดต่อจากโหนดอินพุท i ไปยังนิวรอน j และ θ^h_j คือค่าเทอม ไบแอส เมื่อได้ค่าเน็ทแล้วขั้นถัดไปคือคำนวณค่าเอาต์พุทของแต่ละนิวรอนในชั้นซ่อน o_j โดยคำนวณได้ดังนี้

$$o_j = f^h_j(Net^h_j)$$

ใช้สมการทั้งสองคำนวณค่าในชั้นซ่อนทุกๆ ชั้น จากนั้นจึงทำการคำนวณค่าเน็ทและค่าเอาต์พุทที่ชั้นเอาต์พุทโดยใช้สมการที่คล้ายคลึงคือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$Net^{o}_{k} = \sum_{j=1}^{L} o_j w^{o}_{kj} + \theta^{o}_{k}$$

$$o_k = f^{o}_{k}(Net^{o}_{k})$$

ส่วนฟังก์ชั่นไบนารีซิกมอยด์กำหนดได้ดังนี้

$$f(Net_j) = 1 / (1 + Exp(-Net))$$

การหาค่าผิดพลาดและการปรับน้ำหนักที่ชั้นเอาต์พุท

สมการปรับค่าน้ำหนักที่ชั้นเอาต์พุท แสดงได้ดังนี้

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

$$\Delta w^{o}_{kj} = \eta (y_k - o_k) f'^{o}_{k}(Net^{o}_{k}) o_j$$

$$o_j = f^{h}_{j}(Net^{h}_{j})$$

$$f^{h}_{j}(Net^{h}_{j}) = (1 + Exp(-Net^{h}_{j}))^{-1}$$

$$f'^{o}_{k} = f^{o}_{k}(1 - f^{o}_{k})$$

$$= o_k(1 - o_k)$$

$$w^{o}_{kj}(t+1) = w^{o}_{kj}(t) + \eta (y_k - o_k)\ o_k (1 - o_k)\ o_j$$

เขียนสมการให้ดูง่ายขึ้นได้ดังนี้

$$\sigma^{o}_{k} = (y_k - o_k) f'^{o}_{k}(Net^{o}_{k})$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$w^{o}_{kj}(t+1) = w^{o}_{kj}(t) + \eta \sigma^{o}_{k} o_j$$

การหาค่าผิดพลาดและการปรับน้ำหนักที่ชั้นซ่อน

การปรับแต่งค่าน้ำหนักที่ชั้นซ่อนจะต้องทำการคำนวณหาค่าผิดพลาดที่ชั้นซ่อน (Hidden-layer error term) ก่อนโดยใช้สมการ

$$\sigma^{h}_{j} = f'^{h}_{j}(Net^{h}_{j}) \sum \sigma^{o}_{k} w^{o}_{kj}$$

จากนั้นจึงสามารถปรับแต่งค่าน้ำหนัก โดยสมการที่คล้ายคลึงกับสมการปรับแต่งค่าน้ำหนักที่ชั้น เอาต์พุทคือ

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

สรุปขั้นตอนการฝึกหัดโครงข่ายประสาทเทียมโดยวิธีการแพร่ย้อนกลับ

1. กำหนดอินพุทเวคเตอร์ แก่โหนดอินพุท

$$x_p = (x_{p1}, x_{p2}, \ldots, x_{pn})$$

2. คำนวนหาค่าเน็ทอินพุทของแต่ละนิวรอนในชั้นซ่อนโดยสมการ

$$Net^{h}_{j} = \sum_{i=1}^{N} x_i w^{h}_{ji} + \theta^{h}_{j}$$

3. คำนวนค่าเอาต์พุทของแต่ละนิวรอนในชั้นซ่อน

$$o_j = f^{h}_{j}(Net^{h}_{j})$$

4. คำนวนค่าเน็ทของแต่ละนิวรอนในชั้นเอาต์พุท

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$Net^{o}_{k} = \sum_{j=1}^{L} o_j w^{o}_{kj} + \theta^{o}_{k}$$

5. คำนวณค่าเอาต์พุทของแต่ละนิวรอนในชั้นเอาต์พุท

$$o_k = f^{o}_{k}(Net^{o}_{k})$$

6. คำนวนเทอมค่าผิดพลาดของแต่ละนิวรอนในชั้นเอาต์พุท

$$\sigma^{o}_{k} = (y_k - o_k)\, f'^{o}_{k}(Net^{o}_{k})$$

7. คำนวณเทอมค่าผิดพลาดของแต่ละนิวรอนในชั้นซ่อน

$$\sigma^{h}_{j} = f'^{h}_{j}(Net^{h}_{j}) \sum \sigma^{o}_{k} w^{o}_{kj}$$

หมายเหตุ ค่าเทอมผิดพลาดของแต่ละนิวรอนในชั้นซ่อนต้องคำนวณก่อนที่จะมีการปรับแต่งน้ำหนักที่ ชั้นเอาต์พุท

8. ปรับแต่งค่าน้ำหนักที่ชั้นเอาต์พุท

$$w^{o}_{kj}(t+1) = w^{o}_{kj}(t) + \eta\, \sigma^{o}_{k}\, o_j$$

9. ปรับแต่งค่าน้ำหนักที่ชั้นซ่อน

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

และคำนวณเทอมค่าผิดพลาด (SSE – Sum Square Error)

$$SSE = \sum (y_p - o_p)^2$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซึ่งเป็นปริมาณการวัดว่าโครงข่ายเรียนรู้ได้ดีเพียงไร เมื่อค่าผิดพลาดมีค่าต่ำลงจนสามารถยอมรับได้ (Error Tolerance) สำหรับทุกๆ คู่อันดับ อินพุท-เอาต์พุท ของกลุ่มข้อมูลทดสอบการฝึกหัดนี้ก็ยุติลงได้

ฟังก์ชั่นกระตุ้น

ฟังก์ชั่นกระตุ้นสำหรับหลักการแพร่ย้อนกลับนี้มีความสำคัญมาก ที่จะเป็นตัวกำหนดคุณสมบัติให้กับ โครงข่าย ฟังก์ชั่นกระตุ้นที่นิยมใช้กันมากคือฟังก์ชั่นซิกมอยด์ได้แก่ไบนารีซิกมอยด์ (binary sigmoid) ซึ่งได้ เคยกล่าวถึงมาแล้ว ค่าเอาต์พุทที่ได้จากฟังก์ชั่นนี้อยู่ในช่วงระหว่าง 0 ถึง 1

ฟังก์ชั่นไบนารีซิกมอยด์กำหนดได้ ดังนี้

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

จะได้ฟังก์ชั่นอนุพันธ์ คือ

$$f'(x) = f(x)\,(1 - f(x))$$

ฟังก์ชั่นซิกมอยด์อีกชนิดหนึ่งที่นิยมใช้กันมากก็คือไบโพล่าซิมอยด์ (Bipolar Sigmoid) ซึ่งค่าเอาต์พุทที่ ได้จากฟังก์ชั่นอยู่ในช่วงระหว่าง -1 และ 1

ฟังก์ชั่นไบโพล่าซิกมอยด์กำหนดได้ดังนี้

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

จะได้ฟังก์ชั่นอนุพันธ์ คือ

$$f'(x) = \tfrac{1}{2}\,(1 + f(x))\,(1 - f(x))$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6 การกำหนดจำนวนชั้นซ่อน

ยังไม่มีวิธีการใดที่สามารถกำหนดจำนวนของชั้นซ่อนให้มีค่าที่แน่นอนได้ แต่ก็มีทฤษฎีที่กล่าวไว้ว่า ชั้นซ่อนเพียงชั้นเดียวก็พอเพียงแล้วสำหรับการนำไปใช้กับปัญหาที่ไม่เป็นเชิงเส้นใดๆ ก็ได้ แต่ในบางกรณีการเพิ่มชั้นซ่อนก็สามารถช่วยให้การเรียนรู้ง่ายขึ้น รวดเร็วขึ้น และอีกปัญหาหนึ่งคือในชั้นซ่อนควรจะมีนิวรอนกี่นิวรอน ซึ่งเช่นเดียวกันคือ ยังไม่มีกฎเกณฑ์ตายตัวใดๆ กำหนดได้อย่างแน่นอนได้ เนื่องจากจำนวนนิวรอนในชั้นซ่อนนี้ขึ้นอยู่กับความซับซ้อนของปัญหา มีหลายเทคนิคถูกนำเสนอออกมาเพื่อใช้หาจำนวนนิวรอนที่เหมาะสมในชั้นซ่อน ซึ่งวิธีที่นับว่าแพร่หลายพอสมควรคือ ใช้นิวรอนให้มีจำนวนมากไว้ก่อนแล้วค่อยๆ ลดจำนวนลงมา จนได้จำนวนที่น้อยที่สุดที่โครงข่ายประสาทเทียมยังคงเรียนรู้ได้ เทคนิคดังกล่าวเรียกว่า พรุนนิ่ง (Pruning)

2.7 การประกันภัยรถยนต์

ในที่นี้จะกล่าวถึง การประกันภัยรถยนต์ ที่กำหนดไว้ในพระราชบัญญัติประกันวินาศภัย พ.ศ. 2535

- ความคุ้มครองตามกรมธรรม์ประกันภัยรถยนต์
  1. ความคุ้มครองความรับผิดชอบต่อ ชีวิต ร่างกาย หรืออนามัยของบุคคลภายนอก
  2. ความคุ้มครองความรับผิดชอบต่อ ทรัพย์สินของบุคคลภายนอก
  3. ความคุ้มครองความรับผิดชอบต่อ ความเสียหายของตัวรถยนต์
  4. ความคุ้มครองความรับผิดชอบต่อ ความสูญหาย และไฟไหม้ของตัวรถยนต์

- ประเภทของกรมธรรม์
  1. กรมธรรม์ประเภท 1 คือ กรมธรรม์ประกันภัยที่มีความคุ้มครองครบทั้ง 4 ข้อ
  2. กรมธรรม์ประเภท 2 คือ กรมธรรม์ประกันภัยที่มีความคุ้มครองตามข้อ 1, 2 และ 4
  3. กรมธรรม์ประเภท 3 คือ กรมธรรม์ประกันภัยที่มีความคุ้มครองตามข้อ 1 และ 2

- ประเภทรถยนต์
  1. ประเภทรถยนต์นั่ง
  2. ประเภทรถยนต์โดยสาร
  3. ประเภทรถยนต์บรรทุก
  4. ประเภทรถยนต์ลากจูง
  5. ประเภทรถพ่วง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ห้องสมุดคณะเทคโนโลยีสารสนเทศ สจล.

6. ประเภทรถจักรยานยนต์
7. ประเภทรถยนต์นั่งรับจ้างสาธารณะ
8. ประเภทรถยนต์เบ็ดเตล็ด

- ลักษณะการใช้งานรถยนต์
  1. การใช้งานส่วนบุคคล
  2. การใช้เพื่อการพาณิชย์
  3. การใช้รับจ้างสาธารณะ
  4. การใช้เพื่อการพาณิชย์พิเศษ
  5. รถยนต์ป้ายแดง
  6. รถพยาบาล
  7. รถดับเพลิง
  8. รถใช้ในการเกษตร
  9. รถใช้ในการก่อสร้าง
  10. รถอื่นๆ

- ขนาดรถยนต์
  1. ขนาดเครื่องยนต์
  2. จำนวนที่นั่ง
  3. น้ำหนักบรรทุก

บทสรุป

ในบทนี้ได้กล่าวถึงประวัติความเป็นมาของการวิจัยศาสตร์ทางด้านโครงข่ายประสาทเทียม พร้อมทั้งแสดงทฤษฎีการทำงานของรูปแบบอะคาไลน์ มาดาไลน์ และหลักการแพร่ย้อนกลับ ซึ่งเป็นความรู้พื้นฐานของงานวิจัยนี้ ในบทที่ 3 จะแสดงแนวคิดที่จะแปลงรูปแบบทฤษฎีการทำงานในบทนี้เป็นภาษาโปรแกรมคอมพิวเตอร์ เพื่อทำให้ไมโครคอมพิวเตอร์สามารถจำลองการทำงานตามทฤษฎีเหล่านี้ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
วิธีดำเนินการศึกษา

ในการศึกษาโครงงานนี้ จะเป็นการนำทฤษฎีโครงข่ายประสาทเทียม แบบแบคพรอพเกชั่น (Back-Propagation Neural Network) มาพัฒนาเป็นโปรแกรมประยุกต์ เพื่อใช้ในการจำแนกลักษณะลูกค้า โดยแบ่งการดำเนินการศึกษาออกเป็น 6 ขั้นตอน คือ

1. ทำการศึกษาทฤษฎีที่เกี่ยวข้อง
2. ทำการรวบรวมข้อมูลประกันภัย
3. ทำการออกแบบฐานข้อมูล
4. ทำการพัฒนาโปรแกรม และทดสอบ
5. สรุปผลที่ได้จากการศึกษา

โครงงานนี้แบ่งออกเป็น 4 ส่วนใหญ่ๆ คือ

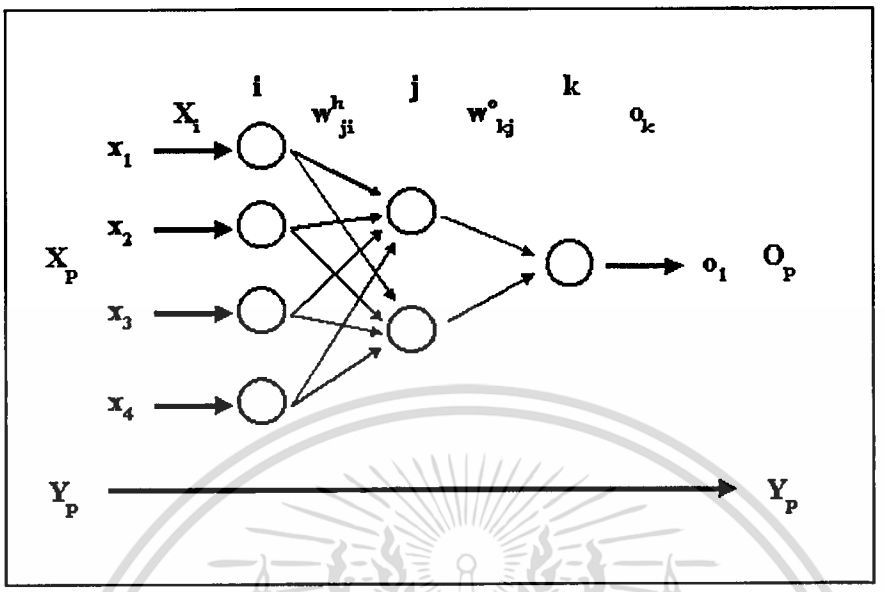
1. ส่วนของโครงสร้างแบบจำลอง
2. ส่วนของอัลกอริธึม
3. ส่วนของฐานข้อมูล
4. ส่วนของโปรแกรม

3.1 ส่วนของโครงสร้างแบบจำลอง

โครงสร้างแบบจำลองของโครงข่ายประสาทเทียมแบบแบคพรอพเกชั่น ที่ใช้ในโครงงานนี้ประกอบ 3 Layer ดังแสดงในภาพ 3.1

- ชั้นข้อมูลเข้า (Input Layer)
- ชั้นซ่อน (Hidden Layer)
- ชั้นข้อมูลออก (Output Layer)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 3.1 แสดงโครงสร้างแบบจำลองโครงข่ายประสาทเทียม ที่ใช้ในโครงงานนี้

3.2 ส่วนของอัลกอริธึม

ขั้นตอนของโปรแกรมที่ใช้ในการ Train มีดังนี้

1. อ่านค่าตัวแปรจากหน้าจอ ประกอบด้วย LearningRate, ErrorAccept, HiddenNode, MaxLoop, StartRecordTrain, TotalRecordTrain, StartRecordTest, TotalRecordTest
2. อ่านข้อมูลจากฐานข้อมูล ลง Array 2 มิติ aryTableInput, AryTableTarget
3. ทำการแปลงค่าข้อมูลใน Array ให้อยู่ในช่วง 0 ถึง 1 เก็บไว้ใน aryConvInput, aryConvTarget โดยใช้ ConvSex(), ConvAge(), ConvCareer(), ConvSumInsure (), ConvPaid()
4. กำหนดค่าน้ำหนักด้วยการสุ่มในช่วง –1 ถึง 1 เก็บไว้ใน aryHiddenWeight, aryOutputWeight โดยใช้ RandomWeight()
5. ทดสอบค่า Error ที่ยอมรับได้ (โครงงานนี้กำหนดไว้ที่ 0.05)
6. แปลงข้อมูล 1 ชุด จาก Array 2 มิติ สู่ Array 1 มิติ เก็บไว้ใน aryInput, aryTarget
7. คำนวณหาค่า Net ของชั้น Hidden Layer เก็บไว้ใน aryHiddenNode โดยใช้ Net()
8. ส่งค่าผ่านฟังก์ชั่นกระตุ้น โดยใช้ Sigmoid() เพื่อให้ได้ Output ของชั้น Hidden
9. คำนวณหาค่า Net ของชั้น Output Layer เก็บไว้ใน aryOutputNode โดยใช้ Net()
10. ส่งค่าผ่านฟังก์ชั่นกระตุ้น โดยใช้ Sigmoid() เพื่อให้ได้ Output ของชั้น Output

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

11. คำนวณหาค่า Delta ของชั้น Otutput Layer Layer เก็บไว้ใน aryDeltaOutput โดยใช้ DeltaOutput()
12. คำนวณหาค่า Delta ของชั้น Hidden Layer Layer เก็บไว้ใน aryDeltaHidden โดยใช้ DeltaHidden()
13. คำนวณหาค่า Delta Weight ของชั้น Output และ Hidden โดยใช้ DeltaWeight()
14. นำค่า Delta Weight เข้าปรับ Weight ชั้น Output โดยใช้ UpdateWeight()
15. นำค่า Delta Weight เข้าปรับ Weight ชั้น Hidden โดยใช้ UpdateWeight()
16. คำนวณหาค่า SSE โดยใช้ SSE()
17. กลับสู่ขั้นตอนที่ 5

ขั้นตอนของโปรแกรมที่ใช้ในการ Test มีดังนี้

1. อ่านค่าตัวแปรจากหน้าจอ ประกอบด้วย LearningRate, ErrorAccept, HiddenNode, MaxLoop, StartRecordTrain, TotalRecordTrain, StartRecordTest, TotalRecordTest
2. อ่านข้อมูลจากฐานข้อมูล ลง Array 2 มิติ aryTableInput, AryTableTarget
3. ทำการแปลงค่าข้อมูลใน Array ให้อยู่ในช่วง 0 ถึง 1 เก็บไว้ใน aryConvInput, aryConvTarget โดยใช้ ConvSex(), ConvAge(), ConvCareer(), ConvSumInsure(), ConvPaid()
4. แปลงข้อมูล 1 ชุด จาก Array 2 มิติ สู่ Array 1 มิติ เก็บไว้ใน aryInput, aryTarget
5. คำนวณหาค่า Net ของชั้น Hidden Layer เก็บไว้ใน aryHiddenNode โดยใช้ Net()
6. ส่งค่าผ่านฟังก์ชั่นกระตุ้น โดยใช้ Sigmoid() เพื่อให้ได้ Output ของชั้น Hidden
7. คำนวณหาค่า Net ของชั้น Output Layer เก็บไว้ใน aryOutputNode โดยใช้ Net()
8. ส่งค่าผ่านฟังก์ชั่นกระตุ้น โดยใช้ Sigmoid() เพื่อให้ได้ Output ของชั้น Output
9. คำนวณหาค่า SSE โดยใช้ SSE()
10. กลับสู่ขั้นตอนที่ 4

ขั้นตอนการแสดงข้อมูลด้วยกราฟ

1. นำ Array ของค่า Target กับค่า Output แสดงเป็นกราฟ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 ส่วนของฐานข้อมูล

ในการศึกษาโครงงานนี้ ได้ออกแบบฐานข้อมูลให้มีแค่ Table เดียว เฉพาะฟิลด์ที่สนใจ เพื่อความสะดวกในการใช้งาน โดยใช้โปรแกรม Microsoft Access เป็นฐานข้อมูลให้แก่โปรแกรม ซึ่งประกอบไปด้วยฟิลด์ต่างๆ ดังนี้

ตารางที่ 3.1 ตารางแสดงฟิลด์ของฐานข้อมูล

ชื่อ	ชนิดข้อมูล	รายละเอียด
Sex	Text	เพศ
Age	Number	อายุ
Career	Text	อาชีพ (ลักษณะการใช้งานรถยนต์)
SumInsure	Number	ทุนประกัน
Paid	Number	มูลค่าความเสียหายที่เกิด

Input

- Sex
- Age
- Career
- SumInsure

Output

- Ratio ของ Paid กับ SumInsure โดยแบ่งเป็น 4 กลุ่ม
  - 0 – 10 %
  - 10 – 20 %
  - 20 – 50 %
  - 50 % ขึ้นไป

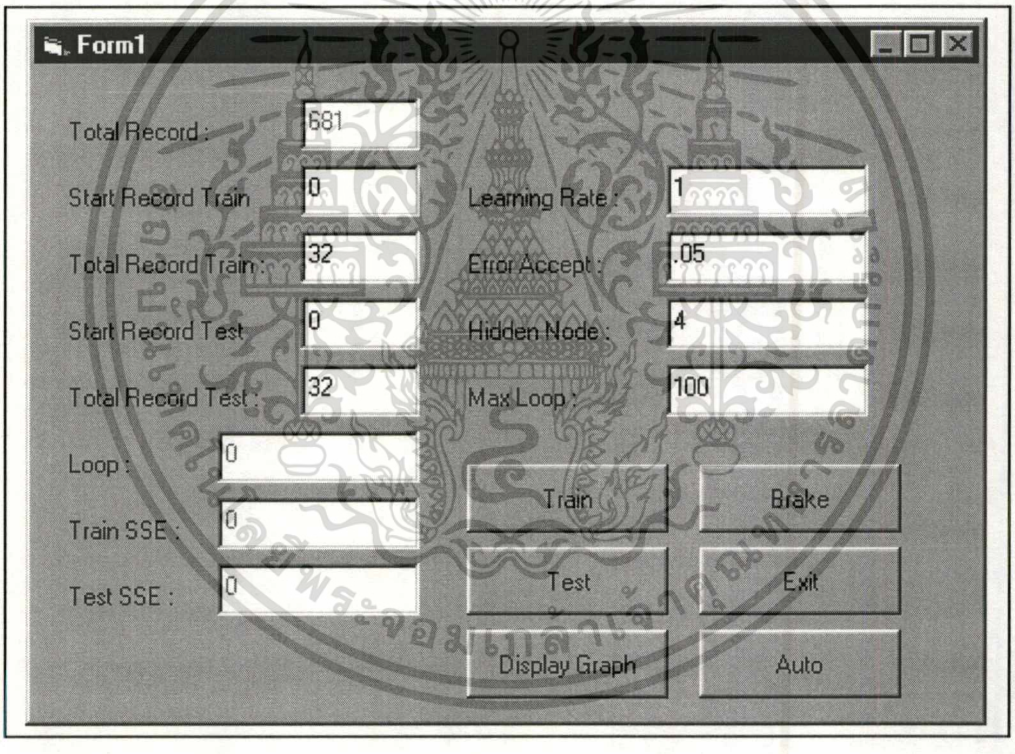
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4 ส่วนของโปรแกรม

ในการศึกษาโครงงานนี้ ได้ออกแบบและพัฒนาโปรแกรม โดยใช้โปรแกรม Microsoft Visual Basic Version 6.0 เป็นเครื่องมือในการพัฒนาโปแกรม เพื่อนำทฤษฎีโครงข่ายประสาทเทียมแบบแบคพรอพพาเกชั่น มาประยุกต์ใช้ในการจำแนกกลุ่มลูกค้า เนื่องจากเป็นโปรแกรมที่มีเครื่องมือที่ช่วยในการออกแบบหน้าจอ, ช่วยในการติดต่อ Database และสามารถแสดงผลเป็นกราฟ

3.4.1. หน้าจอหลัก

เมื่อเริ่มเรียกการใช้งานโปรกแกรม จะแสดงหน้าจอนี้ให้ผู้ใช้เลือกการใช้งานประกอบด้วย



ภาพที่ 3.2 แสดงหน้าจอหลัก

ฟิลด์ที่สามารถใส่ค่าได้

- Learning Rate

  ใช้กำหนดค่าอัตราการเรียนรู้ของโครงข่ายประสาทเทียม

- Error Accept

  ใช้กำหนดค่า Error ที่ยอมรับได้ ที่ใช้ในการหยุดการเรียนรู้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Hidden Node

  ใช้กำหนดจำนวน Node ใน Hidden Layer

- Max Loop

  ใช้กำหนดจำนวนรอบสูงสุดในการหยุดการเรียนรู้

- Start Record Train

  ใช้กำหนดตำแหน่ง Record เริ่มต้นในการเรียนรู้

- Total Record Train

  ใช้กำหนดจำนวน Record ในการเรียนรู้

- Start Record Test

  ใช้กำหนดตำแหน่ง Record เริ่มต้นในการทดสอบ

- Total Record Test

  ใช้กำหนดจำนวน Record ในการทดสอบ

ฟิลด์ที่ใช้แสดงผล

- Total Record

  ใช้แสดง Record ทั้งหมดที่มีอยู่

- Loop

  ใช้แสดงรอบที่ในการเรียนรู้

- Train SSE

  ใช้แสดง SSE ในการเรียนรู้

- Test SSE

  ใช้แสดง SSE ในการทดสอบ

ปุ่ม

- Train

  ใช้สั่งให้โปรแกรมเริ่มการเรียนรู้

- Test

  ใช้สั่งให้โปรแกรมเริ่มการทดสอบ

- Display Graph

  ใช้สั่งให้โปรแกรมแสดงกราฟเปรียบเทียบค่าที่ได้จากการทดสอบกับค่าจริง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Auto

  ใช้สั่งให้โปรแกรม เริ่มการเรียนรู้ ทดสอบ แล้วแสดงกราฟ

- Break

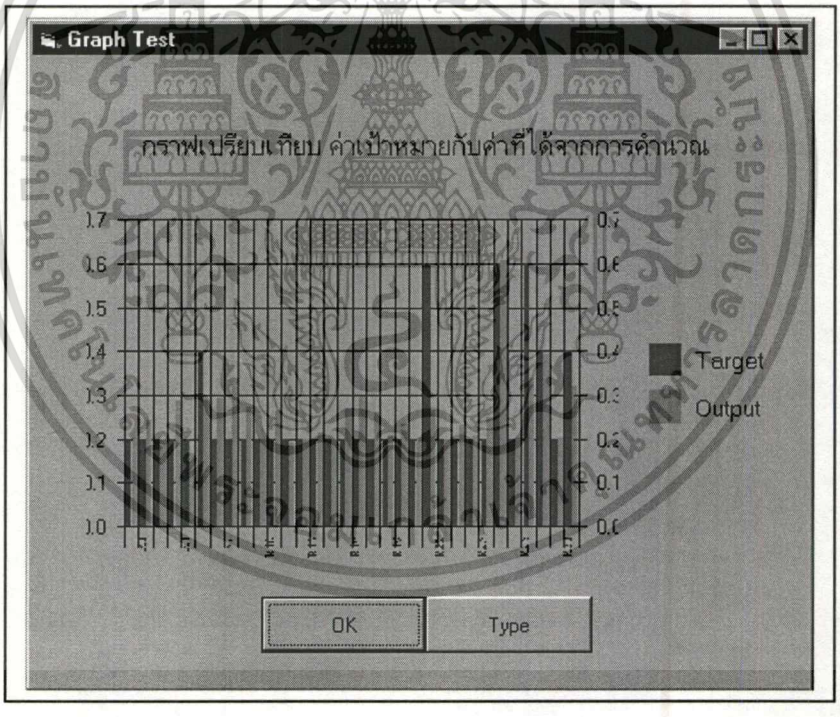
  ใช้สั่งให้โปรแกรมหยุดการทำงาน

- Exit

  ใช้สั่งให้ออกจากโปรแกรม

3.4.2. หน้าจอแสดงกราฟ

หน้าจอส่วนนี้ใช้แสดงผลลัพธ์ที่ได้จากโปรแกรมในรูปของกราฟ เพื่อเปรียบเทียบค่าจริงกับค่าที่ได้จากการคำนวณ ว่ามีความใกล้เคียงกันมากน้อยแค่ไหน



ภาพที่ 3.3 แสดงกราฟเปรียบเทียบ ค่าจริงกับค่าที่ได้จากการคำนวณ

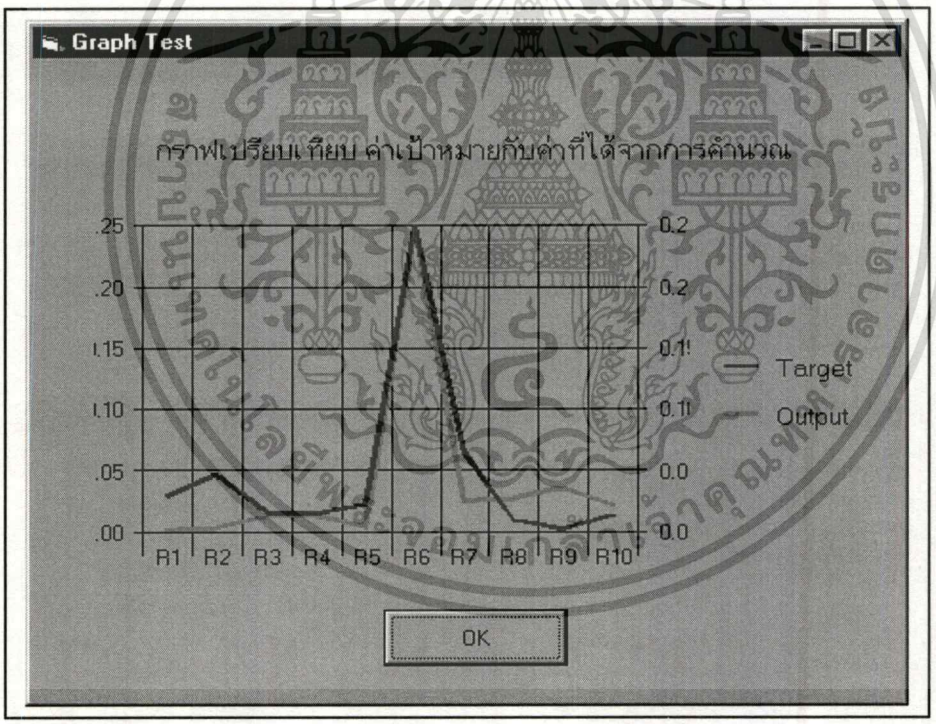
ในบทนี้แสดงให้เห็นถึงวิธีการดำเนินการศึกษาโครงงานนี้ว่า ได้มีขั้นตอนดำเนินการศึกษาและพัฒนาโปรแกรม เพื่อนำมาประยุกต์ใช้อย่างไร ในส่วนของการทำงานได้บรรยายเพียงว่าเรียกใช้ Sub Procedure ใดเท่านั้น ซอร์สโค้ดของโปรแกรมทั้งหมด ได้แสดงไว้ในภาคผนวกท้ายเล่มไว้แล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4
สรุปผลการศึกษา และข้อเสนอแนะ

4.1 สรุปผลการศึกษา

ผลที่ได้จากการศึกษา และพัฒนาโปรแกรมสำหรับการจำแนกกลุ่มลูกค้า โดยใช้โครงข่ายประสาทเทียมแบบแบคพรอพพาเกชั่น โดยใช้ข้อมูลประกันภัยรถยนต์เป็นข้อมูลสำหรับใช้ในการฝึกสอน และทำการทดลองจำแนกลูกค้าด้วยข้อมูลทดสอบ ผลที่ได้จากการดำเนินการศึกษาสามารถสรุปได้ ดังนี้



ภาพที่ 4.1 ผลการคำนวณ

ผลที่ได้จากการคำนวณด้วยโปรแกรมกับค่าจริง จะเห็นได้ว่ามีค่าใกล้เคียงกัน แต่จะมีบางค่าที่ไม่ใกล้เคียง เนื่องมาจากผลของปัจจัยหลายประการด้วยกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 ข้อเสนอแนะ

การดำเนินการศึกษาในครั้งนี้เป็นการศึกษาเฉพาะปัจจัยบางประการเท่านั้น เนื่องจากข้อมูลที่คิดว่าน่าสนใจบางอย่างไม่มีเก็บในฐานข้อมูล การดำเนินการศึกษาในครั้งนี้จึงใช้ข้อมูลเท่าที่หามาได้เท่านั้น ดังนั้นการนำโปรแกรมที่พัฒนานี้ไปใช้งานจริง อาจจะมีความคลาดเคลื่อนไปบ้าง จึงควรนำปัจจัยอื่นๆ ที่เกี่ยวข้องเข้ามามีส่วนร่วมในการคำนวณด้วย เช่น อายุรถยนต์ ฯลฯ ซึ่งน่าจะมีผลต่อการคำนวณมาเป็นส่วนประกอบด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

กิตติ ภักดีวัฒนะกุล และ จำลอง ครูอุตสาหะ. 2541. **Visual Basic 5 ฉบับ โปรแกรมเมอร์.** กรุงเทพมหานคร : ไทยเจริญการพิมพ์

สุทธิชัย มณีรัตนรุ่งโรจน์. 2539. **การจำลองการทำงานโครงข่ายประสาทเทียมแบบแพร่ย้อนกลับ.** วิทยานิพนธ์วิทยาศาสตรมหาบัณฑิต คณะเทคโนโลยีสารสนเทศ บัณฑิตวิทยาลัย, สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

James A. Freeman and David M. Skapura. 1991. **Neural Networks Algorithms Application and Programimg Technigues.** New York : Addison-Wesley.

Peter Cabena. et al. 1997. **Discovering Data Mining From Concept to Implementation.** New Jersey : Prentice Hall.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก
ซอร์สโค้ด

```
Option Explicit

Dim InputNode As Integer
Dim HiddenNode As Integer
Dim OutputNode As Integer
Dim TargetNode As Integer
Dim TotalRecordTrain As Integer
Dim TotalRecordTest As Integer
Dim StartRecordTrain As Integer
Dim StartRecordTest As Integer
Dim MaxLoop As Integer
Dim CountLoop As Integer
Dim ErrorAccept As Single
Dim NetError As Single
Dim TrainSSE As Single
Dim TestSSE As Single
Dim LearningRate As Single
Dim Brake As Boolean
Dim aryTableInput()
Dim aryTableTarget()
Dim aryConvInput()
Dim aryConvTarget()
Dim aryInput()
Dim aryHidden()
Dim aryOutput()
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Dim aryTarget()
Dim aryHiddenWeight()
Dim aryOutputWeight()
Dim aryDeltaHidden()
Dim aryDeltaOutput()
Dim Fmt As String

Sub RandomWeight()
   Dim i As Integer

   Randomize
   For i = 1 To (InputNode * HiddenNode)
      aryHiddenWeight(i) = (2 * Rnd) - 1
   Next i
   For i = 1 To (HiddenNode * OutputNode)
      aryOutputWeight(i) = (2 * Rnd) - 1
   Next i
End Sub

Function Sigmoid(NetInput) As Single
   Sigmoid = 1 / (1 + Exp(-NetInput))
End Function

Sub Net(ByRef aryIn, ByRef aryWeight, ByRef aryOut)
   Dim i, j As Integer
   Dim NodeIn, NodeOut As Integer

   NodeIn = UBound(aryIn)
   NodeOut = UBound(aryOut)
   For j = 1 To NodeOut
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
      aryOut(j) = 0
      For i = 1 To NodeIn
         aryOut(j) = aryOut(j) + (aryIn(i) * aryWeight(((i - 1) * NodeOut) + j))
      Next i
      aryOut(j) = Sigmoid(aryOut(j))
   Next j
End Sub

Sub DeltaOutput(ByRef aryTarget, ByRef aryOutput)
   Dim i As Integer
   Dim NodeTarget, NodeOutput As Integer
   ReDim aryDeltaOutput(UBound(aryOutput))

   NodeTarget = UBound(aryTarget)
   NodeOutput = UBound(aryOutput)
   For i = 1 To NodeOutput
      aryDeltaOutput(i) = 0
      aryDeltaOutput(i) = (aryTarget(i) - aryOutput(i)) * (aryOutput(i) * (1 - aryOutput(i)))
   Next i
End Sub

Sub DeltaHidden(ByRef aryIn, ByRef aryDelta, ByRef aryWeight)
   Dim i, j As Integer
   Dim NodeIn, NodeOut As Integer
   ReDim aryDeltaHidden(UBound(aryIn))

   NodeIn = UBound(aryIn)
   NodeOut = UBound(aryDelta)

   For i = 1 To NodeIn
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        aryDeltaHidden(i) = 0
        For j = 1 To NodeOut
            aryDeltaHidden(i) = aryDeltaHidden(i) + (aryDelta(j) * aryWeight(((i - 1) * NodeOut) +
j))
        Next j
        aryDeltaHidden(i) = aryDeltaHidden(i) * ((1 - aryIn(i)) * aryIn(i))
    Next i
End Sub

Function DeltaWeight(Delta, InputVector) As Single
    DeltaWeight = LearningRate * Delta * InputVector
End Function

Sub UpdateWeight(ByRef aryDelta, ByRef aryInput, ByRef aryWeight)
    Dim i, j As Integer
    Dim NodeIn, NodeOut As Integer
    Dim C As Integer

    NodeIn = UBound(aryInput)
    NodeOut = UBound(aryDelta)
    For i = 1 To NodeIn
        For j = 1 To NodeOut
            C = ((i - 1) * NodeOut) + j
            aryWeight(C) = aryWeight(C) + DeltaWeight(aryDelta(j), aryInput(i))
        Next j
    Next i
End Sub

Function SSE(Sum, Target, Output) As Single
    SSE = Sum + ((Target - Output) ^ 2)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
End Function

Function SquareError(Target, Output) As Single
   SquareError = (Target - Output) ^ 2
End Function

Private Sub cmdBrake_Click()
   Brake = True
End Sub

Sub ScreenIn()
   'Input Variable to Screen
   LearningRate = txtLearningRate.Text
   ErrorAccept = txtErrorAccept.Text
   HiddenNode = txtHiddenNode.Text
   MaxLoop = txtMaxLoop.Text
   StartRecordTrain = txtStartRecordTrain.Text
   TotalRecordTrain = txtTotalRecordTrain.Text
   StartRecordTest = txtStartRecordTest.Text
   TotalRecordTest = txtTotalRecordTest.Text
End Sub

Sub ScreenOut()
   'Output Variable to Screen
   txtLearningRate.Text = LearningRate
   txtErrorAccept.Text = ErrorAccept
   txtHiddenNode.Text = HiddenNode
   txtMaxLoop.Text = MaxLoop
   txtStartRecordTrain.Text = StartRecordTrain
   txtTotalRecordTrain.Text = TotalRecordTrain
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    txtStartRecordTest.Text = StartRecordTest
    txtTotalRecordTest.Text = TotalRecordTest
End Sub

Sub DisplayLoopError()
    txtLoop.Text = CountLoop
    txtTrainSSE.Text = Format(TrainSSE, Fmt)
    txtTestSSE.Text = Format(TestSSE, Fmt)
End Sub

Private Sub Form_Initialize()
    datData.Recordset.MoveLast
    txtTotalRecord = datData.Recordset.RecordCount
End Sub

Private Sub Form_Load()
    'Initial Fix Variable
    InputNode = 4
    OutputNode = 1
    Brake = False
    Fmt = "0.00000000"
    datDataTest.DatabaseName = App.Path & "\Neural.mdb"
    datData.DatabaseName = App.Path & "\Neural.mdb"
    datData.RecordSource = "QryNeural"

    'Initial Vary Variable
    LearningRate = 1
    ErrorAccept = 0.05
    HiddenNode = 4
    MaxLoop = 100
```

เอกสารนี้สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    StartRecordTrain = 0
    TotalRecordTrain = 32
    StartRecordTest = 0
    TotalRecordTest = 32

    Call ScreenOut
End Sub

Function ConvSex(Sex)
    If Sex = "F" Then
        ConvSex = 0
    Else
        ConvSex = 1
    End If
End Function

Function ConvAge(Age)
    ConvAge = (Age - 18) / (75 - 18)
End Function

Function ConvCareer(Career)
    Select Case Career
        Case "39"
            Career = 0
        Case "62"
            Career = 1
        Case "63"
            Career = 2
        Case "82"
            Career = 3
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        Case "84"
            Career = 4
        Case "85"
            Career = 5
        Case "95"
            Career = 6
        Case "99"
            Career = 7
        Case "C3"
            Career = 8
        Case "E1"
            Career = 9
        Case "E2"
            Career = 10
        Case "G1"
            Career = 11
        Case "M1"
            Career = 12
        Case "P1"
            Career = 13
        Case "S1"
            Career = 14
        Case "S3"
            Career = 15
        Case "ZZ"
            Career = 16
    End Select
    ConvCareer = Career / 16
End Function
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Function ConvSumInsure(SumInsure)
   If SumInsure > 4000000 Then SumInsure = 4000000
   ConvSumInsure = (SumInsure - 10000) / (4000000 - 10000)
   'ConvSumInsure = (SumInsure - 10000) / (20000000 - 10000)
End Function

Function ConvPaid(Paid)
   If Paid < 0 Then Paid = 0
   If Paid > 300000 Then Paid = 300000
   ConvPaid = (Paid - 0) / (300000 - 0)
   'ConvPaid = (Paid - (-20491)) / (912265 - (-20491))
End Function

Private Sub cmdExit_Click()
   End
End Sub

Private Sub cmdTrain_Click()
   Dim i, j As Integer

   Call ScreenIn

   'Load Data from Table to Array
   ReDim aryTableInput(TotalRecordTrain, InputNode)
   ReDim aryTableTarget(TotalRecordTrain, OutputNode)

   i = 1
   datData.Recordset.MoveFirst
   datData.Recordset.Move StartRecordTrain
   Do While (Not datData.Recordset.EOF) And (i <= TotalRecordTrain)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    aryTableInput(i, 1) = datData.Recordset.Sex
    aryTableInput(i, 2) = datData.Recordset.Age
    aryTableInput(i, 3) = datData.Recordset.Career
    aryTableInput(i, 4) = datData.Recordset.SumInsure
    aryTableTarget(i, 1) = datData.Recordset.Paid
    i = i + 1
    datData.Recordset.MoveNext
Loop
TotalRecordTrain = i - 1
Call ScreenOut

'Convert Data in Format 0-1
ReDim aryConvInput(TotalRecordTrain, InputNode)
ReDim aryConvTarget(TotalRecordTrain, OutputNode)

For i = 1 To TotalRecordTrain
    aryConvInput(i, 1) = Format(ConvSex(aryTableInput(i, 1)), Fmt)
    aryConvInput(i, 2) = Format(ConvAge(aryTableInput(i, 2)), Fmt)
    aryConvInput(i, 3) = Format(ConvCareer(aryTableInput(i, 3)), Fmt)
    aryConvInput(i, 4) = Format(ConvSumInsure(aryTableInput(i, 4)), Fmt)
    aryConvTarget(i, 1) = Format(ConvPaid(aryTableTarget(i, 1)), Fmt)
Next i

'Train NeuralNet
ReDim aryInput(InputNode)
ReDim aryTarget(OutputNode)
ReDim aryHidden(HiddenNode)
ReDim aryOutput(OutputNode)
ReDim aryHiddenWeight(InputNode * HiddenNode)
ReDim aryOutputWeight(HiddenNode * OutputNode)
```

เอกสารนี้ใช้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Call RandomWeight
Brake = False
CountLoop = 1
NetError = 1
Do While (CountLoop <= MaxLoop) And (Not Brake) And (NetError > ErrorAccept)
  NetError = 0
  For i = 1 To TotalRecordTrain
    For j = 1 To InputNode
      aryInput(j) = aryConvInput(i, j)
    Next j
    For j = 1 To OutputNode
      aryTarget(j) = aryConvTarget(i, j)
    Next j

    Call Net(aryInput, aryHiddenWeight, aryHidden)
    Call Net(aryHidden, aryOutputWeight, aryOutput)
    Call DeltaOutput(aryTarget, aryOutput)
    Call DeltaHidden(aryHidden, aryDeltaOutput, aryOutputWeight)
    Call UpdateWeight(aryDeltaOutput, aryHidden, aryOutputWeight)
    Call UpdateWeight(aryDeltaHidden, aryInput, aryHiddenWeight)
    For j = 1 To (OutputNode)
      aryOutput(j) = Format(aryOutput(j), Fmt)
      NetError = SSE(NetError, aryTarget(j), aryOutput(j))
    Next j
  Next i
  TrainSSE = NetError
  Call DisplayLoopError
  CountLoop = CountLoop + 1
  DoEvents
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    Loop
    CountLoop = CountLoop - 1
End Sub

Private Sub cmdTest_Click()
    Dim i, j As Integer

    Call ScreenIn

    ReDim gbl_aryDisplayTest(1 To TotalRecordTest, 1 To 2)

    'Load Data from Table to Array
    ReDim aryTableInput(TotalRecordTest, InputNode)
    ReDim aryTableTarget(TotalRecordTest, OutputNode)

    i = 1
    datData.Recordset.MoveFirst
    datData.Recordset.Move StartRecordTest
    Do While (Not datData.Recordset.EOF) And (i <= TotalRecordTest)
        aryTableInput(i, 1) = datData.Recordset.Sex
        aryTableInput(i, 2) = datData.Recordset.Age
        aryTableInput(i, 3) = datData.Recordset.Career
        aryTableInput(i, 4) = datData.Recordset.SumInsure
        aryTableTarget(i, 1) = datData.Recordset.Paid
        i = i + 1
        datData.Recordset.MoveNext
    Loop
    TotalRecordTest = i - 1
    Call ScreenOut
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
'Convert Data in Format 0-1
ReDim aryConvInput(TotalRecordTest, InputNode)
ReDim aryConvTarget(TotalRecordTest, OutputNode)

For i = 1 To TotalRecordTest
   aryConvInput(i, 1) = Format(ConvSex(aryTableInput(i, 1)), Fmt)
   aryConvInput(i, 2) = Format(ConvAge(aryTableInput(i, 2)), Fmt)
   aryConvInput(i, 3) = Format(ConvCareer(aryTableInput(i, 3)), Fmt)
   aryConvInput(i, 4) = Format(ConvSumInsure(aryTableInput(i, 4)), Fmt)
   aryConvTarget(i, 1) = Format(ConvPaid(aryTableTarget(i, 1)), Fmt)
Next i

'Test NeuralNet
ReDim aryInput(InputNode)
ReDim aryTarget(OutputNode)
ReDim aryHidden(HiddenNode)
ReDim aryOutput(OutputNode)

Brake = False
NetError = 0
For i = 1 To TotalRecordTest
   For j = 1 To InputNode
      aryInput(j) = aryConvInput(i, j)
   Next j
   For j = 1 To OutputNode
      aryTarget(j) = aryConvTarget(i, j)
   Next j

   Call Net(aryInput, aryHiddenWeight, aryHidden)
   Call Net(aryHidden, aryOutputWeight, aryOutput)
```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    For j = 1 To (OutputNode)
      aryOutput(j) = Format(aryOutput(j), Fmt)
      NetError = SSE(NetError, aryTarget(j), aryOutput(j))
    Next j

    gbl_aryDisplayTest(i, 1) = aryTarget(1)
    gbl_aryDisplayTest(i, 2) = aryOutput(1)

  Next i
    TestSSE = NetError
    Call DisplayLoopError
End Sub

Private Sub cmdDisplayGraph_Click()
  frmGraphTest.Show
End Sub

Private Sub cmdAuto_Click()
  Call cmdTrain_Click
  Call cmdTest_Click
  Call cmdDisplayGraph_Click
End Sub
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประวัติผู้เขียน

ชื่อผู้เขียน	นายชัยวุฒิ โฆษิตจินดา	
วันเดือนปีเกิด	5 สิงหาคม 2514	
สถานที่เกิด	จังหวัดกรุงเทพมหานคร	
วุฒิการศึกษาระดับปริญญาตรี	วท.บ. คอมพิวเตอร์	
สถานที่สำเร็จการศึกษา	มหาวิทยาลัยรามคำแหง	
ปีที่สำเร็จการศึกษา	2536	
ประวัติการทำงาน	2536	ทำงานฝ่ายคอมพิวเตอร์ บริษัท ซิสเต็ม-โปร จำกัด
	2537	ทำงานฝ่ายคอมพิวเตอร์ บริษัทเงินทุนหลักทรัพย์ เอ็ม ซี ซี จำกัด (มหาชน)
	2541 - ปัจจุบัน	ทำงานฝ่ายคอมพิวเตอร์ บริษัท ซี จี ยู ประกันภัย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้