โปรแกรมทำนายระดับน้ำโดยใช้โครงข่ายประสาทเทียม
Water Level Prediction by Backpropagation Neural Network

โดย
นายสุพานิช อังศิริกุล

วิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของ
รายวิชา CSC690

ประจำภาคเรียนที่ 1 ปีการศึกษา 2546

กิตติกรรมประกาศ

การพัฒนาโปรแกรมทำนายระดับน้ำโดยใช้โครงข่ายประสาทเทียมจนสำเร็จลุล่วงไปได้ด้วยดีนั้น ผู้พัฒนาต้องขอกราบขอบพระคุณ ดร.วรวุฒิ ไพรวัน ที่ปรึกษาปริญญานิพนธ์ที่ได้ให้คำแนะนำและชี้แนวทางในการศึกษาต่างๆ รวมถึง อาจารย์สิริพร ศุภราทิตย์ ที่ได้ให้คำแนะนำในการพัฒนาโปรแกรมรวมไปถึงให้การสนับสนุนข้อมูลปริมาณน้ำฝนรายวันและระดับน้ำรายวันของลุ่มแม่น้ำยม บริเวณอำเภอเมือง จ. สุโขทัย

ประโยชน์และความดีของโครงงานนี้ ขอมอบแด่ผู้มีพระคุณทุกๆ ท่านที่ได้ให้การช่วยเหลือและสนับสนุนจนโครงงานนี้สำเร็จลุล่วงไปได้ด้วยดี

สุพานิช อังศิริกุล

29 กันยายน 2546

สารบัญ

หน้า

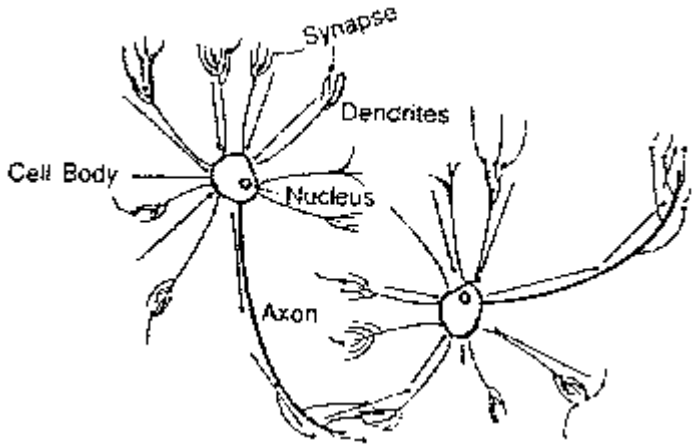
บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

การทำนายหรือการคำนวณระดับน้ำรายวัน เป็นงานที่สำคัญสำหรับวิศวกรแม่น้ำ ในการที่จะวางแผนและจัดการเกี่ยวกับปัญหาน้ำท่วมและการบรรเทาภัยพิบัติจากน้ำท่วม การทำนายดังกล่าวสามารถทำได้หลายวิธี เช่น การวิเคราะห์แบบ Multiple regression หรือการใช้ Conceptual model เช่น Tank model วิธีหนึ่งที่นิยมทำกันคือ การใช้แบบจำลองทางอุทกวิทยาชนิด Physically based distributed model ที่จำลองขั้นตอนต่างๆของขบวนการเกิดการไหลของน้ำในแม่น้ำด้วยความสัมพันธ์ทางฟิสิกส์ที่แท้จริงหรือการใช้แบบจำลองเชิงเส้น เนื่องจากกระทำได้ง่ายและรวดเร็ว อย่างไรก็ตามผลการคำนวณยังมีความคลาดเคลื่อนอยู่พอสมควร เนื่องจากปรากฏการณ์ ที่เกิดขึ้นไม่สามารถจำลองด้วยวิธีการประมาณที่เป็นแบบเชิงเส้นได้ เพราะหากสังเกตค่าข้อมูลระดับน้ำจะเห็นได้ว่ารูปแบบของข้อมูลไม่ได้อยู่ในรูปแบบที่เป็นเชิงเส้น ประกอบกับแบบจำลองแต่ละตัวก็ให้ผลลัพธ์จากการคำนวณที่แตกต่างกันมาก นอกจากนี้ขั้นตอนดังกล่าวก็มีความซับซ้อนและยังประกอบด้วยตัวพารามิเตอร์ที่มีความเกี่ยวข้องซึ่งต้องใช้ในการคำนวณจำนวนมาก จึงเป็นการยากในการประมาณค่า ซึ่งเป็นสาเหตุหนึ่งที่ทำให้ผลลัพธ์จากการทำนายมีความถูกต้องน้อยโดยเฉพาะอย่างยิ่งเมื่อไปประยุกต์ใช้กับ แม่น้ำที่มีขนาดใหญ่

Artificial Neural Network หรือโครงข่ายประสาทเทียม เป็นแนวความคิดที่ถูกออกแบบให้ทำงานในลักษณะที่คล้ายกับสมองของมนุษย์ ซึ่งประกอบด้วย processing elements ที่เป็นเซลล์หลายตัว โดยที่เซลล์แต่ละตัวจะติดต่อกันโดยส่งสัญญาณเป็นเอาท์พุทผ่านส่วนที่เรียกว่า Synapses กลายมาเป็นอินพุทในส่วนที่เรียกว่า Dendrites และเมื่อ

ผ่านการประมวลผลจะได้เอาท์พุทเพียง 1 เอาท์พุทและจะส่งออกมาจากส่วนที่เรียกว่า Axon ซึ่งในแต่ละเซลล์ประสาทจะรับข้อมูลจากหลายๆทางแล้วส่งต่อไปยังเซลล์ประสาทตัวอื่นๆ โดยใช้หลักการ “Synaptic Strength” ของการเชื่อมโยงเซลล์ประสาทดังรูปที่ 1.1



รูปที่ 1.1การส่งข้อมูลระหว่างเซลล์ประสาท

จากการศึกษาเรื่องโครงข่ายประสาทเทียมทำให้เกิดรูปแบบโครงข่ายประสาทเทียมในหลายๆรูปแบบซึ่งสามารถแบ่งออกเป็น 2 ประเภทใหญ่ๆ คือแบบที่ให้โครงข่ายประสาทเทียมเรียนรู้โดยต้องมีการแนะว่าผลลัพธ์ที่ต้องการมีหน้าตาเป็นอย่างไรหรือเรียกว่า Supervise Learning และแบบที่ให้โครงข่ายประสาทเทียมเรียนรู้โดยไม่ต้องมีการแนะว่าผลลัพธ์ที่ได้จะมีลักษณะเป็นอย่างไรให้โครงข่ายประสาทเทียมทำการหาผลลัพธ์เอง ซึ่งเรียกว่า Unsupervised Learning ได้มีการนำโครงข่ายประสาทเทียมไปใช้งานที่หลากหลายไม่ว่าจะเป็นการใช้โครงข่ายประสาทเทียมเพื่อการระบุหรือจัดกลุ่ม การใช้โครงข่ายประสาท

เทียมสำหรับหาความเกี่ยวข้องกันของรูปแบบต่างๆ การใช้โครงข่ายประสาทเทียมสำหรับการหาค่าที่เหมาะสม ซึ่งข้อดีของระบบโครงข่ายประสาทเทียมคือจะมีความยืดหยุ่นในการประยุกต์ใช้งาน หากนำไปใช้ในการคำนวณจะให้ผลลัพธ์ได้ในแบบที่เป็นเชิงเส้นและแบบที่ไม่เป็นเชิงเส้น และสามารถทำการคำนวณข้อมูลที่อยู่ในรูปแบบที่มีความซับซ้อนได้ดีโดยที่ผู้ใช้ไม่จำเป็นต้องทำความเข้าใจถึงรูปแบบของข้อมูล

ถ้าหากเรานำระบบโครงข่ายประสาทเทียมที่สามารถคำนวณข้อมูลและให้ผลลัพธ์ในรูปแบบที่ไม่เป็นเชิงเส้นมาใช้ในการทำนายระดับน้ำก็จะทำให้เราสามารถทำนายระดับน้ำที่จะเกิดขึ้นในอนาคตและสามารถประมาณการณ์ได้ว่าจะเกิดน้ำท่วมหรือไม่

1.2 วัตถุประสงค์ของการศึกษา

การวิจัยในครั้งนี้ ผู้วิจัยได้นำเอาเทคโนโลยีระบบโครงข่ายประสาทเทียมมาประยุกต์ใช้ในการสร้างแบบจำลองที่จะใช้ในการทำนายหาระดับน้ำที่จะเกิดขึ้นในอนาคต โดยมีวัตถุประสงค์ดังต่อไปนี้

1.2.1 ศึกษารูปแบบของโครงข่ายประสาทเทียมเพื่อนำมาใช้ในการสร้างแบบจำลองที่จะใช้ในการทำนายหาระดับน้ำที่จะเกิดขึ้นในอนาคต

1.2.2 เพื่อพัฒนาโปรแกรมโดยอาศัยรูปแบบโครงข่ายประสาทเทียมให้สามารถทำนายระดับน้ำที่จะเกิดขึ้นในอนาคตได้

1.2.3 เพื่อศึกษาถึงตัวแปรต่างๆที่มีผลต่อการเรียนรู้ของโครงข่ายประสาทเทียมที่เหมาะสมกับการทำนายระดับน้ำ

1.3 ขอบเขตของการศึกษา

การวิจัยครั้งนี้จะเน้นที่การพัฒนาโปรแกรมที่อาศัยระบบโครงข่ายประสาทเทียมให้สามารถทำนายหาระดับน้ำที่จะเกิดขึ้นในอนาคตได้จริง โดยมีขอบเขตต่อไปนี้

1.3.1 หารูปแบบของโครงข่ายที่ให้ผลลัพธ์ในการทำนายได้ดีที่สุด
1.3.2 ศึกษาถึงค่าอัตราการเรียนรู้ที่เหมาะสมที่ทำให้โครงข่ายสามารถเรียนรู้ได้เร็ว
1.3.3 สามารถวิเคราะห์ได้ว่าโครงข่ายแบบใดให้ผลลัพธ์ดีที่สุด โดยใช้เครื่องมือวัดประสิทธิภาพคือ Efficiency Index (EI), Root Mean Square Error, Mean Absolute Error (MAD)
1.3.4 พัฒนาโปรแกรมให้สามารถแสดงกราฟเปรียบเทียบผลลัพธ์ที่ได้จากทำนายกับค่าที่ถูกต้องเพื่อให้สามารถมองเห็นประสิทธิภาพของการทำนายได้
1.3.5 พัฒนาโปรแกรมให้ผู้ใช้สามารถปรับเปลี่ยนรูปแบบของโครงข่ายที่จะใช้ในการเรียนรู้ได้

1.4 ประโยชน์ที่คาดว่าจะได้รับ

การวิจัยครั้งนี้ จะสามารถเป็นต้นแบบในการพัฒนาครั้งต่อๆไป สำหรับผู้สนใจ และประโยชน์ที่คาดว่าจะรับจากงานวิจัยในครั้งนี้คือ

1.4.1 โปรแกรมที่สามารถใช้ในการทำนายระดับน้ำรายวัน
1.4.2 สามารถใช้โปรแกรมในการทำนายค่าระดับน้ำรายวันเพื่อพยากรณ์ว่าจะเกิดเหตุการณ์น้ำท่วมขึ้นหรือไม่

1.4.3สามารถใช้โปรแกรมในการทำนายข้อมูลอื่นๆที่มีรูปแบบใกล้เคียงกันได้

1.4.4เป็นแนวทางในการศึกษาและพัฒนาระบบการทำนายในรูปแบบอื่นๆ

บทที่ 2
ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

จากการพัฒนาคอมพิวเตอร์ให้มีความสามารถในการประมวลผลที่สูงขึ้น นักวิทยาศาสตร์ได้พยายามใช้ความสามารถของคอมพิวเตอร์ในการทำงานที่ใกล้เคียงกับการทำงานของมนุษย์จึงได้เกิดศาสตร์ที่เรียกว่าเครือข่ายประสาทเทียม (Artificial Neural Network) ขึ้น โดยเครือข่ายประสาทเทียมก็คือระบบการประมวลผลข้อมูลที่มีลักษณะการทำงานเหมือนกับเครือข่ายประสาทของสิ่งมีชีวิตและได้ถูกพัฒนามาจากความรู้ความเข้าใจในโมเดลทางคณิตศาสตร์ของมนุษย์และระบบประสาทวิทยา ซึ่งต้องตั้งอยู่บนพื้นฐานดังต่อไปนี้

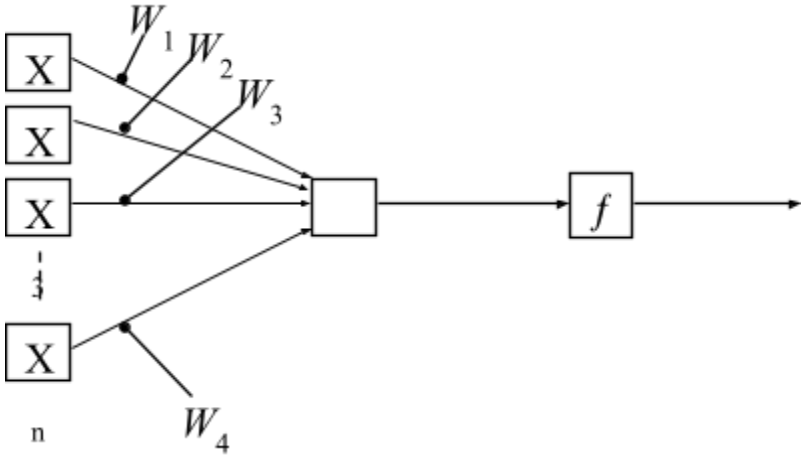
1. การประมวลผลจะเกิดขึ้นกับส่วนที่เรียกว่า นิวรอน (neuron) หรือเซลล์ประสาทหลายๆตัว
2. การส่งข้อมูลระหว่างนิวรอนจะใช้เส้นทางที่เรียกว่าคอนเน็คชั่นลิงค์ (Connection links)
3. แต่ละ Connection link จะประกอบด้วยน้ำหนัก (Weight) ซึ่งถ้าเป็น neural network โดยทั่วไปจะมีการคูณสัญญานที่ส่งมาด้วย weight
4. และแต่ละนิวรอนจะใช้แอคติเวชั่นฟังก์ชั่น (Activation Function) กับค่าอินพุทเพื่อคำนวณหาค่าเอาพุทหรือสัญญาณส่งออก
5. ตัวนิวรอนสามารถส่งสัญญาณเพียง 1 สัญญาณไปยังนิวรอนอื่นใน 1 ช่วงเวลาเท่านั้นแม้ว่าสัญญาณที่ส่งจะเป็นการส่งแบบกระจายไปยังนิวรอนหลายๆตัว

เราสามารถจำแนกนิวรอนเน็ตเวิร์คออกไปได้หลายรูปแบบโดยดูจาก (1) รูปแบบการเชื่อมต่อระหว่างนิวรอน (2) วิธีการในการหาค่า weight บนคอนเน็คชั่นลิงค์ (เรียกว่า training หรือ learning

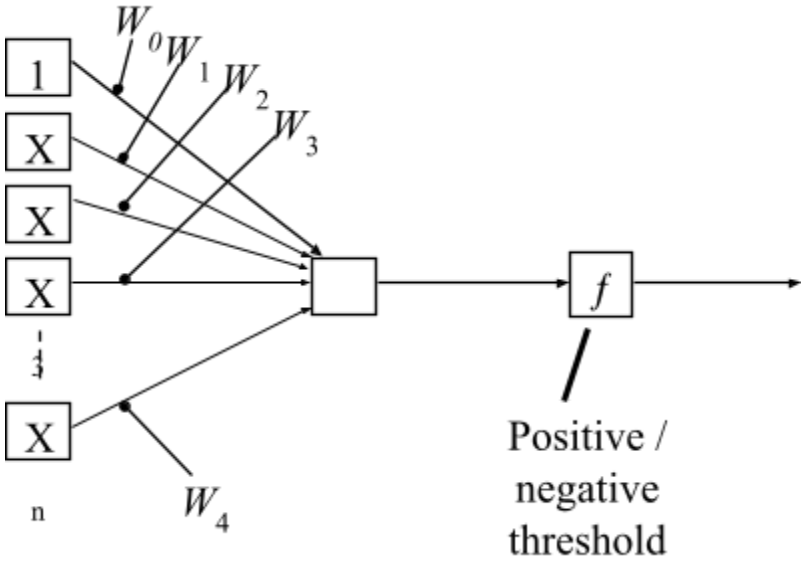
algorithm) และ (3) ดูจาก activation function ซึ่งทำให้ได้นิวรอนเน็ตเวิร์คในแบบต่างๆดังนี้

2.1 Perceptron

นิวรอนเน็ตเวิร์คแบบเพอเซพตรอนถือได้ว่าเป็นพื้นฐานของนิวรอนเน็ตเวิร์คโดยทั่วไป จากรูปที่ 2.1จะเห็นว่าโครงสร้างเครือข่ายในแบบของเพอเซพตรอนมีจำนวนอินพุท n ตัว เขียนสัญลักษณ์ในรูปของเวคเตอร์ ได้เป็น (X_1, X_2, …, X_n) และ weight ที่คอนเน็ชั่นลิงค์ (W_1, W_2, …, W_n) ตัวแปรเหล่านี้โดยปรกติจะมีค่าเป็นเลขจำนวนจริงบวกหรือลบ ตัวเพอเซพตรอนประกอบด้วยตัวประมวลผลที่ทำหน้าที่ในการรวมค่าอินพุทแล้วนำมาเปรียบเทียบกับค่าพิกัด (Threshold Value) เพื่อความสะดวก (ในเชิงสมการคณิตศาสตร์) จะแทนค่าของพิกัดด้วยค่าน้ำหนักอีกตัวหนึ่งโดยเพิ่มที่อินพุท ซึ่งแทนด้วย W_0 และอินพุทที่ค่าพิกัดส่วนนี้จะเป็น 1 ดังรูป 2.2



รูปที่ 2.1 รูปแบบนิวรอนเน็ตเวิร์คแบบเพอเซพตรอน



รูปที่ 2.2 นิวรอนเน็ตเวิร์คแบบเพอเซพตรอนที่แทนที่องค์ประกอบของค่าพิกัด (Threshold) ซึ่งเป็นค่าที่เปรียบเทียบทาง output ด้วยการเพิ่ม weight ทางด้าน input อีกหนึ่งตัวซึ่งทำให้ผลการตรวจสอบผลบวกของ input ว่าเป็นเพียงตัวเลขบวกหรือลบ (Positive / negative threshold)

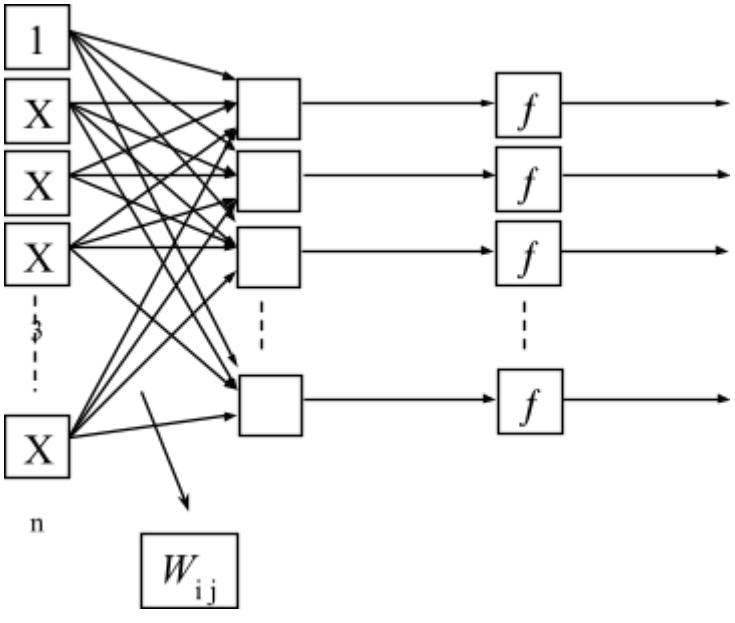
เพอเซพตรอนเป็นระบบนิวรอนเน็ตเวิร์คที่ทำการประมวลผลให้สัญญาณ output เป็นฟังก์ชั่นที่ขึ้นอยู่กับสัญญาณ input และ weight ที่

กำหนดอยู่ในคอนเน็กชั่นลิงค์ input แต่ละตัวและค่าพิกัด ซึ่งค่าพิกัดนี้สามารถปรับค่าได้ นั่นคือ output (o(x)) ของนิวรอนจะส่งผลลัพธ์เป็น 1 ถ้าผลบวกของผลคูณ (g(x)) ระหว่างสัญญาณ input และ weight ของคอนเน็กชั่นลิงค์ที่ input นั้นมีค่ามากกว่าค่า Threshold ที่ถูกำหนดไว้ มิฉะนั้น output จะเป็น 0 ถ้าผลบวกนั้นมีค่าน้อยกว่าค่า Threshold จากการทำงานของเพอเซพตรอนที่ผ่านมานั้นสามารถเขียนเป็นความสัมพันธ์ได้ดังนี้

ฟังก์ชั่นผลบวกของผลคูณ $g(x) = \sum_{i=0}^{n} X_i W_i$

ฟังก์ชั่น output ของเพอเซพตรอน $o(x) = 1$ if $g(x) > 0$ หรือ 0 if $g(x) < 0$

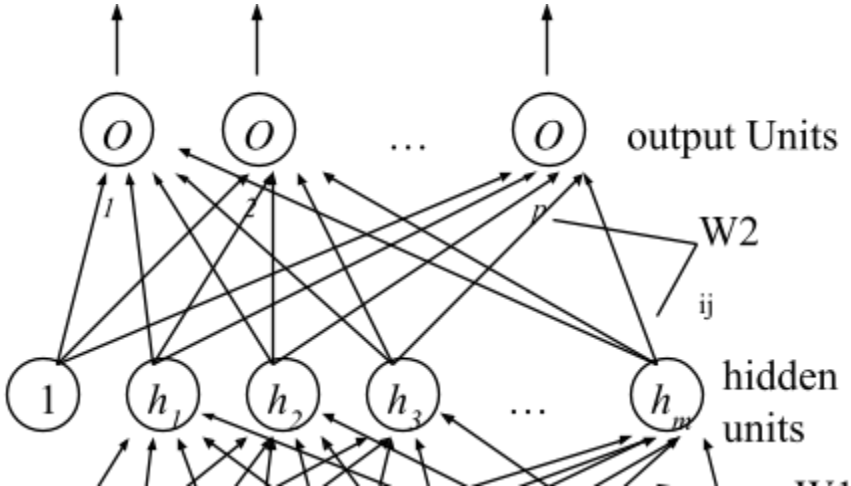
จากความสัมพันธ์จะเห็นว่าเพอเซพตรอนในลักษณะนี้จะให้ผลลัพธ์เป็น 0 หรือ 1 (Binary Function) ถ้าเรานำเพอเซพตรอนในลักษณะเดียวกันนี้หลายๆตัวมาต่อเชื่อมกันดังรูปที่ 2.3 จะทำให้สามารถคำนวณฟังก์ชั่นที่ซับซ้อนขึ้น



รูปที่ 2.3 เพอเซพตรอนที่มีหลาย input และหลาย output

2.2 Back-Propagation

เป็นนิวรอนเน็ตเวิร์คแบบที่ใช้งานได้ดีสำหรับการแก้ไขปัญหาเกี่ยวกับการจดจำที่ซับซ้อนและปัญหาที่มีความไม่แน่นอนในส่วนของ input สามารถยืดหยุ่นได้ แต่คำตอบของ Output นั้นยังคงมีความแน่นอนอยู่



รูปที่ 2.4 Back Propagation Model

จากรูปที่ 2.4 วงกลมหมายถึงนิวรอนอาจเรียกว่า node หรือ unit ก็ได้ จะเห็นว่า Network นี้ประกอบด้วย node ที่เรียงกันอยู่ 3 ชั้น (Layer) คือ Input Layer, Hidden Layer และ Output Layer ลักษณะการไหลของข้อมูลจะเป็นลักษณะเคลื่อนที่ไปข้างหน้าจาก Input Layer ผ่านไปยัง Hidden Layer และเข้าสู่ Output Layer จึงเรียกเน็ตเวิร์คนี้ว่า Multilayer Feedforword Network เน็ตเวิร์ดในลักษณะนี้มีความสามารถในการเรียนรู้โดยที่ต้องมีผู้ช่วยสอนจึงจัดอยู่ในกลุ่มที่เรียกว่า Supervise Learning

สำหรับส่วนประกอบต่างๆที่รวมกันเป็น Black Propagation Neural Network มีรายละเอียดดังต่อไปนี้

1. Input Layer เป็นชั้นที่ทำการรับข้อมูลเข้ามาสู่เน็ตเวิร์คโดยอินพุทข้อมูลจะต้องอยู่ในรูปตัวแปรชุดหนึ่งมิติ (one dimensional array) ข้อมูลแต่ละตัวจะถูกส่งเข้าไปยัง node ใน input layer โดยจะส่งข้อมูล 1 ตัวต่อ 1 node

2. Hidden Layer เป็นชั้นที่เชื่อมโยงระหว่างชั้น input layer และ output layer ซึ่งจำนวน node ในชั้นนี้ไม่ได้มีทฤษฎีใดกำหนดไว้แน่นอนว่าจะต้องมีจำนวน node เท่าไหร่ แต่โดยทั่วไปแล้วการกำหนดจำนวน node ในชั้นนี้จะกำหนดให้น้อยกว่าครึ่งหนึ่งของจำนวน node

ในชั้น input layer ซึ่งจำนวน node ในชั้นนี้มีความสำคัญต่อกระบวนการเรียนรู้คือ ถ้าจำนวน node ในชั้นนี้น้อยเกินไปจะทำให้การแปรผันของค่าต่างๆในเน็ตเวิร์คมีน้อยเกินไปอาจทำให้กระบวนการในการเรียนรู้นั้นล้มเหลวได้ ในขณะเดียวกันถ้าจำนวน node ในชั้นนี้มีมากเกินไปผลดีคือค่าต่างๆในเน็ตเวิร์คมีการแปรผันได้มาก ทำให้การเรียนรู้มีโอกาสประสบความสำเร็จได้สูงแต่ก็ต้องเสียเวลามากในการทำให้เน็ตเวิร์คเกิดการเรียนรู้

3. Output Layer เป็นชั้นที่ทำการส่งค่าต่างๆที่ได้จากการคำนวณในเน็ตเวิร์คออกมาซึ่งค่าที่ได้นี้จะนำไปใช้ใน 2 ลักษณะคือ นำค่าที่ได้ไปคำนวณหาค่าผิดพลาดที่เกิดขึ้นระหว่างเอาท์พุทที่ต้องการ (Target Output) และ เอาท์พุทที่ได้ (Actual Output) แล้วนำค่าผิดพลาดนี้ไปเป็นตัวช่วยในการคำนวณปรับน้ำหนักของสายเชื่อมโยงเพื่อให้เอาท์พุทครั้งต่อไปใกล้เคียงกับเอาท์พุทที่กำหนดมากขึ้น อีกลักษณะหนึ่งคือ นำค่าเอาท์พุทที่ได้ไปผ่านค่า Threshold ตามที่กล่าวมาในกระบวนการของ Perceptron แล้วนำไปเปรียบเทียบกับเอาท์พุทที่กำหนด ถ้าเท่ากันแล้วก็ถือว่าสิ้นสุดกระบวนการในการเปรียบเทียบ

เนื่องจากค่าเอาท์พุทของเน็ตเวิร์คที่ได้ต้องนำไปใช้ในการหาค่าความผิดพลาด ดังนั้นค่าเอาท์พุทที่ได้จะไม่มีการนำไปผ่านค่า Threshold แต่จะต้องทำให้ค่าเอาท์พุทนั้นมีความต่อเนื่อง ซึ่งค่าเอาท์พุทที่จะนำไปใช้จริงนั้นจะได้จากสมการ

$$\text{Output} = \frac{1}{1 + e^{-\text{sum}}}$$

เส้นกราฟที่ได้จากการพล็อตค่าด้วยฟังก์ชั่นนี้จะออกมาเป็นรูปตัว S และเรียกฟังก์ชั่นในลักษณะนี้ว่า Sigmoid Function ในรูป 2.5 แสดงลักษณะของกราฟที่เกิดจากเอาท์พุทของ Perceptron และเอาท์พุทที่เกิดจาก Sigmoid Function คือหากค่า Sum เป็น 0 ค่าเอาท์พุทจะ

ออกมาเป็น 0.5 ซึ่งตรงข้ามกับ perceptron ที่ค่าเอาท์พุทจะออกมาเป็น 0 หรือ 1 ในทำนองเดียวกันถ้า sum มีค่ามากๆเอาท์พุทจะเท่ากับ 1 ถ้า sum น้อยมากๆค่าเอาท์พุทจะเท่ากับ 0

รูปที่ 2.5 ลักษณะความแตกต่างที่ได้จากค่า Output โดยใช้ Binary Function และ Sigmoid Function

4. weight เป็นชุดของเลขจำนวนจริงทั้งบวกและลบชุดหนึ่งที่ได้จากการสุ่มในช่วงเล็กๆเช่น –0.1 ถึง 0.1 หรือ –0.3 ถึง 0.3 เป็นต้น ชุดของ weight นี้จะมี 2 ชุดคือ จาก input layer ไปยัง hidden layer และจาก hidden layer ไปยัง output layer ซึ่งชุดของ weight นี้จะแทนด้วยลูกศรทิศทางเดียวที่เชื่อมระหว่างแต่ละ node ของชั้น input layer ไปยัง node ของชั้น hidden layer และจากแต่ละ node ของชั้น hidden layer ไปยังทุกๆ node ของชั้น output layer

5. จุดอ้างอิง (Reference Point or Bias) เป็น node พิเศษที่มีเฉพาะใน input และ hidden layer เท่านั้นซึ่ง node นี้จะมีค่าเท่ากับ

1 ตลอดกระบวนการเรียนรู้ ถ้าไม่มี node นี้แล้วหากค่าอินพุทเป็น 0 หมดอาจจะทำให้กระบวนการเรียนรู้ล้มเหลวได้

2.2.1 หลักการของ Back Propagation Network

ก่อนที่จะนำนิวรอนเน็ตเวิร์คไปใช้งานในการเรียนรู้รูปแบบใดๆจะต้องทำการฝึกสอนเน็ตเวิร์คก่อนว่าจะรู้จำอะไร ขั้นตอนในการเรียนรู้คือเน็ตเวิร์คจะได้รับชุดของตัวอย่างข้อมูลอินพุทและเอาท์พุทเป็นคู่ เหมือนกับกำลังจะบอกว่าอินพุทเป็นแบบนี้ แล้วเอาท์พุทจะต้องเป็นแบบนี้ สำหรับแต่ละคู่ของข้อมูล เน็ตเวิร์คจะต้องดำเนินการเรียนรู้ 2 ขั้นตอนคือ การส่งผลและการปรับค่า (Propagation and Adaptation) ข้อมูลอินพุทที่ป้อนเข้าสู่นิวรอนในชั้นของ input layer จะถูกประมวลผลและส่งผลลัพธ์ไปยังชั้นต่อไป จนถึงชั้นเอาท์พุท ค่าเอาท์พุทที่ได้จากการประมวลผลของเน็ตเวิร์คจะถูกนำไปเปรียบเทียบกับตัวอย่างเอาท์พุทที่กำหนดมา ค่าของความผิดพลาดที่เกิดขึ้นในการเปรียบเทียบเอาท์พุททั้ง 2 นั้นจะถูกคำนวณขึ้นมาสำหรับ node แต่ละ node

ค่าของความผิดพลาดจะถูกส่งถอยหลังกลับจากชั้นเอาท์พุทไปยัง node ต่างๆของ hidden layer แต่ละ node นี้จะรับค่าของความผิดพลาดนั้นเพียงบางส่วนขึ้นอยู่กับว่า node นั้นเป็นตัวที่ส่งผลมากหรือน้อยไปสู่เอาท์พุทนั้น ขบวนการของการส่งค่าความผิดพลาดกลับไปนั้นจะทำซ้ำสำหรับชั้นถัดลงมาอีก จนกระทั่งทุก node ในเน็ตเวิร์คได้รับส่วนแบ่งความผิดพลาดนั้นหลังจากนั้น weight ที่เชื่อมต่อของแต่ละ node จะถูกปรับค่าไปตามความมากหรือน้อยของสัญญาณค่าผิดพลาดที่ได้รับ เพื่อที่จะทำให้เน็ตเวิร์คปรับตัวเองไปอยู่ในสถานะที่จะทำให้อินพุท-เอาท์พุทได้รับการบันทึก ขบวนการของการส่งผลและการปรับค่า

จะดำเนินต่อไป จนกระทั่งค่าความผิดพลาดต่ำกว่าค่าที่กำหนดไว้ค่าหนึ่งหรือค่าเอาท์พุทที่ได้สามารถอยู่ในช่วงที่ยอมรับได้แล้วจึงหยุด

จุดสำคัญจุดหนึ่งซึ่งเป็นข้อดีของระบบการเรียนรู้แบบ Neural Network คือในช่วงจังหวะที่เน็ตเวิร์คได้รับการฝึกฝน node ต่างๆในแต่ละชั้นจะทำการปรับค่าของ node แตกต่างกันไปเพื่อที่จะทำการรู้จำลักษณะที่แตกต่างของข้อมูลอินพุท ดังนั้นหลังจากที่ได้รับการฝึกฝนแล้วเมื่อจะนำเน็ตเวิร์คไปใช้งานเพื่อการรู้จำอินพุทอีกชุดหนึ่ง ซึ่งอาจมีสัญญาณรบกวนหรือข้อมูลที่ไม่สมบูรณ์ node ต่างๆของ hidden layer จะให้ผลลัพธ์ของเอาพุทได้ถูกต้อง ถ้าอินพุทที่ป้อนเข้ามามีลักษณะใกล้เคียงกับ pattern ที่เคยเรียนรู้ไปแล้ว ในทางตรงกันข้ามลักษณะข้อมูลที่ไม่มีความใกล้เคียงกับ pattern ที่เคยผ่านการเรียนรู้ หน่วยนิวรอนหน่วยนั้นจะไม่ส่งผลลัพธ์ คือจะไม่มีการเดาสุ่มเอาพุทที่จะออกมา

2.2.2 Back Propagation Neural Network Algorithm

สิ่งที่กำหนดให้ : ชุดของคู่อินพุท – เอาท์พุท
สิ่งที่ต้องคำนวณหา : ชุดของweight สำหรับเน็ตเวิร์คที่มี 3 layer ที่สามารถจับคู่อินพุทและเอาท์พุทได้ถูกต้อง

1. กำหนดตัวแปรต่างๆที่ต้องใช้ในการคำนวณดังนี้

A : จำนวน node ใน input layer
B : จำนวน node ใน hidden layer
C : จำนวน node ใน output layer

แต่เนื่องจากใน input layer และ hidden layer ต้องมี Bias อีก 1 ตัว ดังนั้นตัวเลขที่ใช้สำหรับการคำนวณในโปรแกรมจะเป็น (0,…,A) , (0,…,B) และ (1,…,C) สำหรับ input, hidden และ output layer ตามลำดับ

x_j : node แต่ละตัวใน input layer

h_j : node แต่ละตัวใน hidden layer

o_j : node แต่ละตัวใน output layer (ที่หาได้จากเน็ตเวิร์ค)

y_j : ชุดของเอาท์พุทที่กำหนด

$w1_{ij}$: weight ระหว่าง input layer กับ hidden layer

$w2_{ij}$: weight ระหว่าง hidden layer กับ output layer

2. ทำการกำหนดค่าเริ่มต้นของน้ำหนักในเน็ตเวิร์คโดยการสุ่มตัวเลขระหว่าง –0.5 ถึง 0.5

$w1_{ij}$ = random(-0.5,0.5) for all i = 0…A, j = 1…B

$w2_{ij}$ = random(-0.5,0.5) for all i = 0…B, j = 1…C

3. กำหนดค่าของ Bias สำหรับ input และ hidden layer ซึ่งจะมีค่าคงที่ตลอดคือ 1

$x_0 = 1$

$h_0 = 1$

4. ทำการเลือกชุดค่าของ input และ output เช่นถ้าเลือก input เป็น x_i แล้ว output ก็ต้องเป็น y_i

5. คำนวณหาค่าของแต่ละ node ใน hidden layer จากสมการ

$$h_j = \frac{1}{1+e^{-\sum_{i=0}^{A} W1_{ij} x_i}} \quad \text{for all } j = 1, \ldots, B$$

6. คำนวณหาค่าของแต่ละ node ใน output layer จากสมการ

$$h_j = \frac{1}{1+e^{-\sum_{i=0}^{B} W2_{ij} y_i}} \quad \text{for all } j = 1, \ldots, C$$

7. คำนวณหาค่าความผิดพลาดที่เกิดขึ้นใน output layer ระหว่าง output ที่หาได้กับ output ที่กำหนดให้จากสมการ

$$\delta 2_j = o_j(1 - o_j)(y_i – o_j) \text{ for all } j = 1, \ldots , C$$

ซึ่งสมการนี้มาจากการหาอนุพันธ์อันดับหนึ่งของส่วนที่เป็น input ของ output layer คูณด้วยค่าผิดพลาดระหว่าง output ที่ได้กับ output ที่กำหนด

$$\text{Input ของ output layer } (i_0) = \sum_{i=0}^{n} W2_{ij} h_i \text{ for all } j = 1 , \ldots , C$$

$$\delta 2_j = f\,'(i_0)(y_i – o_j) \text{ for all } j = 1, \ldots , C$$

8. คำนวณหาค่าความผิดพลาดที่เกิดขึ้นใน hidden layer จากสมการ

$$\delta 1_j = h_j(1 - h_j) \sum_{1}^{c} \delta 2_j W2_{ij} \quad \text{for all } j = 1 , \ldots , B$$

9. ปรับค่า weight ระหว่าง hidden layer กับ output layer โดยคูณ learning rate (η) จากสมการ

$$\Delta W2_{ij} = \eta \delta 2_j h_i \quad \text{for all } i = 0 , \ldots , B , j = 1 , \ldots , C$$

$$W2_{ij}(\text{New}) = W2_{ij}(\text{old}) + \Delta W2_{ij}$$

10. ปรับค่า weight ระหว่าง input layer กับ hidden layer จากสมการ

$$\Delta W1_{ij} = \eta\, \delta 1_j\, x_i \qquad \text{for all } i = 0, \ldots, A,\ j = 1, \ldots, B$$

$$W1_{ij}(\text{New}) = W1_{ij}(\text{old}) + \Delta W1_{ij}$$

กลับไปยังขั้นตอนที่ 4 แล้วทำซ้ำไปเรื่อยๆจนกระทั่งหมดชุดคู่อินพุท-เอาท์พุท และค่าความผิดพลาดเกิดขึ้นอยู่ในช่วงที่ยอมรับได้ เช่น

Output ที่กำหนด	1	0	0	1
Output ที่ได้จากระบบ	0.828	0.003	0.034	0.887
Output ที่ผ่านค่า Threshold(0.5)	1	0	0	1

นั่นแสดงว่า Output ที่ได้มีค่าเท่ากับ Output ที่กำหนดในช่วงที่ยอมรับได้

บทที่ 3
ขั้นตอนการดำเนินงาน

3.1 การดำเนินการ

ขั้นตอนการดำเนินงานทั้งหมดสามารถแบ่งออกเป็น 6 ขั้นตอนต่อไปนี้

1. กำหนดปัญหา วัตถุประสงค์และขอบเขตของการแก้ไขปัญหา
2. ศึกษาและทำความเข้าใจในข้อกำหนดต่างๆของโมเดล Backpropagation Neural Network รวมไปถึงกระบวนการในการทำงานของโมเดลนี้
3. ศึกษารูปแบบของข้อมูลและทำการแปลงข้อมูลให้อยู่ในรูปแบบที่จะสามารถนำไปใช้ในการบวนการเรียนรู้ของ Backpropagation Neural Network
4. ศึกษาอุปกรณ์อาร์ดแวร์และซอฟต์แวร์เพื่อที่จะใช้ในการพัฒนาระบบ
5. ออกแบบและพัฒนาโปรแกรมให้สามารถใช้งานได้ นำข้อมูลมาใช้กับโปรแกรม ทำการปรับค่าต่างๆและทำการวิเคราะห์ว่าโมเดลแบบไหนของ Backpropagation Neural Network ที่ให้ค่าที่ถูกต้องมากที่สุด โดยใช้เครื่องมือวัดประสิทธิภาพคือ Efficiency Index (EI), Root Mean Square Error, Mean Absolute Error (MAD)

จากขั้นตอนทั้ง 6 ขั้นตอนนั้น ในขั้นตอนที่ 1 และ 2 ได้กล่าวไว้แล้วในบทที่ 1 และ 2 ตามลำดับ ส่วนขั้นตอนที่ 3 , 4 และ 5 เป็นขั้นตอนที่จะกล่าวถึงในบทนี้ และขั้นตอนที่ 6 จะกล่าวถึงในบทที่ 4

3.2 รูปแบบข้อมูลและการแปลงข้อมูลก่อนนำไปใช้ในกระบวนการเรียนรู้

โดยทั่วไปข้อมูลที่เหมาะสมจะนำมาใช้ในการทำนายระดับน้ำจะประกอบไปด้วยข้อมูลระดับน้ำรายวัน และข้อมูลปริมาณน้ำฝน ดังนั้นผู้ทำวิจัยจึงได้ทำการขอข้อมูลที่จะนำมาใช้ในการวิเคราะห์เป็นข้อมูลของระดับน้ำรายวันของลุ่มแม่น้ำยม บริเวณอำเภอเมือง จังหวัดสุโขทัย และปริมาณน้ำฝนรายวันของบริเวณดังกล่าวเป็นเวลา 5 ปี ซึ่งข้อมูลทั้งสองได้มีการบันทึกแยกจากกันจึงต้องมีการรวมข้อมูลเหล่านี้เข้าด้วยกัน ส่วนตัวอย่างของข้อมูลจะแสดงไว้ในภาคผนวก ก ดังนั้นข้อมูลที่ได้จะอยู่ในรูปแบบดังนี้

น้ำฝน (mm.)	ระดับน้ำ (meter)
30.5	44.54
0	44.53
…	…
12.3	44.34
…	…
0	47.64
0	47.50

ตารางที่ 3.1 ปริมาณน้ำฝนและระดับน้ำรายวัน

รูปที่ 3.1

จะเห็นได้ว่าข้อมูลที่ได้ในแต่ละ record ยังมีจำนวนน้อยเกินไปที่จะใช้ในการเรียนรู้ของโครงข่าย และเนื่องจากว่าปริมาณน้ำของวันปัจจุบันจะขึ้นอยู่กับปริมาณฝนที่ตกลงมาของวันก่อนหน้า ดังนั้นจึงควรมีการเพิ่มข้อมูลปริมาณน้ำฝนและระดับน้ำใน 1 record เข้าไปอีกโดยจะ

เพิ่มข้อมูลของวันก่อนหน้าวันปัจจุบัน 1 วัน และ 2 วัน ดังนั้นรูปแบบของข้อมูลที่ได้จะอยู่ในลักษณะดังนี้

น้ำฝนก่อนหน้า 2 วัน (mm.)	น้ำฝนก่อนหน้า 1 วัน (mm.)	น้ำฝน (mm.)	ระดับน้ำก่อนหน้า 2 วัน (meter)	ระดับน้ำก่อนหน้า 1 วัน (meter)	ระดับน้ำ (meter)
0	0	30.5	0	0	44.54
0	30.5	12.5	0	44.54	44.53
30.5	12.5	0	44.54	44.53	44.40
…	…	…	…	…	…
12.3	0	0	44.35	44.25	44.34
…	…	…	…	…	…
0	30.8	18.9	47.62	47.65	47.64
30.8	18.9	0	47.65	47.64	47.67

ตารางที่ 3.2 ปริมาณน้ำฝนและระดับน้ำรายวันเป็นเวลา 3 วัน

จากตารางข้างต้นจะเห็นข้อมูลนี้คือข้อมูลที่จะใช้เป็นอินพุทสำหรับ Backpropagation Neural Network ยังขาดข้อมูลส่วนที่เป็นเอท์พุทเพื่อใช้ในการเปรียบเทียบ(แนะแนวทาง) ให้กับ Backpropagation Neural Network ซึ่งจำเป็นอย่างยิ่งในช่วงที่เน็ตเวิร์คกำลังทำการเรียนรู้ ดังนั้นจึงต้องทำการเพิ่มข้อมูลเข้าไปในแต่ละ record อีก และข้อมูลที่เราจะเพิ่มเข้าไปนี้ก็คือข้อมูลของระดับน้ำในวันรุ่งขึ้น ก็จะได้ข้อมูลดังนี้

น้ำฝนก่อนหน้า 2 วัน (mm.)	น้ำฝนก่อนหน้า 1 วัน (mm.)	น้ำฝน (mm.)	ระดับน้ำก่อนหน้า 2 วัน (meter)	ระดับน้ำก่อนหน้า 1 วัน (meter)	ระดับน้ำ (meter)	ระดับน้ำวันรุ่งขึ้น (meter)
0	0	30.5	0	0	44.54	44.53
0	30.5	12.5	0	44.54	44.53	44.40
30.5	12.5	0	44.54	44.53	44.40	44.45
…	…	…	…	…	…	…
12.3	0	0	44.35	44.25	44.34	44.40
…	…	…	…	…	…	…
0	30.8	18.9	47.62	47.65	47.64	44.67
30.8	18.9	0	47.65	47.64	47.67	0

ตารางที่ 3.3 ปริมาณน้ำฝนและระดับน้ำรายวันเป็นเวลา 3 วัน และระดับน้ำวันรุ่งขึ้น

จากการเพิ่มข้อมูลเข้าไปจะพบว่า record ที่ 1 กับ 2 และ record สุดท้ายจะมีข้อมูลที่ไม่ถูกต้องเนื่องจากไม่มีข้อมูลจริงของปริมาณน้ำฝนวันก่อนหน้า 2 วันและ 1 วัน และปริมาณระดับน้ำก่อนหน้า 2 วัน และ 1 วัน สำหรับเรคอร์ดที่ 1 และ 2 จึงใส่ค่าเป็นศูนย์ และไม่มีข้อมูลของระดับน้ำวันรุ่งขึ้นของเรคอร์ดสุดท้าย จึงต้องทำการตัด record ทั้ง 3 record นี้ทิ้งไป

3.3 อุปกรณ์ฮาร์ดแวร์และซอฟต์แวร์ที่จะใช้ในการพัฒนาระบบ

ในการพัฒนาโปรแกรมนั้นจะต้องคำนึงถึง อุปกรณ์ฮาร์ดแวร์และซอฟต์แวร์ที่เป็นที่ยอมรับในวงการและมีผู้นิยมใช้กันอย่างแพร่หลาย เพื่อที่จะเปิดโอกาสให้ผู้ที่สนใจต้องการจะนำงานวิจัยนี้ไปพัฒนาต่อไปอุปกรณ์ฮาร์ดแวร์และซอฟต์แวร์ที่จะใช้ในการพัฒนาระบบมีดังต่อไปนี้

3.3.1 ไมโครคอมพิวเตอร์ 1 ชุด

- มีหน่วยความจำอย่างน้อย 32 MB. และ เป็นเครื่องที่มีความสามารถในระดับ Pentium 100 MHz. ขึ้นไป
- จอภาพสี มีความละเอียดไม่ต่ำกว่า 640 x 480 เป็นอย่างน้อย
- ฮาร์ดดิสก์ความจุไม่ต่ำกว่า 500 MB.

3.3.2 โปรแกรม MS Access ตั้งแต่เวอร์ชั่น 97 ขึ้นไป

3.3.3 ชุดพัฒนาโปรแกรมภาษา JAVA , Java Development Kit เวอร์ชั่น 1.2 หรือเวอร์ชั่นที่สูงกว่า รวมไปถึงโปรแกรม JAVA run time ซึ่งสามารถทำงานภายใต้ระบบปฏิบัติการ Window

3.3.4 ระบบปฏิบัติการ Window เวอร์ชั่น 98

3.4 การออกแบบและพัฒนาโปรแกรม

จากการที่ระบบโครงข่ายประสาทเทียมจะต้องมีการรับข้อมูลเข้ามาใช้ในการเรียนรู้ของโครงข่ายดังนั้นโปรแกรมที่จะทำการพัฒนาจึงต้องสามารถเลือกเปิดไฟล์ที่จะใช้เป็นข้อมูลอินพุตให้กับโครงข่าย โดยไฟล์ข้อมูลนี้จะอยู่ในรูปแบบของเท็กซ์ไฟล์เนื่องจากหากมีการเก็บข้อมูลโดยใช้ฐานข้อมูลไม่ว่าจะเป็น MS Access ,Excel หรือฐาน

ข้อมูลในรูปแบบอื่นๆ ก็สามารถที่จะแปลงให้เป็นเท็กซ์ไฟล์ได้ซึ่งจะทำให้เกิดความยืดหยุ่นในการนำไปใช้งาน

ในหัวข้อ 3.2 ที่ได้กล่าวถึงรูปแบบของข้อมูลนั้น ข้อมูลที่ได้จากขั้นตอนดังกล่าวเมื่อนำมาเก็บในรูปเท็กซ์ไฟล์จะใช้ช่องว่างในการแบ่งแยกข้อมูลแต่ละฟิลด์ และข้อมูลในแต่ละเรคอร์ดจะแยกกันโดยใช้การขึ้นบรรทัดใหม่ และในบรรทัดบนสุดของเท็กซ์ไฟล์จะต้องเป็นชื่อฟิลด์ ดังนี้

R2	R1	R	H2	H1	H	Hnew	
0	0	0	44.34		44.35	44.32	44.26
0	0	0	44.35		44.32	44.26	44.24
0	0	40.8	44.32		44.26	44.24	44.34
0	40.8	12.3	44.26		44.24	44.24	44.32
40.8	12.3	0	44.24		44.24	44.32	44.27

…

3.5 โครงสร้างของโปรแกรมและการทำงาน

โครงสร้างของโปรแกรมจะประกอบด้วย 4 ขั้นตอนหลัก ดังนี้

1. ส่วนที่ทำหน้าที่รับอินพุตไฟล์ (เท็กซ์ไฟล์) และแปลงข้อมูลลงฐานข้อมูล MS Access
2. ส่วนที่รับข้อมูลการเซตค่าโมเดลที่ต้องการซึ่งค่าที่ทำการเซตประกอบด้วย ค่าจำนวน input node , hidden node , output node , learning rate, momentum , จำนวนรอบ และค่า maximum error
3. ส่วนที่ทำการสร้างโมเดลโครงข่ายประสาทเทียม และทำการเรียนรู้
4. ส่วนที่ทำการแสดงกราฟเปรียบเทียบข้อมูลจากการเรียนรู้และทดสอบ พร้อมทั้งแสดงค่า EI, RMSE และ MAD

3.5.1 ส่วนที่ทำหน้าที่รับอินพุตไฟล์ (เท็กซ์ไฟล์) และแปลงข้อมูลลงฐานข้อมูล MS Access

เมื่อมีการเลือกข้อมูลอินพุตไฟล์ที่ใช้ในการเรียนรู้และทดสอบ โปรแกรมจะทำการแปลงข้อมูลจากเท็กซ์ไฟล์ให้อยู่ในรูปของฐานข้อมูล MS Access โดยจะทำการสร้างตารางชื่อ input และทำการตรวจดูเท็กซ์ไฟล์ว่าในบรรทัดแรกมีฟิลด์ข้อมูลเท่าใด เพื่อนำมาสร้างเป็นชื่อฟิลด์ของตาราง จากนั้นก็ทำการแปลงข้อมูลในแต่ละบรรทัดให้เป็นเรคอร์ดของตาราง input จากนั้นโปรแกรมจะทำการสร้างส่วนของ user interface โดยจะแสดงจำนวน input node , hidden node, output node และค่าอื่นๆเช่น learning rate และ momentum โดยค่า output node จะมีค่า default เป็น 1 และ input node จะมีค่า default เท่ากับจำนวนฟิลด์ทั้งหมดของตาราง input ลบด้วย 1

3.5.2 ส่วนที่รับข้อมูลการเซตค่าสำหรับโมเดลที่ต้องการ

ก่อนที่จะเริ่มกระบวนการเรียนรู้ สามารถที่จะทำการเปลี่ยนแปลงค่า input node , hidden node, output node, leaning rate, momentum, maximum error, epoch (จำนวนรอบ) , training data, testing data โดยในกรณีของค่า input node และ output node จะมีความสัมพันธ์กัน หากมีการเปลี่ยนแปลงค่าใดค่าหนึ่งจะมีการแก้ไขค่าของอีกตัวหนึ่งด้วยเพื่อให้สัมพันธ์กับจำนวนข้อมูลที่รับเข้ามา สำหรับค่า momentum และ learning rate ค่าที่รวมกันจะต้องไม่เกิน 1.0 หากมีการเปลี่ยนแปลงค่าใดค่าหนึ่ง จะมีการแก้ไขค่าของอีกตัวหนึ่งด้วยเพื่อให้สัมพันธ์กัน นอกจากนี้ข้อมูลที่จะนำมาทำการเรียนรู้และทดสอบจะอยู่ในอัตรา 3 ต่อ 2 ตามลำดับ หรือสามารถตั้งค่าอื่นได้ตามต้องการแต่รวมกันจะไม่เกินจำนวนข้อมูลทั้งหมดที่มีอยู่

3.5.3 ส่วนที่ทำการสร้างโมเดลโครงข่ายประสาทเทียมและทำการเรียนรู้

เมื่อมีการกดปุ่ม training โปรแกรมจะทำการสร้างโครงข่ายประสาทเทียมตามค่าที่ตั้งไว้ โดยใช้คลาส BackProp ในการสร้าง ซึ่งค่าพารามิเตอร์ที่ส่งให้คลาสนี้จะประกอบด้วย input pattern , output pattern, input node, output node, hidden node, learning rate, momentum, maximum error และ epoch นอกจากนี้ยังมีการเตรียมพื้นที่สำหรับเก็บ weight สำหรับ input layer ไป hidden layer และ weight สำหรับ hidden layer ไป output layer แล้วจึงทำการเรียนรู้

กระบวนการเรียนรู้ของโครงข่ายจะสิ้นสุดลงใน 2 กรณี คือ เมื่อถึงจำนวนรอบที่ได้ตั้งไว้ หรือค่า error ที่เกิดขึ้นน้อยกว่าค่าที่ตั้งไว้ ซึ่งในแต่ละรอบของการเรียนรู้จะมีการคำนวณค่า error เพื่อใช้หาค่า sum square error ซึ่งจะใช้เปรียบเทียบกับค่า maximum error หากค่า error ที่ได้น้อยกว่าค่า maximum error โครงข่ายจึงจะหยุดการเรียนรู้ จากนั้นจะทำการบันทึกค่า weight ลงในฐานข้อมูลในตารางชื่อ weight โดย 1 เรคอร์ดจะหมายถึง 1 ลิงค์ ซึ่งตารางนี้จะประกอบด้วยฟิลด์ดังนี้

1. StartFrom เป็นฟิลด์ที่ใช้สำหรับบันทึกว่าเป็นโหนดเริ่มต้นของลิงค์ของโครงข่ายประสาทเทียม
2. EndAt เป็นฟิลด์ที่ใช้สำหรับบันทึกว่าเป็นโหนดปลายทางของลิงค์ของโครงข่ายประสาทเทียม
3. WeightValue เป็นฟิลด์ที่ใช้สำหรับบันทึกค่า weight สำหรับลิงค์ของโหนดคู่นั้นๆ

เพื่อให้รู้ว่าค่า ที่เก็บในฟิลด์ WeightValue เป็นของลิงค์ใด จึงต้องมีการกำหนดรูปแบบของข้อมูลในฟิลด์ StartFrom และ EndAt ใน

รูปของ XN โดย X จะหมายถึงชนิดของ layer และ N หมายถึงโหนดที่เท่าไหร่ใน layer นั้นๆ ซึ่งค่า X คือ

- I แสดงถึงโหนดที่อยู่ใน input layer
- H แสดงถึงโหนดที่อยู่ใน hidden layer
- O แสดงถังโหนดที่อยู่ใน output layer

ตัวอย่างเช่น StartFrom เป็น I1 และ EndAt เป็น H2 หมายถึงลิงค์จากโหนดที่ 1 ของ input layer ไปยังโหนดที่ 2 ของ hidden layer

หลังจากที่โปรแกรมทำการบันทึกค่า weight เสร็จสิ้น โปรแกรมจะทำการนำข้อมูลที่เหลือจากการ training (คือข้อมูลในส่วน testing) นำมาผ่านโครงข่ายประสาทเทียมเพื่อหาผลลัพธ์จากการทำนายของโครงข่ายนั้นแล้วเก็บค่าผลลัพธ์ที่ได้ไว้สำหรับการสร้างกราฟต่อไป

3.5.4 ส่วนที่กราฟเปรียบเทียบข้อมูลจากการเรียนรู้และทดสอบพร้อมทั้งแสดงค่า EI, RMSE และ MAD

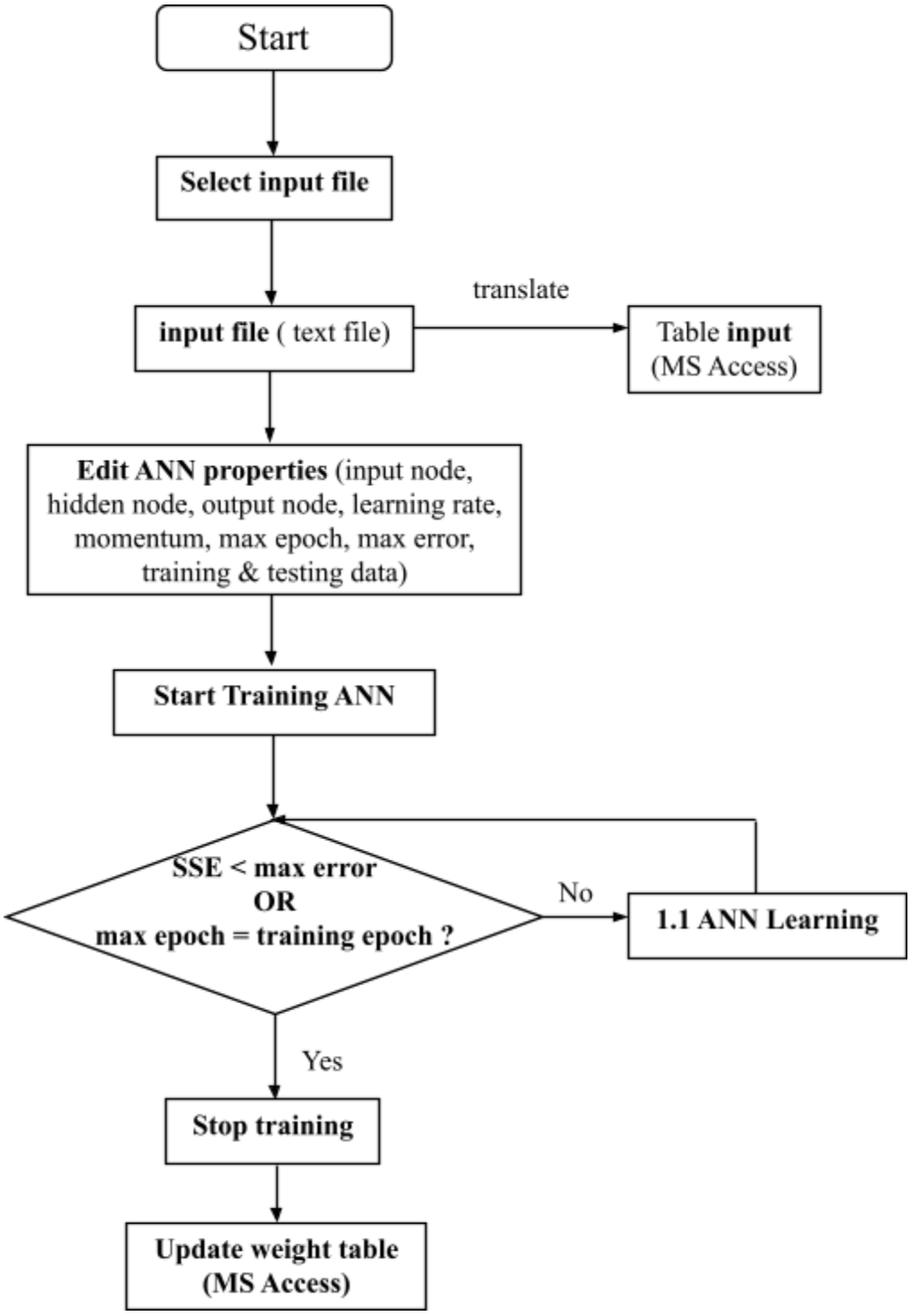
ในขั้นตอนนี้จะนำผลลัพธ์ที่ได้จากขั้นตอนการเรียนรู้ และทดสอบโครงข่ายประสาทเทียมมาทำการสร้างกราฟ โดยกราฟที่สร้างจะประกอบกราฟ 4 รูปแบบ ดังนี้

1. กราฟแสดงค่า Sum Square Error ในช่วงของการ training ในแต่ละรอบ
2. กราฟเปรียบเทียบข้อมูลจริงและข้อมูลที่ได้จากการทำนายของโครงข่ายในช่วงของการ training
3. กราฟเปรียบเทียบข้อมูลจริงและข้อมูลที่ได้จากการทำนายของโครงข่ายในช่วงของการ testing
4. กราฟ dispersion แสดงการกระจายของข้อมูลจริงและข้อมูลจากการทำนาย

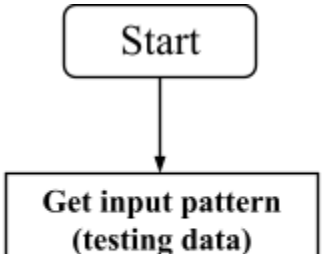
สำหรับค่า EI, RMSE และ MAD จะแสดงไว้ส่วนบนของกราฟเปรียบเทียบข้อมูลจริงและข้อมูลที่ได้จากการทำนายของโครงข่ายในช่วงของการ training

3.6 Program FlowChart

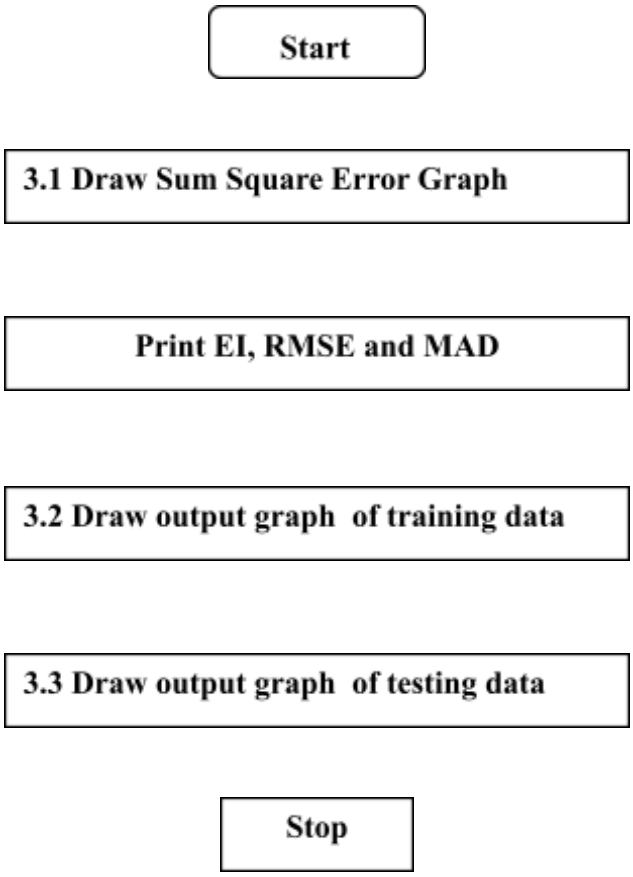
3.6.1 Select Input Data & Start Training



3.6.2. Testing Artificial Neural Network



3.6.3. Generate Graph



บทที่ 4
ผลการศึกษา

การจะวิเคราะห์ว่าโครงข่ายแบบใดที่มีประสิทธิภาพสามารถดูได้จากค่า EI, RMSE, MAD ซึ่งวิธีการคำนวณค่าต่างๆมีดังต่อไปนี้

ก่อนการคำนวณค่า EI และ RMSE จะต้องทำการคำนวณค่าความผิดพลาด (SSE = Sum Squared Error) ของข้อมูลทุกชุดก่อนดังนี้

$$V_i = t_i - z_i$$

$$SSE = \sum_{i=1}^{N} V_i^2$$

เมื่อ V_i = ความผิดพลาดของแต่ละชุดของข้อมูล N ชุด

t_i = target output

z_i = output จากการคำนวณ

1. Efficiency Index (EI)

$$EI = (ST - SSE) / ST$$

$$ST = \frac{1}{N} \sum_{i=1}^{N} (t_i - \bar{t})^2$$

เมื่อ $\bar{t}$ = ค่าเฉลี่ยของ target output

2. Root Mean Squared Error (RMSE)

$$RMSE = (SSE/N)^{1/2}$$

3. Mean Absolute Error (MAD)

$$MAD = \frac{1}{N} \sum_{i=1}^{N} |t_i - z_i|$$

โครงข่ายใดให้ค่า EI สูงสุดและค่า RMSE, MAD ต่ำสุดถือว่าเป็นโครงข่ายที่ดีที่สุด

4.1 ผลการทดสอบโครงข่าย

โครงข่าย	EI	RMSE	MAD	Epoch
6-2-1	-208.47	0.015306	0.013804	325
6-3-1	-153.08	0.013127	0.011596	293
6-4-1	-152.21	0.013091	0.014129	270
6-5-1	-63.36	0.008484	0.007183	212
6-6-1	-71.28	0.008991	0.008243	160
6-7-1	-66.84	0.008109	0.007947	141
6-8-1	-56.05	0.007988	0.006304	122
6-9-1	-55.29	0.007934	0.005919	130

ตารางที่ 4-1 เปรียบเทียบค่า EI, RMSE , MAD และ จำนวนรอบของโครงข่ายแต่ละแบบ

จากตารางที่ 4-1 แสดงผลจากการทดลองโครงข่ายแบบต่างๆรวมทั้งจำนวนรอบการทำงานพบว่า โครงข่ายที่ดีที่สุด คือโครงข่ายที่มีจำนวนโหนดใน hidden layer ตั้งแต่ 5 โหนดไปจนถึง 9 โหนด และ

โครงข่ายที่ให้ค่า EI มากที่สุด และให้ค่า RMSE กับ MAD น้อยที่สุด คือโครงข่าย 6-9-1 แต่หากสังเกตตั้งแต่โครงข่าย 6-5-1 ไปจนถึงโครงข่าย 6-9-1 จะพบว่า ค่า EI, RMSE และ MAD ไม่แตกต่างกันมากนัก

บทที่ 5
สรุปผล

การศึกษาครั้งนี้ได้ทำการพัฒนาโปรแกรมโครงข่ายใยประสาทเทียมโดยใช้ภาษาจาวาและพัฒนาบนระบบปฏิบัติการวินโดวส์ เพื่อใช้ในการทำนายระดับน้ำรายวันของลุ่มแม่น้ำยม อำเภอเมือง จังหวัดสุโขทัย โดยเลือกใช้โครงข่าย Back Propagation Neural Network มาเป็นต้นแบบ อย่างไรก็ตามการศึกษาครั้งนี้ได้ทดสอบโครงข่ายใยประสาทเทียม 8 โครงข่ายแล้วพิจารณาว่าโครงข่ายที่ดีที่สุด ซึ่งพอจะสรุปผลได้ดังนี้

1. โครงข่าย 6-9-1 เป็นโครงข่ายที่ดีที่สุด สำหรับการศึกษาในครั้งนี้ โดยให้ค่า EI สูงที่สุด และให้ค่า RMSE และ MAD ต่ำที่สุด
2. การมีจำนวนโหนดในชั้น hidden layer ของโครงข่ายมากขึ้นไม่ได้หมายความว่าผลการทำนายจะดีขึ้น
3. การมีจำนวนโหนดในชั้น hidden layer ของโครงข่ายมากขึ้นจะมีผลให้โครงข่ายสามารถเรียนรู้ได้เร็วขึ้น
4. การประยุกต์ใช้ระบบโครงข่ายใยประสาทเทียมสามารถใช้งานได้ดีในการทำนายระดับน้ำรายวัน อย่างไรก็ตามยังคงมีความยุ่งยากในการออกแบบโครงข่ายรวมทั้งการกำหนดค่า weight เริ่มต้นให้กับ
5. จำนวนข้อมูลที่นำมาใช้ในการทดลองมีผลต่อการทำนายผลของระบบ

ภาคผนวก

วิธีการติดตั้งและใช้งานโปรแกรม

1. การติดตั้งโปรแกรม

ก่อนที่จะใช้งานโปรแกรมจะต้องทำการเตรียมติดตั้งซอฟต์แวร์ที่เกี่ยวข้องกับโปรแกรม ซึ่งประกอบด้วยซอฟต์แวร์ต่างๆดังนี้

1. Java Software Development Kit (Java SDK) เวอร์ชั่น 1.2 ขึ้นไป สามารถทำการดาวน์โหลดได้ที่ www.java.sun.com
2. ODBC driver ซึ่งจะรวมอยู่ในโปรแกรม MS office
3. Edit plus 2.0

เมื่อทำการติดตั้งโปรแกรมดังกล่าวข้างต้นแล้ว ต้องทำการเซตค่าให้กับโปรแกรม Edit plus ให้สามารถทำการคอมไพล์ซอสโค้ดภาษาจาวา และสั่งให้โปรแกรมทำงาน ดังนี้

- เปิดโปรแกรม Edit plus จากนั้นไปที่เมนู Tool เลือกคำสั่ง Configure User Tools…

- กดปุ่ม Add Tool >> เลือกคำสั่ง program ให้ใส่ข้อความในเท็กซ์บ็อกซ์ต่างๆดังนี้
  i. Menu text : Java Compiler
  ii. Command : ให้เลือกโปรแกรม javac.exe ซึ่งอยู่ในไดเรคทอรี่ bin ที่ได้ทำการติดตั้ง Java SDK ไว้
  iii. Argument : ให้เลือกคำสั่ง file name

iv. Initial directory : ให้เลือกคำสั่ง file directory

- กดปุ่ม Add Tool >> เลือกคำสั่ง program ให้ใส่ข้อความในเท็กซ์บ็อกซ์ต่างๆดังนี้
  i. Menu text : Java Run Time
  ii. Command : ให้เลือกโปรแกรม java.exe ซึ่งอยู่ในไดเรคทอรี่ bin ที่ได้ทำการติดตั้ง Java SDK ไว้
  iii. Argument : ให้เลือกคำสั่ง file name without extension
  iv. Initial directory : ให้เลือกคำสั่ง file directory

- เมื่อเลือกเมนูอีกครั้งก็จะปรากฏคำสั่ง Java Compiler และ Java Run Time

ก่อนที่จะทำการรันโปรแกรมจะต้องทำการเซตค่า ODBC ให้รู้จักกับฐานข้อมูล MS Access ก่อนเนื่องจากโปรแกรมมีการเรียกใช้ข้อมูลจากฐานข้อมูล MS Access ซึ่งมีวิธีการดังนี้

- ไปที่เมนู start เลือกคำสั่ง control panel ในกรณีของ windows 98 และ windows ME จะสามารถมองเห็นไอคอนของ ODBC ได้ทันที หากเป็น windows 2000 หรือ windows XP จะต้องคลิ๊กเข้าไปที่ Admin's Tool แล้วจึงทำการเปิด ODBC ขึ้นมาเซตค่าดังนี้
  i. เลือก tab System DSN กดปุ่ม Add เลือก driver เป็น Microsoft Access Database (*.mdb) แล้วจึงกดปุ่ม finish

ii. ตั้งค่า data source name เป็น predict จากนั้นเลือก ฐานข้อมูลชื่อ BackProp.mdb ซึ่งอยู่ในไดเรคทอรี่เดียวกับซอสโปรแกรม

สำหรับการเปิดโปรแกรมให้ทำงานสามารถทำได้โดยใช้โปรแกรม Edit Plus เปิดไฟล์ชื่อ supanit1.java แล้วเรียกใช้คำสั่ง java compiler ทำการแปลงซอสโค้ดให้เป็นไฟล์นามสกุล class (ไบต์โค้ด) จากนั้นจึงเรียกใช้คำสั่ง java rum time ทำการเปิดโปรแกรม

2. **การใช้งานโปรแกรม**

การเริ่มใช้งานโปรแกรมก่อนอื่นต้องทำการเลือกไฟล์ที่จะใช้เป็นข้อมูลสำหรับการฝึกฝนโครงข่าย โดยไปที่คำสั่ง Create Model แล้วกดปุ่ม input file ทำการเลือกไฟล์ input2.txt จากนั้นโปรแกรมจะสร้างหน้าจอสำหรับเซตค่าต่างๆ

บรรณานุกรม

วีระศักดิ์ ซึงถาวร. Fundamental of Java programming volume 1. ซีเอ็ด, 2541.

Fausett, L. “Fundamentals of Neural Networks” Florida. Prentice-Hall, 1994.

Hagan,M, T. “NEURAL NETWORKS (COMPUTER SCIENCE).” Boston. PWS Pub, 1996.

Linden,V, Peter.“Just Java 1.2” Palo Alto. Calif. Sun Micro systems, 1999.

Morelli, R. “Java Java Java! : object-oriented problem solving.” Upper Saddle River. N.J. Prentice-Hall, 2000.