อัลกอริทึมการเรียนรู้แบบใหม่สำหรับการจำแนกกลุ่มด้วยโครงข่ายประสาทเทียมโดยใช้พีซีและอีโวลูชั่นนารี่อัลกอริทึม

A NEW LEARNING ALGORITHM FOR NEURAL NETWORK CLASSIFIER USING FUZZY AND EVOLUTIONARY ALGORITHMS

อัศวิน สีเงิน

ASAVIN MEENGEN

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต

สาขาวิชาเทคโนโลยีสารสนเทศ

บัณฑิตวิทยาลัย

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ. 2548

ISBN 974-15-1941-9

อัลกอริทึมการเรียนรู้แบบใหม่สำหรับการจำแนกกลุ่มด้วยโครงข่ายประสาทเทียมโดยใช้ฟัซซี่และอีโวลูชั่นนารี่อัลกอริทึม

A NEW LEARNING ALGORITHM FOR NEURAL NETWORK CLASSIFIER USING FUZZY AND EVOLUTIONARY ALGORITHMS

อัศวิน มีเงิน

ASAVIN MEENGEN

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต

สาขาวิชาเทคโนโลยีสารสนเทศ

บัณฑิตวิทยาลัย

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ.2548

ISBN 974-15-1941-9

A NEW LEARNING ALGORITHM FOR NEURAL NETWORK CLASSIFIER USING FUZZY AND EVOLUTIONARY ALGORITHMS

ASAVIN MEENGEN

A THESIS SUBMITTED IN PARTIAL FULFILLMENT

OF THE REQUIREMENT FOR THE DEGREE OF

MASTER OF SCIENCE IN INFORMATION TECHNOLOGY

SCHOOL OF GRADUATE STUDIES

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

2005

ISBN 974-15-1941-9

หัวข้อวิทยานิพนธ์	**อัลกอริทึมการเรียนรู้แบบใหม่สำหรับการจำแนกกลุ่มด้วยโครงข่ายประสาทเทียมโดยใช้ฟัซซี่และอีโวลูชั่นนารี่อัลกอริทึม**
นักศึกษา	นายอัศวิน มีเงิน
รหัสประจำตัว	44067427
ปริญญา	วิทยาศาสตรมหาบัณฑิต
สาขาวิชา	เทคโนโลยีสารสนเทศ
พ.ศ.	2548
อาจารย์ผู้ควบคุมวิทยานิพนธ์	รศ.ดร.อาริต ธรรมโน

บทคัดย่อ

วิทยานิพนธ์ฉบับนี้นำเสนอวิธีการจำแนกกลุ่ม ด้วยอัลกอริทึมการเรียนรู้แบบใหม่สำหรับการจำแนกกลุ่มด้วยโครงข่ายประสาทเทียมโดยใช้ฟัซซี่และอีโวลูชั่นนารี่อัลกอริทึม ซึ่งประยุกต์ใช้วิธีการจัดกลุ่มแบบ Fuzzy C-Mean Clustering และ Genetic algorithm เข้าด้วยกัน เพื่อหาจุดศูนย์กลางและส่วนเบี่ยงเบนมาตรฐานภายในชั้นซ่อนของโครงข่ายประสาทเทียม โดยอัลกอริทึมที่นำเสนอได้ใช้วิธีการจัดกลุ่มแบบ Fuzzy C-Mean Clustering ในการกำหนดจุดศูนย์กลางเริ่มต้นของแต่ละกลุ่มก่อน และใช้การวัดระยะทางแบบ Euclidian ในการกำหนดส่วนเบี่ยงเบนมาตรฐาน จากนั้นจึงใช้ Genetic algorithm ทำการปรับปรุงจุดศูนย์กลางและส่วนเบี่ยงเบนมาตรฐานที่ได้อีกครั้งเพื่อให้โครงข่ายประสาทเทียมสามารถจำแนกกลุ่มได้อย่างมีประสิทธิภาพ ในส่วนผลการทดลองได้นำไปเปรียบเทียบกับโครงข่ายประสาทเทียมแบบแพร่กระจายย้อนกลับ และโครงข่ายประสาทเทียมแบบเรเดียลเบสิกฟังก์ชั่น

Thesis	A New Learning Algorithm for Neural Network Classifier using Fuzzy and Evolutionary Algorithms
Student	Mr.Asavin Meengen
Student ID	44067427
Degree	Master of Science
Programme	Information Technology
Year	2005
Thesis Advisor	Assoc.Prof.Dr.Arit Thammano

ABSTRACT

This thesis proposes A New Learning Algorithm for Neural Network Classifier using Fuzzy and Evolutionary Algorithms which apply concepts of Fuzzy C-Mean clustering (FCM) and Genetic Algorithms (GAs) to construct centers and standard deviations in hidden layer of Neural Network. The proposed algorithm employs FCM to initial centers of each class and Euclidian distance is used to find standard deviation of each center. After that, those centers and their standard deviations are improved by GAs. The experimental results are compared against Back-propagation and Radial Basis Function Neural Networks.

กิตติกรรมประกาศ

วิทยานิพนธ์เล่มนี้สามารถสำเร็จลุล่วงได้ด้วยความกรุณาจากอาจารย์ที่ปรึกษา รศ.ดร.อาริต ธรรมโน ที่คอยให้คำปรึกษาและให้ความช่วยเหลือ ตลอดจนชี้แนะวิธีการแก้ไขปัญหา ให้แก่ข้าพเจ้า

ขอขอบคุณ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ที่ได้ประสิทประสาทวิชาให้ข้าพเจ้าตลอดจนห้องสมุดสถาบันและห้องสมุดคณะต่างๆ ที่ให้ข้าพเจ้าสามารถค้นคว้าหาข้อมูลความรู้ในด้านต่างๆ

สำหรับคุณงามความดีอันใดที่เกิดจากวิทยานิพนธ์ฉบับนี้ ข้าพเจ้าขอมอบให้กับบิดาและมารดาของข้าพเจ้า ตลอดจนครูบาอาจารย์ที่เคารพทุกท่านที่อบรมสั่งสอนข้าพเจ้า

อัศวิน มีเงิน

สารบัญ

สารบัญ (ต่อ)

สารบัญ (ต่อ)

สารบัญตาราง

สารบัญรูป

สารบัญรูป (ต่อ)

บทที่ 1

บทนำ

1.1 ความเป็นมาของปัญหา

ปัจจุบันได้มีความพยายามในการพัฒนาให้เครื่องจักร สามารถทำงานที่ใกล้เคียงกับความสามารถของมนุษย์มากขึ้น เช่น การจดจำใบหน้า, การจดจำเสียง, การจดจำลายนิ้วมือ หรือแม้กระทั้งการคัดแยกวัตถุต่างๆ ซึ่งงานเหล่านี้จะอาศัยหลักการในการ Classification หรือการจัดกลุ่มของข้อมูล จึงได้มีการคิดค้นวิธีการ Classification ด้วยวิธีการต่างๆขึ้นมาหลากหลายวิธีซึ่งโครงข่ายประสาทเทียมก็เป็นหนึ่งในนั้น

ได้มีงานวิจัยมากมายที่กล่าวถึงการใช้โครงข่ายประสาทเทียมในงาน Classification จึงปรากฏโครงข่ายประสาทเทียมแบบใหม่ขึ้นมาอยู่ตลอดเวลา ทั้งแบบมีผู้ฝึกสอน (Supervised learning) และแบบเรียนรู้ด้วยตนเอง (Unsupervised learning) โดยที่โครงข่ายประสาทเทียมมักจะประกอบไปด้วย Layer และ Node ของแต่ละ Layer และจะใช้ข้อมูลที่เรียกว่า Training data ในการฝึกฝนเพื่อใช้ในการปรับปรุงพารามิเตอร์ต่างๆให้โครงข่ายสามารถ Classify ได้อย่างถูกต้อง

ในการสร้างโครงข่ายประสาทเทียมเพื่อใช้ในงาน Classification (Neural Network Classifier) โดยทั่วไปจะใช้โครงข่ายประสาทเทียมแบบมีผู้ฝึกสอน ซึ่งโครงข่ายที่นิยมใช้และเป็นที่รู้จักกันดีเช่น Back-propagation, Radial Basis Function และ Probabilistic Neural Networks เป็นต้น ซึ่งโครงข่ายประสาทเทียมเหล่านี้จำนวนของ Node ใน Hidden layer จะไม่มีการเปลี่ยนแปลงผู้ฝึกสอนจำเป็นจะต้องกำหนดจำนวนของ Hidden node ขึ้นมาเอง ซึ่ง Hidden node นี้คือตัวแทนของ Cluster หรือก็คือจุด Center จึงมีการนำวิธีการเรียนรู้ด้วยตนเองมาประยุกต์ใช้ในการหาจำนวนของ Hidden node ที่เหมาะสมกับ Training data ได้ด้วยตัวเอง โดยถ้าหากไม่สามารถจัด data ให้เข้ากับ Cluster ใดได้ก็จะสร้าง Hidden Node หรือ Cluster ขึ้นมาใหม่ แต่เนื่องจากการเรียนรู้แบบนี้จะไม่สามารถระบุคำตอบที่ถูกต้อง (ระบุ Class) ให้กับแต่ละ Cluster ได้อย่างชัดเจน ทำให้ในระหว่างการเรียนรู้ data ที่ควรอยู่ใน Cluster เดียวกันอาจไปปรากฏอยู่ใน Cluster อื่นเป็นต้น จึงจำเป็นต้องมีการปรับปรุงค่าน้ำหนัก (Weight) ในชั้นระหว่าง Hidden กับ Output layer เพื่อให้น้ำหนักความน่าเชื่อถือของแต่ละ Cluster ในแต่ละ Class ดังนั้นหากสามารถหา Center ที่เป็นตัวแทนที่ดีของแต่ละ Cluster ในแต่ละ Class ได้อย่างเหมาะสมก็จะทำให้โครงข่ายประสาทเทียมสามารถทำการ Classify ได้ดีมากยิ่งขึ้น

ในวิทยานิพนธ์ฉบับนี้จึงรวบรวมแนวคิดและวิธีการของ Fuzzy C-Mean Clustering และ Genetic Algorithms เข้าด้วยกันเพื่อสร้างโครงข่ายประสาทเทียมที่สามารถปรับเปลี่ยนจำนวนและตำแหน่งของ Center ได้ด้วยตัวเอง โดยจะนำ Fuzzy C-Mean Clustering มาใช้ในการหาจุด Center หรือตัวแทนของ Cluster ในแต่ละ Class ให้กับโครงข่ายประสาทเทียมในตอนต้น จากนั้นจะนำ Genetic Algorithms มาใช้ในการปรับเปลี่ยนตำแหน่งและจำนวนของจุด Center ที่ได้จาก Fuzzy C-Mean Clustering ให้เหมาะสมกับข้อมูลที่นำมาฝึกฝนกับโครงข่ายประสาทเทียม

1.2 วัตถุประสงค์ของการศึกษา

1. เพื่อพัฒนาโครงข่ายประสาทเทียมสำหรับการ Classification โดยใช้แนวคิดของ Fuzzy C-Mean Clustering และ Genetic Algorithms
2. เพื่อเปรียบเทียบผลในการ Classification ด้วยวิธี Neural Network Classifier using Fuzzy and Evolutionary Algorithms กับการ Classification ด้วยวิธีการอื่น

1.3 ขอบเขตการวิจัย

1. กลุ่มตัวอย่างและกลุ่มทดสอบจะต้องเป็นข้อมูลที่สามารถจำแนกกลุ่มได้
2. พารามิเตอร์ของกลุ่มตัวอย่างและกลุ่มทดสอบจะต้องเป็นค่าจำนวนจริง (Real) เท่านั้น
3. การทำงานจะอยู่ในระบบ Off-line

1.4 ขั้นตอนของการศึกษา

1. ศึกษาการ Classification โดยใช้โครงข่ายประสาทเทียม
2. ศึกษาวิธีการ Clustering โดยวิธี Fuzzy C-Mean Clustering
3. ศึกษาแนวคิด (Algorithm) ของ Genetic Algorithms
4. ศึกษาและทดลองปรับแนวคิดทั้งหมดมาใช้ร่วมกัน
5. สร้างโครงสร้างและกระบวนการ Classification ตามวิธีที่ได้ศึกษามา
6. ทดสอบโครงสร้างและกระบวนการ Classification กับข้อมูลแบบต่างๆ
7. สรุปและอภิปรายผลการทดลอง

บทที่ 2

ทฤษฎีที่เกี่ยวข้อง

2.1 การจำแนกกลุ่ม

ในปัจจุบันมีความนิยมนำวิธีการ Classification ไปประยุกต์ใช้กับงานในด้านต่างๆมากมาย ทำให้เกิดทฤษฎีและแนวคิดใหม่ๆขึ้นเพื่อใช้ในการ Classification เช่น ทฤษฎีเบย์เชียลดีซิชั่น (Bayesian Decision Theory), แบบจำลองฮิทเดนมาคอฟ (Hidden Markov Model) และโครงข่ายประสาทเทียม (Neural Networks) เป็นต้น

มีงานวิจัยอยู่จำนวนมาก ที่ให้ความสนใจในการออกแบบแนวคิดในการ Classification เนื่องจาก คุณภาพของการจำแนกกลุ่มที่ดีนั้นมีความสำคัญกับงานในด้านต่างๆ ที่ต้องการระบุความน่าจะเป็นในแต่ละรูปแบบที่เป็นไปได้ [3] เช่น การรู้จำ (Recognition), การพยากรณ์ (Forecasting) เป็นต้น ซึ่งความยากง่ายในการจำแนกกลุ่มนั้น ขึ้นอยู่กับข้อมูล (Data) ในแต่ละกลุ่ม เนื่องจากข้อมูลในกลุ่มหนึ่งอาจไปสัมพันธ์กับข้อมูลในอีกกลุ่มหนึ่ง หรือข้อมูลในกลุ่มเดียวกันอาจจะเป็นข้อมูลที่ซ้ำซ้อน (Complexity) หรืออาจจะเป็นสิ่งรบกวน (Noise) ก็ได้

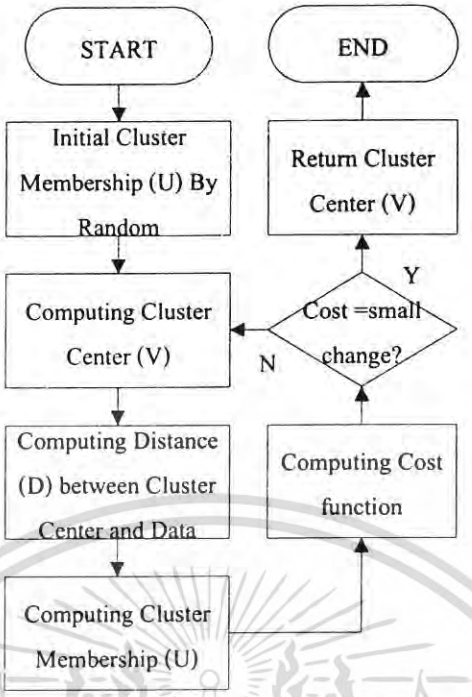
ในบทที่ 2 จะกล่าวถึงทฤษฎี Fuzzy C-Mean Clustering, โครงข่ายประสาทเทียม (Neural Networks) และขั้นตอนวิธีทางพันธุกรรม (Genetic Algorithms) ซึ่งเป็นวิธีที่จะนำมาเป็นพื้นฐานในงานวิจัยนี้

2.2 Fuzzy C-Mean Clustering

หลักการของ Fuzzy C-Mean Clustering นั้นจะเป็นการแบ่งกลุ่มของข้อมูล โดยที่แต่ละข้อมูลจะไม่ขึ้นอยู่กับกลุ่มใดกลุ่มหนึ่งโดยสมบูรณ์ แต่จะมีระดับความเป็นสมาชิก (Membership) เพื่อบ่งบอกถึงน้ำหนักของข้อมูลนั้นๆในแต่ละกลุ่ม

โดยที่แนวคิด Fuzzy C-Mean Clustering นั้นจะหาค่า Cost function [3] ที่มีค่าน้อยที่สุด เพื่อให้สามารถหาจุดศูนย์กลางของกลุ่ม (Cluster Centers) ได้อย่างแม่นยำซึ่งสามารถเขียนได้ดังสมการที่ 2.1

$$OBJ(U,V) = \sum_{j=1}^{C} \sum_{i=1}^{P} u_{ij}^{m} D_{ij}^{2} \qquad (2.1)$$



รูปที่ 2.1 Flowchart การหาค่าจุด Center ของกลุ่มด้วย FCM

1. กำหนดค่าเริ่มต้น (Initial) ให้กับระบบ ซึ่งประกอบด้วย
   1.1 จำนวน Data Pattern (P)
   1.2 จำนวน Cluster Center (C)
   1.3 ค่าตัวแปรอิสระ m โดยที่ $m \geq 1$
   1.4 ระดับความเป็นสมาชิกในกลุ่ม (Cluster Membership) ของแต่ละ Input Pattern โดยจะใช้วิธีการสุ่ม (Random) ในการกำหนดค่า
2. คำนวณหาจุดศูนย์กลางของกลุ่ม (Cluster Center) โดยใช้สมการที่ 2.5
3. คำนวณหา Distance จากสมการที่ 2.4 โดยที่ Distance คือระยะทางระหว่าง Dataกับจุดศูนย์กลางของกลุ่ม (Cluster Center)
4. คำนวณหาค่า Cluster Membership โดยใช้สมการที่ 2.3
5. คำนวณหาค่า Cost function โดยใช้สมการที่ 2.1
6. หากค่า Cost function ไม่มีการเปลี่ยนแปลง หรือเปลี่ยนแปลงเพียงเล็กน้อยให้กระทำในขั้นตอนถัดไป แต่ถ้าหากมีการเปลี่ยนแปลงให้กลับไปทำยังขั้นตอนที่ 2
7. ผลลัพธ์ที่ได้จากระบบคือค่าจุดศูนย์กลางของกลุ่ม (Cluster Center)

2.3 Neural Networks

โครงข่ายประสาทเทียม (Neural Networks) คือ รูปแบบการประมวลผลที่ได้แนวคิดมาจากการทำงานของระบบประสาทภายในสมองของมนุษย์ โดยที่การทำงานของโครงข่ายประสาทเทียมจะเลียนแบบการทำงานของสมอง เช่น การรู้จำและจำแนกรูปแบบต่างๆ การประมวลผลสัญญาณ การทำความเข้าใจในรูปภาพและเสียง การควบคุมหุ่นยนต์ ฯลฯ โดยที่การจะทำให้โครงข่ายประสาทเทียมสามารถทำงานเหล่านี้ได้จะต้องให้โครงข่ายประสาทเทียมกระทำการเรียนรู้ โดยสามารถแบ่งประเภทของการเรียนรู้ได้เป็น 2 ประเภทใหญ่ๆ คือ

1. การเรียนรู้โดยมีผู้สอน (Supervised learning) โดยที่วิธีการนี้ จะต้องทราบผลลัพธ์ที่แท้จริงของข้อมูลที่นำมาเรียนรู้เสียก่อน จากนั้นจะนำผลลัพธ์ที่ได้จากระบบ มาเปรียบเทียบกับผลลัพธ์ที่แท้จริง ปรับค่าน้ำหนักจนได้ผลลัพธ์ที่ใกล้เคียงหรือเหมือนผลลัพธ์ที่แท้จริงมากที่สุด

2. การเรียนรู้ด้วยตนเอง (Unsupervised learning) ซึ่งการเรียนรู้วิธีนี้จะไม่ทราบผลลัพธ์ที่แท้จริง ดังนั้นจะต้องทำการฝึกสอนโครงข่ายประสาทเทียมด้วยข้อมูลหลากหลายรูปแบบ โดยที่โครงข่ายประสาทเทียมจะทำการแบ่งแยกข้อมูลออกเป็น Cluster โดยสังเกตที่ระยะทางจากข้อมูลไปยังจุดศูนย์กลางของ Cluster หากมีข้อมูลที่ไม่สามารถจัดเข้ากับ Cluster ใดได้ โครงข่ายประสาทเทียมจะสร้าง Cluster ขึ้นมาใหม่

โครงข่ายประสาทเทียมจะมีแบบจำลอง (Model) อยู่หลากหลายรูปแบบ ซึ่งพอจะยกตัวอย่างมาพอสังเขปได้ดังนี้

2.3.1 Radial Basis Function Networks

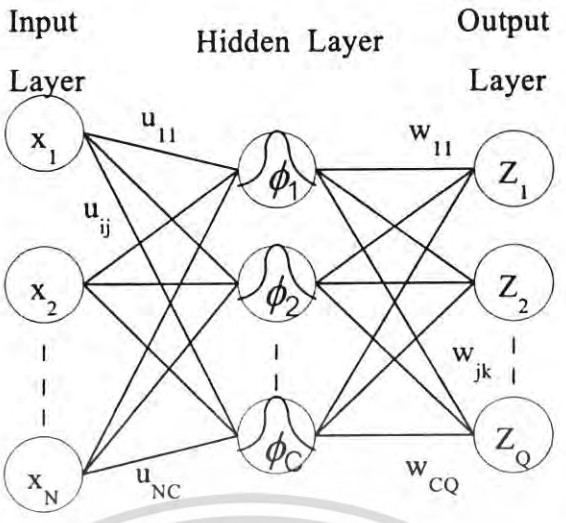
Radial Basis Function Networks (RBFN) เป็นโครงข่ายประสาทเทียมแบบหลายชั้น (Multilayer Neural Networks) โดยที่ Radial Basis Function Networks จะประกอบไปด้วยจำนวน Layer ด้วยกัน 3 Layer คือ 1. Input layer, 2. Hidden layer และ 3. Output layer ซึ่งจะแตกต่างกับโครงข่ายประสาทเทียมแบบแพร่กระจายย้อนกลับ (Back-propagation Neural Networks) ตรงที่ RBFN จะมี Hidden layer อยู่เพียงชั้นเดียวเท่านั้นส่วน Back-propagation สามารถมี Hidden layer ได้มากกว่าหนึ่ง layer

2.3.1.1 Input Layer

ในชั้น Input layer นี้ จะมีจำนวน Node ที่ทำหน้าที่รับ Data เข้ามายังโครงข่ายเท่ากับขนาด (Attribute) ของ Data เอง เมื่อกำหนดให้ $X^P = \{x_1, x_2, \ldots\ldots, x_N\} \in R$

โดยที่ N หมายถึงขนาดของ Data (Attribute)

P หมายถึงจำนวน Pattern ของ Data (Input Pattern)



รูปที่ 2.2 โครงสร้างของ Radial Basis Function Networks

2.3.1.2 Hidden Layer และการหาจุด Center

ในส่วนของ Hidden layer จะประกอบไปด้วย Radial Basis Function อาจกล่าวได้ว่า ในส่วนนี้เป็นหัวใจหลักของ Radial Basis Function Networks ซึ่ง Radial Basis Function นั้นมี Function ต่างๆมากมายเช่น

1. Gaussian Function

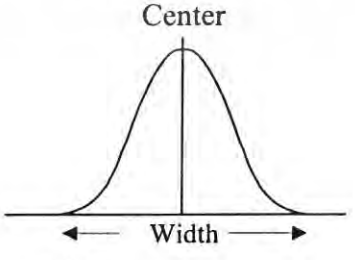
$$\varphi(r)=\exp(-\frac{r^2}{\sigma^2}) \quad ; \sigma>0 \tag{2.6}$$

2. Multi-Quadric Function

$$\varphi(r)=(r^2+\sigma^2)^{1/2} \quad ; \sigma>0 \tag{2.7}$$

3. Cubic Function

$$\varphi(r) = r^3 \tag{2.8}$$



รูปที่ 2.3 Gaussian Function

โดยทั่วไป Radial Basis Function ที่เลือกใช้ในส่วน Hidden Layer นี้ จะเลือกใช้ Gaussian Function เนื่องจากจะสังเกตได้ว่า Gaussian Function นั้นจะให้ความสำคัญกับข้อมูลใดๆ ก็ตามที่มีค่าใกล้เคียงกับจุด Center มากที่สุด และจะลดลั่งกันไปตามระยะ โดยที่รัศมีของฐาน (ความกว้างของฐาน) จะเท่ากับค่า σ

เมื่อพิจารณาสมการที่ 2.6 จะพบว่าสมการ Gaussian สามารถเขียนใหม่ให้อยู่ในรูปแบบที่เหมาะสมสำหรับใช้งานใน Radial Basis Function Networks ได้ดังสมการต่อไปนี้

$$\varphi_j(X)=\exp\left(-\left\|X-v^j\right\|^2 \Big/ \sigma_j^2\right) \qquad ; j=\{1,2,.....,C\} \tag{2.9}$$

โดยที่ C หมายถึงจำนวนจุด Center

v^j หมายถึงจุด Center ของกลุ่มที่ j

σ_j หมายถึงค่าเบี่ยงเบนมาตรฐาน (Standard deviation) ของกลุ่มที่ j

เนื่องจากจำนวนจุด Center และรัศมี จะมีจำนวนเท่ากันกับจำนวน Node ในชั้น Hidden Layer ดังนั้นการเลือกจุด Center และรัศมีที่เหมาะสมจะสามารถลดจำนวน Node ในชั้น Hidden Layer ลงได้ จึงมีวิธีในการหาค่าจุด Center และรัศมีอยู่มากมายดังที่จะยกตัวอย่างดังต่อไปนี้

1. การเลือกข้อมูลทั้งหมด

วิธีการเลือกข้อมูลทั้งหมดนี้จะเลือกทุก Input Pattern มาเป็นจุด Center วิธีการนี้ไม่เป็นที่นิยม เนื่องจากโครงข่ายที่ได้จะให้ผลลัพธ์ที่ดีก็ต่อเมื่อข้อมูลทดสอบ (Test data) ต้องคล้ายกับข้อมูลการเรียน (Training data) เท่านั้น

$$v^C = X^P \qquad ; C = P \tag{2.10}$$

โดยที่ P คือจำนวน Input Pattern

C คือจำนวนของจุด Center หรือจำนวนของ Hidden Node

v^C คือจุด Center เมื่อ $v^C=\{v_1,v_2,.....,v_N\}$

2. Fixed centers selected at random

วิธีการเลือกข้อมูลบางส่วนโดยการสุ่ม (Fixed centers selected at random) นี้จะเลือกบางส่วนจาก Input Pattern มาเป็นจุด Center โดยจะเลือกมาเท่ากับ C ตัวด้วยวิธีการสุ่ม ซึ่งสมการที่ใช้ในวิธีนี้จะดัดแปลงจากสมการที่ 2.9

$$\varphi_j(X)=\exp\left(-\frac{C}{d_{max}^2}\left\|X-v^j\right\|^2\right) \quad ; v^j \subset X^P \tag{2.11}$$

โดยที่ C คือจำนวนจุด Center ที่ต้องการ

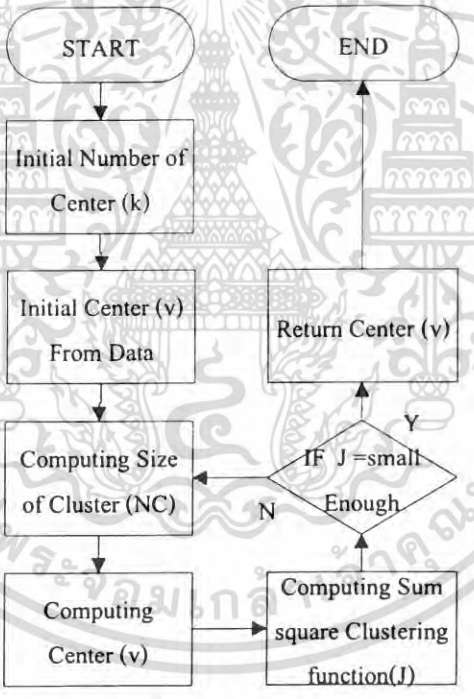
d_{max} คือระยะทาง (Distance) ระหว่างจุด Center ที่อยู่ไกลกันที่สุด ซึ่งโดยทั่วไปจะหาระยะทางโดยใช้ Euclidian Distance

X คือ Data เมื่อ $X=\{x_1,x_2,.....,x_N\}$

v คือจุด Center

3. k-Mean Clustering

การจัดกลุ่มแบบค่าเฉลี่ย k ตัว (k-Mean Clustering) นี้เป็นวิธีการที่ดีกว่าวิธีการ 1 และ 2 เนื่องจากเป็นวิธีในการหาจุด Center โดยสังเกตจากความสัมพันธ์และการกระจายตัวของข้อมูลและการหาค่าเฉลี่ย



รูปที่ 2.4 Flowchart การหาค่าจุด Center โดยวิธี k-Mean Clustering

จากรูปที่ 2.4 สามารถอธิบายเป็นขั้นตอนได้ดังนี้

1. กำหนดค่า k หรือจำนวนจุด Center ที่ต้องการ

2. กำหนดค่าจุด Center (v) ขึ้นมาเท่ากับ k ตัว โดยทั่วไปจะเลือกเอามาจาก Data ที่ใช้ในการเรียนรู้ (Training data) นั้นเอง

3. คำนวณหาขนาดของกลุ่ม (NC) ของแต่ละจุด Center ซึ่งวิธีการนี้อาจใช้ Euclidian Distance หาระยะทางจากจุด Center ไปยังทุกๆ Data และตัดสินใจเลือกสมาชิกจากระยะทางที่ใกล้ที่สุด ซึ่งจำนวนสมาชิกก็คือขนาดของกลุ่มนั้นเอง

4. เมื่อทราบขนาดของกลุ่มเป็นที่เรียบร้อย ทำการคำนวณหาค่า Center (v) ใหม่ โดยใช้วิธีหาค่าเฉลี่ยซึ่งแสดงในสมการที่ 2.12

$$v^j = \left(\frac{1}{NC_j}\right) \sum_{P \in S_j} X^P \qquad ; j=\{1, 2, \ldots, C\} \tag{2.12}$$

เมื่อกำหนดให้ $X^P = \{x_1, x_2, \ldots, x_N\} \in R$

โดยที่ X^P หมายถึง Input Pattern

S หมายถึงสมาชิกภายในกลุ่มของจุด Center

NC หมายถึงจำนวนสมาชิกภายในกลุ่มของจุด Center

v หมายถึงจุด Center

5. คำนวณหาความเหมาะสมของจุด Center ที่คำนวณได้ โดยพิจารณาจาก การหาผลรวมกำลังสอง (Sum squared clustering) ที่น้อยที่สุด

$$J = \sum_{j=1}^{C} \sum_{P \in S_j} \left\| X^P - v^j \right\|^2 \tag{2.13}$$

โดยที่ X^P หมายถึง Input Pattern

S หมายถึงสมาชิกภายในกลุ่มของจุด Center

C หมายถึงจำนวนจุด Center

v หมายถึงจุด Center

6. เมื่อคำนวณค่า J แล้วยังไม่ได้ค่าที่เหมาะสม ให้กลับไปทำขั้นตอนที่ 3 จนกว่าได้ค่าที่เหมาะสม (ค่า J มีการเปลี่ยนแปลงน้อย) แสดงว่าจุด Center ที่ได้คือจุด Center ที่เหมาะสม

4. EM algorithm

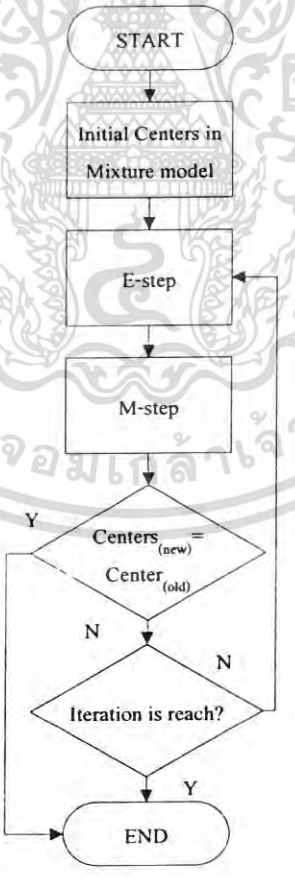
EM (Expectation Maximization) algorithm เป็นวิธีการ Clustering วิธีการหนึ่ง ซึ่ง การ Clustering นั้นเป็นการจัดกลุ่มข้อมูลโดยไม่มีผู้ฝึกฝน (Unsupervised learning) โดยตัว EM algorithm จะคล้ายๆกับ FCM แบ่งเป็นขั้นตอนหลักๆ 2 ขั้นตอนคือ E-step (Expectation-step) และ M-step (Maximization-step) ซึ่ง EM algorithm พยายามหาความเหมาะสมระหว่างข้อมูล (Data) กับ Mixture model ให้มากที่สุด โดยมักจะใช้ Gaussian function เป็น Mixture model ดังสมการที่ 2.14

$$\exp\left(-\left\|X-v^{j}\right\|^{2} \Big/ \sigma_{j}^{2}\right) \qquad ; j=\{1, 2, \ldots, C\} \qquad (2.14)$$

โดยที่ C หมายถึงจำนวนจุด Center

v หมายถึงจุด Center

σ หมายถึงค่าเบี่ยงเบนมาตรฐาน (Standard deviation)



รูปที่ 2.5 Flowchart วิธีการของ EM algorithm

จากรูปที่ 2.5 เป็น Flowchart ของ EM algorithm ที่ใช้ Gaussian function เป็น Mixture model โดยสามารถอธิบายเป็นขั้นตอนได้ดังนี้

1. กำหนดจำนวน Center (C) ที่ต้องการ

2. กำหนดค่า v เริ่มต้นให้กับ Mixture mode ในทุกๆ Center ซึ่งในขั้นตอนนี้อาจจะใช้วิธีการสุ่มเลือกข้อมูลจาก Input pattern หรือสุ่มตัวเลขขึ้นมาก็ได้โดยค่า σ จะกำหนดให้เท่ากับ 1

3. E-step เป็นการหาความน่าจะเป็นของ Input pattern ที่ i จะอยู่กับ Mixture mode ที่ j ดังสมการที่ 2.15 ซึ่งจะคล้ายกับการหาค่าความเป็นสมาชิกของ FCM Clustering

$$E_{ij}=\frac{\exp(-\left\|X^i-v^j\right\|^2/\sigma_j^2)}{\sum_{r=1}^{C}\exp(-\left\|X^i-v^r\right\|^2/\sigma_j^2)} \quad ;j=\{1,2,.....,C\},\ i=\{1,2,.....,P\} \qquad (2.15)$$

โดยที่ C หมายถึงจำนวนจุด Center

P หมายถึงจำนวน Input pattern

v หมายถึงจุด Center

σ หมายถึงค่าเบี่ยงเบนมาตรฐาน (Standard deviation)

4. M-step เป็นการปรับจุด Center (v) โดยอาศัยค่า Expectation ดังสมการที่ 2.16 ใน Step นี้จะปรับให้ค่า Center อยู่ในตำแหน่งที่เหมาะสมที่สุดในกลุ่ม Input pattern ที่อยู่ใกล้ที่สุด

$$v^j=\frac{1}{P}\sum_{i=1}^{P}E_{ij}X^i \quad ;j=\{1,2,.....,C\} \qquad (2.16)$$

โดยที่ C หมายถึงจำนวนจุด Center

P หมายถึงจำนวน Input pattern

E หมายถึงค่า Expectation

5. หากจุด Center ไม่มีการเปลี่ยนแปลงหรือดำเนินมาถึงจำนวนรอบที่กำหนดให้หยุดการทำงาน หากไม่ตรงตามเงื่อนไขเหล่านี้ให้วนกลับไปทำขั้นตอนที่ 3

2.3.1.3 Output Layer

ในชั้น Output layer นี้เชื่อมต่อกับ Hidden Node ในชั้น Hidden Layer โดยแต่ละเส้นทางจะประกอบด้วยค่าน้ำหนัก (Weight) ดังแสดงในรูปที่ 2.6

ซึ่งในชั้น Output Layer นี้จะรับ Data จากชั้น Hidden Layer มาคำนวณด้วยสมการที่ 2.17 ซึ่งเป็นสมการ Linear Function เพื่อให้ทราบผลลัพธ์ (Output) ที่ได้จาก Radial Basis Function Networks

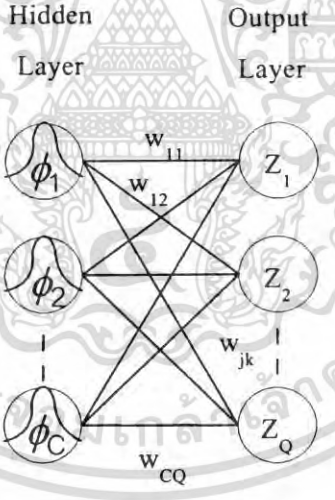
$$Z_k = \sum_{j=1}^{C} W_{jk}\varphi_j(X) \qquad ; k = \{1, 2, \ldots, Q\} \qquad (2.17)$$

โดยที่ Q คือจำนวนของ Output Node

C คือจำนวนของ Hidden Node

W คือค่า Weight ระหว่าง Hidden Node กับ Output Node

$\varphi_j(X)$ คือ Output ที่ได้จาก Hidden Node



รูปที่ 2.6 องค์ประกอบในชั้น Output Layer

2.3.1.4 การวัดค่าความผิดพลาด และการหาค่า Weight

การวัดค่าความผิดพลาด มีไว้สำหรับการปรับค่าน้ำหนัก (Weight) ให้โครงข่ายมีความสามารถจำแนกกลุ่มได้ดียิ่งขึ้น ยิ่งอัตราการผิดพลาด (Error rate) ยิ่งน้อยยิ่งจำแนกได้ดีขึ้นโดยสมการที่ใช้ในการวัดอัตราการผิดพลาดนั้นแสดงในสมการที่ 2.18

$$E(w)=\frac{1}{2}\sum_{i=1}^{P}\sum_{k=1}^{Q}(T_{ik}-Z_{ik})^2 \qquad (2.18)$$

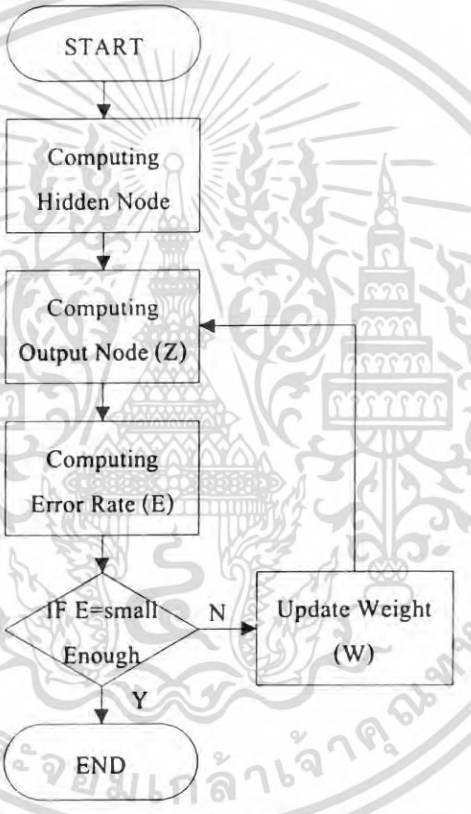
โดยที่ P คือจำนวน Input Pattern

Q คือจำนวนของ Output Node

Z คือ Output จาก Output Layer

T คือ Output ที่แท้จริง (Target)

การปรับปรุงค่าน้ำหนัก (Update weight) นั้นจะกระทำได้โดยใช้สมการที่ 2.19



รูปที่ 2.7 Flowchart ในส่วนการ Update Weight

$$W_{jk}^{new}=W_{jk}^{old}+\Delta W_{jk} \qquad ;j=\{1,2,\ldots,C\}\ \ k=\{1,2,\ldots..,Q\} \qquad (2.19)$$

โดยที่ W คือค่า Weight

C คือจำนวน Hidden Node

Q คือจำนวน Output Node

เมื่อ $$\Delta W_{jk} = -\eta \frac{\partial E}{\partial W_{jk}} \quad (2.20)$$

โดยที่ η คืออัตราการเรียนรู้ (Learning rate)

เมื่อ $$-\frac{\partial E}{\partial W_{jk}} = \sum_{i=1}^{P} (T_{ik} - Z_{ik})\varphi_j(X) \qquad ; j=\{1, 2,\ldots,C\} \; k=\{1, 2,\ldots..,Q\} \quad (2.21)$$

โดยที่ $\varphi_j(X)$ คือ Output ที่ได้จาก Hidden Layer

P คือจำนวน Input Pattern

Q คือจำนวนของ Output Node

C คือจำนวนของ Hidden Node

Z คือ Output จาก Output Layer

T คือ Output ที่แท้จริง (Target)

จากรูปที่ 2.7 แสดงวิธีการปรับปรุงค่า Weight โดยสามารถอธิบายเป็นขั้นตอนได้ดังนี้

1. คำนวณหาค่า Output ของ Hidden Node ($\varphi_j(X)$) แต่ละตัวโดยใช้สมการที่ 2.9 กับทุก Input Pattern

2. คำนวณหา Output ของ Output Layer โดยใช้สมการที่ 2.17 ซึ่ง Output ที่ได้นี้เป็น Output ของ Radial Basis Function Networks

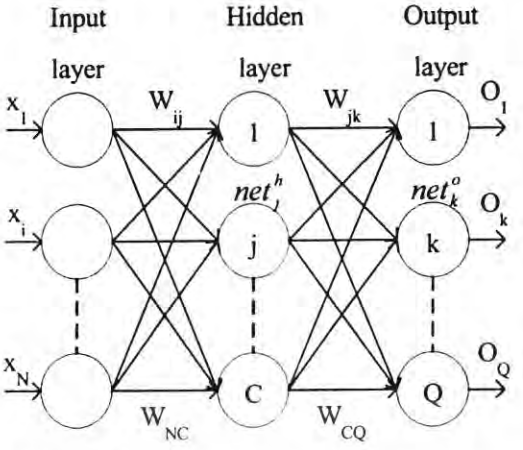
3. คำนวณหาค่า Error Rate (E) โดยใช้สมการที่ 2.18

4. หากค่า Error Rate (E) มีค่าน้อยให้จบการทำงาน

5. หากค่า Error Rate ยังมีค่ามากให้ปรับปรุง Weight โดยใช้สมการที่ 2.19 และกลับไปยังขั้นตอนที่2

2.3.2 Back-propagation neural networks

โครงข่ายประสาทเทียมแบบแพร่กระจายย้อนกลับ (Back-propagation neural networks) ซึ่งจะใช้โครงสร้างแบบ Multilayer Feedforward Neural Networks เป็นโครงสร้าง ดังแสดงในรูปที่ 2.8 Back-propagation neural networks สามารถนำไปใช้ได้ดีกับปัญหาชนิดต่างๆไม่ว่าจะเป็นการจำแนกกลุ่ม การทำนาย การประมวลผลสัญญาณเชิงเลข [4] เป็นต้น



รูปที่ 2.8 โครงสร้าง Back-propagation neural networks

หลักการคร่าวๆ ของ Back-propagation คือ ในระหว่างทำการเรียนรู้ (Learning) จะต้องเป็นการเรียนรู้แบบมีผู้สอน เมื่อป้อน Data เข้าสู่ระบบ ระบบจะทำการประมวลผลไปที่ละชั้น (layer) จนกระทั่งได้ Output ของระบบที่ Output layer จากนั้นทำการเปรียบเทียบ Output ที่ได้กับ Target ผลต่างที่เกิดขึ้นจะถูกป้อนกลับไปปรับปรุงค่า Weight โดยจะทำการปรับปรุงถอยหลังไปเรื่อยๆ จากชั้น Output Layer จนถึงชั้น Input Layer

รูปที่ 2.8 แสดงโครงสร้าง Back-propagation neural networks ซึ่งประกอบไปด้วย 3 ชั้น และค่า Weight ในชั้น Hidden Layer และ Output Layer โดยที่ในชั้น Hidden Layer นั้นในแต่ละ Node จะประกอบไปด้วย Transfer function ดังสมการที่ 2.22

$$sgm(x)=\frac{1}{1+e^{-x}} \tag{2.22}$$

โดยในส่วน Hidden node จะหาผลรวมของค่า Weight กับ Input pattern ดังสมการที่ 2.23

$$net_j^h=\sum_{i=1}^{N} X_i W_{ij} \qquad ;j=\{1\ 2,\ldots,C\} \tag{2.23}$$

หลังจากนั้นจะนำผลรวมจาก Hidden node ไปผ่าน Transfer function จากสมการที่ 2.22

$$H_j=f(net_j^h)=sgm(net_j^h) \tag{2.24}$$

โดยที่ H_j คือ Output ของ Hidden Node

C คือจำนวนของ Hidden Node

N คือขนาดของ Data หรือจำนวนของ Input Node

X คือ Input Data

W_{ij} คือค่า Weight ระหว่าง Hidden Layer กับ Input Layer

และข้อมูลจะถูกส่งไปคำนวณยังส่วนของ Output Layer โดยใช้สมการดังนี้

$$net_k^o = \sum_{j=1}^{C} H_j W_{jk} \qquad ; k = \{1, 2, \ldots, Q\} \tag{2.25}$$

หลังจากนั้นจะนำผลรวมจากแต่ละ Output node ไปผ่าน Transfer function เช่นเดียวกัน

$$O_k = f(net_k^o) = sgm(net_k^o) \tag{2.26}$$

โดยที่ O_k คือ Output ของ Output Node

Q คือจำนวนของ Output Node

C คือจำนวนของ Hidden Node

W_{jk} คือค่า Weight ระหว่าง Hidden Layer กับ Output Layer

เมื่อได้ผลลัพธ์จาก Output แล้ว นำมาหาค่า Error Rate เพื่อจะนำไปปรับปรุงค่า Weight ในแต่ละ Layer ซึ่งการปรับปรุงค่า Weight ของ Back-propagation นี้จะเปรียบเทียบความแตกต่างระหว่าง Output ที่ได้กับ Target โดยจะใช้ LMS algorithm เพื่อหาความแตกต่างนี้

$$E(W) = \frac{1}{2} \sum_{k=1}^{Q} (T_k - O_k)^2 \tag{2.27}$$

โดยที่ O_k คือ Output ของ Output Node

Q คือจำนวนของ Output Node

T_k คือ Output ที่แท้จริง (Target)

หลังจากนั้นทำการปรับปรุงค่าน้ำหนัก (Update weight) ซึ่งกระทำได้โดยใช้สมการที่ 2.28

$$W(new) = W + \Delta W \tag{2.28}$$

โดยที่ W คือค่า Weight

เมื่อ $$\Delta W = -\eta \frac{\partial E}{\partial W} \quad (2.29)$$

โดยที่ η คืออัตราการเรียนรู้ (Learning rate)

จากรูปที่ 2.8 เป็นรูปโครงสร้าง Back-propagation neural networks ซึ่งมีจำนวน Layer เท่ากับ 3 Layer และมี Weight อยู่ 2 ช่วงคือ จาก Hidden to Output และจาก Input to Hidden ซึ่งในขั้นตอนแรกจะต้องพิจารณาปรับค่า Weight จากส่วน Hidden to Output เสียก่อน

$$\Delta W_{jk} = -\eta \frac{\partial E}{\partial W_{jk}} = \eta \delta_K \frac{\partial net_k^o}{\partial W_{jk}} \quad (2.30)$$

โดยที่ W_{jk} คือค่า Weight ระหว่าง Hidden Layer กับ Output Layer

เนื่องจาก $$net_k^o = \sum_{j=1}^{C} H_j W_{jk}$$

ดังนั้น $$\frac{\partial net_k^o}{\partial W_{jk}} = \frac{\partial \sum_{j=1}^{C} H_j W_{jk}}{\partial W_{jk}} = H_j \quad (2.31)$$

โดยที่ H_j คือ Output ของ Hidden Node

C คือจำนวนของ Hidden Node

และ $$\delta_K = -\frac{\partial E}{\partial net_k^o} = (T_k - O_k) f'(net_k^o) \quad (2.32)$$

โดยที่ O_k คือ Output ของ Output Node

T_k คือ Output ที่แท้จริง (Target)

เมื่อนำสมการที่ 2.30, 2.31 และ 2.32 แทนลงไปในสมการที่ 2.28 จะได้สมการที่ใช้ในการปรับปรุงค่า Weight จาก Hidden Node ไปยัง Output Node ดังสมการที่ 2.33

$$W_{jk}(new) = W_{jk} + \eta[(T_k - O_k) f'(net_k^o) H_j] \tag{2.33}$$

โดยที่ H_j คือ Output ของ Hidden Node

O_k คือ Output ของ Output Node

T_k คือ Output ที่แท้จริง (Target)

W_{jk} คือค่า Weight ระหว่าง Hidden Layer กับ Output Layer

η คืออัตราการเรียนรู้ (Learning rate)

ในขั้นตอนถัดมาเป็นการปรับปรุง Weight ในส่วน Input to Hidden ซึ่งจะใช้หลักการเหมือนกับการปรับปรุง Weight ในส่วน Hidden to Output โดยจะใช้สมการที่ 2.28 และ 2.29 เป็นหลัก

$$\Delta W_{ij} = -\eta \frac{\partial E}{\partial W_{ij}} = \eta \delta_j \frac{\partial net_j^h}{\partial W_{ij}} \tag{2.34}$$

โดยที่ W_{ij} คือค่า Weight ระหว่าง Input Layer กับ Hidden Layer

เนื่องจาก $net_j^h = \sum_{i=1}^{N} X_i W_{ij}$

ดังนั้น $$\frac{\partial net_j^h}{\partial W_{ij}} = \frac{\partial \sum_{i=1}^{N} X_i W_{ij}}{\partial W_{ij}} = X_i \tag{2.35}$$

โดยที่ N คือขนาดของ Data หรือจำนวนของ Input Node

X คือ Input Data

และ $$\delta_j = -\frac{\partial E}{\partial net_j^h} = (\sum_{k=1}^{Q} \delta_k W_{jk}) f'(net_j^h) \tag{2.36}$$

โดยที่ W_{jk} คือค่า Weight ระหว่าง Hidden Layer กับ Output Layer

δ_k หาได้จากสมการที่ 2.32

เมื่อนำสมการที่ 2.34, 2.35 และ 2.36 แทนลงไปในสมการที่ 2.28 จะได้สมการที่ใช้ในการปรับปรุงค่า Weight จาก Input Node ไปยัง Hidden Node ดังสมการที่ 2.37

$$W_{ij}(new) = W_{ij} + \eta\left(\left[\sum_{k=1}^{Q} \delta_k W_{jk} \right] f'(net_j^h) X_i \right) \quad (2.37)$$

โดยที่ X คือ Input Data

W_{ij} คือค่า Weight ระหว่าง Hidden Layer กับ Input Layer

δ_k หาได้จากสมการที่ 2.32

Q คือจำนวนของ Output Node

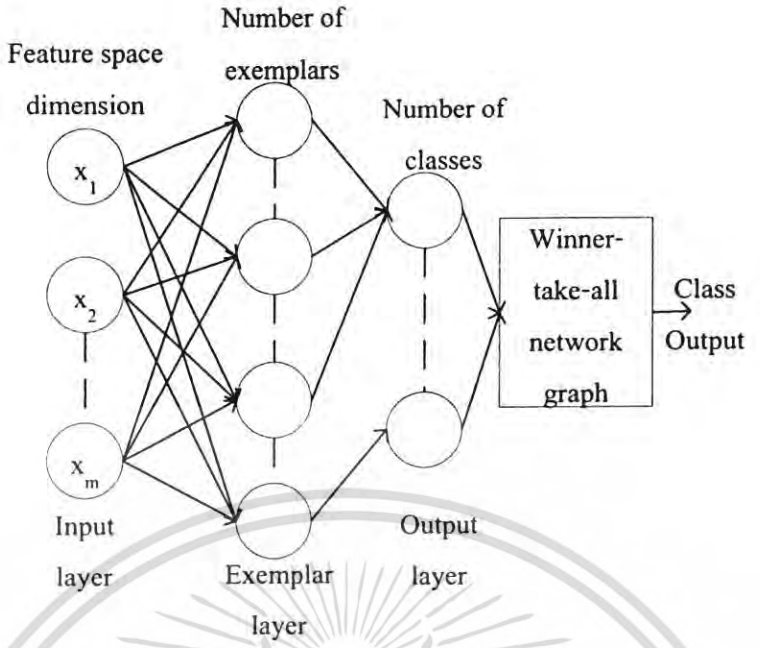
η คืออัตราการเรียนรู้ (Learning rate)

2.3.3 Probabilistic neural networks

Probabilistic neural networks เป็นโครงข่ายแบบ Multilayer Feedforward Neural Networks ชนิดหนึ่ง ซึ่ง Probabilistic neural networks สามารถถูกสร้างได้โดยแค่ผ่านการเรียนรู้ (Learning) เพียงแค่รอบเดียว โดยที่ฟังก์ชั่นในการกระทำ (Activation function) ของนิวรอน (Neural) ชนิดนี้จะใช้หลักของสถิติในการประมาณค่า Probability density functions: PDFs กับ Data ที่นำมาเรียนรู้ (Training Data) [5]

จากรูปที่ 2.9 แสดงโครงสร้างของ Probabilistic neural networks ซึ่งจะประกอบไปด้วย 3 Layer โดยที่ชั้น Input Layer จะมีจำนวน Node เท่ากับ Attribute ของ Data ส่วนในชั้น Hidden Layer จะมีจำนวน Node เท่ากับจำนวนของ Data Pattern และในชั้น Output Layer จะมีจำนวน Node เท่ากับจำนวน Class ที่ต้องการ Output ที่ได้จากชั้น Output Layer จะถูกส่งไปตัดสินใจเพื่อหาคำตอบอีกครั้งโดย Output Node ใดให้ค่าสูงสุด Node นั้นจะเป็นผู้ชนะ

ซึ่งในส่วนของ Hidden Node นั้นอาจใช้ Gaussian Function ดังสมการที่ 2.9 ก็ได้ขึ้นอยู่กับการออกแบบ ซึ่ง Probabilistic neural networks มีข้อเสียคือจำนวนหรือขนาดของ Hidden Node จะมีจำนวนมากถ้าหาก Data ที่นำมาเรียนรู้มีจำนวนมากเนื่องจากในชั้น Hidden Layer จะต้องนำ Data ทั้งหมดใน Training Data มาสร้างเป็น Hidden Node และบรรจุไว้ในชั้นนี้

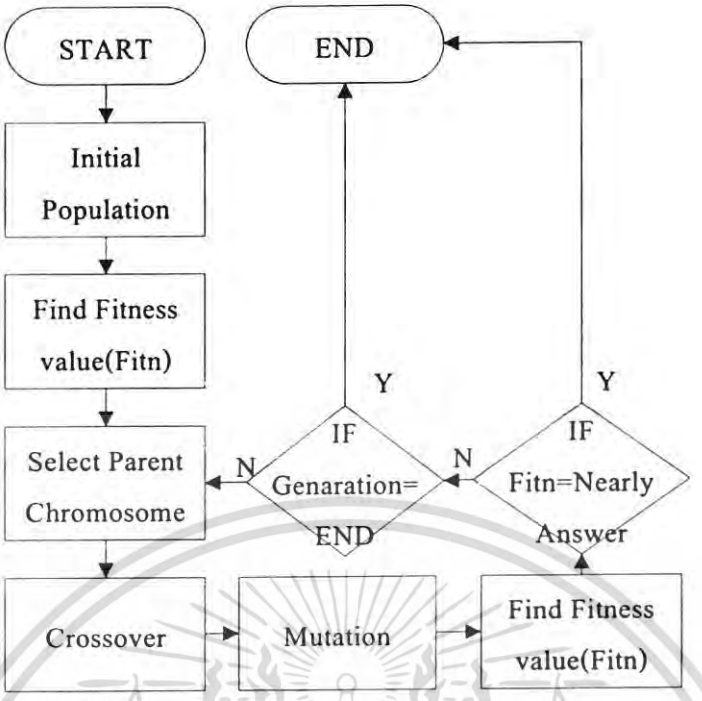


รูปที่ 2.9 โครงสร้าง Probabilistic neural networks

2.4 Genetic Algorithms

ขั้นตอนวิธีทางพันธุกรรม (Genetic Algorithms) ถูกเสนอโดย Holland [6] ซึ่งใช้แนวคิดในการคำนวณโดยใช้หลักการทางชีววิทยาของทฤษฎีพัฒนาการว่า สิ่งมีชีวิตทุกชนิดประกอบไปด้วยเซลล์ ซึ่งแต่ละเซลล์จะมีกลุ่มของโคโมโซม (Chromosomes) ที่เหมือนกัน โดยแต่ละโคโมโซมจะประกอบไปด้วยยีนส์ (Genes) ซึ่งจะสะท้อนลักษณะเฉพาะของสิ่งมีชีวิตนั้น เช่น สีตา สีผม เป็นต้น [6]

เมื่อนำโครโมโซมทั้งหมดมาประกอบกันจะเรียกว่า Genome และเรียกกลุ่มของยีนส์ใน Genome นี้ว่า Genotype หรือรหัสพันธุกรรม ซึ่งจะเปลี่ยนเป็นอวัยวะของสิ่งมีชีวิตต่อไป ตามทฤษฎีของการวิวัฒนาการสิ่งมีชีวิตที่แข็งแรงที่สุดจะมีโอกาสสืบทอดสายพันธุ์ต่อไป การผสมยีนส์ของพ่อและแม่จะทำให้เกิดลูกซึ่งคัดลอกยีนส์ของพ่อและแม่ที่ผสมกัน ทำให้เกิดสายพันธุ์ที่แข็งแรงยิ่งขึ้น บางครั้งการคัดลอกยีนส์ของพ่อแม่ไม่สมบูรณ์ อาจทำให้เกิดการผ่าเหล่า (Mutation) ในรุ่นลูก การผ่าเหล่าทำให้สิ่งมีชีวิตมีโอกาสพัฒนาสายพันธุ์ใหม่ที่ยิ่งแข็งแรงขึ้น หากการผ่าเหล่าทำให้เกิดสายพันธุ์ด้อย สายพันธุ์ด้อยจะไม่สามารถสืบทอดต่อไป [6]



รูปที่ 2.10 Flowchart กระบวนการ Genetic Algorithms

2.4.1 Initial Population

เนื่องจากข้อมูลในโลกนี้มีอยู่ด้วยกันหลายหลายชนิด ดังนั้นจึงจำเป็นที่จะต้องแปลงข้อมูลต่างๆ ให้อยู่ในรูปแบบที่เหมาะสมสำหรับใช้งานกับ Genetic Algorithms ซึ่งผลลัพธ์ที่ได้จากการแปลงจะอยู่ในรูปของโครโมโซม ตัวแปรที่ต้องการค้นหาหมายถึงยีนส์หนึ่งยีนส์เมื่อนำยีนส์มาประกอบกันจะได้โครโมโซมซึ่งก็คือผลลัพธ์ของปัญหา การแปลงทำได้หลายวิธี เช่นการแปลงเป็นเลขฐานสอง (Binary) หรือการแปลงเป็นค่าจริง (Real value) โดยจะพิจารณาจากลักษณะของปัญหา ตัวอย่างเช่น เลขฐานสองใช้สำหรับปัญหาที่มีรูปแบบคำตอบเพียงสองรูปแบบ การใช้ตัวเลขลำดับใช้สำหรับหาคำตอบของปัญหาที่ต้องการจัดอันดับ เช่น ปัญหาการเดินทางของพนักงานขาย (Traveling saleman Problem) การแปลงค่าจริงใช้สำหรับการหา Weight ที่เหมาะสมกับการคำนวณ

เมื่อได้ข้อมูลที่ถูกแปลงให้เหมาะสมกับ Genetic Algorithms แล้ว จะต้องทำการสุ่มโครโมโซมเหล่านี้ขึ้นมาเป็นประชากรเริ่มต้น (Initial Population) โดยจำนวนโครโมโซมหรือประชากรแต่ละรุ่น ขึ้นอยู่กับพารามิเตอร์ที่กำหนดขึ้นเอง ส่วนความยาวของโครโมโซมคือจำนวนตัวแปรของปัญหาโดยจะแทนด้วยจำนวนยีนส์ ซึ่งจะมีผลต่อการหาค่าคำตอบที่ดีที่สุดด้วย

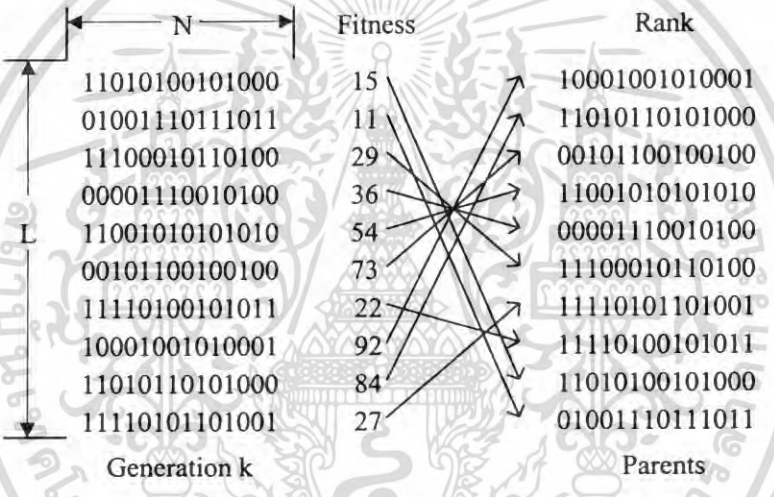
2.4.2 Fitness Function

ในส่วนของการประเมินความเหมาะสม (Fitness function) นี้จะเป็นส่วนที่วิเคราะห์ได้ยากที่สุด เนื่องจากไม่มีวิธีการที่ตายตัวในการกำหนด ซึ่งการประเมินความเหมาะสมนี้จะต้องวิเคราะห์ที่ลักษณะของปัญหาและรูปแบบของคำตอบที่ต้องการ

เหตุที่ Fitness Function มีความสำคัญเนื่องจาก Fitness Function จะให้ค่า Fitness value กับแต่ละโครโมโซม เพื่อใช้ในการตัดสินใจในการเลือก (Selection) โครโมโซม Parents หรือโครโมโซมที่จะกลายเป็นรุ่นต่อไป โดยถ้าหากโครโมโซมใดมีค่า Fitness value ต่ำก็จะถูกทิ้งไป เปิดโอกาสให้โครโมโซมที่มีค่า Fitness value สูงได้สืบทอดเป็นรุ่นต่อไป

2.4.3 Selection

หลักการทั่วไปของการคัดเลือก (Selection) ก็คือโครโมโซมที่เหมาะสมที่สุดหรือโครโมโซมที่มีค่า Fitness value สูง จะมีโอกาสที่จะถูกเลือกมากกว่าโครโมโซมที่ด้อยกว่าโดย วิธีการคัดเลือกจะมีอยู่ด้วยกันหลายวิธีเช่น Roulette-wheel selection, Local selection, Stochastic universal sampling เป็นต้น

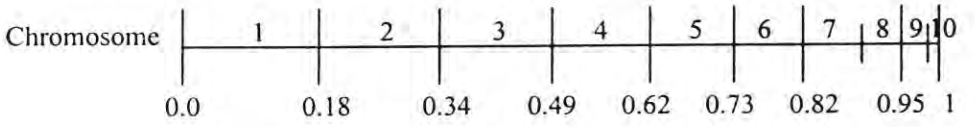


รูปที่ 2.11 ตัวอย่างการ Selection แบบจัดอันดับ

1. Roulette-wheel selection

Roulette-wheel selection จะช่วยให้โครโมโซมที่เหมาะสมที่สุดมีโอกาสถูกเลือกมากที่สุด โดยที่วิธีนี้เปรียบเสมือน การกำหนดช่องให้กับทุกโครโมโซม โดยที่ขนาดของช่องจะขึ้นอยู่กับค่าความเหมาะสม ยิ่งโครโมโซมใดมีค่าความเหมาะสมมาก โครโมโซมนั้นจะมีขนาดของช่องที่กว้างกว่าโครโมโซมที่ด้อยกว่าตน

เมื่อกำหนดช่องให้กับทุกโครโมโซมแล้ว จะทำการสุ่มเพื่อที่จะหาโครโมโซม Parent โดยจะพบว่าโครโมโซมที่มีขนาดของช่องที่กว้างกว่า ย่อมมีโอกาสถูกเลือกได้มากกว่านั้นเองดังแสดงในรูปที่ 2.12



รูปที่ 2.12 Roulette-wheel selection

จากรูปที่ 2.12 จะเรียงลำดับโครโมโซมโดยเรียงจากค่า Fitness value จากมากไปหาน้อยซึ่งโครโมโซมที่มีค่า Fitness value มากจะกว้างกว่าโครโมโซมที่มี Fitness value น้อยลดหลั่นกันไปตามลำดับ

2. Stochastic universal sampling

Stochastic universal sampling นี้จะใช้วิธีการ Roulette-wheel selection มาประยุกต์ใช้งานโดยแตกต่างกันตรงที่ การเลือกโครโมโซม Parents ซึ่งวิธี Roulette-wheel selection จะทำการสุ่มตัวเลขเพื่อเลือกโครโมโซม Parents ถ้าหากต้องการโครโมโซม Parents จำนวน k ตัว ก็จะทำการสุ่มทั้งหมด k ครั้ง แต่สำหรับวิธีการ Stochastic universal sampling นั้นจะไม่ใช้การสุ่มในการเลือกโครโมโซม Parents ทุกตัวแต่จะสุ่มเฉพาะค่าแรกเท่านั้นส่วนค่าถัดไปใช้วิธีการคำนวณจากสูตร 1/k โดยที่ k คือจำนวน Parents ที่ต้องการหาซึ่งสามารถแสดงวิธีการได้ดังรูปที่ 2.13

รูปที่ 2.13 การเลือกโดยวิธี Stochastic universal sampling

จากรูปที่ 2.13 กำหนดให้จำนวนโครโมโซม Parents เท่ากับ 6 ดังนั้นจึงนำ 1/6 ซึ่งจะได้ผลลัพธ์เท่ากับ 0.167 และนำค่านี้ไปเป็นค่าช่วงห่างระหว่างจุดที่เลือก ซึ่งผลลัพธ์ที่ได้จากรูปที่ 2.13 คือเลือกโครโมโซมที่ 1, 2, 3, 5, 6, 9 มาเป็นโครโมโซม Parents

2.4.4 Crossover

วิธีการสร้างโครโมโซมลูก (Offspring) นั้นจะเกิดจากโครโมโซม Parents โดยจะเลือกยีนส์ในโครโมโซม Parents มาสร้างเป็นโครโมโซม Offspring ซึ่งจะใช้วิธีการ Crossover ซึ่งการ Crossover นี้จะทำให้เกิดโครโมโซม Offspring ขึ้นมาสองโครโมโซม โดยที่สามารถแบ่งวิธีการ Crossover ได้ดังต่อไปนี้

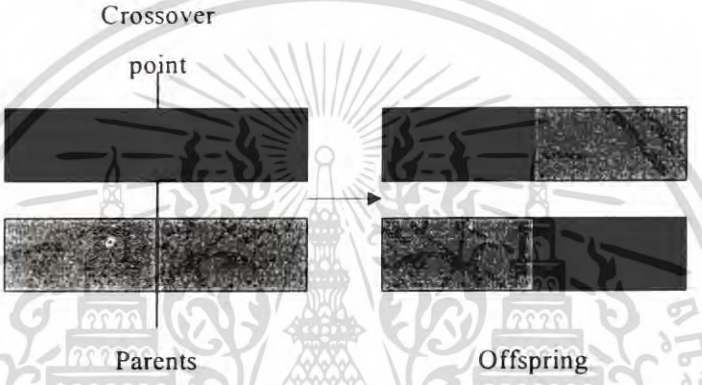
1. Single-point crossover

วิธีการ Single-point crossover นี้ จะเลือกจุดเพียงจุดเดียวในการ Crossover ซึ่งจะเลือกตำแหน่งโดยวิธีการสุ่ม

ตัวอย่าง Single-point crossover แสดงเป็นเลขฐานสองซึ่งมีจำนวนยีนส์ในแต่ละโครโมโซมให้เท่ากับ 10

Parent 1 1 0 0 1 0 0 0 0 1 1

Parent 2 1 1 1 0 1 1 0 1 0 0



รูปที่ 2.14 Single-point crossover

เลือกจุด Crossover โดยวิธีการสุ่ม ซึ่งได้ผลลัพธ์เป็นตำแหน่งที่ 5

Parent 1 1 0 0 1 0 | 0 0 1 1

Parent 2 1 1 1 0 1 | 1 0 1 0 0

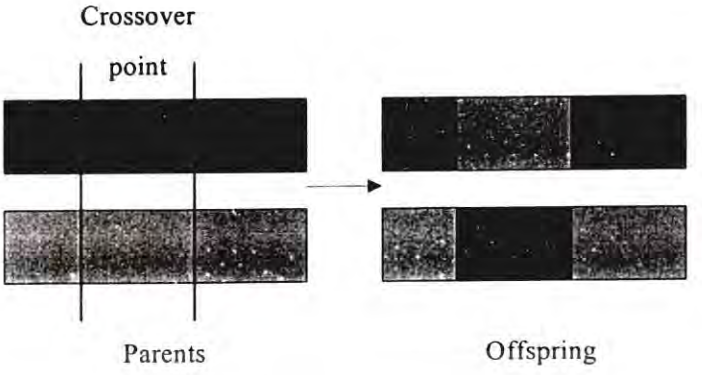
จากนั้นทำการ Crossover ดังรูปที่ 2.14 ก็จะได้โครโมโซม Offspring ซึ่งสืบทอดยีนส์มาจากโครโมโซม Parents

Offspring 1 1 0 0 1 0 1 0 1 0 0

Offspring 2 1 1 1 0 1 0 0 0 1 1

2. Multi-point crossover

วิธีการ Multi-point crossover นี้จะคล้ายกับการ Crossover แบบ Single-point crossover โดยเพิ่มจำนวนจุดใน Crossover ซึ่งจำนวนจุดในการ Crossover จะต้องเป็นพารามิเตอร์ที่กำหนดมาแล้ว ส่วนตำแหน่งในการ Crossover นั้นจะเกิดขึ้นมาจากการสุ่ม



รูปที่ 2.15 Multi-point crossover

ตัวอย่าง Multi-point crossover แสดงเป็นเลขฐานสองและขนาดของยีนส์ในโครโมโซม Parents มีขนาดเท่ากับ 12

Parent 1 1 0 0 1 0 0 0 0 1 1 0 1

Parent 2 1 1 1 0 1 1 0 1 0 0 1 1

ต้องการจุดในการ Crossover เท่ากับ 3 ดังนั้นจึงทำการสุ่มตำแหน่งมาทั้งหมด 3 ตำแหน่ง คือ จุดที่ 2 , 6 และจุดที่ 9

Parent 1 1 0 | 0 1 0 0 | 0 0 1 | 1 0 1

Parent 2 1 1 | 1 0 1 1 | 0 1 0 | 0 1 1

ทำการ Crossover โดยอ้างอิงจากรูปที่ 2.15 จะได้โครโมโซม Offspring ซึ่งสืบทอดยีนส์มาจากโครโมโซม Parents

Offspring 1 1 0 1 0 1 1 0 0 1 0 1 1

Offspring 2 1 1 0 1 0 0 0 1 0 1 0 1

2.4.5 Mutation

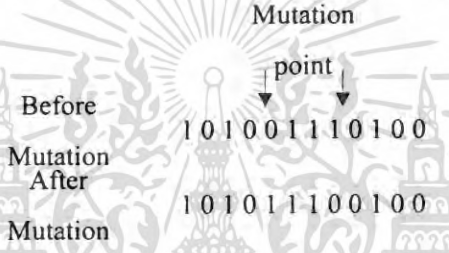
เนื่องจากโครโมโซม Offspring ที่เกิดจากโครโมโซม Parents นั้น อาจมีความคล้ายคลึงกับโครโมโซม Parents มากเกินไป ทำให้โครโมโซมในรุ่นถัดไปมีค่า Fitness value ที่ใกล้เคียงกัน และอาจทำให้ได้ผลลัพธ์ที่ไม่ดีพอเนื่องจากไม่เกิดการพัฒนาในรุ่นถัดไป

ดังนั้นจึงต้องมีกระบวนการการผ่าเหล่า (Mutation) เพื่อให้โครโมโซม Offspring แตกต่างออกไปจากเดิมแต่จะยังคงรักษาเค้าโครงของโครโมโซม Parents เอาไว้ การ Mutation นั้นมีทั้งข้อดี

และข้อเสีย ข้อดีคือโครโมโซม Offspring ที่ผ่าเหล่าอาจจะให้ค่า Fitness value ดีกว่าโครโมโซม Parents แต่ข้อเสียก็คือในบางครั้งโครโมโซม Offspring ที่ผ่าเหล่านั้นอาจเป็นโครโมโซมที่ไม่เหมาะสมจำเป็นต้องกำจัดทิ้งไป ดังนั้นอัตราการ Mutation จึงจำเป็นต้องมีค่าต่ำเพื่อไม่ให้โครโมโซม Offspring นั้น แตกต่างจากโครโมโซม Parents มากเกินไป

1. Binary mutation

Binary mutation จะใช้กับโครโมโซมที่มียีนส์เป็นข้อมูลแบบเลขฐานสอง การ Mutation นั้นจึงมีอยู่แค่ 2 ค่าเท่านั้นคือ "0" และ "1" ส่วนตำแหน่งและจำนวนของการ Mutation นั้นหาได้จากวิธีการสุ่ม รูปที่ 2.16 แสดงตัวอย่าง Binary mutation โดยจำนวนในการ Mutation เท่ากับ 2 ตรงตำแหน่ง 5 และ 8



รูปที่ 2.16 แสดงตัวอย่าง Binary mutation

2. Real value mutation

Real value mutation จะใช้กับโครโมโซมที่มียีนส์เป็นค่าจริง ซึ่งยีนส์ประเภทนี้อาจใช้ในการหาพารามิเตอร์คำตอบของปัญหา ดังนั้นการ Mutation จึงระบุเจาะจงค่าและวิธีการที่แน่นอนไม่ได้ จะต้องวิเคราะห์ชนิดของปัญหาด้วย

โดยส่วนใหญ่ตำแหน่งและจำนวนของการ Mutation จะขึ้นอยู่กับการสุ่ม แต่สำหรับค่าที่ใช้ในการ Mutation นั้นอาจหาค่าโดยใช้สมการที่ 2.38

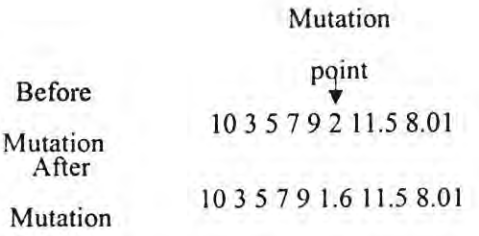
$$After = Before + (Range * Before) \tag{2.38}$$

โดยที่ After คือ ยีนส์หลังจากการ Mutation

Before คือ ยีนส์ก่อนการ Mutation

Range คือ อัตราการกลายพันธุ์ $-1 \le Range \le 1$

โดยทั่วไปค่า Range กำหนดให้มีค่าต่ำเพื่อไม่ให้โครโมโซม Offspring มียีนส์ที่แตกต่างจาก Parents มากเกินไป ค่า Range อาจหามาได้โดยวิธีการสุ่ม



รูปที่ 2.17 แสดงตัวอย่าง Real value mutation

ตัวอย่างในรูปที่ 2.17 เป็นการ Mutation โดยวิธี Real value mutation โดยที่ ตำแหน่งในการ Mutation เท่ากับ 6 และค่า Range มีค่าเท่ากับ -0.2 เมื่อแทนค่าลงในสมการที่ 2.38 จะได้ผลลัพธ์ดังต่อไปนี้ 2 + (-0.2 * 2) = 2 – 0.4 =1.6

2.4.6 การหยุดการเรียนรู้

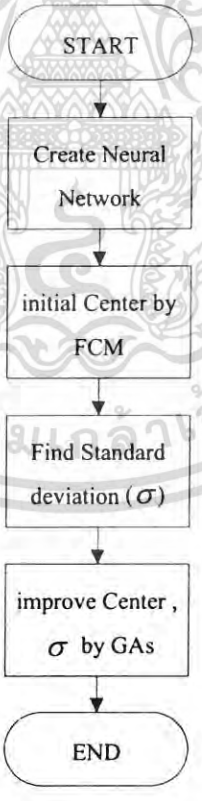
หลังจากระบวนการ Mutation โครโมโซมลูก (Offspring) จะถูกนำไปหาค่า Fitness value หากค่า Fitness value ที่ได้มีความเหมาะสมใกล้เคียงกับคำตอบหรือสิ้นสุดจำนวนรอบในการเรียนรู้ Genetic Algorithms จะจบการทำงานและโครโมโซมที่ให้ค่า Fitness value ที่ดีที่สุดในประชากรรุ่นสุดท้าย (Last generation) จะเป็นผลลัพธ์ของปัญหาที่เหมาะสมที่สุดในการรู้ครั้งนี้

บทที่ 3

Fuzzy and Evolutionary Neural Network Classifier

3.1 ขั้นตอนการดำเนินงาน

ในงานวิจัยนี้เป็นการสร้างอัลกอริทึมการเรียนรู้แบบใหม่สำหรับ Neural Network Classifier โดยใช้ Fuzzy และ Evolutionary Algorithms โดยนำโครงข่ายประสาทเทียม (Neural networks), Fuzzy C-mean clustering (FCM) และ Genetic algorithms (GAs) มาประยุกต์ใช้งานร่วมกัน ซึ่งผู้วิจัยขอเรียกว่า Fuzzy and Evolutionary Neural Network (FENN) โดยงานวิจัยนี้จะนำ Fuzzy C-mean clustering และ Genetic algorithms มาใช้หาจุด Center และค่าเบี่ยงเบนมาตรฐาน (Standard derivation: σ) ที่ดีที่สุดของ Gaussian function ซึ่งเป็นฟังก์ชั่นที่อยู่ในชั้น Hidden layer ของ Fuzzy and Evolutionary Neural Network โดยสามารถกำหนดเป็นขั้นตอนการดำเนินงานได้ดังต่อไปนี้



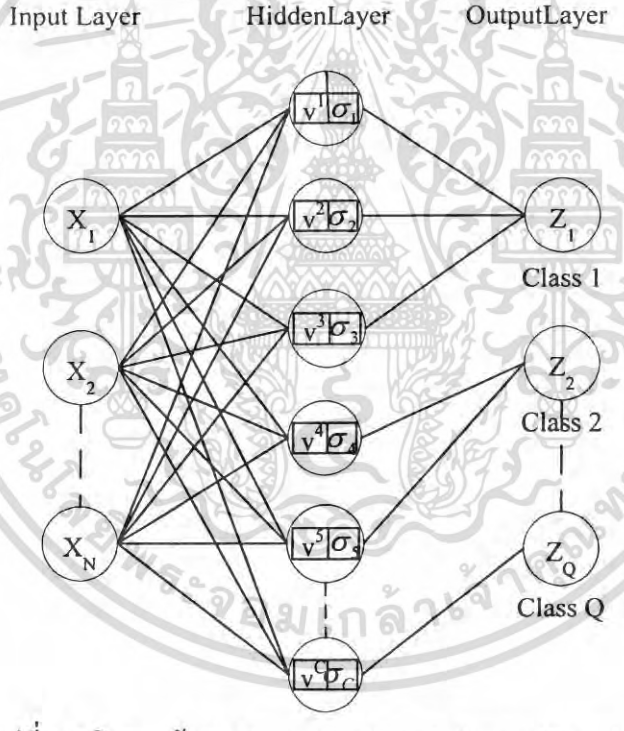
รูปที่ 3.1 ขั้นตอนการดำเนินงาน

1. สร้างโครงสร้าง Fuzzy and Evolutionary Neural Network ที่ใช้ในงานวิจัยนี้
2. กำหนดจุด Center เริ่มต้นโดยใช้วิธีการของ Fuzzy C-mean clustering
3. หาค่าเบี่ยงเบนมาตรฐาน (σ) จากแต่ละจุด Center ที่หาได้จากวิธีการของ Fuzzy C-mean clustering
4. ปรับปรุงจุด Center และ σ โดยใช้ Genetic Algorithms เพื่อหาจุด Center และ σ ที่เหมาะสมกับ Data ที่นำมาเรียนรู้ (Training data)

หมายเหตุ ขั้นตอนที่ 2 ถึง 4 เป็นส่วนของ Learning Algorithm

3.2 โครงสร้างของ Fuzzy and Evolutionary Neural Network

ในรูปที่ 3.2 แสดงโครงสร้างของ Fuzzy and Evolutionary Neural Network (FENN) ซึ่งจะประกอบไปด้วยจำนวน Layer 3 Layer คือ Input layer, Hidden layer และ Output layer



รูปที่ 3.2 โครงสร้างของ Fuzzy and Evolutionary Neural Network

3.2.1 Input Layer

ในชั้น Input layer นี้จะประกอบไปด้วย Input Node ที่มีขนาดเท่ากับจำนวน Attribute ของ Input Data ตัวอย่างเช่น X=[1.02, 0.54, 2] แสดงว่า X เป็นข้อมูลที่มีขนาด 3 Attributes ดังนั้น Input node จะมีจำนวนเท่ากับ 3 เป็นต้น โดยแต่ละ Input node จะเชื่อมต่อกับทุก Node ในชั้น Hidden Layer

ส่วนชุดข้อมูลที่สามารถใช้กับ Fuzzy and Evolutionary Neural Network จะต้องเป็นข้อมูลแบบจำนวนจริง (Real values) โดยที่ $X^P \in R$ เท่านั้น

3.2.2 Hidden Layer

ในชั้น Hidden layer นี้จะประกอบไปด้วย Node ที่บรรจุฟังก์ชั่นเก๊าส์เซียล (Gaussian function) เอาไว้โดยแต่ละ Node ในชั้นนี้จะเชื่อมต่อเฉพาะกับ Output node ที่ Hidden node นั้นเป็นสมาชิกอยู่ จากสมการที่ 3.1 จะพบว่า Gaussian function จะประกอบไปด้วยตัวแปรที่เป็นจุด Center (v) และค่าเบี่ยงเบนมาตรฐาน (σ) ซึ่งในส่วนนี้จะใช้ Fuzzy C-Mean Clustering และ Genetic Algorithms เพื่อหาค่าจุด Center และค่าเบี่ยงเบนมาตรฐานเหล่านี้

$$Y_j = e^{\left(-\left\|X - v^j\right\|^2 \Big/ \sigma_j^2\right)} \qquad : j = \{1, 2, ..., C\} \tag{3.1}$$

โดยที่ Y คือ ผลลัพธ์ที่ได้จากชั้น Hidden Layer

X คือ Input Data

v คือ จุด Center ของแต่ละ Node

σ คือ ค่าเบี่ยงเบนมาตรฐาน (Standard derivation)

C คือ จำนวนของ Hidden Node

3.2.3 Output Layer

ในชั้น Output layer นี้จำนวนของ Output Node เท่ากับจำนวนของกลุ่ม (Class) แทนด้วยค่า $k = \{1, 2, ..., Q\}$ จากสมการที่ 3.2 แสดงการหาค่าผลลัพธ์ของแต่ละ Output Node โดยเลือกผลลัพธ์ที่มากที่สุดของ Hidden Node (Y_j) ที่เป็นสมาชิกของ Class ใน Output Node (Z_k) นั้นมาเป็นคำตอบ (เลือกผลลัพธ์ของ Node ที่ชนะจาก Hidden node ที่อยู่ใน Class เดียวกันมาเป็นคำตอบ)

$$Z_k = \mathrm{Max}(Y_j) \qquad ; j = \{1, 2, ..., NK\}, k = \{1, 2, ..., Q\} \tag{3.2}$$

โดยที่ Z คือ ผลลัพธ์ที่ได้จากชั้น Output Layer

Y คือ ผลลัพธ์ที่ได้จากชั้น Hidden Layer เมื่อ $Y_j \in Z_k$

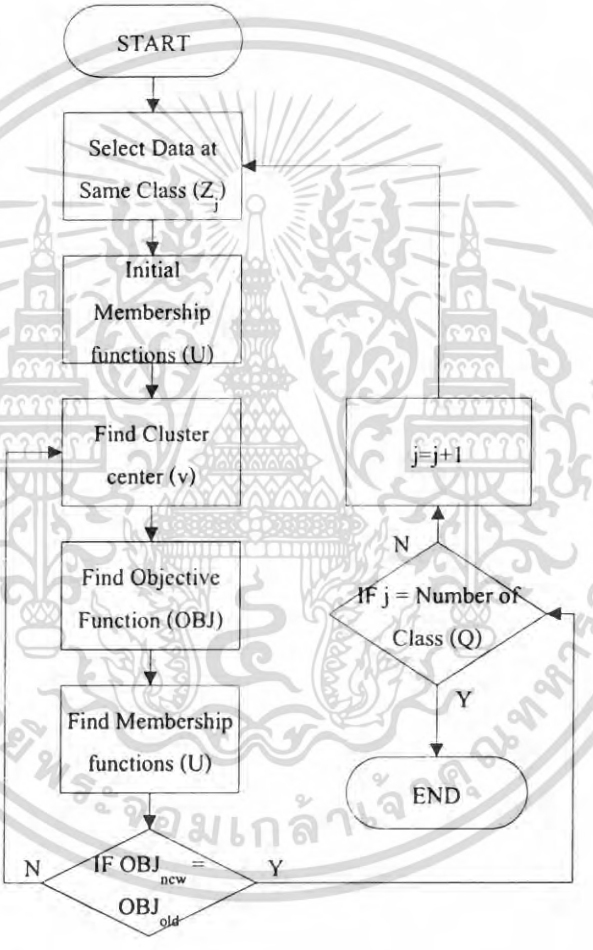
NK คือ จำนวนสมาชิกของ Z_k

Q คือ จำนวน Output node หรือ จำนวน Class

3.3 การหาจุด Center เริ่มต้นโดยวิธี Fuzzy C-Mean Clustering

ในงานวิจัยนี้ Genetic Algorithms จะถูกนำมาใช้ในการกำหนดจุด Center และเนื่องจาก Genetic Algorithms จะต้องมีการกำหนดจำนวนประชากรเริ่มต้น (Initial Population) ก่อน จึงใช้ FCM ในการกำหนดประชากรเริ่มต้น เพื่อให้ Genetic Algorithms สามารถทำงานได้อย่างมีประสิทธิภาพที่สุดในการหาจุด Center และค่าเบี่ยงเบนมาตรฐานที่เหมาะสมกับ Training Data

วิธีการหาจุด Center โดยวิธี Fuzzy C-Mean Clustering สามารถเขียนเป็น Flowchart ได้ดังรูปที่ 3.3 ซึ่งสามารถอธิบายเป็นขั้นตอนได้ดังนี้



รูปที่ 3.3 Flowchart วิธีการหาจุด Center โดยวิธี FCM

1. โดยทั่วไปแล้ว FCM นั้นเป็นกระบวนการ Clustering ซึ่งเป็นการเรียนรู้แบบไม่มีผู้ฝึกสอน ดังนั้นแต่ละจุด Center จะไม่มีกลุ่ม (Class) ที่แน่นอนเนื่องจาก FCM จะหาจุด Center โดยพิจารณาจากข้อมูลทั้งหมด เพื่อให้เหมาะสมกับโครงข่ายแบบ Fuzzy and Evolutionary Neural Network แต่ละจุด Center จำเป็นต้องทราบ Class ที่แน่นอน จึงต้องนำข้อมูลที่ใช้ในการฝึกฝน (Training Data)

มาจัดแบ่ง Class ให้ข้อมูลที่อยู่ใน Class เดียวกันอยู่ด้วยกันด้วยสมการที่ 3.3 ก่อนนำไปหาจุด Center โดยวิธี FCM

$$X = \{X_k^1, X_k^2, \ldots\ldots, X_k^P\} \qquad ; k=\{1, 2, \ldots, Q\} \tag{3.3}$$

เมื่อ Q คือจำนวน Class

P คือจำนวน Input Pattern ที่อยู่ใน Class เดียวกัน

X_k คือ Input Pattern ของ Class ที่ k

2. กำหนดค่าเริ่มต้น (Initial) ให้กับ Membership functions (U) โดยวิธีการสุ่ม

3. หาค่า Cluster center จากสมการที่ 3.4

$$v^j = \sum_{i=1}^{P} u_{ij}^m X^i \Big/ \sum_{i=1}^{P} u_{ij}^m \qquad ; j=\{1, 2, \ldots, C\} \tag{3.4}$$

โดยที่ C คือจำนวนของ Cluster center ที่ต้องการ

P คือจำนวน Input Pattern

m คือ Degree of Fuzzification $m \geq 1$

u คือ Membership Function

v คือ Cluster Center

4. หาค่า Objective function จากสูตร

$$OBJ(U,V) = \sum_{j=1}^{C} \sum_{i=1}^{P} u_{ij}^m D_{ij}^2 \tag{3.5}$$

โดยที่ C คือ จำนวนของ Cluster Center ที่ต้องการ

P คือจำนวน Input Pattern

OBJ คือ Objective function

D คือ ระยะทาง (Distance) ระหว่าง Input Pattern กับ Cluster Center

ซึ่งหาค่าได้จากสูตร $D_{ij}^2 = \left\| X^i - v^j \right\|^2$ (3.6)

โดยที่ X คือ Input Pattern

v คือ Cluster Center

5. หาค่า Membership function จากสูตร

$$u_{ij}=\left[\sum_{r=1}^{C}\left[\frac{D_{ij}}{D_{ir}}\right]^{\frac{2}{m-1}}\right]^{-1} \quad ; j=\{1,2,\ldots,C\} \ , i=\{1,2,\ldots,P\} \tag{3.7}$$

โดยที่ C คือ จำนวนของ Cluster Center

P คือจำนวน Input Pattern

m คือ Degree of Fuzzification $m \geq 1$

u คือ Membership Function

D คือ ระยะทาง (Distance) ระหว่าง Input Pattern กับ Cluster Center

6. หากค่า Objective function (OBJ_{new}) ที่คำนวณได้จากสมการที่ 3.5 มีค่าไม่เท่ากับ Objective function เดิม (OBJ_{old}) ให้กลับไปทำยังขั้นตอนที่ 3

7. หากยังกระทำไม่ครบทุก Class ให้กลับไปทำยังขั้นตอนที่ 1

3.4 การหาค่าเบี่ยงเบนมาตรฐานจากจุด Center

หลังจากกำหนดจุด Center เริ่มต้นด้วยวิธี Fuzzy C-Mean Clustering ได้แล้วขั้นตอนถัดมาคือการหาค่าเบี่ยงเบนมาตรฐาน (Standard deviation : σ) ที่เหมาะสมให้แก่ทุกจุด Center โดยคำนวณจากการหาระยะทางแบบ Euclidian distance ซึ่งในวิทยานิพนธ์ฉบับนี้จะใช้ระยะทางที่สั้นที่สุดระหว่างจุด Center ที่อยู่ต่างกลุ่ม (Class) มากำหนดเป็นค่า σ

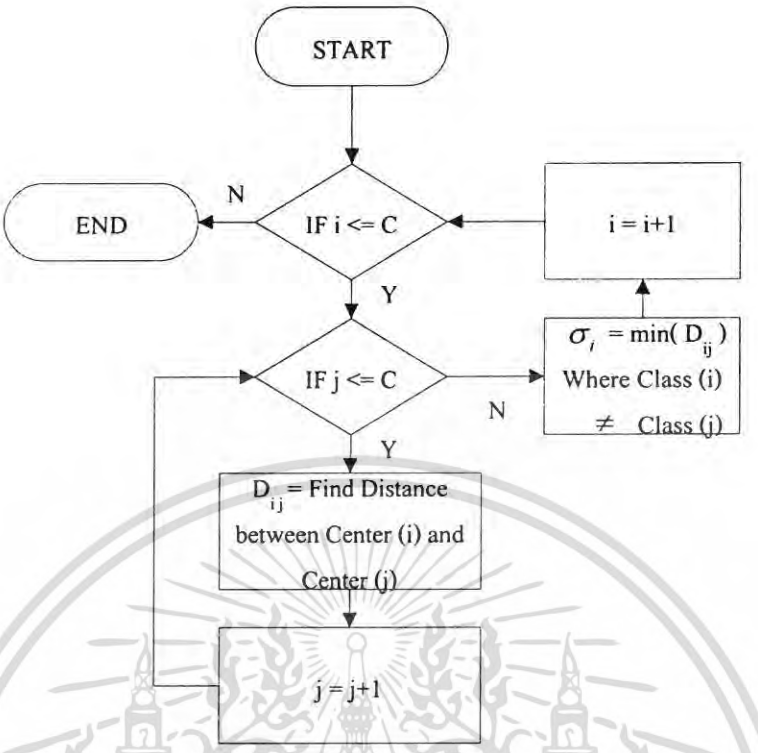
รูปที่ 3.4 เป็น Flowchart แสดงวิธีการหาค่าเบี่ยงเบนมาตรฐาน (σ) โดยเริ่มจากหาระยะทางแบบ Euclidian จากสูตร

$$D_{ij}=\left\|v^{i}-v^{j}\right\| \quad ; i=\{1,2,\ldots,C\} \ , \ j=\{1,2,\ldots,C\} \ , \ i\neq j \tag{3.8}$$

โดยที่ C คือ จำนวนจุด Center ทั้งหมด

v คือ จุด Center

D คือ ระยะทางที่ได้จาก Euclidian distance



รูปที่ 3.4 Flowchart วิธีการหาค่าเบี่ยงเบนมาตรฐาน

โดยค่า Distance (D_{ij}) ที่ได้ของแต่ละจุด Center จะถูกเรียงลำดับเพื่อหาค่าที่น้อยที่สุดโดยคิดเฉพาะค่า Distance (D_{ij}) ระหว่างจุด Center ที่กำลังพิจารณากับจุด Center ที่อยู่ต่าง Class กันเท่านั้นมาเป็นค่า σ ให้กับจุด Center นั้น

$$\sigma_i = \text{Min}(D_{ij}) \qquad \text{ที่ Class}(v^i) \neq \text{Class}(v^j) \tag{3.9}$$

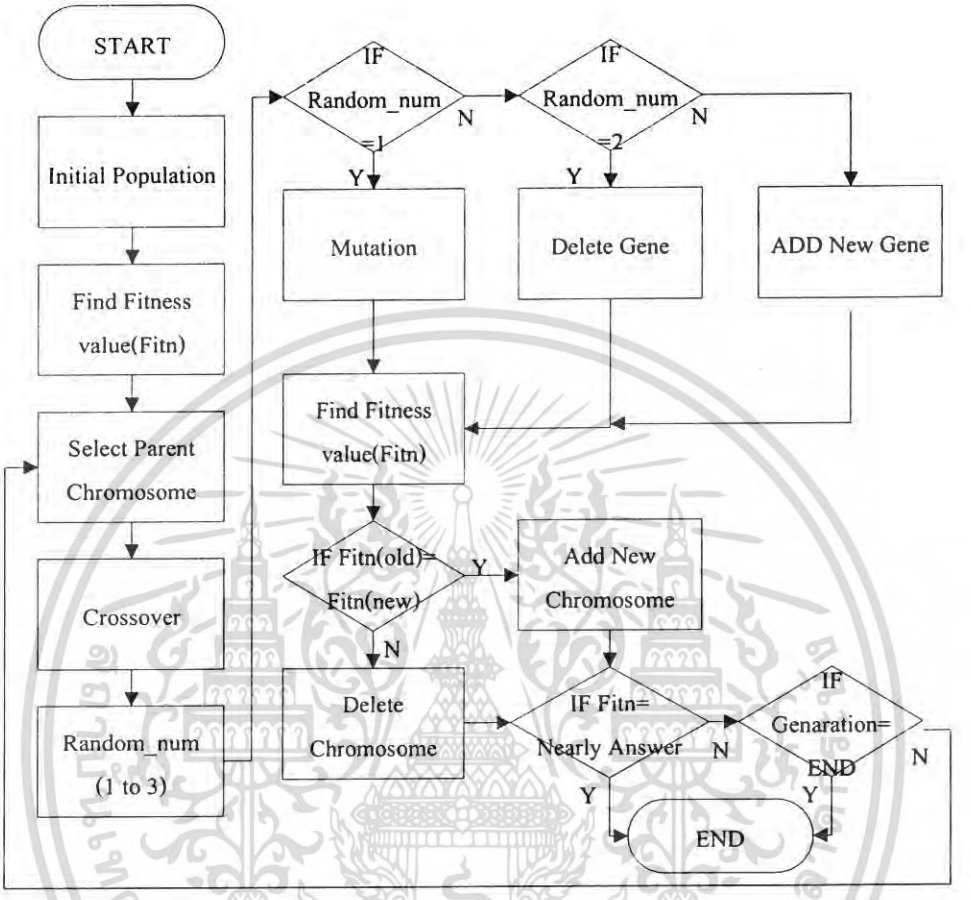
เมื่อ $i=\{1, 2, \ldots, C\}$, $j=\{1, 2, \ldots, C\}$

v คือ จุด Center

D_{ij} คือ Distance ระหว่างจุด Center

σ คือค่าเบี่ยงเบนมาตรฐานของจุด Center

3.5 ปรับปรุงจุด Center และค่าเบี่ยงเบนมาตรฐานโดยใช้ GAs



รูปที่ 3.5 Flowchart แสดงการใช้ GAs ในการหาจุด Center และค่าเบี่ยงเบนมาตรฐาน

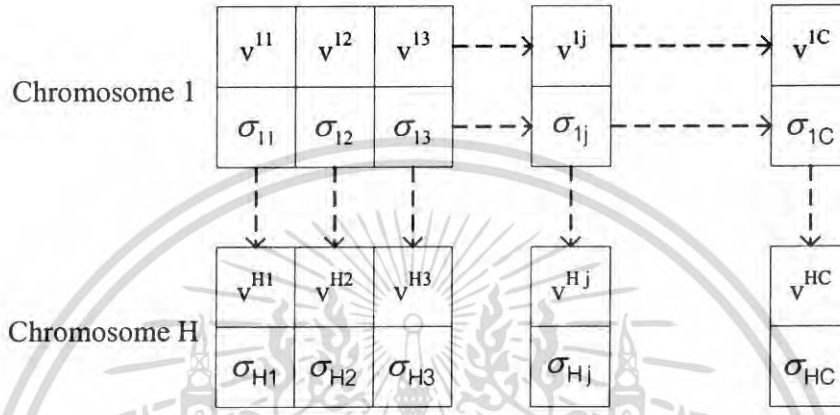
หลังจากที่หาจุด Center และค่าเบี่ยงเบนมาตรฐาน (σ) ของแต่ละ Class ได้แล้ว จะแน่ใจได้อย่างไรว่า จุด Center และค่า σ ที่หามาได้นั้นเป็นค่าที่เหมาะสมที่สุดสำหรับข้อมูลที่นำมาเรียนรู้

ดังนั้นจึงได้นำ Genetic Algorithms (GAs) เข้ามาเพื่อช่วยในการหาจุด Center และค่า σ ที่เหมาะสม เนื่องจากหลักการของ Genetic Algorithms จะสร้างประชากรรุ่นใหม่ที่ดีกว่าประชากรรุ่นเดิม จึงสามารถคาดการณ์ได้ว่าหากกำหนดกลุ่มประชากรเป็นจุด Center และค่า σ ให้กับ GAs ประชากรที่จะสืบทอดในรุ่นถัดไป (Next Generation) จะให้ค่าจุด Center และค่า σ ที่เหมาะสมแก่ Fuzzy and Evolutionary Neural Network ได้

จากรูปที่ 3.5 เป็น Flowchart แสดงขั้นตอนการใช้ Genetic Algorithms ในการหาจุด Center และค่า σ ที่เหมาะสม ซึ่งสามารถอธิบายเป็นขั้นตอนได้ดังนี้

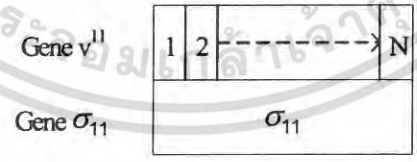
3.5.1 การกำหนดประชากรเริ่มต้น

เป็นการกำหนดประชากรเริ่มต้น (Initial Population) ให้กับ Genetic Algorithms ซึ่งโครโมโซมแต่ละเส้นจะประกอบไปด้วยยีนส์ (Genes) อยู่ 2 ชนิด คือยีนส์ที่เป็นค่าจุด Center และยีนส์ที่เป็นค่าเบี่ยงเบนมาตรฐาน (σ) ของ Fuzzy and Evolutionary Neural Network ดังที่ได้แสดงในรูปที่ 3.6



รูปที่ 3.6 แสดงองค์ประกอบของยีนส์ในแต่ละโครโมโซม

จากรูปที่ 3.6 จะพบว่าจำนวนของโครโมโซม (Chromosome) แทนด้วยตัว H ในแต่ละโครโมโซมจะประกอบไปด้วยยีนส์ซึ่งแทนจำนวนด้วยตัว C ซึ่งเท่ากับจำนวนของจุด Center หรือจำนวน Hidden Node ใน Hidden Layer โดยแต่ละโครโมโซมจะประกอบไปด้วยยีนต์จุด Center และยีนต์ค่า σ



รูปที่ 3.7 แสดงโครงสร้างภายในยีนส์

จากรูปที่ 3.7 แสดงโครงสร้างภายในยีนส์ ซึ่งจะประกอบไปด้วยจุด Center (v) และค่าเบี่ยงเบนมาตรฐาน (σ) โดยที่ขนาดของจุด Center ในแต่ละยีนส์จะมีขนาดเท่ากับ N ซึ่ง N จะเท่ากับจำนวน Attribute ของ Input Pattern และขนาดของค่า σ จะมีค่าเท่ากับ 1

ซึ่งประชากรในส่วนของ Initial Population นี้ได้มาจาก การหาจุด Center โดยวิธี Fuzzy C-Mean Clustering โดยจะกำหนดเป็น Chromosome เส้นแรกและทำการเลือก Input Pattern ที่ใกล้เคียงกับจุด Center ของแต่ละ Class มากำหนดเป็น Chromosome เส้นที่เหลือ โดยการคัดเลือกจะใช้การหาระยะทางแบบ Euclidian เลือกตัวที่ใกล้จุด Center ที่สุด ส่วนค่าเบี่ยงเบนมาตรฐานของแต่ละจุด Center หาได้จากวิธีการเดียวกันในหัวข้อที่ 3.4

3.5.2 การหาค่า Fitness value

การหาค่า Fitness value เป็นการประเมินค่าของประชากรในแต่ละรุ่นว่ามีความเหมาะสมกับงานหรือสิ่งที่ต้องการจะทราบมากน้อยเพียงไร โดยค่า Fitness value จะแปรผันตรงกับคุณภาพของประชากรและช่วยในการตัดสินใจในการเลือก Parent chromosome ของประชากรในรุ่นถัดไป โดยค่า Fitness value นั้นหาได้จากการนำ Chromosome แต่ละเส้นผ่าน Fitness function ซึ่งใน Fuzzy and Evolutionary Neural Network นี้จะใช้ Fitness function ดังสมการที่ 3.10 โดยใช้ผลรวมของความผิดพลาดมากำหนดเป็นค่า Fitness value ให้กับแต่ละ Chromosome โดยหากผลรวมของความผิดพลาดมีค่าสูง Fitness value จะมีค่าต่ำและหากผลรวมของความผิดพลาดมีค่าน้อย Fitness value จะมีค่าสูง

$$Fit\_Val^{f} = \left[\sum_{r=1}^{H} (Wrong^{r} + NewErr^{r}) \right] - (Wrong^{f} + NewErr^{f}) \quad ;f=\{1, 2, \ldots, H\} \qquad (3.10)$$

โดยที่ H คือ จำนวน Chromosome ในแต่ละรุ่น

Fit_Val คือ ค่า Fitness value

Wrong คือ ผลรวมของคำตอบที่ระบุผิด Class

NewErr คือ ค่า Error rate

จากสมการที่ 3.10 จะพบว่ามีตัวแปรที่ไม่ทราบค่าอยู่ด้วยกัน 2 ตัวแปรคือ Wrong และ NewErr โดยค่า Wrong คือผลรวมของจำนวนคำตอบที่ระบุผิด Class ตามสมการที่ 3.11

$$Wrong^{f} = \sum_{i=1}^{P} A^{i} \qquad ; f = \{1, 2, \ldots, H\} \qquad (3.11)$$

เมื่อ $$A^{i} = \begin{cases} 0 & ,\text{if } O^{i} = T^{i} \\ 1 & ,\text{Otherwise} \end{cases} \qquad (3.12)$$

โดยที่ P คือ จำนวน Input Pattern

H คือ จำนวน Chromosome

T คือ Target ของ Input pattern

O คือ ผลลัพธ์ของ FENN แสดง Class ที่ชนะ

Wrong คือ ผลรวมของจำนวนคำตอบที่ระบุผิด Class

โดยค่า O นั้นหาได้จากสมการที่ 3.13 ซึ่งเป็นคำตอบที่ระบุ Class ของ Hidden node ที่ชนะ (Node ที่ให้ค่าสูงสุด) ของแต่ละ Input pattern

$$O^i = Class(Max(Y_j)) \quad ; j = \{1, 2, \ldots, C\}, i = \{1, 2, \ldots, P\} \tag{3.13}$$

โดยที่ C คือ จำนวนของ Hidden node

P คือ จำนวนของ Input pattern

O คือ ผลลัพธ์ที่ระบุ Class ของ Hidden node ที่ชนะ

Y คือ ผลลัพธ์ที่ได้จากชั้น Hidden layer

ซึ่งค่า Y เป็นผลลัพธ์ที่ได้จากสมการ Gaussian function ที่อยู่ภายใน Hidden node ดังสมการที่ 3.14

$$Y_j = e^{\left(-\left\|X - v^j\right\|^2 / \sigma_j^2\right)} \quad : j = \{1, 2, \ldots, C\} \tag{3.14}$$

โดยที่ Y คือ ผลลัพธ์ที่ได้จากชั้น Hidden layer

X คือ Input pattern

v คือ จุด Center

σ คือ ค่าเบี่ยงเบนมาตรฐาน (Standard derivation)

C คือ จำนวนของ Hidden Node

จากสมการที่ 3.10 จะพบว่านอกจากตัวแปร Wrong แล้วยังมีตัวแปร NewEff อีก 1 ตัวแปร ที่ยังไม่ทราบค่าซึ่งค่า NewEff นั้นหาได้จากค่า Err ซึ่งเป็นผลรวมของอัตราส่วนระหว่าง Node ที่ชนะในกลุ่มที่มี Class ไม่ตรงกับ Target กับ Node ที่ชนะในกลุ่มที่มี Class ตรงกับ Target ในแต่ละ Input pattern ดังสมการที่ 3.15

$$Err^f = \sum_{i=1}^{P} (False^i / True^i) \quad ; f = \{1, 2, \ldots, H\} \tag{3.15}$$

เมื่อ $False^i$ คือ ค่า $Max(Y_j)$ ที่ $Class(Max(Y_j)) \notin T^i$; $j = \{1, 2, \ldots\ldots, C\}$

$True^i$ คือ ค่า $Max(Y_j)$ ที่ $Class(Max(Y_j)) \in T^i$; $j = \{1, 2, \ldots\ldots, C\}$

โดยที่ H คือ จำนวน Chromosome ในแต่ละรุ่น

P คือ จำนวน Input pattern

C คือ จำนวนของ Hidden Node

Y คือ ผลลัพธ์ที่ได้จากชั้น Hidden layer ในสมการที่ 3.14

T คือ Target ของ Input pattern

จากสมการที่ 3.15 จะพบว่าค่าความผิดพลาด (Err) จะมีค่าน้อยก็ต่อเมื่อผลลัพธ์จาก Hidden node ของ Class ที่ตอบถูก (True) จะต้องมีค่ามากกว่าผลลัพธ์จาก Hidden node ของ Class ที่ตอบผิด (False) เพื่อให้เกิดผลต่างระหว่างผลลัพธ์ที่ตอบถูกต้องกับผลลัพธ์ที่ตอบผิดมากยิ่งขึ้น

ในวิทยานิพนธ์ฉบับนี้จะให้ความสำคัญกับค่า Wrong มากกว่าค่า Err เนื่องจากค่า Wrong จะเป็นตัวระบุว่า Chromosome เส้นนี้มีความผิดพลาดมากน้อยเพียงใด หากค่า Wrong มีค่าเท่ากันจึงจะพิจารณาค่า Err ดังนั้นเพื่อให้การเปรียบเทียบค่า Err ง่ายขึ้นจึงแปลงค่า Err ให้อยู่ในรูปทศนิยมมีค่าตั้งแต่ 0 ถึง 1 ตามสมการที่ 3.16 โดยค่า Wrong จะคงไว้ในรูปเลขจำนวนเต็มบวก

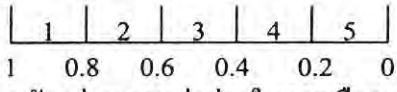
$$NewErr^f = Err^f \Big/ \sum_{r=1}^{H} Err^r \qquad ; f = \{1, 2, \ldots\ldots, H\} \tag{3.16}$$

โดยที่ H คือ จำนวน Chromosome

3.5.3 Selecting Parent Chromosomes

ในขั้นตอนนี้จะใช้วิธี Roulette wheel selection ในการเลือก Parent Chromosome เพื่อกระทำขั้นตอน Crossover ในการสร้างโครโมโซมลูก (Offspring) ต่อไป

วิธีการเลือกแบบ Roulette wheel selection นี้จะสุ่มตัวเลขในช่วง 0 ถึง 1 ขึ้นมาดังนั้นจึงต้องจัดช่วงของโครโมโซม ให้อยู่ในระหว่าง 0 ถึง 1 ด้วยวิธีการที่ง่ายที่สุดคือกำหนดช่วงในการเลือกทุกช่วงให้เท่ากันหมด เช่น หากมีโครโมโซมทั้งหมด 5 เส้น ก็ทำการแบ่งช่วงได้เป็น 5 ช่วงเท่าๆกันดังแสดงในรูปที่ 3.8



รูปที่ 3.8 ตัวอย่างการแบ่งช่วงในการเลือกเท่าๆกัน

จากรูปที่ 3.8 หากสุ่มตัวเลขในช่วง 0 ถึง 1 ขึ้นมาได้เท่ากับ 0.7 ซึ่งตกอยู่ในช่วงที่ 2 แสดงว่าจะต้องเลือกโครโมโซม เส้นที่ 2 ขึ้นมาเป็น Parent เพื่อใช้ในการ Crossover การแบ่งช่วงเท่าๆ กันแบบนี้ทำให้ทุกโครโมโซมจะมีความสำคัญเท่ากันหมด แต่เนื่องจากโครโมโซมแต่ละเส้นมีค่า Fitness value ที่ไม่เท่ากัน ซึ่งหมายความว่าเส้นที่มีค่า Fitness value สูง น่าจะถูกเลือกโดยการสุ่มมากกว่าโครโมโซม ทีมีค่า Fitness value ต่ำ

ดังนั้นช่วงในการเลือกของแต่ละโครโมโซม ควรขึ้นอยู่กับค่า Fitness value ด้วยดังสมการที่ 3.17

$$Fit\_len^f = Fit\_Val^f \Big/ \sum_{r=1}^{H} Fit\_Val^r \; ; f = \{1, 2, \ldots, H\} \tag{3.17}$$

โดยที่ H คือ จำนวนของโครโมโซมทั้งหมดในประชากร (Population) แต่ละรุ่น

Fit_Val คือ ค่า Fitness value

Fit_len คือ ช่วงในการถูกเลือกของแต่ละ Chromosome

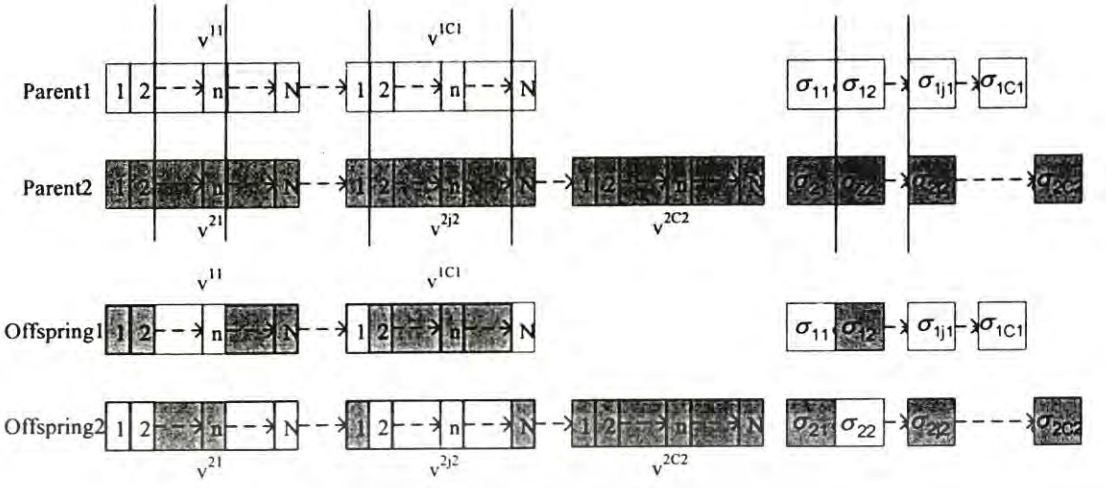
เนื่องจากในวิทยานิพนธ์นี้ จำนวนโครโมโซม จะไม่เท่ากันในทุกรุ่น (Generation) ดังนั้นค่า H จึงสามารถเปลี่ยนแปลงได้ตลอดเวลา

3.5.4 Crossover

การ Crossover เป็นกระบวนการที่เกิดขึ้นหลังจากกระทำการ Selection Parent Chromosome เรียบร้อยแล้ว เพื่อทำการสร้างโครโมโซมลูก (Offspring)

เนื่องจากในวิทยานิพนธ์ฉบับนี้แต่ละโครโมโซม อาจมีจำนวนยีนส์ (Gene) ไม่เท่ากันได้ ($C_1 \neq C_2$) ดังแสดงในรูปที่ 3.9 ดังนั้นยีนส์ ที่เกินออกมาจะไม่ถูก Crossover แต่อย่างใด แต่จะผูกติดไปให้กับ Offspring ตัวใดก็ได้ขึ้นอยู่กับการสุ่ม โดยจะผูกไปทั้งยีนส์ Center และยีนต์ค่าเบี่ยงเบนมาตรฐาน (σ) ดังที่แสดงในรูปที่ 3.9

ในส่วนของยีนส์ที่สามารถ Crossover กันได้ จะทำการ Crossover กันเฉพาะภายในคู่ของยีนส์นั้นๆไม่ข้ามคู่กันแต่อย่างใดดังแสดงในรูปที่ 3.9 ซึ่งยีนส์ของจุด Center กับยีนส์ของค่าเบี่ยงเบนมาตรฐานจะแยกกัน Crossover โดยไม่ยุ่งเกี่ยวกัน ส่วนตำแหน่งในการ Crossover นั้นจะขึ้นอยู่กับการสุ่ม



รูปที่ 3.9 กระบวนการ Crossover

3.5.5 Mutation, Add Gene, Delete Gene

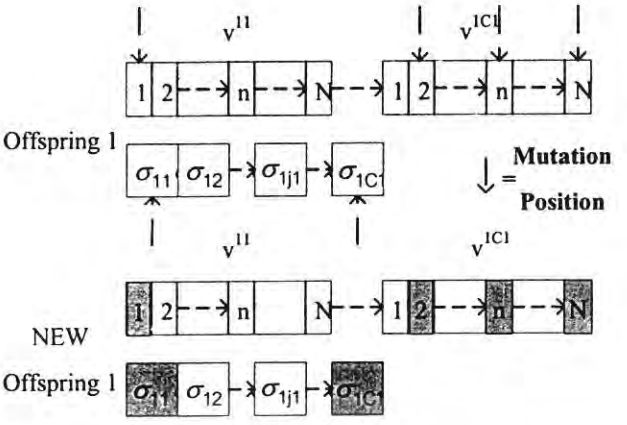
จะเกิดขึ้นหลังจากกระบวนการ Crossover แล้ว ซึ่งจะกระทำกับ Offspring เพื่อให้โครงข่ายประสาทเทียมที่ได้ไม่ Over fit จนเกิดไป จาก Flowchart ในรูปที่ 3.5 จะพบว่า การ Mutation, Add Gene และ Delete Gene จะไม่เกิดขึ้นพร้อมๆกัน แต่จะเกิดเพียงอย่างใดอย่างหนึ่งเท่านั้น ซึ่งจะกำหนดโดยวิธีการสุ่มตัวเลขจำนวนเต็มระหว่าง 1 ถึง 3

1. Mutation

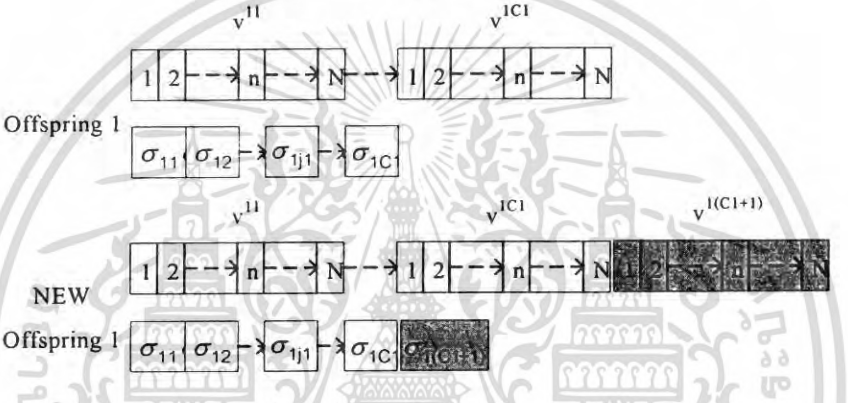
วิธีการ Mutation จะใช้วิธีเลือกสุ่มพารามิเตอร์ ที่จะถูก Mutation ของทุกยีนส์ในแต่ละโครโมโซมของ Offspring โดยจะทำการ Mutation ทั้งส่วนยีนส์ที่เป็นค่าจุด Center และยีนส์ที่เป็นค่าเบี่ยงเบนมาตรฐาน โดยจำนวนของการ Mutation ก็จะขึ้นอยู่กับการสุ่มเช่นกันดังแสดงในรูปที่ 3.10

2. Add Gene

วิธีการ Add Gene นั้นจะเพิ่มยีนส์ต่อท้ายให้กับโครโมโซม Offspring โดยจะเพิ่มทั้ง ยีนส์ที่เก็บค่าจุด Center และยีนส์ที่เก็บค่าเบี่ยงเบนมาตรฐาน ส่วนค่าที่นำมาบรรจุให้กับยีนส์ที่เกิดใหม่นี้ได้มาจากการสุ่มเลือกพารามิเตอร์ จากยีนส์ในโครโมโซมของประชากรรุ่นนั้นๆ โดยยีนส์ที่เกิดขึ้นใหม่นี้จะต้องถูกสุ่มเพื่อหา Class อีกด้วย เนื่องจากยีนส์แต่ละตัวเป็นตัวแทนของ Hidden node ซึ่ง Node เหล่านี้จะต้องเชื่อมต่อกับ Output node ใด node หนึ่ง (Class ใด Class หนึ่ง) เท่านั้นโดยวิธีการ Add Gene แสดงได้ดังรูปที่ 3.11



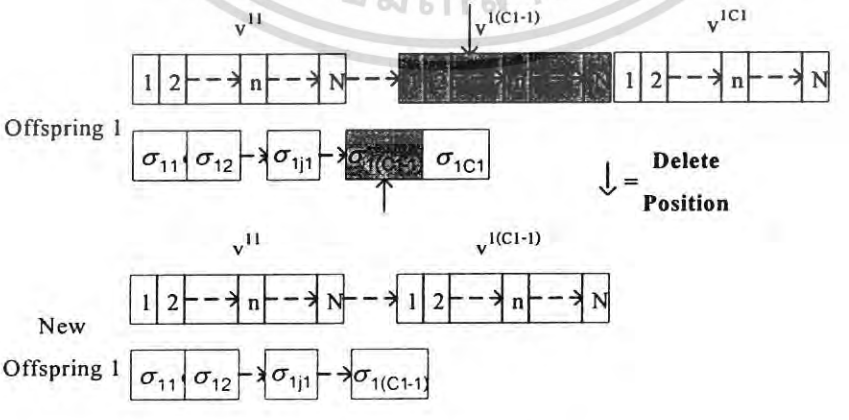
รูปที่ 3.10 แสดงวิธีการ Mutation



รูปที่ 3.11 แสดงวิธีการ Add Gene

3. Delete Gene

วิธีการ Delete Gene จะใช้วิธีการสุ่มเลือกยีนส์ที่อยู่ภายในโครโมโซม Offspring เมื่อได้ตำแหน่งจะทำการลบทั้งยีนส์ที่เก็บค่าจุด Center และยีนส์ที่เก็บค่าเบี่ยงเบนมาตรฐาน วิธีการลบแสดงดังรูปที่ 3.12



รูปที่ 3.12 แสดงวิธีการ Delete Gene

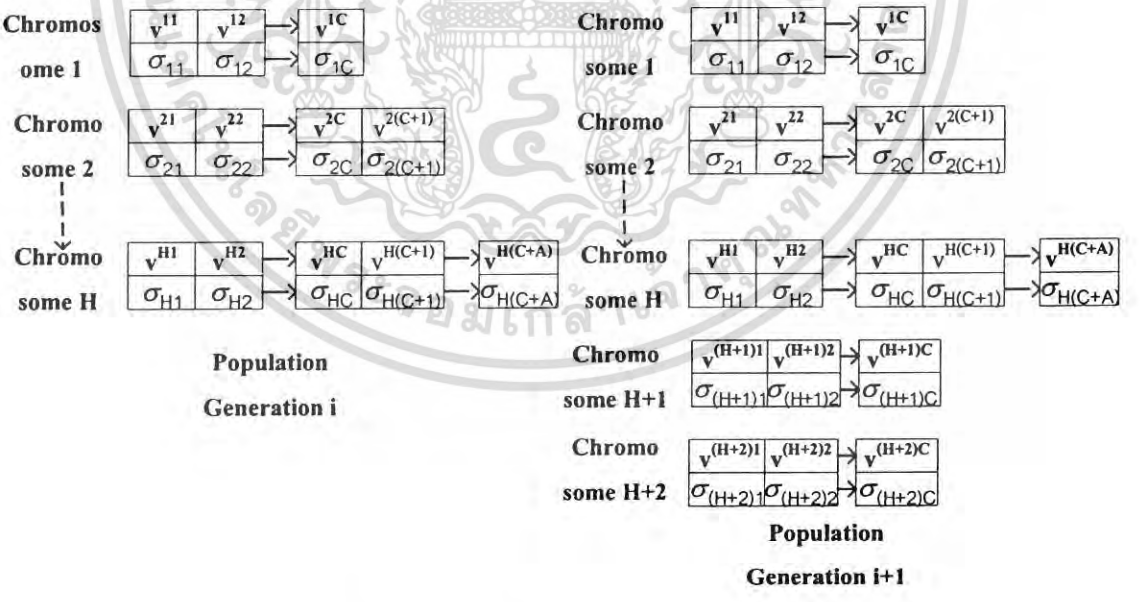
3.5.6 การ Add Chromosomes และการ Delete Chromosomes

จาก Flowchart ในรูปที่ 3.5 จะพบว่าในกรณีที่ค่า Fitness value ไม่มีการเปลี่ยนแปลง หรือเปลี่ยนแปลงน้อยมาก ทำให้ต้องเสียเวลาในการประมวลผลนานขึ้น เนื่องจากการเปลี่ยนแปลงในแต่ละรุ่น (Generation) จะแตกต่างกันน้อยมากเพราะข้อมูลในแต่ละโครโมโซมมีความใกล้เคียงกัน ทำให้ต้องมีการเพิ่มโครโมโซมขึ้นมาเพื่อ ให้จำนวนประชากร (Population) มากขึ้น ทำให้ข้อมูลมีความหลากหลายมากขึ้นช่วยให้ค่า Fitness value เปลี่ยนแปลงมากขึ้น หลังจากค่า Fitness value เปลี่ยนแปลงแล้วจำเป็นจะต้องลดขนาดของโครโมโซมให้กลับมาอยู่ในระดับปกติ เนื่องจากยิ่งจำนวนประชากรมากขึ้นก็จะยิ่งเสียเวลาในการประมวลผลมากขึ้นเช่นกัน

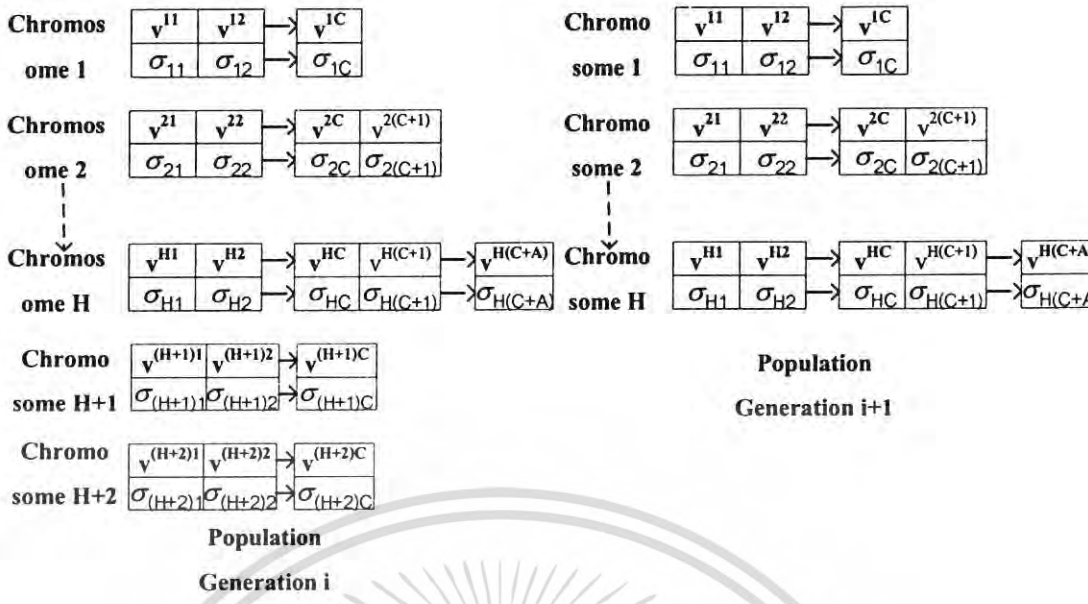
1. Add Chromosome

การเพิ่มโครโมโซม (Add Chromosomes) นั้นจะเพิ่มโครโมโซมครั้งละ 2 เส้นและจะเพิ่มในทุกรุ่นโดยโครโมโซมที่เพิ่มขึ้นมานี้ จะใช้วิธีการสุ่มเลือกยีนส์ภายในโครโมโซมที่เป็นประชากรในรุ่นเดียวกัน โดยองค์ประกอบของยีนส์ที่เลือกมานั้นจะประกอบไปด้วยยีนส์ที่เป็นจุด Center, ค่าเบี่ยงเบนมาตรฐาน และ Class (เพื่อระบุว่าเป็นสมาชิกของ Output Node ใด)

ในวิทยานิพนธ์ฉบับนี้จะเพิ่มโครโมโซมมากที่สุดไม่เกิน 10 เส้น เมื่อรวมกับประชากร (Population) ดั้งเดิมแล้วจะไม่เกิน 30 เส้นและการเพิ่มโครโมโซมนั้นจะทำการเพิ่มก็ต่อเมื่อค่า Fitness value ของโครโมโซมเส้นที่ดีที่สุด (ค่า Fitness value มากที่สุด) ไม่มีการเปลี่ยนแปลง



รูปที่ 3.13 แสดงวิธีการ Add Chromosomes



รูปที่ 3.14 แสดงวิธีการ Delete Chromosomes

2. Delete Chromosome

การลบโครโมโซม (Delete Chromosomes) นั้นจะเกิดขึ้นได้ก็ต่อเมื่อ จำนวนของโครโมโซมในกลุ่มของประชากรมีการเพิ่มขึ้นมาจากเดิม (เพิ่มจากการ Add Chromosomes) และค่าที่ได้จาก Fitness value ของโครโมโซมเส้นที่ดีที่สุดเริ่มมีการเปลี่ยนแปลง จึงต้องลดจำนวนของโครโมโซมลง เนื่องจากจำนวนของโครโมโซมจะมีผลต่อการประมวลผลดังนั้นจำเป็นจะต้องลดโครโมโซม ให้กลับไปยังสภาวะเดิม ซึ่งวิทยานิพนธ์ฉบับนี้กำหนดจำนวนประชากร (Population) เท่ากับ 20 ส่วนโครโมโซมเส้นที่จะถูกลบนั้นจะเลือกโครโมโซมเส้นที่ด้อยที่สุด (ค่า Fitness value น้อยที่สุด) และจะลบไปเรื่อยๆจนกระทั่งจำนวนประชากรลดลงจนเท่ากับ 20

3.5.7 การหยุดการเรียนรู้

การที่ Genetic Algorithms จะหยุดการเรียนรู้ได้มีอยู่ 2 เงื่อนไขคือ

1. ค่า Fitness value มีค่าเหมาะสมหรือเท่ากับที่ต้องการ
2. จำนวนรอบของรุ่น (Generation) ได้ดำเนินมาถึงรอบที่กำหนด

ซึ่งหลังจากหยุดการเรียนรู้แล้วโครโมโซมเส้นที่ให้ค่า Fitness value สูงสุดภายในประชากรรุ่นสุดท้าย จะถูกนำมากำหนดเป็นค่าจุด Center และค่าเบี่ยงเบนมาตรฐาน (Standard deviation : σ) ของ Fuzzy and Evolutionary Neural Network (FENN)

บทที่ 4

การทดลองและผลการทดลอง

การทดลองในส่วนนี้ เป็นการทดลองเปรียบเทียบประสิทธิภาพในเชิงความถูกต้อง ในการจัดกลุ่มของ Fuzzy and Evolutionary Neural Network (FENN) ซึ่งเป็น Model ที่นำเสนอ โดยแบ่งการทดลองออกเป็น 3 การทดลองได้แก่ (1) เปรียบเทียบผลการทดลองของ FENN กับ Back-propagation และ Radial Basis Function (2) เปรียบเทียบผลการทดลองของ FENN โดยใช้ชุดข้อมูลที่ไม่ผ่านการ Normalization กับข้อมูลที่ผ่านการ Normalization (3) เปรียบเทียบผลการทดลองของ FENN แบบกำหนดขอบเขต (Boundary) กับแบบไม่กำหนดขอบเขต ของจำนวน Hidden node ในชั้น Hidden layer โดยทั้ง 3 การทดลองจะเลือกโครงสร้างของทั้ง FENN, Back-propagation และ Radial basis function ที่ให้ผลการทดลองที่ดีที่สุดมานำเสนอ

การทดลองทั้งหมดใช้โปรแกรม MATLAB 6.5 ในการสร้าง Model ต่าง ๆ ที่ใช้ในการทดลอง โดยทั้ง 3 การทดลองใช้ข้อมูลมาตรฐานจาก UCI machine learning [10] จำนวน 8 ชุดข้อมูล ที่ใช้ทดสอบอัลกอริทึมแบบ Classification ซึ่งข้อมูลแต่ละชุดมีลักษณะดังนี้

1. Vowel Recognition

เป็นชุดข้อมูลของการรู้จำ (Recognition) เสียงสระในภาษาอังกฤษจำนวน 11 เสียง (hid, hId, hEd, hAd, hYd, had, hOd, hod, hUd, hud, และ hed) โดยไม่ขึ้นอยู่กับผู้พูด (Independent speaker) ในชุดข้อมูลนี้จะเก็บข้อมูลจากผู้พูด 15 คน จากผู้ชาย 8 คนและผู้หญิง 7 คน โดยแบ่งชุดข้อมูลที่ใช้ในการเรียนรู้ (Training data) จากผู้ชาย 4 คนผู้หญิง 4 คนเป็นจำนวน 528 เสียงและชุดข้อมูลที่ใช้ในการทดสอบ (Testing data) จากผู้ชาย 4 คนผู้หญิง 3 คนเป็นจำนวน 462 เสียง รวมทั้งสิ้นเป็นจำนวน 990 เสียงโดยเสียงเหล่านี้จะถูกดึงคุณลักษณะ (Feature) โดยผ่าน Low pass filter ที่ 4.7 kHz, อัตราการ sampling ที่ 10 kHz ขนาด 12 bits, Linear predictive อันดับที่ 12 และผ่าน Hamming windowed เพราะฉะนั้นจะได้จำนวนคุณสมบัติ (Attribute) เท่ากับ 10 และจำนวนกลุ่ม (Class) เท่ากับ 11

2. Sonar: Mines vs. Rocks

ข้อมูลชุดนี้เป็นการวิเคราะห์คลื่นโซน่าร์เพื่อจำแนกโลหะกับหินที่อยู่ใต้ดิน โดยจะมีข้อมูลของกลุ่มโลหะอยู่ 111 ตัวและข้อมูลของกลุ่มหินอยู่ 97 ตัวโดยที่จะแบ่งเป็น Training data เท่ากับ 105 ตัวและ Testing data เท่ากับ 103 ตัวซึ่งแต่ละข้อมูลจะมีอยู่ด้วยกัน 60 attribute แต่ละ attribute มีค่าอยู่ระหว่าง 0 ถึง 1

```
-3.639,0.418,-0.67,1.779,-0.168,1.627,-0.388,0.529,-0.874,-0.814  :hid
-3.327,0.496,-0.694,1.365,-0.265,1.933,-0.363,0.51,-0.621,-0.488  :hId
-2.12,0.894,-1.576,0.147,-0.707,1.559,-0.579,0.676,-0.809,-0.049  :hEd
-2.287,1.809,-1.498,1.012,-1.053,1.06,-0.567,0.235,-0.091,-0.795 :hAd
-2.598,1.938,-0.846,1.062,-1.633,0.764,0.394,-0.15,0.277,-0.396   :hYd
-2.852,1.914,-0.755,0.825,-1.588,0.855,0.217,-0.246,0.238,-0.365 :had
-3.482,2.524,-0.433,1.048,-1.995,0.902,0.322,0.45,0.377,-0.366   :hOd
-3.941,2.305,0.124,1.771,-1.815,0.593,-0.435,0.992,0.575,-0.301  :hod
-3.86,2.116,-0.939,0.688,-0.675,1.679,-0.512,0.928,-0.167,-0.434 :hUd
-3.648,1.812,-1.378,1.578,0.065,1.577,-0.466,0.702,0.06,-0.836    :hud
-3.032,1.739,-1.141,0.737,-0.834,1.386,-0.575,0.679,-0.018,-0.823:hed
```

รูปที่ 4.1 ตัวอย่างชุดข้อมูล Vowel Recognition

```
0.0200,0.0371,0.0428,0.0207,0.0954,0.0986,0.1539,0.1601,0.3109,0.2111,0.1609,0.1582,0.2238
,0.0645,0.0660,0.2273,0.3100,0.2999,0.5078,0.4797,0.5783,0.5071,0.4328,0.5550,0.6711,0.641
5,0.7104,0.8080,0.6791,0.3857,0.1307,0.2604,0.5121,0.7547,0.8537,0.8507,0.6692,0.6097,0.49
43,0.2744,0.0510,0.2834,0.2825,0.4256,0.2641,0.1386,0.1051,0.1343,0.0383,0.0324,0.0232,0.0
027,0.0065,0.0159,0.0072,0.0167,0.0180,0.0084,0.0090,0.0032   :Rock
0.0260,0.0363,0.0136,0.0272,0.0214,0.0338,0.0655,0.1400,0.1843,0.2354,0.2720,0.2442,0.1665
,0.0336,0.1302,0.1708,0.2177,0.3175,0.3714,0.4552,0.5700,0.7397,0.8062,0.8837,0.9432,1.000
0,0.9375,0.7603,0.7123,0.8358,0.7622,0.4567,0.1715,0.1549,0.1641,0.1869,0.2655,0.1713,0.09
59,0.0768,0.0847,0.2076,0.2505,0.1862,0.1439,0.1470,0.0991,0.0041,0.0154,0.0116,0.0181,0.0
146,0.0129,0.0047,0.0039,0.0061,0.0040,0.0036,0.0061,0.0115   :Mine
```

รูปที่ 4.2 ตัวอย่างชุดข้อมูล Sonar: Mines vs. Rocks

3. Ionosphere database

เป็นข้อมูลที่ได้จากสัญญาณเรดาร์ ที่ถูกส่งออกไปตรวจสอบสภาพชั้นบรรยากาศในชั้น Ionosphere โดยสามารถจำแนกออกได้ 2 ประเภท คือสภาพชั้นบรรยากาศดีกับสภาพชั้นบรรยากาศเสีย โดย Attribute ทั้งหมดในแต่ละข้อมูลเท่ากับ 34 และจำนวนของข้อมูลทั้งหมดเท่ากับ 351 โดยจะแบ่งเป็น Training data เท่ากับ 200 และ Testing data เท่ากับ 151

1,0,0.99539,-0.05889,0.85243,0.02306,0.83398,-0.37708,1,0.03760,0.85243,-0.17755,0.59755,-0.44945,0.60536,-0.38223,0.84356,-0.38542,0.58212,-0.32192,0.56971,-0.29674,0.36946,-0.47357,0.56811,-0.51171,0.41078,-0.46168,0.21266,-0.34090,0.42267,-0.54487,0.18641,-0.45300 :good

1,0,1,-0.18829,0.93035,-0.36156,-0.10868,-0.93597,1,-0.04549,0.50874,-0.67743,0.34432,-0.69707,-0.51685,-0.97515,0.05499,-0.62237,0.33109,-1,-0.13151,-0.45300,-0.18056,-0.35734,-0.20332,-0.26569,-0.20468,-0.18401,-0.19040,-0.11593,-0.16626,-0.06288,-0.13738,-0.02447 :bad

รูปที่ 4.3 ตัวอย่างชุดข้อมูล Ionosphere database

4. SPECTF heart data

ข้อมูลการวินิจฉัยโรคหัวใจจากภาพถ่าย Single proton Emission Computed Tomography (SPECT) โดยจะแบ่งคนไข้เป็น 2 Class คือปกติกับไม่ปกติ โดยในแต่ละข้อมูลจะมีอยู่ด้วยกัน 44 attribute ซึ่งแต่ละ attribute จะเป็นเลขจำนวนเต็มตั้งแต่ 0 ถึง 100 และมีจำนวนข้อมูลทั้งหมด 349 โดยแบ่งเป็น Training data เท่ากับ 80 และ Testing data เท่ากับ 269

68,64,65,68,63,64,77,73,75,72,80,77,70,71,61,61,73,68,63,62,76,73,69,69,48,59,62,44,66,59,75,74,64,64,63,61,70,69,74,67,51,48,45,45 :0

62,67,64,70,59,58,67,74,60,66,68,68,73,71,60,63,64,74,64,65,74,77,69,73,59,58,58,67,65,69,78,76,61,62,64,67,72,74,71,71,71,69,66,61 :1

รูปที่ 4.4 ตัวอย่างชุดข้อมูล SPECTF heart data

5. Image Segmentation data

ข้อมูลชุดนี้ใช้ในการจำแนกรูปภาพแบบ Outdoor จำนวน 7 รูปประกอบไปด้วย Brickface, Sky, Foliage, Cement, Window, Path, Grass โดยแต่ละรูปมีขนาด 3x3 region ขนาดของ Training data ในแต่ละภาพเท่ากับ 30 รวมทั้งสิ้น 210 ภาพและขนาดของ Testing data ในแต่ละภาพเท่ากับ 300 รวมทั้งสิ้น 2100 ภาพ จำนวนของ Attribute เท่ากับ 19 โดยตัวอย่างของ attribute ที่ใช้เช่น ค่าเฉลี่ยของสี, ค่าเฉลี่ยของความสว่าง, ค่าสีแดง, ค่าสีน้ำเงิน, ค่าสีเขียว, ขนาดของภาพ เป็นต้น

BRICKFACE,140.0,125.0,9,0.0,0.0,0.2777779,0.06296301,0.66666675,0.31111118,6.185185,7.3
333335,7.6666665,3.5555556,3.444444,4.444447,-7.88889,7.777777,0.5456349,-1.1218182
SKY,140.0,25.0,9,0.0,0.0,0.99999875,1.4666697,1.1111107,0.118517555,128.0,117.77778,142.3
3334,123.888885,-30.666666,43.0,-12.333333,142.33334,0.17250364,-2.3545518
FOLIAGE,101.0,121.0,9,0.11111111,0.0,0.6666667,0.843274,1.5,0.88819396,3.4074075,1.11111
12,6.0,3.1111112,-6.888889,7.7777777,-0.8888889,6.0,0.8435185,-2.5191648
CEMENT,191.0,119.0,9,0.0,0.0,1.1111107,1.2938615,0.9444459,0.772202,39.851852,36.22222,4
8.22222,35.11111,-10.888889,25.11111,-14.222222,48.22222,0.27171725,-2.0059998
WINDOW,189.0,144.0,9,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0
PATH,134.0,187.0,9,0.0,0.0,1.8888887,1.5153537,2.1111114,1.7847083,59.37037,52.666668,74.
111115,51.333332,-20.11111,44.22222,-24.11111,74.111115,0.30745777,-2.03348
GRASS,223.0,185.0,9,0.0,0.0,0.5,0.34960312,2.3888886,2.0807757,12.962963,11.555555,9.7777
8,17.555555,-4.2222223,-9.55555,13.777778,17.555555,0.44541803,1.8388497

รูปที่ 4.5 ตัวอย่างชุดข้อมูล Image Segmentation data

6. Wine recognition

ข้อมูลชุดนี้เป็นการจดจำไวน์ 3 ชนิดซึ่งจะวิเคราะห์ความแตกต่างได้จากองค์ประกอบทางเคมี โดยข้อมูลไวน์เหล่านี้จะมีอยู่ด้วยกัน 13 attribute เช่น แอลกอฮอล์, สี, ความเปรี้ยว เป็นต้น ซึ่งข้อมูลทั้งหมดจะมีอยู่ด้วยกัน 3 กลุ่ม กลุ่มแรกเท่ากับ 59 กลุ่มที่สองเท่ากับ 71 และกลุ่มสุดท้ายเท่ากับ 48 โดยที่จะแบ่งเป็น Training data เท่ากับ 90 และ Testing data เท่ากับ 88

1: 14.23,1.71,2.43,15.6,127,2.8,3.06,.28,2.29,5.64,1.04,3.92,1065
1: 13.2,1.78,2.14,11.2,100,2.65,2.76,.26,1.28,4.38,1.05,3.4,1050
2: 12.37,.94,1.36,10.6,88,1.98,.57,.28,.42,1.95,1.05,1.82,520
2: 12.33,1.1,2.28,16,101,2.05,1.09,.63,.41,3.27,1.25,1.67,680
3: 13.17,2.59,2.37,20,120,1.65,.68,.53,1.46,9.3,.6,1.62,840
3: 14.13,4.1,2.74,24.5,96,2.05,.76,.56,1.35,9.2,.61,1.6,560

รูปที่ 4.6 ตัวอย่างชุดข้อมูล Wine recognition

7. Iris database

ข้อมูลมูลชุดนี้ได้รับความนิยมกันมากในงานด้าน Classification โดยเป็นข้อมูลของดอกไม้ Iris จำนวน 3 ชนิด โดยแต่ละชนิดจะมีอยู่ 4 attribute คือ sepal length, sepal width, petal length,

petal width ข้อมูลทั้งหมดมีอยู่ด้วยกัน 150 โดยแบ่งเป็น Training data เท่ากับ 120 และ Testing data เท่ากับ 30

5.3	3.7	1.5	0.2	Iris-setosa
5	3.3	1.4	0.2	Iris-setosa
7	3.2	4.7	1.4	Iris-versicolor
6.4	3.2	4.5	1.5	Iris-versicolor
6.2	3.4	5.4	2.3	Iris-virginica
5.9	3	5.1	1.8	Iris-virginica

รูปที่ 4.7 ตัวอย่างชุดข้อมูล Iris database

8. Haberman's Survival Data

เป็นชุดข้อมูลเพื่อทำนายการตายจากการผ่าตัดเร็งเต้านมในระหว่างปี ค.ศ. 1958 ถึง 1970 ซึ่งมีจำนวนทั้งหมด 306 ข้อมูล ข้อมูลละ 3 attribute ได้แก่ อายุของคนไข้, จำนวนปีที่เข้ารับการรักษา, จำนวนเต้านมที่ตรวจพบว่าเป็นมะเร็ง และมีอยู่ด้วยกัน 2 Class โดยที่ Class 1 หมายถึงมีชีวิตอยู่ตั้งแต่ 5 ปีขึ้นไป Class 2 หมายถึงเสียชีวิตภายในระยะเวลา 5 ปี โดยจะแบ่งเป็น Training data เท่ากับ 214 และ Testing data เท่ากับ 92

30,62,3 :1
30,65,0 :1
78,65,1 :2
83,58,2 :2

รูปที่ 4.8 ตัวอย่างชุดข้อมูล Haberman's Survival Data

4.1 เปรียบเทียบผลการทดลองระหว่าง FENN กับ Back-propagation และ RBF

สำหรับการเปรียบเทียบผลการทดลองของทั้ง 3 Models นี้ได้กำหนดพารามิเตอร์ที่ใช้ในการทดลองของแต่ละ Model ดังต่อไปนี้

1. พารามิเตอร์ของ Back-propagation Neural Network

Learning rate = 0.01

Learning Algorithm = Gradient descent

จำนวนรอบในการฝึกสอน = 5000

จำนวน Layer = 3 (1 Input layer, 1 Hidden layer, 1 Output layer)

Transfer function = sigmoid ทั้ง Hidden และ Output layers

จำนวน Node ในชั้น Hidden layer = 20%, 30%, 40% ของ Training data

2. พารามิเตอร์ของ Radial Basis Function Neural Network

Learning Algorithms = - ใช้ EM algorithm ในการหาจุด Center

- ใช้ Mean of nearest neighbour distance ในการหาค่า Standard deviation (σ)
- ใช้ Pseudo-inverse ในการปรับค่า Weight

จำนวนรอบในการฝึกสอน = 1000

จำนวน Node ในชั้น Hidden layer = 20%, 30%, 40% ของ Training data

3. พารามิเตอร์ของ Fuzzy and Evolutionary Neural Network

Mutation rate = 0.05

จำนวนของ Chromosome = 20 ถึง 30 เส้น

จำนวนรอบในการฝึกสอน = 1000

จำนวน Node เริ่มต้นในชั้น Hidden layer = 20%, 30%, 40% ของ Training data

Boundary ของ Hidden node = $\pm$ 20% ของ Node เริ่มต้น

ตารางที่ 4.1 ผลการทดลองของ Back-propagation, RBF และ FENN กับชุดข้อมูลที่ใช้ในการทดสอบ

ชุดข้อมูลที่ใช้ในการทดสอบ (Testing Data) กับจำนวน pattern	Back-propagation		Radial Basis Function		Fuzzy and Evolutionary	
	Accuracy (%)	จำนวน Node	Accuracy (%)	จำนวน Node	Accuracy (%)	จำนวน Node
1. Vowel (462)	48.48	211	59.09	211	59.74	249
2. Sonar (103)	71.84	21	63.11	42	83.5	49
3. Ionosphere (151)	93.38	40	97.35	60	98.68	95
4. SPECTF (269)	77.32	24	74.35	32	81.78	29
5. Image (2100)	87.29	63	90.05	84	77.95	99
6. Wine (88)	92.05	18	88.64	27	70.45	43
7. Iris (30)	100	36	100	24	100	26
8. Haberman (92)	66.3	64	64.13	86	70.65	49

ตารางที่ 4.1 เป็นตารางผลการทดลองของ Back-propagation, Radial Basis Function Neural Networks และ Fuzzy and Evolutionary Neural Network โดยทดลองกับชุดข้อมูลทั้ง 8 ชุด ซึ่งนำเสนอในส่วนของข้อมูลที่ใช้ในการทดสอบ (Testing data) โดยแสดงผลที่ได้ในรูปเปอร์เซ็นต์ความถูกต้อง (Accuracy) และจำนวนของ Node ในชั้น Hidden layer ของแต่ละ Model ที่ให้เปอร์เซ็นต์ความถูกต้องกับชุดข้อมูลในการเรียนรู้ (Training data) มากที่สุด

จากชุดข้อมูลทั้ง 8 ชุดสามารถแบ่งชุดข้อมูลออกได้เป็น 2 กลุ่มคือ

กลุ่มที่ 1 ได้แก่ Vowel, Sonar, Ionosphere และ SPECTF โดยกลุ่มข้อมูลนี้จะดึงคุณลักษณะ (Feature) โดยผ่านฟังก์ชั่นทางคณิตศาสตร์หรือผ่านการ Pre-processing มาก่อนทำให้ไม่สามารถบอกถึงความหมายในแต่ละ Attribute ได้

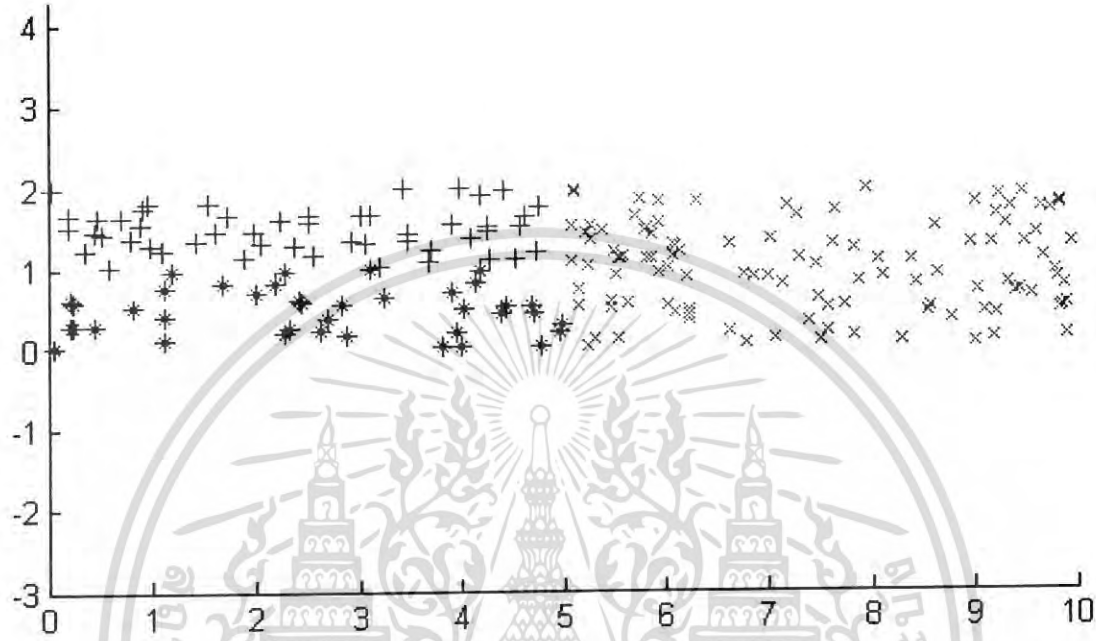
กลุ่มที่ 2 ได้แก่ Image Segmentation, Wine, Iris และ Haberman's Survival โดยกลุ่มข้อมูลนี้จะไม่ผ่านการ Pre-processing ทำให้ Feature ในแต่ละ Attribute สามารถบอกถึงความหมายของข้อมูลได้ เช่น Iris data ในแต่ละ Attribute จะบอกถึงความกว้างความยาวของกลีบดอก Iris เป็นต้น

จากผลการทดลองในตารางที่ 4.1 จะพบว่าในชุดข้อมูลกลุ่มที่ 1 นั้น FENN จะให้ค่า Accuracy ที่ดีกว่า Back-propagation และ Radial Basis Function

ส่วนข้อมูลในกลุ่มที่ 2 นั้น ชุดข้อมูล Image Segmentation และ Wine ให้ค่า Accuracy ของ FENN น้อยกว่า Back-propagation และ Radial Basis Function สาเหตุอันเนื่องมาจากในช่วงข้อมูลของแต่ละ Attribute สำหรับชุดข้อมูล Image Segmentation และ Wine นั้นมีความแตกต่างของช่วงข้อมูลที่สูงมากทำให้ ค่า Standard deviation (σ) ไม่ครอบคลุม Attribute ทีมีช่วงข้อมูลที่สูง เนื่องจาก FENN จะใช้ Euclidian distance ระหว่าง Center ที่อยู่ต่าง Class กันมากำหนดเป็นค่า Standard deviation (σ) ให้แต่ละ Center (แต่ละ Center มีค่า σ แตกต่างกัน) ดังนั้นการกระจายตัวของ Gaussian function ในบาง Center ที่มีจุด Center ของกลุ่มอื่นอยู่ใกล้เคียง จะให้ค่า Standard deviation ที่ได้มีค่าน้อย จนไม่ครอบคลุมไปถึง Attribute อื่นที่มีช่วงข้อมูลที่สูงกว่า ต่างกับ Radial Basis Function ที่ใช้ค่าเฉลี่ยของ Euclidian distance จากแต่ละ Center ที่อยู่ใกล้ที่สุดมากำหนดเป็นค่า Standard deviation (แต่ละ Center ใช้ค่า σ เดียวกัน) ทำให้การกระจายตัวของ Gaussian function จะเฉลี่ยกันไปในทุก Attribute

ในส่วนของชุดข้อมูล Iris และ Haberman's Survival ซึ่งเป็นข้อมูลในกลุ่มที่ 2 เช่นกันแต่กลับให้ผลการทดลองที่ดีกว่า อันเนื่องมาจากข้อมูลในแต่ละ Attribute มีค่าและช่วงของข้อมูลใกล้เคียงกัน ซึ่งจะคล้ายกับข้อมูลในกลุ่มที่ 1 ทำให้การกระจายตัวของ Gaussian function ครอบคลุมเกือบทุก Attribute ทำให้ผลการทดลองที่ได้ใกล้เคียงกับ Back-propagation และ Radial Basis Function Neural Networks

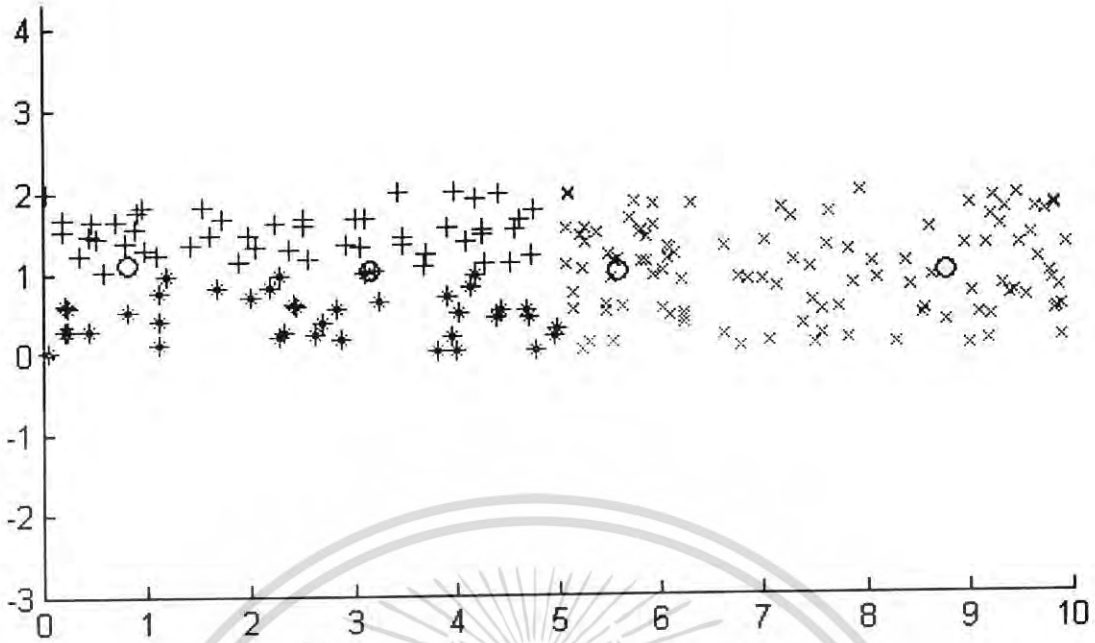
เพื่อให้การอธิบายมีความสมบูรณ์มากยิ่งขึ้น จึงขอยกตัวอย่างชุดข้อมูลที่มีจำนวน Attribute เท่ากับ 2 (แกน X และแกน Y) ประกอบด้วยข้อมูลทั้งหมด 200 patterns มีจำนวน Class เท่ากับ 3 (Class 1 = *, Class 2 = +, Class 3 = x) ดังแสดงในรูปที่ 4.9



รูปที่ 4.9 แสดงข้อมูลจากตัวอย่างชุดข้อมูลที่ใช้ในการอธิบาย

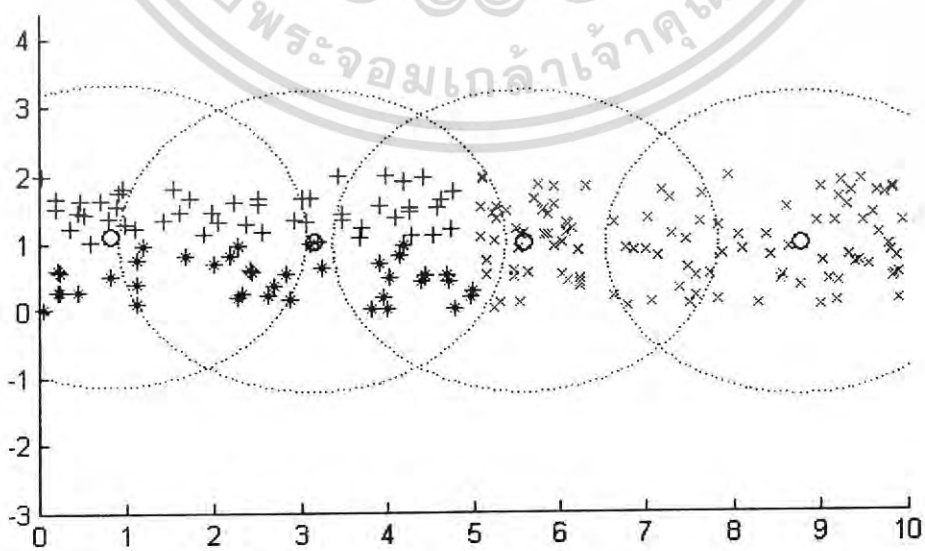
จากรูปที่ 4.9 จะพบว่าชุดข้อมูลนี้ Attribute ในแกน X (แนวนอน) จะมีช่วงของข้อมูลกว้างกว่า Attribute ในแกน Y (แนวตั้ง) โดยแนวแกน X จะมีค่าระหว่าง 0 ถึง 10 แนวแกน Y จะมีค่าระหว่าง 0 ถึง 2 โดยข้อมูลทั้งหมดเกิดจากการสุ่ม เพื่อให้ง่ายต่อการอธิบายการกระจายตัวของข้อมูลในแต่ละ Class จะแบ่งแยกกันอย่างชัดเจนโดยแต่ละ Class จะแสดงด้วยรูปสัญลักษณ์ที่แตกต่างกัน ดังรูปที่ 4.9

รูปที่ 4.10 แสดงจุด Center ที่ได้จากการหาจุด Center ของ Radial Basis Function (โดยจุด Center = o และกำหนดให้หาจุด Center = 4จุด) จะพบว่าจุด Center ที่ได้จะกระจายอยู่ในกลุ่มข้อมูลในระยะที่เท่าๆกัน ทั้งนี้เนื่องจากการหาจุด Center ของ Radial Basis Function จะมองข้อมูลทั้งหมดอยู่ใน Class เดียวกัน ทำให้จุด Center ที่เกิดขึ้นเป็นจุด Center ของข้อมูลทั้งหมดไม่ใช่ของ Class ใด Class หนึ่ง



รูปที่ 4.10 แสดงจุด Center ที่ได้จาก Radial Basis Function

จากรูปที่ 4.11 แสดงรัศมีของแต่ละ Center ที่ได้จากการหาจุด Center ของ Radial Basis Function โดยค่ารัศมีเหล่านี้ถูกนำมากำหนดเป็นค่า Standard deviation (σ) โดยใช้ค่าเฉลี่ยของ Euclidian distance ระหว่าง Center ที่อยู่ใกล้ที่สุดมากำหนดเป็นค่าดังกล่าว ซึ่งค่า σ นี้จะมีค่าเท่ากันในทุกๆ Center จากรูปจะพบว่ารัศมีของแต่ละ Center จะครอบคลุมข้อมูลทั้งหมด เนื่องจากการกระจายตัวของแต่ละ Center จะกระจายตัวเท่าๆกันในชุดข้อมูล ทำให้ระยะทางระหว่างแต่ละ Center มีค่าใกล้เคียงกันในทุกๆจุด จึงทำให้ค่า σ ของแต่ละ Center สามารถครอบคลุมข้อมูลในทุก Attribute ได้



รูปที่ 4.11 รัศมีของแต่ละ Center ที่ได้จาก Radial Basis Function

หากมองเพียงจุด Center ที่ได้จากทั้ง 2 Model จะพบว่า FENN ควรจะสามารถจำแนกข้อมูลในแต่ละ Class ได้ดีกว่า Radial Basis Function แต่ยังมีอีกสิ่งที่จะต้องพิจารณาควบคู่กันนั้นคือค่า Standard deviation (σ) จากรูปที่ 4.13 แสดงรัศมีของแต่ละจุด Center ที่หาโดยวิธี FENN ซึ่งค่ารัศมีเหล่านี้คือค่า σ นั้นเองโดยสามารถหาได้จาก Euclidian distance ระหว่าง Center ที่อยู่ต่าง Class กันมากำหนดเป็นค่า σ ทำให้แต่ละ Center มีค่า σ แตกต่างกันไปดังรูปที่ 4.13 จะพบว่าระยะทางระหว่าง Center + และ Center * จะอยู่ใกล้กันมากเนื่องจาก Attribute ในแนวแกน Y มีช่วงข้อมูลอยู่ในค่าน้อยทำให้ค่า σ ที่ได้มีค่าน้อยตามระยะทางเช่นกัน จนไม่สามารถครอบคลุมในทุก Attribute ของข้อมูลใน Class ตนเองได้ ซึ่งตรงกันข้ามกับ Center x ซึ่งมีระยะทางไกลจาก Center ของ Class อื่น (Center +, Center *) เนื่องจาก Attribute ในแนวแกน X มีช่วงข้อมูลที่ห่างกันมากทำให้ค่า σ ที่ได้มีค่ามาก ซึ่งนอกจากจะครอบคลุมทุก Attribute ของข้อมูลใน Class ตนเองแล้วยังไปคาบเกี่ยว (Over lap) กับข้อมูลใน Class อื่นทำให้การจำแนกข้อมูลผิดพลาดได้ง่าย

ซึ่งการแก้ปัญหาสามารถกระทำได้โดยนำชุดข้อมูลไปกระทำการ Normalization เสียก่อนดังจะแสดงในหัวข้อการทดลองถัดไป

4.2 เปรียบเทียบผลการทดลองกับข้อมูลที่ผ่านการ Normalization

ในการทดลองนี้จะเป็นการเปรียบเทียบผลการทดลองของ Fuzzy and Evolutionary Neural Network โดยใช้ชุดข้อมูลที่ผ่านการ Normalization กับข้อมูลที่ไม่ผ่านการ Normalization ซึ่งการทำ Normalization กับชุดข้อมูล เป็นการทำให้ทุก Attribute มีช่วงข้อมูลอยู่ในระหว่าง 0 ถึง 1 โดยมีพารามิเตอร์ดังนี้

1. พารามิเตอร์ของ Fuzzy and Evolutionary Neural Network

   Mutation rate = 0.05

   จำนวนของ Chromosome = 20 ถึง 30 เส้น

   จำนวนรอบในการฝึกสอน = 1000

   จำนวน Node เริ่มต้นในชั้น Hidden layer = 20%, 30%, 40% ของ Training data

   Boundary ของ Hidden node = $\pm$ 20% ของ Node เริ่มต้น

ตารางที่ 4.2 ผลการทดลองของ FENN กับชุดข้อมูลที่ไม่ผ่านการ Normalization และชุดข้อมูลที่ผ่านการ Normalization

ชุดข้อมูล และจำนวน pattern (Training data, Testing data)	ข้อมูลไม่ผ่านการ Normalization			ข้อมูลผ่านการ Normalization		
	Accuracy (%) ของ Training data	Accuracy (%) ของ Testing data	จำนวน Node	Accuracy (%) ของ Training data	Accuracy (%) ของ Testing data	จำนวน Node
1. Vowel (528,462)	100	59.74	249	100	53.25	193
2. Sonar (105,103)	98.1	83.5	49	99.05	85.44	59
3.Ionosphere(200,151)	99	98.68	95	97	95.36	83
4. SPECTF (80/269)	98.75	81.78	29	87.5	84.76	20
5. Image (210,2100)	95.24	77.95	99	98.57	90.52	59
6. Wine (90,88)	94.44	70.45	43	100	97.73	37
7. Iris (120,30)	100	100	26	99.17	100	21
8. Haberman (214,92)	93.46	70.65	49	90.65	69.57	88

จากตารางที่ 4.2 จะพบว่าผลการทดลองในส่วนของ Testing data กับข้อมูลในกลุ่มที่ 1 (แบ่งกลุ่มตามการทดลองที่ 4.1) ส่วนใหญ่ให้ Accuracy กับข้อมูลที่ไม่ผ่านการ Normalization มากกว่าข้อมูลที่ผ่านการ Normalization เล็กน้อยทั้งนี้เนื่องจากข้อมูลในกลุ่มนี้ได้ผ่านการทำ Pre-processing มาก่อนแล้ว ทำให้ช่วงข้อมูลในแต่ละ Attribute ไม่แตกต่างกันมากนัก เมื่อผ่านการ Normalization อาจทำให้สูญเสียคุณลักษณะบางประการไป เพราะจะต้องบีบข้อมูลให้อยู่ในช่วง 0 ถึง 1

ส่วนข้อมูลในกลุ่มที่ 2 (แบ่งกลุ่มตามการทดลองที่ 4.1) สำหรับชุดข้อมูล Iris และ Haberman ให้ผลการทดลองกับ Testing data ทั้งแบบที่ผ่านการ Normalization และไม่ผ่านการ Normalization ไม่แตกต่างกัน เนื่องจาก Attribute มีจำนวนน้อยและช่วงข้อมูลในแต่ละ Attribute อยู่ในช่วงใกล้เคียงกัน ซึ่งแตกต่างกับชุดข้อมูล Image Segmentation และ Wine โดยข้อมูลที่ผ่านการ Normalization จะมี Accuracy ของ Testing data มากกว่าข้อมูลที่ไม่ผ่านการ Normalization ทั้งนี้เพราะการ Normalization จะทำให้ช่วงข้อมูลในแต่ละ Attribute มีค่าอยู่ในช่วงเดียวกัน (ระหว่าง 0 ถึง 1) ต่างจากเดิมที่ช่วงข้อมูลแต่ละ Attribute ของข้อมูลทั้ง 2 ชุดนี้มีความแตกต่างกันมาก (บาง Attribute อยู่ในหลักร้อยบาง Attribute อยู่ในช่วงทศนิยม) และเมื่อ Attribute มีค่าอยู่ในช่วงเดียวกันทำให้การกระจายตัว Gaussian function ครอบคลุมในทุก Attribute ทำให้การหาจุด Center และค่า Standard deviation (σ) เป็นไปได้โดยง่าย ซึ่งจะสังเกตได้จากตารางที่ 4.2 ในส่วน

ของ Training data ของ Image segment และ Wine นั้นจะให้ Accuracy กับข้อมูลที่ผ่านการ Normalization มากกว่าข้อมูลที่ไม่ผ่านการ Normalization อีกทั้งจำนวน Hidden node จะมีจำนวนที่น้อยกว่าอีกด้วย

4.3 เปรียบเทียบผลการทดลองแบบกำหนดขอบเขตของ Hidden node

ในการทดลองส่วนนี้เป็นการทดลองเปรียบเทียบผลระหว่าง FENN แบบกำหนด Boundary และแบบไม่กำหนด Boundary ของการเพิ่มหรือลดจำนวน Hidden node ในชั้น Hidden layer เพื่อหาความสัมพันธ์ระหว่างอัตราการเพิ่มหรือลดจำนวน Hidden node กับ Accuracy ของ Training data โดยกำหนดพารามิเตอร์ต่างๆดังต่อไปนี้

1. พารามิเตอร์ของ Fuzzy and Evolutionary Neural Network แบบกำหนด Boundary

   Mutation rate = 0.05

   จำนวนของ Chromosome = 20 ถึง 30 เส้น

   จำนวนรอบในการฝึกสอน = 1000

   จำนวน Node เริ่มต้นในชั้น Hidden layer = 20%, 30%, 40% ของ Training data

   Boundary ของ Hidden node = $\pm$ 20% ของ Node เริ่มต้น

2. พารามิเตอร์ของ Fuzzy and Evolutionary Neural Network แบบไม่กำหนด Boundary

   Mutation rate = 0.05

   จำนวนของ Chromosome = 20 ถึง 30 เส้น

   จำนวนรอบในการฝึกสอน = 2000

   จำนวน Node เริ่มต้นในชั้น Hidden layer = 20%, 30%, 40% ของ Training data

   Boundary ของ Hidden node = ไม่กำหนด

ตารางที่ 4.3 ผลการทดลองของ FENN แบบกำหนด Boundary และไม่กำหนด Boundary ของการเพิ่มหรือลดจำนวน Hidden node

ชุดข้อมูล และจำนวน pattern (Training data, Testing data)	กำหนด Boundary			ไม่กำหนด Boundary		
	Accuracy (%) ของ Training data	Accuracy (%) ของ Testing data	จำนวน Node	Accuracy (%) ของ Training data	Accuracy (%) ของ Testing data	จำนวน Node
1. Vowel (528,462)	100	59.74	249	100	60.82	351
2. Sonar (105,103)	98.1	83.5	49	100	82.52	117
3.Ionosphere(200,151)	99	98.68	95	99.5	98.01	115
4. SPECTF (80/269)	98.75	81.78	29	98.75	78.07	89
5. Image (210,2100)	95.24	77.95	99	96.19	81.38	158
6. Wine (90,88)	94.44	70.45	43	95.56	69.32	97
7. Iris (120,30)	100	100	26	100	100	92
8. Haberman (214,92)	93.46	70.65	49	94.39	64.13	195

จากตารางที่ 4.3 จะพบว่าการปล่อย Boundary จะทำให้ Accuracy ของ Training data นั้นดีขึ้น แต่จำนวน Node ก็จะมากขึ้นตามไปด้วยเช่นกัน จนในบางชุดข้อมูลจำนวน Node เพิ่มขึ้นจนใกล้เคียงกับจำนวน Pattern ของ Training data

ถึงแม้ว่าการปล่อย Boundary จะช่วยให้ Accuracy ของ Training data นั้นดีขึ้นแต่เมื่อเทียบกับการกำหนด Boundary จะพบว่าต่างกันเพียงเล็กน้อยเท่านั้น และการเพิ่มขึ้นของ Accuracy นี้จะทำให้เกิดการ Over fit ซึ่งมีผลกระทบต่อ Accuracy ของ Testing data (Accuracy ของ Testing data ลดต่ำลง) และจำนวน Node ที่มีมากกว่า จะทำให้การประมวลผลของระบบช้าลงอีกด้วย

บทที่ 5

สรุปผลการทดลอง

วิทยานิพนธ์ฉบับนี้ได้นำเสนอ Fuzzy and Evolutionary Neural Network (FENN) ซึ่งเป็นโครงข่ายประสาทเทียมชนิดใหม่ที่ใช้ในงาน Classification โดยประยุกต์ใช้ทฤษฎีของ Fuzzy C-Mean Clustering และ Genetic Algorithm ร่วมกับ Neural Network และเปรียบเทียบประสิทธิภาพกับโครงข่ายประสาทเทียมอื่นอีก 2 ชนิด คือ Back-propagation และ Radial Basis Function จากการทดลองสามารถสรุปข้อดีข้อเสียรวมถึงข้อเสนอแนะดังต่อไปนี้

1. ประเภทข้อมูลที่เหมาะสมกับ FENN

จากการทดลองในหัวข้อที่ 4.1 จะพบว่าข้อมูลทั้ง 8 ชุดที่ใช้ในการทดลองจะต้องเป็นข้อมูลชนิดตัวเลขที่เป็นจำนวนจริงเท่านั้นจึงจะสามารถใช้กับ FENN ได้โดยข้อมูลเหล่านี้ประกอบไปด้วยข้อมูลที่ผ่านและไม่ผ่านกระบวนการ Pre-processing ซึ่งสามารถแบ่งลักษณะของข้อมูลออกเป็น 2 กลุ่มใหญ่ๆคือ 1.ข้อมูลในแต่ละ Attribute มีช่วงของข้อมูลที่ใกล้เคียงกัน 2.ข้อมูลในแต่ละ Attribute มีช่วงของข้อมูลที่แตกต่างกัน จากการทดลองจะพบว่า FENN จะเหมาะสมกับข้อมูลที่ข้อมูลในแต่ละ Attribute มีช่วงของข้อมูลที่ใกล้เคียงกันมากที่สุด เนื่องจากข้อมูลเหล่านี้จะเหมาะกับโครงข่ายที่ใช้สมการ Gaussian function ที่มีค่า Standard deviation (σ) เพียงค่าเดียวในแต่ละจุด Center สังเกตได้จากการทดลองที่ 4.2 จะพบว่าหากนำชุดข้อมูลที่ข้อมูลในแต่ละ Attribute มีช่วงของข้อมูลที่แตกต่างกัน ไปทำการ Normalization (การ Normalization เป็นกระบวนการ Pre-processing แบบหนึ่ง) จะทำให้ผลการทดลองดีขึ้น

2. ข้อแตกต่างระหว่าง FENN กับ Back-propagation และ Radial Basis Function

ข้อแตกต่างที่ทำให้ FENN ดีกว่า Back-propagation และ Radial Basis Function คือ FENN จะหาจุด Center ในแต่ละ Class ทำให้จุด Center ที่ได้คือจุด Center ของ Class นั้นจริงๆจึงไม่จำเป็นต้องปรับค่าน้ำหนัก (Weight) ระหว่างชั้น Hidden กับ Output layer

แม้ว่าจะไม่ต้องเสียเวลาในการปรับค่า Weight แต่ FENN จะใช้เวลาในหนึ่งรอบของการเรียนรู้ (Training) มากกว่า Back-propagation และ Radial Basis Function เนื่องจาก FENN จะเสียเวลาให้กับกระบวนการ Genetic Algorithm ที่มีอยู่หลายขั้นตอน แต่จำนวนรอบของการเรียนรู้จะเท่าๆกับ Radial Basis Function และน้อยกว่า Back-propagation

ข้อแตกต่างอีกข้อหนึ่งที่มีผลต่อประสิทธิภาพสำหรับโครงข่ายที่ใช้จุด Center ในการจำแนกกลุ่มนั้นก็คือค่า σ ถึงแม้ Radial Basis Function และ FENN จะใช้ Euclidian distance ระหว่างจุด Center มากำหนดเป็นค่า σ เหมือนกันแต่ตำแหน่งจุด Center ของ Radial Basis

Function จะกระจายกันอยู่ทั่วทั้งชุดข้อมูลเนื่องจาก Radial Basis Function จะพิจารณาข้อมูลทั้งชุดเสมือนอยู่ใน Class เดียวกันทำให้ระยะทางระหว่างจุด Center ใกล้เคียงกันค่า σ จึงครอบคลุมทุกข้อมูล แม้ว่าข้อมูลชุดนั้นจะมีความแตกต่างกันมากของช่วงข้อมูลในแต่ละ Attribute ซึ่งต่างจาก FENN ที่แต่ละจุด Center จะหาโดยการวิเคราะห์จากข้อมูลที่อยู่ใน Class เดียวกัน ทำให้ไม่เหมาะสมกับชุดข้อมูลที่มีช่วงของข้อมูลในแต่ละ Attribute ที่แตกต่างกันมาก (ส่วนใหญ่เป็นข้อมูลที่ไม่ได้ผ่านกระบวนการ Pre-processing) เพราะบางจุด Center ที่อยู่ต่าง Class กันแต่อยู่ใกล้กันจะมีค่า σ น้อยจนไม่ครอบคลุมข้อมูลที่อยู่ใน Class เดียวกันแต่มี Attribute อยู่ในช่วงสูงๆได้จนทำให้การจำแนกกลุ่มผิดพลาดไป

3. การกำหนดขอบเขตจำนวน Hidden node ของ FENN

จากการทดลองในหัวข้อที่ 4.3 ทำให้ทราบว่าการกำหนดขอบเขต (Boundary) ให้ประสิทธิภาพทั้งในด้านความถูกต้องของข้อมูลที่ใช้ในการทดสอบ (Testing data) ที่ดีกว่าและจำนวน Hidden node หรือจำนวน Center ที่น้อยกว่าการไม่กำหนดขอบเขต สาเหตุอันเนื่องมาจากจำนวน Center ที่เพิ่มขึ้นจะทำให้โครงข่ายเกิดการ Over fit มากจนเกินไป ทำให้ประสิทธิภาพในด้านความถูกต้องของข้อมูลที่ใช้ในการทดสอบลดลง และเนื่องจากจำนวนของจุด Center ก็คือจำนวนของ Hidden node ในชั้น Hidden layer ฉะนั้นหากจำนวนของจุด Center มากขึ้นนั้นก็หมายความว่าเวลาในการประมวลผลในแต่ละรอบของการฝึกสอนจะนานขึ้นตามไปด้วย

จากการทดลองที่ 4.1 จะพบว่าผลการทดลองที่ดีที่สุดได้จากการกำหนดจำนวนของ Hidden node เริ่มต้นสำหรับการฝึกสอนสำหรับ FENN ที่ประมาณ 40% ของจำนวน Training patterns โดยมีการเพิ่มหรือลดของจำนวน Node ในระหว่างการฝึกสอนที่ 20% จาก Node เริ่มต้น ซึ่งในกรณีดังกล่าว ได้ค่าเฉลี่ยของจำนวน Node จะอยู่ที่ประมาณ 47% ในทุกชุดข้อมูลที่ใช้ในการทดลอง อย่างไรก็ดีพบว่าการเพิ่มขึ้นของจำนวน Node นั้นจะเพิ่มขึ้นจนกระทั่งขึ้นไปติด Boundary ทำให้ยังไม่สามารถสรุปได้ว่าจำนวน Node ที่เหมาะสมที่สุดสำหรับ FENN นั้นควรจะอยู่ที่เท่าใด เนื่องจากไม่ได้มีการทดลองกำหนดจำนวน Node เริ่มต้นที่มากกว่า 40 %

4. ข้อเสนอแนะ ในการทำวิจัยครั้งต่อไป

ข้อจำกัดของ FENN คือไม่เหมาะสมกับชุดข้อมูลที่ข้อมูลในแต่ละ Attribute มีช่วงของข้อมูลที่แตกต่างกัน อันเนื่องมาจากผลกระทบของการกำหนดค่า Standard deviation (σ) ซึ่งในโครงข่ายนี้จะจำกัดให้แต่ละจุด Center สามารถมีค่า σ ได้เพียงค่าเดียวเท่านั้นทำให้การกระจายตัวของแต่ละจุด Center ไม่ครอบคลุมข้อมูลในทุก Attribute ดังนั้นหากกำหนดให้มีค่า σ ในแต่ละ Attribute ของแต่ละจุด Center (แต่ละจุด Center มีค่า σ ได้มากกว่า 1) จะทำให้ FENN สามารถใช้กับชุดข้อมูลที่ข้อมูลในแต่ละ Attribute มีช่วงของข้อมูลที่แตกต่างกันได้ดียิ่งขึ้น

เนื่องจาก FENN สามารถเพิ่มจำนวนจุด Center หรือจำนวน Hidden node ได้ ดังนั้นในกรณีที่จุด Center ที่เกิดขึ้นใหม่มีความเหมาะสมหรือเป็นตัวแทนที่ดีกว่าจุด Center ก่อนหน้าจนทำ

ให้จุด Center ก่อนหน้าหมดความสำคัญไป (จุด Center ที่หมดความสำคัญคือจุด Center ที่ไม่เคยชนะในหนึ่งรอบของการฝึกสอน) หากกำจัดจุด Center ดังกล่าวทิ้งจะสามารถลดจำนวน Hidden node ลงได้ เวลาในการประมวลผลก็จะสั้นลงและจุด Center ที่ได้จะเป็นจุด Center ที่เหมาะสมมากที่สุดสำหรับข้อมูลที่ใช้ในการฝึกสอนนั้นๆ

เอกสารอ้างอิง

[1] Steve A. Billings and Guag L. Zheng. "Radial Basis Function Network Configuration Using Genetic Algorithms" **Elsevier Science Ltd.** No.8 1995 pp. 877-890

[2] Mahid Amiri. 2003. **"Fuzzy C-Means Clustering."** [Online] Available : http://mehr.sharif.edu/~amiri/Downloads/Y_FCMC/Y_FCMC_Presentation_Ver0.8.ppt.

[3] Richard O. Duda, Peter E. Hart and David G. Stork. **Pattern Classification** 2nd USA : A Wiley-Interscience Publication. 2000

[4] Daniel Franklin. 2003. **"Back-Propagation Neural Network Tutorial."** [Online]. Available : http://ieee.uow.edu.au/~daniel/software/libneural/BPN_tutorial /BPN_English/ BPN_English/node8.html.

[5] N.K. Bose and P. Liang. **Neural Network Fundamentals with Graphs, Algorithms, Applications** 1st Singapore : McGraw-Hill, Inc. 1996

[6] ดร.อาณัติ ลีมัคเดช. 2547. **"การสร้างพอร์ทหุ้นด้วยวิธีคำนวณเชิงพันธุกรรม."** [Online]. Available : http://mif.bus.tu.ac.th/documents/genetic.pdf.

[7] Tom M. Mitchell. **Machine Learning** 1st Singapore : McGraw-Hill companies,Inc. 1997

[8] Shreekanth Mandayam and Robi Polikar. 2004. **"Artificial Neural Netwoks."** [Online]. Available : http://users.rowan.edu/~shreek/spring02/ann/lectures /RBFlecture1.ppt.

[9] Jose C. Principe, Neil R. Euliano and W. Curt Lefebvre. **Neural and Adaptive Systems** 1st USA : John Wiley & Sons,Inc. 2000

[10] Blake, C. L., Merz, C. J.: 1998. **"UCI Repository of machine learning databases."** [Online]. Available : http//www.ics.uci.edu/~mlearn/MLRepository.html. University of California, Department of Information and Computer Science.

[11] Shotaro Akaho. 1995. **"Mixture model for image understanding and the EM algorithm."** [Online]. Available : http://ieeexplore.ieee.org/iel2/3505/10435/00487742.pdf.

ภาคผนวก

ผลงานวิจัยที่ได้รับการตีพิมพ์

Arit Thammano and Asavin Meengen. "A New Evolutionary Neural Network Classifier". **Proceedings of 9th Pacific-Asia Conference (PAKDD),** May 2005, Springer, pp. 249-255.

A New Evolutionary Neural Network Classifier

Arit Thammano* and Asavin Meengen**

Faculty of Information Technology
King Mongkut's Institute of Technology Ladkrabang,
Bangkok, 10520 Thailand
E-mail: arit@it.kmitl.ac.th* and asv_kmitl@hotmail.com**

Abstract. This paper proposes two new concepts: (1) the new evolutionary algorithm and (2) the new approach to deal with the classification problems by applying the concepts of the fuzzy c-means algorithm and the evolutionary algorithm to the artificial neural network. During training, the fuzzy c-means algorithm is initially used to form the clusters in the cluster layer; then the evolutionary algorithm is employed to optimize those clusters and their parameters. During testing, the class whose cluster node returns the maximum output value is the result of the prediction. This proposed model has been benchmarked against the standard backpropagation neural network, the fuzzy ARTMAP, C4.5, and CART. The results on six benchmark problems are very encouraging.

1 Introduction

In the past decades, data are being collected and accumulated at a dramatic pace. Therefore, there is an urgent need for a new generation of computational techniques and tools to assist humans in extracting useful information (knowledge) from the rapidly growing volumes of data [1]. This arouses many researchers to study into the area of data mining. One of the data mining functionalities, which plays an important role in business decision-making tasks, is classification. Classification is the process of finding a set of models that describe and distinguish data classes or concepts, for the purpose of being able to use the model to predict the class of objects whose class label is unknown [2]. A variety of techniques have been applied to deal with the classification problems, such as neural networks, decision trees, and statistical methods. However, many previous research works show that neural network classifiers have a better performance, lower classification error rate, and more robust to noise than the other two methods mentioned above. The proposed evolutionary neural network classifier described in this paper employs the concept of the fuzzy c-mean clustering and the evolutionary algorithm to find and optimize the center and the standard deviation of each cluster. The performance of the proposed network is evaluated against the fuzzy ARTMAP, the backpropagation neural network, C4.5, and CART.

This paper is organized as follows. Following this introduction, section 2 presents the architecture of the evolutionary neural network classifier. The learning algorithm is described in section 3. In section 4, the experimental results are demonstrated and discussed. Finally, section 5 is the conclusions.

2 The Proposed Model

The architecture of the evolutionary neural network classifier is a three-layer feedforward neural network as shown in Figure 1. The first layer is the input layer, which consists of N nodes. Each node represents a feature component of the input data. The second layer is the cluster layer. The nodes in this second layer are constructed during the training phase; each node represents a cluster that belongs to one of the classes. The third layer is the output layer. Each node in the output layer represents a class. In this paper, the input vector is denoted by $X_i = (x_{i1}, \ldots, x_{iN})$, where i is the i^{th} input pattern, and N is the number of features in X. The nodes in the cluster layer are fully connected to the nodes in the input layer. Therefore, once the model receives the input and its associated target output (X_i, Y_i), the input vector X_i is directly transmitted to the cluster layer via these connections. Each node in the cluster layer then calculates the membership degree to which the input vector X_i belongs to its cluster j.

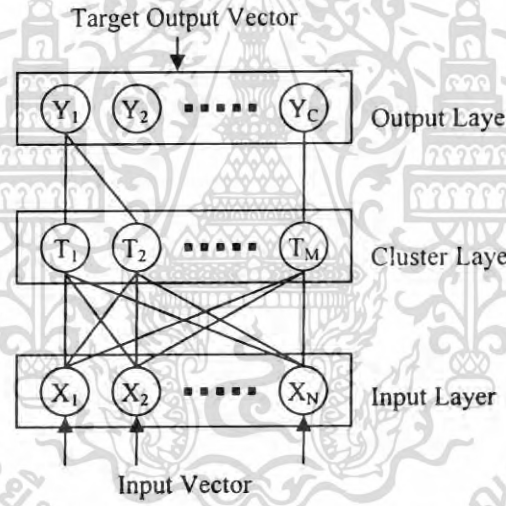


Fig. 1. Architecture of the evolutionary neural network classifier

$$T_j = \mu_j(X_i) = \text{Gaussian}(X_i, c_j, \sigma_j) = \exp\left[\frac{-d_{ji}^2}{2\sigma_j^2}\right]. \qquad (1)$$

where j = 1, 2, ..., M. It denotes the j^{th} node in the cluster layer.

M is the total number of nodes in the cluster layer.

$d_{ji} = \|X_i - c_j\|$. It is the distance between the input vector X_i and the center of the j^{th} cluster.

c_j is the center of the j^{th} cluster.

σ_j is the width of the j^{th} cluster.

The cluster node with the highest degree of membership is selected to be the winning node.

$$T_J = \max\{T_j : j = 1, 2, ..., M\}. \quad (2)$$

Then, the class of the input vector X_i is predicted as the one whose cluster node exhibits the highest degree of membership (the winning node).

3 The Learning Algorithm

During the training period of the evolutionary neural network classifier, two phases of learning are involved: Phase 1 (Initialize the cluster node) and Phase 2 (Optimize the cluster node).

3.1 Phase 1: Initialize the Cluster Node

The fuzzy c-means algorithm is used in this phase to partition the input space into M clusters, which are then used to be the nodes in the cluster layer. The goal of the fuzzy c-means algorithm is to minimize the objective function J_m, which is

$$J_m = \sum_{j=1}^{M}\sum_{i=1}^{P} \mu_{ji}^{m} d_{ji}^{2}. \quad (3)$$

where P is the total number of patterns in the training data set.

m is a weight exponent in the fuzzy membership. It can be any real number which is greater than 1.

μ_{ji} is the membership value of the i^{th} data in the j^{th} cluster.

d_{ji} is the distance between the i^{th} data and the center of the j^{th} cluster.

The detailed procedure of the fuzzy c-means algorithm is as follows:

A. Initialize the center values of all clusters.

B. Calculate μ_{ji} as follows:

$$\mu_{ji} = \left[\sum_{k=1}^{M} \left(d_{ji}/d_{ki}\right)^{2/(m-1)}\right]^{-1}. \quad (4)$$

C. Update the center value c_j by using the following formula:

$$c_j = \sum_{i=1}^{P} \mu_{ji}^{m} x_i \Big/ \sum_{i=1}^{P} \mu_{ji}^{m}. \quad (5)$$

where x_i is the i^{th} data in the series.

D. Calculate J_m using Equation 3.

E. If a convergence criterion is reached, stop the loop. Otherwise return to step B.

After the centers of the clusters are obtained, the width of the j^{th} cluster is calculated as follows:

$$\sigma_j = \min_{k=1}^{M} (\| c_j - c_k \|), \quad k \text{ is not in the same class as } j. \tag{6}$$

3.2 Phase 2: Optimize the Cluster Node

After the centers and the widths of the cluster nodes are initially determined in Phase 1, this phase employs the evolutionary algorithm to optimize those clusters and their parameters (c_j and σ_j) so that the prediction error of the system is minimal. A description of the process used in this phase is given below.

A. Define the size (S) of the initial population. Encode the initial parameters of all clusters in the cluster layer into a starting chromosome. A gene in the chromosome represents the center and the width of each cluster. For each cluster node j, create a list of input patterns whose membership value μ_{ji} are in the top S-1. Then repeat the following steps S-1 times:
   A.1. Randomly select an input pattern from each list, and assign it to be the center of the cluster. Then calculate the width of the cluster by using equation 6.
   A.2. Encode the centers and the widths of all clusters obtained from step A.1 into a new chromosome.
B. Combine the newly generated S-1 chromosomes with the starting chromosome to form the initial population.
C. Evaluate the fitness of each chromosome s in the population. The fitness function used in this research is as follows:

$$f(s) = \left[\sum_{q=1}^{S} (w_q + e_q) \right] - (w_s + e_s). \tag{7}$$

$$w_s = \sum_{i=1}^{P} A_i. \tag{8}$$

$$A_i = \begin{cases} 1, & \text{if } J \notin Y_i \\ 0, & \text{otherwise} \end{cases} \tag{9}$$

$$e_s = \sum_{i=1}^{P} \frac{T_b(X_i)}{T_g(X_i)} \Big/ \sum_{q=1}^{S} \sum_{i=1}^{P} \frac{T_b(X_i)}{T_g(X_i)}. \tag{10}$$

$$T_g(X_i) = \max_{\forall j, j \in Y_i} (T_j(X_i)). \tag{11}$$

$$T_b(X_i) = \max_{\forall j, j \notin Y_i} (T_j(X_i)). \tag{12}$$

where S is the total number of chromosomes in the population.
J is the winning cluster node.
Y_i is the target output of the i^{th} input pattern.

If the highest fitness value of the current population is equal to that of the last generation, increase the size (S) of the current population by 2 chromosomes. As long as there is no improvement in the fitness value of the current population, the size of the current population is continuously increased by 2 chromosomes. On the contrary, when the improvement is significant enough, the size of the current population will be reduced back to the initial stage.

D. Create a new population by repeating the following steps S/2 times:
   D.1. Select two parent chromosomes from a current population using the roulette wheel technique.
   D.2. With a crossover probability, cross over the parent chromosomes to form the new offspring.
   D.3. Randomly pick a number from a set of {1, 2, 3}. If the picked number is 1, perform the mutation on the new offspring as follows:
      - Randomly select mutation positions.
      - With a mutation probability, mutate the selected positions by adding or subtracting a small random number to/from the original value of the selected positions.

      If the picked number is 2, add a new gene to the end of the new offspring. However, if the picked number is 3, delete a randomly-selected gene from the new offspring.
   D.4. Place the new offspring into a new population.
E. Combine a new population with the current population. Then select S chromosomes from the combined list according to their fitness to form the next generation.
F. If a predetermined number of iteration is reached or the end condition is satisfied, stop the loop and return the best chromosome in the current population. The genes of the best chromosome are then used as the parameters of the nodes in the cluster layer. If not, go to step C.

4 Experimental Results

To test the performance of the proposed approach, the experiments have been conducted on 6 benchmark data sets: the iris data [3], the vowel recognition problem [4], the Pima Indians diabetes database [5], the ionosphere data [5], the BUPA liver disorders data [5], and the heart disease problem [6].

Results of the experiments are shown in Table 1. For the iris data (data set 1), all three neural network models -- the fuzzy ARTMAP, the backpropagation neural network (BPNN), and the proposed model – give a perfect accuracy (100%). However, it is the opposite way round for the data set 2, where both decision trees outpace all three neural network models. For the data set 3, the proposed model comes out to be the best among the compared methods, while the performance of the rest is about the same. For the data set 4, there is no significant difference in the performance of the three neural network methods. For the data set 5, the best prediction performance (71.68%) is obtained from the fuzzy ARTMAP neural network using the vigilance parameter (ρ_a) = 0.5 and the learning rate (β) = 0.7. However, the proposed model is not far behind the performance of the fuzzy

ARTMAP (50 versus 49 misclassifications). For the data set 6, the proposed model, which yields an accuracy of 85.56%, outperforms other methods by a wide margin.

Table 1. Experimental results

Data Set	% Correct				
	C4.5	CART	Fuzzy ARTMAP	BPNN	Proposed Model
1	95.0 [7]	93.0 [8]	100	100	100
2	80.0 [7]	78.2 [9]	56.71	55.19	66.23
3	73.0 [9]	74.5 [9]	75.52	76.04	80.73
4	94.9 [9]	88.9 [9]	98.68	99.34	100
5	65.0 [7]	-	71.68	67.05	71.10
6	77.5 [10]	-	77.78	78.88	85.56

5 Conclusions

In this paper, the new evolutionary neural network, which applies the concepts of the fuzzy c-means algorithm and the evolutionary algorithm to the artificial neural network, is proposed and its performances are compared with two of the best decision trees and two of the best neural networks. The results of the proposed evolutionary neural network are the best among those of the compared methods.

References

1. Fayyad, U., Piatetsky-Shapiro, G., Smyth, P.: Knowledge Discovery and Data Mining: Towards a Unifying Framework. Proceedings of the Second International Conference on Knowledge Discovery and Data Mining. (1996)
2. Han, J., Kamber, M.: Data Mining: Concepts and Techniques. Academic Press. (2001)
3. Fisher, R. A.: The Use of Multiple Measurements in Taxonomic Problems. Annual Eugenics, Vol. 7, Part II. (1936) 179-188
4. Deterding, D. H.: Speaker Normalization for Automatic Speech Recognition. Ph.D. Dissertation. (1989)
5. Blake, C. L., Merz, C. J.: UCI Repository of Machine Learning Databases [http://www.ics.uci.edu/~mlearn/MLRepository.html]. University of California, Department of Information and Computer Science. (1998)
6. Statlog Project Datasets: Retrieved March 28, 2003. From http://www.liacc.up.pt/ML/statlog/datasets/heart/
7. Ventura, D., Martinez, T. R.: An Empirical Comparison of Discretization Methods. Proceedings of the Tenth International Symposium on Computer and Information Sciences. (1995) 443-450
8. Tschichold-Gürman, N.: Fuzzy RuleNet: An Artificial Neural Network Model for Fuzzy Classification. Proceedings of the 1994 ACM Symposium on Applied Computing. (1994) 145-149
9. Datasets used for Classification: Comparison of Results: Retrieved March 5, 2005. From http://www.phys.uni.torun.pl/kmk/projects/datasets.html
10. Last, M., Maimon, O.: A Compact and Accurate Model for Classification. IEEE Transactions on Knowledge and Data Engineering, Vol. 16, No. 2. (2004) 203-215

ประวัติผู้เขียน

ชื่อ-นามสกุล	นายอัศวิน มีเงิน
วัน เดือน ปีเกิด	25 กุมภาพันธ์ 2522 ที่เชียงใหม่
ที่อยู่	270 หมู่ 4 ตำบล เวียง อำเภอ ฝาง จังหวัด เชียงใหม่ 50110
ประวัติการศึกษา	2544 ครุศาสตรอุตสาหกรรมบัณฑิต สาขาอิเล็กทรอนิกส์และคอมพิวเตอร์ (เกียรตินิยมอันดับ2) สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ความชำนาญเฉพาะด้าน	1.) ออกแบบระบบไมโครโปรเซสเซอร์และไมโครคอนโทรเลอร์ 2.) ออกแบบวงจรดิจิตอล