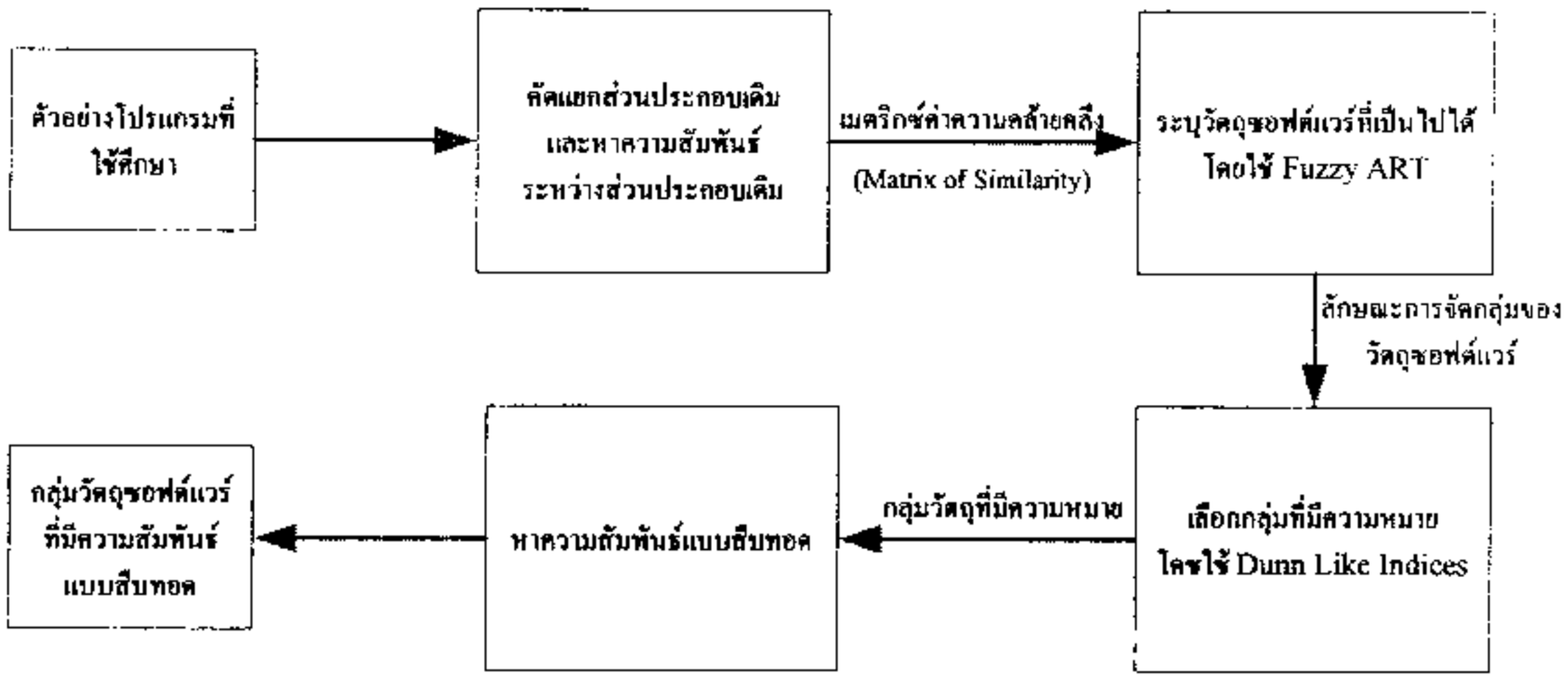
บทที่ 3

ขั้นตอนการดำเนินงาน

การระบุวัตถุซอฟต์แวร์ที่เป็นไปได้ หมายถึง การค้นหาและปรับเปลี่ยนส่วนประกอบของโปรแกรมเชิงโครงสร้างให้สามารถนำไปใช้งานในโปรแกรมเชิงวัตถุได้ ซึ่งวิทยานิพนธ์ฉบับนี้ มีขั้นตอนการดำเนินงานทั้งหมด 4 ขั้นตอน คือ 1) คัดส่วนจำเพาะประกอบเดิมและหาความสัมพันธ์ระหว่างส่วนประกอบเดิม 2) ระบุวัตถุซอฟต์แวร์ที่เป็นไปได้ โดยใช้วิธีการจัดกลุ่มข้อมูลแบบฟัซซี่เออาร์ที 3) เลือกกลุ่มที่มีความหมาย 4) หาความสัมพันธ์แบบสืบทอดของวัตถุที่เลือก ดังแสดงในภาพที่ 3.1



ภาพที่ 3.1 ขั้นตอนการดำเนินงาน

3.1 คัดแยกส่วนประกอบเดิม

การระบุวัตถุซอฟต์แวร์ที่เป็นไปได้ หมายถึง การค้นหาและปรับเปลี่ยนส่วนประกอบของโปรแกรมเชิงโครงสร้างให้สามารถนำไปใช้งานในโปรแกรมเชิงวัตถุได้

3.1.1 การคัดแยกส่วนประกอบเดิม (Extracting Artifacts)

การคัดแยกส่วนประกอบเดิม หมายถึง การนำโปรแกรมเชิงโครงสร้างที่เราต้องการจะทำการปรับเปลี่ยนมาวิเคราะห์เพื่อคัดแยกส่วนประกอบเดิมที่ต้องการออกมา โดยส่วนประกอบเดิมที่แยกออกมานั้นจะมีลักษณะ 2 แบบ คือ

3.1.1.1 ส่วนของข้อมูล (Data Artifacts) คือ ส่วนประกอบที่ทำหน้าที่เก็บข้อมูลในโปรแกรมเชิงโครงสร้าง เช่น สตรักท์ (Struct) ตัวแปรส่วนกลาง (Global Variable) ฯลฯ

3.1.1.2 ส่วนของคำสั่ง (Operation Artifacts) คือ ส่วนประกอบที่ทำหน้าที่ในการทำงานตามคำสั่งต่าง ๆ ภายในโปรแกรมที่ถูกเขียนโดยผู้เขียนโปรแกรม เช่น กระบวนคำสั่ง (Procedure) ฟังก์ชัน (Function) ฯลฯ

3.1.2 การค้นหาความสัมพันธ์ระหว่างส่วนประกอบเดิม (Exploring Relations)

การค้นหาความสัมพันธ์ระหว่างส่วนประกอบเดิม หมายถึง การวิเคราะห์ว่าส่วนประกอบเดิมเหล่านั้นมีความสัมพันธ์กันอย่างไร เช่น การสร้างกราฟอ้างอิง (Reference Graph) หรือการทำโค้ดสไลซ์ซิ่ง (Code Slicing) การค้นหาความสัมพันธ์ระหว่างส่วนประกอบเดิม ในส่วนของการเตรียมข้อมล มีขั้นตอนการทำงานดังนี้

การเตรียมข้อมูล คือ การบันทึกรายละเอียดของข้อมูลที่เราต้องการจัดกลุ่มลงในเมตริกซ์โดยใช้คุณสมบัติ (Attribute) อย่างใดอย่างหนึ่งเป็นเกณฑ์ ทั้งนี้เพื่อจะได้ทราบว่าข้อมูลแต่ละอันนั้นมีคุณสมบัติเหมือนกันหรือต่างกันอย่างไร โดยจะบันทึกลงในเมตริกซ์ที่เรียกว่า เมตริกซ์ข้อมูล (Data Matrix) ซึ่งมีลักษณะตามตารางที่ 3.1

ตารางที่ 3.1 ตัวอย่างเมตริกซ์ข้อมูล

	Attribute 1	Attribute 2	Attribute 3
Object 1	1.0	0.0	1.0
Object 2	0.0	1.0	0.0
Object 3	1.0	0.0	1.0

หลังจากนั้นนำค่าคุณสมบัติที่แสดงในตารางมาคำนวณเพื่อหาค่าระยะทาง (Distance) ค่าระยะทาง คือ ตัวเลขที่บอกถึงความแตกต่างของข้อมูลแต่ละคู่ ในที่นี้จะเลือกใช้สูตรซึ่งมีรายละเอียดดังนี้

$$sim(x,y) = \frac{\sum(xy)}{\sqrt{\sum x^2} \times \sqrt{\sum y^2}}$$

หลังจากคำนวณแล้วจะบันทึกค่าระยะทางที่ได้ลงในเมตริกซ์ความแตกต่าง (Dissimilarity Matrix) ซึ่งจะมีขนาด n x n เสมอ เมื่อ n คือ จำนวนของข้อมูล และจะแสดงข้อมูลไว้เพียงครึ่งเดียว ตามตารางที่ 3.2

ตารางที่ 3.2 ตัวอย่างเมตริกซ์ความแตกต่าง

	Attribute 1	Attribute 2	Attribute 3
Object 1	1.0	0.0	1.0
Object 2		1.0	0.0
Object 3			1.0

3.2 ระบุหาวัตถุซอฟต์แวร์ที่เป็นไปได้

การระบุวัตถุซอฟต์แวร์ที่เป็นไปได้ หมายถึง การเลือกกลุ่มของส่วนชุดคำสั่งที่มีความหมายมารวมกับส่วนข้อมูลที่มีความเกี่ยวข้องกันเพื่อประกอบเป็นวัตถุซอฟต์แวร์ที่เป็นไปได้ ทั้งนี้ขึ้นอยู่กับวิจารณญาณของแต่ละบุคคลว่าจะเลือกระบุวัตถุซอฟต์แวร์แบบใด ดังนั้นงานวิจัยส่วนใหญ่จึงไม่ค่อยอธิบายการทำงานในขั้นตอนนี้ เพียงแต่กล่าวว่า ให้เลือกกลุ่มส่วนประกอบเดิมที่มีความหมายมาประกอบกัน แล้วสร้างเป็นวัตถุซอฟต์แวร์ที่เป็นไปได้ สำหรับการระบุวัตถุซอฟต์แวร์ที่เป็นไปได้นั้น จะใช้เทคนิคฟัซซี่ เออาร์ที (Fuzzy ART)

ฟัซซี่ เออาร์ที (Fuzzy ART)

Adaptive Resonance Theory (ART) พัฒนาขึ้นโดย Carpenter และ Grossberg ในปี ค.ศ. 1987 (Fausett, 2001 : 218) เพื่อวิเคราะห์กระบวนการรับรู้ข่าวสารของมนุษย์ (Human Cognitive Information Processing) หลักการทำงานถูกออกแบบเพื่อให้หน่วยความจำไม่มีการเปลี่ยนแปลงไม่ว่าจะเกิดการเรียนรู้แบบรวดเร็ว (Fast Learning) หรือการเรียนรู้แบบช้า (Slow Learning) กับทุกสภาพข้อมูลนำเข้า

ART มีคุณสมบัติพื้นฐานดังนี้

1. Self-Scaling Computational Units การทำงานของ Attentional Subsystem ขึ้นอยู่กับ Competitive Learning ช่วยเสริมลักษณะพิเศษของรูปแบบข้อมูล (Pattern) และกำจัดสิ่งรบกวน (Noise)

2. Self-Adjusting Memory Search ระบบสามารถทำการค้นหาไปพร้อม ๆ กับการปรับค่าน้ำหนัก

3. ระบบจะทำการปรับ Attentional Vigilance โดยใช้สภาพแวดล้อมเป็นต้นแบบ ถ้าสภาพแวดล้อมไม่สอดคล้องกับการจำได้ปัจจุบัน (Current Recognition) ของระบบ ก็จะมีการปรับไปใช้ค่า Vigilance อื่น

นอกจากนี้ ART (Fausett, 2001 : 218-219) ยังมีคุณสมบัติพิเศษหลายประการ เช่น

1. สามารถเลียนแบบพฤติกรรมทางชีววิทยา โดยเน้นที่การเรียนรู้ที่ไม่สามารถควบคุมได้ (Unsupervised Learning) และ Self-Organisation ซึ่งสามารถกำหนดรูปแบบของคลัสเตอร์หรือขั้นตอนของประเภทการเรียนรู้

2. ART Network ถูกพัฒนาเพื่อให้ผู้ใช้สามารถกำหนดค่าความคล้ายคลึง (The Degree of Similarity) ของแต่ละรูปแบบที่อยู่ในคลัสเตอร์เดียวกันได้

3. ART สามารถแก้ปัญหา Stability-Plasticity Dilemma

4. สามารถนำมาประยุกต์ใช้ในงานด้านต่าง ๆ เช่น ใช้ในการแยกให้เห็นความแตกต่างระหว่างเห็ดที่สามารถนำมารับประทานกับเห็ดที่มีพิษ การวิเคราะห์เชื้อโรคต่าง ๆ หรือแม้แต่แบบจำลองของกระบวนการทางชีววิทยาของระบบประสาท

ART มีด้วยกันหลายประเภท ขึ้นอยู่กับการนำไปประยุกต์ใช้งาน เช่น

1. ART1 เหมาะสำหรับข้อมูลที่มีลักษณะเป็นไบนารี่ โดยแบ่งรูปแบบของการเรียนรู้ออกเป็น 2 รูปแบบ คือ 1) การเรียนรู้แบบช้า (Slow-Learning) 2) การเรียนรู้แบบรวดเร็ว (Fast-Learning)

2. ART2 เหมาะสำหรับข้อ มูลที่มีลักษณะเป็นแอนาล็อก (Continuous-Value or Analog)

3. Fuzzy ART เหมาะสำหรับข้อมูลที่มีลักษณะเป็นทั้งไบนารี่ หรือ แอนาล็อก

โครงสร้างพื้นฐานของ ART ประกอบด้วย 2 ระบบย่อย

1. Attentional Subsystem ซึ่งขั้นตอนของการเรียนรู้และการกระตุ้น จะเกิดขึ้นในส่วนนี้ ประกอบด้วย

1) Input Process Field หรือ F_1 Layer แบ่งย่อยออกเป็น 2 ส่วนด้วยกันคือ

(1) Input Portion เราจะแทนด้วย $F_1(a)$

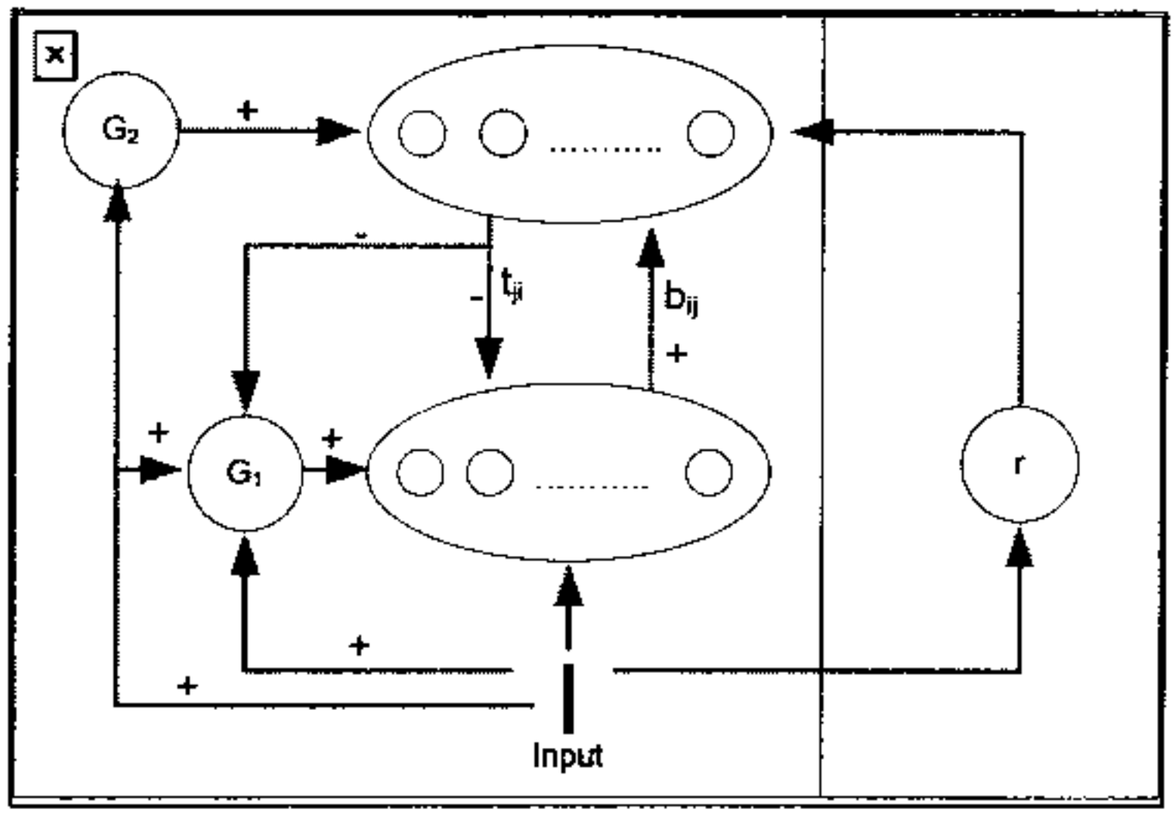
(2) Interface Portion เราจะแทนด้วย $F_1(b)$ โดยจะทำการเปรียบเทียบค่าของน้ำหนักที่ได้จาก Input Portion กับ F_2 Layer

2) Cluster Unit หรือ F_2 Layer ซึ่งบางครั้งอาจเรียกว่า Competitive Layer จะทำการเลือกค่าน้ำหนักที่มากที่สุดให้เป็นวินเนอร์ (Winner) และทำการปรับค่าน้ำหนักที่ไม่ได้เป็น วินเนอร์ ให้เป็น 0

3) Gain Control ควบคุมการทำงานของ F_1 และ F_2 Layer

2. Orienting Subsystem จะทำการควบคุมการทำงานของ Attentional Subsystem เมื่อเกิดการจับคู่กันอย่างไม่เหมาะสม (Mismatch) ขึ้นที่ Attentional Subsystem โดยจะประกอบด้วยกลไกการทำงาน (Reset Mechanism) ไว้ใช้สำหรับควบคุมค่าความคล้ายคลึงของรูปแบบข้อมูลที่จะอยู่ในคลัสเตอร์เดียวกัน

แต่สำหรับ Kumar นั้นไม่ได้แบ่ง F_1 layer ออกเป็น 2 ส่วนเหมือนกับ Fausett ซึ่งวิทยานิพนธ์ฉบับนี้จะใช้แผนภาพของ Kumar อธิบายหลักการทำงานของ ART ดังแสดงในภาพที่ 3.2



ภาพที่ 3.2 โครงสร้างการทำงานของ ART

แหล่งที่มา: Kumar, 2005: 491.

ART มีหลักการทำงานโดยทั่วไป ดังนี้ (Kumar, 2005 : 491)

1. การทำงานจะเริ่มต้นเมื่อมีข้อมูลเข้าสู่ F_1 Layer โดยข้อมูลที่จะเข้าสู่ F_1 นั้นมาจาก 3 แหล่งด้วยกันคือ

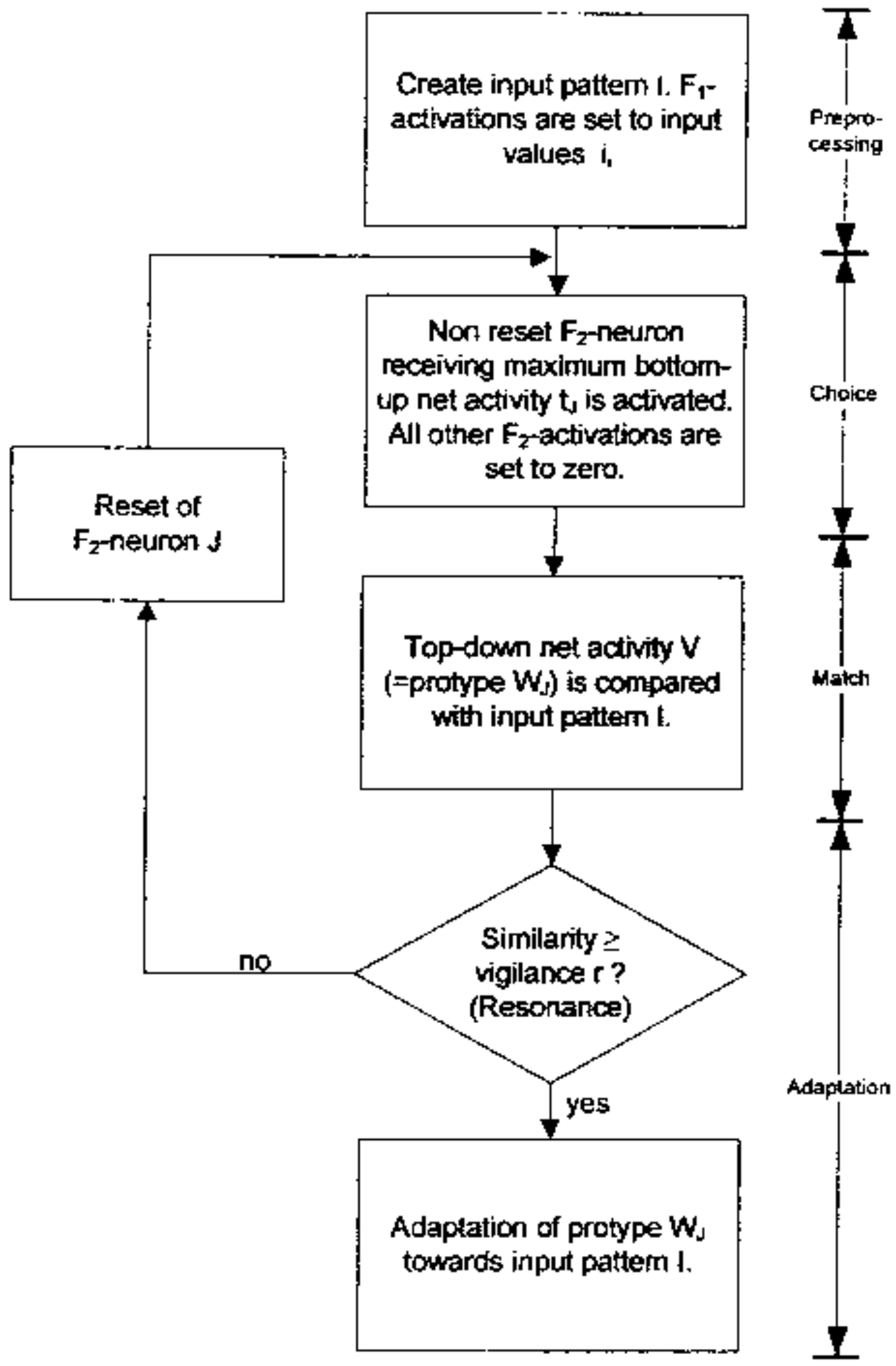
- ข้อมูลนำเข้า (Input) ที่มาจาก I
- G หรือ Gain Control
- t_{ji} หรือ top-down weight

ระบบจะทำงานได้นั้นต้องมีข้อมูลนำเข้าอย่างน้อย 2 แหล่งเข้ามาในระบบ จึงจะทำงานได้ ซึ่งจะเป็นไปตามกฎ 2/3 rule ซึ่งในที่นี้จะเรียกว่า X โดยข้อมูลจะถูกส่งเข้าไปแบบกระจายอย่างทั่วถึง (Non-Specific) และส่งสัญญาณไประงับการทำงานของ Reset Node จากนั้นค่า X จะถูกส่งไปยัง F_2 Layer เราเรียกว่า bottom-up weight โดยจะแทนด้วยสัญลักษณ์ b_{ij} ซึ่งหมายถึงค่าน้ำหนักที่ส่งมาจากโหนด ที่ i ของ F_1 layer ไปยังโหนดที่ j ของ F_2 layer

2. F_2 Layer จะทำงานได้ก็ต้องเป็นไปตามกฎ 2/3 rule เช่นเดียวกับ F_1 Layer โดย F_2 Layer จะทำการเปรียบเทียบน้ำหนักทุกค่าที่เข้ามาว่าน้ำหนักไหนมีค่ามากที่สุด ก็จะเลือกน้ำหนักดังกล่าวให้เป็นวินเนอร์ (Winner) และปรับค่าน้ำหนักที่เหลือให้เป็น 0 จากนั้น F_2 Layer ก็จะส่งค่าน้ำหนักที่เป็นวินเนอร์ กลับมายัง F_1 Layer อีกครั้ง ซึ่งเรียกค่าดังกล่าวว่า top-down weight โดยจะแทนด้วยสัญลักษณ์ t_{ji} ซึ่งหมายถึงค่าน้ำหนักที่ส่งมาจากโหนด ที่ j ของ F_2 Layer ไปยังโหนด ที่ i ของ F_1 Layer

3. F_1 Layer นี้จะทำการจับคู่น้ำหนักที่ได้จาก F_2 Layer กับ Input Pattern ที่เข้ามาว่าเหมาะสมกันหรือไม่ สำหรับสาเหตุที่อาจทำให้เกิดการจับคู่อย่างไม่เหมาะสมกัน ก็เนื่องมาจากข้อมูลมีลักษณะแตกต่างกันมาก ถ้าเกิดข้อมูลที่ได้จับคู่กันอย่างเหมาะสม ระบบก็จะเกิดการเรียนรู้ (Learning or Resonance) และมีการปรับค่าน้ำหนักใหม่เกิดขึ้น แต่ถ้าผลการเปรียบเทียบจับคู่กันอย่างไม่เหมาะสม ระบบก็จะส่งสัญญาณไปบอกยัง Reset Node ว่าข้อมูลจับคู่กันไม่เหมาะสม จากนั้น Reset Node จะส่งสัญญาณไปยัง F_2 layer เพื่อให้เลือกค่าน้ำหนักที่เหลือที่มีค่ามากที่สุดมาใหม่อีกครั้ง

4. จากนั้นก็จะทำซ้ำตามขั้นตอนที่ 2-3 จนกว่าข้อมูลจะจับคู่กันอย่างเหมาะสม จากหลักการดังกล่าวสามารถแบ่งขั้นตอนการทำงานออกเป็น 4 ขั้นตอน คือ 1) Preprocessing 2) Choice 3) Match 4) Adaptation ตามภาพที่ 3.3



ภาพที่ 3.3 ขั้นตอนการทำงานของ ART

แหล่งที่มา: Frank, 1998: 545.

พารามิเตอร์ที่สำคัญของ ART ที่สำคัญ ได้แก่ (Fausett, 2001 : 225)

n	จำนวนของ Input Vector
m	จำนวนคลัสเตอร์ ที่ต้องการ
b_{ij}	ค่าน้ำหนักจากโหนด X_i ของ F_1 Layer ไปยังโหนด Y_j ของ F_2 Layer
t_{ji}	ค่าน้ำหนักจากโหนด Y_j ของ F_2 Layer ไปยังโหนด X_i ของ F_1 Layer
ρ	ค่า vigilance หรือค่าความคล้ายคลึงของกลุ่มข้อมูลที่จะอยู่ในกลุ่มเดียวกัน
s	Input Vector
x	Activation Vector
$\|x\|$	ขนาดของเวกเตอร์ x ซึ่งหาได้จากผลรวมของทุกค่า x_i

ขั้นตอนการทำงานของ ART 1 (Fausett, 2001 : 227)

ขั้นตอนที่ 1 กำหนดค่าพารามิเตอร์

$L > 1$ (ใช้ในการปรับค่า bottom-up weight)

$0 < \rho \leq 1$

กำหนดค่าเริ่มต้นให้กับน้ำหนักต่าง ๆ โดย

$$0 < b_{i,j}(0) < \frac{L}{L-1+n}$$

$$t_{ji}(0) = 1$$

ขั้นตอนที่ 2 ให้ทำขั้นตอนที่ 3-14 จนกว่าเงื่อนไขจะเป็นเท็จ

ขั้นตอนที่ 3 สำหรับแต่ละ Training Input ให้ทำตั้งแต่ขั้นตอนที่ 4-12

ขั้นตอนที่ 4 กำหนดให้ Activation ทั้งหมดของ F_2 มีค่าเท่ากับ 0 และกำหนดให้ activation ของ F_1 เป็น input

ขั้นตอนที่ 5 คำนวณค่า Norm ของ s จากสูตร $\|s\| = \sum s_i$

ขั้นตอนที่ 6 ส่งค่าของ input จาก F_1 layer ไปยัง F_2 layer โดย

$x_i = s_i$

ขั้นตอนที่ 7 สำหรับทุก node ของ F_2 ที่ยังไม่ inhibited

ถ้า $y_j \neq -1$ แล้ว $y_j = \sum b_{ij} x_i$

ขั้นตอนที่ 8 ในขณะที่ reset node ยังเป็น true ให้ทำขั้นตอนที่ 9-12

ขั้นตอนที่ 9 เลือกค่า J ที่ทำให้ y_J มีค่ามากที่สุด ($y_J \geq y_j$) สำหรับทุก node j และ ถ้า y_J มีค่า -1 จะทำให้ทุก node มีสถานะเป็น inhibited และ pattern นี้จะไม่สามารถจัดกลุ่มได้

ขั้นตอนที่ 10 คำนวณค่า x ของ F_1 อีกครั้ง โดย

$x_i = s_i t_{Ji}$

ขั้นตอนที่ 11 คำนวณขนาดของ vector x โดย

$\|x\| = \sum x_i$

ขั้นตอนที่ 12 ทดสอบ reset โดย

ถ้า $\frac{\|x\|}{\|s\|}$ น้อยกว่า ρ แล้ว $y_J = -1$ และกลับไปทำขั้นตอนที่ 8

แต่ถ้า $\frac{\|x\|}{\|s\|}$

มากกว่า ρ ให้ไปทำขั้นตอนที่ 13

ขั้นตอนที่ 13 ทำการปรับค่าน้ำหนัก ที่ node J โดย

$$b_{iJ}(new) = \frac{Lx_i}{L-1+\|x\|},$$

$$t_{Ji}(new) = x_i$$

ขั้นตอนที่ 14 จะหยุดการทำงานเมื่อเกิดเหตุการณ์ตามเงื่อนไข

- ไม่มีการเปลี่ยนแปลงของนำหนัก (no weight changes)
- ไม่เกิดการรีเซ็ท (no unit reset)

Fuzzy ART (Carpenter and others, 1991 : 759) ถูกพัฒนาขึ้นในปี ค.ศ. 1991 จากการรวมทฤษฎี Fuzzy Set กับ ART1 เพื่อให้ใช้งานได้กับข้อมูลที่มีลักษณะเป็นไบนารี่หรือ แอนาล็อก หลักการโดยทั่วไป คือ แทนที่ลักษณะของ Intersection Operator (∩) ของ ART1 ซึ่ง Intersection Operator (∩) เป็นตัวดำเนินการทางเซต ด้วย MIN Operator (^) ของ Fuzzy Set Theory โดย MIN Operator จะช่วยลด Intersection Operator ในกรณีที่ข้อมูลเป็นแบบไบนารี่ การเพิ่มขึ้นอย่างรวดเร็วของหมวดหมู่ (Category) สามารถป้องกันได้โดยการทำให้เวกเตอร์นำเข้ากลับเข้าสู่สภาพปกติ (Normalize Input Vector) ในขั้นตอน Preprocessing หรือที่เรียกว่า Complement Coding ซึ่งจะนำไปสู่ Symmetric Theory โดยมี MIN Operator และ MAX Operator เป็นตัวช่วยทำให้สมบูรณ์ยิ่งขึ้น

Fuzzy ART แบ่งการทำงานออกเป็น 4 ขั้นตอน ตามภาพที่ 3.2 ดังนี้ (Carpenter and others, 1991 : 763)

1. Preprocessing

❖ **Input Vector**

ให้ I เป็นเวกเตอร์ ที่มีมิติขนาด M โดยที่สมาชิกแต่ละตัวในเวกเตอร์ I มีค่าอยู่ตั้งแต่ 0 ถึง 1

$$i_k \in [0,1] \quad \forall k.$$

เราสามารถนำเอา Complement Coding เข้ามาเพื่อแก้ปัญหาการเพิ่มขึ้นอย่างรวดเร็วของ Cluster เพราะองค์ประกอบภายในเวกเตอร์ของ Prototype สามารถมีขนาดเล็กลงได้ในขั้นตอน Adaptation โดย Complement Coding สามารถคำนวณได้จากสูตรยูคลิเดียน เพื่อจะทำการแปลงรูปแบบข้อมูลนำเข้า (Input Pattern) A ไปเป็น Code Input (I) แต่ว่าวิธีนี้มีข้อเสียคือข้อมูลบางอย่างที่เก็บไว้ในรูปแบบข้อมูลนำเข้า (Input Pattern) จะสูญหายไป ดังนั้น จึงได้มีการคิดสูตรใหม่ขึ้นมา ให้เวกเตอร์ดั้งเดิม (Original Vector) $A = (a_1, \ldots, a_k)$ ที่จะประมวลใส่รหัสไป

ยังข้อมูลนำเข้า (Input Pattern) $I = (i_1,\ldots, i_m)$ โดยเพิ่ม Complement ของ A ไปยัง Original Vector และเวกเตอร์ดังกล่าวก็จะมีขนาดเพิ่มเป็น 2 เท่า

$$I = (A, A^c)$$
$$= (a_1,\ldots,a_k,1-a_1,\ldots,1-a_k) \quad a_i \in [0,1] \quad \forall i.$$

❖ **Weight vector**

แต่ละ category (j) $(1 \le j < N)$ ใน F_2 Layer ที่สอดคล้องกับเวกเตอร์ w_j โดยที่ $w_j \equiv (w_{j1},\ldots,w_{jM})$ ของ Adaptive Weight หรือ LTM traces โดยกำหนดให้ weight ดังกล่าวมีค่าเท่ากับ

$$w_{j1} = \ldots = w_{jM} = 1 \qquad \ldots\ldots (1)$$

❖ **Parameters**

Fuzzy ART จะมีค่า parameter ดังนี้

- Choice Parameter (α) ต้องมีค่ามากกว่า 0

$$\alpha > 0$$

- Learning Parameter (β) หรือ Learning Rate จะช่วยให้ Prototype W_j ไม่มีการเปลี่ยนแปลงอย่างรวดเร็วจนเกินไป โดยมีค่าตั้งแต่ 0 ถึง 1

$$\beta \in [0,1]$$

- Vigilance Parameter (ρ) คือ ค่าความคล้ายคลึง ใช้ในการบอกว่าข้อมูลมีระยะห่างเท่าใดจึงจะอยู่ในกลุ่มข้อมูลเดียวกัน ซึ่งมีค่าตั้งแต่ 0 ถึง 1

$$\rho \in [0,1]$$

2. Choice

แต่ละ input (I) และ category j กำหนดให้ Choice Function (T_j) หาได้จาก

$$T_j(I) = \frac{|I \wedge w_j|}{\alpha + |w_j|} \qquad \ldots\ldots (2)$$

โดยที่ fuzzy AND operator ($\wedge$) นิยามว่า

$$(x \wedge y)_i \equiv \min(x_i, y_i), \qquad \ldots\ldots (3)$$

และ norm $\|\cdot\|$ หาได้จากสูตร

$$|x| \equiv \sum_{i=1}^{M} |x_i| \qquad \text{..... (4)}$$

กำหนดให้ Tj แทน $T_j(I)$ ในสมการ (2) เมื่อ Input (I) ถูกกำหนดแน่นอน และมี Category Choice ที่ชี้ด้วย J โดย

$$T_J = \max\{T_j : j = 1,..., N\}. \text{..... (5)}$$

ถ้ามีค่า T_j ที่มากที่สุดมากกว่า 1 ค่าแล้ว category (j) ที่มีค่าดัชนีน้อยที่สุดจะถูกเลือก

3. Match

Resonance จะเกิดขึ้นเมื่อฟังก์ชันที่จับคู่กัน (Match Function) ของหมวดหมู่ (Category) ที่เลือกมีค่ามากกว่า Vigilance

$$\frac{|I \wedge W_J|}{|I|} \geq \rho \qquad \text{..... (6)}$$

และจะเกิดการจับคู่กันอย่างไม่เหมาะสม ถ้า

$$\frac{|I \wedge W_J|}{|I|} < \rho \qquad \text{..... (7)}$$

ค่าของ Choice Function T_j จะถูกกำหนดให้มีค่าเป็น -1 ในช่วงเวลาของ Input Presentation เพื่อป้องกันเรื่อง Persistent Selection ในระหว่างการ Search ค่าดัชนี J อันใหม่ที่ถูกเลือกโดยสมการ (5) และกระบวนการ Search จะดำเนินต่อไปจนกระทั่งสามารถเลือกค่า J ที่เหมาะสมได้จากสมการ (6)

4. Adaptation

Weight Vector จะถูก update ตามสมการ

$$W_j^{(new)} = \beta(I \wedge w_j^{(old)}) + (1 - \beta)w_j^{(old)} \qquad \text{..... (8)}$$

3.3 ขั้นตอนการทดลอง

ขั้นตอนที่ 1 : คัดแยกส่วนประกอบเดิมออกจากโปรแกรมต้นฉบับที่เราต้องการศึกษา

ขั้นตอนที่ 2 : หาความสัมพันธ์ระหว่างส่วนประกอบเดิมที่คัดแยกได้

ขั้นตอนที่ 3 : จัดเรียงส่วนประกอบเดิม โดยสร้างตารางเมตริกซ์ของข้อมูล โดยพิจารณาว่าแต่ละฟังก์ชันสอดคล้องกับความสัมพันธ์ใดบ้างที่ได้กำหนดไว้ในขั้นตอนที่ 2 ซึ่งถ้าข้อมูลสอดคล้องกับความสัมพันธ์ที่กำหนด ให้มีค่าเท่ากับ 1 แต่ถ้าข้อมูลไม่สอดคล้องกับเงื่อนไขที่กำหนด ให้มีค่าเท่ากับ 0

ขั้นตอนที่ 4 : คำนวณค่าความคล้ายคลึง (Similarity) ระหว่างฟังก์ชัน

$$sim(x,y) = \frac{\sum(xy)}{\sqrt{\sum x^2} \times \sqrt{\sum y^2}}$$

ขั้นตอนที่ 5 : นำค่าความคล้ายคลึง (Similarity) ที่คำนวณได้ไปทำการจัดกลุ่มข้อมูลด้วยโปรแกรมฟัซซี่ เออาร์ที (Fuzzy ART) โดยกำหนดค่า Vigilance และ Learning Rate ต่าง ๆ กัน

ขั้นตอนที่ 6 : ผลลัพธ์ที่ได้จะสามารถบอกได้ว่าแต่ละฟังก์ชันสามารถจัดกลุ่มในลักษณะใด

ขั้นตอนที่ 7 : นำผลที่ได้จากขั้นตอนที่ 6 มาหาค่า Dunn like Indices ซึ่งจะทำให้ทราบว่าค่า Vigilance และ Learning Rate ที่เหมาะสมมีค่าเท่าไหร่ โดยแต่ละ Cluster จะแยกกันอย่างชัดเจนเมื่อระยะห่างระหว่าง Cluster มีค่ามาก และ เส้นผ่านศูนย์กลางของแต่ละ Cluster มีค่าน้อย ดังนั้นเมื่อนำมาหาค่า Dunn like Indices โดยมีสูตรดังนี้

$$D = \min_{i=1\ldots n_c}\left\{\min_{j=i+1\ldots n_c}\left(\frac{d(c_i,c_j)}{\max_{k=1\ldots n_c}(diam(c_k))}\right)\right\}$$

โดยที่ $$d(c_i,c_j) = \min_{x\in c_i, y\in c_j}\{d(x,y)\}$$

และ $$diam(c_i) = \max_{x,y\in c}\{d(x,y)\}$$

ขั้นตอนที่ 8 : ตรวจสอบว่ามีความสัมพันธ์ในลักษณะของ Inheritance โดย

8.1 จัดกลุ่มของวัตถุให้อยู่ในรูปของระดับที่แตกต่างกัน โดยขึ้นอยู่กับความถี่ในการถูกเรียกใช้งาน

8.2 วัตถุใดที่มีความถี่ในการถูกเรียกใช้งานสูงสุดจะอยู่ในระดับที่ 0 (Level 0) และวัตถุใดที่มีความถี่ในการถูกเรียกใช้งานน้อยกว่าก็จะอยู่ในระดับที่ต่ำลงมา เช่น ระดับที่ 1(Level 1) เป็นต้น

8.3 สำหรับความสัมพันธ์แบบพ่อกับลูก (Parent-Child) นั้นจะพิจารณาจากความสัมพันธ์ระหว่างระดับนั้น กับระดับที่อยู่สูงขึ้นไป กล่าวคือ จะคำนวณหาค่าความคล้ายคลึง (Similarity) ของแต่ละวัตถุที่อยู่ในระดับที่ i กับทุกวัตถุที่อยู่ในระดับที่ i-1 โดยวัตถุ t ในระดับที่ i จะกลายเป็น ลูก (Child) ของระดับที่ i-1 ก็ต่อเมื่อวัตถุในระดับที่ i มีค่าความใกล้เคียงกับวัตถุในระดับที่ i-1 มากที่สุด ดังแสดงในภาพที่ 3.4 โดยเส้นลูกศรที่บางจะแสดงลักษณะการถูกเรียกใช้งาน และเส้นลูกศรที่หนา จะแสดงถึงความสัมพันธ์แบบพ่อกับลูก