ภาคผนวก ก

การพัฒนาโปรแกรมด้วย MATLAB

โปรแกรม MATLAB (Matrix Laboratory) เป็นโปรแกรมที่ใช้เพื่อคำนวณเชิงตัวเลข แสดงกราฟิกที่ซับซ้อน และสร้างแบบจำลองที่ทำให้สามารถมองเห็นภาพการทำงานได้ง่ายและชัดเจน นิยมนำโปรแกรม MATLAB มาใช้แก้ปัญหาในหลายสาขา เช่นสาขาทางด้าน วิทยาศาสตร์ และวิศวกรรมศาสตร์ เป็นต้น ลักษณะการใช้งานเป็นแบบโต้ตอบกับผู้ใช้ (Interactive) และมีฟังก์ชันการใช้งานในสาขาต่างๆ ให้เลือกใช้มากมาย

ก.1 โครงสร้างของ MATLAB

โครงสร้างของโปรแกรม MATLAB ประกอบด้วย 5 ส่วนใหญ่ๆ ดังนี้คือ

1. ภาษาโปรแกรม MATLAB (MATLAB Language) เป็นภาษาโปรแกรมชั้นสูงที่ใช้ควบคุมลำดับการทำงาน ลักษณะการเขียนโปรแกรมเป็นแบบออบเจ็ก (Object Oriented Programing) ทำให้การเขียนโปรแกรมไม่ยุ่งยากเมื่อเทียบกับการเขียนโปรแกรมด้วยภาษาอื่นๆ เช่น C เป็นต้น

2. สิ่งแวดล้อมในการทำงานด้วย MATLAB (MATLAB Working Environment) โปรแกรม MATLAB จะมีกลุ่มของเครื่องมือที่เป็นประโยชน์สำหรับผู้เขียนโปรแกรม และผู้ใช้โปรแกรม เช่นการจัดการตัวแปร การนำข้อมูลเข้า การนำข้อมูลออก การเขียนโปรแกรม การตรวจสอบและจัดการข้อผิดพลาดของโปรแกรม เป็นต้น ในส่วนการติดต่อกับผู้ใช้โปรแกรม MATLAB มีให้เลือก 2 แบบคือแบบที่ให้ผู้ใช้ป้อนคำสั่งผ่านทางคีย์บอร์ดโดยตรง (Command Line) และแบบกราฟฟิก (Graphic User Interface)

3. ฟังก์ชันในการคำนวณทางคณิตศาสตร์ (MATLAB Mathematical Function Library) โปรแกรม MATLAB มีฟังก์ชันที่ใช้ในการคำนวณทางคณิตศาสตร์อย่างหลากหลาย เช่น ฟังก์ชันในการคำนวณพีชคณิต ฟังก์ชันในการหาค่าทางตรีโกณมิติ เป็นต้น ทำให้สามารถแก้ปัญหาทางด้านคณิตศาสตร์ที่ซับซ้อนได้

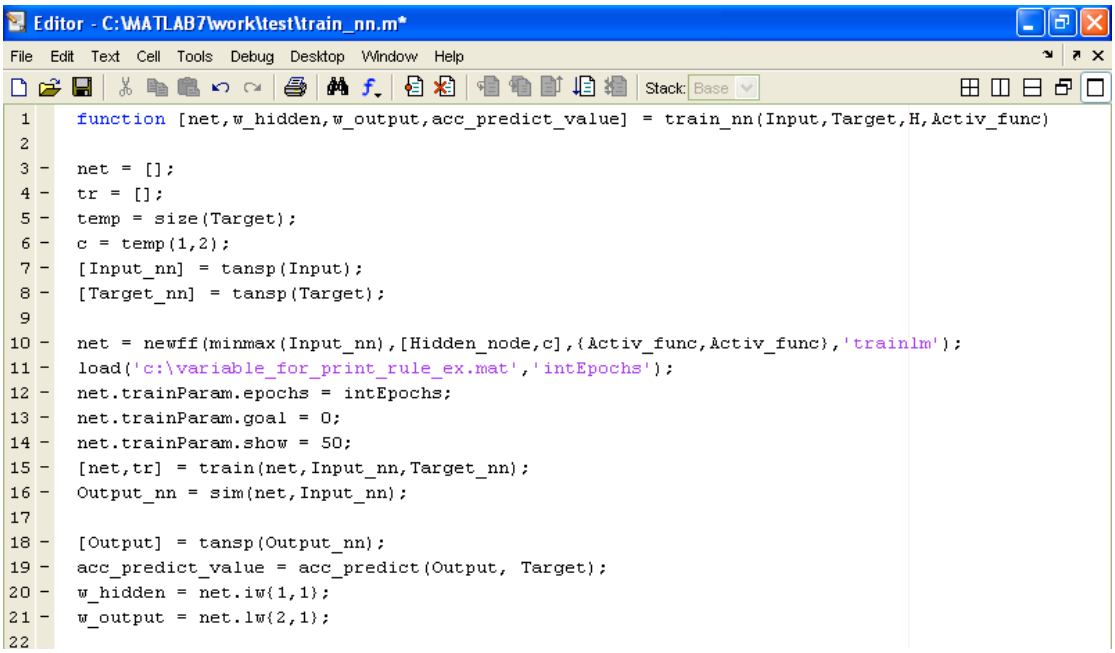
4. การจัดการกราฟฟิก (Handle Graphics) การจัดการกราฟิกของ MATLAB ประกอบด้วยคำสั่งชั้นสูงสำหรับการสร้างกราฟโดยมีพื้นฐานอยู่บนแนวความคิดที่ว่าทุกๆ สิ่งบนหน้าต่างรูปภาพของโปรแกรม MATLAB เป็นวัตถุ (Object) นอกจากนี้โปรแกรม MATLAB ยังมีฟังก์ชันที่ใช้สำหรับแสดงภาพสองมิติ ภาพสามมิติและภาพเคลื่อนไหวด้วย

5. ส่วนสนับสนุนการติดต่อจากภายนอก (MATLAB Application Program Interface : API) API จะใช้เพื่อสนับสนุนการติดต่อจากภายนอกโปรแกรม MATLAB เช่น การติดต่อจากโปรแกรม Visual Basic เป็นต้น

นอกจากนี้โปรแกรม MATLAB ยังมีการเพิ่มส่วนของการทำงานแบบพิเศษหรือกล่องเครื่องมือ (Toolboxes) ที่ช่วยในการหาคำตอบ โดยแต่ละกล่องเครื่องมือจะเหมาะกับงานในแต่ละสาขา เช่น กล่องเครื่องมือการประมวลสัญญาณ (Signal Processing Toolbox) กล่องเครื่องมือการประมวลผลภาพ (Image Processing Toolbox) และกล่องเครื่องมือโครงข่ายประสาทเทียม (Neural Networks Toolbox) เป็นต้น ซึ่งในงานวิจัยนี้ได้ใช้กล่องเครื่องมือ โครงข่ายประสาทเทียมมาช่วยในการสกัดกฎภาษาธรรมชาติด้วย

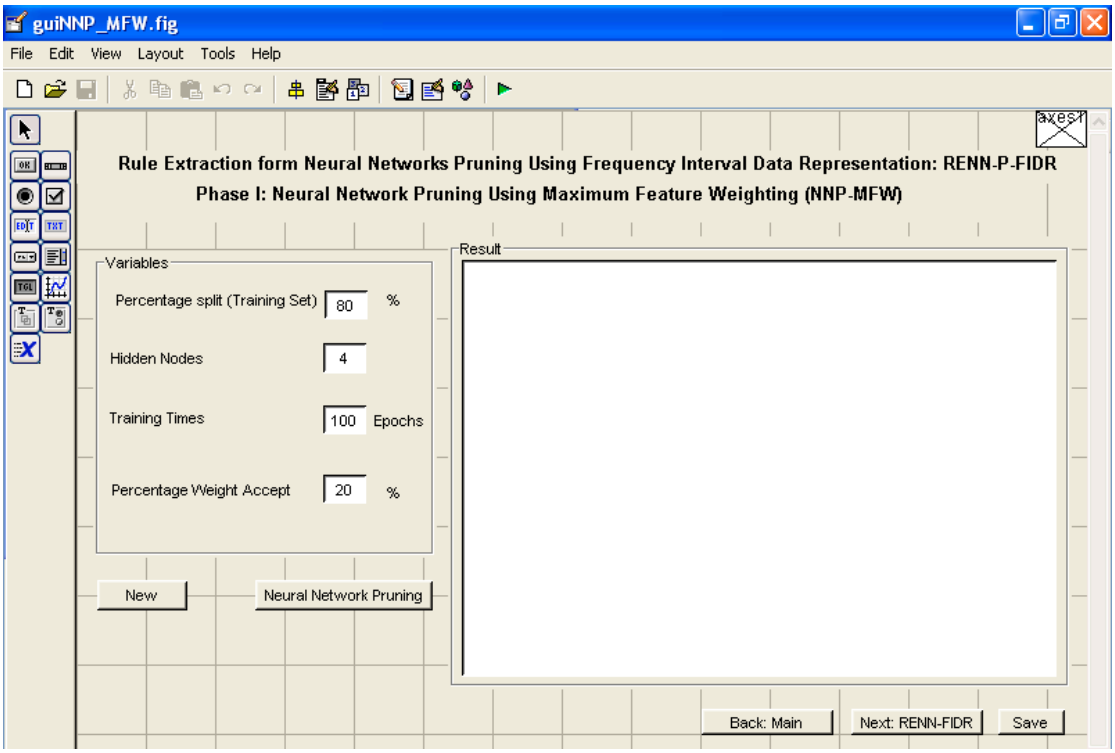
ก.2 การเขียนโปรแกรมด้วย MATLAB

ในการเขียนโปรแกรมด้วย MATLAB จะต้องบันทึกโปรแกรมเป็นนามสกุล .m ซึ่งเรียกว่า M-file ตัวอย่างเช่นโปรแกรม train_nn.m เป็นโปรแกรมสำหรับการสอนโครงข่ายประสาทเทียมแสดงดังภาพประกอบ ก.1 ซึ่งในโปรแกรมนี้มีการเรียกใช้ฟังก์ชันที่มีอยู่ใน MATLAB ตัวอย่างเช่นในบรรทัดที่ 5 มีการเรียกใช้ฟังก์ชัน size เพื่อหาขนาดของข้อมูลผลลัพธ์ มีการเรียกใช้ฟังก์ชันที่เขียนเพิ่มเติมขึ้นมาเอง ตัวอย่างเช่นในบรรทัดที่ 19 มีการเรียกใช้ฟังก์ชัน acc_predict เพื่อคำนวณหาค่าความถูกต้องของข้อมูล และมีการเรียกใช้ฟังก์ชันที่มีอยู่ในกล่องเครื่องมือโครงข่ายประสาทเทียม ตัวอย่างเช่นในบรรทัดที่ 15 มีการเรียกใช้ฟังก์ชัน train เพื่อสอนโครงข่ายประสาทเทียม



ภาพประกอบ ก.1 ตัวอย่าง M-file

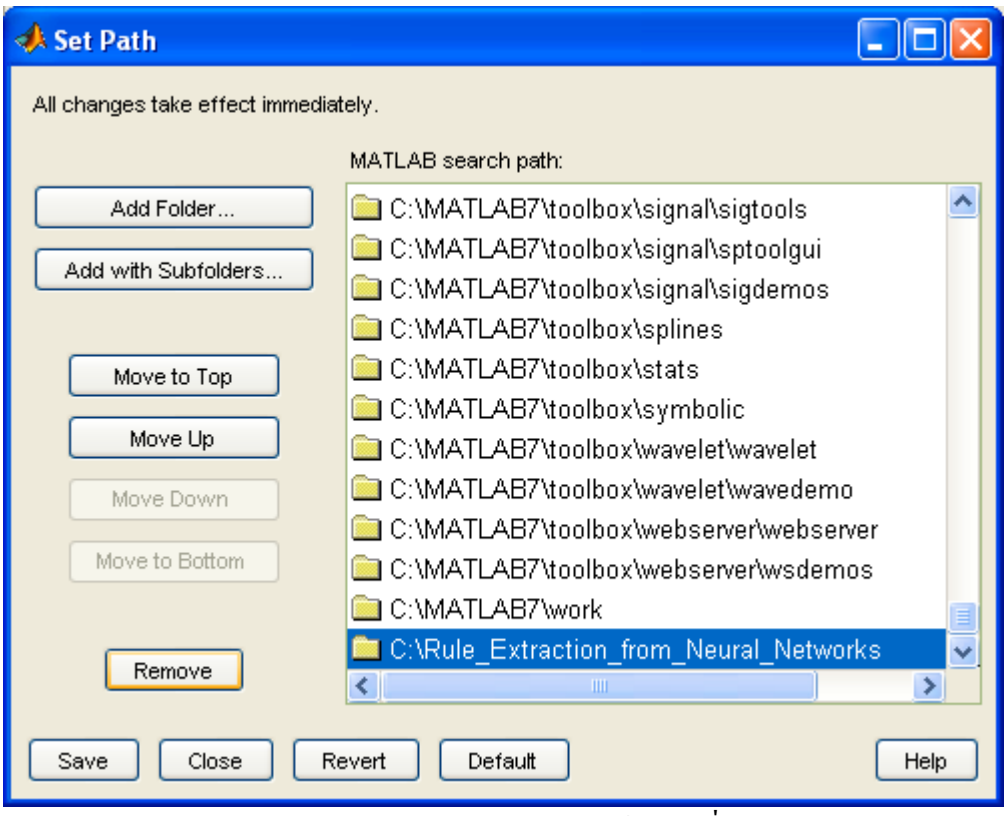
จากภาพประกอบ ก.1 เป็นตัวอย่าง M-file ที่ผู้ใช้จะต้องป้อนคำสั่งผ่านทางคีย์บอร์ดโดยตรง เพื่อให้ผู้ใช้สามารถใช้งานได้ง่าย และสะดวกยิ่งขึ้น จึงต้องเขียนโปรแกรมเชื่อมต่อกับผู้ใช้ทางกราฟฟิก (Graphic User Interface) โดยบันทึกโปรแกรมเป็นนามสกุล .fig ตัวอย่างเช่นโปรแกรม NNP_MFW. fig เป็นโปรแกรมในการตัดโหนดโครงข่ายประสาทเทียม ดังภาพประกอบ ก.2 ประกอบด้วยวัตถุ (Objects) ต่างๆ เช่น หน้าต่าง กล่องข้อความ ป้ายข้อความ และปุ่ม เป็นต้น โดยโปรแกรม MATLAB จะสร้างโปรแกรม M-file สำหรับการสร้างวัตถุต่างๆ ให้โดยอัตโนมัติ และสามารถเพิ่มคำสั่งในโปรแกรม M-file นั้นได้



ภาพประกอบ ก.2 ตัวอย่างโปรแกรมเชื่อมต่อกับผู้ใช้ทางกราฟฟิก

ก.3 การจัดการไฟล์และโฟลเดอร์

ในการประมวลผลของโปรแกรม MATLAB จะทำการค้นหาฟังก์ชันหรือโปรแกรมต่างๆ ตามเส้นทาง (Path) ที่ได้กำหนดไว้ ดังนั้นถ้าต้องการบันทึกโปรแกรมไว้ใน เส้นทางใหม่ที่ยังไม่ได้กำหนดไว้ จะต้องกำหนดเส้นทางใหม่นั้นเพิ่มด้วย โดยใช้เมนู File > Set Path ซึ่งในโปรแกรมการสกัดกฎภาษาธรรมชาติได้ทำการเพิ่มเส้นทาง C:\Rule_Extraction_from_Neural_Networks ดังภาพประกอบ ก.3



ภาพประกอบ ก.3 การกำหนดเส้นทางเพิ่ม

ก.4 การสร้างกราฟในโปรแกรม MATLAB

การสร้างกราฟมีความสำคัญมากสำหรับการวิเคราะห์ผลการทดลอง โปรแกรม MATLAB มีฟังก์ชันที่ใช้ในการสร้างกราฟในรูปแบบต่างๆ มากมายทั้งในลักษณะกราฟ 2 มิติ และ 3 มิติ เป็นต้น ซึ่งสามารถบันทึกไฟล์หน้าต่างที่แสดงกราฟเป็นนามสกุล .fig เพื่อนำมาใช้งานได้อีก ตัวอย่างโปรแกรมสร้างกราฟจำนวนกฎ เมื่อกำหนดให้ค่าความกำกวมที่ยอมรับได้ $\phi = 30$ $\phi = 50$ และ $\phi = 80$ ตามลำดับ แสดงดังภาพประกอบ ก.4 และตัวอย่างกราฟที่สร้างได้เมื่อประมวลผลโปรแกรมแสดงดังภาพประกอบ ก.5

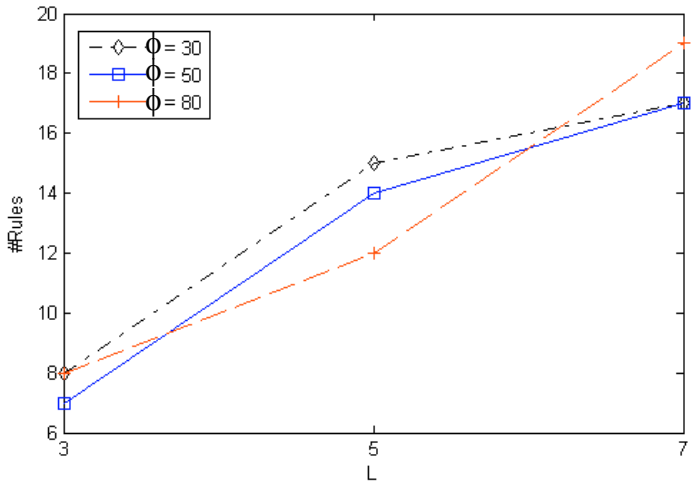
Editor - C:\MATLAB7\work\test\p_graph_RENN_P_FIDR_vs_Thr.m*

File Edit Text Cell Tools Debug Desktop Window Help

Stack: Base

```
1   function p_graph_RENN_P_FIDR_vs_Thr(Ambig_30,Ambig_50, Ambig_80)
2
3 -  [Ambig_30] = transp(Ambig_30);
4 -  [Ambig_50] = transp(Ambig_50);
5 -  [Ambig_80] = transp(Ambig_80);
6
7 -  set(gcf,'Color',[1,1,1]);
8
9 -  L = [3 5 7];
10 - plot(L, Ambig_30, 'k-.d',L, Ambig_50, 'b-s',L, Ambig_80, 'r--+');
11 - title(' ');
12 - xlabel('L'); ylabel('#Rule');
13 - legend('  = 30','  = 50','  = 80');
```

ภาพประกอบ ก.4 โปรแกรมสร้างกราฟจำนวนกฎ



ภาพประกอบ ก.5 ผลลัพธ์ของโปรแกรมสร้างกราฟจำนวนกฎ

ในการสร้างกราฟในโปรแกรม MATLAB จะมีการกำหนดรูปแบบเส้นแสดงดังตารางที่ ก.1 เครื่องหมายแสดงดังตารางที่ ก.2 และสีแสดงดังตารางที่ ก.3

ตารางที่ ก.1 รูปแบบเส้น

สัญลักษณ์	รูปแบบเส้น
.	เส้นจุด
-	เส้นทึบ
-.	เส้นประและเส้นจุด

- -	เส้นประ

ตารางที่ ก.2 เครื่องหมาย

สัญลักษณ์	เครื่องหมาย
.	จุด
o	วงกลม
+	บวก
x	กากบาท
*	ดอกจัน
s	สี่เหลี่ยม
d	รูปข้าวหลามตัด
v	สามเหลี่ยมล่าง
^	สามเหลี่ยมบน
>	สามเหลี่ยมขวา
<	สามเหลี่ยมซ้าย

ตารางที่ ก.3 สีเส้น

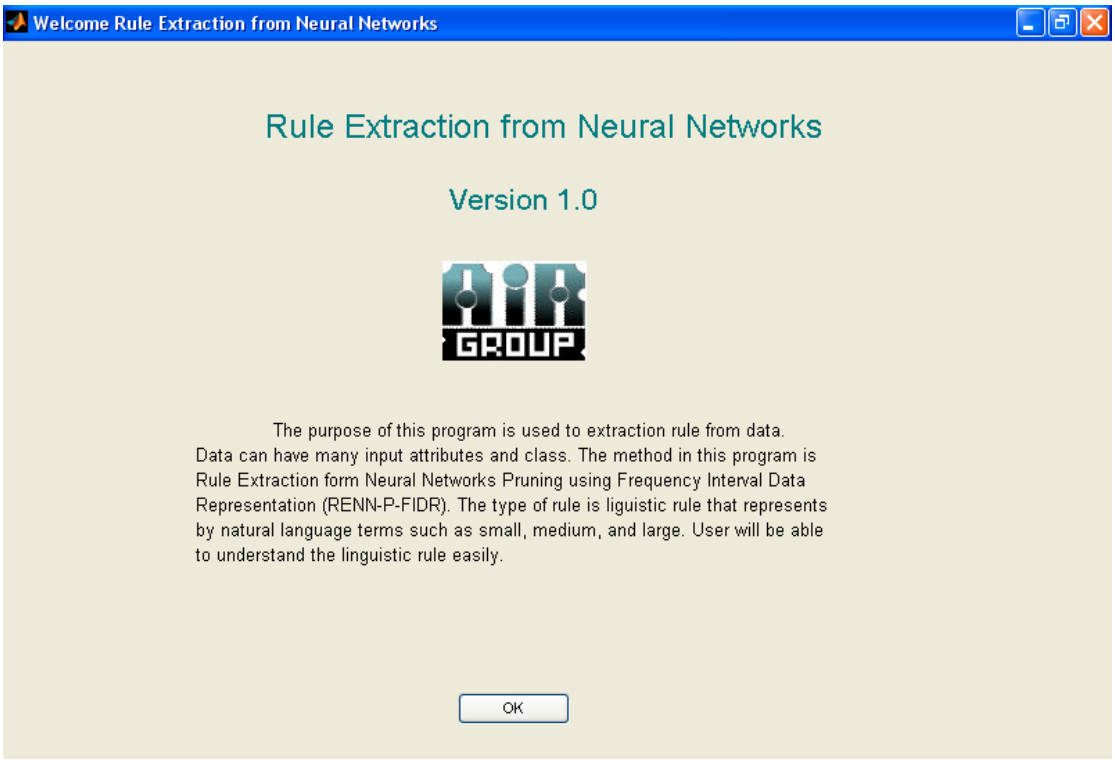
สัญลักษณ์	สี
r	red
g	green
b	blue
c	cyan
m	magenta
y	yellow
k	black
w	white

โดยสามารถดูรายละเอียดคำสั่งและฟังก์ชันอื่นๆได้ในคู่มือผู้ใช้ในโปรแกรม MATLAB 7.0

ภาคผนวก ข

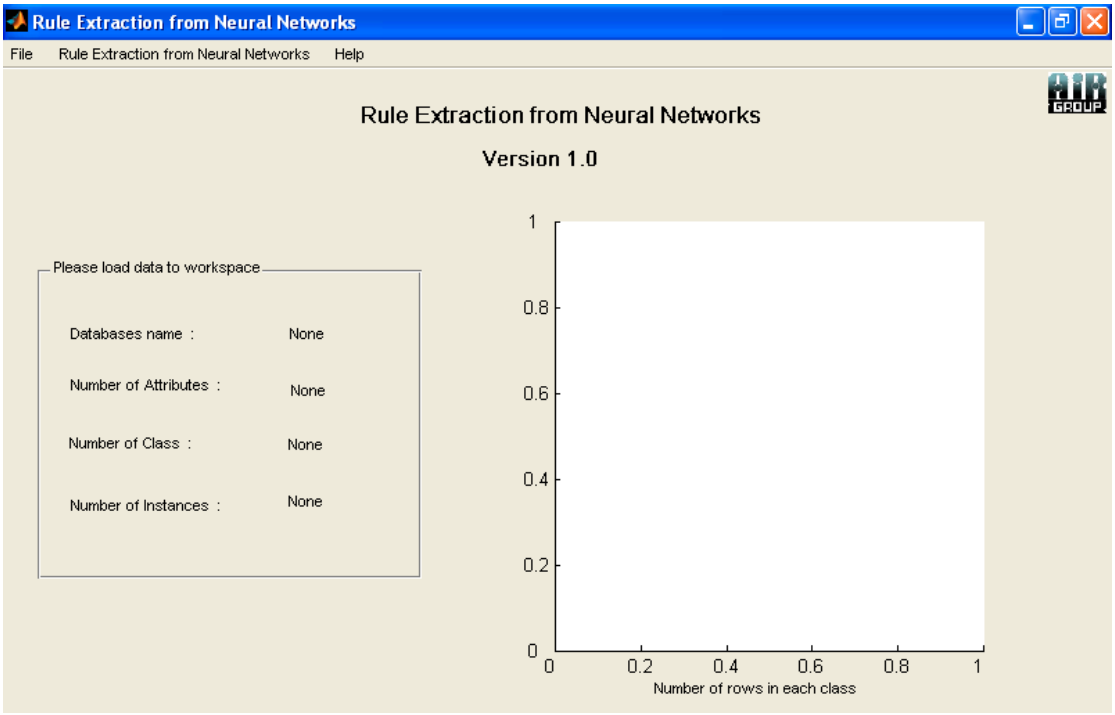
คู่มือสำหรับผู้ใช้

เมื่อเปิดโปรแกรมการสกัดกฎภาษาธรรมชาติโดยใช้โครงข่ายประสาทเทียมจะปรากฏหน้าจอที่อธิบายข้อมูลเบื้องต้นเกี่ยวกับโปรแกรม ดังภาพประกอบ ข.1



ภาพประกอบ ข.1 โปรแกรมการสกัดกฎภาษาธรรมชาติโดยใช้โครงข่ายประสาทเทียม

เริ่มต้นสกัดกฎภาษาธรรมชาติโดยใช้โครงข่ายประสาทเทียมได้โดยคลิกปุ่ม OK ซึ่งจะปรากฏหน้าจอสำหรับโหลดข้อมูลเข้าไปยังโปรแกรมดังภาพประกอบ ข.2



ภาพประกอบ ข.2 หน้าจอสำหรับโหลดข้อมูลเข้าไปยังโปรแกรม

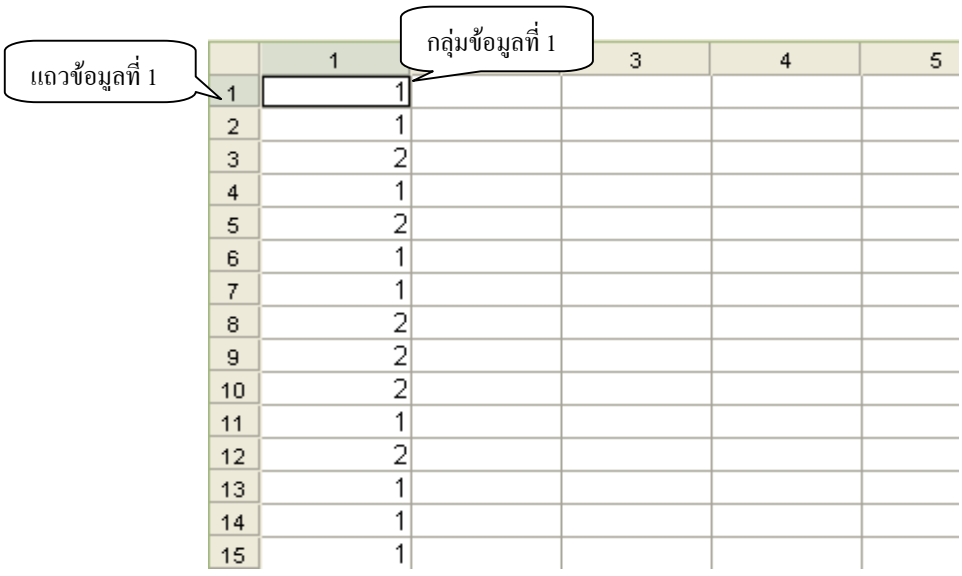
จากนั้นโหลดข้อมูลจากไฟล์ข้อมูลเข้ามาใน workspace ของโปรแกรม MATLAB ซึ่งในไฟล์ข้อมูลจะต้องเก็บข้อมูลไว้ในตัวแปร 2 ตัวแปรคือ Input_total ไว้เก็บค่าตัวแปรข้อมูลเข้า และ Target_total ไว้เก็บค่าผลลัพธ์หรือกลุ่มข้อมูล โดยทั้ง 2 ตัวแปรมีรูปแบบในการเก็บดังนี้

- ตัวแปร Input_total ในตำแหน่งของแถวใช้แทนลำดับของแถวข้อมูล และตำแหน่งของหลักใช้แทนลำดับของตัวแปรข้อมูลเข้า ดังภาพประกอบ ข.3

	1	2	3	4	5
1	8	10	10	8	
2	5	3	3	3	
3	8	7	5	10	
4	7	4	6	4	
5	10	7	7	6	
6	7	3	2	10	
7	10	5	5	3	
8	8	4	5	1	
9	5	2	3	4	
10	10	7	7	3	
11	10	10	10	8	
12	5	4	4	9	
13	2	5	3	3	
14	10	4	3	1	
15	6	10	10	2	

ภาพประกอบ ข.3 การเก็บค่าตัวแปรข้อมูลเข้า

- ตัวแปร Target_total ในตำแหน่งของแถวใช้แทนลำดับของแถวข้อมูล และใช้หลักที่ 1 เพียงหลักเดียวเท่านั้นในการแทนค่าผลลัพธ์ ซึ่งถ้าค่าผลลัพธ์เป็น 1 หมายถึงเป็นข้อมูลกลุ่มที่ 1 ถ้าค่าผลลัพธ์เป็น 2 หมายถึงเป็นข้อมูลกลุ่มที่ 2 เป็นต้น ดังภาพประกอบ ข.4

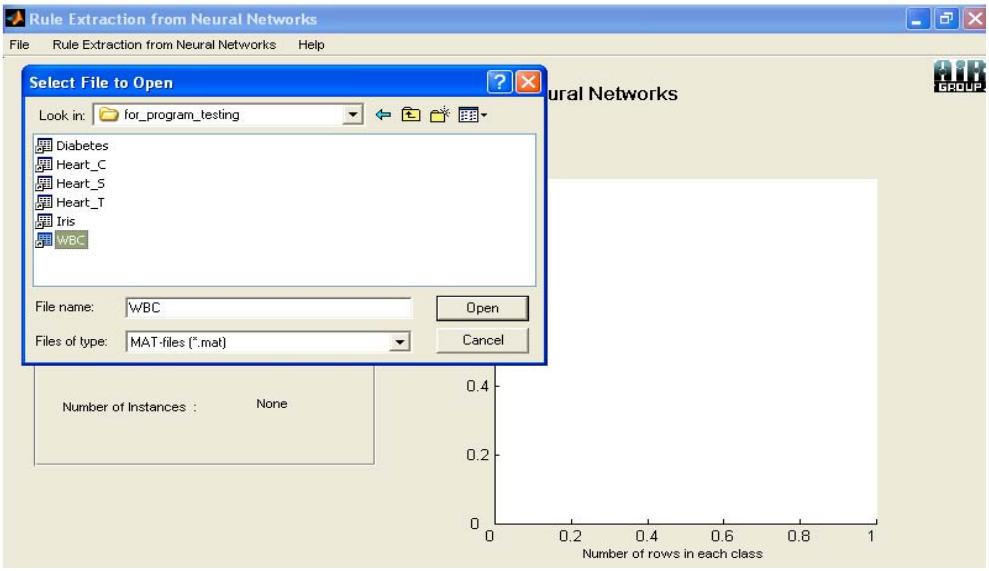


	1	3	4	5
1	1			
2	1			
3	2			
4	1			
5	2			
6	1			
7	1			
8	2			
9	2			
10	2			
11	1			
12	2			
13	1			
14	1			
15	1			

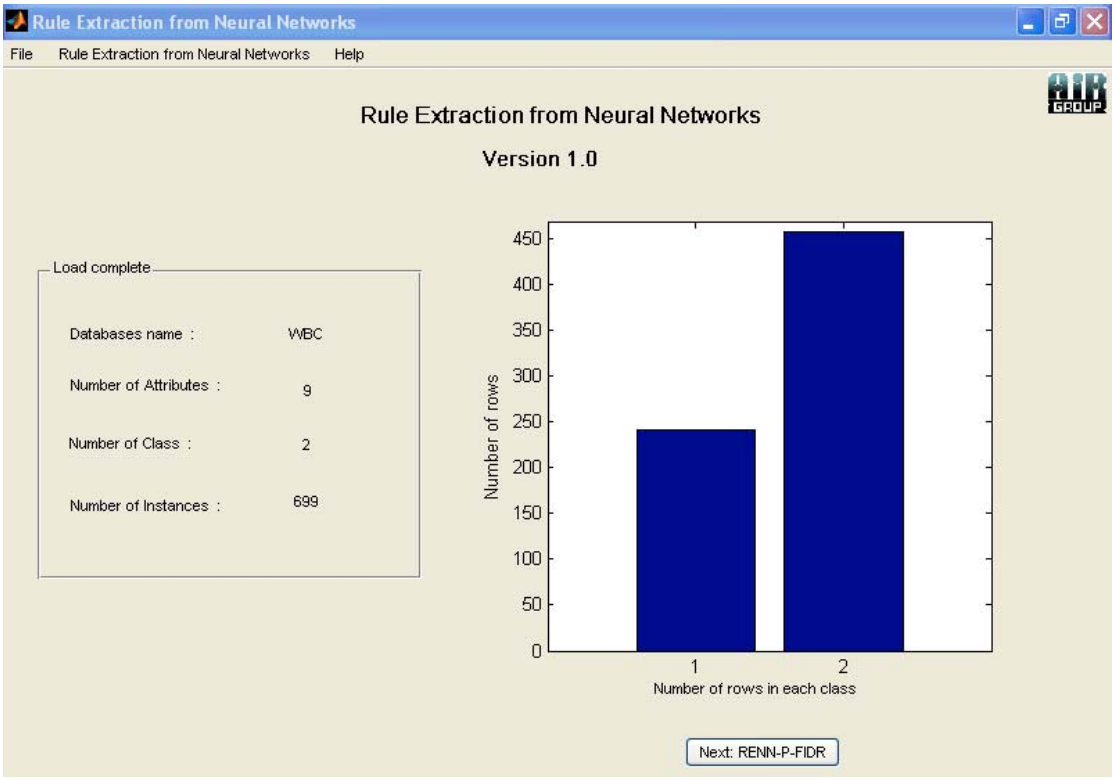
ภาพประกอบ ข.4 การเก็บค่าผลลัพธ์หรือกลุ่มข้อมูล

ในการโหลดไฟล์ข้อมูลให้คลิกเลือกเมนู File > Load database ดังภาพประกอบ ข.5 เมื่อโหลดไฟล์ข้อมูลเรียบร้อยแล้วโปรแกรมจะแสดงรายละเอียดของข้อมูลประกอบด้วย ชื่อฐานข้อมูล จำนวนตัวแปรข้อมูลเข้า จำนวนกลุ่มข้อมูล และจำนวนแถวข้อมูล นอกจากนี้โปรแกรม

ยังแสดงกราฟความถี่จำนวนแถวข้อมูลในแต่ละกลุ่มข้อมูล ตัวอย่างจากฐานข้อมูลโรคมะเร็ง เต้านมดังภาพประกอบ ข.6

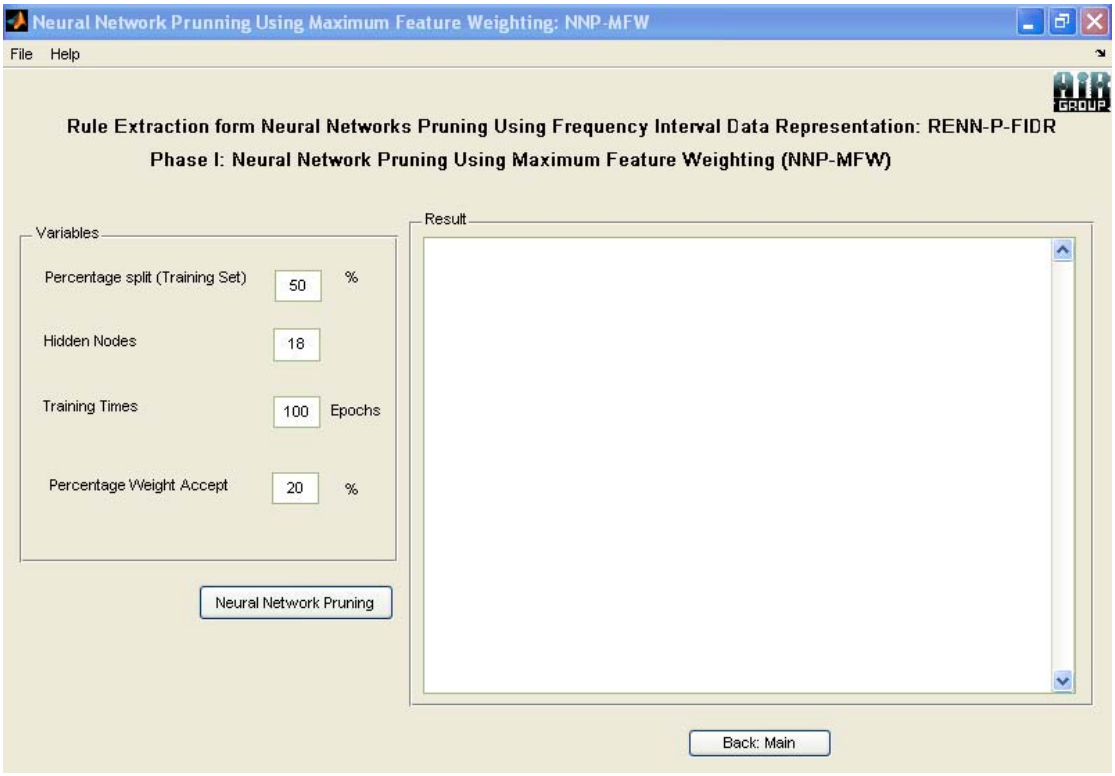


ภาพประกอบ ข.5 การโหลดไฟล์ข้อมูล



ภาพประกอบ ข.6 การแสดงรายละเอียดของข้อมูล

คลิกเลือกเมนู Rule Extraction > Rule Extraction form Neural Networks Pruning using Frequency Interval Data Representation: RENN-P-FIDR หรือคลิกปุ่ม Next: RENN-P-FIDR เพื่อสกัดกฎภาษาธรรมชาติโดยใช้โครงข่ายประสาทเทียมที่ผ่านการตัดโหนดโดยใช้การแทนค่าความถี่ของช่วงข้อมูล ซึ่งจะเข้าสู่หน้าจอขั้นตอนหลักที่ I การตัดโหนดโครงข่ายประสาทเทียมโดยใช้ค่าน้ำหนักสูงสุด (NNP-MFW) แสดงดังภาพประกอบ ข.7



ภาพประกอบ ข.7 หน้าจอขั้นตอนหลักที่ I การตัดโหนดโครงข่ายประสาทเทียม
โดยใช้ค่าน้ำหนักสูงสุด

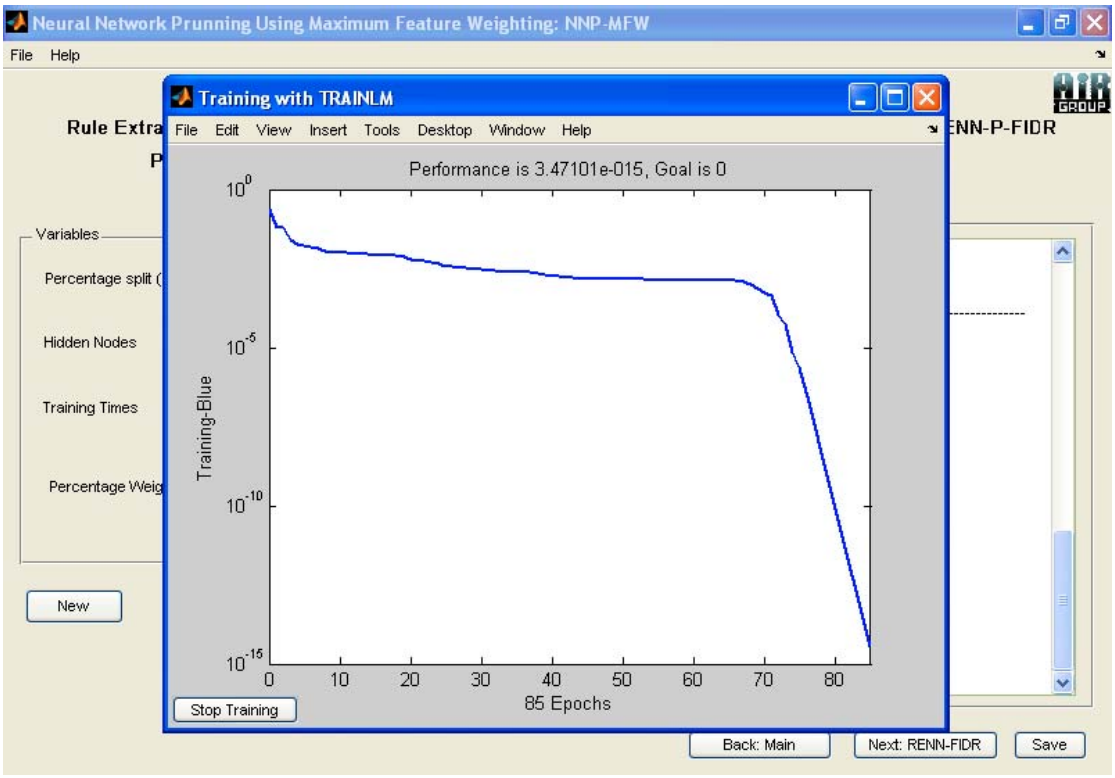
ในการตัดโหนดโครงข่ายประสาทเทียมโดยใช้ค่าน้ำหนักสูงสุดจะต้องกำหนดตัวแปรต่างๆ ดังตารางที่ ข.1 ดังนี้

ตารางที่ ข.1 ตัวแปรในขั้นตอนหลักที่ I (NNP-MFW)

ชื่อตัวแปร	คำอธิบาย	ค่าที่เป็นไปได้ (เลขจำนวนเต็ม)	ค่าที่แนะนำ
1. Percentage Split	ร้อยละที่ใช้ในการสุ่มข้อมูลชุดสอน (%)	1-99	80

2. Hidden Nodes	จำนวนโหนดซ่อนในโครงข่ายประสาท-เทียม (nodes)	1-99	1 ถึง 2 เท่าของจำนวนตัวแปรข้อมูลเข้า
3. Training Times	จำนวนรอบในการสอนโครงข่ายประสาท-เทียม (epochs)	1-10000	100
4. Percentage Weight Accept	จำนวนร้อยละของน้ำหนักที่ยอมรับได้ ถ้ามีค่ามากจะตัดโหนดโครงข่ายประสาท-เทียมได้มาก ถ้ามีค่าน้อยจะตัดโหนดโครงข่ายประสาทเทียมได้น้อย (%)	0-100	20

ตัวอย่าง กำหนดให้ร้อยละของข้อมูลชุดสอน = 80 ร้อยละของน้ำหนักที่ยอมรับได้ = 20 จำนวนโหนดซ่อน = 9 โหนด และจำนวนรอบในการเรียนรู้ = 200 รอบ คลิกปุ่ม Neural Network Pruning เพื่อตัดโหนดโครงข่ายประสาทเทียม โปรแกรมจะทำการสอนโครงข่ายประสาทเทียมดังภาพประกอบ ข.8



ภาพประกอบ ข.8 การสอนโครงข่ายประสาทเทียมในการตัดโหนดโครงข่ายประสาทเทียม

หน้าจอผลลัพธ์ที่ได้แสดงดังภาพประกอบ ข.9 ซึ่งมีปุ่มให้เลือกคลิก 5 ปุ่มโดยแต่ละปุ่มมีหน้าที่ดังนี้

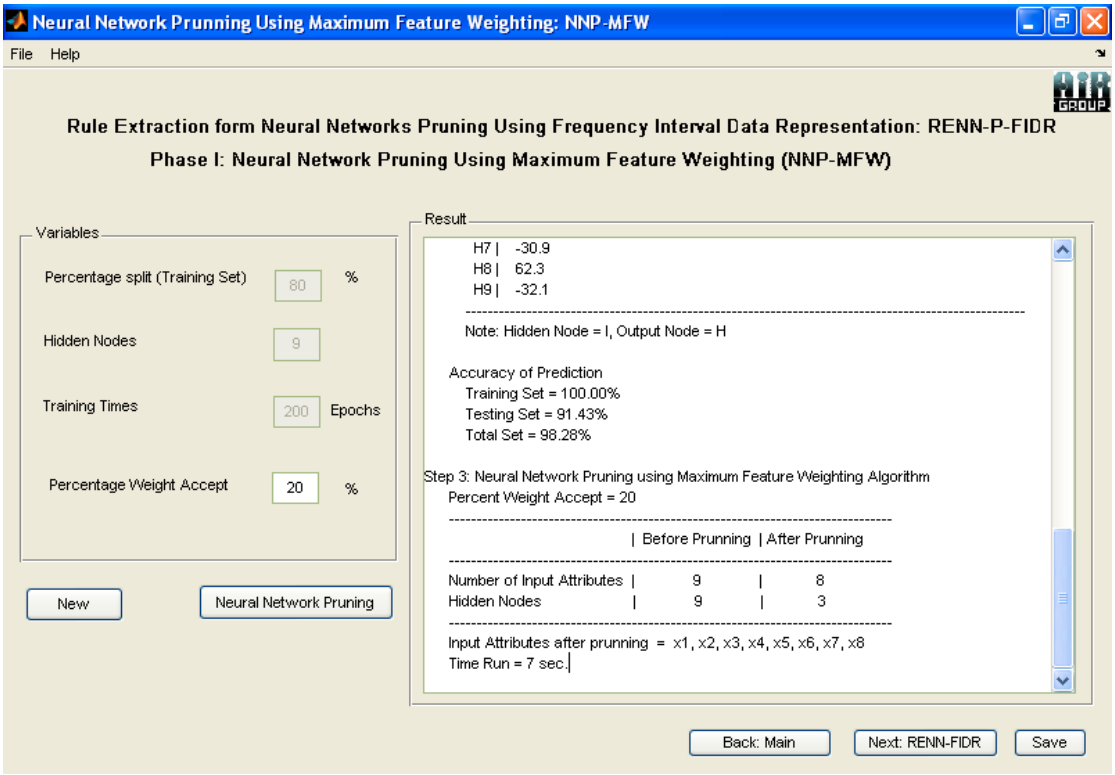
- New ตัดโหนดโครงข่ายประสาทเทียมโดยสอนโครงข่ายประสาทเทียมใหม่จะต้องคลิกปุ่ม New
- Neural Network Pruning ตัดโหนดโครงข่ายประสาทเทียมโดยไม่ต้องสอนโครงข่ายประสาทเทียมใหม่ แต่เปลี่ยนค่าร้อยละของน้ำหนักที่ยอมรับได้
- Back: Main กลับไปยังหน้าจอโหลดข้อมูล เพื่อโหลดไฟล์ข้อมูลใหม่
- Next: RENN-FIDR ไปยังหน้าจอขั้นตอนหลักที่ II การสกัดกฎภาษาธรรมชาติจากโครงข่ายประสาทเทียมโดยการแทนค่าความถี่ของช่วงข้อมูล (RENN-FIDR)
- Save บันทึกผลลัพธ์ที่ได้จากการตัดโหนดดังภาพประกอบ ข.910

ผลลัพธ์ที่ได้ในแต่ละขั้นตอนแสดงดังภาพประกอบ ข.11 ซึ่งมีรายละเอียดดังต่อไปนี้

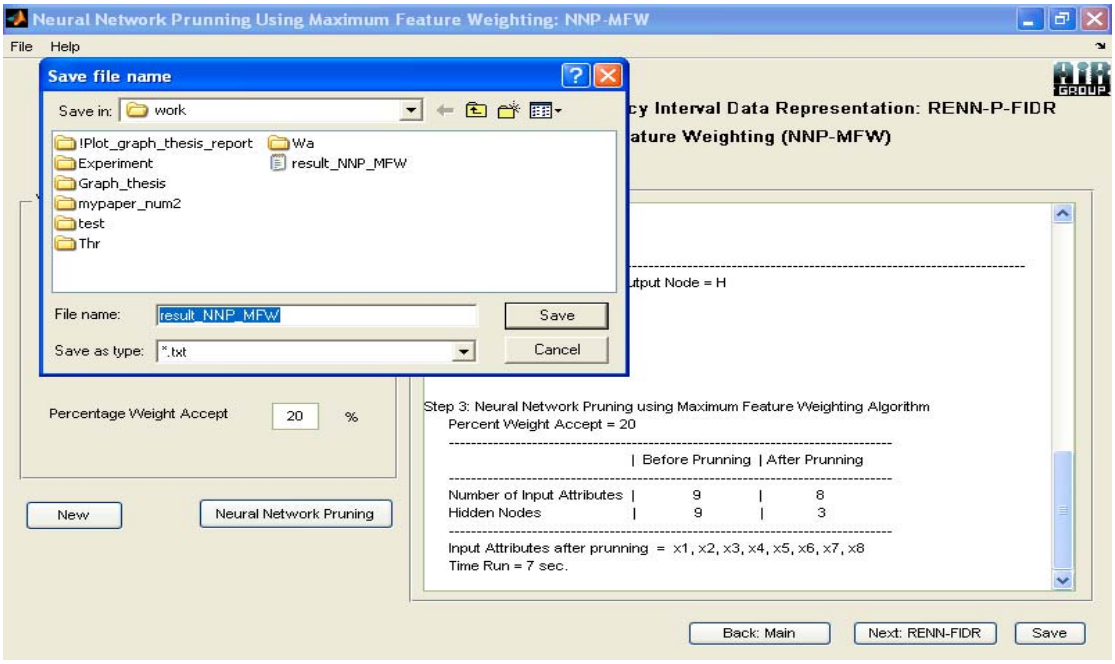
ขั้นตอนที่ 1 การเตรียมข้อมูล ประกอบด้วยลักษณะของข้อมูล การสุ่มข้อมูลชุดสอนและชุดทดสอบ

ขั้นตอนที่ 2 การสอนโครงข่ายประสาทเทียมด้วยการเรียนรู้แบบแพร่ย้อนกลับประกอบด้วย ตารางค่าน้ำหนักของโครงข่ายประสาทเทียมที่ได้จากการสอนโครงข่ายประสาทเทียม โดยตารางที่ 1 เป็นตารางแสดงค่าน้ำหนักจากโหนดข้อมูลเข้าไปยังโหนดซ่อน และตารางที่ 2 เป็นตารางแสดงค่าน้ำหนักจากโหนดซ่อนไปยังโหนดผลลัพธ์ นอกจากนี้ยังแสดง ค่าความถูกต้องของข้อมูลจากโครงข่ายประสาทเทียมด้วย

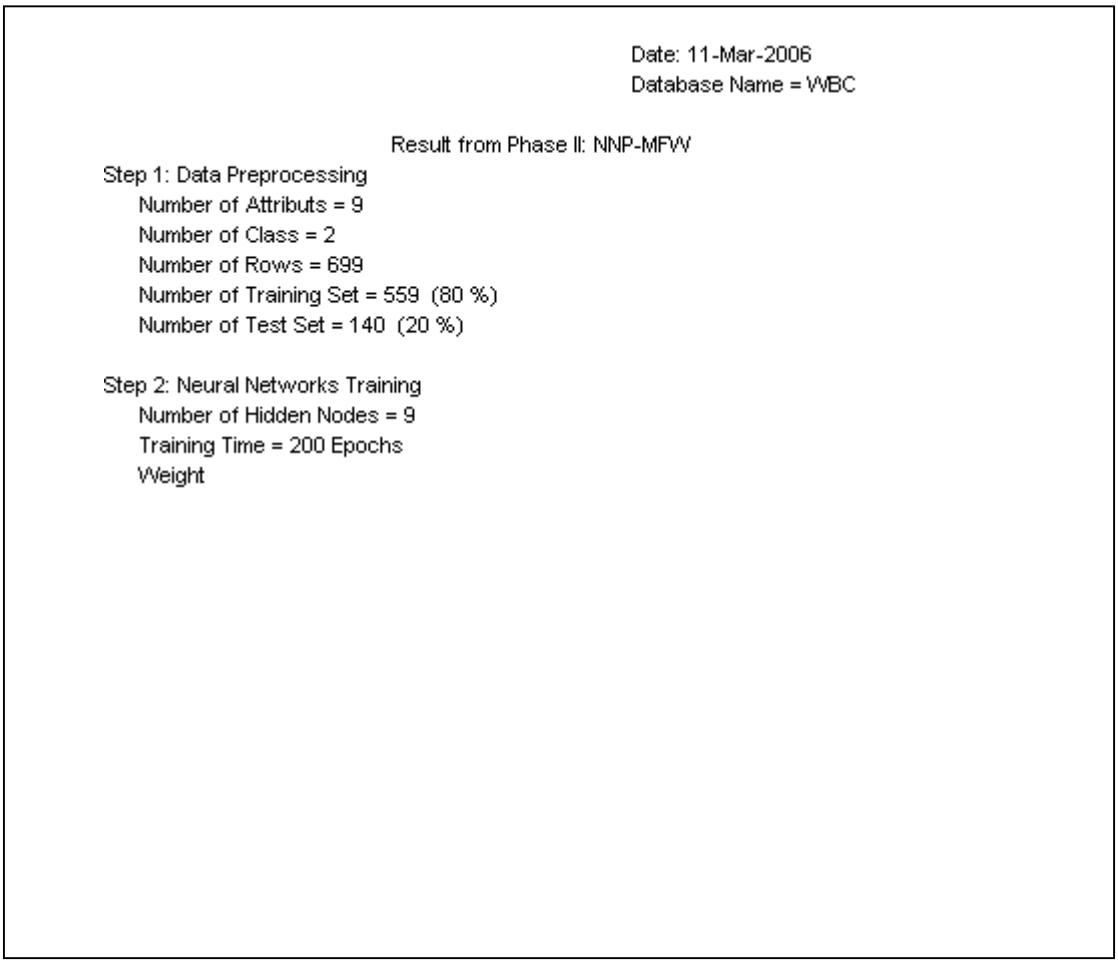
ขั้นตอนที่ 3 ขั้นตอนวิธีการตัดโหนดโครงข่ายประสาทเทียมโดยใช้ค่าน้ำหนักสูงสุด ประกอบด้วยตัวแปรข้อมูลเข้าและโหนดข้อมูลเข้าหลังผ่านการตัดโหนด ซึ่งจะถูกนำมาใช้ในขั้นตอนหลักที่ II การสกัดกฎภาษาธรรมชาติจากโครงข่ายประสาทเทียมโดยการแทนค่าความถี่ของช่วงข้อมูล (RENN-FIDR) ต่อไป



ภาพประกอบ ข.9 หน้าจอผลลัพธ์ของขั้นตอนหลักที่ I (NNP-MFW)



ภาพประกอบ ข.10 การบันทึกผลลัพธ์ลงในไฟล์



Input Layer to Hidden Layer

I/H	H1	H2	H3	H4	H5	H6	H7	H8	H9
I1	-0.1	-0.5	-0.4	1.5	-5.6	-4.5	2.1	2.4	1.0
I2	-3.2	-2.0	2.8	-2.5	8.4	-0.2	1.9	-0.3	-1.3
I3	1.3	0.4	-0.6	-0.1	4.3	-2.3	0.2	1.8	-1.3
I4	1.5	-1.7	-2.0	-2.0	4.3	-3.3	-0.1	0.9	-2.1
I5	-0.1	-3.9	-0.5	-3.1	-5.7	-2.0	-4.2	4.0	-0.8
I6	2.7	3.3	-2.5	2.9	-0.3	4.9	2.0	3.9	4.3
I7	-2.1	4.8	-0.8	2.7	1.1	-1.8	-4.4	-6.0	-0.8
I8	0.9	-1.1	-0.8	2.2	-1.0	0.8	-4.6	-2.7	1.2
I9	-4.3	5.3	3.6	0.5	-4.8	5.7	2.5	0.4	3.1

Note: Input Node = I, Hidden Node = H

ภาพประกอบ ข.11 ผลลัพธ์ของขั้นตอนหลักที่ I (NNP-MFW) ในแต่ละขั้นตอน

Hidden Layer to Output Layer

H/O	O1
H1	-34.0
H2	3.5
H3	-2.7
H4	61.7
H5	34.2
H6	-32.7
H7	-30.9
H8	62.3
H9	-32.1

Note: Hidden Node = I, Output Node = H

Accuracy of Prediction
Training Set = 100.00%
Testing Set = 91.43%
Total Set = 98.28%

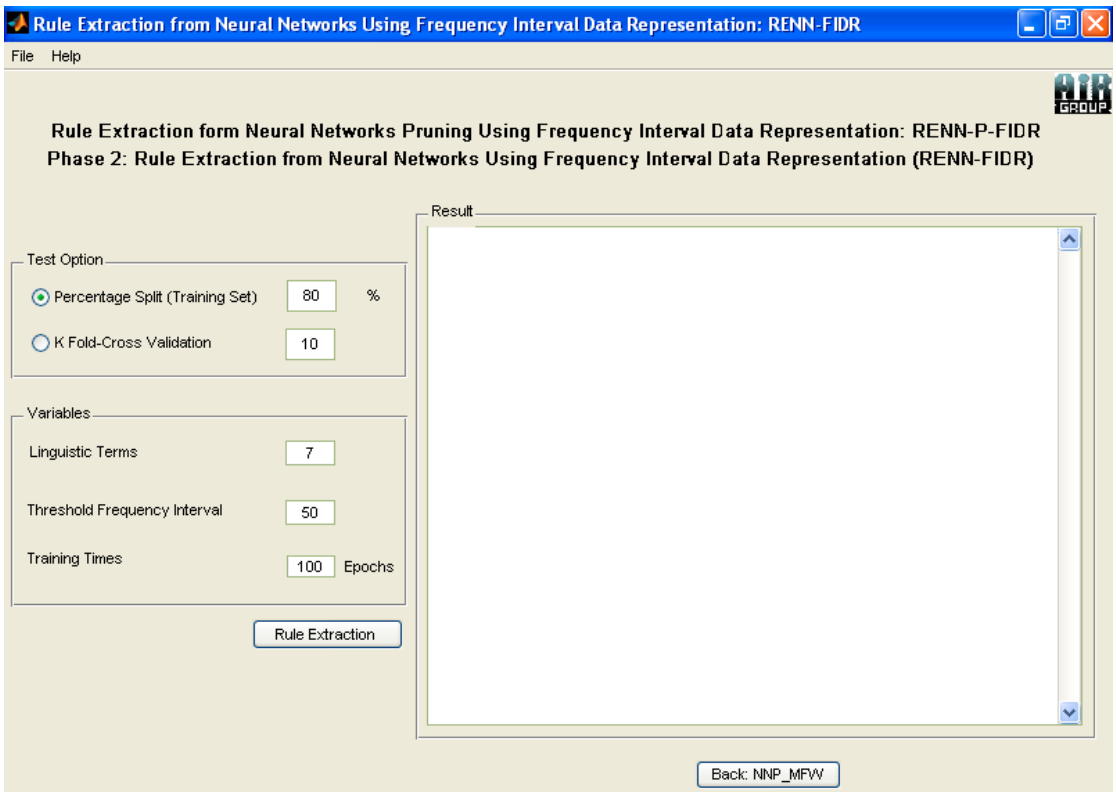
Step 3: Neural Network Pruning using Maximum Feature Weighting Algorithm
Percent Weight Accept = 20

	Before Prunning	After Prunning
Number of Input Attributes	9	8
Hidden Nodes	9	3

Input Attributes after prunning = x1, x2, x3, x4, x5, x6, x7, x8
Time Run = 7 sec.

ภาพประกอบ ข.11 (ต่อ)

เมื่อคลิกปุ่ม Next: RENN-FIDR จะปรากฏหน้าจอขั้นตอนหลักที่ II การสกัดกฎภาษาธรรมชาติจากโครงข่ายประสาทเทียมโดยการแทนค่าความถี่ของช่วงข้อมูล (RENN-FIDR) ดังภาพประกอบ ข.12



ภาพประกอบ ข.12 หน้าจอขั้นตอนหลักที่ II การสกัดกฎภาษาธรรมชาติจากโครงข่ายประสาทเทียมโดยการแทนค่าความถี่ของช่วงข้อมูล

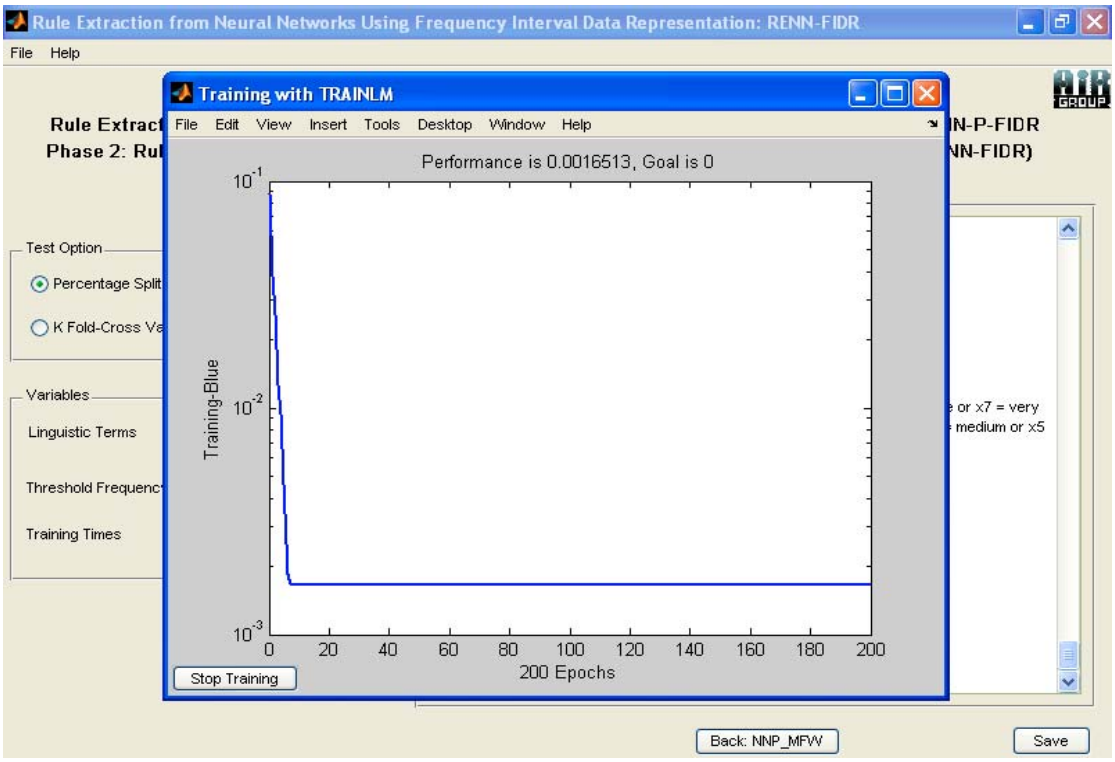
ในการสกัดกฎภาษาธรรมชาติจากโครงข่ายประสาทเทียมโดยการแทนค่าความถี่ของช่วงข้อมูลจะต้องกำหนดตัวแปรต่างๆ ดังตารางที่ ข.2

ตารางที่ ข.1 ตัวแปรในขั้นตอนหลักที่ II (RENN-FIDR)

ชื่อตัวแปร	คำอธิบาย	ค่าที่เป็นไปได้ (เลขจำนวนเต็ม)	ค่าที่แนะนำ
1. Test Option 1.1 Percentage Split	1. การเลือกสุ่มข้อมูล (เลือกแบบเดียว)	1.1 = 1-99 1.2 = 1-10	1.1 = 80 1.2 = 10

1.2 K-Fold Cross Validation	1.1 แบบร้อยละใช้ในการสุ่มข้อมูลชุดสอน (%) 1.2 แบบความเที่ยงตรง K กลุ่ม		
2. Linguistic Terms	จำนวนรูปแบบภาษาธรรมชาติ	2-20	7
3. Threshold Frequency Interval	ค่าความกำกวมที่ยอมรับได้	0-100	50
4. Training Times	จำนวนรอบในการสอนโครงข่ายประสาทเทียม (epochs)	1-10000	100

ตัวอย่าง เลือกสุ่มข้อมูลแบบร้อยละ โดยกำหนดให้ร้อยละของข้อมูลชุดสอน = 80 เปอร์เซ็น จำนวนรูปแบบภาษาธรรมชาติ = 5 และจำนวนรอบในการเรียนรู้ = 200 รอบ คลิกปุ่ม Rule Extraction เพื่อสกัดกฎภาษาธรรมชาติ โปรแกรมจะทำการสอนโครงข่ายประสาทเทียมดังภาพประกอบ ข.13



ภาพประกอบ ข.13 การสอนโครงข่ายประสาทเทียมในการสกัดกฎภาษาธรรมชาติ

หน้าจอผลลัพธ์ที่ได้แสดงดังภาพประกอบ ข.14 ซึ่งมีปุ่มให้เลือกคลิก 3 ปุ่มโดยแต่ละปุ่มมีหน้าที่ดังนี้

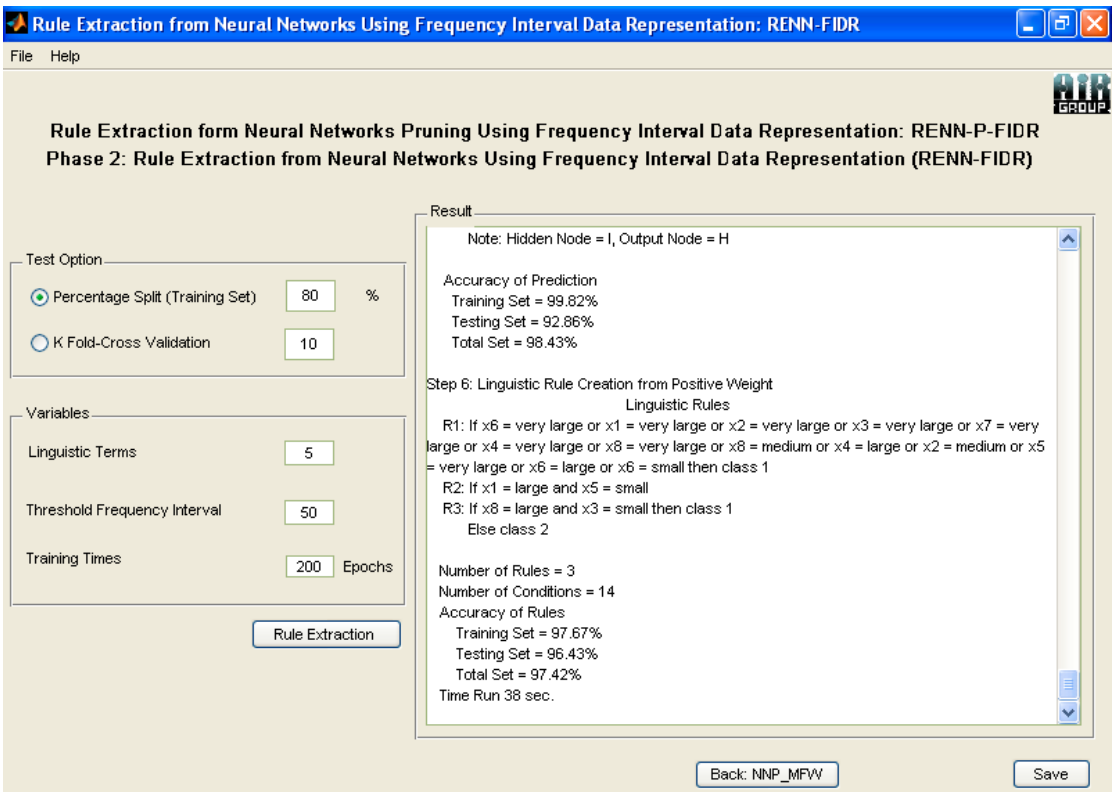
- Rule Extraction เพื่อสกัดกฎภาษาธรรมชาติจากโครงข่ายประสาทเทียมโดยการแทนค่าความถี่ของช่วงข้อมูล
- Back: NNP-MFW ไปยังหน้าจอขั้นตอนหลักที่ I การตัดโหนดโครงข่ายประสาทเทียมโดยใช้ค่าน้ำหนักสูงสุด (NNP-MFW)
- Save บันทึกผลลัพธ์ที่ได้จากการสกัดกฎภาษาธรรมชาติ

ผลลัพธ์ที่ได้ในแต่ละขั้นตอนแสดงดังภาพประกอบ ข.15 ซึ่งมีรายละเอียดดังต่อไปนี้

ขั้นตอนที่ 4 ขั้นตอนวิธีการสร้างรูปแบบภาษาธรรมชาติจากความถี่ของช่วงข้อมูลประกอบด้วยช่วงภาษาธรรมชาติที่แบ่งได้

ขั้นตอนที่ 5 การสอนโครงข่ายประสาทเทียมด้วยรูปแบบภาษาธรรมชาติประกอบด้วยตารางค่าน้ำหนักของโครงข่ายประสาทเทียมที่ได้จากการสอนโครงข่ายประสาทเทียมโดยตารางที่ 1 เป็นตารางแสดงค่าน้ำหนักจากโหนดข้อมูลเข้าไปยังโหนดซ่อน และตารางที่ 2 เป็นตารางแสดงค่าน้ำหนักจากโหนดซ่อนไปยังโหนดผลลัพธ์ นอกจากนี้ยังแสดงค่าความถูกต้องของข้อมูลจากโครงข่ายประสาทเทียมด้วย

ขั้นตอนที่ 6 ขั้นตอนวิธีการสร้างกฎภาษาธรรมชาติโดยใช้ค่าน้ำหนักที่เป็นบวกประกอบด้วยกฎที่สร้างได้ และค่าความถูกต้องของข้อมูลของกฎ



ภาพประกอบ ข.14 หน้าจอผลลัพธ์ในขั้นตอนหลักที่ II (RENN-FIDR)

Date: 11-Mar-2006
Database Name = WBC
Percent Weight Accept = 20

Result from Phase II: RENN-FIDR

Test Option = Percentage
Number of Training Set = 559 (80 %)
Number of Testing Set = 140 (20 %)
Number of Total Set = 699 (100 %)

Step 4: Linguistic Intervals Creation from Frequency Algorithm
Linguistic Terms = 5
Threshold Frequency Interval = 50

Linguistic Intervals

x / ling. terms	very small,	small,	medium,	large,	very large
x1	[1.0,5.0),	[5.0,6.0),	[6.0,7.0),	[7.0,9.0),	[9.0,10.0]
x2	[1.0,4.0),	[4.0,5.0),	[5.0,6.0),	[6.0,10.0),	[10.0,10.0]
x3	[1.0,3.0),	[3.0,4.0),	[4.0,6.0),	[6.0,9.0),	[9.0,10.0]
x4	[1.0,3.0),	[3.0,4.0),	[4.0,7.0),	[7.0,10.0),	[10.0,10.0]
x5	[1.0,3.0),	[3.0,4.0),	[4.0,6.0),	[6.0,10.0),	[10.0,10.0]
x6	[1.0,6.0),	[6.0,7.0),	[7.0,9.0),	[9.0,10.0),	[10.0,10.0]
x7	[1.0,4.0),	[4.0,5.0),	[5.0,6.0),	[6.0,8.0),	[8.0,10.0]
x8	[1.0,3.0),	[3.0,4.0),	[4.0,6.0),	[6.0,10.0),	[10.0,10.0]

Step 5: Neural Networks Training with Linguistic Terms
Number of Hidden Nodes = 3
Training Times = 200 Epochs
Weight

Input Layer to Hidden Layer

I/H	H1	H2	H3
I1_1	-2.4	5.9	4.1
I1_2	-4.1	0.8	-0.0
I1_3	3.4	1.3	-4.6
I1_4	1.2	-2.1	2.6
I1_5	2.7	-9.1	-1.2
I2_1	0.8	-5.9	-2.6
I2_2	2.1	6.9	1.2
I2_3	0.5	-3.9	-0.9
I2_4	0.1	-0.3	-1.3
I2_5	-0.5	-1.9	2.3
I3_1	-2.2	0.7	-2.7
I3_2	1.6	0.5	0.9
I3_3	5.1	-3.0	-4.6
I3_4	-0.8	-0.0	4.8
I3_5	-0.6	-2.5	2.2
I4_1	-1.3	-1.5	-5.7
I4_2	-1.5	2.4	-0.0
I4_3	2.6	0.8	-0.3
I4_4	0.1	-1.0	3.0
I4_5	0.1	-3.3	5.5

ภาพประกอบ ข.15 ผลลัพธ์ในขั้นตอนหลักที่ II (RENN-FIDR) ในแต่ละขั้นตอน

```
I5_1 |   -2.9   4.1   4.5
I5_2 |   1.0   0.4   -4.5
I5_3 |   -2.4   3.3   0.6
I5_4 |   3.1   1.1   1.3
I5_5 |   0.7   -7.7   0.8
I6_1 |   -2.4   3.6   -1.6
I6_2 |   2.4   -2.1   1.2
I6_3 |   0.8   -0.9   -4.6
I6_4 |   1.5   -2.3   5.4
I6_5 |   0.7   -1.3   2.5
I7_1 |   0.2   3.7   1.4
I7_2 |   4.6   -0.6   -2.1
I7_3 |   -0.6   -1.2   -3.4
I7_4 |   0.1   2.2   0.9
I7_5 |   -0.0   -7.7   0.4
I8_1 |   -2.6   1.1   -1.9
I8_2 |   2.5   -2.8   -3.3
I8_3 |   2.0   5.5   4.0
I8_4 |   1.8   -2.5   -0.4
I8_5 |   0.6   -2.8   -1.1
-------------------------------------------------------------------------------------------
      Note: Input Node = I, Hidden Node = H

                              Hidden Layer to Output Layer
-------------------------------------------------------------------------------------------
H/O   |   O1
-------------------------------------------------------------------------------------------
H1    |   7.2
H2    |   -28.8
H3    |   29.6
-------------------------------------------------------------------------------------------
      Note: Hidden Node = I, Output Node = H

  Accuracy of Prediction
    Training Set = 99.82%
    Testing Set = 92.86%
    Total Set = 98.43%

Step 6: Linguistic Rule Creation from Positive Weight
                              Linguistic Rules
  R1: If x6 = very large or x1 = very large or x2 = very large or x3 = very large or x7 = very
large or x4 = very large or x8 = very large or x8 = medium or x4 = large or x2 = medium or x5
= very large or x6 = large or x6 = small then class 1
  R2: If x1 = large and x5 = small
  R3: If x8 = large and x3 = small then class 1
      Else class 2

 Number of Rules = 3
 Number of Conditions = 14
 Accuracy of Rules
    Training Set = 97.67%
    Testing Set = 96.43%
    Total Set = 97.42%
 Time Run 38 sec.
```

ภาพประกอบ ข.15 (ต่อ)

ตัวอย่าง ถ้าเลือกสุ่มข้อมูลแบบความเที่ยงตรง K กลุ่ม โดย K = 10 กำหนดให้จำนวนรูปแบบภาษาธรรมชาติ = 5 และจำนวนรอบในการเรียนรู้ = 200 รอบ คลิกปุ่ม Rule Extraction เพื่อสกัดกฎภาษาธรรมชาติ โปรแกรมจะทำการประมวลผล 10 รอบและสรุปผลที่ได้ดังภาพประกอบ ข.16

```
                                                  Date: 11-Mar-2006
                                                  Database Name = WBC
                                                  Percent Weight Accept = 20
                              Result from Phase II: RENN-FIDR

Test Option = K-Fold Crossvalidation
  K = 10
  Number of Training Set = 629  (90 %)
  Number of Testing Set = 70  (10 %)
  Number of Total Set = 699  (100 %)

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ Experiment 1 ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

Step 4: Linguistic Intervals Creation from Frequency Algorithm
 Linguistic Terms = 5
 Threshold Frequency Interval = 50
                              Linguistic Intervals
-----------------------------------------------------------------------------------------
x / ling. terms |   very small,   small,   medium,   large,   very large
-----------------------------------------------------------------------------------------
     x1     | [1.0,5.0), [5.0,6.0), [6.0,7.0), [7.0,9.0), [9.0,10.0]
     x2     | [1.0,4.0), [4.0,7.0), [7.0,8.0), [8.0,10.0), [10.0,10.0]
     x3     | [1.0,3.0), [3.0,4.0), [4.0,6.0), [6.0,9.0), [9.0,10.0]
     x4     | [1.0,2.0), [2.0,4.0), [4.0,7.0), [7.0,9.0), [9.0,10.0]
     x5     | [1.0,3.0), [3.0,4.0), [4.0,9.0), [9.0,10.0), [10.0,10.0]
     x6     | [1.0,6.0), [6.0,7.0), [7.0,9.0), [9.0,10.0), [10.0,10.0]
     x7     | [1.0,4.0), [4.0,6.0), [6.0,7.0), [7.0,8.0), [8.0,10.0]
     x8     | [1.0,3.0), [3.0,4.0), [4.0,5.0), [5.0,10.0), [10.0,10.0]
-----------------------------------------------------------------------------------------

Step 5: Neural Networks Training with Linguistic Terms
 Number of Hidden Nodes = 3
 Training Times = 200 Epochs
 Weight
                              Input Layer to Hidden Layer
-----------------------------------------------------------------------------------------
 I/H   |    H1    H2    H3
-----------------------------------------------------------------------------------------
 I1_1 |   0.9   19.9   -18.2
 I1_2 |   32.4   -198.6   -32.1
 I1_3 |   -34.6   31.2   -27.1
 I1_4 |   12.0   94.9   24.9
 I1_5 |   -5.2   51.4   61.7
```

ภาพประกอบ ข.16 ผลลัพธ์ในขั้นตอนหลักที่ II (RENN-FIDR) เมื่อเลือกสุ่มข้อมูล

แบบความเที่ยงตรง 10 กลุ่ม

I2_1	-21.0	-39.1	-34.7
I2_2	20.4	31.7	13.9
I2_3	5.4	-2.5	26.3
I2_4	-21.6	-13.9	28.7
I2_5	26.6	22.9	-20.1
I3_1	-29.9	-62.3	-5.3
I3_2	-13.4	-44.1	44.0
I3_3	58.2	-16.1	-35.5
I3_4	-6.4	29.2	-17.8
I3_5	-0.3	89.4	22.9
I4_1	-94.0	137.1	-1.9
I4_2	-12.7	-101.8	6.6
I4_3	84.5	20.3	-49.5
I4_4	15.7	0.9	4.0
I4_5	15.7	-58.5	55.9
I5_1	5.8	-8.1	-13.5
I5_2	-19.0	-9.0	-41.6
I5_3	39.5	-42.6	37.7
I5_4	0.2	1.9	0.9
I5_5	-17.5	58.5	31.0
I6_1	-37.5	-110.1	-45.6
I6_2	-12.7	10.0	25.0
I6_3	35.2	-2.0	-16.7
I6_4	4.2	-14.3	1.2
I6_5	24.2	114.5	47.0
I7_1	-43.4	-118.3	18.0
I7_2	-9.7	34.1	39.8
I7_3	0.1	-12.3	-15.5
I7_4	42.8	74.5	-58.1
I7_5	18.5	18.2	29.7
I8_1	-26.6	51.4	-36.6
I8_2	18.4	-112.7	-12.6
I8_3	1.0	9.9	43.9
I8_4	-11.4	-4.4	-14.1
I8_5	27.5	53.1	34.5

Note: Input Node = I, Hidden Node = H

Hidden Layer to Output Layer

H/O	O1
H1	5.6
H2	82.8
H3	17.7

Note: Hidden Node = I, Output Node = H

Accuracy of Prediction
Training Set = 97.14%
Testing Set = 97.14%
Total Set = 97.14%

ภาพประกอบ ข.16 (ต่อ)

```
Step 6: Linguistic Rule Creation from Positive Weight
                              Linguistic Rules
   R1: If x6 = very large or x1 = very large or x2 = very large or x3 = very large or x8 = very
large or x7 = very large or x4 = large or x6 = medium or x8 = medium or x2 = medium or x6 =
large or x6 = small then class 1
   R2: If x4 = medium and x3 = medium
   R3: If x4 = very large and x1 = large then class 1
        Else class 2

  Number of Rules = 3
  Number of Conditions = 13
  Accuracy of Rules
     Training Set = 97.62%
     Testing Set = 97.14%
     Total Set = 97.57%
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^ End of Experiment 1 ^^^^^^^^^^^^^^^^^^^^^^^^^^^^
                                        •
                                        •

                                        •

^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^ End of Experiment 10 ^^^^^^^^^^^^^^^^^^^^^^^^^^^^
Conclusion
  Number of Rules = 3
  Number of Conditions = 11
  Accuracy of Rules
     Training Set = 96.62%
     Testing Set = 94.41%
     Total Set = 96.39%
  Time Run Average = 15 sec.
```

ภาพประกอบ ข.16 (ต่อ)

จะเห็นผู้ใช้สามารถใช้งานโปรแกรมการสกัดกฎภาษาธรรมชาติโดยใช้โครงข่ายประสาทเทียมได้ง่ายเนื่องจากมีส่วนการติดต่อกับผู้ใช้เป็นแบบ Graphic User Interface และแสดงผลลัพธ์ของการทำงานในแต่ละขั้นตอนอย่างชัดเจน โดยในการทำงานงานมี 3 หน้าจอหลักคือ หน้าจอสำหรับโหลดข้อมูล หน้าจอสำหรับการตัดโหนดโครงข่ายประสาทเทียมโดยใช้ค่าน้ำหนักสูงสุด (NNP-MFW) และหน้าจอสำหรับการสกัดกฎภาษาธรรมชาติจากโครงข่ายประสาทเทียมโดยการแทนค่าความถี่ของช่วงข้อมูล (RENN-FIDR)