หอสมุดกลาง สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 2

ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

ในบทนี้จะได้กล่าวถึงทฤษฎีและงานวิจัยที่เกี่ยวข้องในงานวิจัยนี้ โดยทฤษฎีที่เกี่ยวข้อง มีเนื้อหาเกี่ยวกับทฤษฎีที่ใช้ในงานวิจัย ซึ่งประกอบด้วย การเรียนรู้ของเครื่อง การโปรแกรมตรรกะเชิงอุปนัย และแบ็กพรอพาเกชันนิวรอลเน็ตเวิร์ก ส่วนงานวิจัยที่เกี่ยวข้องประกอบด้วยงานวิจัยในด้านการรู้จำตัวพิมพ์อักษรไทย งานวิจัยด้านการโปรแกรมตรรกะเชิงอุปนัย งานวิจัยที่เกี่ยวกับแบ็กพรอพาเกชันนิวรอลเน็ตเวิร์ก

2.1 ทฤษฎีที่เกี่ยวข้อง

ทฤษฎีที่เกี่ยวข้องในงานวิจัยนี้ประกอบด้วย การเรียนรู้ของเครื่อง (Machine Learning) การโปรแกรมตรรกะเชิงอุปนัย (Inductive Logic Programming) วิธีการสร้างกฎโดยระบบโปรกอล (Progol) และวิธีการแบ็กพรอพาเกชันนิวรอลเน็ตเวิร์ก (Backpropagation Neural Network)

2.1.1 การเรียนรู้ของเครื่อง (Machine Learning)

การเรียนรู้ของเครื่องตั้งอยู่บนพื้นฐานของการพยายามทำให้โปรแกรมคอมพิวเตอร์ทำการพัฒนาความสามารถของตัวเองโดยอัตโนมัติ โดยใช้ประสบการณ์ที่ได้รับจากภายนอกหรือจากผู้สอน ตัวอย่างเช่น การเรียนรู้จากข้อมูลทางการแพทย์ซึ่งได้รับการเก็บไว้ เพื่อทำการรักษาโรคใหม่ๆ ได้อย่างมีประสิทธิภาพ หรือการสร้างซอฟต์แวร์เพื่อช่วยรับข่าวที่น่าสนใจจากระบบข่าวออนไลน์ โดยเรียนรู้จากความต้องการของผู้ใช้แต่ละคน

แม้ว่ามีการศึกษาถึงเรื่องการเรียนรู้ของเครื่องอย่างแพร่หลาย แต่ในปัจจุบันการเรียนรู้ของเครื่องยังไม่สามารถทำได้ดีเท่ากับการเรียนรู้ของมนุษย์ อย่างไรก็ตามได้มีการคิดค้นและศึกษาเพื่อหาขั้นตอนวิธีใหม่ๆ เพื่อให้สามารถใช้งานกับการเรียนรู้หลายๆ แบบได้อย่างมีประสิทธิภาพ ในด้านการประยุกต์ใช้โปรแกรมหลายโปรแกรมใช้ประโยชน์จากการพัฒนาให้ใช้การเรียนรู้ได้เป็นอย่างดี เช่น มีการพัฒนาโปรแกรมประยุกต์ทางด้านธุรกิจโดยใช้การเรียนรู้ขึ้น หรือการใช้ขั้นตอนวิธีที่มีพื้นฐานจากการเรียนรู้ของเครื่องทำงานในด้านการรู้จำเสียงพูด การทำนายอัตราการฟื้นตัวของผู้ป่วยโรคปอดอักเสบ การสืบค้นกรณีชำระเงินไม่ตรงเวลาของเจ้าของบัตรเครดิต หรือการสร้างกลยุทธ์ต่างๆ ในการเล่นเกม ในทางทฤษฎีก็เช่นกัน ได้มีการศึกษาทางด้านวิธีการเรียนรู้ของเครื่อง เช่น การหาความสัมพันธ์พื้นฐานของตัวอย่างที่ป้อนให้เพื่อทำการเรียนรู้ การหาจำนวนที่เหมาะสมของสมมติฐานที่ควรนำมาพิจารณา การคาดหมายความผิดพลาดของสมมติฐานต่างๆ สำหรับในด้านการศึกษาแขนงใหม่ ได้มีการศึกษาถึงวิธีการเรียนรู้ของเครื่องในด้านต่างๆ มากขึ้น วิธีการดาต้าไมนิ่ง (Data Mining) เป็นวิธีการหนึ่งซึ่งพัฒนาขึ้นจากการเรียนรู้ของเครื่อง ทำการค้นหาความรู้ที่ซ่อนอยู่ภายในระบบฐานข้อมูลที่มีขนาดใหญ่ จะเห็นได้ว่าวิธีการเรียนรู้ของเครื่องได้กลายเป็นปัจจัยสำคัญอย่างยิ่งในการพัฒนาของโปรแกรมคอมพิวเตอร์ในปัจจุบัน

ตัวอย่างของการนำวิธีการเรียนรู้ของเครื่องไปใช้งานได้อย่างมีประสิทธิภาพ

- การเรียนรู้เพื่อทำการรู้จำเสียงพูด
  ระบบที่ทำการรู้จำเสียงพูดที่ประสบความสำเร็จส่วนใหญ่ได้นำวิธีการเรียนรู้ของเครื่องไปใช้ในรูปแบบต่างๆ เช่น ระบบ SPHINX [15] ได้ทำการเรียนรู้เพื่อการทำการรู้จำองค์ประกอบพื้นฐานของเสียงพูด (phonemes) และคำศัพท์ (words) จากสัญญาณเสียงที่ได้ทำการเรียนรู้ไว้ก่อน ใช้นิวรอลเน็ตเวิร์ก [24] และแบบจำลองฮิดเด็นมาคอฟ (Hidden Markov Model) [15] เพื่อทำการรู้จำเสียงของผู้พูดแต่ละคน
- การเรียนรู้เพื่อบังคับพาหนะโดยอัตโนมัติ
  ใช้การเรียนรู้ของเครื่องเพื่อทำการสอนรถยนต์ที่ควบคุมด้วยเครื่องคอมพิวเตอร์ ให้ควบคุมได้อย่างถูกต้องเมื่อขับเคลื่อนอยู่บนถนนหลายแบบ ดังตัวอย่างเช่น ระบบ ALVINN [20] สามารถทำการขับเคลื่อนรถยนต์ด้วยความเร็ว 70 ไมล์ต่อชั่วโมง เป็นระยะทาง 90 ไมล์ บนทางหลวง ในขณะที่มีรถยนต์คันอื่นเคลื่อนอยู่ด้วย
- การเรียนรู้เพื่อทำการจำแนกโครงสร้างใหม่ทางดาราศาสตร์
  มีการนำวิธีการเรียนรู้ของเครื่องไปประยุกต์ใช้เพื่อทำการค้นหาลักษณะพื้นฐานในระบบฐานข้อมูลขนาดใหญ่ เช่น องค์การอวกาศแห่งสหรัฐอเมริกา หรือ NASA ได้ใช้ขั้นตอนวิธีการเรียนรู้ต้นไม้การตัดสินใจ (decision tree learning algorithm) ทำการเรียนรู้เพื่อจำแนกวัตถุที่อยู่บนท้องฟ้าจากภาพที่มีขนาดใหญ่มาก [14]

2.1.2 การโปรแกรมตรรกะเชิงอุปนัย (Inductive Logic Programming)

เป็นวิธีการเรียนรู้ของเครื่องที่ใช้ร่วมกับวิธีการโปรแกรมเชิงตรรกะ (Logic Programming) โดยใช้วิธีการโปรแกรมเชิงตรรกะเพื่ออธิบายตัวอย่างที่ป้อนให้กับระบบเพื่อทำการเรียนรู้ ระบบจะทำการเรียนรู้เพื่อสร้างแนวความคิดในรูปแบบของโปรแกรมเชิงตรรกะจากตัวอย่างและความรู้ภูมิหลัง (Background Knowledge) ที่ผู้สอนป้อนให้ ซึ่งแนวความคิดที่ได้จากระบบนี้สามารถใช้นำไปใช้เพื่อทำนายตัวอย่างที่ยังไม่เคยพบได้

เนื่องจากการโปรแกรมตรรกะเชิงอุปนัยเป็นวิธีการเรียนรู้แบบหนึ่งของเครื่อง ดังนั้นจุดมุ่งหมายของการโปรแกรมตรรกะเชิงอุปนัย จึงคล้ายกับจุดมุ่งหมายของการเรียนรู้ของเครื่อง นั่นคือ การพัฒนาเครื่องมือและวิธีการเพื่อทำการสร้างสมมติฐานจากการสังเกตหรือจากตัวอย่าง และเพื่อสังเคราะห์ความรู้ใหม่จากประสบการณ์ที่ได้รับ แต่การใช้โปรแกรมเชิงตรรกะเป็นเครื่องมือในการอธิบายสมมติฐานและตัวอย่างทำให้การโปรแกรมตรรกะเชิงอุปนัยมีจุดเด่นเหนือวิธีการเรียนรู้ของเครื่องแบบอื่น ดังนี้

1. การใช้โปรแกรมตรรกะเป็นรูปแบบในการอธิบายความรู้ (knowledge) เนื่องจากความรู้ส่วนใหญ่ของมนุษย์มักได้รับการอธิบายในรูปของตรรกะอันดับหนึ่ง (first order logic) หรือเป็นตัวแปรของตรรกะอันดับหนึ่ง (variant of first order logic) แต่ในวิธีการเรียนรู้ของเครื่องส่วนใหญ่มักจะอธิบายความรู้เหล่านี้โดยการใช้ตรรกศาสตร์ประพจน์ (propositional logic) และข้อได้เปรียบสำคัญอีกประการหนึ่งของการใช้โปรแกรมเชิงตรรกะ คือ การใช้โปรแกรมตรรกะอธิบายแนวความคิดที่สังเคราะห์ได้จากตัวอย่าง

สามารถอธิบายแนวความคิดได้ซับซ้อนมากยิ่งขึ้น ซึ่งการอธิบายด้วยตรรกศาสตร์ประพจน์แต่เพียงอย่างเดียว จะไม่สามารถครอบคลุมแนวคิดได้ทั้งหมด

2. สามารถใช้ความรู้ภูมิหลังในกระบวนการเรียนรู้ได้ง่ายกว่าการเรียนรู้ของเครื่องแบบอื่นที่ไม่ใช้การโปรแกรมเชิงตรรกะ

ตัวอย่างต่อไปนี้แสดงให้เห็นถึงข้อได้เปรียบของโปรแกรมตรรกะ ซึ่งสามารถใช้ตัวแทนตรรกะอันดับหนึ่ง (first order representation) เมื่อเทียบกับตัวแทนแบบพรอพอซิชันนอล (propositional representation)

ในการเรียนรู้ของเครื่องเพื่อให้สร้างแนวความคิด *Daughter(x,y)* ซึ่งเป็นแนวความคิดที่ใช้อธิบายความสัมพันธ์ของบุคคล 2 คน คือ x และ y ค่าของ *Daughter(x,y)* จะเป็นจริง เมื่อ x เป็นบุตรสาวของ y นอกเหนือจากนี้จะให้ค่าเป็นเท็จ สมมติให้ข้อมูลของแต่ละคนประกอบด้วย *Name, Mother, Father, Male, Female* ดังนี้ในขั้นตอนการเรียนรู้เพื่อสร้างแนวความคิด *Daughter(x,y)* จะประกอบด้วยข้อมูลของบุคคล 2 คน และค่าของแนวคิดเป้าหมาย *Daughter* ดังตัวอย่างบวกต่อไปนี้ ซึ่งเป็นความสัมพันธ์ระหว่าง *Sharon* กับ *Bob* โดย *Sharon* เป็นลูกสาวของ *Bob*

$(Name_1 = Sharon,\ Mother_1 = Louise,\ Father_1 = Bob,$
$Male_1 = False,\ Female_1 = True,$
$Name_2 = Bob,\ Mother_2 = Nora,\ Father_2 = Victor,$
$Male_2 = True,\ Female_2 = False,\ Daughter_{1,2} = True)$

เมื่อตัวเลขแต่ละตัวแสดงว่าองค์ประกอบนั้นเป็นองค์ประกอบของคนใด หลังจากทำการป้อนตัวอย่างบวกจำนวนหนึ่งให้กับระบบเพื่อทำการเรียนรู้ ในกระบวนการเรียนรู้ของตัวเรียนรู้แบบพรอพอซิชันนอล (propositional learner) เช่น CN2 หรือ C4.5 ผลที่ได้อาจเป็นกฎที่มีขีดจำกัดมากเกินไปเช่น

IF $(Father_1 = Bob) \wedge (Name_2 = Bob) \wedge (Female_1 = True)$
THEN $Daughter_{1,2} = True$

แม้ว่ากฎนี้จะถูกต้องและสามารถนำไปใช้ได้ แต่กฎนี้เป็นกฎที่มีข้อจำกัดมากเกินไป ซึ่งไม่สามารถใช้จำแนกความสัมพันธ์ของคนสองคนอื่นๆ ได้ดี ต่างจากการใช้การแทนอันดับหนึ่ง ที่สามารถทำการเรียนรู้และสร้างแนวความคิดหรือกฎได้ดังนี้

IF Father(y,x) $\wedge$ Female(y), THEN Daughter(x,y)

เมื่อ x และ y เป็นตัวแปรซึ่งสามารถใช้แทนคนใดคนหนึ่งก็ได้

จะเห็นว่ากฎที่ได้เมื่อมีการใช้การแทนอันดับหนึ่ง จะสามารถนำไปใช้จำแนกความสัมพันธ์ของคนอื่นๆ ในอนาคตได้ดีกว่ากฎที่ไม่มีการแทนอันดับหนึ่ง

หลักการสำคัญของการโปรแกรมตรรกะเชิงอุปนัย คือ การสร้างแนวคิด หรือกฎ หรือสมมติฐาน (h) ที่สามารถอธิบายตัวอย่าง (E) ได้ โดยประกอบกับความรู้ภูมิหลัง (B) ดังความสัมพันธ์

$$B \wedge h \vdash E$$

ซึ่งจุดมุ่งหมายสำคัญของวิธีการโปรแกรมตรรกะเชิงอุปนัยคือ การสร้างสมมติฐานที่สามารถอิมพลาย (imply) ตัวอย่างได้ จากกลุ่มตัวอย่างและความรู้ภูมิหลังที่ป้อนให้ ซึ่งมีทฤษฎีที่เกี่ยวข้องในการสร้างสมมติฐานหลายวิธี เช่น วิธีการอนุมานโดยอุปนัย (Inductive Inference) ไออาร์ (Inverse Resolution, IR) อาร์แอลจีจี (Relative Least General Generalization, RLGG) ไออี (Inverse Entailment, IE) ฯลฯ ในที่นี้ได้นำเสนอข้อกำหนดของการสร้างสมมติฐาน คือ วิธีการอนุมานโดยอุปนัย และวิธีการสร้างสมมติฐาน คือ วิธีการไออาร์

2.1.2.1 วิธีการอนุมานโดยอุปนัย (Inductive Inference) [16]

ในการสร้างสมมติฐานขึ้นจากตัวอย่างและความรู้ภูมิหลัง จำเป็นต้องสร้างข้อกำหนดขึ้นดังนี้ "กำหนดให้ D เป็นเซตของตัวอย่างที่ใช้สอนเพื่อทำการเรียนรู้ซึ่งมีรูปแบบ $\langle x_i, f(x_i)\rangle$ โดย x_i คือ ตัวอย่างที่ i และ $f(x_i)$ เป็นค่าที่ต้องการ (target value) และ B คือความรู้ภูมิหลัง คำนิยามของการเรียนรู้คือการสร้างสมมติฐาน h ซึ่งค่า $f(x_i)$ ของแต่ตัวอย่าง x_i แต่ละตัวสามารถอิมพลายได้ด้วยสมมติฐาน h ลักษณะของ x_i และความรู้ภูมิหลัง B"

$$(\forall \langle x_i, f(x_i)\rangle \in D)\ (B \wedge h \wedge x_i) \vdash f(x_i) \qquad (2.1)$$

นิพจน์ $X \vdash Y$ หมายความว่า "Y สามารถอิมพลายได้โดยใช้ X" ซึ่งในนิพจน์ (2.1) หมายความว่า สำหรับทุกตัวอย่าง x_i ค่าของ $f(x_i)$ สามารถอิมพลายได้โดยใช้ความรู้ภูมิหลัง B สมมติฐาน h และตัวอย่าง x_i

ตัวอย่างเช่น ในการสร้างแนวความคิดของความสัมพันธ์ระหว่างคนสองคน $\langle u,v\rangle$ เมื่อ u เป็นบุตรของ v แทนด้วยเพรดิเคต Child(u,v) สมมติให้มีตัวอย่างบวกเพียงตัวอย่างเดียวคือ Child (Bob,Sharon) โดยมีเพรดิเคตที่ใช้อธิบายตัวอย่างนี้คือ Male(Bob), Female(Sharon) และ Father (Sharon,Bob) และมีความรู้ภูมิหลังเป็น Parent(u,v) ← Father(u,v) จะสามารถอธิบายโดยใช้นิพจน์ (2.1) ได้ดังนี้

x_i : *Male(Bob), Female(Sharon), Father(Sharon,Box)*

$F(x_i)$: *Child(Bob,Sharon)*

B : *Parent(u,v) ← Father(u,v)*

ในกรณีนี้สามารถสร้างสมมติฐานได้หลายแบบซึ่งสามารถครอบคลุมตัวอย่างได้ ตามคำนิยามในนิพจน์ (2.1) $(B \wedge h \wedge x_i) \vdash f(x_i)$ ซึ่งในที่นี้จะยกตัวอย่างสมมติฐาน 2 ข้อ ดังนี้

h_1 : *Child(u,v) ← Father(u,v)*

h_2 : *Child(u,v) ← Parent(u,v)*

จะเห็นว่าเพรดิเคตเป้าหมาย *Child(Bob,Sharon)* อยู่ภายใต้เงื่อนไขของ $h_1 \wedge x_i$ โดยไม่ต้องใช้ความรู้ภูมิหลัง B สำหรับในกรณีของสมมติฐาน h_2 อยู่ภายใต้เงื่อนไข $B \wedge h_2 \wedge x_i$ แต่ไม่อยู่ภายใต้เงื่อนไข $h_2 \wedge x_i$ ตัวอย่างนี้แสดงให้เห็นถึงบทบาทสำคัญของความรู้ภูมิหลังในการขยายเซตของสมมติฐานจากตัวอย่าง ดังเช่นในสมมติฐาน h_2 ซึ่งเพรดิเคตในสมมติฐานไม่ได้อยู่ในตัวอย่าง x_i หรือเพรดิเคตที่ใช้อธิบายตัวอย่างนั้น แต่อยู่ในความรู้ภูมิหลัง

ในการค้นหาสมมติฐานตามความสัมพันธ์ $(\forall \langle x_i, f(x_i)\rangle \in D)\ (B \wedge h \wedge x_i) \vdash f(x_i)$ มีลักษณะที่น่าสนใจหลายประการ เช่น

- ความสัมพันธ์นี้สนับสนุนคำนิยามพื้นฐานของการเรียนรู้ คือ การค้นหาแนวความคิดทั่วไปซึ่งสามารถใช้อธิบายเซตของตัวอย่างที่ใช้ในการสอนได้
- เมื่อใช้ความรู้ภูมิหลังประกอบ สามารถสร้างสมมติฐานที่ครอบคลุมตัวอย่างได้มากขึ้น
- โดยการใช้ความรู้ภูมิหลัง สามารถช่วยให้กระบวนการเรียนรู้สามารถค้นหาสมมติฐานได้ง่ายขึ้น

2.1.2.2 วิธีการไออาร์ (Inverse Resolution, IR) [16]

กฎรีโซลูชัน (resolution rule) เป็นวิธีการหนึ่งที่ได้รับความนิยมในการนำมาใช้พิจารณาตรรกะอันดับหนึ่ง (first order logic) ดังนั้นจึงสามารถนำกฎรีโซลูชันมาใช้ในการสร้างตัวดำเนินการไออี (Inverse Entailment Operator) โดยระบบ CIGOL [18] ซึ่งเป็นระบบที่ใช้ในการโปรแกรมตรรกะเชิงอุปนัยวิธีหนึ่งได้นำวิธีการนี้มาใช้

กำหนดให้ L เป็นเพรดิเคตแบบพรอพอซิชันนอล P และ R เป็นอนุประโยคแบบพรอพอซิชันนอล จะได้กฎรีโซลูชัน คือ

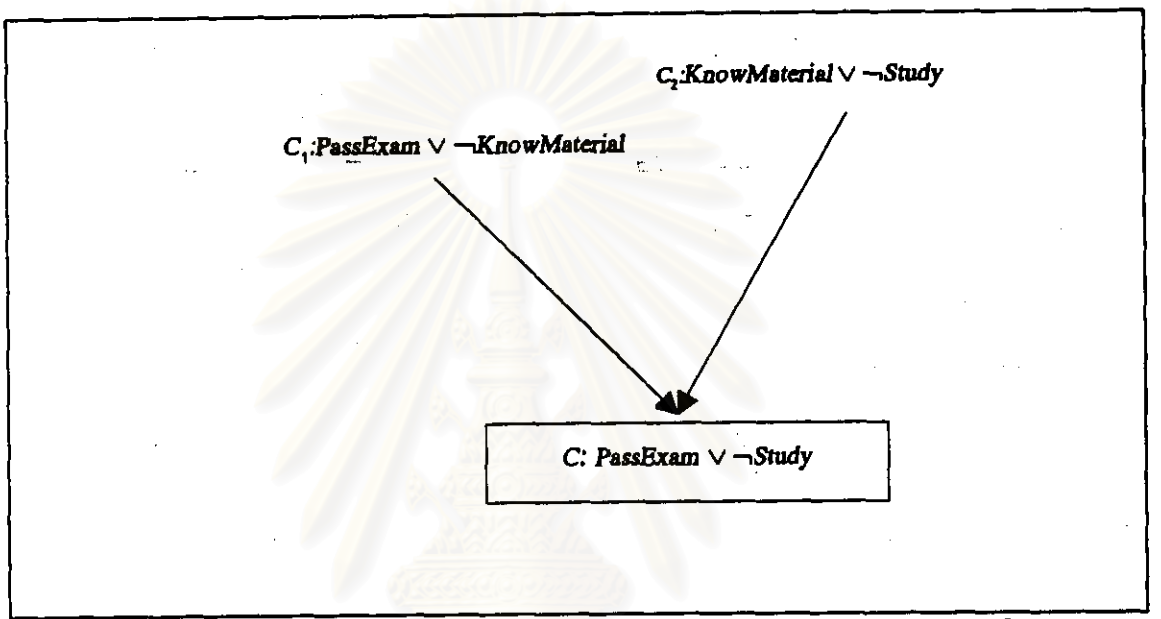
$$\frac{\begin{array}{ll} P & \vee\ L \\ \neg L & \vee\ R \end{array}}{P \vee R}$$

หมายความว่า สามารถสรุปอนุประโยค $P \vee R$ ได้จากอนุประโยค $P \vee L$ และอนุประโยค $\neg L \vee R$ ซึ่งสามารถนำไปใช้ในการสร้างสมมติฐานที่ครอบคลุมตัวอย่างได้ ดังนี้

1. กำหนดอนุประโยค C_1 และ C_2 ให้หาสัญพจน์ L จากอนุประโยค C_1 ซึ่งมีสัญพจน์ $\neg L$ ปรากฏอยู่ในอนุประโยค C_2
2. สร้างอนุประโยคสรุป C โดยรวมทุกสัญพจน์จาก C_1 และ C_2 ยกเว้นสัญพจน์ L และ $\neg L$ ดังสมการ

$$C = (C_1 - \{L\}) \cup (C_2 - \{\neg L\})$$

การใช้กฎรีโซลูชันในการสรุปสมมติฐานจากตัวอย่าง โดยการใช้อนุประโยคที่มีอยู่ C_1 และ C_2 ให้กำหนดสัญพจน์ L ซึ่งเป็นสัญพจน์แบบบวก (positive literal) ในอนุประโยคหนึ่ง และเป็นสัญพจน์แบบลบ (negative literal) ในอนุประโยคที่เหลือ จะสามารถสร้างอนุประโยคสรุป C ได้ ดังรูปที่ 2.1 กำหนดให้มีอนุประโยค C_1 และ C_2 ในขั้นแรกต้องกำหนดสัญพจน์ $L = \neg KnowMaterial$ ซึ่งปรากฏอยู่ในอนุประโยค C_1 และสัญพจน์ที่ตรงข้ามคือ $\neg(\neg KnowMaterial) = KnowMaterial$ ซึ่งปรากฏอยู่ในอนุประโยค C_1 ดังนั้นผลสรุปของสองอนุประโยคนี้ คือ การรวมกันของสัญพจน์ $C_1 - \{L\} = PassExam$ และ $C_2 - \{\neg L\} = \neg Study$



รูปที่ 2.1 แสดงตัวอย่างของการใช้กฎรีโซลูชันในการหาข้อสรุปจากอนุประโยค 2 อนุประโยค

เมื่อสามารถนำกฎรีโซลูชันมาใช้เพื่อหาข้อสรุปจากอนุประโยคได้ ก็เป็นการง่ายที่จะสร้างตัวดำเนินการไออี (inverse entailment operator) $O(C,C_1)$ ซึ่งสามารถทำการอนุมานโดยอุปนัยได้ โดยการสร้างอนุประโยคเริ่มแรก C_2 จากข้อสรุป C และอนุประโยคเริ่มแรกอีกประโยคหนึ่งคือ C_1 ตัวอย่างเช่น มีข้อสรุป $C = A \vee B$ และอนุประโยคเริ่มต้น $C_1 = B \vee D$ จุดมุ่งหมายคือ การพยายามสร้างอนุประโยค C_2 ซึ่ง $C_1 \wedge C_2 \vdash C$

วิธีการ ขั้นแรก จากนิยามของกฎรีโซลูชัน สัญพจน์ใดๆ ที่ปรากฏในอนุประโยค C แต่ไม่ปรากฏในอนุประโยค C_1 จะต้องปรากฏใน C_2 ในตัวอย่างก่อนหน้านี้ ในอนุประโยค C_2 จะต้องมีอนุประโยค A ขั้นที่สอง สัญพจน์ที่ปรากฏใน C_1 แต่ไม่ปรากฏใน C จะต้องเป็นสัญพจน์ที่ถูกดึงออกโดยกฎรีโซลูชัน ดังนั้นนิเสธของสัญพจน์นี้จะต้องปรากฏใน C_2 จากตัวอย่างเดิม อนุประโยค C_2 จึงต้องมีสัญพจน์ $\neg D$ ปรากฏอยู่ด้วย สุดท้ายจะได้สัญพจน์ $C_2 = A \vee \neg D$ ซึ่งสามารถสรุปการดำเนินการไออีได้ดังนี้

1. กำหนดอนุประโยค C_1 และ C ให้หาสัญพจน์ L จากอนุประโยค C_1 แต่ไม่ปรากฏอยู่ในอนุประโยค C

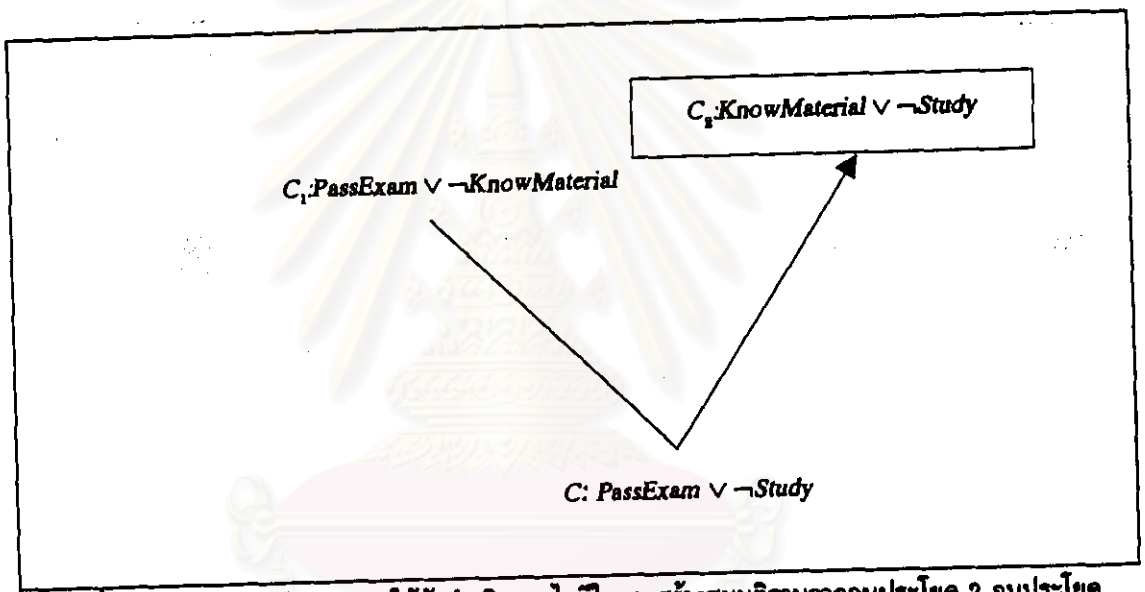
2. สร้างอนุประโยค C_2 ดังสมการ

$$C_2 = (C - (C_1 - \{L\})) \cup \{\neg L\}$$

ตัวอย่างในรูป 2.2 แสดงการใช้ตัวดำเนินการไออี ในการสร้างสมมติฐาน C_2 = *KnowMaterial* $\vee$ $\neg$*Study* จากอนุประโยค C = *PassExam* $\vee$ $\neg$*Study* และอนุประโยค C_1 = *PassExam* $\vee$ $\neg$*KnowMaterial*

2.1.2.2 <u>การรีโซลูชันอันดับหนึ่ง</u> (First-Order Resolution) [16]

ในหัวข้อที่แล้วได้นำเสนอการกระจายในกรณีของพรอพอซิชันนอล ทำการหาอนุประโยคที่สามจากอนุประโยคสองอนุประโยคที่กำหนดให้ เมื่อนำมาประยุกต์ใช้กับตัวแปร เพื่อสร้างการรีโซลูชันอันดับหนึ่ง จะใช้พื้นฐานของตัวแทนรวม (unifying substitution)



รูปที่ 2.2 แสดงตัวอย่างของการใช้ตัวดำเนินการไออีในการสร้างสมมติฐานจากอนุประโยค 2 อนุประโยค

กำหนดตัวแทน θ = $\{x/Bob, y/z\}$ หมายความว่า ใช้ค่า *Bob* แทนตัวแปร x และใช้ตัวแปร z แทนตัวแปร y และใช้เครื่องหมาย $W\theta$ แทนผลการแทนโดยตัวแทน θ ในนิพจน์ W ตัวอย่างเช่น ถ้า L เป็นสัญพจน์ *Father*$(x,Bill)$ และ θ = $\{x/Bob, y/z\}$ จะได้ $W\theta$ = *Father*$(Bob,Bill)$

ตัวแทนรวม θ สำหรับสัญพจน์สองสัญพจน์ คือ L_1 และ L_2 ตัวอย่างเช่น ถ้า L_1 = *Father*(x,y), L_2 = *Father*$(Bill,z)$, และ θ = $\{x/Bill, z/y\}$, แล้ว θ เป็นตัวแทนรวมสำหรับ L_1 และ L_2 เพราะ $L_1\theta = L_2\theta$ = *Father*$(Bill,y)$ ซึ่งกฎรีโซลูชันอันดับหนึ่งจะสร้างอนุประโยคสรุป C ด้วยวิธีการดังนี้

1. หาสัญพจน์ L_1 จากอนุประโยค C_1 สัญพจน์ L_2 จากอนุประโยค C_2 และตัวแทน θ ซึ่งทำให้ $L_1\theta = L_2\theta$
2. สร้างอนุประโยคสรุปโดยรวมทุกสัญพจน์จาก $C_1\theta$ = $C_2\theta$ ยกเว้นสัญพจน์ $L_1\theta$ และ $\neg L_2\theta$ ดังสมการ

$$C = (C_1 - \{L_1\})\theta \cup (C_2 - \{L_2\})\theta$$

2.1.2.3 วิธีการไออาร์ ในกรณีอันดับหนึ่ง (Inverting Resolution: First-Order Case) [16]

จากหัวข้อที่แล้วในการรีโซลูชันอันดับหนึ่งจะได้

$$C = (C_1 - \{L_1\})\theta_1 \cup (C_2 - \{L_2\})\theta_2$$

กำหนดให้ในอนุประโยค C_2 ไม่มีสัญพจน์ที่เหมือนกับสัญพจน์ในอนุประโยค C_1 เราสามารถเปลี่ยนสมการข้างต้นเป็น

$$C - (C_1 - \{L_1\})\theta_1 = (C_2 - \{L_2\})\theta_2$$

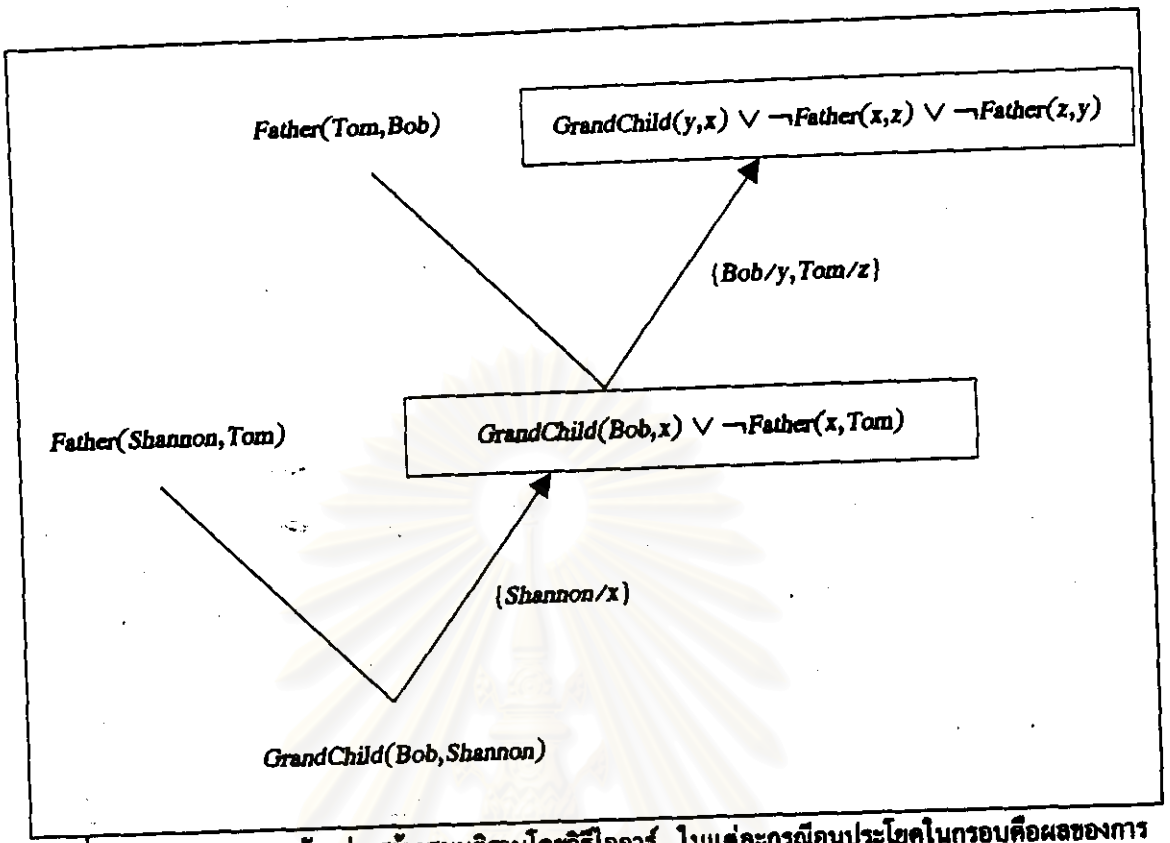
สุดท้ายเราใช้นิยามของกฎการรีโซลูชัน $L_2 = \neg L_1\theta_1\theta_2^{-1}$ เราจะสามารถสรุปอนุประโยค C_2 เป็นกฎไออาร์ได้ดังนี้

$$C_2 = (C - (C_1 - \{L_1\})\theta_1)\theta_2^{-1} \cup \{\neg L_1\theta_1\theta_2^{-1}\}$$

ตัวอย่างในรูปที่ 2.3 ต้องการเรียนรู้กฎสำหรับเพรดิเคตเป้าหมาย *GrandChild*(*y*,*x*) โดยกำหนดข้อมูลที่ใช้ทำการเรียนรู้ D = *GrandChild*(*Bob*,*Shannon*) และข้อมูลซึ่งเป็นความรู้ภูมิหลัง B = {*Father*(*Shannon*,*Tom*), *Father*(*Tom*,*Bob*)} พิจารณาส่วนล่างที่สุดของรูปที่ 2.3 ในที่นี้ให้ข้อสรุป C เป็นตัวอย่างที่ใช้ในการเรียนรู้ *GrandChild*(*Bob*,*Shannon*) และเลือกอนุประโยค C_1 = *Father*(*Shannon*,*Tom*) จากกลุ่มความรู้ภูมิหลังในการใช้ตัวดำเนินการไออาร์สำหรับสัญพจน์ L_1 คือ *Father*(*Shannon*,*Tom*) เลือกตัวแทนย้อนกลับ θ_1^{-1} = {} และ θ_2^{-1} = {*Shannon*/*x*} ในกรณีนี้เราสามารถสร้างอนุประโยค C_2 เป็นการรวมกันของอนุประโยค $(C - (C_1 - \{L_1\})\theta_1)\theta_2^{-1} = (C_1\theta_1)\theta_2^{-1}$ = *GrandChild*(*Bob*,*x*) และอนุประโยค $\{\neg L_1\theta_1\theta_2^{-1}\}$ = ¬*Father*(*x*,*Tom*) ดังนั้นผลลัพธ์ คือ อนุประโยค (*GrandChild*(*Bob*,*x*) ∨ ¬*Father*(*x*,*Tom*)) ซึ่งมีค่าเท่ากับอนุประโยค (*GrandChild*(*Bob*,*x*) ← *Father*(*x*,*Tom*)) ด้วยวิธีเดียวกันนี้ เราสามารถสร้างอนุประโยคสุดท้าย *GrandChild*(*y*,*x*) ← *Father*(*x*,*z*) ∧ *Father*(*z*,*y*) ได้

2.1.3 การสร้างกฎโดยระบบโปรกอล

โปรกอลเป็นระบบที่ใช้วิธีการไออีร่วมกับการค้นหาแบบเจนเนอรัลไปสู่สเปคซิฟิก (general-to-specific search) ในการสร้างคำจำกัดความของกฎ โปรกอลใช้วิธีการไออีสร้างอนุประโยคที่เฉพาะมากที่สุด (most-specific clause) จากนั้นนำอนุประโยคที่ได้มาทำการเจนเนอรัลไรซ์ (generalization) ใช้การค้นหาแบบ A* (A*-like search) โดยใช้ฮิวริสติก (heuristic) เป็นค่าการบีบอัด (compression) เพื่อให้สามารถครอบคลุมตัวอย่างได้มากขึ้น สุดท้ายกฎที่ได้จะเป็นกฎที่มีการบีบอัดสูงสุด ซึ่งสามารถสรุปขั้นตอนที่ระบบโปรกอลใช้ในการสร้างกฎได้ดังนี้



รูปที่ 2.3 แสดงตัวอย่างสร้างสมมติฐานโดยวิธีไออาร์ ในแต่ละกรณีอนุประโยคในกรอบคือผลของการสรุปในแต่ละขั้น C คืออนุประโยคด้านล่าง C_1 คืออนุประโยคด้านซ้าย และ C_2 คืออนุประโยคที่อยู่ในกรอบ โดยในแต่ละขั้น $\theta_1 = \{\}$ และ θ_2^{-1} คือตัวแทนที่อยู่บนลูกศรทางขวา ผลสรุปสุดท้าย คือ $GrandChild(y,x) \leftarrow Father(x,z) \wedge Father(z,y)$

ขั้นตอนที่ 1 เลือกอนุประโยคจากตัวอย่างบวกตัวอย่างแรก

ขั้นตอนที่ 2 สร้างอนุประโยคที่เฉพาะมากที่สุดจากอนุประโยคในขั้นตอนที่ 1 โดยใช้วิธีการอาร์แอลจีจี [17]

ขั้นตอนที่ 3 ทำการเจนเนอรัลไรซ์อนุประโยคในขั้นตอนที่ 2 ใช้การค้นหาแบบ A* โดยใช้ฮิวริสติก (heuristic) เป็นค่าการบีบอัด เพื่อหาอนุประโยคที่มีค่าการบีบอัดสูงสุด

ขั้นตอนที่ 4 ลบตัวอย่างทั้งหมดที่สอดคล้องกับอนุประโยคในขั้นตอนที่ 3

ขั้นตอนที่ 5 ทำซ้ำตั้งแต่ขั้นตอนที่ 1 ถึง 4 จนกว่าไม่เหลือตัวอย่างอีก

การวัดค่าการบีบอัดในการทำเจนเนอรัลไรซ์ใช้การคำนวณดังสมการ 2.2

$$f = p - (n + c + h) \tag{2.2}$$

เมื่อ f - ค่าการบีบอัด

p - จำนวนตัวอย่างบวกที่อนุประโยคนั้นครอบคลุม

n - จำนวนตัวอย่างลบที่อนุประโยคนั้นครอบคลุม

c - จำนวนสัญพจน์ที่ปรากฏอยู่ในส่วนเนื้อความ (body) ของอนุประโยค

h - จำนวนสัญพจน์ที่จำเป็นของความสัมพันธ์ระหว่างอินพุตและเอาต์พุตในส่วนหัว (head) ของอนุประโยค

โดยตัวอย่างแสดงการสร้างกฎโดยระบบโปรกอลแสดงในภาคผนวก ข

2.1.4 แบ็กพรอพาเกชันนิวรอลเน็ตเวิร์ก (Backpropagation Neural Network)

แบ็กพรอพาเกชันนิวรอลเน็ตเวิร์กแบบหลายชั้น (multilayer backpropagation neural network) เป็นขั้นตอนวิธีแบบหนึ่งของนิวรอลเน็ตเวิร์ก ซึ่งมีความสามารถในการสร้างดิชิชันเซอร์เฟสแบบไม่เป็นเชิงเส้น (nonlinear decision surface) ตัวอย่างเช่น ในรูปที่ 2.4 เป็นการรู้จำเสียงพูด โดยทำการจำแนกเสียงสระ 10 เสียง โดยทั้งหมดอยู่ในรูปแบบของคำศัพท์ "h_d" (เช่น "hid", "had", "head", "hood", ฯลฯ) โดยมีสัญญาณอินพุตเป็นตัวเลขพารามิเตอร์ 2 ตัว จากการวิเคราะห์สเปกตรอล (spectral analysis) ของเสียง ทำให้สามารถสร้างดิชิชันเซอร์เฟสได้ใน 2 มิติ ซึ่งสามารถทำการแบ่งแยกตัวอย่างได้ดีกว่าดิชิชันเซอร์เฟสแบบเชิงเส้น (linear decision surface)

แบ็กพรอพาเกชันนิวรอลเน็ตเวิร์กต่างจากเพอร์เซบตรอน (perceptron) ที่องค์ประกอบย่อยของเน็ตเวิร์ก ซึ่งเป็นองค์ประกอบที่สามารถทำงานกับฟังก์ชันแบบไม่เชิงเส้น (nonlinear function) โดยใช้องค์ประกอบซิกมอยด์ (sigmoid unit) ดังรูปที่ 2.5

องค์ประกอบซิกมอยด์จะทำงานเหมือนกับเพอร์เซบตรอนในส่วนที่หาผลคูณของอินพุต จากนั้นนำผลที่ได้ผ่านตัวกรอง (thresbold) แต่ในกรณีขององค์ประกอบซิกมอยด์ เอาต์พุตจะเป็นฟังก์ชันต่อเนื่อง (continuous function) ของอินพุต ดังนี้

$$o = \sigma(\vec{w} \cdot \vec{x})$$

โดย

$$\sigma(y) = \frac{1}{1+e^{-y}}$$

เมื่อ

o - เอาต์พุต

$\vec{x}$ - อินพุต

$\vec{w}$ - ค่าน้ำหนักของอินพุตนั้น ๆ

σ - ฟังก์ชันซิกมอยด์ (sigmoid function) ซึ่งให้ค่าเอาต์พุตระหว่าง 0 และ 1

<u>ขั้นตอนวิธีแบ็กพรอพาเกชันนิวรอลเน็ตเวิร์ก</u> (The BACKPROPAGATION Algorithm) [16]

ขั้นตอนวิธีแบ็กพรอพาเกชันนิวรอลเน็ตเวิร์กจะทำการเรียนรู้เพื่อปรับค่าน้ำหนักสำหรับนิวรอลเน็ตเวิร์กแบบหลายชั้น โดยใช้วิธีการเกรเดียนเดซเซนต์ (gradient descent) เพื่อทำการลดกำลังสองของ

ค่าผิดพลาด (squared error) ระหว่างเอาต์พุตที่ได้จากเน็ตเวิร์กและค่าเป้าหมายของอินพุตเหล่านี้ให้น้อยที่สุด โดยมีขั้นตอนวิธีสำหรับการปรับค่าน้ำหนักในแบบแบ็กพรอพาเกชันดังนี้

กำหนดให้ตัวอย่างที่ใช้ในการเรียนรู้แต่ละตัวอย่างอยู่ในรูป $(\vec{x}, \vec{t})$ เมื่อ $\vec{x}$ เป็นเวกเตอร์ของอินพุตของเน็ตเวิร์ก และ $\vec{t}$ เป็นเวกเตอร์ของเป้าหมายของเอาต์พุตของเน็ตเวิร์ก

η เป็นค่าอัตราการเรียนรู้ (learning rate)

อินพุตขององค์ประกอบ j ซึ่งมาจากองค์ประกอบ i แทนด้วย x_{ji} และค่าน้ำหนักขององค์ประกอบ j ซึ่งมาจากองค์ประกอบ i แทนด้วย w_{ji}

1. สร้างนิวรอลเน็ตเวิร์กตามโครงสร้างที่ต้องการ กำหนดจำนวนนิวรอนของแต่ละชั้น
2. กำหนดค่าน้ำหนักเริ่มต้นแบบสุ่มให้มีค่าน้อย ๆ (เช่น ระหว่าง -0.05 ถึง 0.05)
3. ทำการปรับค่าน้ำหนักด้วยขั้นตอนวิธีดังนี้

- For each $(\vec{x}, \vec{t})$ in *trainning_examples*, Do

Propagate the input forward through the network:

1. Input the instance x to the network and computer the output o_u of every unit u in the network.

Propagate the errors backward through the network:

2. For each network output unit k, calculate its error term δ_k

$$\delta_k \leftarrow o_k(1 - o_k)(t_k - o_k)$$

3. For each hidden unit h, calculate its error term δ_h

$$\delta_h \leftarrow o_h(1 - o_h) \sum_{k \in outputs} w_{kh}\delta_k$$

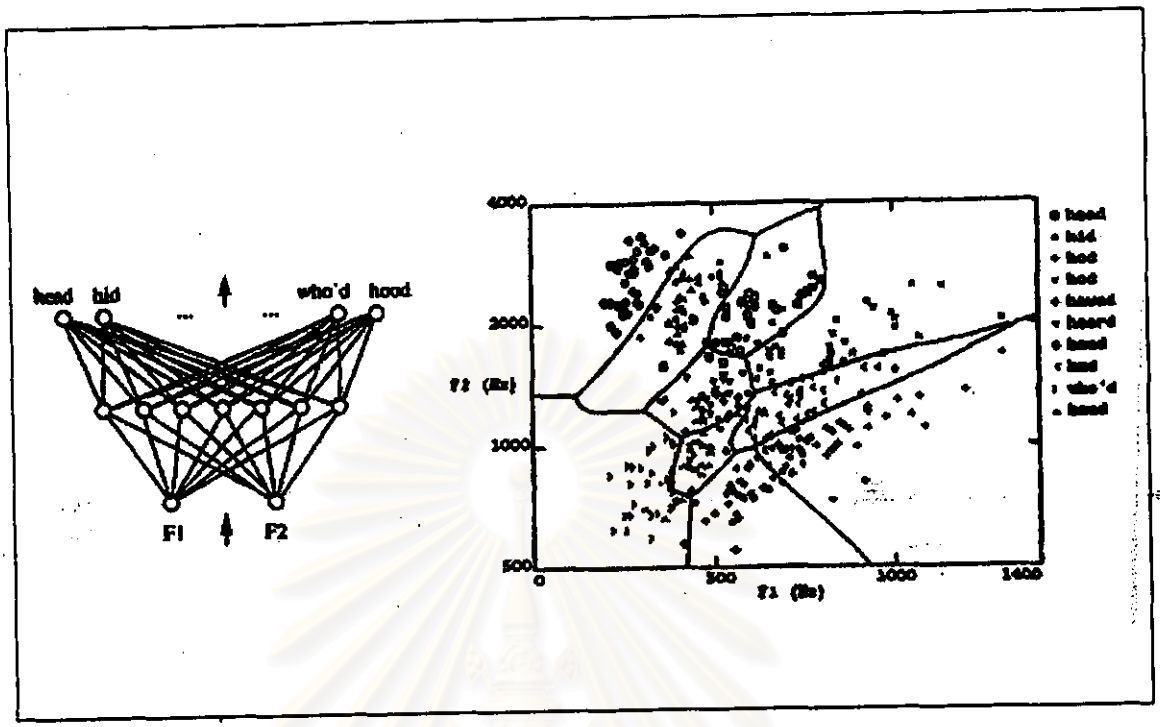
4. Update each network weight w_{ji}

$$w_{ji} \leftarrow w_{ji} + \Delta w_{ji}$$

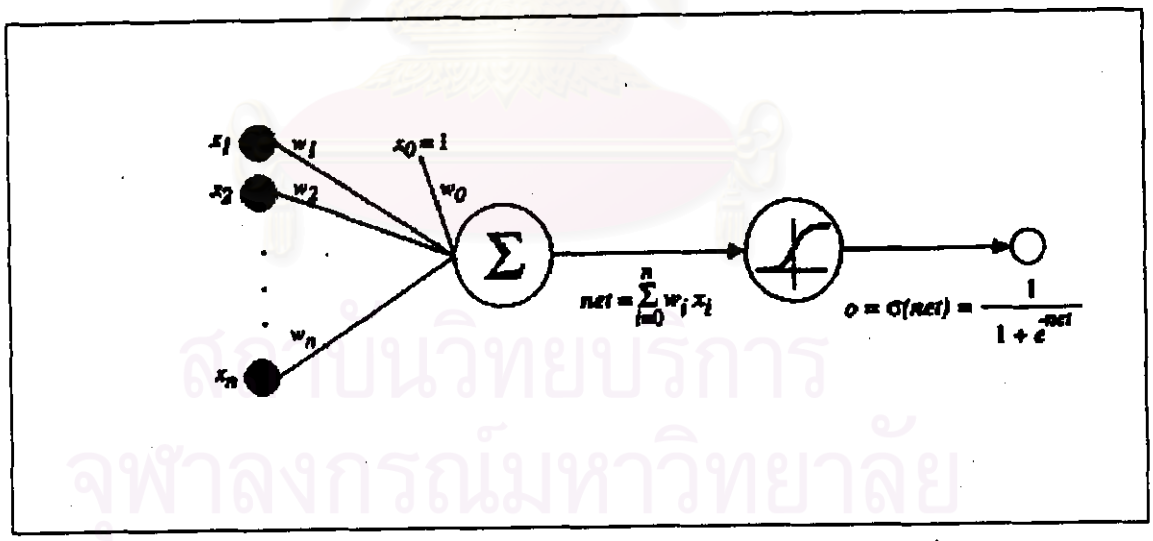
where

$$\Delta w_{ji} = \eta \delta_j x_{ji}$$

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ 2.4 แสดงตัวอย่างการนำนิวรอลเน็ตเวิร์กแบบหลายชั้นไปประยุกต์ใช้งานกับการรู้จำเสียงพูด โดยทำการจำแนกเสียงพูดจากคำศัพท์ที่มีเสียงสระต่างกัน 10 เสียง และอยู่ในรูปของคำศัพท์ "h_d" (เช่น "had", "hid" ฯลฯ) [16]



รูปที่ 2.5 องค์ประกอบซิกมอยด์ (sigmoid unit)

2.2 งานวิจัยที่เกี่ยวข้อง

2.2.1 งานวิจัยทางด้านการรู้จำตัวพิมพ์และตัวเขียนอักษรไทย

2.2.1.1 พิพัฒน์ ทิรัญย์วณิชชากร (2525 และ 2527) [5]

พ.ศ. 2525 เสนอวิธีการรู้จำตัวพิมพ์อักษรไทยโดยการวิเคราะห์เส้นแสดงขอบของอักษร และให้รหัสเพื่อแสดงทิศทางที่เปลี่ยนไปตามลักษณะความโค้งเว้าของเส้นแสดงขอบของอักษร โดยนำหลักพื้นฐานทางคณิตศาสตร์มาใช้เพื่อกำจัดสัญญาณรบกวน แล้วตัดส่วนความโค้งเว้าของเส้นแสดงขอบของอักษรออกเป็นส่วนย่อย จากนั้นจึงนำลักษณะทางเรขาคณิตของส่วนโค้งที่ตัดได้ ได้แก่ ความยาวของเส้นโค้ง ระยะห่างระหว่างจุดเริ่มต้นกับจุดสิ้นสุดของส่วนโค้ง ระยะห่างระหว่างจุดเริ่มต้นกับจุดศูนย์ถ่วง (centroid) ของส่วนโค้ง ระยะห่างระหว่างจุดสิ้นสุดกับจุดศูนย์ถ่วงของส่วนโค้ง และมุมที่เกิดจากการทำมุมกันของเส้นที่ลากจากจุดศูนย์ถ่วงไปยังจุดเริ่มต้นกับเส้นที่ลากจากจุดศูนย์ถ่วงไปยังจุดสิ้นสุดของส่วนโค้ง ลักษณะทางเรขาคณิตเหล่านี้จะนำมาใช้ในการคำนวณค่าความคล้าย (similarity) ระหว่างส่วนโค้งของอักษรที่รับเข้ามากับส่วนโค้งของอักษรต้นแบบ เมื่อได้ค่าความคล้ายระหว่างแต่ละคู่ส่วนโค้งของอักษรแล้ว ก็จะนำไปใช้ในการคำนวณค่าความคล้ายระหว่างอักษร ผลการวิจัยกับตัวพิมพ์อักษรไทย 670 ตัว ซึ่งประกอบด้วยรูปแบบการหมุน 5 แบบ และ 2 ขนาด มีความถูกต้อง 99.7 เปอร์เซ็นต์ สำหรับตัวอักษร 256x256 จุด และมีความถูกต้อง 98.2 เปอร์เซ็นต์ สำหรับตัวอักษร 128x128 จุด

พ.ศ. 2527 เสนอวิธีการรู้จำตัวพิมพ์อักษรไทยโดยการวิเคราะห์เส้นแสดงขอบของอักษร และนำรหัสแบบลูกโซ่ของฟรีแมนกับความแตกต่างของทิศทางของเส้นแสดงขอบของอักษรมาใช้เพื่อกำจัดสัญญาณรบกวน และตัดส่วนความโค้งเว้าของอักษร โดยใช้จุดที่เส้นแสดงขอบของอักษรมีการเปลี่ยนทิศทาง ทำให้ได้ส่วนโค้งย่อยออกมา จากนั้นจะดึงลักษณะทางเรขาคณิตของส่วนโค้งเหล่านั้นออกมา ได้แก่ ความยาวของแต่ละส่วนโค้งย่อย ลักษณะทางเรขาคณิตที่ดึงออกมานี้จะนำมาใช้ในการคำนวณค่าความคล้ายระหว่างส่วนโค้งของอักษรที่รับเข้ามากับส่วนโค้งของอักษรต้นแบบ เมื่อได้ค่าความคล้ายระหว่างแต่ละคู่ส่วนโค้งของอักษรแล้ว ก็จะนำไปใช้ในการคำนวณค่าความคล้ายระหว่างอักษร ผลการวิจัยกับตัวพิมพ์อักษรไทย 345 ตัว ซึ่งมีรูปแบบการหมุน 5 แบบ มีความถูกต้องในการรู้จำ 99.4 เปอร์เซ็นต์

2.2.1.2 ชม กิ้นปาน (2526, 2530 และ 2532) [6]

พ.ศ. 2526 นำวิธีการทับซ้อน (matching method) มาใช้รู้จำตัวพิมพ์อักษรไทยโดยแบ่งเป็น 2 ขั้นตอน ในขั้นตอนของการแบ่งกลุ่มอักษร รูปร่างปลีกย่อยที่ไม่จำเป็นและสัญญาณรบกวนจะถูกกำจัด โดยวิธีการทำอักษรให้มัว (blurring method) วิธีการทำอักษรให้มัวนั้นทำโดยการค้นหาข้อมูลที่มีค่าเป็น 1 ในเมทริกซ์ของอักษรทีละแถว โดยเริ่มจากแถวแรกไปจนถึงแถวสุดท้าย ถ้าพบข้อมูลที่มีค่าเป็น 1 ก็จะเปลี่ยนข้อมูลที่อยู่รอบๆ ตำแหน่งนั้นให้มีค่าเป็น 1 ด้วย อักษรที่มัวแล้ว (blurred characters) จะถูกแบ่งเป็นกลุ่มๆ โดยใช้วิธีวัดค่าสัมประสิทธิ์ความเหมือนของแต่ละรูปแบบ และใช้การกระจายแบบคาร์ฮูเนนโลบหารูปแบบมาตรฐานเพื่อเป็นตัวแทนของแต่ละกลุ่ม ในขั้นตอนของการแยกอักษรจากกลุ่มโดยใช้วิธีทับซ้อนเป็นส่วนๆ (subpattern matching) เพื่อแยกอักษรออกจากกัน ผลที่ได้จากการวิจัยมีความถูกต้อง 98.2 เปอร์เซ็นต์

พ.ศ. 2530 เสนอการรู้จำตัวพิมพ์อักษรไทย โดยใช้การกระจายแบบคาร์ฮูเนนโลบ แบ่งเป็น 2 ขั้นตอน ขั้นตอนแรกเป็นการแบ่งกลุ่มตัวอักษร ไอเกนเวกเตอร์ที่มีค่าไอเกนสูงสุดในการกระจายแบบคาร์ฮูเนนโลบถูกนำมาใช้เป็นรูปแบบมาตรฐานของแต่ละกลุ่มอักษร ขั้นตอนที่สองเป็นการแยกอักษรออกจากกลุ่ม ไอเกนเวนเตอร์ที่ไม่ได้ใช้ในขั้นแรกจะถูกนำมาใช้โดยหาฟังก์ชันการตัดสินใจแบบเชิงเส้นบนระนาบไอเกนเวกเตอร์ที่ได้จากการกระจายแบบคาร์ฮูเนนโลบ เพื่อแยกอักษรออกจากกลุ่ม ผลที่ได้จากการวิจัยมีความถูกต้อง 98 เปอร์เซ็นต์

พ.ศ. 2532 เสนอการรู้จำตัวพิมพ์อักษรไทยโดยใช้วิธีค้นหาจุดเด่นของอักษร (feature concentrated method) ในการเตรียมข้อมูลของอักษร จากนั้นจะเป็นการหาลักษณะเด่นของอักษร โดยการหาคอนเซ็นเทรทเวิร์ด (concentrated word) ณ จุดศูนย์ถ่วง ผลที่ได้จากการวิจัยมีความถูกต้อง 90 เปอร์เซ็นต์

2.2.1.3 ชมทิพ พรพนมชัย (2529) [1]

เสนอวิธีการตรวจรู้อักษรภาษาไทย แบ่งเป็นขั้นตอนดังนี้ ขั้นแรก คือ ขั้นตอนการเปลี่ยนภาพบิตเมทริกซ์ตัวอักษรให้เป็นโครงร่างของอักษร (skeletal form) ขั้นตอนนี้อาจเรียกว่า เป็นขั้นตอนการลดความหนา (thinning process) ของภาพบิตเมทริกซ์ตัวอักษรที่เตรียมไว้ก่อนการนำเข้ามาใช้ในการตรวจรู้อักษร ขั้นตอนที่สอง คือ ขั้นตอนการเปลี่ยนโครงร่างของอักษรให้อยู่ในรูปของรหัส ซึ่งรหัสเหล่านี้จะหมายถึงลักษณะของตัวอักษรตามแนวแถวและแนวสดมภ์ ขั้นตอนที่สาม คือ ขั้นตอนการสร้างความสัมพันธ์ระหว่างรหัสในขั้นที่สองกับอักษรต้นแบบที่สร้างไว้เพื่อการตรวจรู้ โดยความสัมพันธ์นี้จะมีโครงการเป็นรูปต้นไม้ที่เรียกว่า "recognize tree" ผลที่ได้จากการตรวจรู้อักษรภาษาไทยจำนวน 5 แบบพิมพ์ ซึ่งถูกจัดเตรียมไว้ในรูปของภาพบิตเมทริกซ์ตัวอักษร โดยปราศจากสัญญาณรบกวน (noise free) มีความถูกต้อง 70 เปอร์เซ็นต์

2.2.1.4 มนลดา บุญสุวรรณ (2535) [6]

เสนอระบบออฟไลน์สำหรับการรู้จำตัวพิมพ์อักษรไทยหลายรูปแบบ โดยใช้เทคนิคการวิเคราะห์เส้นแสดงขอบของอักษร โดยจะนำรหัสทิศทางแบบลูกโซ่ของฟรีแมนกับความแตกต่างของทิศทางของเส้นแสดงขอบของอักษรมาใช้ในการตัดแบ่งเส้นแสดงขอบของอักษรออกเป็นส่วนโค้งเว้าและส่วนโค้งนูน จากนั้นลักษณะสำคัญ ได้แก่ ความยาวระหว่างจุดบ่งความนูนหรือจุดบ่งความเว้าที่อยู่ติดกัน 2 จุด ภายในส่วนโค้งหนึ่ง ๆ ก็จะถูกนำมาใช้ในขั้นตอนของการเปรียบเทียบแบบการโปรแกรมไดนามิก (dynamic programming) เพื่อหาค่าความแตกต่างระหว่างแต่ละคู่ส่วนโค้งของอักษรที่ต้องการรู้จำกับอักษรต้นแบบ จากนั้นคู่ส่วนโค้งที่มีความคล้ายกันมากที่สุดระหว่างอักษรทั้งสองก็จะถูกตรวจพบได้ และนำมาใช้เป็นคู่ส่วนโค้งเริ่มต้นสำหรับการเปรียบเทียบของคู่ส่วนโค้งอื่น ๆ ที่อยู่ถัดไป จากการใช้วิธีการเปรียบเทียบแบบการโปรแกรมไดนามิก ทำให้สามารถหาค่าความแตกต่างระหว่างแต่ละส่วนของอักษร และความแตกต่างระหว่างอักษร สำหรับนำมาใช้ในการแยกประเภทของอักษรได้ ผลจากการวิจัยโดยใช้รูปแบบของตัวพิมพ์อักษรไทยจำนวน 3 รูปแบบ จำนวนอักษร 1,030 ตัว พบว่ามีความถูกต้อง 94.7 เปอร์เซ็นต์

2.2.1.5 นิธิพัฒน์ ชัชวาลพาณิชย์ (2537) [3]

เสนอระบบออนไลน์สำหรับการรู้จำตัวพิมพ์อักษรไทยและตัวพิมพ์อักษรอังกฤษ โดยแบ่งเป็น 4 ขั้นตอน ดังนี้ ขั้นตอนแรก คือ การวิเคราะห์เส้นแสดงขอบของตัวอักษร โดยใช้การวิเคราะห์จาก

การเปลี่ยนแปลงมุมองศาระหว่างจุดในตัวอักษร ขั้นตอนที่สอง คือ การหาลักษณะสำคัญของตัวอักษร เช่น จำนวนหัวของตัวอักษร จำนวนส่วนโค้ง ความกว้างและความสูงของตัวอักษร ตำแหน่งหัวของตัวอักษร เป็นต้น ขั้นตอนที่สาม คือ การเปรียบเทียบตัวอักษรที่ทำการรู้จำกับตัวอักษรต้นแบบ โดยใช้วิธีการโปรแกรมไดนามิก และใช้ฟังก์ชันการหาค่าผลรวมความแตกต่างสมบูรณ์ ซึ่งเป็นการเปรียบเทียบเพื่อหาค่าความแตกต่างที่น้อยที่สุด ส่วนในขั้นตอนที่สี่ เป็นการนำรหัสของตัวอักษรต้นแบบที่ได้มาเก็บไว้ในแฟ้มข้อมูลตัวอักษร ผลที่ได้จากการทดสอบกับตัวอักษร 3 รูปแบบ มีความถูกต้อง 90.99 เปอร์เซ็นต์

2.2.1.6 สนธยา เมรินทร์(2537) [7]

เสนอการรู้จำตัวพิมพ์อักษรไทยโดยวิธีซินแทกติก วิธีการที่ใช้แบ่งออกเป็น 2 ขั้นตอน คือ การวิเคราะห์โครงร่างของต้นไม้ และการวิเคราะห์ทางฟีเจอร์ (feature) หลังจากที่ข้อมูลภาพถูกทำให้เป็นเวกเตอร์เรียบร้อยแล้ว จะถูกนำไปทำการรู้จำ ขั้นตอนแรกเป็นการจำแนกขั้นต้น ประกอบด้วยการแปลงเวกเตอร์ให้เป็นต้นไม้ของหน่วยสร้างพื้นฐาน (primitive) การวัดค่าระยะระหว่างต้นไม้ของตัวอักษรที่ต้องการรู้จำกับตัวอักษรต้นแบบ โดยเลือกเปรียบเทียบเฉพาะตัวอักษรต้นแบบที่มีหัวของตัวอักษรอยู่ในบริเวณเดียวกับหัวของตัวอักษรที่ต้องการรู้จำเท่านั้น ส่วนขั้นตอนหลังเป็นการจำแนกโดยละเอียด โดยนำเอาลักษณะเด่นของตัวอักษรมาวิเคราะห์ เพื่อเพิ่มความถูกต้องของการรู้จำให้มากขึ้น หากผลการรู้จำไม่อยู่ในเกณฑ์ที่ยอมรับได้ เวกเตอร์ของตัวอักษรจะถูกนำไปปรับปรุงเพื่อตัดส่วนเกินออก หรือต่อส่วนขาดของตัวอักษรเข้าด้วยกัน จากนั้นจึงถูกนำไปรู้จำโดยวิธีเดิมอีกจนกว่าผลการรู้จำจะอยู่ในเกณฑ์ที่น่าพอใจ หรือจนกว่าไม่สามารถปรับปรุงเวกเตอร์ได้อีก จากการทดสอบกับตัวอักษร 966 ตัว ซึ่งมีรูปแบบตัวอักษร 2 แบบ และ 6 ขนาด ผลที่ได้มีความถูกต้อง 97 เปอร์เซ็นต์

2.2.1.7 อภิรักษ์ จิรายุสกุล (2538) [4]

เสนอการนำซีพีเอ็น (Counterpropagation Neural Network, CPN) มาประยุกต์ใช้ในงานตัวอักษรภาษาไทย โดยเปลี่ยนแปลงโครงสร้างหลัก 2 ส่วน ส่วนแรก คือ วิธีการเรียนรู้โดยใช้วิธีการแยกสอนตัวอย่างข้อมูลนำเข้าทีละกลุ่ม ส่วนที่สองคือ โครงสร้างใน "competitive layer" ของซีพีเอ็น สามารถเพิ่มจำนวนโหนดในขั้นตอนการเรียนรู้ ซึ่งช่วยลดเวลาการเรียนรู้และความผิดพลาดในการจดจำตัวอักษร งานวิจัยนี้ได้แสดงให้เห็นถึงปัญหา และวิธีแก้ไขโครงสร้างของซีพีเอ็นเพื่อช่วยในการจดจำตัวอักษรภาษาไทย ผลการทดสอบการจดจำตัวอักษรภาษาไทย 75 ตัว ซึ่งใช้ข้อมูลอย่างละ 15,000 ตัว ที่พิมพ์จากเครื่องเลเซอร์ และจากกระดาษถ่ายเอกสาร ได้ผลความถูกต้องประมาณ 99 เปอร์เซ็นต์ โดยข้อมูลตัวอย่างที่นำมาสอนจำนวน 1,500 ตัว

2.2.1.8 เดชา รัตนาธาร (2538) [2]

เสนอการรู้จำตัวพิมพ์อักษรไทยโดยใช้เทคนิคฟัซซีโลจิก และวิธีซินแทกติก ซึ่งเป็นงานวิจัยที่พัฒนาต่อเนื่องจากงานวิจัยของสนธยา เมรินทร์ (2537) ขบวนการแบ่งออกเป็น 3 ส่วนหลัก คือ ส่วนแรก เป็นการปรับปรุงคุณภาพของข้อมูล โดยการกำจัดสัญญาณรบกวน และการทำตัวอักษรให้บาง โดยใช้เทคนิคเอสพีทีเอ (Save Point Thinning Algorithm, SPTA) ส่วนที่สอง เป็นการแทนค่ารูปแบบ (Pattern Representation) โดยการเข้ารหัสจุดภาพของตัวอักษร แปลงให้เป็นเวกเตอร์ แล้วเปลี่ยนเวกเตอร์ให้เป็นต้นของหน่วยสร้างพื้นฐาน (primitive) ส่วนที่สาม เป็นขั้นตอนการเปรียบเทียบโดยวิธีซินแทกติก ประกอบด้วย

การวิเคราะห์ทางโครงสร้าง และทางรูปร่าง ผลที่ได้ คือ ค่าระยะและค่าความต่าง เพื่อใช้ในการเลือกตัวอักษรต้นแบบที่คล้ายตัวอักษรที่ต้องการรู้จำมากที่สุด ถ้าผลไม่ถึงเกณฑ์ที่ยอมรับได้ จะมีการปรับปรุงเวกเตอร์ และใช้เทคนิคฟัซซีโลจิกเข้ามาช่วยแก้ปัญหา โดยการวัดค่าความเหมือนระหว่างส่วนต่างๆ เพื่อใช้ในการตัดสินผลการรู้จำ จากการทดสอบกับตัวอักษร 2 รูปแบบ แบบละ 6 ขนาด รวม 1,106 ตัวอักษร พบว่ามีอัตราการรู้จำ 99.64 เปอร์เซ็นต์

2.2.1.9 อภิญญา สุพรรณวรรษา (2540) [8]

ใช้การรู้จำตัวพิมพ์อักษรไทยโดยวิธีการโปรแกรมตรรกะเชิงอุปนัย (Inductive Logic Programming) ได้ทำการวิจัยเพื่อเปรียบเทียบผลการทดลองกับวิธีการฟัซซีโลจิก และวิธีซินแทคติก ที่เดชา รัตนาธาร ได้ทดลองไว้ โดยใช้วิธีการปรับปรุงคุณภาพข้อมูลแบบเดียวกัน คือ การกำจัดสัญญาณรบกวน และการทำตัวอักษรให้บาง โดยเทคนิคเอสพีทีเอ และใช้การแทนค่ารูปแบบโดยการเข้ารหัสจุดภาพตัวอักษรแปลงให้เป็นเวกเตอร์ แล้วเปลี่ยนเวกเตอร์ให้เป็นต้นไม้ของหน่วยสร้างพื้นฐาน จากนั้นนำเวกเตอร์พื้นฐานที่ได้มาเป็นตัวอย่างในกระบวนการโปรแกรมตรรกะเชิงอุปนัย ผลการทดลองที่ได้มีอัตราการรู้จำ 87 เปอร์เซ็นต์ จากงานวิจัยนี้พบว่า ปัญหาของการใช้โปรแกรมตรรกะเชิงอุปนัยทำการรู้จำตัวพิมพ์อักษรไทย คือ กรณีที่ไม่สามารถเลือกกฎที่ตรงพอดีในตัวอย่างที่มีสัญญาณรบกวน หรือตัวอย่างใหม่ ซึ่งการโปรแกรมตรรกะเชิงอุปนัยไม่สามารถทำการประมาณเพื่อเลือกกฎที่ใกล้เคียงได้

2.2.1.10 Pisit Phokharatkul และ Chom Kimpan(2541) [19]

ใช้การรู้จำตัวเขียนอักษรไทยโดยการใช้ลักษณะของช่องว่างในตัวอักษรร่วมกับนิวรอลเน็ตเวิร์ก โดยแปลงลักษณะช่องว่างในตัวอักษรออกเป็นแบบต่างๆ แล้วแทนภาพตัวอักษรด้วยลักษณะช่องว่างเหล่านี้ จากนั้นจึงทำการเรียนรู้ด้วยนิวรอลเน็ตเวิร์ก นอกจากนี้ยังมีการใช้เทคนิคพิเศษ เพื่อทำการจำแนกตัวอักษรที่มีลักษณะใกล้เคียงกัน เช่น การพิจารณาลักษณะของส่วนหัวของตัวอักษร ในกรณีของภาพตัวอักษรเป็นภาพอักษร 'ด' หรือ 'ค' ฯลฯ ผลการทดลองที่ได้มีอัตราการรู้จำ 98.3 เปอร์เซ็นต์ เมื่อทำการทดสอบกับภาพตัวอักษร 3,200 ตัว จากผู้เขียนทั้งหมด 40 คน

2.2.1.11 Pitak Thumwarin และ Suphamit Chtayasothon (2541) [23]

ใช้การเข้ารหัสแบบค่าคอนเซนเทรตโคด (concentrated code values) หรือ คิวโคด (Q-codes) เพื่อแทนภาพตัวอักษร จากนั้นใช้ต้นไม้การตัดสินใจ (decision tree) เพื่อทำการจำแนกตัวอักษร และพัฒนาระบบการรู้จำตัวอักษรไทยนี้โดยการใช้การโปรแกรมเชิงวัตถุ (object-oriented programming)

2.2.1.12 Apichart Sajjapong, Wiwat Vatanawood และ Nongluk Covavisaruch (2541) [22]

เป็นงานด้านการรู้จำตัวเขียนอักษรไทยแบบออนไลน์ ทำการรู้จำตัวอักษรซึ่งมีอินพุตจาก digitizing tablet และ pen-like stylus โดยใช้แบ็กพรอพาเกชันนิวรอลเน็ตเวิร์ก ทำการเรียนรู้ลักษณะของจังหวะการเขียนตัวอักษรของผู้ใช้ โดยลักษณะสำคัญที่ใช้เป็นอินพุตเวกเตอร์ของนิวรอลเน็ตเวิร์กประกอบด้วย อัตราส่วนระหว่างความกว้างและความสูงของตัวอักษร รหัสแสดงทิศทางซึ่งใช้เป็นรหัสลูกโซ่ของฟรีแมน รหัสแสดงว่าเป็นจุดเริ่มของตัวอักษรหรือไม่ บริเวณเริ่มต้นขององค์ประกอบพื้นฐาน บริเวณสิ้นสุดขององค์ประกอบ

หอสมุดกลาง สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

พื้นฐาน โดยในการทดลองใช้ตัวอย่างสำหรับทำการเรียนรู้เป็นจำนวนทั้งสิ้น 2,680 ตัวอักษร และใช้ตัวอย่างที่ใช้ทำการทดสอบจำนวน 1,340 ตัวอักษร ผลการทดลองได้เปอร์เซ็นต์ความถูกต้อง 83.43 เปอร์เซ็นต์ ทำการรู้จำผิด 2.16 เปอร์เซ็นต์ และไม่สามารถทำการรู้จำได้ 14.41 เปอร์เซ็นต์

2.2.2 งานวิจัยทางด้านการโปรแกรมตรรกะเชิงอุปนัย

2.2.2.1 Ivan Bratko และ Stephen H. Muggleton (2538) [9]

กล่าวถึงวิธีการโปรแกรมตรรกะเชิงอุปนัย ความสำคัญของความรู้ภูมิหลัง และการนำไปประยุกต์ใช้ในงานด้านต่างๆ เช่น การวิเคราะห์โครงสร้างของเม็ช (mesh) การทำนายค่า การจำแนกทางชีววิทยาของคุณภาพน้ำในแม่น้ำ การสร้างแบบจำลองของโมเลกุลทางชีวภาพ (Biomolecular modelling) การปรับปรุงคุณภาพของข้อมูลในการออกแบบโปรแกรม ฯลฯ เป็นต้น

2.2.2.2 William W. Cohen (2538) [11,12]

ทำการทดลองโดยใช้การระบบการเรียนรู้ในลำดับที่หนึ่ง (first order learning system) ชื่อ FOIL ในงานด้านการจำแนกประเภทข้อความ (Text Categorization) เพื่อเปรียบเทียบวิธีการแบบรีเลชันนอล (relational) และ พรอพอซิชันนอล (propositional) ในการทดลองมีการใช้ความรู้ภูมิหลังเกี่ยวกับตำแหน่งและความสัมพันธ์ของตำแหน่งของคำศัพท์ ผลการทดลองพบว่า ในการจำแนกข้อความจำนวน 51,991 ข้อความเป็น 10 หมวดหมู่ การเรียนรู้แบบรีเลชันนอลให้ผลถูกต้องกว่าแบบพรอพอซิชันนอล 2 หมวดหมู่ และ ทั้งสองแบบให้ผลการทดลองใกล้เคียงกัน 8 หมวดหมู่

2.2.2.3 Mary E. Califf และ Raymond J. Mooney (2540) [10]

นำวิธีการโปรแกรมตรรกะเชิงอุปนัยมาประยุกต์ใช้กับงานด้านการดึงข้อมูลสำคัญจากข้อความภาษาธรรมชาติ (Natural language information extraction) ในการทดลองได้ใช้ตัวอย่างในขั้นตอนการเรียนรู้เป็นประกาศรับสมัครงานคู่กับข้อมูลสำคัญที่มีอยู่ในประกาศนั้น เช่น ลักษณะงาน เงินเดือน สถานที่บริษัทนายจ้าง ฯลฯ เป็นต้น ซึ่งผู้สอนเป็นผู้ป้อนข้อมูลเหล่านี้ให้ จากนั้นจึงทำการทดสอบโดยให้ระบบทำการดึงลักษณะต่างๆ เหล่านี้จากข้อความในประกาศ ผลการทดลองปรากฏว่า ได้ค่าพรีซิชัน (precision) เฉลี่ย 83.7 เปอร์เซ็นต์ และค่ารีคอล (recall) เฉลี่ย 53.1 เปอร์เซ็นต์

2.2.2.4 Sašo Džeroski, Nico Jacobs, Martin Molina, Carlos Moure, Stephen Muggleton และ Wim Van Laer (2541) [13]

ทำการเรียนรู้เพื่อหาลักษณะของทางแยกซึ่งทำให้เกิดอุบัติเหตุทางรถยนต์ โดยใช้ข้อมูลจากเครื่องตรวจจับสัญญาณซึ่งติดตั้งในบริเวณทางแยกต่างๆ โดยตัวอย่างที่ใช้ในการเรียนรู้ประกอบด้วยความเร็วของรถยนต์ ความหนาแน่นของรถยนต์ ลักษณะของถนน ทำการทดลองโดยเปรียบเทียบผลระหว่างวิธีการโปรแกรมตรรกะเชิงอุปนัย 3 วิธี กับวิธีการเรียนรู้แบบพรอพอซิชันนอล ผลการทดลองปรากฏว่า วิธีการโปรแกรมตรรกะเชิงอุปนัยทั้งสามวิธีให้เปอร์เซ็นต์ความถูกต้องสูงกว่าวิธีการเรียนรู้แบบพรอพอซิชันนอล

2.2.2.5 Sam Roberts, Wimi Van Laer, Nico Jacobs, Stephen Muggleton และ Jeremy Broughton (2541) [21]

ทำการทดลองเปรียบเทียบเมื่อใช้วิธีการโปรแกรมตรรกะเชิงอุปนัย และวิธีการเรียนรู้แบบพรอพอซิชันนอล เรียนรู้ข้อมูลตัวอย่างอุบัติเหตุทางรถยนต์ของวัยรุ่นชายที่อยู่ในรูปแบบพรอพอซิชันนอล จากข้อมูลอุบัติเหตุทางรถยนต์ที่เกิดขึ้นทั้งหมด ผลการทดลองปรากฏว่า เปอร์เซ็นต์ความถูกต้องของทุกวิธีการต่ำกว่าเปอร์เซ็นต์ความถูกต้องของวิธีการเลือกโดยคลาสที่มีตัวอย่างมากกว่า (majority class) เพราะว่า จำนวนตัวอย่างบวกมีน้อยกว่าจำนวนตัวอย่างลบมาก จึงได้ทำการทดลองโดยปรับจำนวนตัวอย่างที่ใช้ในการเรียนรู้ให้จำนวนตัวอย่างบวกเท่ากับตัวอย่างลบ ทำให้เปอร์เซ็นต์ความถูกต้องของทุกวิธีสูงกว่าเปอร์เซ็นต์ความถูกต้องของวิธีการเลือกโดยคลาสที่มีตัวอย่างมากกว่า และเปอร์เซ็นต์ความถูกต้องของวิธีการโปรแกรมตรรกะเชิงอุปนัยและวิธีการเรียนรู้แบบพรอพอซิชันนอลมีค่าใกล้เคียงกัน

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย