

การประมาณการผลการจัดเก็บภาษีมูลค่าเพิ่ม
โดยใช้โครงข่ายประสาทเทียมแบบเบคพรอพพาเกชั่น

Forecasting of VAT Collection
by Backpropagation Neural Network



รายงานนี้เป็นส่วนหนึ่งของวิชาโครงการพัฒนาระบบงาน
หลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ

ภาคเรียนที่ 1 ปีการศึกษา 2542

คณะเทคโนโลยีสารสนเทศ

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชื่อหัวข้อ	การประมาณการผลการจัดเก็บภาษีมูลค่าเพิ่ม โดยใช้โครงข่ายประสาทเทียมแบบแบคพรอพเกชัน (Backpropagation Neural Network)
นักศึกษา	นายสนิทวงศ์ กมลภากรณ์
อาจารย์ที่ปรึกษา	ดร.อาริต ธรรมโน
ระดับการศึกษา	วิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ
แขนงวิชา	วิทยาการสารสนเทศ
ปีการศึกษา	2542

บทคัดย่อ

กรมสรรพากรมีหน้าที่จัดเก็บภาษีอันเป็นรายได้หลักสำคัญอันดับหนึ่งของประเทศ โดยได้รับมอบหมายให้ทำหน้าที่จัดเก็บภาษีเพิ่มขึ้นในแต่ละปีตามงบประมาณรายได้รายจ่ายของรัฐบาลในการบริหารการจัดเก็บภาษีอากรที่ดี ต้องมีการควบคุมและติดตามผลการจัดเก็บภาษีที่ได้ดำเนินการมาแล้ว นำมาวิเคราะห์เพื่อที่จะพยายามเร่งรัดจัดเก็บภาษีต่อไปให้ได้ตามเป้าหมายที่ได้วางไว้ กรมสรรพากรมีหน้าที่จัดเก็บภาษีหลายประเภท ปัจจุบันได้มีการนำเทคโนโลยีสารสนเทศมาใช้ในการประมวลผลสำหรับภาษีบางประเภทเท่านั้น

โครงการนี้จะเป็นการศึกษาทฤษฎีที่เกี่ยวข้องกับโครงข่ายประสาทเทียมแบบแบคพรอพเกชัน (Backpropagation Neural Network) และนำทฤษฎีโครงข่ายประสาทเทียมแบบแบคพรอพเกชันมาประยุกต์ใช้เพื่อช่วยในการทำประมาณการผลการจัดเก็บภาษีมูลค่าเพิ่มในแต่ละปี โดยทำการเปรียบเทียบผลลัพธ์ที่ได้จากการนำทฤษฎีโครงข่ายประสาทเทียมแบบแบคพรอพเกชันมาใช้กับผลที่ได้จากการจัดเก็บจริงของปีที่ผ่านมาว่าจะได้ผลลัพธ์ที่ถูกต้องหรือใกล้เคียงมากน้อยแค่ไหน เพื่อที่จะได้เป็นแนวทางในการนำมาประยุกต์ใช้กับงานการทำประมาณการการจัดเก็บภาษีอื่นๆ ของกรมสรรพากรได้ต่อไปอย่างมีประสิทธิภาพ

Title	Forecasting of VAT Collection by Backpropagation Neural Network
Student	Mr. Sanitwong Kamonparkorn
Advisor	Dr. Arit Thammano
Level of Study	Master of Science in Information Technology
Major	Information Science
Academic Year	1999

ABSTRACT

The Revenue Department has, as its main duty, a function to collect taxes which become the primary source of income for the nation. Its target is set when the government has concluded the budgeting decision. After the tax collecting process has carried out, the result will be analyzed against the target set forth. Any deviations will be closely examined during the period and planned action be activated accordingly. The result from each tax collecting process would be gathered together statistically and later evaluated by utilizing the computer technology. However, as there are many tax categories involved, the information technology is presently being applied to some tax categories only.

This project studies the theory of Backpropagation Neural Network. Then Backpropagation Neural Network is used to Forecast the VAT (Value Added Tax) Collection. By comparing the result received from Backpropagation Neural Network with the result real data in past. This model apply to forecast another tax.

กิตติกรรมประกาศ

ขอขอบพระคุณท่านอาจารย์ ดร. อาริต ธรรมโน ที่กรุณาให้คำปรึกษา และช่วยให้แนวทางในการทำโครงการนี้

ขอบคุณพี่ๆ เพื่อนๆ ที่เรียนด้วยกัน และพี่ๆ ที่สำนักเทคโนโลยีสารสนเทศกรมสรรพากร ที่กรุณาเอื้อเพื่อให้ยืมเครื่องคอมพิวเตอร์ใช้ในการพัฒนาโปรแกรม

สนิทวงศ์ กมลภากรณ์



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 วัตถุประสงค์	1
1.3 ขอบเขตของการศึกษา	2
1.4 แผนการดำเนินการศึกษา	2
1.5 ประโยชน์ที่คาดว่าจะได้รับจากโครงการ	2
2. ทฤษฎีที่เกี่ยวข้อง	4
2.1 ทฤษฎีเซลล์ประสาททางชีววิทยาเบื้องต้น (Biological Neurons)	4
2.2 ตัวแบบของเซลล์ประสาทเทียม (Artificial Neural Model)	5
2.3 สถาปัตยกรรมของโครงข่ายประสาทเทียม (Neural Network Architecture)	9
2.4 การฝึกสอนโครงข่ายประสาทเทียม (Training)	10
2.5 เพอเซปตรอน (Perceptron)	11
2.6 กฎเดลต้า (The Delta Rule)	11
2.7 โครงข่ายนิวรอนแบบแบคพรอพพาคชัน (Backpropagation Neural Network)	12
3. วิธีดำเนินการศึกษา	17
3.1 ส่วนของโครงสร้างแบบจำลอง	17
3.2 ส่วนของอัลกอริทึม	18
3.3 ส่วนของฐานข้อมูล	24
3.4 ส่วนของโปรแกรม	25

เอกสารนี้เป็นเอกสารลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. ผลการดำเนินการศึกษา	29
4.1 ส่วนของกราฟแสดงผล	29
4.2 ส่วนของข้อมูลที่เป็นตัวเลขเพื่อใช้เปรียบเทียบกัน	30
5. สรุปผลการศึกษาและข้อเสนอแนะ	31
5.1 สรุปผลที่ได้จากการดำเนินการศึกษา	31
5.2 ข้อเสนอแนะ	31
บรรณานุกรม	32
ภาคผนวก ก ข้อมูลผลการจัดเก็บภาษีมูลค่าเพิ่มที่จัดเก็บได้ทั่วประเทศ	33
ภาคผนวก ข ตัวอย่างซอร์สโค้ดโปรแกรม	35
ประวัติผู้เขียน	70



สารบัญตาราง

	หน้า
ตารางที่	
3.1 ตารางแสดงพีลด์ในฐานข้อมูล	25
4.1 แสดงการเปรียบเทียบผลที่ได้จากการทำนายกับค่าเป้าหมาย	30



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ

	หน้า
ภาพที่	
2.1 แสดงลักษณะเซลล์ของประสาทมนุษย์ทางชีววิทยา	5
2.2 แสดงตัวแบบของเซลล์ประสาทเทียม	6
2.3 แสดงลักษณะของสมการ threshold	7
2.4 แสดงกราฟของฟังก์ชัน ไปนารีซิกมอยด์	8
2.5 แสดงกราฟของฟังก์ชัน ไปโพล่าซิกมอยด์	8
2.6 แสดงลักษณะ โครงข่ายแบบชั้นเดียว (Single Layer)	9
2.7 แสดงลักษณะ โครงสร้างแบบหลายชั้น (Multilayer)	10
2.8 ลักษณะของเพอเซปตรอน (Perceptron)	11
2.9 แสดงลักษณะ โครงข่ายนิเวรอนแบบแบคพรอพพาเกชั่น	13
3.1 แสดง โครงสร้างแบบจำลอง โครงข่ายประสาทเทียม แบบแบคพรอพพาเกชั่นที่ใช้ในโครงการนี้	18
3.2 แสดงหน้าจอเมนูหลัก	25
3.3 แสดงหน้าจอสำหรับบันทึกข้อมูลผลการจัดเก็บภาหิมูลค่าเพิ่ม	26
3.4 แสดงหน้าจอประมวลผลหาค่าประมาณการ	27
3.5 แสดงหน้าจอสำหรับแสดงผลด้วยกราฟ	28
4.1 กราฟแสดงผลที่ได้จากการทำนาย	29

บทที่ 1

บทนำ

1.1 ความเป็นมา

กรมสรรพากรเป็นหน่วยงานสังกัดกระทรวงการคลัง ที่มีหน้าที่ในการจัดเก็บภาษีอากร ซึ่งระบบภาษีอากรที่กรมสรรพากรจัดเก็บประกอบด้วย ภาษีเงินได้บุคคลธรรมดา ภาษีเงินได้นิติบุคคล ภาษีเงินได้หัก ณ ที่จ่าย ภาษีธุรกิจเฉพาะ และระบบภาษีมูลค่าเพิ่ม โดยระบบภาษีมูลค่าเพิ่มได้เริ่มนำมาใช้ตั้งแต่วันที่ 1 มกราคม พ.ศ. 2535

โครงสร้างของกรมสรรพากรตามพระราชกฤษฎีกา แบ่งส่วนราชการกรมสรรพากร กระทรวงการคลัง พ.ศ. 2540 โดยมีหน่วยงานที่เกี่ยวข้องกับระบบสารสนเทศ และหน่วยงานบริหารที่มีหน้าที่เกี่ยวข้องกับการวางแผนในการบริหารการจัดเก็บภาษีมูลค่าเพิ่ม คือ

1. สำนักเทคโนโลยีสารสนเทศ มีหน้าที่ในการวางแผน และพัฒนาระบบงานคอมพิวเตอร์ เป็นศูนย์กลางข้อมูล และประมวลผลการจัดเก็บภาษี
2. สำนักแผนภาษี มีหน้าที่กำหนดนโยบาย และเป้าหมายการจัดเก็บภาษี พร้อมทั้งการจัดทำแผนแม่บท
3. สำนักมาตรฐานกรรมวิธีภาษี มีหน้าที่กำหนดปรับปรุงพัฒนามาตรฐาน หลักเกณฑ์ ขั้นตอนการปฏิบัติงานเกี่ยวกับภาษี

การจัดทำแผนการปฏิบัติงานของกรมสรรพากรเป็นหน้าที่ของสำนักแผนภาษี ได้รวบรวมข้อมูลทำแผนปฏิบัติงานประจำปีทุกปี โดยสรุปแผนและนโยบายเสนอผู้บริหารของกรมสรรพากร ซึ่งการรวบรวมข้อมูลเพื่อทำการประมาณการผลการจัดเก็บภาษีมูลค่าเพิ่ม ยังคงใช้การรวบรวมด้วยเจ้าหน้าที่แบบ Manual และยังไม่มีการใช้ หรือวิธีที่ใช้ในการทำการประมาณการที่ตายตัว และเชื่อถือได้เพียงพอ ซึ่งอาจก่อให้เกิดความผิดพลาดในเรื่องของตัวเลขได้ง่าย

1.2 วัตถุประสงค์

1. เพื่อศึกษาและทำความเข้าใจเกี่ยวกับทฤษฎีโครงข่ายประสาทเทียมแบบแบคพรอพเกชัน (Backpropagation Neural Network)
2. เพื่อจะศึกษาถึงแนวทางความเป็นไปได้ในการที่จะนำมาประยุกต์ใช้กับการทำการประมาณการผลการจัดเก็บภาษีมูลค่าเพิ่มของกรมสรรพากร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. เพื่อเป็นต้นแบบในการที่จะนำไปประยุกต์ใช้กับการทำประมาณการจัดเก็บภาษีอากรอื่นๆ ของกรมสรรพากร
4. เพื่อเป็นแนวทางในการนำไปประยุกต์ใช้ในการสนับสนุนการตัดสินใจของผู้บริหารสำหรับวางแผนในด้านการจัดเก็บภาษีมูลค่าเพิ่ม
5. เพื่อศึกษาเปรียบเทียบผลที่ได้จากการทำประมาณการผลการจัดเก็บแบบเดิมกับแบบที่นำทฤษฎีของโครงข่ายนิวรอนแบบคพรอพเกชั่น (Backpropagation Neural Network) มาใช้ว่าแบบใดจะมีความคลาดเคลื่อนต่ำกว่ากัน

1.3 ขอบเขตของโครงการ

โครงการนี้ได้ทำการศึกษาโดยใช้ข้อมูลของผลการจัดเก็บภาษีมูลค่าเพิ่มประจำเดือน ที่จัดเก็บได้จริงทั่วประเทศตั้งแต่ เดือน ม.ค. 2536 ถึง เดือน เม.ย. 2542 เป็นข้อมูลสำหรับการฝึกสอน และใช้ข้อมูลที่ได้จากการทำประมาณการ ตั้งแต่ เดือน ม.ค. 2542 ถึง เดือน เม.ย. 2542 เป็นค่าเป้าหมายในการเปรียบเทียบ ของโครงข่ายนิวรอนแบบคพรอพเกชั่น (Backpropagation Neural Network)

1.4 แผนการดำเนินการศึกษา

งาน	มิ.ย.	ก.ค.	ธ.ค.	ก.ย.	ต.ค.
1. ทำการศึกษาทฤษฎีที่เกี่ยวข้อง	—	—			
2. ทำการรวบรวมข้อมูลผลการจัดเก็บภาษีมูลค่าเพิ่ม		—			
3. ทำการออกแบบฐานข้อมูล		—			
4. ออกแบบอัลกอริทึมของโครงข่าย ที่นำมาใช้ในระบบที่พัฒนา			—		
5. ทำการพัฒนาโปรแกรม และทดสอบ				—	—
6. สรุปผลที่ได้จากการศึกษา และจัดทำรายงาน					—

1.5 ประโยชน์ที่คาดว่าจะได้รับจากโครงการ

1. เข้าใจถึงหลักการของโครงข่ายนิวรอนแบบคพรอพเกชั่น (Backpropagation Neural Network)

เอกสารนี้เป็นเอกสารราชการ
 2. สามารถประยุกต์ใช้โครงข่ายนิวรอนแบบคพรอพเกชั่น (Backpropagation Neural Network) ในการคำนวณภาษีมูลค่าเพิ่มได้
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Network) ใช้ในการช่วยแก้ปัญหาเกี่ยวกับการทำประมาณการผลการจัดเก็บภาษีมูลค่าเพิ่มแต่ละปีของกรมสรรพากรได้

3. สามารถลดเวลาที่ใช้ในการจัดทำประมาณการผลการจัดเก็บภาษีมูลค่าเพิ่มได้
4. เพิ่มประสิทธิภาพการทำงานของกรมสรรพากร
5. สามารถเป็นต้นแบบในการนำไปประยุกต์ใช้กับการทำประมาณการผลการจัดเก็บภาษีอากรอื่นๆ ของกรมสรรพากรได้
6. เป็นข้อมูลสำหรับผู้บริหาร เพื่อช่วยในการตัดสินใจและวางแผนปฏิบัติงานด้านการจัดเก็บภาษีมูลค่าเพิ่มของกรมสรรพากรได้อย่างมีประสิทธิภาพ



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

ทฤษฎีที่เกี่ยวข้อง

มนุษย์และคอมพิวเตอร์ต่างก็มีความสามารถในการทำงานบางอย่างที่แตกต่างกัน เช่น งานในการจัดการฐานข้อมูล งานประมวลผลการคำนวณ เป็นต้น ซึ่งงานเหล่านี้คอมพิวเตอร์สามารถทำได้รวดเร็วกว่ามนุษย์หลายเท่า แต่งานบางอย่าง เช่น การจดจำเสียง การจดจำภาพหน้าคน มนุษย์สามารถทำงานได้ดีกว่าและเร็วกว่า ซึ่งถ้าเป็นคอมพิวเตอร์ทำแล้วอาจจะไม่สามารถจดจำได้ และต้องใช้เวลาในการประมวลผลนานด้วย

สมองมนุษย์ประกอบด้วยเซลล์ประสาทจำนวนมากที่เชื่อมต่อกัน ซึ่งเซลล์เหล่านี้ทำหน้าที่เป็นหน่วยประมวลผลให้กับสมองมนุษย์ การส่งสัญญาณของเซลล์ประสาทระหว่างเซลล์หนึ่งสู่อีกเซลล์หนึ่งใช้เวลาเร็วมาก คือ ประมาณสิบล้านล้านวินาที ซึ่งรูปแบบการส่งสัญญาณทำในลักษณะขนานกัน และวิธีการแก้ปัญหาของสมองมนุษย์ใช้ประสบการณ์ที่เกิดจากการเรียนรู้ในอดีตนำมาวิเคราะห์เพื่อแก้ปัญหา ขณะที่สถาปัตยกรรมของคอมพิวเตอร์ประกอบด้วยหน่วยประมวลผลเพียงหน่วยเดียว และการทำงานใช้ชุดคำสั่งที่ส่งงานแบบลำดับขั้น และวิธีการในการแก้ปัญหานั้นจะต้องทำตามลำดับขั้นตอนของการทำงานที่แน่นอน

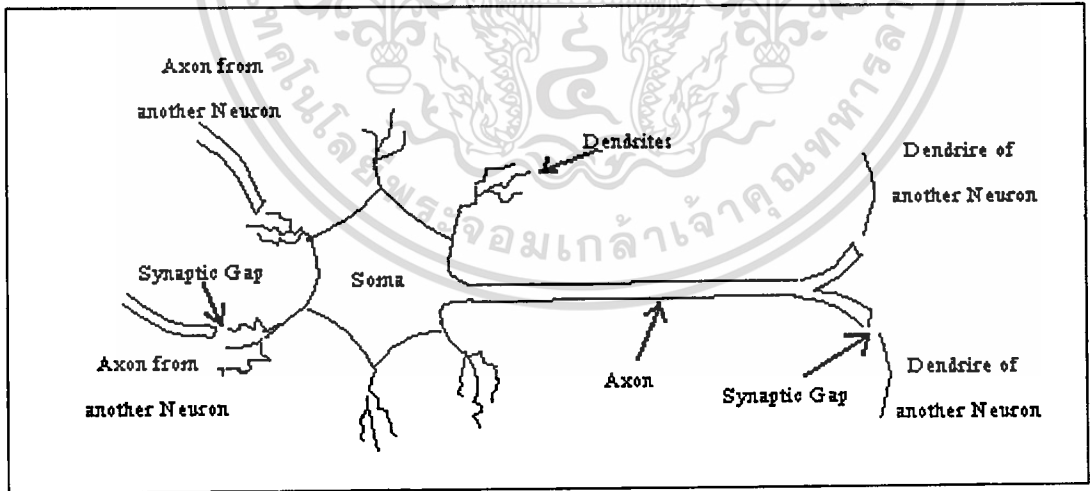
ดังนั้น เมื่อเราต้องการให้คอมพิวเตอร์สามารถจัดการกับปัญหาในลักษณะที่คล้ายกับมนุษย์ จึงได้มีการนำสถาปัตยกรรมของสมองมนุษย์มาเป็นตัวแบบในการประมวลผลด้วยคอมพิวเตอร์ ซึ่งเรียกว่า โครงข่ายประสาทเทียม (Neural Network) จะมีลักษณะที่คล้ายคลึงกับสมองและระบบประสาท ซึ่งประกอบด้วยส่วนของการประมวลผลต่างๆ ที่เรียกว่า นิวรอน (Neuron) ทุกๆ นิวรอนสามารถมีอินพุตได้หลายอินพุต แต่มีเอาต์พุตเพียงเอาต์พุตเดียว และทุกๆ เอาต์พุตจะแยกไปยังอินพุตของนิวรอนอื่นๆ ภายในโครงข่าย การติดต่อกันภายในระหว่างนิวรอน อินพุตจะมีน้ำหนักเป็นตัวกำหนดกำลังของการติดต่อภายในและช่วยในการตัดสินใจ การทำงานของนิวรอนในบางโครงข่ายจะถูกกำหนดไว้ตายตัว แต่บางโครงข่ายสามารถที่จะปรับแต่งได้ ซึ่งอาจจะเป็นการปรับแต่งจากภายนอกโครงข่าย หรือนิวรอนสามารถปรับได้ด้วยตัวมันเอง ในจุดนี้เองที่แสดงถึงความสามารถในการเรียนรู้ และจดจำของโครงข่ายประสาทเทียม

2.1 ทฤษฎีเซลล์ประสาททางชีววิทยาเบื้องต้น (Biological Neurons)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

องค์ประกอบของสมองมนุษย์นั้นประกอบไปด้วยเซลล์ประสาท หรือนิวรอน ประมาณ 10^{11} ตัว โดยที่เซลล์ประสาทหนึ่งๆ มีเส้นเชื่อมโยงต่อไปยังอีกเซลล์ประสาทอื่นๆ ประมาณ 10^4 ตัว และเวลาในการสับเปลี่ยนข้อมูลระหว่างนิวรอนประมาณ 10^{-3} วินาที ซึ่งนับว่าช้ามากเมื่อเทียบกับเวลาในการ switch ของคอมพิวเตอร์ที่ใช้เวลาเพียง 10^{-10} วินาที ลักษณะเซลล์ประสาททางชีวภาพ ประกอบไปด้วย 3 ส่วนหลักคือ เดนไดรต์ (Dendrites) โซมา (Soma) หรือตัวเซลล์ (Cell body) และแอกซอน (Axon) ส่วนที่เชื่อมระหว่างตัวเซลล์กับแอกซอนของเซลล์อื่นๆ ที่อยู่รอบๆ ข้างเรียกว่า ไชนแนปส์ (Synapse) โดยตัวเดนไดรต์จะรับสัญญาณจากเซลล์ประสาทที่อยู่รอบๆ ข้างผ่านทาง ไชนแนปส์ ด้วยปฏิกิริยาทางเคมี ซึ่งสัญญาณที่เข้ามาในที่นี้คือ ประจุอิเล็กตรอน โดยปฏิกิริยาเคมีที่เกิดขึ้น จะทำการปรับเปลี่ยนความถี่ของสัญญาณที่เข้ามา ตัวเซลล์ทำหน้าที่รวมสัญญาณที่เข้ามาแล้วทำการส่งสัญญาณออกไปให้กับแอกซอน เพื่อผ่านต่อไปให้กับเซลล์อื่นๆ ซึ่งสัญญาณที่ออกจากตัวเซลล์มีลักษณะเป็นแบบสัญญาณกระตุ้นให้กับเซลล์อื่น กล่าวคือ ถ้าค่าสัญญาณที่ส่งออกจากตัวเซลล์มีจำนวน 100 ครั้งต่อวินาที ถือว่าค่าสัญญาณที่ส่งออกเป็นสถานะการกระตุ้น (Fire) โดยทั่วไปเราสนใจสัญญาณที่ออกจากเซลล์ประสาทในรูปของสถานะการกระตุ้น และไม่กระตุ้น (Not Fire) ให้กับเซลล์ที่อยู่รอบข้างมากกว่าสนใจเงื่อนไขที่เกี่ยวข้องที่ทำให้เซลล์ประสาทเกิดสถานะนี้ขึ้นมา

2.1 แสดงลักษณะของเซลล์ประสาททางชีววิทยา



ภาพที่ 2.1 แสดงลักษณะของเซลล์ของประสาทมนุษย์ทางชีววิทยา

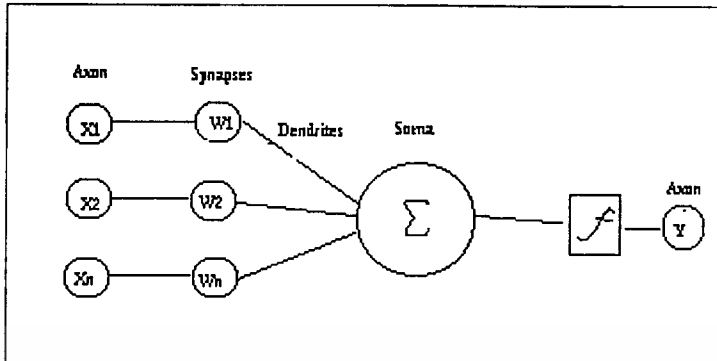
2.2 ตัวแบบของเซลล์ประสาทเทียม (Artificial Neural Model)

ลักษณะของตัวแบบเซลล์ประสาทที่ใช้กันโดยทั่วไป ซึ่งใช้พื้นฐานจากตัวแบบที่เสนอโดย

MCCulloch และ Pitt ในปี ค.ศ. 1943 ดังภาพที่ 2.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 2.2 แสดงตัวอย่างของเซลล์ประสาทเทียม

เซลล์ประสาทเทียมจะมีลักษณะคล้ายกับเซลล์ของประสาทมนุษย์ทางชีววิทยา คือ มีส่วนของสัญญาณอินพุต (input) จากรูป คือ X_1 ถึง X_n จากเซลล์ที่อยู่รอบข้าง ที่ส่งเข้ามาจะมีค่าน้ำหนัก W_1 ถึง W_n ซึ่งเปรียบเสมือนส่วนของไซแนปส์ (Synapse) ในเซลล์ประสาทของมนุษย์ ซึ่งสามารถปรับค่าได้ ผ่านเข้ามาที่ส่วนที่ทำหน้าที่รวมสัญญาณอินพุต (input) ที่เข้ามาและออกเป็นค่าสัญญาณเอาต์พุต (output) ค่าหนึ่ง สำหรับค่า สัญญาณเอาต์พุต ที่ออกมาสามารถคำนวณตามสมการ ดังนี้คือ

$$\text{net}_i = X_1 W_1 + X_2 W_2 + \dots + X_n W_n \quad (2.1)$$

หรือ
$$\text{net}_i = \sum X_j W_{ij} \quad (2.2)$$

$$Y_i = f(\text{net}_i) \quad i=1,2,\dots,n \quad (2.3)$$

- โดยที่ X_j คือ ค่าอินพุตลำดับที่ j ที่เข้าสู่เซลล์โหนดที่ i
 W_{ij} คือ ค่าน้ำหนักลำดับที่ j ที่เข้าสู่เซลล์โหนดที่ i
 net_i คือ ค่า net input ของเซลล์โหนดที่ i
 $f()$ คือ ฟังก์ชันกระตุ้นของเซลล์
 Y_i คือ ค่า เอาต์พุตที่ออกจากเซลล์โหนดที่ i
 n คือ จำนวนเซลล์ประสาทเทียมของโครงข่าย

ค่าของส่วนเชื่อมต่อของเซลล์ประสาทเทียมอาจมีค่าเป็น บวก ลบ หรือศูนย์ก็ได้ ดังนั้นในการปรับแต่งค่าสัญญาณที่เข้าสู่เซลล์ประสาทด้วยค่าน้ำหนักของส่วนที่เชื่อมต่อ ถ้าข้อมูลที่เข้าสู่เซลล์ประสาทเทียมมีค่าเป็นบวก ก็จะก่อให้เกิดสภาพเสริมกัน และถ้าค่าของส่วนเชื่อมต่อมีค่าเป็นลบจะเป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

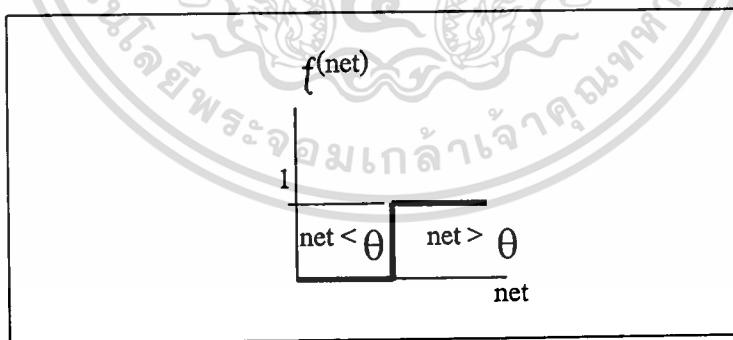
ลบ ก็จะลดค่าความเข้มของข้อมูลที่เข้ามานั้น ส่วนค่าของส่วนที่เชื่อมต่อมีค่าเป็นศูนย์จะหมายถึง ส่วนที่เชื่อมต่อตรงส่วนนั้นไม่มีการทำงาน

โดยทั่วไปแล้ว ส่วนเอาต์พุตที่ออกมา จะถูกกระทำต่อไปโดยส่วนที่เรียกว่า ฟังก์ชันกระตุ้น (Activation Function) ค่าที่ได้จากฟังก์ชันกระตุ้นนี้ทำให้เซลล์ประสาทเทียมมีลักษณะเหมือนกับ เซลล์ประสาททางชีววิทยา กล่าวคือ ค่าผลลัพธ์ที่ได้จากฟังก์ชันกระตุ้นเป็นค่าที่อยู่ในช่วง 0 ถึง 1 ถ้า ค่าสัญญาณที่ออกจากเซลล์ทางชีววิทยามีลักษณะเป็นสถานะของการกระตุ้น เราสามารถแทนความหมายของสถานะการกระตุ้นของเซลล์ประสาทด้วยค่า 1 และการไม่กระตุ้นด้วยค่า 0 รูปแบบฟังก์ชันนี้มีหลายรูปแบบซึ่งอาจมีลักษณะดังนี้

1. Step Function เป็นฟังก์ชันที่ใช้ในโครงข่ายแบบชั้นเดียว (Single Layer) ซึ่งใช้ในการแปลงค่า net input ที่อยู่ในรูปของค่าต่อเนื่องให้อยู่ในรูปของไบนารี (Binary) คือ 1 และ 0 หรือ ไบโพลาร์ (Bipolar) คือ 1 และ -1 ฟังก์ชันนี้จะใช้ค่าเทรชโอล์ด (Threshold) ในการกำหนดการแปลงค่า ซึ่งเรียกฟังก์ชันนี้ว่า Threshold Function หรือ Heavisid Function ดังสมการที่ 2.4

$$f(\text{net}) = \begin{cases} 1 & \text{if } \text{net} \geq \theta \\ 0 & \text{if } \text{net} < \theta \end{cases} \quad (2.4)$$

โดยที่ net คือ ค่า net input ของเซลล์ประสาทเทียม
 θ คือ ค่าเทรชโอล์ด (Threshold)

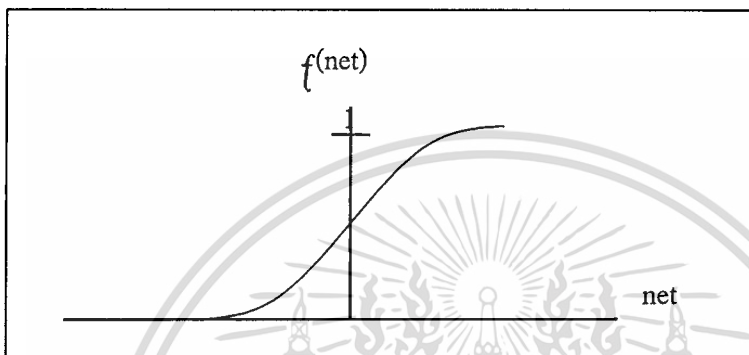


ภาพที่ 2.3 แสดงลักษณะของสมการ threshold

2. Sigmoid Function เป็นฟังก์ชันที่ใช้ในการแปลงค่า net input ให้อยู่ในช่วง 0 ถึง 1 ซึ่งเรียกว่า Binary Sigmoid function หรือ อยู่ในช่วง -1 ถึง 1 ซึ่งเรียกว่า Bipolar Sigmoid ดังแสดงในสมการที่ 2.5 และ 2.6

$$f(\text{net}) = 1/(1+e^{-\text{net}}) \quad (2.5)$$

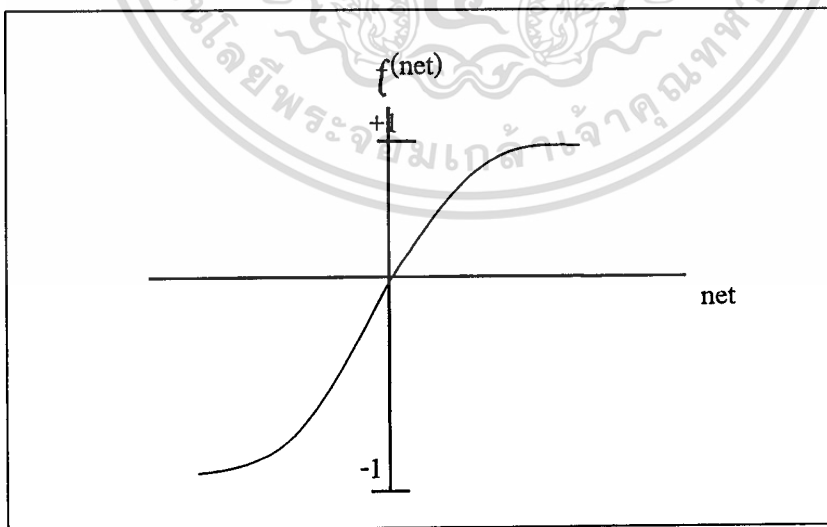
โดยที่ net คือ ค่า net input ของเซลล์ประสาทเทียม
e คือ ฟังก์ชันเอกซ์โปเนนเชียล



ภาพที่ 2.4 แสดงกราฟของฟังก์ชันนารีซิมมอยด์

$$f(\text{net}) = (e^{\text{net}} - e^{-\text{net}}) / (e^{\text{net}} + e^{-\text{net}}) \quad (2.6)$$

โดยที่ net คือ ค่า net input ของเซลล์ประสาทเทียม
e คือ ฟังก์ชันเอกซ์โปเนนเชียล



ภาพที่ 2.5 แสดงกราฟของฟังก์ชันโพล่าซิมมอยด์

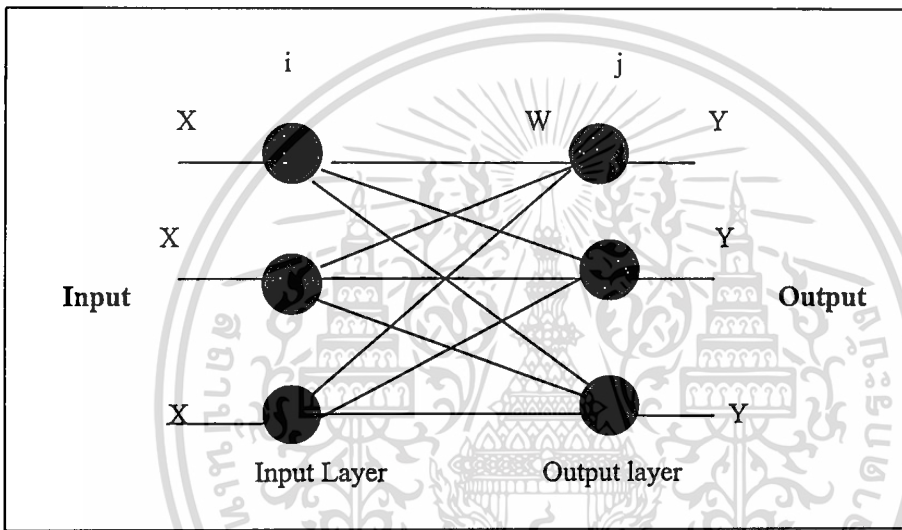
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3 สถาปัตยกรรมของโครงข่ายประสาทเทียม (Neural Network Architecture)

โครงข่ายประสาทเทียมประกอบด้วยเซลล์ประสาทเทียม หรือโหนดจำนวนมาก ที่เชื่อมต่อกัน ซึ่งการเชื่อมต่อนี้จะแบ่งออกเป็นกลุ่มย่อยๆ เรียกว่า ชั้น (Layer) โดยทั่วไปแล้ว แบ่งออกเป็น 2 แบบ คือ

2.3.1 โครงข่ายแบบชั้นเดียว (Single Layer)

เป็นลักษณะรูปแบบอย่างง่ายที่สุด ดังแสดงในภาพที่ 2.6

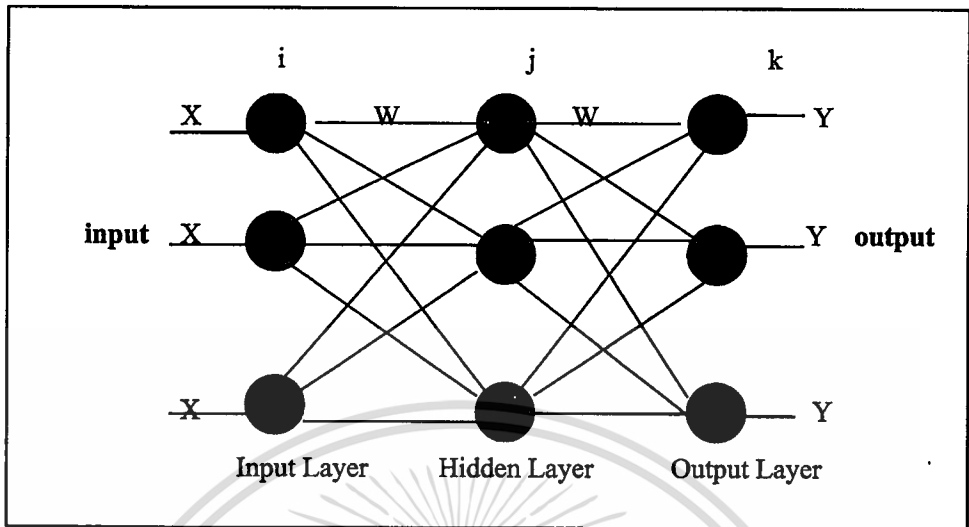


ภาพที่ 2.6 แสดงลักษณะ โครงข่ายแบบชั้นเดียว (Single Layer)

สำหรับโครงสร้างในภาพที่ 2.6 อาจจะมีการเชื่อมโยงบางส่วนถูกตัดทิ้งไป หรืออาจจะมีการป้อนกลับจาก output ไปยัง input ได้เรียกว่ามีลักษณะย้อนกลับ (Backward Learning) เพื่อส่งสัญญาณผิดพลาดกลับไปปรับแต่งค่าน้ำหนัก (weight) ให้ดีขึ้นเพื่อลดสัญญาณผิดพลาดที่เกิดขึ้น โครงสร้างอย่างนี้สามารถใช้ฝึกสอนกลุ่มอินพุตที่มีลักษณะง่ายๆ ได้

2.3.2 โครงข่ายแบบหลายชั้น (Multilayer)

สำหรับโครงการพัฒนาระบบงานที่ศึกษานี้ใช้โครงข่ายแบบหลายชั้น ลักษณะของโครงข่ายแบบชั้นเดียวถึงแม้จะทำงานได้ดีขึ้นจากนิวรอนเพียง 1 นิวรอน แต่ประสิทธิภาพในการทำงานจะเพิ่มมากขึ้นถ้าเราจัดรูปแบบโครงสร้างในลักษณะหลายๆ ชั้น (Multilayer) ดังแสดงในภาพที่ 2.7



ภาพที่ 2.7 แสดงลักษณะ โครงสร้างแบบหลายชั้น (Multilayer)

สำหรับประสิทธิภาพในการทำงานของโครงข่ายแบบหลายชั้นนี้จะไม่เพิ่มขึ้น หรือไม่แตกต่างจากโครงข่ายแบบชั้นเดียวเลย ถ้าในส่วนของฟังก์ชันกระตุ้นมีลักษณะเป็นเชิงเส้น ดังนั้น ในการเพิ่มประสิทธิภาพในการทำงานของโครงข่ายแบบหลายชั้น ทำได้โดยการกำหนดฟังก์ชันกระตุ้น (Activation Function) ในลักษณะไม่เชิงเส้น เช่น $f(\text{net}) = 1/(1+e^{-\text{net}})$ สำหรับโครงการพัฒนาระบบงานนี้ใช้ฟังก์ชันแบบนี้ และ $f(\text{net}) = (e^{\text{net}} - e^{-\text{net}}) / (e^{\text{net}} + e^{-\text{net}})$

2.4 การฝึกสอนโครงข่ายประสาทเทียม (Training)

โครงข่ายประสาทเทียมจะถูกฝึกให้เรียนรู้จากตัวอย่าง เพื่อที่จะได้มีความรู้ที่จะนำมาใช้ในการแก้ปัญหาต่อไป แบบจำลองต่างๆ มีขั้นตอนในการฝึกที่แตกต่างกัน สามารถแบ่งตามลักษณะของการเรียนรู้ได้ 2 ประเภท คือ

2.4.1 การเรียนรู้แบบมีผู้สอน (Supervised Learning)

ข้อมูลจะต้องประกอบด้วยตัวอย่างข้อมูลที่เรากำลังต้องการสอนและผลลัพธ์ที่เราต้องการให้โครงข่ายสร้างเมื่อมีที่นำข้อมูลเข้าในลักษณะเดียวกันมาเป็นข้อมูลเข้า (input) ขณะสอน โครงข่ายชนิดนี้จะกำหนดค่าผลลัพธ์เป้าหมายให้กับข้อมูลเข้าแต่ละตัว โครงข่ายจะนำค่าที่ผิดพลาดที่ได้จากการคำนวณโครงข่ายกับผลลัพธ์ที่ต้องการให้โครงข่ายสร้างนั้นมาเป็นค่าน้ำหนัก เพื่อปรับให้ได้ค่าข้อมูลออกตรงตามผลลัพธ์เป้าหมายที่ต้องการสำหรับโครงการพัฒนาระบบงานนี้ใช้การเรียนรู้แบบมีผู้สอน

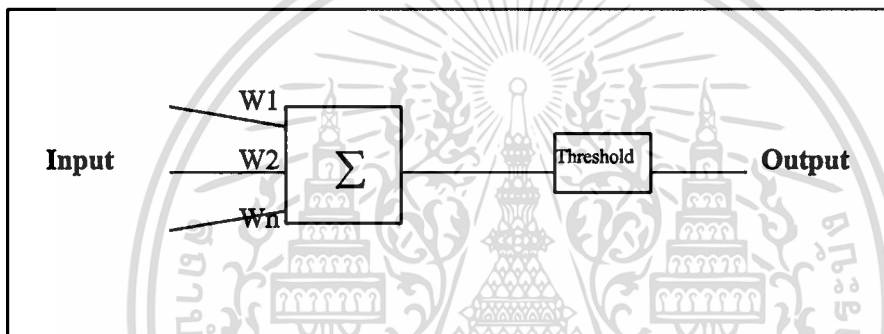
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.2 การเรียนรู้แบบไม่มีผู้สอน (Unsupervised Learning)

ลักษณะนี้จะสอนโครงข่ายด้วยการป้อนแบบข้อมูลอย่างต่อเนื่องเพียงอย่างเดียว โดยไม่มีการส่งค่าผลลัพธ์เป้าหมายให้กับข้อมูลเข้าแต่ละตัว การปรับค่าน้ำหนักใช้ข้อมูลที่นำมาสอนเป็นตัวปรับค่า โดยค่าน้ำหนักจะปรับตามกลุ่มที่ข้อมูลเข้า

2.5 เพอเซปตรอน (Perceptron)

ลักษณะของนิวรอนในยุคแรกๆ ของโครงข่ายนิวรอน เรียกว่า เพอเซปตรอน (Perceptron) ซึ่งมีลักษณะดังภาพที่ 2.8



ภาพที่ 2.8 ลักษณะของเพอเซปตรอน (Perceptron)

จะมีลักษณะเฉพาะ คือ ในส่วนของฟังก์ชันกระตุ้น (Activation Function) จะเป็นฟังก์ชันเทรชโลด (Threshold Function) ซึ่งจะให้ค่า 1 ออกมา ถ้าค่าผลรวมที่เข้ามามีค่ามากกว่าค่าที่กำหนดค่าหนึ่ง และจะให้ค่า 0 ออกมา ถ้าค่าผลรวมมีค่าน้อยกว่าค่าที่กำหนด

การฝึกสอนในช่วงแรกๆ ใช้ลักษณะเพียง 1 ชั้น ซึ่งประสบความสำเร็จมาก แต่ต่อมาได้ประสบปัญหาที่ลักษณะของเพอเซปตรอน 1 ชั้น ไม่สามารถที่จะแก้ปัญหาเกี่ยวกับการเอกคลูซิฟออร์ (XOR problem) ซึ่งปัญหานี้สามารถแก้ได้โดยใช้ลักษณะของเพอเซปตรอนหลายๆ ชั้น (Multilayer Perceptron) แต่วิธีการให้ได้มาซึ่งค่าน้ำหนัก (weight) ของตัวโครงข่ายเป็นลักษณะของการกำหนดค่าโดยใช้ความคิดขึ้นมาเอง ไม่ใช่ลักษณะของการฝึกสอน เนื่องจากสมัยนั้นยังไม่มีวิธีการฝึกสอนที่ได้ผลในส่วนของเพอเซปตรอนหลายๆ ชั้น

2.6 กฎเดลต้า (The Delta Rule)

เป็นกฎที่สำคัญซึ่งใช้ในการฝึกสอนเพอเซปตรอน (perceptron) และต่อมาได้ถูกขยายเพื่อนำไปใช้ในการฝึกสอนแก่โครงข่ายประสาทเทียมที่มีลักษณะหลายชั้น โดยกฎของเดลตา (Delta) เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปเผยแพร่บนสื่อออนไลน์ใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Rule) นี้จะถูกใช้ในการปรับน้ำหนัก (Weight) ของเพอเซปตรอนได้อย่างสมเหตุสมผล ประกอบด้วยสูตรดังนี้

$$\delta = (T-A) \quad (2.7)$$

โดย T เป็นค่า เอาต์พุต (output) ที่เราต้องการจากเพอเซปตรอน

A เป็นค่า เอาต์พุตที่ออกมาตามการป้อนอินพุต (input)

δ เป็นค่าผลต่างระหว่าง T กับ A

หลังจากที่ได้ค่า δ แล้ว ค่านี้จะถูกนำไปคูณกับค่าอินพุต แต่ละค่า เพื่อใช้ในการปรับค่าน้ำหนักที่เหมาะสมของแต่ละอินพุต นอกจากนั้นยังต้องถูกนำไปคูณกับค่าอัตราการเรียนรู้ (learning rate) เพื่อเพิ่มประสิทธิภาพในการฝึกสอนให้เร็วขึ้น ซึ่งสามารถเขียนเป็นสูตรได้ คือ

$$\Delta_i = n\delta X_i \quad (2.8)$$

ค่า Δ_i คือ ค่าที่จะต้องถูกนำไปใช้ในการปรับค่าน้ำหนักของแต่ละค่าน้ำหนัก

ค่า X_i คือ ค่า อินพุตเวกเตอร์ (input vector)

ค่า n คือ อัตราการเรียนรู้

ดังนั้น สูตรการปรับค่าน้ำหนักเขียนได้ดังนี้

$$W_i(n+1) = W_i(n) + \Delta_i \quad (2.9)$$

ค่า Δ_i เป็นค่าน้ำหนักที่ต้องปรับแต่ละอินพุตใดๆ

ค่า $W_i(n+1)$ เป็นค่าน้ำหนักหลังปรับ

ค่า $W_i(n)$ เป็นค่าน้ำหนักก่อนปรับ

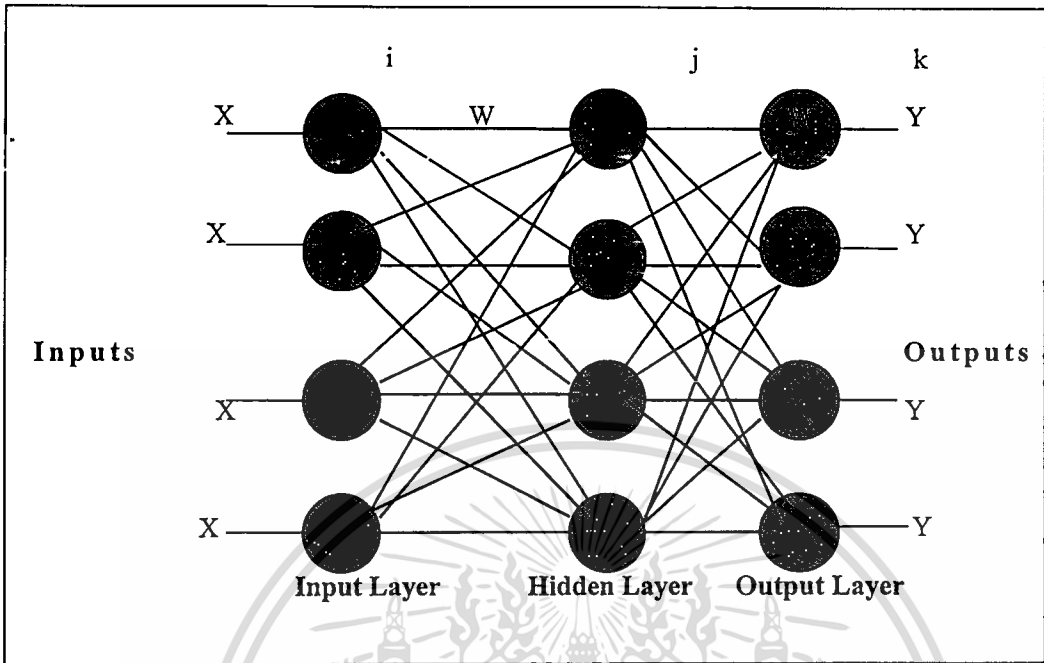
2.7 โครงข่ายนิเวรอนแบบแบคพรอพพาทเกชัน (Backpropagation Neural Network, BPN)

โครงข่ายแบคพรอพพาทเกชันเป็นโครงข่ายที่มีจำนวนชั้นมากกว่า 2 ชั้น คือ มีชั้นอินพุต

(Input Layer) ชั้นซ่อน (Hidden Layer) ชั้นเอาต์พุต (Output Layer) ดังภาพที่ 2.9

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 2.9 แสดงลักษณะ โครงข่ายนิวรอนแบบแบคพรอพพาเกชั่น

โครงข่ายนิวรอนแบบแบคพรอพพาเกชั่นตามภาพที่ 2.9 มีลักษณะหลักๆ คือ

1. มีชั้นของนิวรอนแทรกกระหว่างชั้นอินพุท และชั้นเอาต์พุท ซึ่งเรียกว่า ชั้นซ่อน (Hidden Layer)
2. การเชื่อมต่อระหว่างชั้นที่ติดกันเป็นแบบ Fully Connected Feedforward นั่น คือ ทุกๆ นิวรอนในชั้นอินพุทจะส่งสัญญาณ ไปยังทุกๆ นิวรอนในชั้นซ่อนชั้นแรก และทุกๆ นิวรอนในชั้นซ่อนชั้นแรกจะส่งสัญญาณ ไปยังทุกๆ นิวรอนในชั้นต่อๆ ไป จนในที่สุดทุกๆ นิวรอนในชั้นซ่อนชั้นสุดท้ายจะส่งสัญญาณ ไปยังทุกๆ นิวรอนในชั้นเอาต์พุท
3. ชั้นอินพุทไม่มีการประมวลผล คือ ทำหน้าที่ในการรับสัญญาณเข้าแล้วกระจาย ออกไปยังนิวรอนในชั้นต่อไปเท่านั้น ส่วนชั้นซ่อน และชั้นเอาต์พุทนั้นมีการประมวลผล โครงข่ายนิวรอนเรียกได้ว่า เป็นแมปปีงเน็ตเวิร์ค (Mapping Network) ถ้าหากโครงข่ายสามารถคำนวณฟังก์ชันในบางลักษณะที่กำเนิดความสัมพันธ์ระหว่างอินพุท และเอาต์พุทในโครงข่าย สิ่งที่ทำให้การแมปปีงถูกต้องนั้นจำเป็นต้องมีการฝึกหัดโครงข่าย และการฝึกหัดโครงข่ายโดยอาศัยหลักการแพร่ย้อนกลับนั้น แบ่งออกได้เป็น 3 ขั้นตอน คือ

1. แพร่สัญญาณที่ชั้นอินพุท ไปยังชั้นซ่อนแรก และแพร่สัญญาณ ไปยังชั้นซ่อนถัดไป จนถึงชั้นเอาต์พุท โดยผ่านฟังก์ชันกระตุ้น
 2. คำนวณค่าผิดพลาดของค่าเอาต์พุทกับค่าเป้าหมาย และแพร่ย้อนกลับค่าผิดพลาดเหล่านี้
- เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ไปยังชั้นซ่อน

3. ปรับแต่งค่าน้ำหนักที่ชั้นเอาต์พุท และชั้นซ่อน เพื่อลดค่าผิดพลาด โดยทำให้เอาต์พุทมีค่าใกล้เคียง หรือเท่ากับค่าเป้าหมาย

2.7.1 การฝึกหัดโครงข่ายด้วยวิธีการแพร่ย้อนกลับ

พิจารณาภาพที่ 2.9 ซึ่งเป็นโครงข่ายขนาด 3 ชั้น ให้มีอินพุทเวกเตอร์ $X = (X_1, X_2, \dots, X_n)$ อินพุท X นี้จะถูกป้อนให้กับชั้นอินพุทของโครงข่าย และที่ชั้นอินพุทนี้จะกระจายค่าไปยังนิวรอนในชั้นซ่อน ที่ชั้นซ่อนนี้เองจะทำการคำนวณค่าเน็ต (net^h_j) สมการหาค่าเน็ตที่นิวรอน j ในชั้นซ่อนคือ

$$net^h_j = \sum_{i=1}^n W_{ji}^h X_i + \theta_{h_j} \quad (2.10)$$

เมื่อ W_{ji}^h คือค่าน้ำหนักที่ชั้นซ่อนตรงจุดต่อจากโหนดอินพุท i ไปยังนิวรอน j และ θ_{h_j} คือ ค่าไบแอส (Bias) เมื่อได้ค่าเน็ตแล้วขั้นถัดไปคือ คำนวณค่าเอาต์พุทของนิวรอนในชั้นซ่อน (o_j) โดยคำนวณได้ ดังนี้

$$o_j = f^h_j(net^h_j) \quad (2.11)$$

ใช้สมการทั้งสองคำนวณค่าในชั้นซ่อนทุกๆ ชั้น จากนั้นจึงทำการคำนวณค่าเน็ต (net^o_k) และค่าเอาต์พุทที่ชั้นเอาต์พุท (o_k) เมื่อ W_{kj}^o คือค่าน้ำหนักที่ชั้นซ่อนตรงจุดต่อจากโหนดอินพุท j ไปยังนิวรอน k และ θ_{o_k} คือ ค่าไบแอส (Bias) โดยใช้สมการที่คล้ายคลึงกัน คือ

$$net^o_k = \sum_{j=1}^L W_{kj}^o o_j + \theta_{o_k} \quad (2.12)$$

$$o_k = f^o_k(net^o_k) \quad (2.13)$$

2.7.2 การหาค่าผิดพลาดและการปรับน้ำหนักที่ชั้นเอาต์พุท
สมการปรับค่าน้ำหนักที่ชั้นเอาต์พุทแสดงได้ดังนี้

$$w^o_{kj}(t+1) = w^o_{kj}(t) + \Delta w^o_{kj}(t) \quad (2.14)$$

$$\Delta w^o_{kj} = \eta (y_k - o_k) f'_k(net^o_k) o_j \quad (2.15)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$o_j = f_j^h(\text{net}_j^h) \quad (2.16)$$

$$f_j^h(\text{net}_j^h) = (1 + \exp(-\text{net}_j^h))^{-1} \quad (2.17)$$

$$f_k' = f_k^o(1 - f_k^o) = o_k(1 - o_k) \quad (2.18)$$

$$w_{kj}^o(t+1) = w_{kj}^o(t) + \eta (y_k - o_k) o_k(1 - o_k) o_j \quad (2.19)$$

เขียนเป็นสมการให้ดูง่ายขึ้นได้คือ

$$\delta_k^o = (y_k - o_k) f_k^o'(net_k^o) \quad (2.20)$$

$$w_{kj}^o(t+1) = w_{kj}^o(t) + \eta \delta_k^o o_j \quad (2.21)$$

- η คือ อัตราการเรียนรู้ โดยปกติจะมีค่าเป็นบวก และน้อยกว่า 1
 y_k คือ ค่าเป้าหมาย
 o_j คือ ค่าเอาต์พุตของนิวรอนในชั้นซ่อน
 o_k คือ ค่าเอาต์พุตที่ชั้นเอาต์พุต
 $w_{kj}^o(t)$ คือ ค่าน้ำหนักก่อนถูกปรับค่า
 $w_{kj}^o(t+1)$ คือ ค่าน้ำหนักหลังถูกปรับค่า
 $\exp()$ คือ ฟังก์ชันเอกซ์โปเนนเชียล

2.7.3 การหาค่าผิดพลาดและการปรับน้ำหนักที่ชั้นซ่อน

การปรับแต่งค่าน้ำหนักที่ชั้นซ่อนจะต้องทำการคำนวณค่าผิดพลาดที่ชั้นซ่อนก่อนโดยใช้สมการ

การ

$$\delta_j^h = f_k^h'(net_j^h) \sum \delta_j^o w_{kj}^o \quad (2.22)$$

จากนั้นจึงจะสามารถปรับแต่งค่าน้ำหนักโดยสามการที่คล้ายคลึงกับสมการปรับแต่งค่าน้ำหนักที่ชั้นเอาต์พุต คือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$w_{ji}^h(t+1) = w_{ji}^h(t) + \eta \delta_j^h x_i \quad (2.23)$$

- η คือ อัตราการเรียนรู้ โดยปกติจะมีค่าเป็นบวก และน้อยกว่า 1
- y_k คือ ค่าเป้าหมาย
- $w_{kj}^o(t)$ คือ ค่าน้ำหนักก่อนถูกปรับค่า
- $w_{kj}^o(t+1)$ คือ ค่าน้ำหนักหลังถูกปรับค่า
- x_i คือ ค่าอินพุทเวกเตอร์



บทที่ 3

วิธีการดำเนินการศึกษา

ในการศึกษาโครงการนี้ จะเป็นการนำทฤษฎีโครงข่ายประสาทเทียมแบบแบคพรอพพาเกชัน (Backpropagation Neural Network) มาพัฒนาเป็นโปรแกรมคอมพิวเตอร์เพื่อประยุกต์ใช้ในการทำประมาณการผลการจัดเก็บภาษีมูลค่าเพิ่ม โดยแบ่งการดำเนินการศึกษาออกเป็น 4 ส่วน คือ

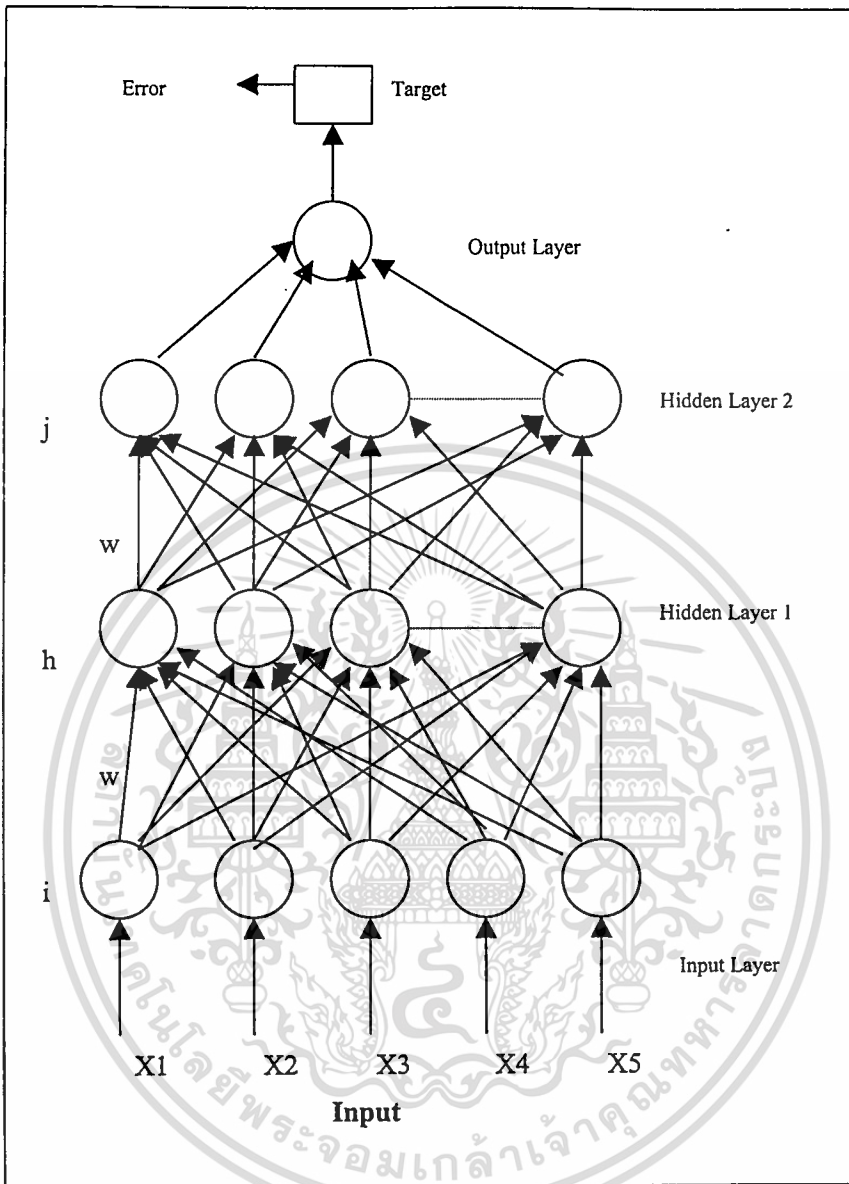
- 3.1 ส่วนของโครงสร้างแบบจำลอง
- 3.2 ส่วนของอัลกอริทึม
- 3.3 ส่วนของฐานข้อมูล
- 3.4 ส่วนของโปรแกรม

3.1 ส่วนของโครงสร้างแบบจำลอง

โครงสร้างแบบจำลองของโครงข่ายประสาทเทียมแบบพรอพพาเกชันที่ใช้ในโครงการนี้ ประกอบด้วย 4 ชั้น คือ

1. ชั้นอินพุท (Input Layer) มี 5 นิวรอน (Neuron)
2. ฮิดเดนชั้นที่ 1 (Hidden Layer 1) มี 20 นิวรอน (Neuron)
3. ฮิดเดนชั้นที่ 2 (Hidden Layer 2) มี 20 นิวรอน (Neuron)
4. ชั้นเอาต์พุท (Output Layer) มี 1 นิวรอน (Neuron)

ดังแสดงในภาพที่ 3.1



ภาพที่ 3.1 แสดง โครงสร้างแบบจำลองโครงข่ายประสาทเทียม
แบบแบคพรอพพาเกชั่นที่ใช้ในโครงการนี้

3.2 ส่วนของอัลกอริทึม

อัลกอริทึมของโครงข่ายประสาทเทียมแบบแบคพรอพพาเกชั่น (Backpropagation Neural Network) ที่ใช้ในการพัฒนาโปรแกรมสำหรับโครงการนี้มีขั้นตอน ดังนี้

ขั้นตอนที่ 0 ทำการแปลงค่าที่อ่านได้จากฐานข้อมูลให้เป็นค่าที่อยู่ระหว่าง 0 ถึง 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นตอนที่ 1 กำหนดค่าตัวแปรเวต (weight) โดยการสุ่มค่าตัวเลขระหว่าง -1 ถึง +1

ขั้นตอนที่ 2 ถ้าข้อความนี้เป็นจริง คือ While (total_error > v_error) ให้ทำขั้นตอนที่ 3 ถึง 21 (สำหรับโครงการนี้กำหนดให้ v_error = 0.05)

ขั้นตอนที่ 3 แต่ละนิวรอนบนชั้นอินพุท (Input Layer) คือ input[1] ถึง input[5] ให้รับค่าสัญญาณอินพุทจากไฟล์ที่ใช้ฝึกสอนที่อ่านเข้ามา และกระจายสัญญาณไปสู่ฮิดเดนชั้นที่ 1 (Hidden Layer 1)

ขั้นตอนที่ 4 คำนวณหาค่าเน็ต (neth1) ของแต่ละนิวรอนบนฮิดเดนชั้นที่ 1 (Hidden Layer 1) เป็นค่าที่ได้จากการหาผลรวมของการคูณกันของตัวแปรเวต (wh1) และอินพุทที่เข้ามาของแต่ละคู่ ซึ่งมีสมการคือ

$$neth1[h] = \sum_{i=1}^5 (input[i] * wh1[i][h]) \quad (3.1)$$

wh1 คือ ค่าน้ำหนักของชั้น Hidden layer 1

h คือ จำนวนนิวรอนของชั้น Hidden layer 1 มีค่าตั้งแต่ 1 ถึง 20

i คือ จำนวนนิวรอนของชั้น Input layer มีค่าตั้งแต่ 1 ถึง 5

input คือ ข้อมูลอินพุทจากชั้น Input layer

ขั้นตอนที่ 5 คำนวณหาค่าเอาต์พุท (output1) ของแต่ละนิวรอนบนฮิดเดนชั้นที่ 1 (Hidden Layer 1) โดย นำผลลัพธ์ จาก neth1[h] เข้าสู่ฟังก์ชันแอคทีเวชัน (Activation Function) ซึ่งมีสมการ คือ

$$output1[h] = 1.0 / (1.0 + \exp(-neth1[h])) \quad (3.2)$$

h คือ จำนวนนิวรอนของชั้น Hidden layer 1 มีค่าตั้งแต่ 1 ถึง 20

neth1 คือ ค่าเน็ตของชั้น Hidden layer 1

exp() คือ ฟังก์ชันเอกซ์โปเนนเชียล

ขั้นตอนที่ 6 คำนวณหาค่าเน็ต (nethi2) ของแต่ละนิวรอนบนฮิดเดนชั้นที่2 (Hidden Layer 2) เป็นค่าที่ได้จากการหาผลรวมของการคูณกันของตัวแปรเวด (wh2) และอินพุทที่เข้ามา (outputhi1(h)) ของแต่ละคู่ ซึ่งมีสมการคือ

$$\text{nethi2}[j] = \sum_{i=1}^{20} (\text{outputhi1}[i] * \text{wh2}[i][j]) \quad (3.3)$$

j คือ จำนวนนิวรอนของชั้น Hidden layer 2 มีค่าตั้งแต่ 1 ถึง 20
 outputhi1 คือ เอาต์พุทของชั้น Hidden layer 1
 wh2 คือ ค่าน้ำหนักของชั้น Hidden layer 2

ขั้นตอนที่ 7 คำนวณหาค่าเอาต์พุท (outputhi2) ของแต่ละนิวรอนบนฮิดเดนชั้นที่2 (Hidden Layer 2) โดย นำผลลัพธ์ จาก nethi2[j] เข้าสู่ฟังก์ชันแอคทีเวชัน (Activation Function) ซึ่งมีสมการ คือ

$$\text{outputhi2}[j] = 1.0 / (1.0 + \exp(-\text{nethi2}[j])) \quad (3.4)$$

j คือ จำนวนนิวรอนของชั้น Hidden layer 2 มีค่าตั้งแต่ 1 ถึง 20
 nethi2 คือ ค่าเน็ตของชั้น Hidden layer 2
 exp() คือ ฟังก์ชันเอกซ์โปเนนเชียล

ขั้นตอนที่ 8 คำนวณหาค่าเน็ต (netout) ของแต่ละนิวรอนบนชั้นเอาต์พุท (Output Layer) เป็นค่าที่ได้จากการหาผลรวมของการคูณกันของตัวแปรเวด (wo) และอินพุทที่เข้ามา (outputhi2(j)) ของแต่ละคู่ ซึ่งมีสมการคือ

$$\text{netout} = \sum_{i=1}^{20} (\text{outputhi2}[i] * \text{wo}[i]) \quad (3.5)$$

i คือ จำนวนนิวรอนของชั้น Input layer มีค่าตั้งแต่ 1 ถึง 5
 outputhi2 คือ ค่าเอาต์พุทของชั้น Hidden layer 2 มีค่าตั้งแต่ 1 ถึง 20
 wo คือ ค่าน้ำหนักของชั้น Output layer

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นตอนที่ 9 คำนวณหาค่าเอาต์พุต (output) ของแต่ละนิวรอนบนชั้นเอาต์พุต (Output Layer) โดยนำผลลัพธ์ จาก netout เข้าสู่ฟังก์ชันแอคทีเวชัน (Activation Function) ซึ่งมีสมการ คือ

$$\text{output} = 1.0 / (1.0 + \exp(-\text{netout})) \quad (3.6)$$

netout คือ ค่าเน็ตของชั้น Output layer

exp() คือ ฟังก์ชันเอกซ์โปเนนเชียล

ขั้นตอนที่ 10 คำนวณหาค่าผิดพลาดจากสมการ

$$\text{errorp} = (1/2) * (\text{target} - \text{output})^2 \quad (3.7)$$

target คือ ค่าเป้าหมาย

output คือ ค่าเอาต์พุตของชั้น Output layer

ขั้นตอนที่ 11 คำนวณหาค่าเดลต้า (deltaout) ของชั้น output layer ซึ่งมีสมการ คือ

$$\text{deltaout} = (\text{target} - \text{output}) * ((1 - \text{output}) * \text{output}) \quad (3.8)$$

target คือ ค่าเป้าหมาย

output คือ ค่าเอาต์พุตของชั้น Output layer

ขั้นตอนที่ 12 คำนวณหาเดลต้าเวต (delta_w) ที่ต้องใช้เป็นค่าสำหรับปรับปรุงเวตที่ชั้นเอาต์พุต (wo) ซึ่งมีสมการ คือ

$$\text{delta_wo}[j] = \text{deltaout} * \text{outputhi2}[j] \quad (3.9)$$

j คือ จำนวนนิวรอนของชั้น Hidden layer 2 มีค่าตั้งแต่ 1 ถึง 20

deltaout คือ ค่าเดลต้าของชั้น Output layer

outputhi2 คือ ค่าเอาต์พุตของชั้น Hidden layer 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นตอนที่ 13 คำนวณหาค่าเคลด้า (deltahi2) ของแต่ละฮิดเดนชั้นที่ 2 ซึ่งมีสมการ คือ

$$\text{deltahi2}[j] = (\text{deltaout} * \text{wo}[j]) * ((1 - \text{outputhi2}[j]) * \text{outputhi2}[j]) \quad (3.10)$$

j คือ จำนวนนิวรอนของชั้น Hidden layer 2 มีค่าตั้งแต่ 1 ถึง 20

deltaout คือ ค่าเคลด้าของชั้น Output layer

outputhi2 คือ ค่าเอาต์พุทของชั้น Hidden layer 2

wo คือ ค่าน้ำหนักของชั้น Output layer

ขั้นตอนที่ 14 คำนวณหาเคลด้าเวต (delta_wh2) ที่ต้องใช้เป็นค่าสำหรับปรับปรุงเวตฮิดเดนชั้นที่ 2 (wh2) ซึ่งมีสมการ คือ

$$\text{delta_wh2}[h][j] = \text{deltahi2}[j] * \text{outputhi1}[h] \quad (3.11)$$

h คือ จำนวนนิวรอนของชั้น Hidden layer 1 มีค่าตั้งแต่ 1 ถึง 20

j คือ จำนวนนิวรอนของชั้น Hidden layer 2 มีค่าตั้งแต่ 1 ถึง 20

deltahi2 คือ ค่าเคลด้าของชั้น Hidden layer 2

outputhi1 คือ ค่าเอาต์พุทของชั้น Hidden layer 1

ขั้นตอนที่ 15 คำนวณหาค่า deltahi1 ของแต่ละฮิดเดนชั้นที่ 1 ซึ่งมีสมการ คือ

20

$$\text{deltahi1}[h] = \left(\sum_{j=1}^{20} (\text{deltahi2}[j] * \text{wh2}[h][j]) \right) * ((1 - \text{outputhi1}[h]) * \text{outputhi1}[h]) \quad (3.12)$$

$j=1$

h คือ จำนวนนิวรอนของชั้น Hidden layer 1 มีค่าตั้งแต่ 1 ถึง 20

j คือ จำนวนนิวรอนของชั้น Hidden layer 2 มีค่าตั้งแต่ 1 ถึง 20

deltahi2 คือ ค่าเคลด้าของชั้น Hidden layer 2

outputhi1 คือ ค่าเอาต์พุทของชั้น Hidden layer 1

wh2 คือ ค่าน้ำหนักของชั้น Hidden layer 2

ขั้นตอนที่ 16 คำนวณหาค่าเคลต้วเวด (δ_{wh1}) ที่ต้องใช้เป็นค่าสำหรับปรับปรุงเวดฮิตเดนชั้นที่ 1 ($wh1$) ซึ่งมีสมการ คือ

$$\delta_{wh1}[i][h] = \delta_{tahi1}[h] * input[i] \quad (3.13)$$

i คือ จำนวนนิวรอนของชั้น Input layer มีค่าตั้งแต่ 1 ถึง 5

h คือ จำนวนนิวรอนของชั้น Hidden layer 1 มีค่าตั้งแต่ 1 ถึง 20

δ_{tahi1} คือ ค่าเคลต้วของชั้น Hidden layer 1

$input$ คือ ข้อมูลอินพุตจากชั้น Input layer

ขั้นตอนที่ 17 ปรับค่าเวดที่ชั้นเอาต์พุต (w_o) ซึ่งมีสมการ คือ

$$w_o[j]_{new} = w_o[j]_{old} + \delta_{wo}[j] \quad (3.14)$$

w_o_{new} คือ ค่านำหนักหลังปรับค่า

w_o_{old} คือ ค่านำหนักก่อนปรับค่า

δ_{wo} คือ ค่าเคลต้วเวดของชั้น Output layer

j คือ จำนวนนิวรอนของชั้น Hidden layer 2 มีค่าตั้งแต่ 1 ถึง 20

ขั้นตอนที่ 18 ปรับค่าเวดที่ฮิตเดนชั้นที่ 2 ($wh2$) ซึ่งมีสมการ คือ

$$wh2[h][j]_{new} = wh2[h][j]_{old} + \delta_{wh2}[h][j] \quad (3.15)$$

h คือ จำนวนนิวรอนของชั้น Hidden layer 1 มีค่าตั้งแต่ 1 ถึง 20

j คือ จำนวนนิวรอนของชั้น Hidden layer 2 มีค่าตั้งแต่ 1 ถึง 20

δ_{wh2} คือ ค่าเคลต้วเวดของชั้น Hidden layer 2

$wh2$ คือ ค่านำหนักของชั้น Hidden layer 2

ขั้นตอนที่ 19 ปรับค่าเวดที่ฮิตเดนชั้นที่ 1 ($wh1$) ซึ่งมีสมการ คือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$wh1[i][h]_{new} = wh1[i][h]_{old} + \delta_{wh1}[i][h] \quad (3.16)$$

h คือ จำนวนนิวรอนของชั้น Hidden layer 1 มีค่าตั้งแต่ 1 ถึง 20

j คือ จำนวนนิวรอนของชั้น Hidden layer 2 มีค่าตั้งแต่ 1 ถึง 20

δ_{wh1} คือ ค่าเดลต้าเวตของชั้น Hidden layer 1

$wh1$ คือ ค่าน้ำหนักของชั้น Hidden layer 1

ขั้นตอนที่ 20 อ่านค่าอินพุตจากไฟล์ใช้ฝึกสอนจนกว่าจะจบไฟล์ แล้วคำนวณหาผลรวมของค่าผิดพลาดที่เกิดขึ้น ($total_error$) ตามสมการ

$$total_error = total_error + errorp \quad (3.17)$$

$errorp$ คือ ค่าผิดพลาดที่ได้จากสมการที่ (3.7)

ขั้นตอนที่ 21 ทดสอบเงื่อนไขตามที่กำหนด คือ ถ้า $total_error \leq 0.05$ ให้หยุดการฝึกสอน ถ้าไม่ใช่หรือเป็นเท็จให้กลับไปทำตั้งแต่ขั้นตอนที่ 3 ใหม่

3.3 ส่วนของฐานข้อมูล

ในการศึกษาโครงการนี้ได้ทำการพัฒนาฐานข้อมูลสำหรับจัดเก็บข้อมูลผลการจัดเก็บภาษีมูลค่าเพิ่มที่จัดเก็บได้จริงทั่วประเทศ ตั้งแต่ เดือน ม.ค. 2536 ถึง เดือน เม.ย. 2542 เพื่อเป็นข้อมูลสำหรับการฝึกสอน (Train) เนื่องจากปัจจุบันกรมสรรพากรได้มีการบันทึกข้อมูลจากแบบภาษีมูลค่าเพิ่ม (ภ.พ.30) ในแต่ละเดือนภาษีจากทั่วประเทศเก็บไว้ในฐานข้อมูล ในการศึกษาโครงการนี้จึงได้ทำการออกแบบและพัฒนาฐานข้อมูล โดยใช้โปรแกรมเอกเซต (MS Access '97) เป็นฐานข้อมูลที่จำลองขึ้นมา เพื่อเป็นแนวทางนำไปประยุกต์ใช้กับงานจริง ในฐานข้อมูลประกอบไปด้วย 1 ตารางซึ่งประกอบด้วยฟิลด์ ดังนี้คือ

ชื่อ	ชนิดข้อมูล	รายละเอียด
Month_train	Text	เดือนภายในปี
Year_train	Text	ปีภายในปี
Money_train	Number	จำนวนเงิน

ตารางที่ 3.1 ตารางแสดงฟิลด์ในฐานข้อมูล

3.4 ส่วนของโปรแกรม

การศึกษาโครงงานนี้ได้ใช้โปรแกรมวิซวลเบสิก เวอร์ชัน 5.0 (Visual basic v 5.0) ใช้เป็นเครื่องมือสำหรับพัฒนาโปรแกรม เพื่อนำทฤษฎีโครงข่ายประสาทเทียมแบบแบคพรอพพาเกชั่น มาประยุกต์ใช้ในการทำประมาณการผลการจัดเก็บภาษีมูลค่าเพิ่ม เนื่องเป็นโปรแกรมที่มีเครื่องมือที่ใช้ช่วยในการออกแบบหน้าจอ และแสดงผลแบบกราฟ ทำให้พัฒนาโปรแกรมได้รวดเร็ว โปรแกรมที่พัฒนาเพื่อใช้สำหรับในการศึกษาโครงงานนี้ แบ่งเป็นแต่ละส่วนดังนี้

3.4.1 หน้าจอเมนูหลัก

เมื่อเริ่มเรียกใช้งาน โปรแกรมสำหรับโครงงานนี้ จะแสดงหน้าจอเมนูหลักสำหรับผู้ที่จะเลือกใช้งานโปรแกรม ซึ่งประกอบไปด้วยปุ่ม 3 ปุ่ม คือ ปุ่มสำหรับใช้เรียกหน้าจอสำหรับบันทึกข้อมูลผลการจัดเก็บภาษีมูลค่าเพิ่ม ปุ่มสำหรับเรียกใช้หน้าจอสำหรับทำการประมวลผลเพื่อหาค่าประมาณการ และปุ่มที่ใช้สำหรับยกเลิกการใช้งานโปรแกรม



ภาพที่ 3.2 แสดงหน้าจอเมนูหลัก

3.4.2 หน้าจอสำหรับทำการบันทึกข้อมูลผลการจัดเก็บภาษีมูลค่าเพิ่ม

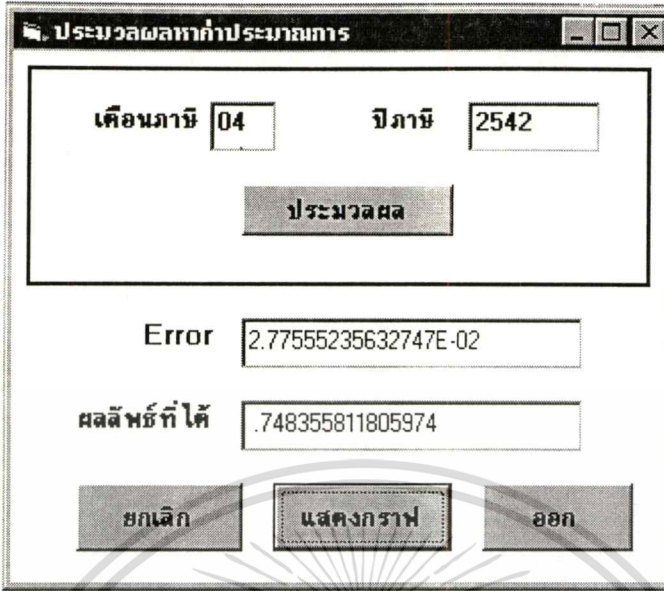
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน้าจอนี้จะใช้สำหรับบันทึกข้อมูลผลการจัดเก็บภาษีมูลค่าเพิ่มที่จัดเก็บได้จริงในแต่ละเดือนภาษีทั่วประเทศ เพื่อใช้เป็นข้อมูลในการฝึกสอน แต่ข้อมูลนี้จะต้องถูกนำไปแปลงค่าให้เป็นค่าตัวเลขระหว่าง 0 ถึง 1 อีกทีตอนที่ทำการประมวลผลเพื่อหาค่าประมาณการ จึงจะใช้เป็นข้อมูลสำหรับทำการฝึกสอนได้

ภาพที่ 3.3 แสดงหน้าจอสำหรับบันทึกข้อมูลผลการจัดเก็บภาษีมูลค่าเพิ่ม

3.4.3 หน้าจอสำหรับประมวลผลหาค่าประมาณการ

หน้าจอนี้ใช้สำหรับทำการประมวลผลหาค่าประมาณการภาษีมูลค่าเพิ่ม โดยใช้ทฤษฎีโครงข่ายประสาทเทียมแบบเบคพรอพพาเกชัน ตามขั้นตอนที่กำหนดในอัลกอริธึม (ข้อ 3.2 ส่วนอัลกอริธึม) โดยโปรแกรมจะทำการอ่านข้อมูลจากฐานข้อมูลผลการจัดเก็บภาษีมูลค่าเพิ่ม หลังเดือน/ปี ภาษีที่ต้องการทำการประมาณการ เช่น ต้องการทำการประมาณการเดือนภาษี 01 (ม.ค.) ปีภาษี 2542 โปรแกรมก็จะทำการอ่านข้อมูลก่อนเดือน 01 ปี 2542 มาทำการแปลงค่าให้เป็นค่าตัวเลขที่อยู่ระหว่าง 0 ถึง 1 เพื่อที่จะใช้เป็นข้อมูลสำหรับทำการฝึกสอน ตามอัลกอริธึมที่กำหนด ซึ่งโปรแกรมจะทำการแสดงผลลัพธ์ที่ได้จากการประมวลผลตามอัลกอริธึม และแสดงค่าผิดพลาดที่ได้จากการประมวลผลตามอัลกอริธึม

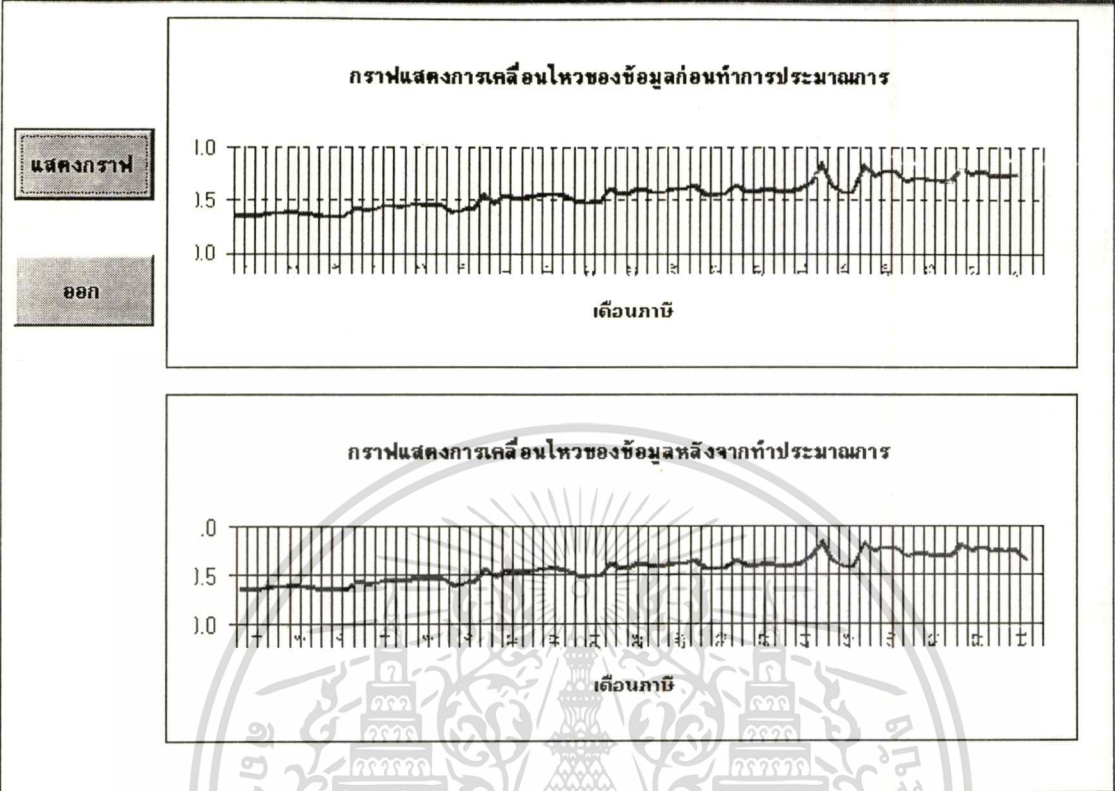


ภาพที่ 3.4 แสดงหน้าจอประมวลผลหาค่าประมาณการ

3.4.4 หน้าจอแสดงกราฟ

หน้าจอส่วนนี้ใช้สำหรับแสดงผลลัพธ์ที่ได้จากโปรแกรมในรูปแบบของกราฟ เพื่อเปรียบเทียบให้เห็นถึงการเคลื่อนไหวของข้อมูลที่ได้จากผลการจัดเก็บภาษีมูลค่าเพิ่มที่จัดเก็บได้จริง กับผลลัพธ์ที่ได้จากการทำประมาณการด้วยโปรแกรมที่พัฒนาใช้สำหรับการศึกษาโครงการนี้ ว่ามีความใกล้เคียงกันมากน้อยแค่ไหน ดังแสดงในภาพที่ 3.5

กราฟแสดงการเคลื่อนไหวของข้อมูล



ภาพที่ 3.5 แสดงหน้าจอสำหรับแสดงผลด้วยกราฟ

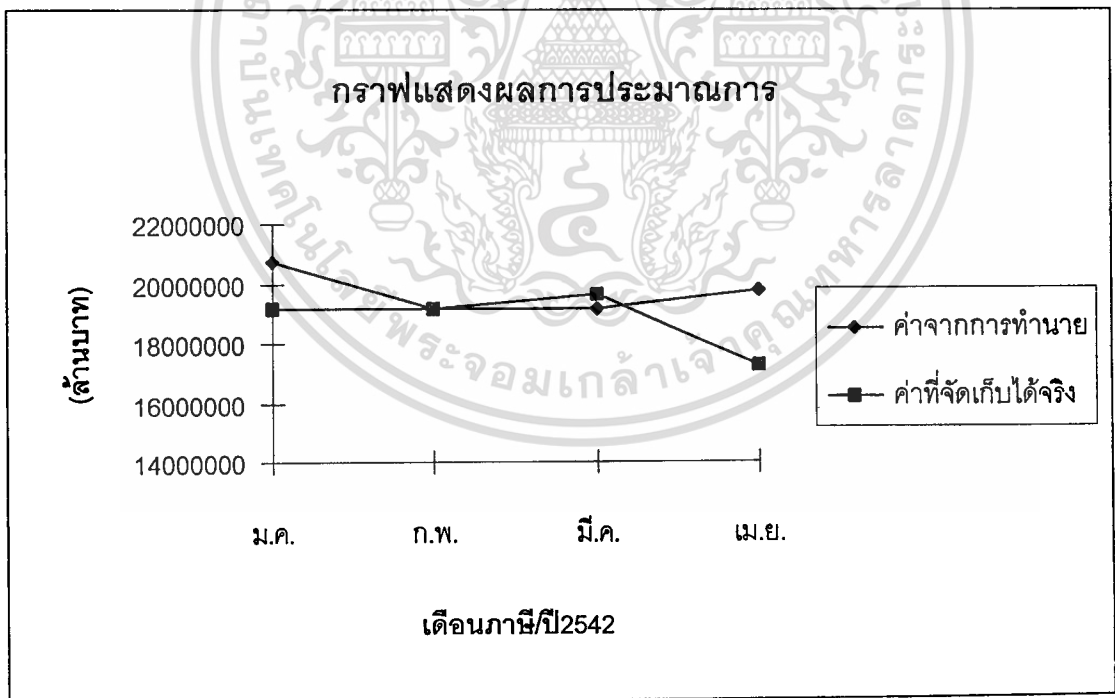
สำหรับเนื้อหาในบทนี้ ได้แสดงให้เห็นถึงวิธีการดำเนินการศึกษาโครงการนี้ว่า ได้มีขั้นตอนดำเนินการศึกษา และพัฒนาโปรแกรมเพื่อนำมาประยุกต์ใช้ในการทำประมวลผลการจัดเก็บภาษีมูลค่าเพิ่มอย่างไร ในส่วนของซอร์สโค้ดสำหรับโปรแกรมทั้งหมด ได้แสดงเอาไว้ในภาคผนวกท้ายเล่ม รวมทั้งข้อมูลผลการจัดเก็บภาษีที่จัดเก็บได้จริงที่ใช้เป็นข้อมูลสำหรับทำการฝึกหัดก็ได้แสดงเอาไว้ที่ภาคผนวกท้ายเล่มด้วยเช่นกัน

บทที่ 4

ผลการดำเนินการศึกษา

ผลที่ได้จากการพัฒนาโปรแกรมสำหรับการทำประมาณการภาษีมูลค่าเพิ่ม โดยใช้โครงข่ายประสาทเทียมแบบแบคพรอพพาเกชั่น โดยใช้ข้อมูลผลการจัดเก็บภาษีมูลค่าเพิ่มตั้งแต่ เดือน ม.ค. 2536 ถึง เดือน เม.ย. 2542 เป็นข้อมูลสำหรับการฝึกสอน และได้ทดลองทำการประมาณการผลการจัดเก็บภาษีมูลค่าเพิ่มจำนวน 4 เดือน คือ เดือน ม.ค. 2542 ถึง เดือน เม.ย. 2542 ผลที่ได้จากการดำเนินการศึกษา สามารถนำมาสรุปแสดงได้เป็น 2 ส่วน ดังนี้

4.1 ส่วนของกราฟแสดงผล



ภาพที่ 4.1 กราฟแสดงผลที่ได้จากการทำนาย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 ส่วนของข้อมูลที่เป็นตัวเลขเพื่อใช้เปรียบเทียบกัน

เดือนภาษี	ผลจากการคำนวณ	ค่าที่เก็บได้จริง
ม.ค.	20,684,928	19,138,487
ก.พ.	19,138,324	19,119,631
มี.ค.	19,127,781	19,597,121
เม.ย.	19,757,314	17,277,230

หน่วย ล้านบาท

ตารางที่ 4.1 แสดงการเปรียบเทียบผลที่ได้จากการทำนายกับค่าเป้าหมาย



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

สรุปผลการดำเนินการศึกษาและข้อเสนอแนะ

5.1 สรุปผลที่ได้จากการดำเนินการศึกษา

การทำประมาณการผลการจัดเก็บภาษีมูลค่าเพิ่มโดยใช้โครงข่ายประสาทเทียมแบบแบคพรอปพากชัน (Backpropagation Neural Network) การดำเนินการศึกษาในครั้งนี้ได้ใช้ข้อมูลผลการจัดเก็บภาษีมูลค่าเพิ่มตั้งแต่ เดือน ม.ค. 2536 ถึง เดือน เม.ย. 2542 เป็นข้อมูลสำหรับการฝึกสอนนั้น จากผลที่ได้จากการดำเนินการศึกษาในบทที่ 4 จะเห็นได้ว่า ผลที่ได้จากการทำนายด้วยโปรแกรมที่พัฒนาสำหรับโครงการนี้ กับ ค่าที่จัดเก็บได้จริง จะเห็นว่ามีเดือน ก.พ. กับเดือน มี.ค. จะมีค่าใกล้เคียงกันมาก ส่วนเดือน ม.ค. และเดือน เม.ย. จะเห็นได้ว่าค่าที่ได้จากการทำนายกับค่าที่จัดเก็บได้จริงมีค่าไม่ใกล้เคียงกัน โดยสังเกตได้จากกราฟที่แสดงผล และตารางแสดงผลที่เป็นค่าตัวเลข

5.2 ข้อเสนอแนะ

การดำเนินการศึกษาในครั้งนี้เป็นการศึกษาเฉพาะปัจจัยของผลการจัดเก็บภาษีมูลค่าเพิ่มเพียงอย่างเดียว ดังนั้นการนำโปรแกรมที่พัฒนาสำหรับโครงการนี้ไปใช้งานจริง อาจมีความคลาดเคลื่อนจากผลที่ได้จากการดำเนินการในครั้งนี้บ้าง จึงควรที่จะนำปัจจัยอื่นๆ ที่มีอิทธิพลต่อการจัดเก็บภาษีมูลค่าเพิ่ม เช่น อัตราเงินเฟ้อ อัตราการขยายตัวทางเศรษฐกิจ อัตรารายได้เฉลี่ยต่อประชากรทั่วประเทศ เป็นต้น ซึ่งปัจจัยต่างๆ เหล่านี้ อาจเป็นปัจจัยที่มีอิทธิพลต่อการเปลี่ยนแปลงผลที่ได้จากการจัดเก็บภาษีมูลค่าเพิ่ม มาพิจารณาเป็นส่วนประกอบด้วย

บรรณานุกรม

- ธนพล ฉันทวีชัย. 2541. การพัฒนาโปรแกรม Windows 95 ด้วยภาษา Visual Basic 4, 5. กรุงเทพมหานคร : เอส.พี.ซี. พรินติ้ง .
- Igor Aleksander, Helen Morton. 1995. **An Introduction to Neural Computing**. 2nd Ed. London : International Thomson Computer Press.
- James A. Freeman, David M. Skapura. 1991. **Neural Networks Algorithms, Applications, and Programming Techniques**. New York : Addison-Wesley.
- Marilyn McCord Nelson, W.T. Illingworth. 1991. **A Practical Guide to Neural Nets**. New York : Addison-Wesley.
- Michael Chester. 1993. **Neural Networks A Tutorial**. New Jersey : Prentice-Hall.
- Philip D. Wasserman. 1993. **Advanced Methods in Neural Computing**. : Van Nostrand Reinhold.

ภาคผนวก ก

ตารางข้อมูลผลการจัดเก็บภาษีมูลค่าเพิ่มที่จัดเก็บได้ทั่วประเทศ



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อมูลผลการจัดเก็บภาษีมูลค่าเพิ่มที่จัดเก็บได้ทั่วประเทศ ตั้งแต่ เดือน ม.ค. 3536 ถึง เดือน

เม.ย. 2542

เดือนภาษี	ปีภาษี						
	2536	2537	2538	2539	2540	2541	2542
ม.ค.	9395990	11216780	14601690	15952490	16788725	21552524	19138487
ก.พ.	9165050	10362990	12272500	14688370	15115399	19249310	19119631
มี.ค.	9008900	11029560	14203740	15126340	15252342	20474618	19597121
เม.ย.	9140080	11823720	13545200	15918720	15979219	20027395	17277230
พ.ค.	9854500	11237550	13774490	15060300	15190683	17708481	
มิ.ย.	9662830	11853830	14385019	14968170	15193267	18759323	
ก.ค.	10242080	11878160	14477035	16061000	15964320	18202051	
ส.ค.	9582240	11793110	14608822	15738810	17549597	17933609	
ก.ย.	9465140	11880710	13918805	16726580	21799229	17834425	
ต.ค.	8971660	9977110	12332630	14700230	16807012	20873760	
พ.ย.	8856970	10531330	12496690	14414950	14887600	19389662	
ธ.ค.	9236450	11205880	12505010	14871460	15285840	20382378	

หน่วย ล้านบาท

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ข
ตัวอย่างชอร์สโค้ด



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซอร์สโค้ดส่วนของฟังก์ชันที่ใช้ในโปรแกรม

```
/* ฟังก์ชันคำนวณค่า net ของชั้น Hidden Layer
```

```
/* wh คือ ค่าน้ำหนักของชั้น Hidden Layer
```

```
/* hinput คือ ค่าอินพุตที่ใช้ในการ Training
```

```
Function nethidden(wh, hinput)
```

```
nethidden = (wh * hinput)
```

```
End Function
```

```
/* ฟังก์ชันคำนวณค่า Activation Functionของชั้น Hidden Layer
```

```
/* nethi คือ ค่า net ของชั้น hidden layer
```

```
Function acthidden(nethi)
```

```
acthidden = 1 / (1 + Exp(-nethi))
```

```
End Function
```

```
/* ฟังก์ชันคำนวณค่า net ของชั้น Output Layer
```

```
/* wo คือ ค่าน้ำหนักของชั้น output layer
```

```
/* acthi คือ ค่า output ที่ผ่านฟังก์ชัน activation จากชั้น hidden layer
```

```
Function netout(wo, acthi)
```

```
netout = (wo * acthi)
```

```
End Function
```

```
/* ฟังก์ชันคำนวณค่า Activation Functionของชั้น Output Layer
```

```
/* neto คือ ค่า net ของชั้น output layer
```

```
Function actout(neto)
```

```
actout = 1 / (1 + Exp(-neto))
```

```
End Function
```

```

/* ฟังก์ชันคำนวณหาค่า delta ของชั้น Output Layer
/* deltar คือ ค่าที่ใช้เป็นค่าเป้าหมาย (target)
/* act_out คือ ค่าผลลัพธ์ที่ได้จากฟังก์ชัน actout
Function deltaout(deltar, act_out)
deltaout = (deltar - act_out) * (act_out * (1 - act_out))
End Function

```

```

/* ฟังก์ชันคำนวณหาค่า deltaweight ของชั้น Output Layer
/* delo คือ ค่าผลลัพธ์ที่ได้จากฟังก์ชัน deltaout
/* fhidden คือ ค่าผลลัพธ์ที่ได้จากฟังก์ชัน acthidden
Function deltaweightout(delo, fhidden)
deltaweightout = (delo * fhidden)
End Function

```

```

/* ฟังก์ชันคำนวณหาค่า deltaweight ของชั้น Hidden Layer
/* delh คือ ค่าผลลัพธ์จากฟังก์ชัน deltahi
/* inhi คือ ค่า input vector
Function deltaweighthi(delh, inhi)
deltaweighthi = (delh * inhi)
End Function

```

```

/* ฟังก์ชันปรับค่าน้ำหนัก
/* delw คือ ค่า delta weight
/* wold คือ ค่าน้ำหนักเก่า
Function newweight(delw, wold)
newweight = wold + delw
End Function

```

```

/* ฟังก์ชันใช้สุ่มหาค่าน้ำหนัก ระหว่าง -1 ถึง +1
Function randweight()

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Randomize

numrnd = (2 * Rnd) - 1

randweight = numrnd

End Function

/* ฟังก์ชันหาค่าจำนวนเงินที่มากที่สุดในฐานข้อมูล

Function max_value()

Dim db As Database

SQL\$ = "select max(money_train) as[mm] from train_vat "

Set db = OpenDatabase("c:\neural\neural_vat")

Set tb = db.OpenRecordset(SQL\$)

max_value = tb!mm

db.Close

End Function

/* ฟังก์ชันใช้แปลงค่าข้อมูลจากฐานข้อมูลให้เป็นค่าตัวเลขระหว่าง 0 ถึง 1

/* h คือ ค่าของจำนวนเงินภาษีที่มากที่สุดในฐานข้อมูล

/* vm คือ เดือนภาษีที่ต้องการทำประมาณการ

/* vy คือ ปีภาษีที่ต้องการทำประมาณการ

Function height_light(h, vm, vy)

Dim dbh As Database

Dim tbh As Recordset

Dim tbh2 As Recordset

vm2 = Val(vm) - 1

vm3 = "0" + Trim(Str(vm2))

see\$ = "select money_train from train_vat where year_train <" & vy & "'order by no_idtrain
desc"

see2\$ = "select money_train from train_vat where year_train = '" & vy & "' and month_train <" &
vm3 & "'order by no_idtrain desc"

Set dbh = OpenDatabase("c:\neural\neural_vat")

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$m = h * 1.2$

Open "c:\neural\num_input.txt" For Output As #1

Set tbh2 = dbh.OpenRecordset(see2\$)

If tbh2.RecordCount > 0 Then

Do While Not tbh2.EOF

 mm2 = tbh2!money_train

 v_in2 = mm2 / m

 Print #1, v_in2

 tbh2.MoveNext

Loop

tbh2.Close

End If

Set tbh = dbh.OpenRecordset(see\$)

Do While Not tbh.EOF

 mm = tbh!money_train

 v_in = mm / m

 Print #1, v_in

 tbh.MoveNext

Loop

tbh.Close

dbh.Close

Close #1

End Function

/* ฟังก์ชันใช้กำหนดค่าที่จะใช้เป็นค่า input ในการทำประมาณการหลังจากการทำ training

/* และได้ค่า error ตามเงื่อนไขที่กำหนดแล้ว

Function result_out()

Open "c:\neural\targen.txt" For Input As #1

Open "c:\neural\num_input.txt" For Input As #2

Open "c:\neural\binary_input.txt" For Output As #3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Line Input #1, v_targeten
Print #3, v_targeten
Do While Not EOF(2)
  For n = 1 To 4
    Line Input #2, v_bi
    Print #3, v_bi
  Next n
Exit Do
Loop
Close #1
Close #2
Close #3
End Function

/* ฟังก์ชันใช้กำหนดค่าเป้าหมาย
/* h คือ ค่าจำนวนเงินที่มากที่สุดในฐานะข้อมูล
/* vm คือ เดือนปีที่ต้องการทำประมาณการ
/* vy คือ ปีปีที่ต้องการทำประมาณการ
Function hili_target(h, vm, vy)
Dim db_gt As Database
Dim tb_gt As Recordset
m = h * 1.2
vvy = Val(vy) - 1
vvm = Val(vm) - 1
vvm2 = "0" + Trim(Str(vvm))
If vm = "01" Then
  setg$ = "select money_train from train_vat where month_train = '12' and year_train = " & vvy
& ""
Else
  setg$ = "select money_train from train_vat where year_train = " & vy & ""

```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
seetg$ = seetg$ + " and month_train = " & vvm2 & ""
```

```
End If
```

```
Set db_gt = OpenDatabase("c:\neural\neural_vat")
```

```
Set tb_gt = db_gt.OpenRecordset(seetg$)
```

```
Open "c:\neural\targen.txt" For Output As #2
```

```
If tb_gt.RecordCount > 0 Then
```

```
    v_tg = tb_gt!money_train
```

```
    v_tg2 = v_tg / m
```

```
    Print #2, v_tg2
```

```
    tb_gt.Close
```

```
    db_gt.Close
```

```
End If
```

```
Close #2
```

```
End Function
```

```
/* ฟังก์ชันใช้กำหนดค่าน้ำหนักให้กับชั้น hidden layer 1
```

```
/* vin คือ จำนวนนิวรอนของชั้น input layer
```

```
/* vhi1 คือ จำนวนนิวรอนในชั้น hidden layer 1
```

```
Function writewhi1(vin, vhi1)
```

```
Dim wh()
```

```
n = (vin * vhi1) - 1
```

```
Open "c:\neural\weighthi1.txt" For Output As #3
```

```
For i = 0 To (n - 1)
```

```
    ReDim wh(i)
```

```
    wh(i) = randweight()
```

```
    Print #3, wh(i)
```

```
Next i
```

```
Close #3
```

```
End Function
```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/* ฟังก์ชันใช้สำหรับกำหนดค่าน้ำหนักของชั้น hidden layer 2
```

```
/* vhi1 คือ จำนวนนิวรอนของชั้น hidden layer 1
```

```
/* vhi2 คือ จำนวนนิวรอนของชั้น hidden layer 2
```

```
Function writewhi2(vhi1, vhi2)
```

```
Dim wh2()
```

```
n = (vhi1 * vhi2) - 1
```

```
Open "c:\neural\weighthi2.txt" For Output As #4
```

```
For i = 0 To n
```

```
    ReDim wh2(i)
```

```
    wh2(i) = randweight()
```

```
    Print #4, wh2(i)
```

```
Next i
```

```
Close #4
```

```
End Function
```

```
/* ฟังก์ชันใช้สำหรับกำหนดค่าน้ำหนักของชั้น output layer
```

```
/* vhi2 คือ จำนวนนิวรอนของชั้น hidden layer 2
```

```
Function writewout(vhi2)
```

```
Dim wh3()
```

```
n = (vhi2 - 1)
```

```
Open "c:\neural\weightout.txt" For Output As #5
```

```
For i = 0 To n
```

```
    ReDim wh3(i)
```

```
    wh3(i) = randweight()
```

```
    Print #5, wh3(i)
```

```
Next i
```

```
Close #5
```

```
End Function
```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/* ฟังก์ชัน ใช้คำนวณหาค่า net และ ค่า output ของชั้น hidden layer 1
```

```
/* vhi1 คือ จำนวนนิวรอนของชั้น hidden layer 1
```

```
Function net_acthi1(vhi1)
```

```
Dim neth2()
```

```
Dim net()
```

```
Dim acthi()
```

```
Dim neth()
```

```
hi1 = vhi1 - 1
```

```
Open "c:\neural\weighthi1.txt" For Input As #7
```

```
Open "c:\neural\acthi1.txt" For Output As #8
```

```
For j = 0 To hi1
```

```
    ReDim neth2(j)
```

```
    ReDim net(j)
```

```
    ReDim acthi(j)
```

```
    sumnet = 0
```

```
    vx = 0
```

```
    Open "c:\neural\binary_input.txt" For Input As #9
```

```
    Do While Not EOF(9)
```

```
        ReDim neth(vx)
```

```
        Line Input #9, vinput
```

```
        If EOF(7) Then
```

```
            Exit Do
```

```
        End If
```

```
        Line Input #7, vwin
```

```
        wh = vwin
```

```
        hinput = vinput
```

```
        neth(vx) = nethidden(wh, hinput)
```

```
        sumnet = sumnet + neth(vx)
```

```
        vx = vx + 1
```

```
    Loop
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Close #9
netih = sumnet
acthi(j) = acthidden(netih)
Print #8, acthi(j)
Next j
Close #8
Close #7
End Function

```

```

/* ฟังก์ชันคำนวณหาค่า net และค่า output ของชั้น hidden layer 2

```

```

/* vhi2 คือ จำนวนนิวรอนของชั้น hidden layer 2

```

```

Function net_acthi2(vhi2)

```

```

Dim v_achi2()

```

```

Dim v_neth()

```

```

1 = 0

```

```

n2 = vhi2 - 1

```

```

Open "c:\neural\weighthi2.txt" For Input As #10

```

```

Open "c:\neural\acthi2.txt" For Output As #11

```

```

For k = 0 To n2 'unit hidden layer 2

```

```

    ReDim v_achi2(k)

```

```

    sumnet2 = 0

```

```

    Open "c:\neural\acthi1.txt" For Input As #12

```

```

    Do While Not EOF(12)

```

```

        ReDim v_neth(1)

```

```

        Line Input #12, vhi1

```

```

        If EOF(10) Then

```

```

            Exit Do

```

```

        End If

```

```

        Line Input #10, vwhi2

```

```

        wh2 = vwhi2

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

hinput2 = vhi1
v_neth(l) = nethidden(wh2, hinput2)
sumnet2 = sumnet2 + v_neth(l)
l = l + 1
Loop
Close #12
netih2 = sumnet2
v_achi2(k) = acthidden(netih2)
Print #11, v_achi2(k)
Next k
Close #10
Close #11
End Function

/* ฟังก์ชันคำนวณหาค่า net และ ค่า output ของชั้น output layer
/* nvhi2 คือ จำนวนนิวรอนของชั้น hidden layer 2
Function net_actout(nvhi2)
Dim output1()
nm2 = nvhi2 - 1
Open "c:\neural\actout.txt" For Output As #13
Open "c:\neural\weightout.txt" For Input As #14
Open "c:\neural\acthi2.txt" For Input As #50
sumout = 0
For ll = 0 To nm2
    ReDim output1(ll)
    Line Input #14, vwout
    Line Input #50, vhi2
    wo = vwout
    acthiout = vhi2
    output1(ll) = netout(wo, acthiout)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
sumout = sumout + output1(ii)
```

```
Next ii
```

```
Close #14
```

```
Close #50
```

```
neto = sumout
```

```
funcout = actout(neto)
```

```
Print #13, funcout
```

```
Close #13
```

```
End Function
```

```
/* ฟังก์ชันใช้คำนวณหาค่า delta ของชั้น output layer
```

```
Function delta_out()
```

```
Open "c:\neural\targen.txt" For Input As #16
```

```
Open "c:\neural\actout.txt" For Input As #17
```

```
Open "c:\neural\delta_out.txt" For Output As #18
```

```
Line Input #16, vtg
```

```
deltar = vtg
```

```
Line Input #17, vacto
```

```
act_out = vacto
```

```
dtout = deltaout(deltar, act_out)
```

```
Print #18, dtout
```

```
Close #16
```

```
Close #17
```

```
Close #18
```

```
End Function
```

```
/* ฟังก์ชันใช้คำนวณหาค่า delta weight ของชั้น output layer
```

```
Function delta_wout()
```

```
Dim del_w()
```

```
i = 0
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Open "c:\neural\delta_out.txt" For Input As #19
Line Input #19, vd_out
delo = vd_out
Close #19

Open "c:\neural\delta_wout.txt" For Output As #20
Open "c:\neural\acthi2.txt" For Input As #21
Do While Not EOF(21)
    ReDim del_w(i)
    Line Input #21, v_achi2
    fhidden = v_achi2
    del_w(i) = deltaweightout(delo, fhidden)
    Print #20, del_w(i)
    i = i + 1
Loop
Close #21
Close #20
End Function

/* ฟังก์ชันใช้คำนวณหาค่า delta ของชั้น hidden layer 2
Function delta_hi2()
Dim deltahi()
hn = 0

Open "c:\neural\delta_hi2.txt" For Output As #22
Open "c:\neural\delta_out.txt" For Input As #24
Line Input #24, v_dout
delout = v_dout
Close #24

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Open "c:\neural\acthi2.txt" For Input As #23
```

```
Open "c:\neural\weightout.txt" For Input As #25
```

```
Do While Not EOF(23)
```

```
    ReDim deltahi(h)
```

```
    Line Input #23, v_achi2
```

```
    Line Input #25, vwout
```

```
    w = vwout
```

```
    fhi1 = v_achi2
```

```
    wdelout = w * delout
```

```
    deltahi(h) = wdelout * (fhi1 * (1 - fhi1))
```

```
    Print #22, deltahi(h)
```

```
    hn = hn + 1
```

```
Loop
```

```
Close #23
```

```
Close #22
```

```
Close #25
```

```
End Function
```

```
/* ฟังก์ชันใช้คำนวณหาค่า delta weight ของชั้น hidden layer 2
```

```
Function del_whi2()
```

```
Dim delwhi2()
```

```
i = 0
```

```
Open "c:\neural\delta_whi2.txt" For Output As #27
```

```
Open "c:\neural\acthi1.txt" For Input As #26
```

```
Do While Not EOF(26)
```

```
    Line Input #26, vachi1
```

```
    inhi = vachi1
```

```
    Open "c:\neural\delta_hi2.txt" For Input As #25
```

```
    Do While Not EOF(25)
```

```
        ReDim delwhi2(i)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Line Input #25, vdthi2
delh = vdthi2
delwhi2(i) = deltaweighthi(delh, inhi)
Print #27, delwhi2(i)
i = i + 1
Loop
Close #25
Loop
Close #26
Close #27
End Function

/* ฟังก์ชันใช้คำนวณหาค่า delta ของชั้น hidden layer 1
/* vhi1 คือ จำนวนนิวรอนของชั้น hidden layer 1
/* vhi2 คือ จำนวนนิวรอนของชั้น hidden layer 2
Function delta_hi1(vhi2, vhi1)
Dim deltahi1()
m1 = (vhi1 - 1)
m2 = (vhi2 - 1)
t = (vhi1 * vhi2) - 1
Open "c:\neural\delta_hi1.txt" For Output As #28
Open "c:\neural\acthi1.txt" For Input As #29
For h = 0 To m1
    ReDim deltahi1(h)
    Line Input #29, v_achi1
    Open "c:\neural\delta_hi2.txt" For Input As #30
    For i = 0 To m2 'mhi2 'ตามจำนวน deltahi2
        Line Input #30, v_dout
        delout = v_dout
        Open "c:\neural\weighthi2.txt" For Input As #31

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

k = vhi1 * i + h 'maxhi2=2
For j = 0 To t' t = max unit hi2 * max unit out
    Line Input #31, vwhi2
    If j = k Then
        w = vwhi2
    End If
Next j
Close #31
wdelout = wdelout + (w * delout)
Next i
Close #30
fhi = v_achi1
deltah1(h) = wdelout * (fhi * (1 - fhi))
Print #28, deltah1(h)
wdelout = 0
Next h
Close #29
Close #28
End Function

/* ฟังก์ชันใช้คำนวณหาค่า delta weight ของชั้น hidden layer 1
Function delta_who1()
Dim inpuh1()
Dim deltawhi1()
j = 0
Open "c:\neural\binary_input.txt" For Input As #32
Open "c:\neural\delta_who1.txt" For Output As #33
Do While Not EOF(32)
    Line Input #32, vnh1
    inhi = vnh1

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Open "c:\neural\delta_hi1.txt" For Input As #34
Do While Not EOF(34)
    ReDim deltawhi1(j)
    Line Input #34, vdelh1
    delh = vdelh1
    deltawhi1(j) = deltaweighthi(delh, inhi)
    Print #33, deltawhi1(j)
    j = j + 1
Loop
Close #34
Loop
Close #32
Close #33
End Function

/* ฟังก์ชันใช้ในการปรับค่าน้ำหนักของชั้น output layer ใหม่
Function upwout()
Dim rw()
i = 0
Open "c:\neural\weightout.txt" For Input As #4
Open "c:\neural\upweightout.txt" For Output As #5
Open "c:\neural\delta_wout.txt" For Input As #6
Do While Not EOF(4)
    ReDim rw(i)
    Line Input #4, w_out
    Line Input #6, dw_out
    delwout = dw_out
    vwout = w_out
    rw(i) = Val(vwout) + Val(delwout)
    Print #5, rw(i)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    i = i + 1
Loop
Close #4
Close #6
Close #5
End Function

/* ฟังก์ชันใช้ในการปรับค่าน้ำหนักของชั้น hidden layer 2 ใหม่
Function upwhi2()
Dim rw()
i = 0
Open "c:\neural\weighthi2.txt" For Input As #40
Open "c:\neural\upweighthi2.txt" For Output As #41
Open "c:\neural\delta_whi2.txt" For Input As #42
Do While Not EOF(40)
    ReDim rw(i)
    Line Input #40, wold
    Line Input #42, del_weight
    volw = wold
    vdw = del_weight
    rw(i) = Val(voldw) + Val(vdw)
    Print #41, rw(i)
    i = i + 1
Loop
Close #40
Close #41
Close #42
End Function

```

/* ฟังก์ชันใช้ในการปรับค่าน้ำหนักของชั้น hidden layer 1 ใหม่

Function upwhi()

Dim upwh()

i = 0

Open "c:\neural\weighthi1.txt" For Input As #35

Open "c:\neural\upweighthi1.txt" For Output As #36

Open "c:\neural\delta_whi1.txt" For Input As #37

Do While Not EOF(35)

ReDim upwh(i)

If EOF(37) Then

Exit Do

GoTo app_whi1

End If

Line Input #37, del_whi

Line Input #35, whi_old

vdhi = del_whi

voldw = whi_old

upwh(i) = Val(voldw) + Val(vdhi)

Print #36, upwh(i)

i = i + 1

Loop

Close #37

Close #35

Close #36

app_whi1:

Close #37

Close #35

Close #36

Open "c:\neural\weighthi1.txt" For Input As #50

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Open "c:\neural\upweighthi1.txt" For Append As #51
cc = 0
Do While Not EOF(50)
    Line Input #50, v_whi1
    If cc = i Then
        Print #51, v_whi1
        i = i + 1
    End If
    cc = cc + 1
Loop
Close #50
Close #51
End Function

```

/* ฟังก์ชันใช้สำหรับกำหนดค่าน้ำหนักให้ชั้น output layer ใหม่

```

Function re_wout()
Open "c:\neural\upweightout.txt" For Input As #38
Open "c:\neural\weightout.txt" For Output As #39
Do While Not EOF(38)
    Line Input #38, vup
    Print #39, vup
Loop
Close #38
Close #39
End Function

```

/* ฟังก์ชันใช้สำหรับกำหนดค่าน้ำหนักให้ชั้น hidden layer 2 ใหม่

```

Function re_whi2()
Open "c:\neural\upweighthi2.txt" For Input As #40
Open "c:\neural\weighthi2.txt" For Output As #41

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Do While Not EOF(40)
```

```
    Line Input #40, vup
```

```
    Print #41, vup
```

```
Loop
```

```
Close #40
```

```
Close #41
```

```
End Function
```

```
/* ฟังก์ชันใช้สำหรับกำหนดค่าน้ำหนักให้กับชั้น hidden layer 1 ใหม่
```

```
Function re_whi1()
```

```
Open "c:\neural\upweighthi1.txt" For Input As #42
```

```
Open "c:\neural\weighthi1.txt" For Output As #43
```

```
Do While Not EOF(42)
```

```
    Line Input #42, vup
```

```
    Print #43, vup
```

```
Loop
```

```
Close #42
```

```
Close #43
```

```
End Function
```

```
/* ฟังก์ชันใช้คำนวณหาค่าผิดพลาดที่เกิดขึ้น
```

```
Function errorp()
```

```
    terror = 0
```

```
    Open "c:\neural\actout.txt" For Input As #28
```

```
    Open "c:\neural\targen.txt" For Input As #29
```

```
    Open "c:\neural\errorp.txt" For Append As #30
```

```
    Line Input #29, vtargen
```

```
    Line Input #28, v_actout
```

```
    fout = v_actout
```

```
    tar = vtargen
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
p_error = (tar - fout) ^ 2
```

```
finerror = p_error / 2
```

```
Print #30, finerror
```

```
Close #28
```

```
Close #29
```

```
Close #30
```

```
End Function
```

```
/* ฟังก์ชันใช้คำนวณหาค่าผลรวมของค่าผิดพลาดที่เกิดขึ้น
```

```
Function sumerror()
```

```
s_errp = 0
```

```
Open "c:\neural\errorp.txt" For Input As #31
```

```
Do While Not EOF(31)
```

```
Line Input #31, errp
```

```
s_errp = s_errp + errp
```

```
Loop
```

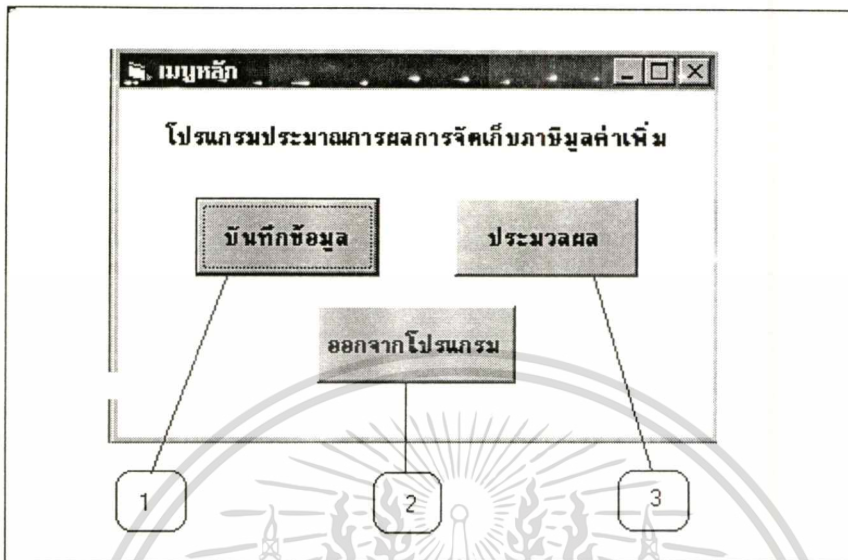
```
Close #31
```

```
sumerror = s_errp
```

```
End Function
```



ซอร์สโค้ดโปรแกรมในส่วนของหน้าจอเมนูหลัก



/* โค้ดของปุ่มหมายเลข 1 เมื่อคลิกที่ปุ่มนี้จะแสดงหน้าจอสำหรับบันทึกข้อมูลผลการจัดเก็บภาษี

```
Private Sub Command1_menu_Click()
Me.Hide
form_inputtrain.Show
End Sub
```

/* โค้ดของปุ่มหมายเลข 2 เมื่อคลิกปุ่มนี้จะแสดงหน้าจอสำหรับการทำประมวลผลภาษี

```
Private Sub Command2_menu_Click()
Me.Hide
Form_process.Show
End Sub
```

/* โค้ดของปุ่มหมายเลข 3 เมื่อคลิกปุ่มนี้จะออกจากโปรแกรม

```
Private Sub Command3_menu_Click()
End
End Sub
```

ซอร์สโค้ดโปรแกรมสำหรับหน้าจอบันทึกข้อมูลผลการจัดเก็บภาษีมูลค่าเพิ่ม

/* โค้ดส่วนนี้เมื่อเรียกหน้าจอบันทึกข้อมูลขึ้นมาแสดง

```
Private Sub Form_Load()
```

```
Dim db_neural As Database
```

```
Set db_neural = OpenDatabase("c:\neural\neural_vat")
```

```
End Sub
```

/* โค้ดของปุ่มหมายเลข 3 คลิกที่ปุ่มนี้จะบันทึกข้อมูลลง table ชื่อ train_vat

```
Private Sub Command1_vat_Click()
```

```
sqladd$ = "insert into train_vat(month_train,year_train,money_train) "
```

```
sqladd$ = sqladd$ + "Values( " & text1_monthvat.Text & ", " & text2_yvat.Text & ", " & Val  
(text3_mvvat.Text) & ")"
```

```
db_neural.Execute sqladd$
```

```
text1_monthvat.Text = ""
```

```
text2_yvat.Text = ""
```

```
text3_mvvat.Text = ""
```

```
End Sub
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/* ใ้ค้ของปุ่มหมายเลข 4 เมื่อคลิกปุ่มนี้จะออกจากหน้าจอบันทึกข้อมูล
```

```
/* และจะแสดงหน้าจอเมนูหลัก
```

```
Private Sub Command2_vat_Click()
```

```
Me.Hide
```

```
Form_menubpn.Show
```

```
End Sub
```

```
/* ใ้ค้ของปุ่มหมายเลข 1 เมื่อคีย์ข้อมูลเดือนภายในที่เป็นชื่อเดือน เช่น ม.ค.
```

```
/* ก็จะถูกแปลงให้เป็นตัวเลขแทน เช่น ม.ค. จะเป็น "01"
```

```
Private Sub text1_monthvat_LostFocus()
```

```
chk_month = text1_monthvat.Text
```

```
Select Case chk_month
```

```
Case "ม.ค."
```

```
text1_monthvat.Text = "01"
```

```
Case "ก.พ."
```

```
text1_monthvat.Text = "02"
```

```
Case "มี.ค."
```

```
text1_monthvat.Text = "03"
```

```
Case "เม.ย."
```

```
text1_monthvat.Text = "04"
```

```
Case "พ.ค."
```

```
text1_monthvat.Text = "05"
```

```
Case "มิ.ย."
```

```
text1_monthvat.Text = "06"
```

```
Case "ก.ค."
```

```
text1_monthvat.Text = "07"
```

```
Case "ส.ค."
```

```
text1_monthvat.Text = "08"
```

```
Case "ก.ย."
```

```
text1_monthvat.Text = "09"
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Case "ต.ค."

text1_monthvat.Text = "10"

Case "พ.ย."

text1_monthvat.Text = "11"

Case "ธ.ค."

text1_monthvat.Text = "12"

End Select

End Sub

/* โค้ดของปุ่มหมายเลข 2 เมื่อกดข้อมูลปีภาษีเสร็จแล้วจะตรวจสอบว่ามีข้อมูลนี้ในฐานข้อมูล
 /* หรือไม่ถ้ามีก็จะไม่บันทึกข้อมูลซ้ำลงในฐานข้อมูล และจะแสดงข้อความเตือน

Private Sub text2_yvat_LostFocus()

Dim tbch As Recordset

sqlch\$ = "select * from train_vat where month_train = " & text1_monthvat & " and year_train =
 " & text2_yvat & ""

Set tbch = db_neural.OpenRecordset(sqlch\$)

If tbch.RecordCount > 0 Then

MsgBox " ข้อมูลซ้ำ! "

text1_monthvat.SetFocus

text1_monthvat.Text = ""

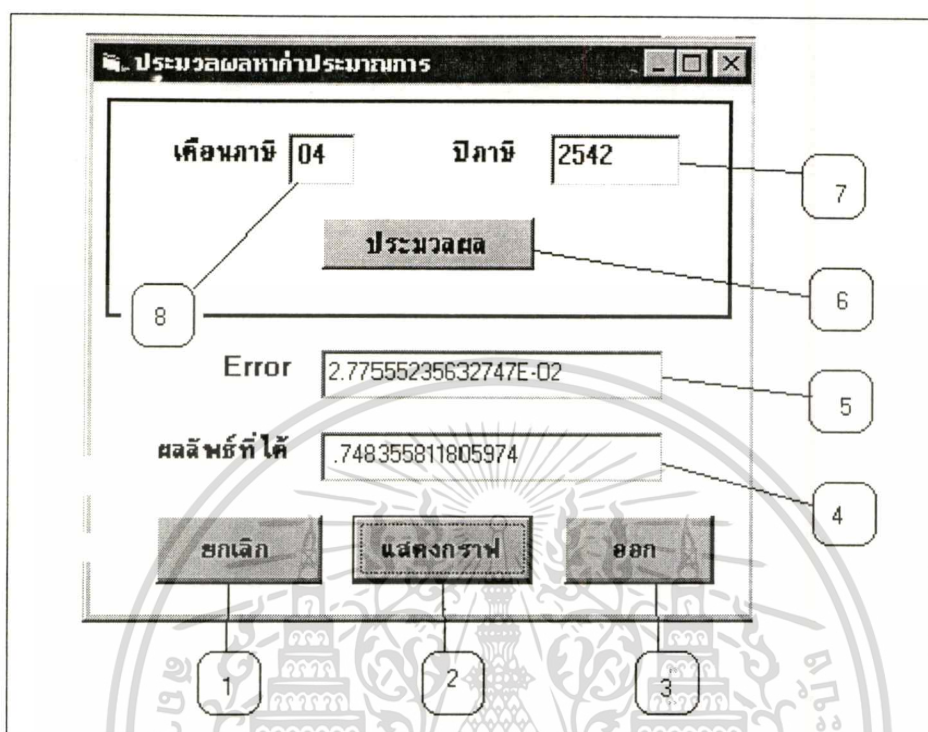
text2_yvat.Text = ""

End If

tbch.Close

End Sub

ซอร์สโค้ดโปรแกรมสำหรับหน้าจอประมวลผลหาค่าประมาณการภาษี



/* ตัวแปรนี้จะถูกประกาศค่าเป็นตัวแปรแบบ โกลบอลสำหรับให้ทุก Sub เรียกใช้งานได้

Dim vin

Dim vhi1

Dim vhi2

Dim vy

Dim h

Dim v_hin

Dim v_hili

Dim vm

/* โค้ดนี้จะใช้สำหรับตอนที่หน้าจอจะถูกเรียก

Private Sub Form_Load()

h = max_value()

End Sub

/* โค้ดของปุ่มหมายเลข 6 เมื่อคลิกปุ่มนี้จะประมวลผลหาค่าประมาณการผลการจัดเก็บภาษี

Private Sub Command_addvarpro_Click()

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับบริการเชิงงานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

vm = Text_monthpro.Text /* คือ text box หมายเลข 8
vy = Text_ypro.Text /* คือ text box หมายเลข 7
Call height_light(h, vm, vy)
Call hili_target(h, vm, vy)
Call data_graph(h, vm, vy)
vin = 5 /* จำนวนนิวรอนในชั้น Input
vhi1 = 20 /* จำนวนนิวรอนในชั้น Hidden layer 1
vhi2 = 20 /* จำนวนนิวรอนในชั้น Hidden layer 2
Call writewhi1(vin, vhi1)
Call writewhi2(vhi1, vhi2)
Call writewout(vhi2)
max_error = 0
Do
  Open "c:\neural\num_input.txt" For Input As #1
  Do While Not EOF(1)
  Open "c:\neural\binary_input.txt" For Output As #2
  For i = 0 To (vin - 1)
    If EOF(1) Then
      Exit For
    End If
    Line Input #1, st_input
    Print #2, st_input
  Next i
  Close #2
  Call net_acthi1(vhi1)
  Call net_acthi2(vhi2)
  nvhi2 = vhi2
  Call net_actout(nvhi2)
  Call errorp
  Call delta_out

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Call delta_wout
Call delta_hi2
Call del_who2
Call delta_hi1(vhi2, vhi1)
Call delta_who1
Call upwout
Call upwho2
Call upwho1
Call re_wout
Call re_who2
Call re_who1
Loop
Close #1
max_error = sumerror()
Kill "c:\neural\errorp.txt"
Loop Until max_error <= 0.05
Call result_out
Call net_acthi1(vhi1)
Call net_acthi2(vhi2)
nvhi2 = vhi2
Call net_actout(nvhi2)
Text_errpro.Text = max_error /* คือ text box หมายเลข 5
Open "c:\neural\actout.txt" For Input As #3
Line Input #3, v_output
Text_resultpro.Text = v_output /* คือ text box หมายเลข 4
Close #3
Command_graphpro.Enabled = True
End Sub
/* ใ้ค้คของปุ่มหมายเลข 3 เมื่อคลิกปุ่มนี้จะออกจากหน้าจอประมวลผลหาค่าประมาณการ
/* และจะแสดงหน้าจอเมนูหลัก

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Private Sub Command_exitpro_Click()
see_dir = Dir("c:\neural\*.txt")
If see_dir <> "" Then
    Kill "c:\neural\*.txt"
End If
Me.Refresh
Text_monthpro.Text = ""
Text_ypro.Text = ""
Text_errpro.Text = ""
Text_resultpro.Text = ""
Me.Hide
Form_menubpn.Show
End Sub

```

/* ใ้ค้ดของปุ่มหมายเลข 2 เมื่อกลิกปุ่มนี้จะแสดงหน้าจอแสดงกราฟ

```

Private Sub Command_graphpro_Click()
Me.Hide
Form_graph.Show
End Sub

```

/* ใ้ค้ดของปุ่มหมายเลข 1 เมื่อกลิกปุ่มนี้เมื่อต้องการลบข้อมูลใน text box

```

Private Sub Command_undopro_Click()
un_dir = Dir("c:\neural\*.txt")
If un_dir <> "" Then
    Kill "c:\neural\*.txt"
End If
Me.Refresh
Text_monthpro.Text = ""
Text_ypro.Text = ""
Text_errpro.Text = ""

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Text_resultpro.Text = ""
```

```
End Sub
```

```
/* ใ้ค้ของ text box หมายเลข 8 สำหรับแปลงค่าเดือนภาษาไทย ให้เป็นตัวเลข
```

```
/* เช่น "ม.ค." เป็น "01"
```

```
Private Sub Text_monthpro_LostFocus()
```

```
chk_mon = Text_monthpro.Text
```

```
Select Case chk_month
```

```
Case "ม.ค."
```

```
Text_monthpro.Text = "01"
```

```
Case "ก.พ."
```

```
Text_monthpro.Text = "02"
```

```
Case "มี.ค."
```

```
Text_monthpro.Text = "03"
```

```
Case "เม.ย."
```

```
Text_monthpro.Text = "04"
```

```
Case "พ.ค."
```

```
Text_monthpro.Text = "05"
```

```
Case "มิ.ย."
```

```
Text_monthpro.Text = "06"
```

```
Case "ก.ค."
```

```
Text_monthpro.Text = "07"
```

```
Case "ส.ค."
```

```
Text_monthpro.Text = "08"
```

```
Case "ก.ย."
```

```
Text_monthpro.Text = "09"
```

```
Case "ต.ค."
```

```
Text_monthpro.Text = "10"
```

```
Case "พ.ย."
```

```
Text_monthpro.Text = "11"
```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Case "ธ.ค."

Text_monthpro.Text = "12"

End Select

End Sub

/* โค้ดของ text box หมายเลข 7 เมื่อคีย์ข้อมูลเดือนภายในและปีภายในเสร็จ โค้ดส่วนนี้จะทำการตรวจ
/* สอบว่ามีข้อมูลที่จะใช้ในการฝึกสอนหรือไม่ ถ้าไม่มีก็จะแสดงข้อความเตือน

Private Sub Text_ypro_LostFocus()

Dim sqldb As Database

Dim sqltb As Recordset

vvy = Val(Text_ypro.Text) - 1

vch = Val(Text_monthpro.Text) - 1

vvch = "0" + Trim(Str(vch))

If Text_monthpro.Text = "01" Then

sqlpro\$ = "select money_train from train_vat where month_train = '12' and year_train = '" &
vvy & """

Else

sqlpro\$ = "select * from train_vat where month_train = '" & vvch & "' and year_train = '" &
Text_ypro.Text & """

End If

Text_ypro.Text & """

Set sqldb = OpenDatabase("c:\neural\neural_vat")

Set sqltb = sqldb.OpenRecordset(sqlpro\$)

If sqltb.RecordCount = 0 Then

MsgBox "ไม่มีข้อมูลผลการจัดเก็บของเดือน/ปีภายในอยู่ในฐานข้อมูล"

sqltb.Close

Text_monthpro.SetFocus

Text_monthpro.Text = ""

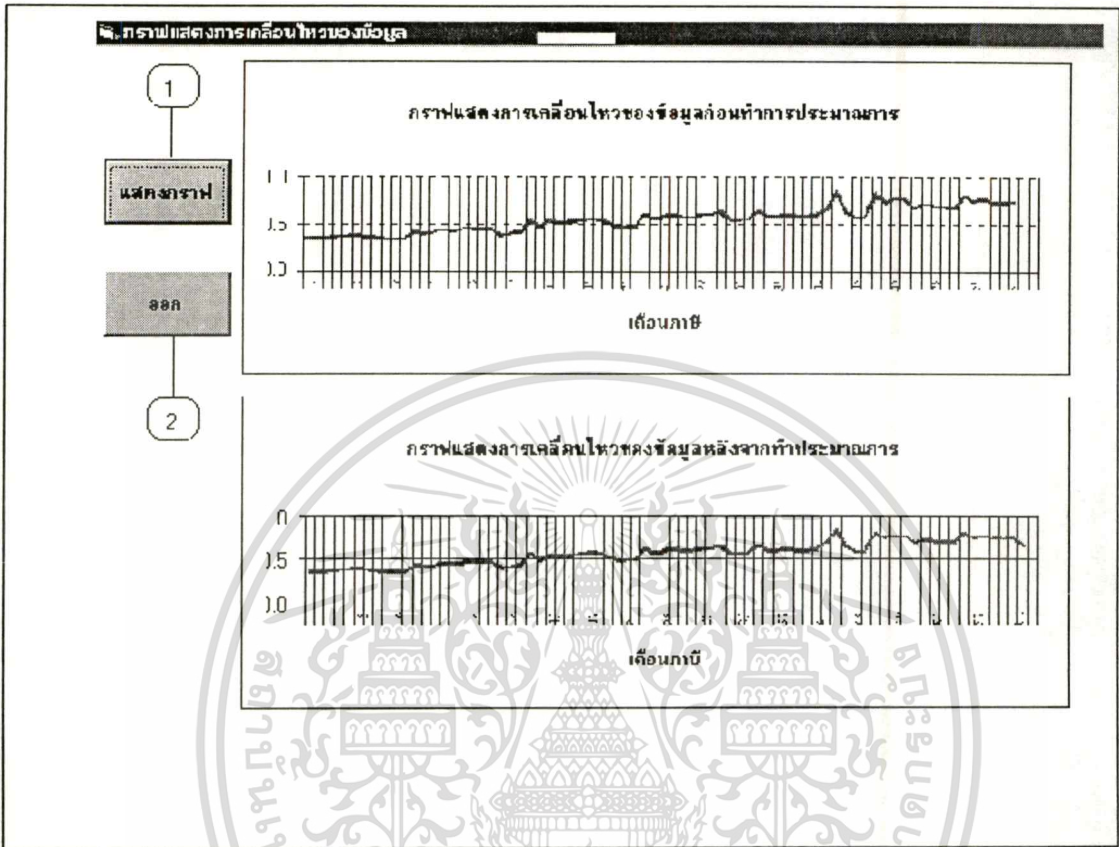
Text_ypro.Text = ""

End If

End Sub

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซอร์สโค้ดโปรแกรมสำหรับหน้าจอแสดงกราฟ



/* โค้ดของปุ่มหมายเลข 1 เมื่อคลิกปุ่มนี้จะแสดงกราฟ

```
Private Sub Command1_graph_Click()
```

```
Dim gdb As Database
```

```
Dim gtb As Recordset
```

```
gsq1$ = "select count(money_train) as[gc] from train_vat"
```

```
Set gdb = OpenDatabase("c:\neural\neural_vat")
```

```
Set gtb = gdb.OpenRecordset(gsq1$)
```

```
v_gm = gtb!gc
```

```
gdb.Close
```

```
i = 0
```

```
j = 0
```

```
ii = 0
```

```
jj = 0
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

MSChart1.ColumnCount = 1
MSChart1.RowCount = v_gm
MSChart1.AutoIncrement = False
Open "c:\neural\text_chart.txt" For Input As #1
Do While Not EOF(1)
Line Input #1, v1
i = i + 1
If i = 13 Then
    j = 0
End If
j = j + 1
MSChart1.Data = v1
MSChart1.Row = i
MSChart1.RowLabel = j
Loop
Close #1
MSChart2.ColumnCount = 1
MSChart2.RowCount = v_gm
MSChart2.AutoIncrement = False
Open "c:\neural\text_chart.txt" For Input As #2
Do While Not EOF(2)
Line Input #2, v2
ii = ii + 1
If ii = 13 Then
    jj = 0
End If
jj = jj + 1
MSChart2.Data = v2
MSChart2.Row = ii
MSChart2.RowLabel = jj

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

s2 = ii
Loop
Close #2
Open "c:\neural\actout.txt" For Input As #3
Line Input #3, gra_output
gr_reout = gra_output
MSChart2.Data = gr_reout
MSChart2.Row = s2
Close #3
End Sub

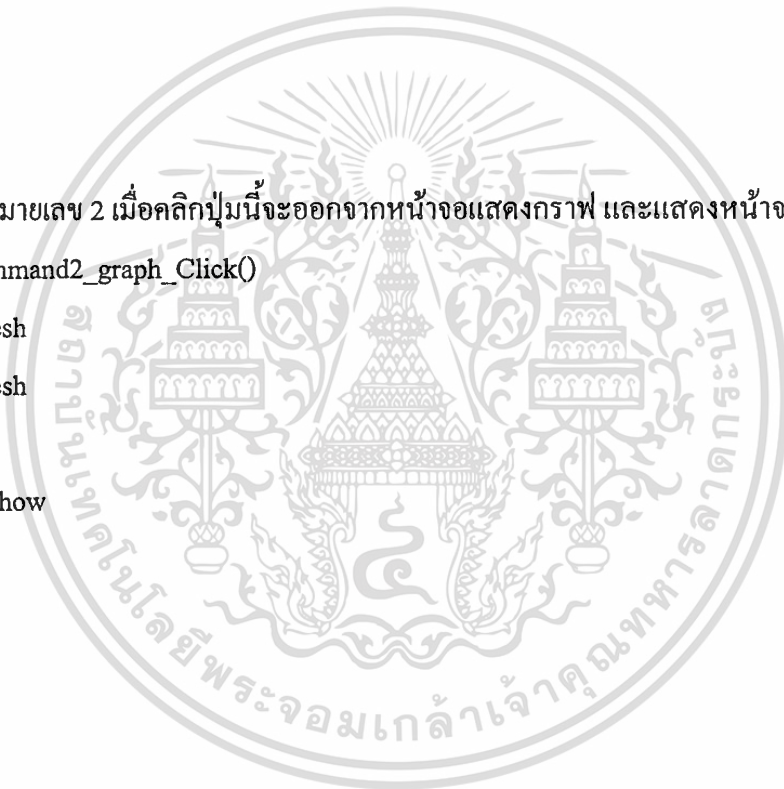
```

/* โค้ดของปุ่มหมายเลข 2 เมื่อคลิกปุ่มนี้จะออกจากหน้าจอแสดงกราฟ และแสดงหน้าจอเมนูหลัก

```

Private Sub Command2_graph_Click()
MSChart1.Refresh
MSChart2.Refresh
Me.Hide
Form_process.Show
End Sub

```



ประวัติผู้เขียน

ชื่อผู้เขียน	นายสนิทวงศ์ กมลภากรณ์
วันเดือนปีเกิด	4 พฤษภาคม 2513
สถานที่เกิด	จังหวัดอุบลราชธานี
วุฒิการศึกษาระดับปริญญาตรี	วท.บ. สถิติประยุกต์
สถานที่สำเร็จการศึกษา	วิทยาลัยครูอุบลราชธานี
ปีที่สำเร็จการศึกษา	2535
ประวัติการทำงาน	2536 รับราชการในตำแหน่งเจ้าหน้าที่ระบบงานคอมพิวเตอร์ ระดับ 3 สังกัดกรมสรรพากร
	2539 – 2540 รับราชการในตำแหน่งนักวิชาการสรรพากร ระดับ 4 สังกัดกรมสรรพากร
	2541 ถึง ปัจจุบัน รับราชการในตำแหน่งนักวิชาการสรรพากร ระดับ 5 สังกัดกรมสรรพากร



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้