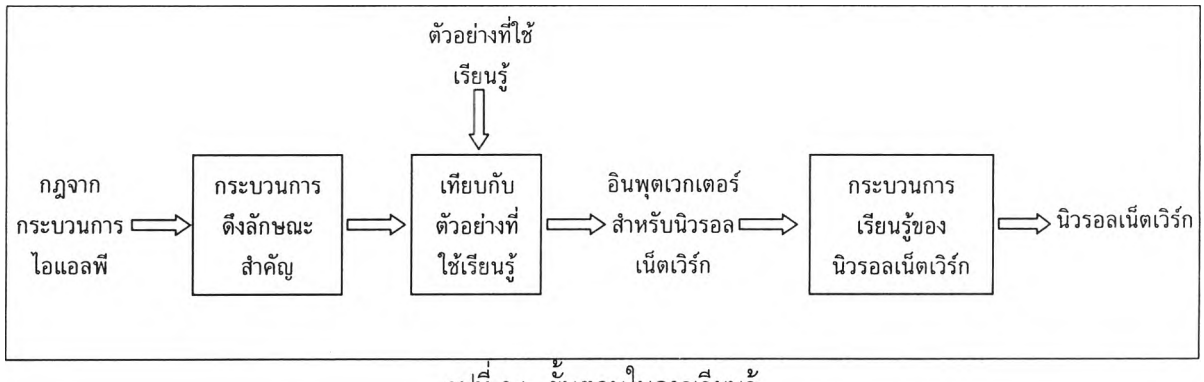
บทที่ 3

วิธีดำเนินการวิจัย

ในบทนี้จะได้กล่าวถึงวิธีดำเนินการทดลอง โครงสร้างของระบบ กระบวนการสำคัญในการวิจัยซึ่งประกอบด้วยกระบวนการดึงลักษณะสำคัญ (feature extraction) การสร้างนิวรอลเน็ตเวิร์กจากลักษณะสำคัญ และกระบวนการเรียนรู้ของนิวรอลเน็ตเวิร์ก

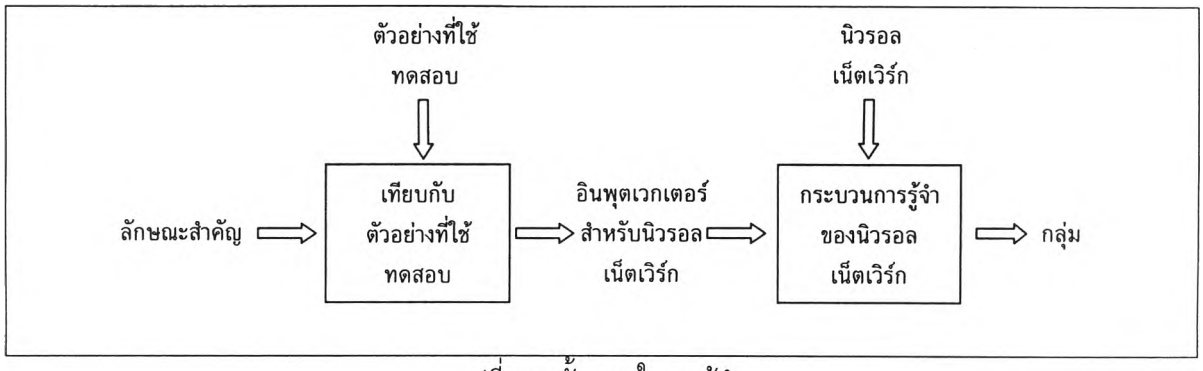
โครงสร้างของระบบ

การประมาณกฎจากระบบไอแอลพีด้วยวิธีการแบ็กพรอพาเกชันนิวรอลเน็ตเวิร์ก แบ่งเป็น 2 ขั้นตอนสำคัญ คือ ขั้นตอนการเรียนรู้ และขั้นตอนการรู้จำ โดยขั้นตอนการเรียนรู้จะเริ่มจากนำกฎที่ได้จากระบบไอแอลพีมาหาลักษณะสำคัญ จากนั้นนำลักษณะสำคัญที่ได้มาเทียบกับตัวอย่าง โดยเลือกการแทนที่ที่ทำให้ลักษณะสำคัญมีค่าความจริงเป็นจริงมากที่สุด จากนั้นนำค่าความจริงที่ได้มาเป็นอินพุตเวกเตอร์เพื่อป้อนให้นิวรอลเน็ตเวิร์กทำการเรียนรู้โดยขั้นตอนวิธีแบ็กพรอพาเกชัน เน็ตเวิร์กที่ได้จะถูกนำไปรู้จำตัวอย่างใหม่ที่ใช้ทำการทดสอบ ขั้นตอนในกระบวนการเรียนรู้ แสดงในรูปที่ 3.1



รูปที่ 3.1 ขั้นตอนในการเรียนรู้

ขั้นตอนการรู้จำจะแตกต่างจากขั้นตอนการเรียนรู้ในส่วนของการดึงลักษณะสำคัญจากกฎ ในการรู้จำไม่ต้องดึงลักษณะสำคัญอีก นำลักษณะสำคัญที่หาได้ในขั้นตอนการเรียนรู้มาใช้เทียบกับตัวอย่าง เลือกการแทนที่ที่ทำให้ลักษณะสำคัญมีค่าความจริงเป็นจริงมากที่สุด นำค่าความจริงที่ได้จัดเป็นอินพุตเวกเตอร์ให้กับนิวรอลเน็ตเวิร์ก จากนั้นเลือกกลุ่มจากเอาต์พุตนิวรอนซึ่งให้ค่าสูงที่สุด ขั้นตอนในการรู้จำ แสดงในรูปที่ 3.2



รูปที่ 3.2 ขั้นตอนในการรู้จำ

กระบวนการดึงลักษณะสำคัญ (Feature Extraction Process)

ปัญหาสำคัญของกฎลำดับที่หนึ่ง คือในกรณีปัญหาที่มีลักษณะเป็นหลายกลุ่ม เมื่อจำแนกตัวอย่างทดสอบจะพบว่าจะมีตัวอย่างบางตัวอย่างที่ไม่ตรงกับกฎข้อใดเลย ระบบไอแอลพีเพียงลำพังจะไม่สามารถบอกได้ว่าตัวอย่างนั้นควรจะถูกจำแนกเป็นกลุ่มใด ในการวิจัยครั้งนี้มีแนวคิดว่าเมื่อไม่มีกฎข้อใดที่ตรงกับตัวอย่างพอดี เราสามารถใช้การครอบคลุมเพียงบางส่วน (partially cover) จากกฎแต่ละข้อเมื่อนำกฎไปเทียบกับตัวอย่าง เพื่อจำแนกตัวอย่างออกเป็นกลุ่มได้ การครอบคลุมเพียงบางส่วนอาจครอบคลุมคุณสมบัติบางอย่างที่สำคัญของกฎในแต่ละข้อ ซึ่งนำไปสู่การจำแนกตัวอย่างนั้นๆ ได้ ในกรณีที่ไม่สามารถหากฎที่ตรงกับตัวอย่างพอดี กฎที่ควรจะตรงกับตัวอย่างที่สุดควรเป็นกฎที่มีลักษณะสำคัญตรงกับตัวอย่างนั้นมากกว่ากฎข้ออื่นๆ โดยส่วนที่ไม่ตรงกับกฎควรเป็นลักษณะที่ไม่สำคัญ ในการทดลองครั้งนี้ได้ใช้แบ็กพรอพาเกชันนิวรอลเน็ตเวิร์กมาทำการเรียนรู้ เพื่อกำหนดน้ำหนักซึ่งเป็นการให้ความสำคัญกับลักษณะสำคัญแต่ละข้อของกฎ

ตัวอย่างเช่น กฎลำดับที่หนึ่งซึ่งทุกสัญพจน์ภายในกฎข้อนั้นมีแต่ตัวแปรซึ่งอยู่ในส่วนหัวของกฎ เราสามารถดึงลักษณะสำคัญ โดยใช้แต่ละสัญพจน์ในกฎข้อนั้นเป็นลักษณะสำคัญของกฎได้ ตัวอย่างเช่น

```
mesh(A,1) :- not_important(A), not_loaded(A).
```

จากตัวอย่างซึ่งเป็นกฎลำดับที่หนึ่งข้างต้นจะเห็นว่า สัญพจน์แต่ละตัวมีแต่ตัวแปรซึ่งอยู่ในส่วนหัวทั้งสิ้นและไม่มีตัวแปรใหม่ปรากฏอยู่ในสัญพจน์นั้นเลย เราจึงสามารถใช้สัญพจน์แต่ละตัวเป็นลักษณะสำคัญได้ และเรียกสัญพจน์ที่มีลักษณะแบบนี้ว่า ลักษณะสำคัญเดี่ยว (singleton feature)

แต่โดยปกติแล้วกฎในลำดับที่หนึ่งที่ได้จากระบบไอแอลพีจะมีความซับซ้อนกว่าตัวอย่างข้างต้นมาก ดังนั้นจึงต้องหาวิธีการที่สามารถดึงลักษณะสำคัญโดยอาศัยตัวแปรใหม่ที่เกิดขึ้นในแต่ละสัญพจน์ได้ ในระบบไอแอลพีโดยทั่วไปสัญพจน์ที่มีตัวแปรใหม่จะไม่สามารถอยู่โดดเดี่ยวโดยปราศจากสัญพจน์อื่นซึ่งสร้างตัวแปรนี้ขึ้นมา และสัญพจน์ส่วนใหญ่ที่สร้างตัวแปรใหม่ จะถูกรวมเข้าอยู่ในกฎลำดับที่หนึ่งเพื่อทำหน้าที่ส่งผ่านตัวแปรใหม่นั้นไปยังสัญพจน์อื่น เพื่อใช้ตรวจสอบคุณสมบัติเฉพาะของตัวอย่างนั้นๆ โดยใช้สัญพจน์อื่นต่อไป ดังนั้นในงานวิจัยนี้

จึงได้นำลักษณะการเกิดตัวแปรใหม่ในสัญพจน์มาสร้างเป็นลำดับของสัญพจน์ เพื่อสร้างเป็นลักษณะสำคัญจากกฎลำดับที่หนึ่ง

ในกระบวนการดึงลักษณะสำคัญมีคำนิยามที่สำคัญอยู่สองคำ คือ สายสัญพจน์ปิด (closed chain) และสายสัญพจน์เปิด (open chain) ทั้งสองสัญพจน์มีคำนิยามดังนี้

สายสัญพจน์ปิด

นิยาม สัญพจน์ใดๆ จะเป็นส่วนหนึ่งของสายสัญพจน์ปิดถ้าตัวแปรใหม่ทุกตัวในสัญพจน์นั้นซึ่งไม่อยู่ในส่วนหัวของกฎปรากฏในสัญพจน์อย่างน้อยสองสัญพจน์ และปรากฏในสัญพจน์อย่างน้อยหนึ่งสัญพจน์ร่วมกับตัวแปรในส่วนหัวของกฎหรือตัวแปรซึ่งเคยมีอยู่แล้วในสายสัญพจน์นั้นๆ

หรือกล่าวอีกนัยหนึ่งว่า สายสัญพจน์ปิดจะเริ่มจาก ตัวแปรซึ่งอยู่ในส่วนหัวของกฎ แล้วตัวแปรเหล่านี้จะไปสร้างตัวแปรใหม่ในสัญพจน์ซึ่งอยู่ในกฎ ตัวแปรใหม่ที่ถูกสร้างขึ้นไปสร้างตัวแปรใหม่ขึ้นอีก เป็นลำดับอย่างนี้ไปจนสิ้นสุดที่สัญพจน์ซึ่งไม่มีการสร้างตัวแปรใหม่อีก สัญพจน์ที่ไม่มีการสร้างตัวแปรใหม่นี้โดยส่วนใหญ่ในกฎลำดับที่หนึ่งจะเป็นสัญพจน์ที่บอกลักษณะพิเศษของตัวอย่างนั้น ตัวอย่างเช่น

จากกฎ

```
mesh(A,1) :- short(A), neighbour_yz_l(B,A), usual(B).
```

จะเห็นว่า ในส่วนหัวของกฎมีตัวแปร A ปรากฏอยู่ สัญพจน์ short(A) ซึ่งมีแต่ตัวแปรที่อยู่ในส่วนหัวเท่านั้น จะถูกจัดเป็นหนึ่งลักษณะสำคัญที่เป็นลักษณะสำคัญเดี่ยว นอกจากลักษณะสำคัญนี้แล้ว ลักษณะสำคัญหนึ่งที่ถูกสร้างขึ้นในลักษณะของสายสัญพจน์ปิด คือ

```
neighbour_yz_l(B,A), usual(B).
```

เป็นลักษณะสำคัญที่เริ่มจากตัวแปร A ในส่วนหัวของกฎ จากนั้นที่สัญพจน์ neighbour_yz_l (B,A) ตัวแปร A สร้างตัวแปรใหม่ B ดังนั้นสัญพจน์ neighbour_yz_l(B,A) จะถูกรวมเข้าไว้ในสายสัญพจน์ จากนั้นตัวแปร B ซึ่งถูกสร้างขึ้นในสัญพจน์ neighbour_yz_l(B,A) ปรากฏอีกครั้งในสัญพจน์ usual(B) และในสัญพจน์ usual(B) นี้มีแต่ตัวแปรที่เคยเกิดขึ้นแล้ว และไม่มีการสร้างตัวแปรใหม่ ทำให้สายสัญพจน์นี้สิ้นสุดที่สัญพจน์ usual(B) และเป็นสายสัญพจน์ปิด

จะเห็นได้ว่า จากตัวอย่างของสายสัญพจน์ปิดข้างต้น สัญพจน์ neighbour_yz_l(B,A) ปรากฏอยู่ในกฎข้อนี้เพื่อทำหน้าที่เพียงสร้างตัวแปร B ขึ้นจากตัวแปร A และส่งต่อตัวแปร B ไปยังสัญพจน์อื่น ในขณะที่สัญพจน์ usual(B) เป็นสัญพจน์ที่ใช้ระบุลักษณะเฉพาะของตัวอย่าง สัญพจน์ usual(B) จะปรากฏอยู่ในลักษณะสำคัญโดยไม่มีสัญพจน์ neighbour_yz_l(B,A) ร่วมอยู่ไม่ได้ ดังนั้นจึงต้องรวมทั้งสองสัญพจน์ไว้ในสายสัญพจน์ปิดด้วยกัน

I20388317

สายสัญพจน์เปิด

นิยาม สัญพจน์ใดๆ จะเป็นส่วนหนึ่งของสายสัญพจน์เปิดถ้ามีตัวแปรใหม่ในสัญพจน์นั้นซึ่งไม่อยู่ในส่วนหัวของกฎปรากฏในสัญพจน์เพียงสัญพจน์เดียวเท่านั้นร่วมกับตัวแปรในส่วนหัวของกฎหรือตัวแปรซึ่งเคยมีอยู่แล้วในสายสัญพจน์นั้นๆ

ลักษณะการเกิดสายสัญพจน์เปิดในกฎลำดับที่หนึ่งจะคล้ายกับการเกิดสัญพจน์ปิด มีการนำลักษณะการเกิดตัวแปรใหม่มาใช้เช่นเดียวกันกับสายสัญพจน์ปิด โดยสัญพจน์ที่มีตัวแปรใหม่จะถูกรวมเข้าไว้ในสายสัญพจน์เช่นกัน เพียงแต่ในสายสัญพจน์เปิดเมื่อมีตัวแปรใหม่เกิดขึ้นและตัวแปรใหม่นี้ไม่ปรากฏในสัญพจน์อื่น นั่นคือตัวแปรใหม่ที่เกิดขึ้นนี้ไม่ได้ทำหน้าที่สร้างตัวแปรอื่น หรือกำหนดลักษณะเฉพาะของตัวอย่าง ต่างจากในสายสัญพจน์ปิด ซึ่งตัวแปรใหม่ทุกตัวที่เกิดขึ้นจะต้องทำหน้าที่อย่างใดอย่างหนึ่ง คือ สร้างตัวแปรอื่น หรือกำหนดลักษณะเฉพาะของตัวอย่าง ตัวอย่างของสายสัญพจน์เปิด เช่น

จากกฎ

```
mesh(A,2) :- short(A), neighbour_xy_1(B,A), neighbour_zx_1(C,B).
```

ลักษณะสำคัญหนึ่งที่ถูกสร้างขึ้นในลักษณะของสายสัญพจน์เปิด คือ

```
neighbour_xy_1(B,A), neighbour_zx_1(C,B).
```

จากตัวอย่างจะเห็นว่า ตัวแปร A ซึ่งอยู่ภายในส่วนหัวของกฎ ทำหน้าที่สร้างตัวแปรใหม่ B ขึ้นภายในสัญพจน์ neighbour_xy_1(B,A) และตัวแปร B ที่เกิดขึ้นใหม่ ทำหน้าที่สร้างตัวแปรใหม่ C ที่สัญพจน์ neighbour_ zx_1(C,B) จะเห็นได้ว่าสัญพจน์ neighbour_xy_1(B,A) ทำหน้าที่ส่งต่อตัวแปร B เพื่อไปสร้างตัวแปรใหม่ C ยังสัญพจน์ถัดไป ดังนั้นสัญพจน์ neighbour_xy_1(B,A) จึงถูกรวมในสายสัญพจน์เปิด ส่วนในสัญพจน์ neighbour_zx_1(C,B) ตัวแปร C ที่ถูกสร้างขึ้นไม่ปรากฏอยู่ในสัญพจน์อื่นอีกเลย ดังนั้นสายสัญพจน์นี้จึงเป็นสายสัญพจน์เปิด

ตัวอย่างสายสัญพจน์ปิดและสายสัญพจน์เปิด

จากกฎ

```
p(A,B) :-  q1(A), q2(A,C), q3(C), q4(C,D), q5(D), q6(A,E,F), q7(E,G),
           q8(E,H), q9(E), q10(F,I), q11(I,B).
```

จะได้สายสัญพจน์ปิดดังต่อไปนี้

```
1) q2(A,C),q3(C)
2) q2(A,C),q4(C,D),q5(D)
3) q2(A,C),q3(C),q4(C,D),q5(D)
4) q6(A,E,F),q9(E),q10(F,I),q11(I,B)
```

จะได้สายสัญพจน์เปิดดังต่อไปนี้

```
1) q6(A,E,F),q7(E,G)
2) q6(A,E,F),q8(E,H)
3) q6(A,E,F),q7(E,G),q8(E,H)
4) q6(A,E,F),q7(E,G),q9(E)
5) q6(A,E,F),q8(E,G),q9(E)
6) q6(A,E,F),q7(E,G),q8(E,H),q9(E)
```

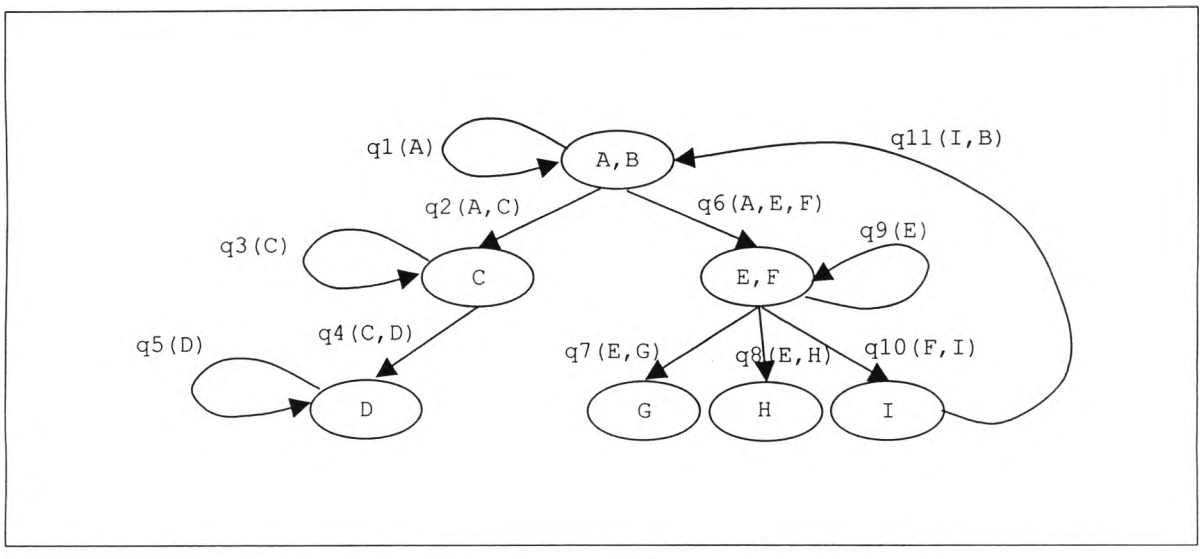
จากตัวอย่างจะเห็นว่า บางสัญพจน์ไม่สามารถปรากฏเพียงลำพังในสายสัญพจน์ได้ คือ สัญพจน์ที่มีตัวแปรใหม่ จะต้องปรากฏร่วมกับสัญพจน์อื่นที่มีตัวแปรซึ่งเคยปรากฏมาแล้ว หรือเป็นตัวแปรในส่วนหัวของกฎ หรือกล่าวอีกนัยหนึ่งว่า สัญพจน์ที่มีตัวแปรใหม่จะต้องปรากฏร่วมกับสัญพจน์ที่ทำหน้าที่สร้างตัวแปรนั้นซึ่งอาจเป็นสัญพจน์ในส่วนหัวของกฎหรือเป็นสัญพจน์ในตัวกฎก็ได้ เพราะถ้าสัญพจน์ซึ่งมีตัวแปรใหม่ปรากฏโดยไม่มีสัญพจน์ซึ่งทำหน้าที่สร้างตัวแปรนั้น จะทำให้ไม่สามารถหาการแทนที่ตัวแปรนั้นได้ ตัวอย่างของสายสัญพจน์เหล่านี้ เช่น "q3(C)", "q1(A), q3(C)", "q5(D)", "q1(A), q5(D)" ฯลฯ ตามคำนิยามของทั้งสายสัญพจน์ปิดและสายสัญพจน์เปิดจะจำกัดไม่ให้มีสายสัญพจน์เหล่านี้เกิดขึ้นอยู่แล้ว เนื่องจาก จากคำนิยามของสายสัญพจน์ทั้งสองแบบ สัญพจน์ใดๆ จะถูกเข้ารวมเข้าไว้ในสายสัญพจน์ ถ้าในสัญพจน์นั้นมีตัวแปรใหม่ปรากฏร่วมกับตัวแปรที่มีอยู่แล้วในส่วนหัวของกฎหรือเคยปรากฏร่วมกับตัวแปรอื่นในสายสัญพจน์นั้นๆ

ขั้นตอนวิธีการดึงลักษณะสำคัญ (feature extraction algorithm)

จากคำนิยามของสายสัญพจน์ปิดและสายสัญพจน์เปิดจะเห็นได้ว่า สายสัญพจน์ทั้งสองมีแนวคิดมาจากการเกิดตัวแปรใหม่ภายในสัญพจน์ ดังนั้นเพื่อให้สามารถเข้าใจวิธีการดึงลักษณะสำคัญได้ง่าย ในงานวิจัยนี้จึงได้ทำการแทนกฎลำดับที่หนึ่งด้วยกราฟ โดยให้โนด (node) ของกราฟแทนตัวแปรใหม่ โนดเริ่มต้น (initial node) แทนตัวแปรซึ่งอยู่ในส่วนหัวของกฎ และเส้นเชื่อม (edge) แทนสัญพจน์ซึ่งมีตัวแปรนั้นๆ จากตัวอย่างของกฎลำดับที่หนึ่ง

```
p(A,B) :-  q1(A), q2(A,C), q3(C), q4(C,D), q5(D), q6(A,E,F), q7(E,G),
           q8(E,H), q9(E), q10(F,I), q11(I,B)
```

จากกฎ ในสัญพจน์ส่วนหัว `p(A,B)` โนดซึ่งแทนตัวแปร `A` และ `B` จะถูกจัดเป็นโนดเริ่มต้นของกราฟ สัญพจน์ `q1(A)` ไม่ได้ทำหน้าที่สร้างตัวแปรใหม่ มีเพียงตัวแปร `A` ปรากฏอยู่ภายในสัญพจน์ จึงมีเส้นเชื่อมซึ่งแทนสัญพจน์ `q1(A)` ออกและวนกลับเข้าที่โนดนั้น จากนั้นที่สัญพจน์ `q2(A,C)` ตัวแปร `A` ทำหน้าที่สร้างตัวแปรใหม่ `C` ดังนั้นจึงมีเส้นเชื่อมซึ่งแทนสัญพจน์ `q2(A,C)` ออกจากโนดเริ่มต้นไปยังโนดใหม่ซึ่งแทนตัวแปร `C` จากกฎข้างต้น เราสามารถแทนด้วยกราฟได้ดังรูปที่ 3.3



รูปที่ 3.3 ตัวอย่างของกราฟซึ่งใช้แทนกฎลำดับที่หนึ่ง

ขั้นตอนวิธีการดึงลักษณะสำคัญ

(1) หาเส้นเชื่อมทุกเส้นซึ่งเริ่มต้นและสิ้นสุดที่โนดเริ่มต้น ใช้สัญพจน์เหล่านั้นเป็นลักษณะสำคัญเดี่ยว

(2) หาสายสัญพจน์ปิดทุกสายที่เป็นไปได้ซึ่งเริ่มต้นจากโนดเริ่มต้น สายสัญพจน์ที่ได้ในขั้นตอนนี้เป็นสายสัญพจน์ปิด

(3) หาทางเดินที่เป็นไปได้ทุกทางซึ่งเริ่มต้นจากโนดเริ่มต้นแล้วไปสิ้นสุดที่โนดสิ้นสุด (terminal node) ซึ่งเป็นโนดที่ไม่มีเส้นเชื่อมออกจากตัวเอง สายสัญพจน์ที่ได้ในขั้นตอนนี้เป็นสายสัญพจน์เปิด

(4) หาการรวม (combination) ทุกแบบที่เป็นไปได้ของสายสัญพจน์เปิดซึ่งได้จากขั้นที่สาม ที่มีตัวแปรใหม่ร่วมกัน

ตัวอย่างกระบวนการดึงลักษณะสำคัญ

```
p(A,B) :-  q1(A), q2(A,C), q3(C), q4(C,D), q5(D), q6(A,E,F), q7(E,G),
           q8(E,H), q9(E), q10(F,I), q11(I,B)
```

จากกฎข้างต้นสามารถดึงลักษณะสำคัญได้ดังต่อไปนี้

ขั้นที่ 1

1) q1(A)

ขั้นที่ 2

2) q2(A,C),q3(C)

3) q2(A,C),q4(C,D),q5(D)

4) q2(A,C),q3(C),q4(C,D),q5(D)

5) q6(A,E,F),q9(E),q10(F,I),q11(I,B)

ขั้นที่ 3

6) q6(A,E,F),q7(E,G)

7) q6(A,E,F),q8(E,H)

8) q6(A,E,F),q9(E),q7(E,G)

9) q6(A,E,F),q9(E),q8(E,H)

ขั้นที่ 4

10) q6(A,E,F),q7(E,G),q8(E,H)

11) q6(A,E,F),q7(E,G),q8(E,H),q9(E)

จากตัวอย่างจะเห็นว่า ตามขั้นตอนวิธีไม่ได้มีการสร้างสายสัญพจน์เปิดทุกสายสัญพจน์ที่เป็นไปได้ โดยสายสัญพจน์เปิดที่ไม่ได้ถูกสร้างขึ้น ได้แก่ สายสัญพจน์เปิดที่เป็นส่วนหนึ่งของสายสัญพจน์ปิด เช่น "q2(A,C)", "q2(A,C),q4(C,D)" และ "q6(A,E,F),q9(E),q10(F,I)" ฯลฯ เนื่องจาก โดยปกติแล้วการสร้างตัวแปรใหม่ในกฎลำดับที่หนึ่ง มีจุดมุ่งหมายเพื่อใช้ตัวแปรใหม่นั้นในสัญพจน์อื่นซึ่งทำหน้าที่ตรวจสอบลักษณะเฉพาะของตัวอย่างนั้นๆ หรือทำหน้าที่สร้างตัวแปรใหม่ตัวอื่น ดังนั้นสายสัญพจน์ทุกสายสัญพจน์ที่สร้างขึ้นจึงควรสิ้นสุดที่สัญพจน์ซึ่งไม่มีตัวแปรใหม่เกิดขึ้น แต่ในบางกรณีที่สร้างสายสัญพจน์ปิดไม่ได้จึงจำเป็นต้องสร้างสายสัญพจน์เปิดด้วย

จากลักษณะสำคัญที่ได้ เราสามารถเปลี่ยนรูปแบบของกฎเพื่อนำไปใช้สร้างนิวรอลเน็ตเวิร์กโดยใช้ลักษณะสำคัญเดี่ยว สายสัญพจน์ปิด และสายสัญพจน์เปิด ประกอบเข้าไปในกฎเพื่อทำให้สามารถหาค่าความจริงของแต่ละลักษณะสำคัญได้โดยไม่ทำให้ความหมายเดิมของกฎเปลี่ยนไป ตัวอย่างเช่น จากตัวอย่างของกระบวนการดึงลักษณะสำคัญข้างต้น กฎที่ผ่านการเปลี่ยนรูปแบบแล้วจะเป็นดังรูปที่ 3.4

การสร้างนิวรอลเน็ตเวิร์กจากลักษณะสำคัญ

จากลักษณะสำคัญที่ได้ กฎที่สร้างจากระบบไอแอลพีบางข้อจำเป็นต้องมีการเปลี่ยนรูปของกฎก่อน เพราะระบบไอแอลพีสามารถสร้างกฎที่แทนตัวแปรด้วยค่าคงที่ภายในส่วนหัวของกฎ หรือสามารถเปลี่ยนตัวแปรภายในส่วนหัวของกฎให้เป็นตัวแปรเดียวกันได้ หากนำกฎซึ่งมีการแทนที่ตัวแปรในส่วนหัวแล้วไปทำการหาลักษณะสำคัญในตัวกฎจะให้ความหมายในส่วนหัวของกฎหายไป ตัวอย่างกฎลักษณะนี้เช่น

illegal(WKf,3,WRf,WRr,BKf,WRr) ← lt(WKf,3).

```
p(A,B) ← q1(A),
         q2(A,C),q3(C),
         q2(A,C),q4(C,D),q5(D),
         q2(A,C),q3(C),q4(C,D),q5(D),
         q6(A,E,F),q9(E),q10(F,I),q11(I,B),
         q6(A,E,F),q7(E,G),
         q6(A,E,F),q8(E,H),
         q6(A,E,F),q9(E),q7(E,G),
         q6(A,E,F),q9(E),q8(E,H),
         q6(A,E,F),q7(E,G),q8(E,H),
         q6(A,E,F),q7(E,G),q8(E,H),q9(E).
```

รูปที่ 3.4 ตัวอย่างของลักษณะสำคัญที่อยู่ภายในกฎ

ตัวอย่างที่จะตรงกับกฎข้อนี้ได้ อาร์กิวเมนต์ตัวที่สองจะต้องมีค่าเป็น 3 อาร์กิวเมนต์ตัวที่สี่และหก จะต้องมีค่าเท่ากัน และคุณสมบัติ `lt(WKf,3)` จะต้องเป็นจริงเมื่อ `WKf` เป็นอาร์กิวเมนต์ตัวที่หนึ่งของตัวอย่างนั้น จะเห็นว่ามีลักษณะเฉพาะสองอย่างที่ปรากฏอยู่ในส่วนหัวโดยไม่ปรากฏในส่วนตัวของกฎ คือ อาร์กิวเมนต์ตัวที่สองของตัวอย่างจะต้องมีค่าเป็น 3 และอาร์กิวเมนต์ตัวที่สี่และหกจะต้องมีค่าเหมือนกัน ดังนั้นเพื่อให้ความหมายของกฎยังอยู่ครบถ้วน จึงจำเป็นต้องมีการแปลงรูปแบบของกฎซึ่งมีค่าคงที่หรือตัวแปรซ้ำกันภายในส่วนหัวของกฎก่อน โดยใช้ตัวแปรที่ไม่ซ้ำกันมาแทน และเพิ่มการแทนที่ของตัวแปรเหล่านั้นเข้าไปในส่วนตัวกฎ จากตัวอย่างกฎข้างต้นจะสามารถแปลงได้เป็น

```
illegal(WKf,WKr,WRf,WRr,BKf,BKr) ← WKr=3,WRr=BKr,lt(WKf,3).
```

แต่ในกรณีของกฎซึ่งมีค่าคงที่ที่ใช้ระบุกลุ่มของตัวอย่างปรากฏอยู่ส่วนหัวของกฎ เช่น "`mesh(A,1) ← not_important(A),not_loaded(A).`" ไม่จำเป็นต้องเปลี่ยนค่าคงที่ในลักษณะนี้ให้ปรากฏในส่วนตัวกฎ เนื่องจากค่าคงที่ค่านี้เป็นการระบุถึงกลุ่มของตัวอย่าง ไม่ได้เป็นการระบุถึงลักษณะเฉพาะที่จำเป็นต้องตรวจสอบของตัวอย่างนั้นๆ

หลังจากการแปลงรูปแบบของกฎและกระบวนการดึงลักษณะสำคัญแล้ว ลักษณะสำคัญที่ได้จะอยู่ในรูปของลักษณะสำคัญเดี่ยว สายสัญพจน์ปิด และสายสัญพจน์เปิด ดังรูปที่ 3.4 ทำการสร้างนิวรอลเน็ตเวิร์กโดยกำหนดให้ประกอบด้วย 3 ชั้น คือ ชั้นอินพุต (input layer) ชั้นซ่อน (hidden layer) และชั้นเอาต์พุต (output layer) นิวรอนในชั้นอินพุตแทนลักษณะสำคัญของกฎแต่ละข้อ ดังนั้นจำนวนนิวรอนที่มีในชั้นอินพุตจะเท่ากับจำนวนลักษณะสำคัญที่มีอยู่ในเซตของกฎ นิวรอนในชั้นซ่อนแทนกฎแต่ละข้อ จำนวนนิวรอนในชั้นนี้จึงมีค่าเท่ากับจำนวนกฎ นิวรอนในชั้นอินพุตซึ่งแทนลักษณะสำคัญภายในกฎแต่ละข้อจะมีเส้นเชื่อมมายังนิวรอนในชั้นซ่อนซึ่ง

แทนกฎข้อนั้นๆ นิวรอนในชั้นเอาต์พุตแทนกลุ่มของตัวอย่าง การเชื่อมต่อจากชั้นซ่อนมายังชั้นเอาต์พุตเป็นการเชื่อมต่อแบบทั้งหมด (fully connected) จำนวนนิวรอนในชั้นเอาต์พุตขึ้นกับจำนวนกลุ่มของตัวอย่าง ในกรณีของปัญหาแบบหลายกลุ่ม จำนวนนิวรอนในชั้นเอาต์พุตจะมีค่าเท่ากับจำนวนกลุ่ม ส่วนในกรณีของปัญหาแบบสองกลุ่ม จำนวนนิวรอนจะมีค่าเท่ากับสอง นิวรอนตัวหนึ่งจะแทนกลุ่มที่เป็นตัวอย่างบวก ในขณะที่นิวรอนอีกตัวหนึ่งแทนกลุ่มที่เป็นกลุ่มตัวอย่าง

ตัวอย่างโครงสร้างของนิวรอลเน็ตเวิร์ก

สมมติว่าในเซตของกฎประกอบด้วยกฎ 4 ข้อ คือ {*C1, C2, C3, C4* } และตัวอย่างแบ่งออกเป็น 3 กลุ่ม คือ { 1, 2, 3 } กฎข้อ *C1* เป็นกฎสำหรับรู้จำตัวอย่างกลุ่ม "1" กฎข้อ *C2* และ *C3* เป็นกฎสำหรับรู้จำตัวอย่างกลุ่ม "2" และ กฎข้อ *C4* เป็นกฎสำหรับรู้จำตัวอย่างกลุ่ม "3"

```
C1:   mesh(A,1) ← not_important(A), not_loaded(A).
C2:   mesh(A,2) ← short(A), opposite_l(B,A).
C3:   mesh(A,2) ← usual(A), neighbour_yz_r(A,B), cont_loaded(B).
C4:   mesh(A,3) ← short(A), neighbour_zx_r(A,B), opposite_r(A,C),
                  short(C).
```

เมื่อผ่านกระบวนการดึงลักษณะสำคัญแล้วจะได้ลักษณะสำคัญดังต่อไปนี้

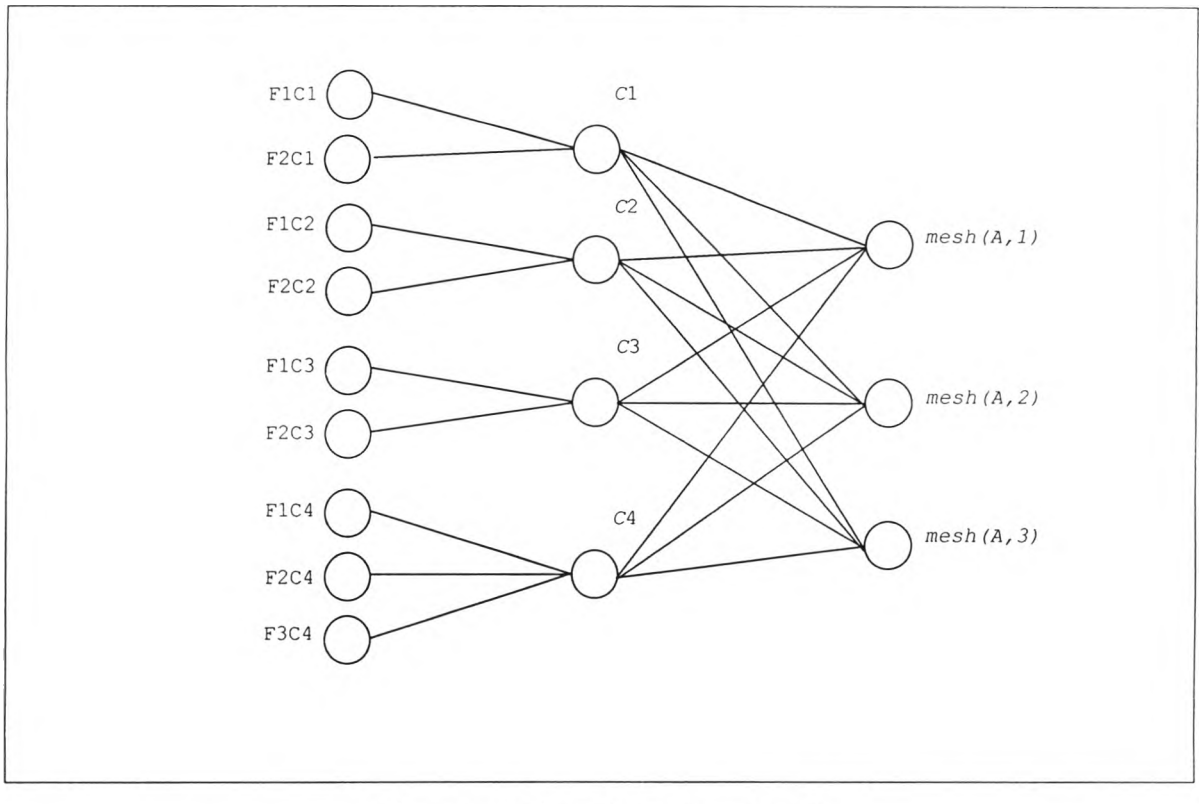
```
F1C1: not_important(A)
F2C1: not_loaded(A)

F1C2: short(A)
F2C2: opposite_l(B,A)

F1C3: usual(A)
F2C3: neighbour_yz_r(A,B), cont_loaded(B)

F1C4: short(A)
F2C4: opposite_r(A,C), short(C)
F3C4: neighbour_zx_r(A,B)
```

เมื่อกำหนดให้ *FiCj* แทนลักษณะสำคัญที่ *i* ในกฎข้อที่ *j*

จากตัวอย่างลักษณะสำคัญ *F1C1, F2C1, F1C2, F1C3* และ *F1C4* เป็นลักษณะสำคัญเดี่ยว *F2C3* และ *F2C4* เป็นสายสัญพจน์ปิด ในขณะที่ *F2C2, F3C4* เป็นสายสัญพจน์เปิด ลักษณะสำคัญเดี่ยวเป็นการตรวจสอบคุณสมบัติของตัวอย่างโดยใช้เพียงสัญพจน์เดียว เช่น ในลักษณะสำคัญ *F2C1* "not_loaded(A)" เป็นการตรวจสอบว่า ตัวอย่าง A มีลักษณะเป็น not_loaded(A) หรือไม่ ตัวอย่างของสายสัญพจน์ปิด เช่น ลักษณะสำคัญ *F2C4* "opposite_r(A,C), short(C)" เป็นการตรวจสอบคุณสมบัติของตัวอย่าง A ว่ามีคุณสมบัติตรงตามสัญพจน์ทั้งสองหรือไม่ คือ A และ C มีความสัมพันธ์ opposite_r(A,C) กัน และ C ยังมีคุณสมบัติเป็น short(C) อีกด้วย จะเห็นในสายสัญพจน์นี้ได้ว่า ตัวแปร C ถูกสร้างขึ้นจากตัวแปร A ในสัญพจน์ opposite_r(A,C) ทำให้สัญพจน์ short(C) ไม่สามารถใช้เป็นลักษณะสำคัญโดยไม่มีสัญพจน์ที่

สร้างมันขึ้นมา คือ opposite_r(A,C) ประกอบอยู่ได้ ส่วนตัวอย่างของสายสัญพจน์เปิดในกรณีนี้คือ ลักษณะสำคัญ *F2C2* ซึ่งจำเป็นต้องใช้เป็นลักษณะสำคัญข้อหนึ่งด้วยเนื่องจาก opposite_l(B,A) เป็นการตรวจสอบคุณสมบัติว่าตัวอย่าง A มีคุณสมบัติ opposite_l(B,A) หรือไม่

ตัวอย่างนี้เป็นการจำแนกตัวอย่างออกเป็น 3 กลุ่ม ประกอบด้วย mesh(A,1), mesh(A,2) และ mesh(A,3) ตามลำดับ ดังนั้นโครงสร้างของนิวรอลเน็ตเวิร์กจะเป็นตามรูปที่ 3.5 นั่นคือ เน็ตเวิร์กประกอบด้วย 3 ชั้น ชั้นอินพุต ชั้นซ่อน และชั้นเอาต์พุต ชั้นอินพุตประกอบด้วย 9 นิวรอน แต่ละนิวรอนแทนลักษณะสำคัญแต่ละข้อ (*F1C1*, *F2C1*, *F1C2*, *F2C2*, *F1C3*, *F2C3*, *F1C4*, *F2C4* และ *F3C4*) ชั้นซ่อนประกอบด้วย 4 นิวรอน แต่ละนิวรอนแทนกฎแต่ละข้อ (*C1*, *C2*, *C3* และ *C4*) ลักษณะสำคัญที่มาจากกฎข้อเดียวกันจะถูกเชื่อมต่อไปยังนิวรอนซึ่งแทนกฎข้อนั้นๆ เช่น นิวรอนซึ่งแทนลักษณะสำคัญ *F1C1* และ *F2C1* จะถูกเชื่อมต่อไปยังนิวรอนซึ่งแทนกฎ *C1* นิวรอนซึ่งแทนลักษณะสำคัญ *F1C2* และ *F2C2* จะถูกเชื่อมต่อไปยังนิวรอนซึ่งแทนกฎ *C2* เป็นต้น ส่วนที่ชั้นเอาต์พุต ประกอบด้วย 3 นิวรอน แต่ละนิวรอนแทนกลุ่มแต่ละกลุ่ม นิวรอนจากชั้นซ่อนและชั้นเอาต์พุต ถูกเชื่อมต่อแบบทั้งหมด



รูปที่ 3.5 ตัวอย่างโครงสร้างของนิวรอลเน็ตเวิร์ก

การเรียนรู้ของนิวรอลเน็ตเวิร์ก

เมื่อกำหนดโครงสร้างของนิวรอลเน็ตเวิร์กได้แล้ว ขั้นตอนต่อไปคือ การทำการเรียนรู้ของนิวรอลเน็ตเวิร์ก อินพุตเวกเตอร์ (input vector) ที่จะป้อนให้กับนิวรอลเน็ตเวิร์กได้มาจากการนำตัวอย่างที่ใช้เรียนรู้มาเทียบกับ

ลักษณะสำคัญ เพื่อหาค่าความจริงของลักษณะสำคัญแต่ละข้อ โดยทั่วไปแล้วค่าความจริงของลักษณะสำคัญสามารถมีได้หลายแบบขึ้นกับการแทนที่ตัวแปรด้วยค่าคงที่ ในการวิจัยครั้งนี้ได้เลือกการแทนที่ที่ทำให้มีจำนวนลักษณะสำคัญที่มีค่าความจริงเป็นจริงมากที่สุด โดยกำหนดให้ค่าของอินพุตนิวรอนสำหรับลักษณะสำคัญที่มีค่าความจริงเป็นจริงเป็น 1 และลักษณะสำคัญที่มีค่าความจริงเป็นเท็จมีค่าเป็น -1 สำหรับส่วนของเอาต์พุตเวกเตอร์กำหนดให้เอาต์พุตนิวรอนที่แทนกลุ่มของตัวอย่างนั้นมีค่าเท่ากับ 1 นอกจากนั้นให้มีค่าเท่ากับ 0

ตัวอย่างการสร้างอินพุตเวกเตอร์และเอาต์พุตเวกเตอร์

จากตัวอย่างโครงสร้างของนิวรอลเน็ตเวิร์กข้างต้น กำหนดให้กลุ่มความรู้ภูมิหลังที่ใช้สร้างกฎเป็นดังนี้

```
not_important(a1).          not_loaded(a2).
short(a1).                  not_loaded(a3).
short(a2).                  cont_loaded(a2).
neighbour_yz_r(a1,a2).      cont_loaded(a3).
neighbour_zx_r(a1,a2).      opposite_r(a1,a2).
usual(a2).                  opposite_r(a1,a3).
```

เมื่อผ่านกระบวนการดึงลักษณะสำคัญในตัวอย่างข้างต้นแล้ว จะสามารถแปลงกฎแต่ละข้อเป็นดังนี้

```
C1:   mesh(A,1) ←
F1C1:     (not_important(A)),
F2C1:     (not_loaded(A)).

C2:   mesh(A,2) ←
F1C2:     (short(A)),
F2C2:     (opposite_l(B,A)).

C3:   mesh(A,2) ←
F1C3:     (usual(A)),
F2C3:     (neighbour_yz_r(A,B), cont_loaded(B)).

C4:   mesh(A,3) ←
F1C4:     (short(A)),
F2C4:     (neighbour_zx_r(A,B)),
F3C4:     (opposite_r(A,C), short(C)).
```

สมมติว่าใช้ตัวอย่าง `mesh(a1,3)` เป็นตัวอย่างที่ใช้เรียนรู้ เมื่อนำไปเทียบกับลักษณะสำคัญแต่ละข้อจะเป็นดังนี้

```
C1:   mesh(a1,1) ←
F1C1:     (not_important(a1)),                          TRUE
F2C1:     (not_loaded(a1)).                             FALSE

C2:   mesh(a1,2) ←
F1C2:     (short(a1)),                                  TRUE
F2C2:     (opposite_l(a2,a1)).                          FALSE

C3:   mesh(a1,2) ←
F1C3:     (usual(a1)),                                  FALSE
F2C3:     (neighbour_yz_r(a1,a2), cont_loaded(a2)).     TRUE
```

```
C4:    mesh(a1,3) ←
F1C4:      (short(a1)),                              TRUE
F2C4:      (neighbour_zx_r(a1,a2)),                  TRUE
F3C4:      (opposite_r(a1,a2), short(a2)).           TRUE
```

เมื่อนำไปจัดเป็นอินพุตเวกเตอร์ให้กับนิวรอลเน็ตเวิร์กเพื่อทำการเรียนรู้จะได้อินพุตเวกเตอร์เป็น <1, -1, 1, -1, -1, 1, 1, 1, 1> ส่วนของเอาต์พุตเวกเตอร์จะเป็น <0, 0, 1> เนื่องจากตัวอย่างนี้เป็นตัวอย่างของกลุ่ม "3" ซึ่งแทนด้วยเอาต์พุตนิวรอนตัวสุดท้าย ดังนั้นเอาต์พุตนิวรอนตัวสุดท้ายจึงมีค่าเป็น "1" นอกจากนั้นมีค่าเป็น "0" จะสังเกตจากตัวอย่างได้ว่า ในส่วนของลักษณะสำคัญจากกฎข้อ *C4* จะสามารถใช้การแทนที่ได้อีกแบบหนึ่งซึ่งให้ค่าความจริงไม่เหมือนกัน คือ

```
C4:    mesh(a1,3) ←
F1C4:      (short(a1)),                              TRUE
F2C4:      (neighbour_zx_r(a1,a2)),                  TRUE
F3C4:      (opposite_r(a1,a3), short(a3)).           FALSE
```

ลักษณะสำคัญ *F3C4* สามารถใช้ค่าคงที่ a3 แทนตัวแปร C ได้ในสัญพจน์ opposite_r(A,C) ซึ่งทำให้ตัวแปร C ในสัญพจน์ short(C) ต้องถูกแทนที่ด้วยค่าคงที่ a3 ด้วยเช่นกัน ทำให้ลักษณะสำคัญ *F3C4* ที่ถูกแทนที่ด้วยค่าคงที่แล้วเปลี่ยนเป็น (opposite_r(a1,a3), short(a3)) ซึ่งให้ค่าความจริงเป็นเท็จ เมื่อเทียบกับการแทนที่โดยใช้ค่าคงที่ a2 แทนตัวแปร C ในตัวอย่างก่อนนี้ที่ให้ค่าความจริงของลักษณะสำคัญในกฎข้อนี้เป็นจริงทั้งหมด ดังนั้นในตัวอย่างนี้จึงเลือกใช้การแทนที่ตัวแปร C ด้วยค่าคงที่ a2

ตัวอย่างที่ใช้ในกระบวนการเรียนรู้เพื่อสร้างกฎจากระบบไอแอลพี จะถูกนำมาเทียบกับลักษณะสำคัญเพื่อสร้างอินพุตเวกเตอร์และเอาต์พุตเวกเตอร์ดังตัวอย่างข้างต้น แต่ในระบบไอแอลพีโดยทั่วไปกฎที่ถูกสร้างขึ้นจะไม่ครอบคลุมตัวอย่างที่ใช้เรียนรู้ทุกตัว จะมีตัวอย่างบางตัวที่ไม่ตรงพอดีกับกฎข้อใดเลย ดังนั้นในการเลือกตัวอย่างที่ใช้เรียนรู้สำหรับนิวรอลเน็ตเวิร์ก จะมีเงื่อนไขดังต่อไปนี้

1) สำหรับปัญหาที่มีลักษณะเป็นหลายกลุ่ม จะเลือกเฉพาะตัวอย่างที่ถูกครอบคลุมด้วยกฎ หรือตัวอย่างที่ตรงพอดีกับกฎมาเป็นตัวอย่างสำหรับนิวรอลเน็ตเวิร์ก
2) ในปัญหาที่มีลักษณะเป็นสองกลุ่ม ตัวอย่างที่ใช้สำหรับสร้างนิวรอลเน็ตเวิร์กจะเป็นตัวอย่างบวกที่ตรงพอดีกับกฎและกลุ่มตัวอย่างลบ

กระบวนการเรียนรู้ของนิวรอลเน็ตเวิร์กจะเริ่มตั้งแต่ทำการสุ่มน้ำหนัก (weight) ของเส้นเชื่อม และค่าไบแอส (bias) ของโนด จากนั้นทำการเรียนรู้โดยทำการปรับน้ำหนักแบบเพิ่มขึ้น (incremental) โดยทำการปรับน้ำหนักเมื่อป้อนตัวอย่างทีละตัว ด้วยขั้นตอนวิธีแบ็กพรอพาเกชัน และทำการทดสอบการลู่เข้าโดยนับจำนวนตัวอย่างที่ถูกติดกันทั้งหมดให้มากกว่าค่าที่กำหนดไว้ในตอนเริ่มทำการทดลอง นิวรอลเน็ตเวิร์กที่สร้างได้จะถูกนำไปใช้รู้จำตัวอย่างต่อไป