บทที่ 3

วิธีการรู้จำตัวพิมพ์อักษรไทย

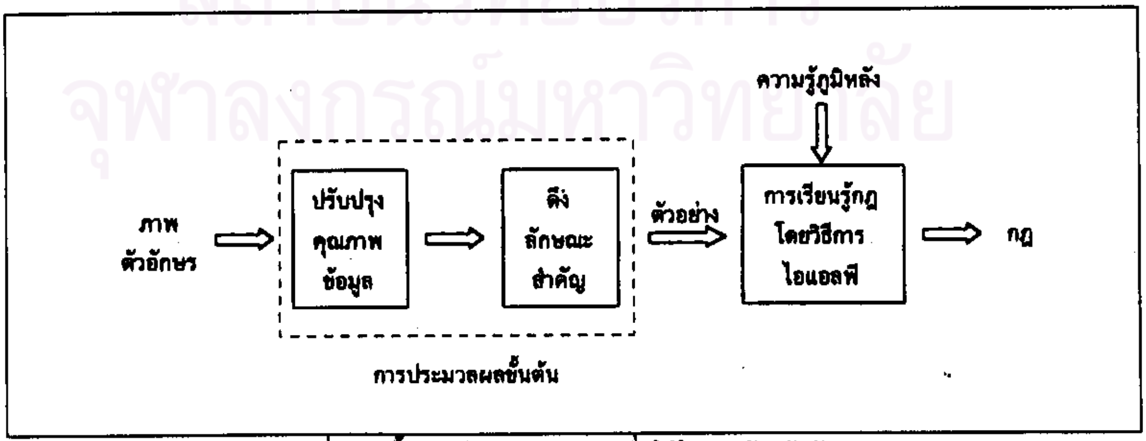
งานวิจัยนี้ใช้การโปรแกรมตรรกะเชิงอุปนัยและแบ็กพรอพาเกชันนิวรอลเน็ตเวิร์กในการรู้จำตัวพิมพ์อักษรไทย ประกอบด้วยขั้นตอนหลายขั้นตอน ซึ่งในบทนี้จะได้กล่าวถึงโครงสร้างของระบบโดยรวมและรายละเอียดของแต่ละขั้นตอน การประมวลผลขั้นต้น คือ การปรับปรุงคุณภาพและการดึงลักษณะสำคัญ (feature extraction) การเรียนรู้กฎโดยวิธีการโปรแกรมตรรกะเชิงอุปนัยหรือไอแอลพี การเรียนรู้เพื่อทำการประมาณกฎที่ใกล้เคียงโดยวิธีการแบ็กพรอพาเกชันนิวรอลเน็ตเวิร์ก และการรู้จำตัวพิมพ์อักษรไทย

3.1 โครงสร้างของระบบ

หลักสำคัญของงานวิจัยนี้ คือ การเรียนรู้กฎโดยใช้วิธีการโปรแกรมตรรกะเชิงอุปนัยจากกลุ่มตัวอย่างซึ่งเป็นลักษณะสำคัญที่ดึงได้จากภาพตัวอักษร แล้วนำกฎนั้นไปทำการรู้จำตัวอย่างอื่น ๆ โดยใช้แบ็กพรอพาเกชันนิวรอลเน็ตเวิร์ก แต่ก่อนนำภาพตัวอักษรผ่านเข้ากระบวนการเรียนรู้ และการรู้จำ จะต้องทำการประมวลผลขั้นต้น (preprocess) ก่อน ซึ่งการขั้นตอนในการประมวลผลขั้นต้นนี้ประกอบด้วยการปรับปรุงคุณภาพข้อมูลและการดึงลักษณะสำคัญ แล้วจึงนำลักษณะสำคัญที่ได้ไปใช้ทำการเรียนรู้ และการรู้จำต่อไป ซึ่งขั้นตอนหลักในงานวิจัยนี้มี 2 ขั้นตอน คือ ขั้นการเรียนรู้ และขั้นการรู้จำ

3.1.1 <u>กระบวนการเรียนรู้</u>

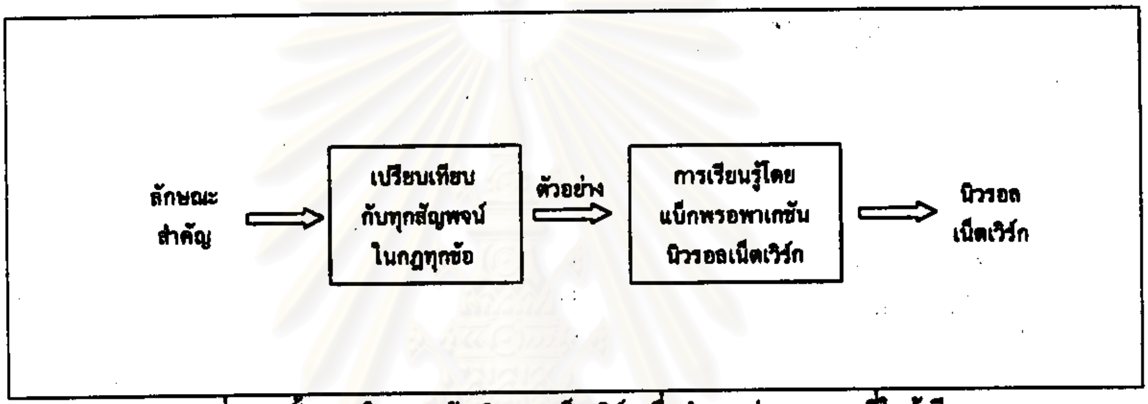
การเรียนรู้ประกอบด้วย 2 ขั้นตอนหลัก คือ ขั้นตอนการเรียนรู้กฎโดยวิธีการโปรแกรมตรรกะเชิงอุปนัย และขั้นตอนการเรียนรู้เพื่อประมาณกฎโดยวิธีการแบ็กพรอพาเกชันนิวรอลเน็ตเวิร์ก โดยลักษณะสำคัญที่ได้จากการประมวลผลขั้นต้น จะถูกใช้เป็นตัวอย่างในการเรียนรู้ร่วมกับความรู้ภูมิหลังเพื่อทำการสร้างเซตของกฎด้วยวิธีการโปรแกรมตรรกะเชิงอุปนัย ดังรูปที่ 3.1



รูปที่ 3.1 ขั้นตอนในการสร้างกฎเพื่อใช้ในการรู้จำตัวอักษร

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

เมื่อทดสอบกฎเหล่านี้ในการรู้จำตัวอักษรพบว่าในกรณีที่ภาพตัวอักษรไม่ชัดเจน หรือภาพที่มีสัญญาณรบกวน (noisy image) วิธีการโปรแกรมตรรกะเชิงอุปนัยไม่สามารถเลือกกฎที่ตรงพอดี (exactly matching) ได้ คือ ไม่มีกฎใดเลยในเซตของกฎทั้งหมดที่ทุกสัญพจน์ในกฎข้อนั้นเป็นจริงสำหรับตัวอักษรนั้น จึงต้องทำการประมาณกฎที่ใกล้เคียง (approximately matching) ในงานวิจัยนี้จึงได้เลือกวิธีการแบ็กพรอพาเกชันนิวรอลเน็ตเวิร์กเพื่อทำการประมาณเพื่อเลือกกฎที่ใกล้เคียงกับภาพตัวอย่างมากที่สุด โดยให้นิวรอลเน็ตเวิร์กทำการเรียนรู้จากตัวอย่างซึ่งเป็นค่าความจริงเมื่อนำลักษณะสำคัญของตัวอักษรไปทดสอบกับทุกสัญพจน์ของทุกกฎในเซต จากนั้นจึงนำนิวรอลเน็ตเวิร์กที่ได้ไปทำการรู้จำภาพตัวอักษรต่อไป ซึ่งในงานวิจัยนี้มีการเปรียบเทียบประสิทธิภาพในการประมาณเพื่อเลือกกฎที่ใกล้เคียงของโครงสร้างที่แตกต่างกัน 2 แบบ ขั้นตอนในการประมาณเพื่อเลือกกฎที่ใกล้เคียงโดยวิธีการแบ็กพรอพาเกชันนิวรอลเน็ตเวิร์กนี้ได้แสดงไว้ในรูปที่ 3.2

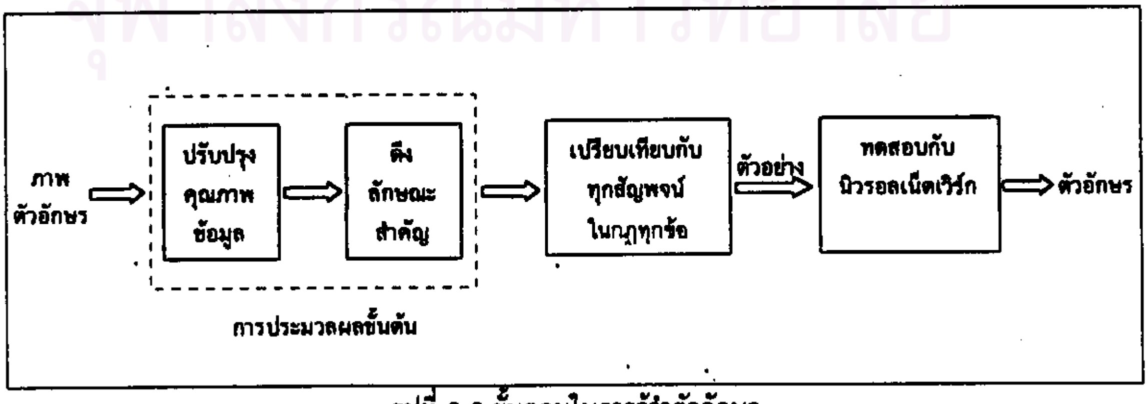


รูปที่ 3.2 ขั้นตอนในการสร้างนิวรอลเน็ตเวิร์กเพื่อทำการประมาณกฎที่ใกล้เคียง

3.1.2 <u>กระบวนการรู้จำ</u>

ขั้นตอนการรู้จำตัวพิมพ์อักษรไทย เมื่อสร้างนิวรอลเน็ตเวิร์กที่สามารถทำการเลือกกฎได้แล้ว เราสามารถนำนิวรอลเน็ตเวิร์กนี้ไปใช้ทำการรู้จำภาพตัวอักษร โดยตัวอย่างที่ใช้ในขั้นตอนนี้ จะผ่านการประมวลผลขั้นต้นก่อน เช่นเดียวกับกระบวนการเรียนรู้ เพื่อให้ได้ลักษณะสำคัญของภาพตัวอักษร แล้วจึงนำลักษณะสำคัญที่ได้นี้ไปทำการทดสอบกับทุกสัญพจน์ของกฎทุกข้อในเซต เพื่อหาค่าความจริงของแต่ละสัญพจน์ ซึ่งค่าความจริงเหล่านี้จะถูกใช้เป็นตัวอย่างในการรู้จำโดยทำการทดสอบกับนิวรอลเน็ตเวิร์กที่ถูกสร้างไว้แล้ว ดังขั้นตอนในรูปที่ 3.3

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ 3.3 ขั้นตอนในการรู้จำตัวอักษร

3.2 การประมวลผลขั้นต้น

จุดประสงค์หลักของการประมวลผลขั้นต้น คือ การหาลักษณะสำคัญของภาพตัวอักษร โดยขั้นตอนในการประมวลผลขั้นต้นนี้ประกอบด้วย การปรับปรุงคุณภาพข้อมูล และการดึงลักษณะสำคัญ เนื่องจากในงานวิจัยชิ้นนี้ทำการทดลองเปรียบเทียบกับงานวิจัยก่อนหน้า คือ "งานวิจัยเรื่องการรู้จำตัวอักษรพิมพ์ภาษาไทยโดยใช้เทคนิคฟัซซีโลจิกและวิธีซินแทกติก" ของ นายเตชา รัตนาธาร [2] และ "งานวิจัยเรื่องการประยุกต์ใช้การโปรแกรมตรรกะเชิงอุปนัยในการรู้จำตัวพิมพ์อักษรไทย" ของนางสาวอภิญญา สุพรรณวรรษา [8] เพื่อให้สามารถทำการเปรียบเทียบได้ จึงได้ใช้วิธีการประมวลผลขั้นต้นแบบเดียวกัน คือ การนำข้อมูลภาพของตัวอักษรมาเข้ากระบวนการปรับปรุงคุณภาพข้อมูล โดยการกำจัดสัญญาณรบกวนและทำตัวอักษรให้บาง เพื่อให้เหลือความกว้างของเส้นเพียง 1 จุดภาพ ได้เป็นลักษณะโครงร่างของตัวอักษร จากนั้นข้อมูลจุดภาพของตัวอักษรจะถูกนำไปวิเคราะห์เพื่อดึงลักษณะสำคัญของตัวอักษร โดยการเข้ารหัสเพื่อหาทิศทางของจุดภาพ แล้วจึงแปลงให้เป็นเวกเตอร์ (vector) แล้วจึงเปลี่ยนเวกเตอร์ให้เหลือเพียงทิศทางของเวกเตอร์เพียงอย่างเดียว เรียกว่า เวกเตอร์พื้นฐาน (primitive) หลังจากที่แทนภาพข้อมูลของตัวอักษรด้วยหน่วยสร้างพื้นฐานแล้ว ข้อมูลเหล่านี้จะนำไปเป็นตัวอย่างให้กับการโปรแกรมตรรกะเชิงอุปนัยเพื่อการเรียนรู้และใช้เป็นข้อมูลเพื่อการรู้จำต่อไป

3.2.1 <u>การปรับปรุงคุณภาพข้อมูล</u> [7]

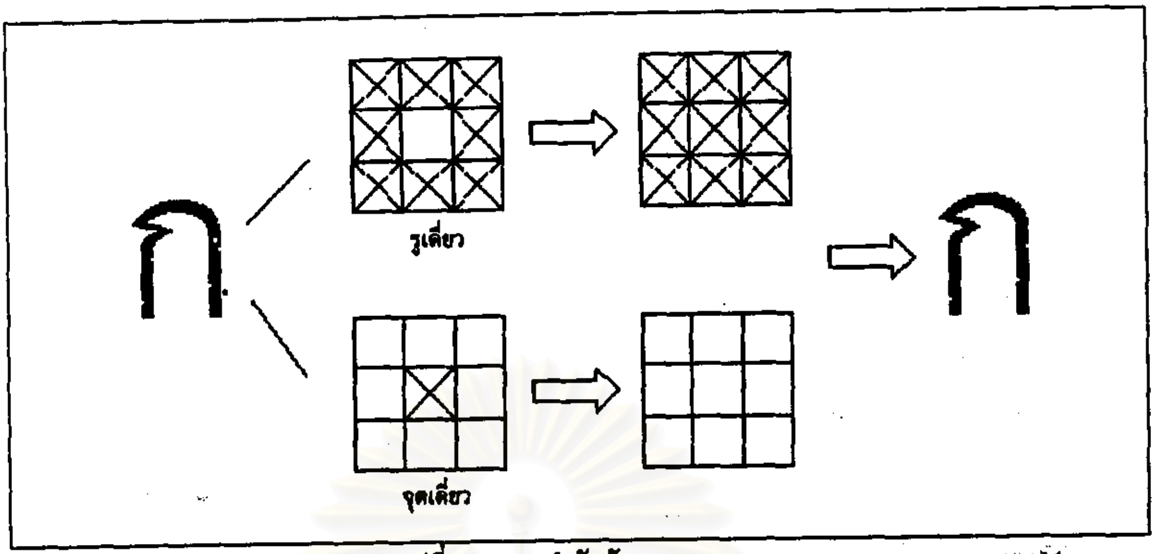
เพื่อให้การเรียนรู้กฎและการรู้จำภาพตัวอักษรมีประสิทธิภาพ ตัวอย่างที่ป้อนให้กับระบบซึ่งก็คือลักษณะสำคัญของภาพตัวอักษรต้องมีความถูกต้องสูง ปัจจัยหนึ่งที่มีความสำคัญอย่างยิ่ง คือ คุณภาพของภาพตัวอักษร จึงจำเป็นต้องมีการปรับปรุงคุณภาพข้อมูลเหล่านี้ โดยมีจุดมุ่งหมายคือ เพื่อหาโครงสร้างของภาพตัวอักษร แล้วนำโครงสร้างนั้นผ่านเข้ากระบวนการดึงลักษณะสำคัญ แล้วจึงนำลักษณะสำคัญที่ได้นี้ผ่านเข้ากระบวนการเรียนรู้ หรือกระบวนการรู้จำ โดยในการปรับปรุงคุณภาพข้อมูล ประกอบด้วยขั้นตอนย่อย 3 ขั้นตอนด้วยกัน คือ

3.2.1.1 <u>การกำจัดสัญญาณรบกวน</u> (Noise reduction)

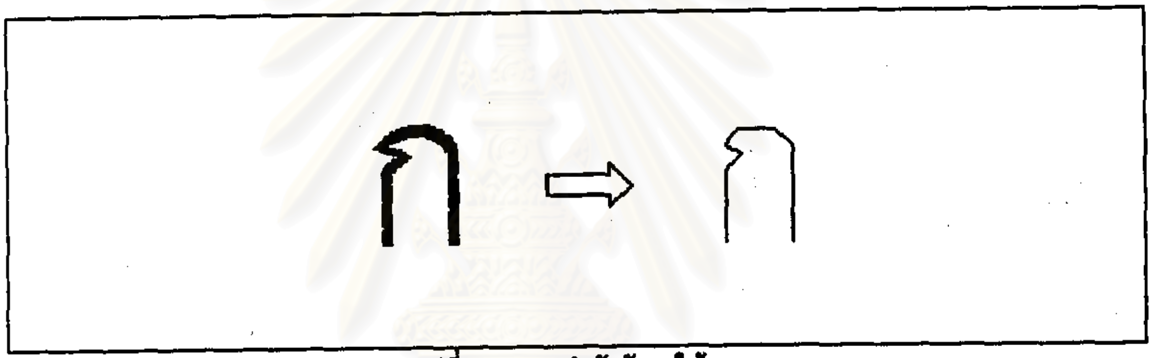
เนื่องจากข้อมูลที่ใช้ในการเรียนรู้และการรู้จำ คือ ภาพของตัวอักษรที่อ่านโดยเครื่องกวาดตรวจ และถูกจัดเก็บเป็นแฟ้มข้อมูลในรูปแบบ BMP แบบขาวดำ ซึ่งภาพตัวอย่างนี้อาจมีสัญญาณรบกวนปะปนอยู่ จึงต้องมีการกำจัดสัญญาณรบกวนก่อน โดยการกำจัดจุดภาพที่เป็นลักษณะจุดเดี่ยวและรูเดี่ยว ดังรูปที่ 3.4

3.2.1.2 <u>การทำตัวอักษรให้บาง</u> (Thinning)

ภาพที่ได้รับการกำจัดสัญญาณรบกวนแล้ว จะถูกส่งเข้าสู่กระบวนการทำตัวอักษรให้บางเพื่อหาโครงร่างของภาพ (skeleton) คือ ข้อมูลของภาพตัวอักษรที่มีความกว้างของเส้น 1 จุดภาพ โดยใช้วิธีการเอสพีทีเอ (Safe Point Thinning Algorithm, SPTA) [2] ดังตัวอย่างในรูปที่ 3.5



รูปที่ 3.4 การกำจัดสัญญาณรบกวน



รูปที่ 3.5 การทำตัวอักษรให้บาง

3.2.1.3 การปรับกรอบของตัวอักษร

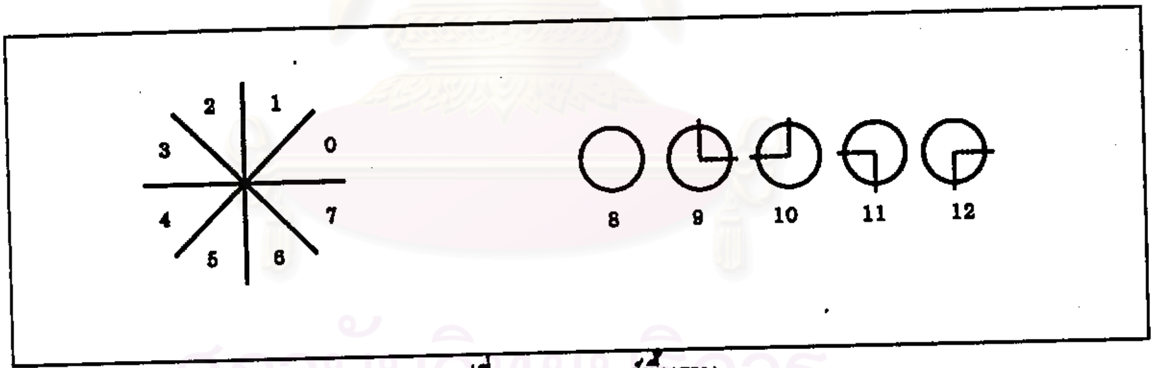
หลังการทำตัวอักษรให้บางแล้ว ภาพตัวอักษรจะเหลือพื้นที่ว่างที่บริเวณขอบมาก จึงต้องทำการปรับขนาดของกรอบภาพให้พอดีกับขนาดของภาพ โดยการตรวจสอบหาจุดภาพจากขอบภาพ จนกระทั่งพบจุดภาพแรกในแนวขอบนั้น ๆ ทำจนครบทั้ง 4 ขอบภาพ โดยเว้นขอบนอกสุดของภาพทั้ง 4 ด้านไว้ด้านละ 1 จุดภาพ เพื่อความสะดวกในการนำข้อมูลไปใช้งาน

3.2.2 การดึงลักษณะสำคัญของตัวอักษร [7]

โครงร่างของภาพที่ได้จากการทำตัวอักษรให้บางจะถูกนำมาวิเคราะห์เพื่อดึงลักษณะสำคัญของภาพตัวอักษรนั้น ซึ่งลักษณะสำคัญที่ใช้ในงานวิจัยนี้ประกอบด้วยลักษณะของเส้นซึ่งถูกแทนด้วยเวกเตอร์พื้นฐาน (primitive vector) อัตราส่วนความกว้างต่อความสูงของภาพตัวอักษร ระดับของตัวอักษร บริเวณที่เป็นรอยต่อของเส้น บริเวณที่มีเส้นหยักขึ้น และบริเวณที่มีเส้นหยักลง ซึ่งรายละเอียดของการดึงลักษณะสำคัญจากภาพตัวอักษร มีดังนี้

3.2.2.1 <u>การแทนโครงร่างของภาพตัวอักษรด้วยเวกเตอร์พื้นฐาน</u>

โครงร่างของภาพจะถูกนำมาแปลงเป็นรหัสเงื่อนไข (condition code) [8] ซึ่งบ่งบอกลักษณะของเส้นนั้นๆ ว่ามีทิศทางอย่างไร เป็นจุดปลายของเส้นหรือไม่ เป็นจุดเชื่อมต่อของเส้นหรือไม่ เส้นมาบรรจบกันเป็นวงกลมหรือไม่ เมื่อได้รหัสเงื่อนไขซึ่งแทนเส้นทุกเส้นในภาพตัวอักษรแล้ว จะนำรหัสเงื่อนไขเหล่านี้มาแปลงเป็นเวกเตอร์พื้นฐาน ซึ่งเวกเตอร์พื้นฐานที่ใช้ในงานวิจัยนี้ประกอบด้วยเวกเตอร์พื้นฐานที่เป็นเส้นตรง 8 รูปแบบ และเวกเตอร์พื้นฐานที่เป็นวงกลมอีก 5 รูปแบบ โดยเวกเตอร์พื้นฐานที่เป็นเส้นตรงจะมีลักษณะเหมือนกับรหัสทิศทางแบบลูกโซ่ของฟรีแมน (freeman chain code) เพื่อใช้แทนเวกเตอร์เส้นตรงที่ทำมุมกับทิศทั้ง 8 โดยที่เวกเตอร์พื้นฐานแต่ละรูปแบบครอบคลุมพื้นที่ 45 องศา ดังแสดงในรูปที่ 3.6 เวกเตอร์พื้นฐานรูปแบบ 0 แทนเวกเตอร์เส้นตรงที่ทำมุมตั้งแต่ 1 ถึง 45 องศา เวกเตอร์พื้นฐานรูปแบบ 1 แทนเวกเตอร์เส้นตรงที่ทำมุมตั้งแต่ 46 ถึง 90 องศา เวกเตอร์พื้นฐานรูปแบบ 2 แทนเวกเตอร์เส้นตรงที่ทำมุมตั้งแต่ 91 ถึง 135 องศา เวกเตอร์พื้นฐานรูปแบบ 3 แทนเวกเตอร์เส้นตรงที่ทำมุมตั้งแต่ 136 ถึง 180 องศา เวกเตอร์พื้นฐานรูปแบบ 4 แทนเวกเตอร์เส้นตรงที่ทำมุมตั้งแต่ 181 ถึง 225 องศา เวกเตอร์พื้นฐานรูปแบบ 5 แทนเวกเตอร์เส้นตรงที่ทำมุมตั้งแต่ 226 ถึง 270 องศา เวกเตอร์พื้นฐานรูปแบบ 6 แทนเวกเตอร์เส้นตรงที่ทำมุมตั้งแต่ 271 ถึง 315 องศา เวกเตอร์พื้นฐานรูปแบบ 7 แทนเวกเตอร์เส้นตรงที่ทำมุมตั้งแต่ 316 ถึง 360 องศา ส่วนเวกเตอร์พื้นฐานที่เป็นวงกลม จะถือเอาบริเวณของจุดเชื่อมต่อจากวงกลมเป็นตัวแบ่งรูปแบบ โดยเวกเตอร์พื้นฐานหมายเลข 8 จะเป็นวงกลมที่ไม่มีจุดเชื่อมต่อกับเส้นตรงอื่นๆ เลย ส่วนเวกเตอร์วงกลมที่เหลืออีก 4 รูปแบบ นั่นคือเวกเตอร์วงกลมที่มีจุดเชื่อมต่อในบริเวณจตุภาคที่ 1, 2, 3 และ 4 ตามลำดับ ดังแสดงในรูปที่ 3.6



รูปที่ 3.6 เวกเตอร์พื้นฐาน

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

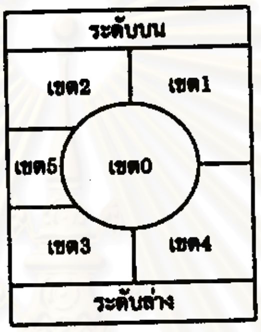
3.2.2.2 <u>การแบ่งระดับและแบ่งเขตย่อยของตัวอักษร</u>

การแบ่งระดับเป็นการจัดกลุ่มของตัวอักษรไทยอย่างง่ายก่อนโดยพิจารณาจากตำแหน่งในแนวตั้งของตัวอักษรนั้น ซึ่งการจัดระดับในงานวิจัยนี้แบ่งเป็น 5 ระดับ คือ

- ระดับที่ 1 เป็นตัวอักษรระดับบน เช่น สระอิ, สระอี และวรรณยุกต์ต่างๆ เป็นต้น
- ระดับที่ 2 เป็นตัวอักษรระดับกลางและมีบางส่วนอยู่ในระดับบน เช่น ป, ไ, โ เป็นต้น
- ระดับที่ 3 เป็นตัวอักษรระดับกลาง เช่น ก, ข, ฃ เป็นต้น

- ระดับที่ 4 เป็นตัวอักษรระดับกลางและมีบางส่วนอยู่ในระดับล่าง เช่น ฎ, ฤ เป็นต้น
- ระดับที่ 5 เป็นตัวอักษรระดับล่าง เช่น สระอุ, สระอู เป็นต้น

นอกจากการจัดกลุ่มของตัวอักษรไทยด้วยการจัดระดับ สิ่งหนึ่งที่สามารถช่วยในการรู้จำตัวอักษรได้ก็คือ บริเวณต่างๆ เพื่อทำการวิจัยเปรียบเทียบกับวิธีการพัชซีโลจิกและวิธีการซินแทกติก [2] และการโปรแกรมตรรกะเชิงอุปนัย [8] ในงานวิจัยนี้จึงใช้การแบ่งบริเวณเหมือนกับในงานวิจัยทั้งสอง โดยแบ่งออกเป็น 7 บริเวณด้วยกัน ดังรูปที่ 3.7



รูปที่ 3.7 เขตย่อยของตัวอักษร

บริเวณทั้ง 7 บริเวณนี้ จะถูกนำไปใช้จำแนกลักษณะของเวกเตอร์พื้นฐานว่ามีจุดเริ่มต้นและจุดสิ้นสุดอยู่ในบริเวณใด มีรอยหยัก หรือจุดเชื่อมต่ออยู่บริเวณใด ซึ่งลักษณะสำคัญของตัวอักษรนี้จะถูกนำไปใช้ในการทดลองต่อไป

3.3 การเรียนรู้กฎโดยวิธีการโปรแกรมตรรกะเชิงอุปนัย

ลักษณะสำคัญของตัวอักษรที่ได้จากกระบวนการประมวลผลขั้นต้น จะถูกใช้เป็นตัวอย่างในการเรียนรู้ของวิธีการโปรแกรมตรรกะเชิงอุปนัย โดยในงานวิจัยนี้เลือกระบบโปรกอล (Progol) [15] ทำหน้าที่ในส่วนของกระบวนการโปรแกรมตรรกะเชิงอุปนัย สร้างกฎจากตัวอย่างบวก (positive example) ตัวอย่างลบ (negative example) และความรู้ภูมิหลัง (background knowledge) โดยที่ทั้งสามส่วนนี้อยู่ในรูปของการโปรแกรมเชิงตรรกะ

3.3.1 <u>ตัวอย่างที่ใช้ในการเรียนรู้กฎ</u>

ตัวอย่างที่ใช้ในกระบวนการเรียนรู้กฎโดยระบบไอแอลพี คือ ลักษณะสำคัญที่ดึงได้จากภาพตัวอักษร ซึ่งประกอบด้วย

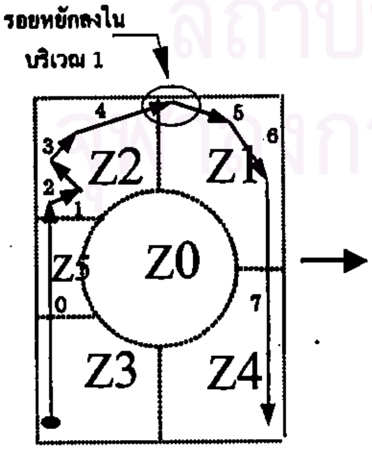
- ระดับของตัวอักษร มีค่าตั้งแต่ 1 ถึง 5
- อัตราส่วนระหว่างความกว้างต่อความสูงของภาพตัวอักษร มีค่าเป็นจำนวนจริง
- ลิสต์ของเวกเตอร์พื้นฐานที่ประกอบอยู่ในภาพตัวอักษรนั้น
- ลิสต์ของบริเวณที่มีจุดเชื่อมต่อของเส้น
- ลิสต์ของบริเวณที่รอยหยักขึ้นและรอยหยักลง

นอกจากรูปแบบทั้ง 13 แบบของเวกเตอร์พื้นฐานแล้ว ลักษณะอื่นที่ถูกเก็บเพื่อทำการรู้จำตัวอักษรยังมีสถานะของเวกเตอร์นั้นว่าเป็นจุดปลายหรือไม่ บริเวณเริ่มต้น และบริเวณสิ้นสุดของเวกเตอร์นั้น นั่นคือ ในเวกเตอร์พื้นฐานหนึ่ง ๆ จะมีข้อมูลดังนี้

- รูปแบบของเวกเตอร์พื้นฐาน ซึ่งแสดงลักษณะและทิศทางของเวกเตอร์นั้น โดยมีเวกเตอร์เส้นตรง 8 รูปแบบ แทนด้วยตัวเลขตั้งแต่ 0 ถึง 7 และเวกเตอร์วงกลม 5 รูปแบบ แทนด้วยตัวเลขตั้งแต่ 8-12 ดังที่เสนอไว้แล้วในหัวข้อ 3.2.2.1
- สถานะแสดงว่าเวกเตอร์นี้เป็นจุดปลายของภาพหรือไม่ โดยใช้ตัวเลข 0 และ -1 แทนค่าสถานะนี้ โดยให้ 0 แทนสถานะไม่เป็นจุดปลาย และ 1 แทนสถานะเป็นจุดปลาย
- บริเวณเริ่มต้นของเวกเตอร์ แทนด้วยตัวเลขตั้งแต่ 0 ถึง 7
- บริเวณสิ้นสุดของเวกเตอร์ แทนด้วยตัวเลขตั้งแต่ 0 ถึง 7

เพื่อความสะดวกในการแทนลักษณะของเวกเตอร์เหล่านี้เป็นลิสต์ จึงใช้ตัวเลขตั้งแต่ 1 ถึง 1664 ซึ่งสามารถแทนลักษณะทั้งหมดที่แตกต่างกันของเวกเตอร์พื้นฐานเหล่านี้ได้ (13 x 2 x 8 x 8 = 1664 รูปแบบ) โดยมีวิธีการคำนวณดังสมการ 3.1 ตัวอย่างเช่น ใช้เลข 1 แทน (0, -1, 0, 0), เลข 2 แทน (0, -1, 0, 1), เลข 1663 แทน (12, 0, 7, 6) และ เลข 1664 แทน (12, 0, 7, 7) เป็นต้น ดังตัวอย่างการแทนโครงร่างของภาพด้วยเวกเตอร์พื้นฐาน ในรูปที่ 3.8

$$\text{number} = (\text{type} \times 128) + ((\text{endpoint} + 1) \times 64) + (\text{start_zone} \times 8) + \text{endzone} + 1 \qquad (3.1)$$



Vector No : (vectorType, endPoint, startingZone, endingZone)	
0	(1, -1, z3, z2)
1	(0, 0, z2, z2)
2	(3, 0, z2, z2)
3	(1, 0, z2, z2)
4	(0, 0, z2, z1)
5	(7, 0, z1, z1)
6	(7, 0, z1, z1)
7	(6, -1, z1, z4)

→

Vector No : VectorID	
0	155
1	83
2	467
3	211
4	82
5	970
6	970
7	781

รูปที่ 3.8 การแทนภาพโครงร่างตัวอักษรด้วยเวกเตอร์พื้นฐาน

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

นอกจากเวกเตอร์พื้นฐานจะถูกใช้แทนภาพตัวอักษรแล้วยังมีระดับของตัวอักษร อัตราส่วนระหว่างความกว้างและความสูง บริเวณที่เป็นจุดเชื่อมต่อของเส้น บริเวณที่รอยหยักลง และบริเวณที่เป็นรอยหยักขึ้น ประกอบกันเป็นตัวอย่างที่ป้อนให้กับระบบโปรกอลเพื่อทำการเรียนรู้กฎ โดยมีรูปแบบเป็นดังรูปที่ 3.9

char(A,B,C,D,E,F)

char	-	ตัวอักษร
A	-	ระดับของตัวอักษร
B	-	อัตราส่วนระหว่างความกว้างและความสูง
C	-	ลิสต์ของเวกเตอร์พื้นฐาน
D	-	ลิสต์ของบริเวณที่มีจุดร่วมของเส้น
E	-	ลิสต์ของบริเวณที่มีรอยหยักลง
F	-	ลิสต์ของบริเวณที่มีรอยหยักขึ้น

รูปที่ 3.9 รูปแบบที่ใช้แทนภาพตัวอักษร

ในการทดลองนี้ ระบบโปรกอลจะทำหน้าที่ในการสร้างกฎสำหรับตัวอักษรครั้งละตัว โดยกำหนดตัวอย่างบวกสำหรับการสร้างกฎของตัวอักษรใด คือ ลักษณะสำคัญของภาพตัวอักษรนั้น และตัวอย่างลบสำหรับการสร้างกฎของตัวอักษรใด คือ ลักษณะสำคัญของภาพตัวอักษรอื่นๆ โดยในการทดลองนี้มีตัวอักษรทั้งสิ้น 77 ตัว (‘ก’, ‘ข’, ‘ฃ’, ... , ‘๙’) จากรูปแบบตัวอักษร 2 รูปแบบ คือ คอร์เดีย (Cordia) และยูโครเซีย (Eucrosia) ขนาดตัวอักษร 7 ขนาด คือ 20, 22, 24, 28, 32, 36, 48 จุด รวมเป็นจำนวนตัวอย่างทั้งหมด 77 x 2 x 7 = 1078 ตัวอย่างในการเรียนรู้กฎ เช่น ตัวอย่างบวกในการเรียนรู้กฎสำหรับตัวอักษร ‘ก’ คือ ลักษณะสำคัญของภาพตัวอักษร ‘ก’ ทั้งหมด 14 ตัว และตัวอย่างลบ คือ ลักษณะสำคัญของภาพตัวอักษรอื่นๆ (‘ข’, ‘ฃ’, ... , ‘๙’) เป็นจำนวน 1064 ตัวอย่าง ก็จะได้กฎสำหรับตัวอักษร ‘ก’ หรือ ตัวอย่างบวกในการเรียนรู้กฎสำหรับตัวอักษร ‘ข’ คือ ลักษณะสำคัญของภาพตัวอักษร ‘ข’ และตัวอย่างลบ คือ ลักษณะสำคัญของภาพตัวอักษรอื่นๆ (‘ก’, ‘ฃ’, ... , ‘๙’) ที่เหลือ ดังแสดงในรูปที่ 3.10

3.3.2 <u>ความรู้ภูมิหลัง</u>

ความรู้ภูมิหลังที่ใช้สำหรับวิธีการโปรแกรมตรรกะเชิงอุปนัย คือ กลุ่มของความรู้ของผู้ใช้ที่สามารถจำแนกตัวอย่างได้ ซึ่งกลุ่มของสัญพจน์ซึ่งเป็นความรู้ภูมิหลังเหล่านี้ จะถูกนำไปใช้เพื่อสร้างคำจำกัดความของกฎ การให้ความรู้ภูมิหลังที่ดีและเหมาะสมแก่ระบบจะทำให้ระบบสามารถสร้างกฎที่ดีได้ ซึ่งกลุ่มของความรู้ภูมิหลังนี้ก็ได้รับการอธิบายโดยใช้โปรแกรมเชิงตรรกะ เช่นเดียวกับตัวอย่างบวก และตัวอย่างลบ โดยความรู้ภูมิหลังในการทดลองเพื่อทำการจำแนกภาพตัวพิมพ์อักษรไทยในงานวิจัยนี้ คือ ความรู้ของผู้ใช้ที่สามารถจำแนกความแตกต่างของกลุ่มตัวอย่างได้ ในงานวิจัยนี้ได้ใช้ความรู้ภูมิหลังทั้งหมด 55 โปรแกรม โดยแบ่งออกเป็น 3 กลุ่มใหญ่ คือ

3.3.2.1 <u>การพิจารณาลักษณะของตัวอักษร</u> เช่น อัตราส่วนความกว้างแต่ความสูงของตัวอักษร ลักษณะส่วนหัวของตัวอักษร ลักษณะส่วนปลายของตัวอักษร ฯลฯ

ตัวอย่างบวก

ก(3,0.78,[155,83,467,211,82,970,970,781],[],[],[z1]).

ก(3,0.76,[155,339,339,211,83,83,978,970,781],[],[],[z2]).

ก(3,0.72,[155,467,467,211,211,83,82,970,842,845,805,19],[z2],[z2],[z1]).

ก(3,0.75,[155,467,211,83,83,978,842,842,714,653,19],[z2],[z2],[z2]).

ก(3,0.75,[155,467,211,83,82,970,845,677,19],[z2],[z2],[z1]).

ก(3,0.78,[155,467,211,83,82,970,970,781,83,979,19],[z2],[z2,z2],[z2,z1]).

ก(3,0.74,[155,339,211,83,83,978,781,19],[z2],[z2],[z2]).

ตัวอย่างลบ

ข(3,0.78,[162,101,989,860,708,705,841,842,978,979,83,1171],[],[z4],[z2]).

ข(3,0.76,[162,101,989,860,708,705,713,978,979,83,211,1171],[],[z4],[z2]).

ข(3,0.75,[162,101,93,988,860,708,705,713,842,978,979,83,211,1171],[],[],[z2]).

ข(3,0.78,[162,93,860,708,705,713,850,979,83,83,1171],[],[],[z2]).

ข(3,0.73,[138,458,202,226,93,988,708,705,849,979,83,83,1171],[],[z3],[z2]).

ข(3,0.76,[162,101,989,860,716,850,979,83,83,1171],[],[z4],[z2]).

ข(3,0.73,[162,101,989,860,708,705,849,979,211,1171],[],[z4],[z2]).

ข(3,0.73,[162,101,989,836,705,713,842,978,83,83,979,83,339,1299],[],[z4,z2],
[z2,z2]).

ข(3,0.77,[162,101,989,836,705,705,841,978,83,979,211,339,1299],[],[z4,z2],
[z2,z2]).

⋮

ฮ(3,0.95,[266,202,74,962,65,773,961,977,83,83,235,350,476,476,1436],[z0],[z0],
[z0,z2]).

ฮ(3,0.89,[138,74,970,66,129,65,773,977,83,235,222,348,476,1436],[z0,z0],[z1,z0],
[z2]).

รูปที่ 3.10 แสดงตัวอย่างบวกและตัวอย่างลบในการเรียนรู้กฎของตัวอักษร *'ก'*

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

3.3.2.2 <u>การพิจารณาตำแหน่งของส่วนต่างๆ ของตัวอักษร</u> เช่น บริเวณของส่วนหัว บริเวณที่มีจุดปลาย บริเวณที่มีจุดเชื่อมต่อ บริเวณที่มีส่วนหยัก บริเวณเริ่มต้น และบริเวณสิ้นสุดของตัวอักษร ฯลฯ

3.3.2.3 <u>การนับจำนวนลักษณะสำคัญของตัวอักษร</u> เช่น จำนวนส่วนย่อยของตัวอักษร จำนวนจุดปลาย จำนวนจุดเชื่อมต่อ จำนวนรอยหยัก จำนวนของส่วนย่อยที่มีลักษณะหรือตำแหน่งตามที่กำหนด ฯลฯ

ตัวอย่างเช่น หากผู้ใช้มีความรู้ว่าบริเวณของหัวของตัวอักษรสามารถนำไปใช้ในการจำแนกตัวอักษร จะทำการเพิ่มความรู้ภูมิหลัง *headzone(A,B) :- head(A,C), startzone(C,B).* เข้าไปในเซตของความรู้ภูมิหลัง ซึ่งความรู้ภูมิหลังโปรแกรมนี้หมายความว่า ในลิสต์ของเวกเตอร์พื้นฐาน *A* จะมีส่วนหัวของตัว

อักษรอยู่ในบริเวณ *B* ถ้าในลิสต์ของเวกเตอร์พื้นฐาน *A* นั้นมีส่วนหัวของตัวอักษรเป็นเวกเตอร์พื้นฐานแบบ *C* และเวกเตอร์พื้นฐานแบบ *C* นี้ มีจุดเริ่มต้นอยู่ที่บริเวณ *B* หรือหากเราเชื่อว่าประเภทของเวกเตอร์พื้นฐานที่เป็นส่วนหัวของตัวอักษร จะสามารถใช้จำแนกตัวอักษรได้ เราจะใช้ความรู้ภูมิหลัง *headprim(A,B) :- head(A,C), primitive(C,B).* ซึ่งหมายความว่า ในลิสต์ของเวกเตอร์พื้นฐาน *A* จะมีส่วนหัวเป็นเวกเตอร์พื้นฐานรูปแบบ *B* ถ้าในลิสต์ของเวกเตอร์พื้นฐาน *A* นั้นมีเวกเตอร์พื้นฐาน *C* เป็นส่วนหัว และ เวกเตอร์พื้นฐาน *C* นี้มีรูปแบบเป็นแบบ *B* เป็นต้น ตัวอย่างความรู้ภูมิหลังแสดงในรูปที่ 3.11 ความรู้ภูมิหลังทั้งหมดที่ใช้ในการทดลองนี้แสดงไว้ในภาคผนวก ค

```
head_zone(A,B)          :- head(A,C), startzone(C,B).
head_primitive(A,B)     :- head(A,C), primitive(C,B).

count_primitive_type4([],0).
count_primitive_type4([A|B],C) :-  primitive(A,4), count_primitive_type4(B,D), C is D+1.
count_primitive_type4([A|B],C) :-  not primitive(A,4), count_primitive_type4(B,C).

v_angle_at_head(A)      :- member(z2,A).

endpoint_primitive(A,B) :- member(C,A), endpoint(C,-1), primitive(C,B).

circle_at_endpoint_in_zone(A,B) :- member(C,A), iscircleendpoint(C), startzone(C,B).

count_circle_at_endpoint([],0).
count_circle_at_endpoint([A|B],C) :- iscircleendpoint(A),count_circle_at_endpoint(B,D), C is D+1.
count_circle_at_endpoint([A|B],C) :- not iscircleendpoint(A),count_circle_at_endpoint(B,D).

count_startzone5([],0).
count_startzone5([A|B],C)    :- startzone(A,5), count_startzone5(B,D), C is D+1.
count_startzone5([A|B],C)    :- not startzone(A,5), count_startzone5(B,C).

right_line(A)           :- member(B,A), endPoint(B,-1),endZone(B,1), primitive(B,1).
right_line(A)           :- member(B,A), endPoint(B,-1),endZone(B,1), primitive(B,2).

member_zone(A,B,C)      :- member(E,A), startzone(E,B), endZone(E,C).

head_primitive_type9or10(A) :- head_primitive(A,9).
head_primitive_type9or10(A) :- head_primitive(A,10).

head_primitive_type10or11(A) :- head_primitive(A,10).
head_primitive_type10or11(A) :- head_primitive(A,11).
```

รูปที่ 3.11 ตัวอย่างความรู้ภูมิหลัง

3.3.3 <u>การเรียนรู้กฎ</u>

ในกระบวนการเรียนรู้กฎโดยระบบโปรกอล ใช้ภาพตัวอย่างที่เป็นพยัญชนะ 44 ตัว สระ วรรณยุกต์ และตัวอักษรพิเศษ ทั้งหมด 23 ตัว ตัวเลขไทย 10 ตัว พิมพ์จากโปรแกรมไมโครซอฟต์เวิร์ด

(Mircrosoft Word) รุ่น 97 ออกทางเครื่องพิมพ์เลเซอร์ ที่ความละเอียดขนาด 300 จุดต่อนิ้ว รวมทั้งสิ้น 77 ตัวอักษร และอ่านกลับเป็นแฟ้มข้อมูลด้วยเครื่องกวาดตรวจ ที่ความละเอียดเท่ากัน คือ 300 จุดต่อนิ้ว แล้วใช้โปรแกรมเพนต์บรัช (Paintbrush) ของไมโครซอฟต์วินโดว์ส 95 (Microsoft Windows 95) ในการตัดตัวอักษรเป็นตัวอักษรเดี่ยวเก็บลงแฟ้มข้อมูลภาพ 1 แฟ้มต่อ 1 ตัวอักษร แล้วจึงนำแฟ้มข้อมูลภาพที่ได้ไปทำการประมวลผลขั้นต้น และใช้ลักษณะสำคัญของภาพเป็นตัวอย่างในการเรียนรู้

ผลการเรียนรู้ คือ กลุ่มของกฎซึ่งสามารถนำไปใช้จำแนกตัวอักษรได้ โดยกฎที่ได้มี 1 กฎต่อ 1 ตัวอักษร ดังแสดงในรูปที่ 3.12 เป็นตัวอย่างของกฎที่ได้จากระบบโปรกอล ตัวอย่างเช่น กฎแรกที่ได้ คือ กฎสำหรับตัวอักษร ‘ก’ ซึ่งหมายความว่า หากภาพตัวอักษรจะถูกรู้จำเป็นตัวอักษร ‘ก’ ตัวอักษรนั้นจะต้องมีส่วนหัวอยู่ในบริเวณเขต 3 หรือมุมล่างซ้าย ส่วนหัวนี้เป็นเวกเตอร์พื้นฐานแบบที่ 1 คือมีทิศทางอยู่ระหว่าง 45 องศา ถึง 90 องศา และไม่มีเวกเตอร์พื้นฐานแบบ 4 ประกอบอยู่ในภาพตัวอักษรนั้นเลย หรือในกฎที่ 2 หากภาพตัวอักษรจะถูกรู้จำเป็นตัวอักษร ‘ข’ ตัวอักษรนั้นจะต้องมีส่วนหัวอยู่ในบริเวณที่ 2 และเป็นเวกเตอร์พื้นฐานรูปแบบที่ 9, 0 หรือ 10 ในลิสต์ของเวกเตอร์พื้นฐานนั้นต้องมีเส้นตั้งทางขวามือ มีจุดปลาย 2 จุด โดยกฎทั้งหมดแสดงในภาคผนวก ง

```
ก(3,A,B,C,D,E)       :-  head_zone(B,3), head_primitive(B,1),
                         count_primitive_type4(B,0).
ข(3,A,B,C,D,E)       :-  not v_angle_at_head(D), head_zone(B,2),
                         endpoint_primitive(B,1),
                         circle_at_endpoint_in_zone(B,2),
                         count_circle_at_endpoint(B,1),
                         right_line(B), member_zone(B,4,1),
                         head_primitive_type9or10(B),
                         count_startzone5(B,0).
ข(3,A,B,C,D,E)       :-  head_zone(B,2), head_primitive(B,10),
                         endpoint_zone(B,4),
                         begin_and_endzone(B,2,1),
                         member_zone(B,2,1), member_zone(B,4,1),
                         have_member(B,7,2,2),
                         head_primitive_type9or10(B),
                         head_primitive_type10or11(B).
ค(3,A,B,[z3],C,D)    :-  not ^_angle([z3]), not v_angle(C), head_zone(B,0),
                         endpoint_primitive(B,6).
```

รูปที่ 3.12 ตัวอย่างกฎที่ได้จากกระบวนการโปรกอล

เนื่องจากกฎเหล่านี้ถูกสร้างขึ้นจากตัวอย่างบวกและตัวอย่างลบเพียง 2 ลักษณะ คือ ตัวอย่างบวกเมื่อต้องการให้ระบบทำการเรียนรู้ลักษณะของภาพตัวอักษรนั้น ๆ และตัวอย่างลบเพื่อให้ระบบทำการเรียนรู้ว่าภาพตัวอักษรลักษณะใดที่ไม่ใช่ภาพตัวอย่างที่ต้องการ ซึ่งในการรู้จำเมื่อนำกฎเหล่านี้ไปจำแนกภาพตัวอักษร หากภาพตัวอักษรไม่ตรงกับกฎของตัวอักษรนั้น ๆ ก็จะถือว่า ไม่ใช่ตัวอักษรนั้น ๆ ดังนั้นในกรณีภาพตัวอย่าง

มีสัญญาณรบกวน (noisy image) หรือภาพตัวอย่างใหม่ อาจไม่มีกฎข้อใดเลยที่ตรงกับภาพตัวอักษรนั้น ๆ หรืออาจมีกฎมากกว่าหนึ่งข้อที่ตรงกับตัวอย่างพอดี ระบบจะไม่สามารถทำการเลือกกฎได้ ในงานวิจัยนี้จึงได้ทำการทดลองประมาณเพื่อเลือกกฎที่ใกล้เคียงกับตัวอย่างมากที่สุด โดยเลือกวิธีการแบ็กพรอพาเกชันนิวรอลเน็ตเวิร์กซึ่งเป็นวิธีการเรียนรู้แบบหนึ่งที่สามารถทำการจำแนกกลุ่มตัวอย่างออกเป็นหลายกลุ่มได้

3.4 การประมาณเลือกกฎที่ใกล้เคียงโดยวิธีการแบ็กพรอพาเกชันนิวรอลเน็ตเวิร์ก

แบ็กพรอพาเกชันนิวรอลเน็ตเวิร์กเป็นวิธีการเรียนรู้ของเครื่องแบบหนึ่งซึ่งถูกนำไปใช้ในงานด้านการรู้จำอย่างแพร่หลาย ในการทดลองนี้ได้ใช้วิธีการนี้ทำการประมาณเพื่อเลือกกฎโดยใช้เครื่องมือสร้างแบ็กพรอพาเกชันนิวรอลเน็ตเวิร์กชื่อ เอเอ็ม 6 (Aspirin/Migraine:AM6) มาทำการทดลอง โดยการสร้างนิวรอลเน็ตเวิร์กขึ้นก่อน แล้วจึงนำนิวรอลเน็ตเวิร์กที่ได้นี้ ไปใช้งานต่อไป โดยในการทดลองนี้ได้ทำสร้างนิวรอลเน็ตเวิร์กที่มีโครงสร้างแตกต่างกัน 2 แบบ นำผลที่ได้เปรียบเทียบกัน

3.4.1 <u>โครงสร้างของนิวรอลเน็ตเวิร์ก</u>

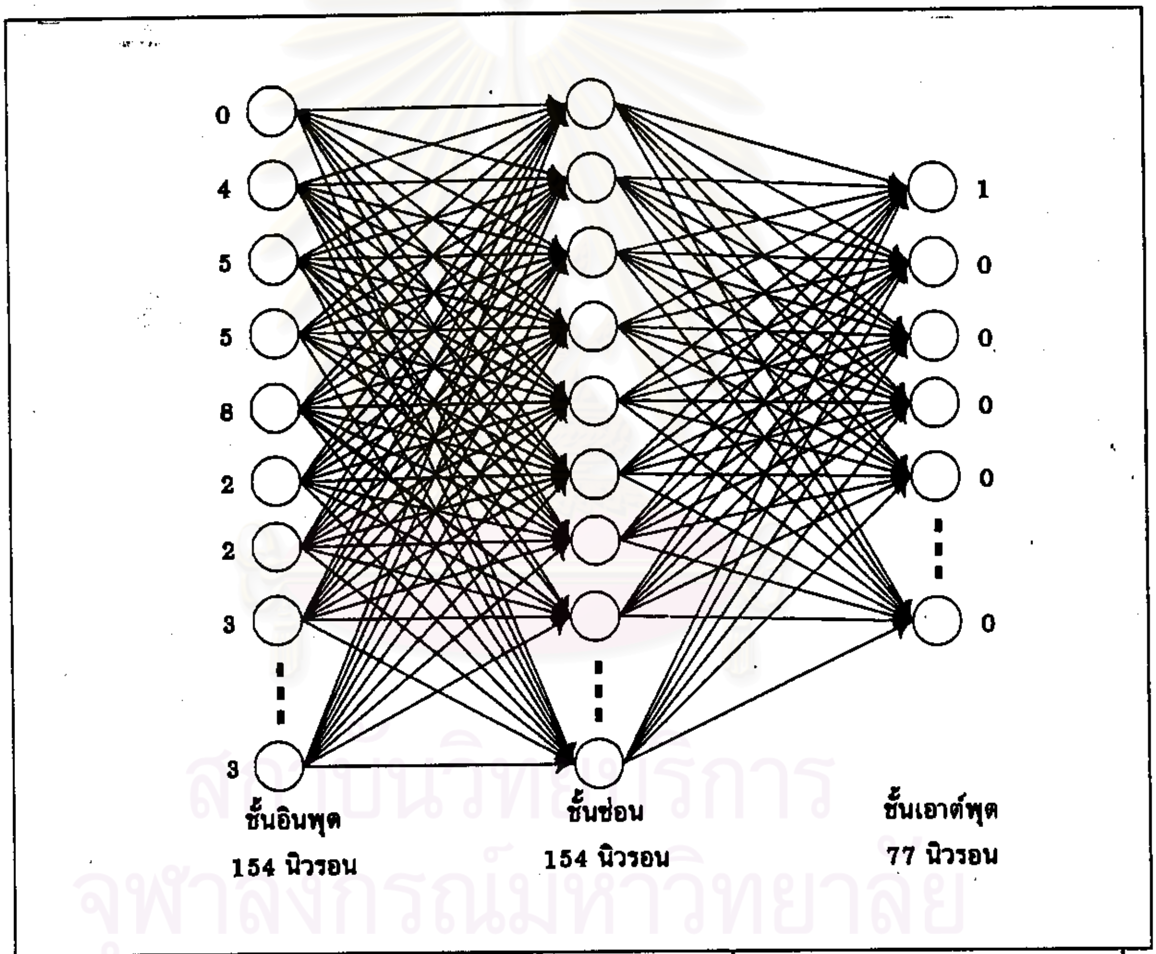
3.4.1.1 <u>นิวรอลเน็ตเวิร์กที่ใช้จำนวนสัญพจน์ที่ตรงและจำนวนสัญพจน์ที่ไม่ตรงกับตัวอย่างเป็นอินพุตเวกเตอร์</u>

เมื่อไม่สามารถเลือกกฎที่ตรงพอดีได้ กฎที่ใกล้เคียงที่สุดจึงควรเป็นกฎที่มีจำนวนสัญพจน์ที่ไม่ตรงกับตัวอย่างน้อย และมีจำนวนสัญพจน์ที่ตรงกับตัวอย่างมาก โดยแนวความคิดนี้จึงได้ออกแบบนิวรอลเน็ตเวิร์กแบบที่หนึ่งนี้ ให้ใช้จำนวนสัญพจน์ที่ไม่ตรงกับตัวอย่างและจำนวนสัญพจน์ที่ตรงกับตัวอย่างเมื่อนำตัวอย่างไปเปรียบเทียบกับกฎทุกข้อ เป็นอินพุตเวกเตอร์ (input vector) ของนิวรอลเน็ตเวิร์ก โดยมีโครงสร้างของนิวรอลเน็ตเวิร์กประกอบด้วย 3 ชั้น คือ ชั้นอินพุต (input layer) ชั้นซ่อน (hidden layer) และ ชั้นเอาต์พุต (output layer) โดยในแต่ละชั้นเชื่อมต่อกันทั้งหมด (fully connected) ในชั้นอินพุตประกอบด้วย 154 นิวรอน (neuron) แต่ละนิวรอนแทนจำนวนสัญพจน์ที่ไม่ตรงและจำนวนสัญพจน์ที่ตรงกับตัวอย่างเมื่อนำภาพตัวอักษรไปทำการเปรียบเทียบกับทุกสัญพจน์ในกฎทุกข้อ ซึ่งในการทดลองนี้ได้กฎทั้งสิ้น 77 กฎ แต่ละกฎใช้สำหรับตัวอักษรแต่ละตัว จึงมีจำนวนตัวเลขทั้งหมด 77 x 2 = 154 จำนวน ชั้นซ่อนประกอบด้วยนิวรอน 154 นิวรอน ซึ่งจำนวนนิวรอนในชั้นซ่อนนี้ได้จากการทดลองพยายามลดจำนวนนิวรอนในชั้นนี้ให้มีค่าน้อยที่สุดซึ่งทำให้นิวรอลเน็ตเวิร์กลู่เข้า (converge) ได้จำนวนนิวรอนในชั้นซ่อน 130 นิวรอน แต่เมื่อนำไปทำการทดลองพบว่าอัตราการรู้จำของนิวรอลเน็ตเวิร์กซึ่งมีจำนวนนิวรอนในชั้นซ่อน 130 นิวรอน มีค่าต่ำกว่านิวรอลเน็ตเวิร์กซึ่งมีจำนวนนิวรอนในชั้นซ่อนเท่ากับจำนวนนิวรอนในชั้นอินพุต คือ 154 นิวรอน จึงกำหนดให้จำนวนนิวรอนในชั้นนี้เท่ากับจำนวนนิวรอนในชั้นอินพุต คือ 154 นิวรอน และกำหนดให้ทุกนิวรอนเป็นองค์ประกอบซิกมอยด์ (zigmoid unit) ส่วนในชั้นเอาต์พุต ประกอบด้วยนิวรอน 77 นิวรอน แต่ละนิวรอนแทนตัวอักษรแต่ละตัว และเป็นองค์ประกอบซิกมอยด์เหมือนกับนิวรอนในชั้นซ่อน ในรูปที่ 3.13 แสดงตัวอย่างของนิวรอลเน็ตเวิร์กแบบนี้

3.4.1.2 <u>นิวรอลเน็ตเวิร์กที่ใช้ค่าความจริงของแต่ละสัญพจน์เป็นอินพุตเวกเตอร์</u>

จากโครงสร้างนิวรอลเน็ตเวิร์กในหัวข้อ 3.4.1.1 ซึ่งมีการกำหนดน้ำหนักความสำคัญให้กับจำนวนสัญพจน์ที่ไม่ตรงกับตัวอย่างและจำนวนสัญพจน์ที่ตรงกับตัวอย่าง ทำให้ไม่สามารถกำหนดน้ำหนักให้กับ

สัญพจน์ที่มีความสำคัญมากให้มีน้ำหนักมากกว่าสัญพจน์อื่นๆ จึงได้ออกแบบโครงสร้างนิวรอลเน็ตเวิร์กขึ้นอีกแบบหนึ่ง ให้สามารถกำหนดน้ำหนักความสำคัญให้กับแต่ละสัญพจน์ไม่เท่ากัน โดยใช้ค่าความจริงในแต่ละสัญพจน์ของกฎทุกข้อเป็นอินพุตเวกเตอร์ ซึ่งโครงสร้างของนิวรอลเน็ตเวิร์กแบบนี้จะคล้ายกับนิวรอลเน็ตเวิร์กในข้อ 3.4.1.1 คือ ประกอบด้วย 3 ชั้น ชั้นอินพุต ชั้นซ่อน และชั้นเอาต์พุต โดยแต่ละนิวรอนในชั้นอินพุตแทนค่าความจริงของแต่ละสัญพจน์และค่าคงที่ในกฎทุกข้อ ชั้นซ่อนประกอบด้วยนิวรอน 77 นิวรอน แต่ละนิวรอนแทนกฎของตัวอักษรและเป็นองค์ประกอบซิกมอยด์ โดยนิวรอนในชั้นอินพุตซึ่งแทนค่าความจริงของสัญพจน์และค่าคงที่ในกฎข้อใด ก็จะเชื่อมกับนิวรอนซึ่งแทนกฎข้อนั้นในชั้นซ่อน ส่วนชั้นเอาต์พุตจะมีลักษณะเหมือนนิวรอลเน็ตเวิร์กในข้อ 3.4.1.1 คือ มีจำนวนนิวรอน 77 นิวรอน แต่ละนิวรอนแทนตัวอักษรแต่ละตัวและเป็นองค์ประกอบซิกมอยด์เช่นเดียวกับนิวรอนในชั้นซ่อน การเชื่อมต่อของนิวรอนในชั้นซ่อนและชั้นเอาต์พุตเป็นแบบเชื่อมต่อกันทั้งหมด ดังแสดงตัวอย่างโครงสร้างของนิวรอลเน็ตเวิร์กแบบนี้ไว้ในรูปที่ 3.14



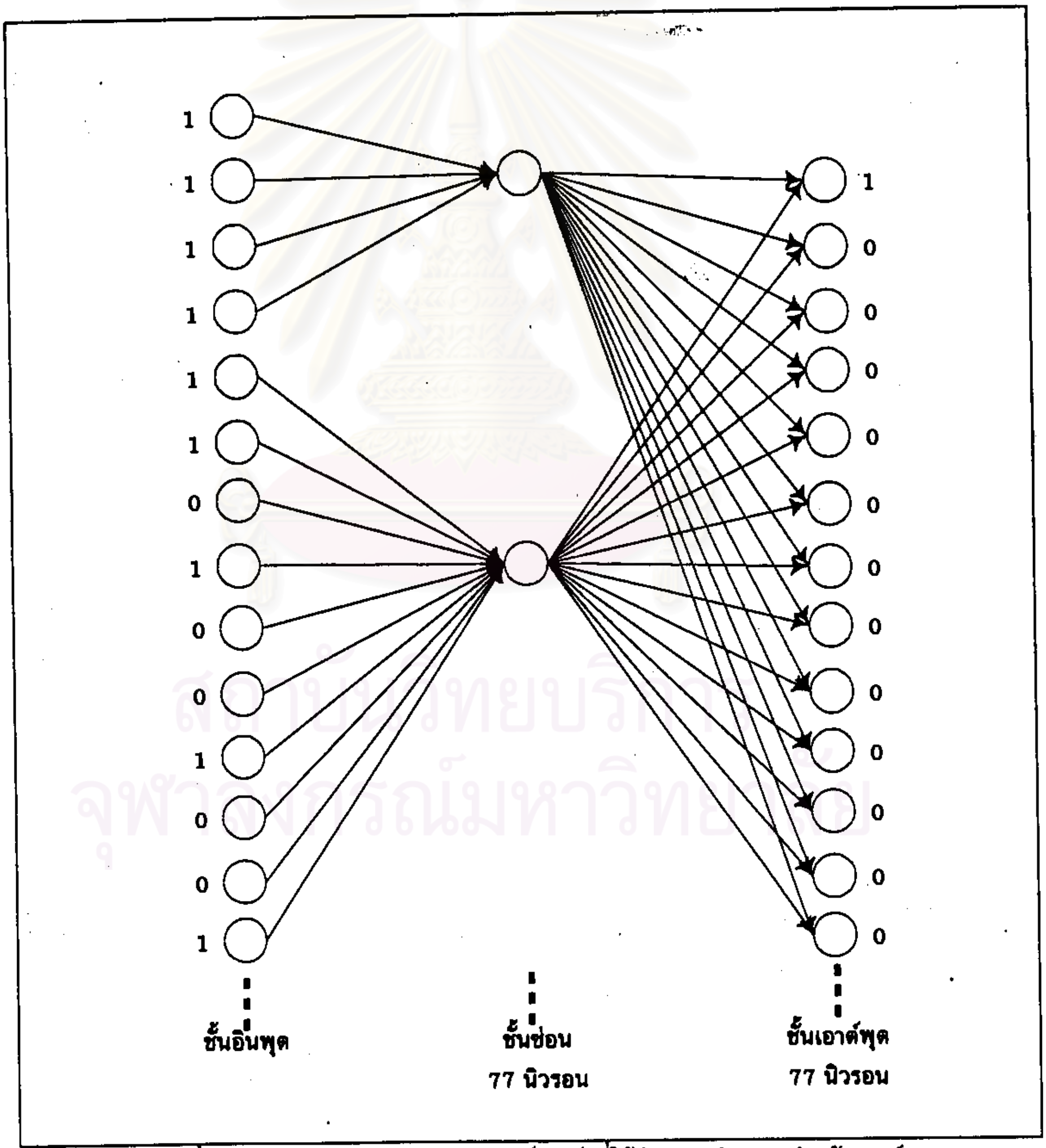
รูปที่ 3.13 แสดงโครงสร้างของนิวรอลเน็ตเวิร์กที่ใช้จำนวนสัญพจน์ที่ไม่ตรงกับตัวอย่างและจำนวนสัญพจน์ที่ตรงกับตัวอย่าง

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

3.4.2 <u>การเรียนรู้เพื่อสร้างนิวรอลเน็ตเวิร์ก</u>

ในการเรียนรู้เพื่อสร้างนิวรอลเน็ตเวิร์กด้วยขั้นตอนวิธีแบ็กพรอพาเกชัน ด้วยระบบเอเอ็ม 6 ได้ใช้ตัวอย่างจำนวนทั้งหมด 1078 ตัวอย่าง ซึ่งเป็นตัวอย่างเดียวกับตัวอย่างในการเรียนรู้กฎโดยระบบโปรกอล กำหนดค่าพารามิเตอร์ต่างๆ เป็นดังนี้

- จำนวนตัวอย่างที่ทดสอบในขณะทำการเรียนรู้ เป็นค่าที่ใช้เพื่อทำการทดสอบการลู่เข้าของนิวรอลเน็ตเวิร์กโดยระบบเอเอ็ม 6 ซึ่งระบบจะทำการทดสอบตัวอย่างตามจำนวนนี้เพื่อหาค่าผิดพลาดที่มากที่สุด หากค่าผิดพลาดนี้มีค่าน้อยกว่าค่าความผิดพลาดที่ยอมรับได้ขณะทำการเรียนรู้ ระบบจะถือว่านิวรอลเน็ตเวิร์กนี้ลู่เข้า โดยในโครงสร้างที่ใช้จำนวนสัญพจน์ที่ไม่ตรงกับตัวอย่างและจำนวนสัญพจน์ที่ตรงกับตัวอย่างเป็นอินพุตเวกเตอร์มีค่าเป็น 400 ตัวอย่าง ส่วนโครงสร้างที่ใช้ค่าความจริงของแต่ละสัญพจน์เป็นอินพุตเวกเตอร์มีค่าเป็น 200 ซึ่งค่าทั้งสองนี้ได้จากการทดลองเพื่อให้ได้ค่ามากที่สุด
- ค่าความผิดพลาดที่ยอมรับได้ขณะทำการเรียนรู้กำหนดเป็น 0.1
- ค่าอัตราการเรียนรู้ (learning rate) กำหนดให้เป็นค่า 0.05



รูปที่ 3.14 แสดงโครงสร้างของนิวรอลเน็ตเวิร์กที่ใช้ค่าความจริงของแต่ละสัญพจน์

เมื่อกำหนดพารามิเตอร์ต่างๆ เสร็จเรียบร้อยแล้ว จึงทำการเตรียมตัวอย่างเพื่อให้ระบบเอเอ็ม 6 ทำการเรียนรู้เพื่อหาค่าน้ำหนัก (weight) ของลิงค์ (link) และค่าไบแอส (bias) ของนิวรอนต่างๆ เพื่อสร้างเป็นนิวรอลเน็ตเวิร์ก โดยในการเตรียมตัวอย่างนี้นิวรอลเน็ตเวิร์กทั้งสองแบบมีขั้นตอนเหมือนกัน คือ การหาค่าความจริงของสัญพจน์และค่าคงที่ ซึ่งเมื่อได้ค่าความจริงเหล่านี้แล้วจึงนำไปจัดรูปแบบตามโครงสร้างของนิวรอลเน็ตเวิร์กเพื่อป้อนเป็นตัวอย่างให้กับระบบเพื่อทำการเรียนรู้ดังนี้

3.4.2.1 <u>นิวรอลเน็ตเวิร์กที่ใช้จำนวนสัญพจน์ที่ไม่ตรงและจำนวนสัญพจน์ที่ตรงกับตัวอย่างเป็นอินพุตเวกเตอร์</u>

ลักษณะสำคัญซึ่งได้จากการส่งภาพตัวอักษรจะผ่านการประมวลผลขั้นต้นเช่นเดียวกับในการเรียนรู้กฎเพื่อหาลักษณะสำคัญของภาพ แล้วนำลักษณะเหล่านี้ไปเปรียบเทียบกับสัญพจน์ของกฎทุกข้อในเซต และนับจำนวนสัญพจน์ที่ไม่ตรงกับตัวอย่างและจำนวนสัญพจน์ที่ตรงกับตัวอย่าง นำตัวเลขเหล่านี้ป้อนเป็นอินพุตเวกเตอร์ ในส่วนของเอาต์พุตกำหนดค่าของเอาต์พุตนิวรอนทุกนิวรอนเป็น '0' ยกเว้นนิวรอนที่แทนตัวอักษรที่เป็นภาพตัวอย่างนั้นให้กำหนดเป็น '1' ตัวอย่างเช่นในรูปที่ 3.13 แสดงตัวอย่างการเรียนรู้ของภาพตัวอักษร 'ก' ซึ่งมีวิธีการดังนี้

เมื่อนำภาพตัวอักษร 'ก' ไปเปรียบเทียบกับกฎของตัวอักษร 'ก' ได้จำนวนสัญพจน์ที่ไม่ตรงกับตัวอย่างเป็น 0 และจำนวนสัญพจน์ที่ตรงกับตัวอย่างเป็น 4 เมื่อนำภาพตัวอักษรเดียวกันนี้ไปเปรียบเทียบกับกฎของตัวอักษร 'ข' ได้จำนวนสัญพจน์ที่ไม่ตรงกับตัวอย่างเป็น 5 และจำนวนสัญพจน์ที่ตรงกับตัวอย่างเป็น 5 ส่วนค่าของนิวรอนอื่นๆ ก็ได้มาด้วยวิธีการเดียวกัน คือ นำภาพตัวอักษรนี้ไปเปรียบเทียบกับทุกสัญพจน์ในกฎทุกข้อ เพื่อหาจำนวนสัญพจน์ที่ไม่ตรงกับตัวอย่างและจำนวนสัญพจน์ที่ตรงกับตัวอย่างของกฎแต่ละข้อ แล้วจัดเป็นอินพุตเวกเตอร์สำหรับทำการเรียนรู้ ในส่วนของชั้นเอาต์พุต เนื่องจาก ตัวอย่างที่ทำการเรียนรู้อยู่นี้คือภาพตัวอักษร 'ก' จึงกำหนดค่าให้กับนิวรอนที่หนึ่ง ซึ่งแทนตัวอักษร 'ก' ให้เป็น '1' ในขณะที่นิวรอนอื่นๆ กำหนดค่าให้เป็น '0'

3.4.2.2 <u>นิวรอลเน็ตเวิร์กที่ใช้ค่าความจริงของแต่ละสัญพจน์เป็นอินพุตเวกเตอร์</u>

เช่นเดียวกับหัวข้อ 3.4.2.1 ภาพตัวอักษรที่ได้รับการอ่านด้วยเครื่องกวาดตรวจ จะถูกส่งผ่านการประมวลผลขั้นต้น จากนั้นลักษณะสำคัญที่ได้จากภาพจะถูกนำไปทดสอบกับสัญพจน์ทุกสัญพจน์ในกฎทุกข้อ ถ้าสัญพจน์ใดตรงกับตัวอย่างกำหนดค่าให้เป็น '1' ถ้าสัญพจน์ใดไม่ตรงกับตัวอย่างกำหนดค่าให้เป็น '0' สำหรับในส่วนของเอาต์พุตให้กำหนดค่าของทุกนิวรอนเป็น '0' ยกเว้นนิวรอนที่แทนตัวอักษรของภาพตัวอักษรนั้นให้กำหนดค่าเป็น '1' ดังตัวอย่างในรูปที่ 3.14 แสดงตัวอย่างการเรียนรู้ของภาพตัวอักษร 'ก' ซึ่งมีวิธีการดังนี้

เมื่อนำตัวอักษร 'ก' ไปเปรียบเทียบกับกฎของตัวอักษร 'ก' จะเห็นว่าภาพตัวอักษรนี้อยู่ในระดับ 3 จึงกำหนดค่านิวรอนที่หนึ่งเป็น '1' ภาพตัวอักษรมีส่วนหัวอยู่ในบริเวณที่ 3 จึงกำหนดค่านิวรอนที่สองเป็น '1' ส่วนหัวของตัวอักษรเป็นรูปแบบที่ 1 กำหนดค่านิวรอนที่สามเป็น '1' ไม่มีเวกเตอร์พื้นฐานรูปแบบที่ 4 อยู่ในภาพเลย กำหนดค่านิวรอนที่สี่เป็น '1' และเมื่อนำภาพตัวอักษรนี้ไปเปรียบเทียบกับค่าคงที่และสัญพจน์ในกฎของตัวอักษร 'ข' ได้ค่าความจริงเป็น '1' '1' '0' '1' '0' '0' '1' '0' '0' '1' ตามลำดับ ใน

ส่วนของเอาต์พุต เนื่องจากตัวอย่างนี้เป็นตัวอย่างของภาพอักษร 'ก' จึงกำหนดให้นิวรอนที่หนึ่งซึ่งแทนตัวอักษร 'ก' มีค่าเป็น '1' ส่วนนิวรอนอื่นกำหนดค่าเป็น '0' ดังแสดงในรูปที่ 3.14

เมื่อจัดตัวอย่างทั้งสองแบบเสร็จ จึงป้อนให้กับระบบเอเอ็ม 6 ทำการเรียนรู้เพื่อหาค่าน้ำหนักของลิงค์ (link) และไบแอส (bias) ของนิวรอน จะได้เป็นนิวรอลเน็ตเวิร์กที่เสร็จเรียบร้อย สามารถนำไปใช้รู้จำตัวอักษรได้ ต่อไป

3.5 ขั้นตอนการรู้จำตัวอักษร

การทดลองในงานวิจัยนี้ทำโดยการนำภาพตัวอักษรที่ใช้ทำการเรียนรู้มาเพิ่มสัญญาณรบกวนโดยการถ่ายเอกสารแบบเข้มและแบบบาง รวมเป็นตัวอย่างที่ใช้ทำการทดสอบจำนวนทั้งสิ้น 2,156 ตัวอย่าง นำตัวอย่างทั้งหมดมาผ่านการประมวลผลขั้นต้นเพื่อหาลักษณะสำคัญของภาพตัวอักษรเช่นเดียวกับการประมวลผลขั้นต้นในการเรียนรู้กฎและการสร้างเน็ตเวิร์ก ลักษณะสำคัญที่ได้จะต่างไปจากตัวอย่างที่ทำการเรียนรู้ จากนั้นจึงนำลักษณะสำคัญเหล่านี้ไปทำการทดสอบเพื่อหาค่าความจริงของแต่ละสัญพจน์ในกฎทุกข้อ แล้วจึงนำค่าความจริงเหล่านี้ไปทดสอบกับนิวรอลเน็ตเวิร์กที่สร้างขึ้นไว้แล้วทั้งสองแบบเพื่อทำการรู้จำว่าตัวอย่างเป็นภาพของตัวอักษรใด และเปรียบเทียบกับวิธีการอื่น ๆ ต่อไป

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย