การประยุกต์นิวรอลเน็ตเวิร์กหลายโครงข่ายบนข้อมูลขนาดใหญ่

นายกฤศณัฏฐ์ บุญเกียรติพงษ์

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต
สาขาวิชาวิทยาศาสตร์คอมพิวเตอร์ ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2554

ลิขสิทธิ์ของจุฬาลงกรณ์มหาวิทยาลัย
บทคัดย่อและแฟ้มข้อมูลฉบับเต็มของวิทยานิพนธ์ตั้งแต่ปีการศึกษา 2554 ที่ให้บริการในคลังปัญญาจุฬาฯ (CUIR)
เป็นแฟ้มข้อมูลของนิสิตเจ้าของวิทยานิพนธ์ที่ส่งผ่านทางบัณฑิตวิทยาลัย
The abstract and full text of theses from the academic year 2011 in Chulalongkorn University Intellectual Repository(CUIR)
are the thesis authors' files submitted through the Graduate School.

APPLYING MULTIPLE NEURAL NETWORKS ON LARGE SCALE DATA

Mr. Kritsanatt Boonkiatpong

A Thesis Submitted in Partial Fulfillment of the Requirements

for the Degree of Master of Science Program in Computer Science

Department of Computer Engineering

Faculty of Engineering

Chulalongkorn University

Academic Year 2011

Copyright of Chulalongkorn University

หัวข้อวิทยานิพนธ์	การประยุกต์นิวรอลเน็ตเวิร์กหลายโครงข่ายบนข้อมูลขนาดใหญ่
โดย	นายกฤศณัฏฐ์ บุญเกียรติพงษ์
สาขาวิชา	วิทยาศาสตร์คอมพิวเตอร์
อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก	ผู้ช่วยศาสตราจารย์ ดร.สุกรี สินธุภิญโญ

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย อนุมัติให้นับวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญามหาบัณฑิต

…….. คณบดีคณะวิศวกรรมศาสตร์
(รองศาสตราจารย์ ดร.บุญสม เลิศหิรัญวงศ์)

คณะกรรมการสอบวิทยานิพนธ์

……ประธานกรรมการ
(ศาสตราจารย์ ดร.บุญเสริม กิจศิริกุล)

…….. อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก
(ผู้ช่วยศาสตราจารย์ ดร.สุกรี สินธุภิญโญ)

……กรรมการ
(รองศาสตราจารย์ ดร.ญาใจ ลิ่มปิยะกรณ์)

……กรรมการภายนอกมหาวิทยาลัย
(ดร.เด่นดวง ประดับสุวรรณ)

กฤศณัฏฐ์ บุญเกียรติพงษ์ : การประยุกต์นิวรอลเน็ตเวิร์กหลายโครงข่ายบนข้อมูลขนาดใหญ่. (APPLYING MULTIPLE NEURAL NETWORKS ON LARGE SCALE DATA) อ. ที่ปรึกษาวิทยานิพนธ์หลัก : ผศ.ดร.สุกรี สินธุภิญโญ, 60 หน้า.

การเรียนรู้และวิเคราะห์ข้อมูลขนาดใหญ่ด้วยนิวรอลเน็ตเวิร์กเป็นเรื่องที่สำคัญมากในการทำเหมืองข้อมูล แต่ก็มักจะเกิดปัญหาในด้านเวลาที่ใช้ในการเรียนรู้ ในงานวิจัยนี้ได้เสนอวิธีการเรียนรู้โดยการใช้นิวรอลเน็ตเวิร์กหลายโครงข่าย ทำการเรียนรู้บนชุดข้อมูลตัวอย่างที่ถูกแบ่งย่อยและสุ่มเลือกมาจากชุดตัวอย่างทั้งหมด จากนั้นจึงทำการรวมโหนดในชั้นแฝงจากเน็ตเวิร์กแต่ละอัน เพื่อหาค่าน้ำหนักประจำโหนดในชั้นแฝงใหม่ที่เหมาะสมสำหรับข้อมูลทั้งชุด ผู้วิจัยได้ทำการพัฒนาอัลกอริทึมการรวมโหนดโดยประยุกต์จากการหาระยะยุคลิดเพื่อระบุความใกล้เคียงกันของโหนด เพื่อที่จะรวมค่าน้ำหนักของโหนดที่ใกล้เคียงเข้าไว้ด้วยกัน ผลการทดลองพบว่า วิธีการที่นำเสนอสามารถลดเวลาในการเรียนรู้ลงได้อย่างมากและยังคงรักษาเปอร์เซ็นต์ความถูกต้องได้เหมือนกับการใช้เซตตัวอย่างทั้งหมด

ภาควิชา วิศวกรรมคอมพิวเตอร์ ลายมือชื่อนิสิต .....
สาขาวิชา วิทยาศาสตร์คอมพิวเตอร์ ลายมือชื่อ อ.ที่ปรึกษาวิทยานิพนธ์หลัก .....
ปีการศึกษา 2554

5171452121 : MAJOR COMPUTER SCIENCE
KEYWORDS : NUERAL NETWORK / LARGE SCALE DATASET / EUCLIDEAN DISTANCE

KRITSANATT BOONKIATPONG : APPLYING MULTIPLE NEURAL NETWORKS ON LARGE SCALE DATA. ADVISOR : ASST.PROF. SUKREE SINTHUPINYO, Ph.D, 60 pp.

Learning and analysis on large scale data sets is very important in data mining. Large amount of data can be a cause of problem in learning time and also in learning capability. This research proposed a novel method to solve that problem by using multiple neural networks to learn from multiple sub datasets that extracted equally from the whole dataset. We employ algorithm by applying Euclidean Distance to integrate weight of hidden nodes from each network.The experimental results show that reducing the learning time for the proposed method and also preserve the accuracy.

Department : Computer Engineering Student's Signature .....
Field of Study : Computer Science Advisor's Signature .....
Academic Year : 2011

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงไปได้ด้วยดี ผู้วิจัยขอกราบขอบพระคุณ ผู้ช่วยศาสตราจารย์ ดร. สุกรี สินธุภิญโญ อาจารย์ที่ปรึกษาวิทยานิพนธ์ ที่ได้ให้ความอนุเคราะห์ สละเวลาสอนความรู้ ให้คำปรึกษา ช่วยตรวจสอบ ให้คำแนะนำแนวทางการวิจัย รวมไปถึงสนับสนุนในเรื่องต่างๆ จนทำให้การวิจัยครั้งนี้สำเร็จออกมาได้ด้วยดี

ขอกราบขอบพระคุณ ศาสตราจารย์ ดร.บุญเสริม กิจศิริกุล รองศาสตราจารย์ ดร.ญาใจ ลิ่มปิยะกรณ์ และ อาจารย์ ดร.เด่นดวง ประดับสุวรรณ กรรมการสอบวิทยานิพนธ์ ที่กรุณาเสียสละเวลา ให้คำแนะนำ ตรวจสอบ และแก้ไขวิทยานิพนธ์ฉบับนี้

ขอกราบขอบพระคุณคุณแม่และครอบครัว ที่ให้การสนับสนุนและเป็นกำลังใจให้เสมอมา

ผู้วิจัยขอขอบคุณเพื่อนร่วมงานทุกคน และผู้บังคับบัญชาในสายงาน ที่คอยติดตาม ให้กำลังใจและสนับสนุน มีส่วนช่วยให้วิทยานิพนธ์สำเร็จได้ด้วยดี

ผู้วิจัยขอขอบคุณเพื่อนๆ ร่วมรุ่น CT19 ทุกคน ที่คอยให้กำลังใจ และให้ความช่วยเหลือในทุกเรื่อง ทุกรูปแบบ เพื่ออำนวยความสะดวกในระหว่างการทำวิจัย

ท้ายที่สุด ผู้วิจัยขอขอบพระคุณเพื่อนๆ ทุกคน ที่คอยติดตามและให้กำลังใจ รวมถึงท่านอื่นๆ ที่มิได้กล่าวชื่อไว้ ณ ที่นี้ที่มีส่วนทำให้วิทยานิพนธ์สำเร็จลุล่วงไปได้ด้วยดี

สารบัญ

หน้า

บทคัดย่อภาษาไทย ..... ง

บทคัดย่อภาษาอังกฤษ ..... จ

กิตติกรรมประกาศ ..... ฉ

สารบัญ ..... ช

สารบัญตาราง ..... ฌ

สารบัญภาพ ..... ฎ

บทที่ 1 บทนำ 1

1.1 ความเป็นมาและความสำคัญของปัญหา 1

1.2 วัตถุประสงค์ของการวิจัย 2

1.3 ขอบเขตของการวิจัย 2

1.4 ประโยชน์ที่คาดว่าจะได้รับ 3

1.5 วิธีดำเนินการวิจัย 3

1.6 ลำดับการจัดเรียงเนื้อหาในวิทยานิพนธ์ 3

1.7 ผลงานที่ตีพิมพ์จากวิทยานิพนธ์ 4

บทที่ 2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง 5

2.1 ทฤษฎีที่เกี่ยวข้อง 5

2.1.1 นิวรอลเน็ตเวิร์กแบบแบ็กพรอพาเกชัน (Backpropagation Neural Network) 5

2.1.2 ระยะยุคลิด (Euclidean Distance) 6

2.2 งานวิจัยที่เกี่ยวข้อง 7

2.2.1 งานวิจัยเกี่ยวกับนิวรอลเน็ตเวิร์กและข้อมูลขนาดใหญ่ 7

2.2.2 งานวิจัยเกี่ยวกับการปรับปรุงประสิทธิภาพนิวรอลเน็ตเวิร์ก 11

บทที่ 3 วิธีดำเนินงานวิจัย 15

3.1 การเตรียมข้อมูลสำหรับการเรียนรู้ 17

3.1.1 ชุดข้อมูลทดสอบที่ 1 Iris 22

3.1.2 ชุดข้อมูลทดสอบที่ 2 Pima Diabetes 22

3.1.3 ชุดข้อมูลทดสอบที่ 3 ข้อมูลสร้างชุดที่ 1 23

หน้า

3.1.4 ชุดข้อมูลทดสอบที่ 4 ข้อมูลสร้างชุดที่ 2 23
3.2 การเรียนรู้ของเน็ตเวิร์กย่อย 24
3.3 การเปรียบเทียบค่าน้ำหนักและการรวมโหนด 25
บทที่ 4 การทดลองและผลการทดลอง 28
4.1 การทดลอง 28
4.1.1 เครื่องมือที่ใช้ในการวิจัย 28
4.1.2 การเรียนรู้ของเน็ตเวิร์กย่อยเพื่อหาเน็ตเวิร์กที่ดีที่สุด 28
4.1.3 การรวมโหนดในชั้นแฝง-ชั้นเอาต์พุต 31
4.1.4 การรวมโหนดในชั้นอินพุต-ชั้นแฝง 32
4.2 ผลการทดลอง 36
4.2.1 เปรียบเทียบความผิดพลาดในการเรียนรู้กับเน็ตเวิร์กตั้งต้น 36
4.2.2 เปรียบเทียบความถูกต้องในการเรียนรู้กับเน็ตเวิร์กตั้งต้น 36
4.2.3 เปรียบเทียบเวลาที่ใช้ในการเรียนรู้กับเน็ตเวิร์กตั้งต้น 37
4.2.3 เปรียบเทียบระดับนัยสำคัญ 37
บทที่ 5 สรุปผลการวิจัยและข้อเสนอแนะ 39
5.1 สรุปผลการวิจัย 39
5.2 ข้อเสนอแนะสำหรับการทำวิจัยต่อ 40
รายการอ้างอิง 41
ภาคผนวก 44
ภาคผนวก ก การเตรียมชุดข้อมูลจากโปรแกรมเวกา 45
ภาคผนวก ข แอสคิวแอลในการแบ่งข้อมูลตรวจสอบและการส่งออก 47
ภาคผนวก ค ค่าน้ำหนักของเน็ตเวิร์กย่อย 52
ภาคผนวก ง การเปรียบเทียบระยะยุคลิดในชั้นแฝง 56
ประวัติผู้เขียนวิทยานิพนธ์ 60

สารบัญตาราง

หน้า

ตารางที่ 1 ค่าเฉลี่ยของความถูกต้องในการแยกแยะของเน็ตเวิร์ก 11
ตารางที่ 2 การเปรียบเทียบประสิทธิภาพของแต่ละอัลกอริทึม 13
ตารางที่ 3 รายละเอียดของข้อมูลทดสอบชุดที่ 1 22
ตารางที่ 4 รายละเอียดของข้อมูลทดสอบชุดที่ 2 22
ตารางที่ 5 รายละเอียดของข้อมูลทดสอบชุดที่ 3 23
ตารางที่ 6 รายละเอียดของข้อมูลทดสอบชุดที่ 4 23
ตารางที่ 7 รายละเอียดเน็ตเวิร์กแต่ละชุดข้อมูลที่ถูกกำหนดในโปรแกรมเอ็มบีพี 24
ตารางที่ 8 ค่าความผิดพลาดรากกำลังสองเฉลี่ยชุดข้อมูลที่ 1 Iris 29
ตารางที่ 9 ค่าความผิดพลาดรากกำลังสองเฉลี่ยชุดข้อมูลที่ 2 Pima Diabetes 29
ตารางที่ 10 ค่าความผิดพลาดรากกำลังสองเฉลี่ยชุดข้อมูลที่ 3 Data Generated 1 30
ตารางที่ 11 ค่าความผิดพลาดรากกำลังสองเฉลี่ยชุดข้อมูลที่ 4 Data Generated 2 30
ตารางที่ 12 ค่าค่าน้ำหนักในชั้นแฝง-ชั้นเอาต์พุตหลังทำการรวมโหนด 31
ตารางที่ 13 ค่าน้ำหนักในชั้นอินพุต-ชั้นแฝงหลังการรวมโหนดชุดข้อมูล Iris 32
ตารางที่ 14 ค่าน้ำหนักในชั้นอินพุต-ชั้นแฝงหลังการรวมโหนดชุดข้อมูล Pima Diabetes 33
ตารางที่ 15 ค่าน้ำหนักในชั้นอินพุต-ชั้นแฝงหลังการรวมโหนดชุดข้อมูล Data Generated 1 34
ตารางที่ 16 ค่าน้ำหนักในชั้นอินพุต-ชั้นแฝงหลังการรวมโหนดชุดข้อมูล Data Generated 2 35
ตารางที่ 17 การเปรียบเทียบความผิดพลาดการเรียนรู้ 36
ตารางที่ 18 การเปรียบเทียบความถูกต้องในการเรียนรู้ 36
ตารางที่ 19 การเปรียบเทียบเวลาที่ใช้ในการเรียนรู้ 37
ตารางที่ 20 เปอร์เซ็นความถูกต้องของข้อมูลทดสอบ 37
ตารางที่ 21 ค่าจำเพาะในการสร้างข้อมูลทดสอบ 3 และ 4 46
ตารางที่ 22 เปรียบเทียบระยะยุคลิดในชั้นแฝงของชุดข้อมูล Iris 56
ตารางที่ 23 เปรียบเทียบระยะยุคลิดในชั้นแฝงของชุดข้อมูล Pima Diabetes 57
ตารางที่ 24 เปรียบเทียบระยะยุคลิดในชั้นแฝงของชุดข้อมูล Data Generated 1 58
ตารางที่ 25 เปรียบเทียบระยะยุคลิดในชั้นแฝงของชุดข้อมูล Data Generated 2 59

สารบัญภาพ

หน้า

ภาพที่ 1 แบบจำลองของนิวรอลเน็ตเวิร์กแบบแบ็กพรอพาเกชัน 5
ภาพที่ 2 แนวคิดในงานวิจัยในการรวมนิวรอลเน็ตเวิร์กชั้นแฝงเดี่ยว 15
ภาพที่ 3 แบบจำลองของนิวรอลเน็ตเวิร์กที่ใช้ในงานวิจัย 17
ภาพที่ 4 แบบจำลองการเตรียมข้อมูลสำหรับการฝึกสอน, การทดสอบ, และการตรวจสอบ 18
ภาพที่ 5 การแบ่งข้อมูลสำหรับการทดสอบแบบ 10-fold Cross Validation 19
ภาพที่ 6 ขั้นตอนและกระบวนการในการทำการวิจัย 21
ภาพที่ 7 ขั้นตอนการทดลองและอัลกอริทึมการรวมโหนด 27
ภาพที่ 8 ผลการทดสอบสมมุติฐานด้วยวิธีการทดสอบความแตกต่างแบบจับคู่ 38
ภาพที่ 9 โปรแกรมเวกาเอ็กพลอเรอร์สำหรับเตรียมข้อมูลทดสอบ 45
ภาพที่ 10 ฟังก์ชันสร้างข้อมูลทดสอบในโปรแกรมเวกาเอ็กพลอเรอร์ 45
ภาพที่ 11 การกำหนดค่าจำเพาะในฟังก์ชันสร้างข้อมูลทดสอบ 46
ภาพที่ 12 เอสคิวแอลในการดึงข้อมูลเพื่อทำเป็นข้อมูลตรวจสอบ 47
ภาพที่ 13 ผลลัพธ์ตัวอย่างของข้อมูลตรวจสอบ 48
ภาพที่ 14 การส่งข้อมูลออกจากฐานข้อมูลเอสคิวแอล 49
ภาพที่ 15 เลือกแหล่งข้อมูลเอสคิวแอลที่ต้องส่งออก 49
ภาพที่ 16 เลือกไฟล์ CSV ที่ต้องการส่งข้อมูลออก 50
ภาพที่ 17 เลือกชนิดการส่งออกเป็นแบบคัดลอกจากตารางข้อมูล 50
ภาพที่ 18 เลือกตารางข้อมูลที่ต้องการส่งออกไปเป็นไฟล์ CSV 51
ภาพที่ 19 ผลลัพธ์ของการส่งออกข้อมูลจากเอสคิวแอล 51
ภาพที่ 20 ค่าน้ำหนักของเน็ตเวิร์กตั้งต้น N_0 ของ Iris 52
ภาพที่ 21 ค่าน้ำหนักของเน็ตเวิร์กคัดเลือก N_{best} ของ Iris 52
ภาพที่ 22 ค่าน้ำหนักของเน็ตเวิร์กที่รวมโหนด N_{best} ของ Iris 53
ภาพที่ 23 ค่าน้ำหนักของเน็ตเวิร์กตั้งต้น N_0 ของ Pima Diabetes 53
ภาพที่ 24 ค่าน้ำหนักของเน็ตเวิร์กคัดเลือก N_{best} ของ Pima Diabetes 53
ภาพที่ 25 ค่าน้ำหนักของเน็ตเวิร์กที่รวมโหนด N_{best} ของ Pima Diabetes 54
ภาพที่ 26 ค่าน้ำหนักของเน็ตเวิร์กตั้งต้น N_0 ของ Data Generated 1 54

หน้า

ภาพที่ 27 ค่าน้ำหนักในเน็ตเวิร์กคัดเลือก N_{best} ของ Data Generated 1..... 54
ภาพที่ 28 ค่าน้ำหนักของเน็ตเวิร์กที่รวมโหนด N_{best} ของ Data Generated 1 54
ภาพที่ 29 ค่าน้ำหนักของเน็ตเวิร์กตั้งต้น N_0 ของ Data Generated 2 55
ภาพที่ 30 ค่าน้ำหนักของเน็ตเวิร์กคัดเลือก N_{best} ของ Data Generated 2..... 55
ภาพที่ 31 ค่าน้ำหนักของเน็ตเวิร์กที่รวมโหนด N_{best} ของ Data Generated 2 55

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ปัจจุบันการศึกษาและงานวิจัยทางด้านนิวรอลเน็ตเวิร์กแพร่หลายไปอย่างกว้างขวาง ซึ่งจะแบ่งการวิจัยออกไปในหลายแขนงด้วยกัน [1] มีทั้งการศึกษาทางด้านโครงสร้างและแบบจำลองของนิวรอลเน็ตเวิร์ก, การออกแบบนิวรอลเน็ตเวิร์ก, การประยุกต์เพื่อใช้งานได้กว้างขวางกว่าเดิมหรือแม้แต่ที่เกี่ยวกับการปรับปรุงประสิทธิภาพในการเรียนรู้เพื่อให้เน็ตเวิร์กมีความสามารถเรียนรู้ได้เร็ว และคัดแยกข้อมูลได้ถูกต้องมากขึ้น นิวรอลเน็ตเวิร์กได้ถูกนำไปประยุกต์ใช้ในงานมากมาย เช่นการพยากรณ์สภาพภูมิอากาศ การจราจร การทำนายหุ้น หรือไม่ว่าจะเป็นการรู้จำในเรื่องต่างๆเช่น เสียง ภาพ เพื่อให้นิวรอลเน็ตเวิร์กทำงานได้คล้ายกับสมองของมนุษย์ให้มากที่สุด ในปัจจุบันสาขาวิชาปัญญาประดิษฐ์ได้มีบทบาทสำคัญเป็นอย่างมากในการประยุกต์ใช้ในอุตสาหกรรมและยากที่จะหาข้อสิ้นสุดได้ เช่นเดียวกับการทำวิจัยในเรื่องปัญญาประดิษฐ์ การเรียนรู้ของเครื่อง และนิวรอลเน็ตเวิร์กก็ยังมีอีกหลากหลายแง่มุมให้ศึกษาและประยุกต์ใช้โดยเฉพาะเพื่อที่จะให้นิวรอลเน็ตเวิร์กประมวลผลได้มีประสิทธิภาพมากขึ้น ประมวลผลได้เร็วขึ้นและความถูกต้องมากขึ้น

ในการทำเหมืองข้อมูลก็เป็นอีกสาขาหนึ่งที่ได้นำเอานิวรอลเน็ตเวิร์กไปประยุกต์ใช้เพื่อให้ได้ข้อมูลที่ถูกต้องและแม่นยำ [2] แต่เนื่องด้วยเหมืองข้อมูลต้องใช้ข้อมูลขนาดใหญ่จำนวนมากในการวิเคราะห์ ทำให้มีปัญหาเกี่ยวกับเรื่องประสิทธิภาพในการวิเคราะห์ข้อมูล รวมไปถึงการใช้นิวรอลเน็ตเวิร์กในการวิเคราะห์เหมืองข้อมูลด้วยซึ่งจะพบปัญหาในการเรียนรู้ไม่ว่าจะเป็นเรื่องความถูกต้องและเวลาที่ใช้ในการวิเคราะห์ซึ่งยังใช้เวลามากในการวิเคราะห์

งานวิจัยนี้จะให้ความสำคัญในการเสนอเทคนิคในการวิเคราะห์ข้อมูลขนาดใหญ่โดยใช้นิวรอลเน็ตเวิร์กในการเรียนรู้ชุดข้อมูลขนาดใหญ่ โดยจากงานวิจัยที่เกี่ยวข้องจะแสดงให้เห็นว่าจำนวนนิวรอนที่อยู่ในชั้นแฝงและจำนวนชั้นของเน็ตเวิร์กในชั้นแฝงจะมีผลต่อประสิทธิภาพ [3] เนื่องด้วยจำนวนชั้นในชั้นแฝงมาก จะทำให้การประมวลผลช้าลงแต่ความถูกต้องจะมากขึ้น ซึ่งการวิจัยนี้ได้เน้นความสนใจในระดับโครงสร้างของนิวรอลเน็ตเวิร์กโดยจะเสนอวิธีการเรียนรู้และปรับปรุงเน็ตเวิร์กให้มีความสามารถในการเรียนรู้ข้อมูลขนาดใหญ่ได้ โดยที่ความถูกต้องไม่แตกต่างจากวิธีการเรียนรู้แบบปกติ แต่เวลาที่ใช้ในการเรียนรู้จะลดลงมาก ซึ่งจะเป็นประโยชน์มากในการนำไปประยุกต์ในการทำเหมืองข้อมูล เพราะโดยปกติการเรียนรู้ข้อมูลขนาดใหญ่ในเหมืองข้อมูลนั้น ทำได้ยากที่จะเรียนรู้ได้ในครั้งเดียว ซึ่งจำเป็นต้องใช้ทั้งทรัพยากรและเวลาในการ

ประมวลผลจำนวนมาก งานวิจัยนี้ยังได้นำเสนอแนวคิดการแบ่งข้อมูลออกเท่าๆกันเป็นชุดข้อมูลย่อยเพื่อที่จะทำการเรียนรู้เซตข้อมูลใหญ่ได้โดยใช้ชั้นแฝงแค่ชั้นเดียว การปรับปรุงความถูกต้องจะใช้เทคนิคการหาค่าน้ำหนักที่เหมาะสมของแต่ละโหนดนิวรอนที่ให้ค่าความผิดพลาดที่น้อยที่สุด แล้วนำเอาค่าน้ำหนักที่ดีสุดของเซตข้อมูลย่อยมาสร้างเน็ตเวิร์กใหม่โดยได้จากการเปรียบเทียบและรวมโหนดจากเน็ตเวิร์กย่อยอื่นๆ และเมื่อผ่านการปรับค่าน้ำหนักที่เหมาะสมแล้ว จะสามารถทำให้นิวรอลเน็ตเวิร์กจะสามารถเรียนรู้เซตข้อมูลขนาดใหญ่ได้โดยที่ความถูกต้องใกล้เคียงกับค่าเดิม โดยไม่จำเป็นต้องเรียนรู้กับเซตข้อมูลขนาดใหญ่ในครั้งเดียว

1.2 วัตถุประสงค์ของการวิจัย

งานวิจัยนี้มีวัตถุประสงค์เพื่อนำเสนอเทคนิคใช้การวิเคราะห์นิวรอลเน็ตเวิร์กเพื่อนำไปประยุกต์กับเซตข้อมูลขนาดใหญ่ โดยการพัฒนาอัลกอริทึมในการวิเคราะห์ข้อมูลเซตย่อยแต่ละเซต เพื่อให้ได้นิวรอลเน็ตเวิร์กที่เหมาะสมที่สุดที่สามารถเรียนรู้แต่ละเซตย่อยได้ โดยให้ผลความผิดพลาดที่ต่ำกว่าหรือใกล้เคียงการเรียนรู้แบบเดิม แต่เวลาในการเรียนรู้ลดลงจากการเรียนรู้แบบเดิม

1.3 ขอบเขตของการวิจัย

1. การเตรียมชุดข้อมูลฝึกฝน ใช้ข้อมูลจากข้อมูลยูซีไอ (UCI Repository) [4] โดยได้ใช้ชุดข้อมูล Iris (Iris) และ Pima Diabetes (Pima Indians Diabetes) โดยเป็นชุดข้อมูลชนิดตัวเลข รูปแบบเป็นแบบคัดแยก (Classification) และข้อมูลที่สร้างจากโปรแกรมเวกา (WEKA) [5]
2. ในการฝึกฝนข้อมูล จะใช้นิวรอลเน็ตเวิร์กแบบแบ็กพรอพาเกชัน (Backpropagation Algorithm) แบบจำนวนรอบแบบคงที่คือ 10,000 รอบ
3. นิวรอลเน็ตเวิร์กที่ใช้ในการเรียนรู้ให้กับชุดข้อมูลแต่ละชุด ไม่ว่าจะทั้งชุดข้อมูลหลักและชุดข้อมูลย่อย จะมีโครงสร้างที่เหมือนกัน คือเป็นนิวรอลเน็ตเวิร์กแบบมีชั้นแฝงเดี่ยว โดยมีชั้นอินพุต (Input Layer) จะมีจำนวนโหนดเท่ากับจำนวนของลักษณะประจำของข้อมูล (Attribute) และชั้นแฝง (Hidden Layer) จะมีจำนวนโหนดเท่ากับจำนวนคลาสที่มี และชั้นเอาต์พุต (Output Layer) จะมีจำนวนโหนด 1 โหนดเพื่อแสดงการคัดแยกข้อมูล
4. นิวรอลเน็ตเวิร์กที่ได้จากการปรับค่าน้ำหนักและรวมโหนด สามารถที่ทำการเรียนรู้ให้กับข้อมูลหลักเพื่อให้ค่าความถูกต้องได้ดีกว่าการเรียนรู้แบบปกติ

1.4 ประโยชน์ที่คาดว่าจะได้รับ

1. งานวิจัยนี้นำเสนอวิธีการใหม่ในการปรับปรุงประสิทธิภาพของนิวรอลเน็ตเวิร์ก โดยวิธีการปรับเปลี่ยนค่าน้ำหนักของเน็ตเวิร์กย่อยและรวมโหนดเข้าด้วยกัน เมื่อนำไปประยุกต์ใช้งานจริงจะสามารถทำให้ค่าน้ำหนักที่ได้นั้นเรียนรู้เซตข้อมูลขนาดใหญ่โดยที่ความถูกต้องใกล้เคียงหรือดีกว่าค่าเดิม
2. สามารถนำวิธีการใหม่นี้ไปใช้อ้างอิง เพื่อการนำไปประยุกต์ใช้กับการวิเคราะห์ข้อมูลขนาดใหญ่ในรูปแบบอื่นๆได้ เช่นการประยุกต์ในการทำเหมืองข้อมูล หรือการนำไปประยุกต์ใช้กับการเรียนรู้ของนิวรอลเน็ตเวิร์กในแบบอื่นที่ใช้วิธีการใกล้เคียงกัน

1.5 วิธีดำเนินการวิจัย

1. ค้นคว้า ศึกษา และวิจัยเรื่องนิวรอลเน็ตเวิร์กแบบแบ็กพรอพาเกชัน (Backpropagation Neural Network) และทฤษฎีที่เกี่ยวข้องในการทำเหมืองข้อมูลโดยใช้นิวรอลเน็ตเวิร์ก

2. ศึกษา ค้นคว้า และ สร้างแบบจำลองการทดลองและสร้างข้อมูลทดสอบจากโปรแกรมประยุกต์เวกา (WEKA) และโปรแกรมประยุกต์เอ็มบีพี Multiple Backpropagation (MBP) [6-7]

3. ออกแบบวิธีการแบ่งข้อมูลทดสอบออกเป็นชุดข้อมูลย่อย เพื่อที่จะทำการเรียนรู้ในเน็ตเวิร์กย่อยและทำการเรียนรู้ให้กับแต่ละชุดข้อมูลย่อยแล้วเก็บผลของค่าความผิดพลาดและค่าน้ำหนักไว้

4. วิเคราะห์ค่าน้ำหนักที่ได้จากเน็ตเวิร์กย่อย เพื่อที่จะสร้างอัลกอริทึมในการเปรียบเทียบเพื่อหาค่าน้ำหนักที่ดีที่สุดมาสร้างเน็ตเวิร์กใหม่ที่ได้จากการรวมโหนดและปรับแต่งค่าน้ำหนักในเน็ตเวิร์กใหม่โดยใช้อัลกอริทึมที่นำเสนอ

5. นำเอาเน็ตเวิร์กใหม่ที่ได้ ทดสอบเปรียบเทียบกับข้อมูลตั้งต้นเพื่อเปรียบเทียบ

6. สรุปผลการทดลอง

7. ตีพิมพ์บทความทางวิชาการ

8. เรียบเรียงวิทยานิพนธ์

1.6 ลำดับการจัดเรียงเนื้อหาในวิทยานิพนธ์

เนื้อหาในวิทยานิพนธ์ฉบับนี้แบ่งออกเป็น 5 บท ดังนี้คือ บทที่ 1 บทนำ บทที่ 2 ทฤษฎีและงานวิจัยที่เกี่ยวข้องกับการทำงานวิจัยฉบบับนี้ บทที่ 3 วิธีการดำเนินงานวิจัยโดยอธิบายถึงขั้นตอนต่างๆ อย่างละเอียด ในส่วนของบทที่ 4 เป็นการทดลองและผลที่ได้จากการทดลองและบทที่ 5

เป็นบทสรุปผลการทดลองและข้อเสนอแนะของงานวิจัยซึ่งจะเป็นประโยน์ต่องานวิจัยอื่นๆในอนาคต

1.7 ผลงานที่ตีพิมพ์จากวิทยานิพนธ์

ส่วนหนึ่งของวิทยานิพนธ์นี้ ได้รับการตีพิมพ์เป็นบทความทางวิชาการ ดังนี้

"Applying Multiple Neural Networks on Large Scale Data" โดย กฤศณัฏฐ์ บุญเกียรติพงษ์ และ สุกรี สินธุภิญโญ ในงานประชุมวิชาการระดับนานาชาติ "2011 International Conference on Information and Electronics Engineering (ICIEE 2011)" ซึ่งจัดขึ้นที่ กรุงเทพฯ ประเทศไทย ระหว่างวันที่ 28-29 พฤษภาคม พ.ศ. 2554

" Integration of Hidden Nodes to Learn Large Scale Data" โดย กฤศณัฏฐ์ บุญเกียรติพงษ์ และ สุกรี สินธุภิญโญ ในงานประชุมวิชาการระดับนานาชาติ "2012 The 8th National Conference on Computer and Information Technology (NCCIT 2012)" ซึ่งจัดขึ้นที่ ชลบุรี ประเทศไทย ระหว่างวันที่ 9-10 พฤษภาคม พ.ศ. 2555

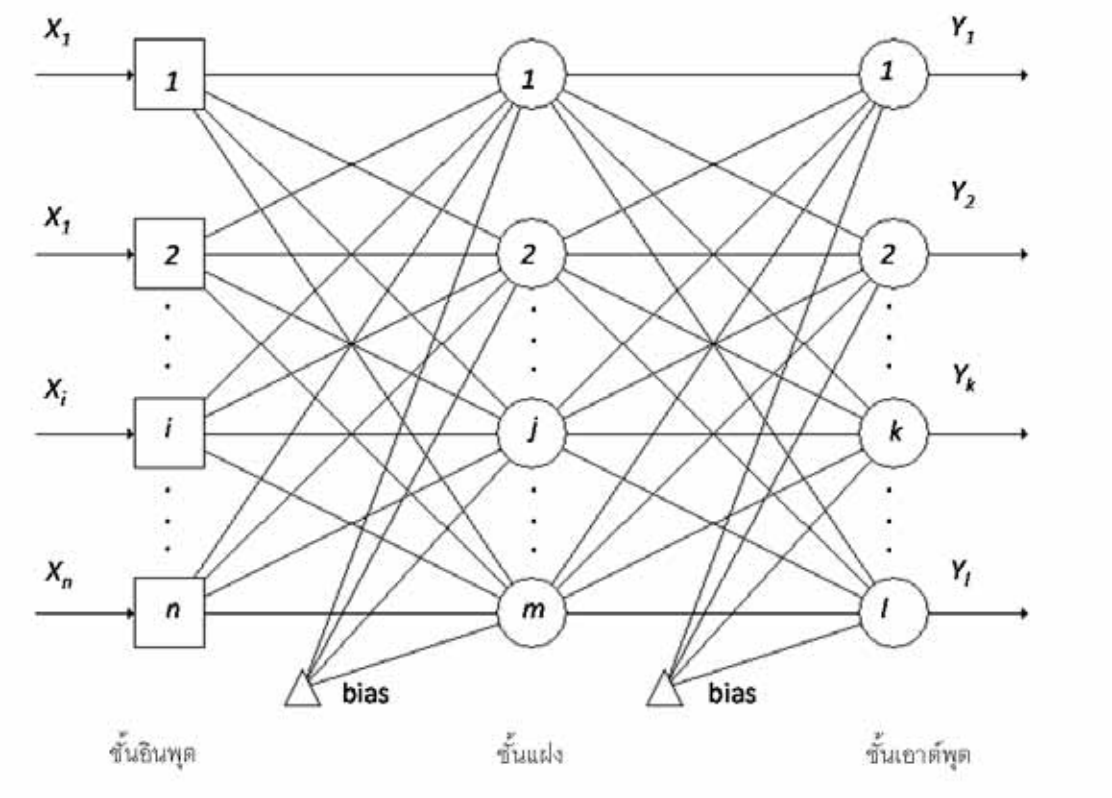
บทที่ 2
ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

2.1 ทฤษฎีที่เกี่ยวข้อง

จากการศึกษาค้นคว้าต่างๆ สามารถสรุปทฤษฎี และแนวคิดสำคัญที่เกี่ยวข้องกับงานวิจัยได้ดังนี้

2.1.1 นิวรอลเน็ตเวิร์กแบบแบ็กพรอพาเกชัน (Backpropagation Neural Network)

ในงานวิจัยนี้ ได้ใช้นิวรอลเน็ตเวิร์กแบบแบ็กพรอพาเกชัน (Backpropagation Neural Network - BPNN) [8] เป็นส่วนในการเรียนรู้ข้อมูล ซึ่งเป็นการเรียนรู้แบบมีผู้สอน (Supervised Learning) โดยนิวรอลเน็ตเวิร์กแบบแบ็กพรอพาเกชันจะมีโครงสร้างแบบง่ายๆและนิวรอนในชั้นต่างๆจะเชื่อมต่อเข้าหากันด้วยค่าน้ำหนัก (Weight) และเมื่อมีการส่งผ่านข้อมูลจากชุดข้อมูลที่ป้อนเข้ามาในนิวรอนอินพุต ค่าน้ำหนักที่ถูกกำหนดจะส่งต่อไปยังนิวรอนในชั้นถัดไปด้วยค่าน้ำหนักที่ต่างออกไป ดังภาพที่ 1



ภาพที่ 1 แบบจำลองของนิวรอลเน็ตเวิร์กแบบแบ็กพรอพาเกชัน

การคำนวณเอาต์พุตของเน็ตเวิร์กจะเกิดจากฟังก์ชันการส่งผ่านโดยนิวรอนในชั้นแฝงและคำนวณค่าเอาต์พุตออกมาโดยสมการที่ 1

$$Y_j(P) = sigmoid\left[\sum_{i-1}^{n} X_i(P) \times W_{ij}(P) - \theta_j\right] \qquad (1)$$

โดยให้ P คือจำนวนรูปแบบของการเรียนรู้ในชุดข้อมูล และ n คือจำนวนของอินพุตของ j นิวรอนในชั้นแฝง และ X_i คืออินพุตที่ i ที่ส่งผ่านมายังนิวรอน และ Y_j คือผลลัพธ์เอาต์พุต โดยค่า W_{ij} คือเป็นค่าน้ำหนักที่แต่ละนิวรอนใช้ θ_j คือค่าเทรสโชลด์ (Theshold) ของฟังก์ชันซิกมอยด์ (*sigmoid)* ที่สามารถแสดงได้ดังสมการที่ *2*

$$Y_{Sigmoid} = \frac{1}{1+e^{-x}} \qquad (2)$$

เมื่อมีการคำนวณหาค่าเอาต์พุตและส่งไปยังนิวรอนในชั้นเอาต์พุต ก็จะแสดงได้ในสมการที่ 3

$$Y_k(P) = \left[\sum_{j=1}^{m} X_{jk}(P) \times W_{jk}(P) - \theta_k\right] \qquad (3)$$

โดยค่า P คือค่ารูปแบบการเรียนรู้ (training pattern) ของชุดข้อมูลเรียนรู้, และ m คือจำนวนของอินพุตของนิวรอนที่ k ในชั้นเอาต์พุต, X_{jk} คือค่าอินพุตเข้าไปในนิวรอนและ Y_k คือค่าผลลัพธ์ของเอาต์พุต W_{jk} คือค่าน้ำหนัก, และ θ_k คือค่าเทรสโชลด์ของฟังก์ชันซิกมอยด์ ที่อ้างอิงในสมการที่ *2*

2.1.2 **ระยะยุคลิด** (Euclidean Distance)

ระยะทางแบบยุคลิด (Euclidean distance) [9] คือระยะทางระหว่างจุดสองจุดในแนวเส้นตรงที่สามารถวัดได้ โดยมาจากทฤษฎีบทพีทาโกรัส (Pythagorean Theorem) ซึ่งเหตุที่เรียกว่า ระยะแบบยุคลิด เพราะเป็นการวัดระยะทางบนปริภูมิแบบยุคลิด คือจะไม่มีความโค้งหรืองอ และอีกทั้งการใช้สูตรที่วัดระยะทางนั้นเป็นปริภูมิอิงระยะทาง ซึ่งจะเรียกได้ว่าเป็น ค่าประจำแบบยุคลิด (Euclidean norm)

ระยะยุคลิดของจุดสองจุด p และ q คือความยาวของส่วนของเส้นตรง pq และถ้าให้ p = (p_1, p_2, …, p_n) และ q = (q_1, q_2, …, q_n) เป็นจุดสองจุดใดๆบนปริภูมิยุคลิดที่มี n มิติ ระยะทางระหว่างจุด p กับ q นั้นจะคำนวณได้จาก

$$d(p,q) = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + \cdots + (p_n - q_n)^2} = \sqrt{\sum_{i=1}^{n}(p_i - q_i)^2} \quad (4)$$

โดยค่าประจำยุคลิดคือระยะทางจากจุดหนึ่งจุดใดๆใน p ไปยังจุดกำเนิดที่มีพิกัด (0,0,…,0) บนปริภูมิยุคลิด

$$\|p\| = \sqrt{p_1^2 + p_2^2 + \cdots + p_n^2} = \sqrt{p.p} \quad (5)$$

ซึ่งสมการที่ 5 นี้ เป็นผลคูณจุด (Dot Product) ซึ่งเป็นขนาดของเวกเตอร์ p จากจุดกำเนิด จะสามารถแสดงได้ว่าระยะยุคลิดจากจุดกำเนิดสามารถนิยามได้อีกแบบคือ

$$\|p - q\| = \sqrt{(p-q).(p-q)} = \sqrt{\|p\|^2 + \|q\|^2 - 2p.q} \quad (6)$$

ระยะยุคลิดในการวิจัยนี้ได้นำมาประยุกต์ในการหาความห่างและความคล้ายกันของเน็ตเวิร์ก เมื่อวัดระยะทางระหว่างสองวัตถุ ถ้าวัตถุห่างกันมากแสดงว่าวัตถุนั้นมีความคล้ายกันน้อย ถ้ามีค่าน้อยก็แสดงว่ามีความคล้ายคลึงกันมาก

2.2 งานวิจัยที่เกี่ยวข้อง

2.2.1 งานวิจัยเกี่ยวกับนิวรอลเน็ตเวิร์กและข้อมูลขนาดใหญ่

การวิจัยของนิวรอลเน็ตเวิร์กที่เกี่ยวกับการจัดการข้อมูลขนาดใหญ่นั้นยังมีแง่มุมในการวิจัยอย่างกว้างขวาง ซึ่งนักวิจัยพยายามที่จะนำเสนอแนวทางการเรียนรู้ข้อมูลขนาดใหญ่ให้มีประสิทธิภาพมากขึ้นและความถูกต้องในการแยกแยะและวิเคราะห์ข้อมูลก็เป็นอีกจุดที่นักวิจัยคำนึงถึง

ปี ค.ศ. 2000 อารอน เจ โอเวนส์ [10] จากดูปองต์ เซ็นทรัล รีเสิร์ซ แอนด์ ดีวีลอฟเมนต์ (DuPont Central Research and Development) สหรัฐอเมริกา ได้เสนอแนวทางการวิจัยเกี่ยวกับการสร้างแบบจำลองสำหรับการวิเคราะห์ข้อมูลขนาดใหญ่โดยใช้นิวรอลเน็ตเวิร์ก โดยนำเสนอว่า

การที่จะทำให้การเรียนรู้ของนิวรอลเน็ตเวิร์กบนข้อมูลขนาดใหญ่มีความรวดเร็วขึ้นจะต้องมีวิธีจัดการที่ประกอบด้วยวิธีการต่อไปนี้

- ลดความซ้ำซ้อนของโครงสร้างของนิวรอลเน็ตเวิร์กลง ขจัดส่วนที่เป็นคอขวดออกจากเน็ตเวิร์ก
- ทำการแบ่งข้อมูลออกเป็นส่วนย่อยหลายๆชุดเพื่อลดเวลาในการเรียนรู้
- จากนั้นก็นำเอาผลของเน็ตเวิร์กย่อยต่างๆมารวมเป็นผลการทำนายสุดท้ายที่แม่นยำและผิดพลาดน้อย

เพื่อที่จะลดความซับซ้อนของเน็ตเวิร์ก งานวิจัยได้เสนอให้ใช้นิวรอลเน็ตเวิร์กแบบชั้นแฝงเดี่ยวและเรียนรู้แบบแบ็กพรอพาเกชัน โดยได้ใช้แบบจำลอง PLS (Partial Least Squares) ในการประมวลผลเพื่อลดปัญหาคอขวดของนิวรอลเน็ตเวิร์กปกติ แต่ PLS เองก็มีข้อจำกัดโดยวิเคราะห์ได้เฉพาะที่เป็นเชิงเส้นเท่านั้น โดยการวิเคราะห์แบบไม่ใช่เชิงเส้นนั้น สัญญาณรบกวนยังมีค่าสูง การใช้แบบจำลอง PLS นั้น จะวิเคราะห์แบบจำลองเชิงเส้นแบ่งออกเป็นส่วนย่อยๆ และทำการวิเคราะห์ซ้ำในส่วนที่แบ่งย่อยนั้น ก็จะลดเวลาในการวิเคราะห์ลงได้มาก

การชักตัวอย่างย่อย (Sub-Sampling) ได้ทดสอบจากข้อมูล 10,000 เร็คคอร์ด โดยแบ่งเป็น 50 กลุ่มที่แตกต่างกัน จะเหลือข้อมูลต่อกลุ่มที่ 200 เร็คคอร์ด แบบจำลองจะถูกใช้กับทั้ง 50 กลุ่มข้อมูลนี้ซึ่งเวลาที่ใช้ในการเรียนรู้ของทั้ง 50 เน็ตเวิร์กนี้น้อยกว่าเวลาที่ใช้เรียนรู้ของชุดข้อมูลกลุ่มเดียว การรวมผลของการเรียนรู้ใช้การโหวตและการรวมน้ำหนักสัมพัทธ์ที่สามารถปรับค่าน้ำหนักโดยอัตโนมัติเพื่อให้ได้ค่าที่ดีที่สุด

ในปี ค.ศ.2005 ในโครงการแดม (DAME) ของมหาวิทยาลัยยอร์ก ประเทศอังกฤษ โบ เจียน เหลียง และ เจม ออสติน [11] นำเสนอวิธีการลดเวลาในการเรียนรู้ของข้อมูลขนาดใหญ่ที่ใช้ในอนุกรมเวลา (Time Series) โดยใช้เทคโนโลยีกริด (Grid Technology) โดยได้นำเสนออัลกอริทึม Fast Time Series Similarity เพื่อเข้ามาช่วยเพื่อลดเวลาการเรียนรู้ของข้อมูลขนาดใหญ่ที่ถูกเรียนรู้โดยอนุกรมเวลา โครงการแดมได้นำเอาอัลกอริทึมนี้ไปประยุกต์กับการออกแบบเครื่องยนต์อากาศพลศาสตร์ของโรลสรอยซ์ (Rolls-Royce)

โดยเสนอว่าอนุกรมเวลาที่มีความยาว n สามารถที่จะแสดงได้ในรูปแบบเวกเตอร์ n-มิติ และ ถ้าให้อนุกรมเวลา 2 ชุด $Q = \{q_1, q_2, \dots, q_n\}$ และ $S = \{s_1, s_2, \dots, s_n\}$ ความสัมพันธ์ของทั้ง 2 อนุกรมจะแสดงได้เป็น

$$Corr(Q,S) = \frac{\sum_{i=1}^{n} s_i q_i}{\sqrt{\sum_{i=1}^{n} s_i^2 \sum_{i=1}^{n} q_i^2}} \qquad (7)$$

$$= \frac{Q.S}{\|Q\|\|S\|} \qquad (8)$$

ความสัมพันธ์ของทั้งสองอนุกรมได้ถูกวัดโดยระยะยุคลิดตามสมการที่ 7 ให้เวกเตอร์ผลผลิต Q และ S และทั้ง 2 เวกเตอร์ ทำมุมโคไซน์ (Cosine) ระหว่างกัน ซึ่งระยะยุคลิดของทั้ง 2 เวกเตอร์จะหาได้จาก

$$D(Q,S) = \|Q - S\| \qquad (9)$$

และจากสมการที่ 8 และ 9 ระยะยุคลิดของทั้ง 2 เวกเตอร์จะเขียนใหม่ได้เป็น

$$D(Q,S) = \sqrt{\|Q\|^2 + \|S\|^2 + 2Q.S} \qquad (10)$$

หลักการหาระยะยุคลิดประยุกต์ในการวัดค่าเวลาในการค้นหา (Searching Time) เพื่อลดเวลาในการเรียนรู้ของเน็ตเวิร์กลง ซึ่งผลการทดสอบคือสามารถค้นหาข้อมูล 10GB ในเวลาที่น้อยกว่า 1 วินาที

และในปี ค.ศ. 2008 อีวานิโกวาส และคณะ [12] ได้นำเสนอการจัดการข้อมูลตัวอย่างขนาดใหญ่ด้วยอัลกอริทึม SAMANN (Sammon's Mapping Neural Network) ได้ทำการแบ่งจำนวนข้อมูลโดยการคัดแยกข้อมูลที่สนใจด้วยเคมีนส์ (k-means) และ เอสโอเอ็ม(SOM) และหลังจากนั้นทำการเรียนรู้ด้วยนิวรอลเน็ตเวิร์ก จากผลที่แสดงให้เห็นว่าค่าความผิดพลาดลดลงจากการเทียบกับการใช้ข้อมูลทั้งหมด และอีกทั้งเวลาในการเรียนรู้ก็ยังลดลงมากกว่า 10% อีกด้วย

ในปีเดียวกัน ยิ ชิ เซา, นู คู หว่อง และเซียง ลี [13] แห่งสถาบันเทคโนโลยีนันยาง สิงคโปร์ นำเสนอวิธีการที่เรียกว่าไฮบริดเวฟเลทนิวรอลเน็ตเวิร์ก (Hybrid Wavelet Neural Networks :HWNN) เป็นการนำเอาข้อดีของ เวฟเลทนิวรอลเน็ตเวิร์ก (Wavelet Neural Network : WNN) และแบ็กพรอพาเกชันนิวรอลเน็ตเวิร์ก มารวมกัน โดย WNN มีประสิทธิภาพสูงในการเรียนรู้และ BPN มีความสามารถในการแยกแยะสูง โดยในงานวิจัยยังได้นำเอาเคโฟลด์ครอสวาลิเดชัน (K-Fold Cross Validation) มาใช้เปรียบเทียบประสิทธิภาพนิวรอลเน็ตเวิร์กแบบระบบแยกแยะหลายชุด (Multiple Classifier System, MCS) เช่น Logiboost Bayesian Classifier (LBC),

Multistage Neural networks Ensemble (MNNE) และ Self-Organizing Neural Grove (SONG) ผลการทดลองแสดงให้เห็นว่า HWNN นั้นมีความสามารถในการแยกแยะได้ถูกต้องมากกว่า 3 วิธีดังกล่าวมาแล้วนั้น นอกจากนั้นแล้ว HWNN ยังเร็วกว่า SONG ในแง่ของการประมวลผล การทดลองทั้ง 2 การทดลองแสดงให้เห็นถึงผลการเรียนทั้งในแง่ของเวลาและความถูกต้องที่น่าพอใจ

นิวรอลเน็ตเวิร์กซึ่งมีประโยชน์มากในการประยุกต์ใช้ ความยืดหยุ่น และใช้ได้กว้างขวาง แต่นิวรอลเน็ตเวิร์กแบบเดี่ยว (Single Neural Network) ก็มีข้อจำกัดและบ่อยครั้งที่จะพบว่าการเลือกพารามิเตอร์สำหรับการฝึกฝนและยังต้องการเวลาในการประมวลผลมากสำหรับข้อมูลขนาดใหญ่ด้วย โดยเทคนิคที่ใช้ในการวิจัย ในงานวิจัยได้ใช้โปรแกรมประยุกต์ที่พัฒนาเองที่มีความสามารถเลือกพารามิเตอร์ของการตัดเล็ม (Prune Parameter) และเลือกพารามิเตอร์ของเน็ตเวิร์กที่เหมาะสม ซึ่งทำให้ลดเวลาในส่วนนี้ได้มาก

อีกส่วนที่ถูกนำมาใช้คือ เคโฟลด์ครอสวาลิเดชัน เพื่อที่วัดประสิทธิภาพ การเปรียบเทียบประสิทธิภาพ HWNN ในแง่ของความถูกต้องในการแยกแยะกับ MCS รวมไปถึง MNNE, LBC และ SONG แม้ว่า HWNN จะให้ผลที่ดีในแง่ประสิทธิภาพ และยังได้นำเสนอการปรับปรุง HWNN ให้มีประสิทธิภาพขึ้นไปอีก โดยได้เพิ่มเติมไปในส่วนโมเมนตัม ซึ่งจะให้ผลที่ดีกว่า เพราะฉะนั้นจึงมีการปรับอัลกอริทึมของ HWNN ให้เป็น iHWNN (improved HWNN) ซึ่งนอกจากจะคงประสิทธิภาพของเน็ตเวิร์กเดิมไว้แล้วยังมีความเร็วในแง่ของการลู่เข้าอีกด้วยHWNN มีโครงสร้างพื้นฐานเป็นแบ็กพรอพาเกชันนิวรอลเน็ตเวิร์ก แต่มีการปรับเปลี่ยนใน 2 จุด อย่างแรกคือ ไบแอส (Bias) ที่ถูกปรับเปลี่ยนให้เพิ่มขึ้นในชั้นอินพุตและชั้นแฝง อย่างที่สองคือการใช้ฟังก์ชันการกระตุ้นในชั้นเอาต์พุต

ไบแอสในแบ็กพรอพาเกชันนิวรอลเน็ตเวิร์กเป็นการป้องกันไม่ให้เน็ตเวิร์กหยุดการเรียนรู้ในกรณีที่ค่าอินพุตมีค่าเป็นศูนย์ ซึ่งถ้าเกิดขึ้นจะทำให้ค่าเมทริกซ์ของชุดน้ำหนักไม่มีการเปลี่ยนแปลง และเน็ตเวิร์กไม่สามารถทำการเรียนรู้ต่อไปได้ โดยได้ให้ค่าคงที่เอาต์พุตค่าหนึ่งและส่งกลับไปยังนิวรอนของชั้นอินพุตและชั้นแฝง ซึ่งจะทำให้ค่าอินพุตของทั้งชั้นแฝงและชั้นเอาต์พุตจะไม่มีทางเป็นศูนย์ และได้ปรับฟังก์ชันการกระตุ้นในชั้นเอาต์พุตจากฟังก์ชันเชิงเส้นเป็นฟังก์ชันซิกมอยด์ เพื่อที่สามารถที่จะเรียนรู้ในช่วงที่ดีกว่าและยืดหยุ่นกว่าโครงสร้างของ HWNN

ในชั้นแฝง ค่า $\Psi(t) = \cos(1.75)\exp(-0.5t^2)$ คือค่า Morlet mother wavelet ที่ถูกใช้ในฟังก์ชันกระตุ้นในชั้นแฝง $a_h(1,2,..,H)$ และ $b_h(1,2,..,H)$ และ คือแฟคเตอร์การส่งผ่านในชั้นแฝง

โดยผลการทดสอบในการวัดค่าเฉลี่ยของความถูกต้องในการแยกแยะของนิวรอลเน็ตเวิร์กที่นำมาทดสอบชุดข้อมูล Iris, Pima-Diabetes, Breast Cancer และ Wine เป็นไปตามตารางที่ 1 ซึ่งจะแสดงให้เห็นว่า iHWNN และ HWNN มีความสามารถในการแยกแยะได้มากว่าเทคนิคอื่นๆที่นำมาเปรียบเทียบ

ตารางที่ 1 ค่าเฉลี่ยของความถูกต้องในการแยกแยะของเน็ตเวิร์ก

Dataset	iHWNN	HWNN	MNNE	LBC	SONG
Iris	98.67	98.67	98.00	94.87	96.2
Pima-Diabetes	80.65	80.5	75.13	74.11	74.9
Breast Cancer	97.1	97.09	96.97	96.18	97.0
Wine	99.3	98.8	N/A	98.14	97.6

ในการวิจัยที่ตั้งใจจะทำต่อไปคือการนำเอา iHWNN และ HWNN ไปประยุกต์ใช้ในการวิเคราะห์และเรียนรู้ในข้อมูลขนาดใหญ่เพื่อทำการปรับปรุงอัลกอริทึมให้ดียิ่งขึ้น

2.2.2 งานวิจัยเกี่ยวกับการปรับปรุงประสิทธิภาพนิวรอลเน็ตเวิร์ก

การปรับปรุงประสิทธิภาพของการเรียนรู้ของนิวรอลเน็ตเวิร์กก็เป็นอีกส่วนที่ได้รับความสนใจในการวิจัยเป็นอย่างมาก โดยเฉพาะการปรับปรุงโครงสร้างเน็ตเวิร์กในเหมาะสมที่สุด เพื่อให้ได้นิวรอลเน็ตเวิร์กโครงสร้างสุดท้ายที่จะสามารถนำไปใช้ในการเรียนรู้ในชุดข้อมูลต่างๆได้อย่างมีประสิทธิภาพ

ในปี ค.ศ. 1996 โกเจน หวัง และ ชิเชียง เซน [14] ได้เสนอการปรับอัลกอริทึมให้กับนิวรอลเน็ตเวิร์กโดยปรับจากค่าน้ำหนัก ซึ่งจะใช้ค่าก่อนหน้าที่ดีที่สุดมาเป็นเกณฑ์ในการปรับหลังจากที่ได้ค่าที่เหมาะสม การปรับปรุงนี้จะถูกทำในชั้นแฝงและชั้นเอาต์พุต โดยปรับปรุงประสิทธิภาพการฝึกสอนจากชั้นต่อชั้น โดยนิวรอลเน็ตเวิร์กที่ใช้ จะใช้เป็นชั้นแฝงชั้นเดียวโดยอัลกอริทึมมีหลักการทำงานดังต่อไปนี้

1) เริ่มต้น สุ่มค่าน้ำหนัก W_1 และ W_2 และกลับค่าเมทริกต์ $1/Q_1$ และ $1/Q_2$
2) เริ่มต้นการฝึกสอนผ่านนิวรอลเน็ตเวิร์ก
3) คำนวณค่าเอาต์พุต โดยใช้สมการ 11 โดยค่า o_{iP} คือค่าเอาต์พุตที่ต้องการสำหรับการใช้ฝึกฝนแบบที่ P

$$d_{2iP} = f^{-1}(o_{iP}) \qquad (11)$$

4) คำนวณค่า $\emptyset_2$ โดยใช้สมการ 12

$$\emptyset_2(P-1) = \beta Y^*(P-1) + (1-\beta)Y(P-1) \qquad (12)$$

โดยที่ $Y(P-1)$ คือค่าตั้งต้นของเอาต์พุตในชั้นแฝง และ β คือค่าคงที่ระหว่างศูนย์และหนึ่ง

5) คำนวณค่า K_2 โดยใช้สมการ 13

$$K_2(P) = \frac{Q_2^{-1}(P-1)\emptyset_2(P-1)}{\lambda_2 + \phi_2^T(P-1)Q_2^{-1}(P-1)\phi_2(P-1)}) \qquad (13)$$

6) ปรับปรุงค่าน้ำหนัก W_2 โดยใช้สมการ 14

$$W_2(P) = W_2(P-1) + K_2(P) \times [d_2(P-1) - \emptyset_2^T(P-1)W_2(P-1)] \qquad (14)$$

7) หาค่าเอาต์พุต Yp^*

8) คำนวณค่าเอาต์พุตที่ต้องการจากชั้นแฝงโดยใช้สมการ 15

$$d_{1iP} = f^{-1}(y_{iP}^*) \qquad (15)$$

9) ปรับปรุงค่าน้ำหนัก W_1 โดยใช้สมการ 16

$$W_1(P) = W_1(P-1) + K_1(P) \times [d_1(P-1) - \emptyset_1^T(P-1)W_1(P-1)] \qquad (16)$$

10) ไปยังขั้นตอนที่ 2 ถ้ายังทำงานไม่จบ

11) ไปยังขั้นตอนที่ 2 ถ้าการลู่เข้ายังไม่เสร็จสิ้น

อัลกอริทึมได้ถูกนำไปทดสอบปัญหาซอร์ (XOR), 4-4 เอนโคดเดอร์ (4-4 Encoder) และ SIMO ซึ่งพบว่าสามารถลดค่าผลรวมความผิดพลาดกำลังสอง (Sum Square Error) ได้อย่างน่าพอใจ

ต่อมาในปี ค.ศ.2002 โบ หยาง, เซาฮง ซู และ ยาดอง หวาง [15] ได้เสนอการปรับปรุงประสิทธิภาพของแบ็กพรอพาเกชันนิวรอลเน็ตเวิร์กโดยใช้การปรับปรุงเจเนติกอัลกอริทึมโดยประยุกต์จาก Evolutionarily Stable Strategy (ESS) ซึ่งเป็นแบบจำลองที่ค่อนข้างเสถียรมาก เมื่อนำเอาเจเนติกอัลกอริทึมมาประยุกต์ใช้ ESS เป็น ESSGA โดยจะสามารถแก้ไข้ปัญหาเรื่องข้อจำกัดตัวแปรให้อยู่ในขอบเขตของตัวเลขไม่อนันต์ (Finite Number) โดยมีรายละเอียดและขั้นตอนต่อไปนี้

- สร้างจำนวนประชากรเริ่มต้นให้เป็น A_0 โดยการสุ่มเลือกจากจำนวน N โดยกำหนดแฟคเตอร์เสถียร w พารามิเตอร์ควบคุมความยืดหยุ่น T_K ค่าความแน่นอนของปัญหา L และความน่าจะเป็น P_c เป็นต้น ในพารามิเตอร์เหล่านี้ ค่า w เป็นพารามิเตอร์ที่มีความสำคัญมากเนื่องจากจะมีอย่างมากในการปรับค่าการเลือกองค์ประกอบ ดังสมการ 17

$$w = \left[\frac{N}{\partial}\right] : N, (\partial \geq 1) \qquad (17)$$

- ประเมินผลที่ได้จากการสร้างของเน็ตเวิร์ก
- จดจำค่าโครโมโซมที่ดีที่สุด
- ใส่ค่าที่เลือกบางส่วนเข้าไปในแมทชิ่งพูล (Matching Pool) (M)
- หาค่าสถานะเพื่อนำไปหาค่าแฟคเตอร์ที่คงที่ (S)
- เปลี่ยนแปลงค่าน้ำหนักโดยหาจากค่า (w, S, M) และสร้างโคโลนีใหม่ M_k
- สร้างโคโลนีใหม่ B_k โดยคำนวณจากแต่ละคู่ของบัพแม่ (Parent Node)
- ตัวแปรบุตรจะเปลี่ยนแปลงภายใต้ความควบคุมของตัวแปรที่สร้างในรอบต่อไป (A_{k+1})
- เปลี่ยนแปลงค่าพารามิเตอร์ T_k
- ทำซ้ำตั้งแต่ขั้นตอนแรกจนกว่าจะถึงเงื่อนไขที่หยุด

การลู่เข้าของ ESSGA เป็นไปได้ดี ซึ่งได้แสดงให้เห็นในการทดลองนี้ว่าการปรับปรุงประสิทธิภาพของนิวรอลเน็ตเวิร์ก ซึ่งได้เปรียบเทียบกับ Canonical Genetic Algorithm (CGA) และ Annealing Evolution Algorithm (AEA) และในแง่ของเวลาที่ใช้ แสดงในตารางที่ 2 ซึ่งแสดงให้เห็นว่าเมื่อนำเอา ESSGA มารวมกับ BPNN แล้ววัดผลในเชิงเวลาที่ใช้ในการเรียนรู้

ตารางที่ 2 การเปรียบเทียบประสิทธิภาพของแต่ละอัลกอริทึม

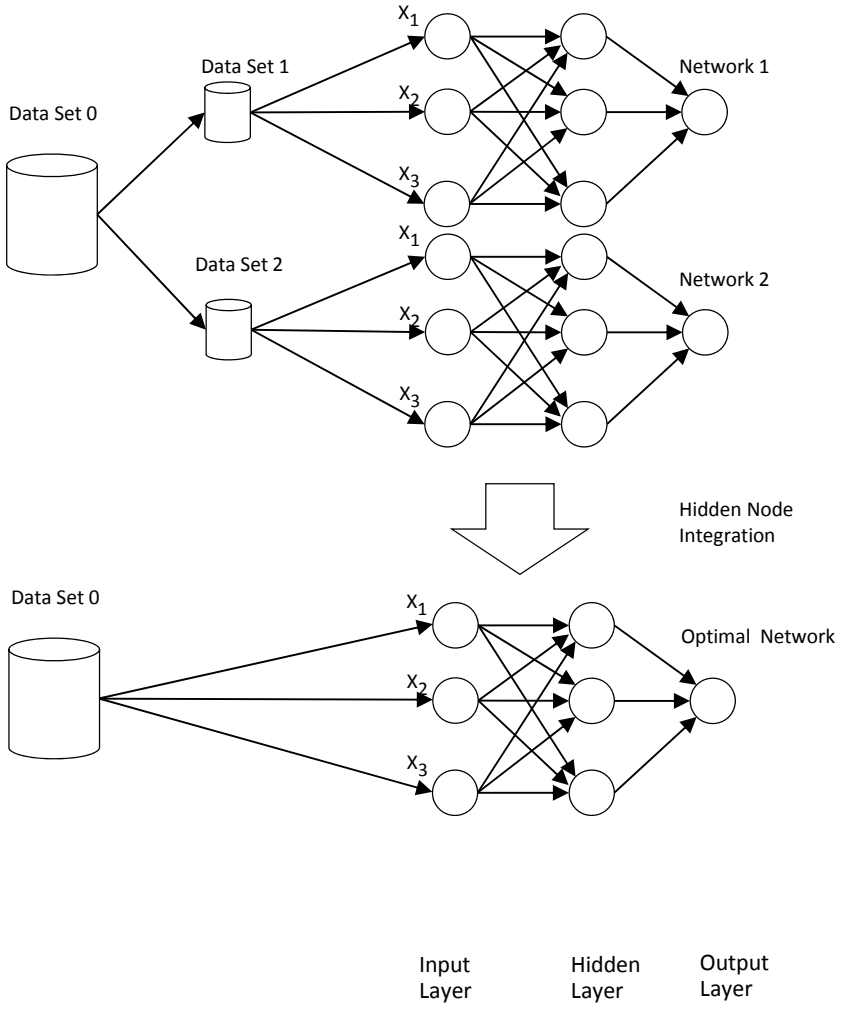
	BP	CGABP	AEABP	ESSBP
Iteration	302350	291162	242366	74144
Running time(sec.)	6.05	6.29	5.49	1.99

จากการประสานแนวคิด ESS เข้ากับเจเนติกอัลกอริทึม ทั้งความเร็วของเน็ตเวิร์กและความถูกต้องได้เพิ่มขึ้นอย่างมากซึ่งจะได้ปรับปรุงในอัลกอริทึมนี้ต่อไป

ในปี ค.ศ. 2009 หยง ลี และ คณะ [16] ได้นำเสนองานวิจัยการใช้วิธีการปรับปรุงการลู่เข้าบนแบ็กพรอพาเกชันนิวรอลเน็ตเวิร์ก ด้วยที่ว่าแบ็กพรอพาเกชันแบบเดิมนั้น อัตราการเรียนรู้ที่ขึ้นกับอัตราการเรียนรู้ (Learning Rate) นั้นจะทำให้การลู่เข้าค่าความผิดพลาดมีอัตราที่ต่ำ ซึ่งได้นำเอาสูตรของเทเลอร์ (Taylor) และให้ค่าอัตราการเรียนรู้ ได้มีการปรับตัวเองเพื่อให้รวดเร็วในการลู่เข้าที่มากขึ้น จะมีผลให้อัตราการลดต่ำลงของความผิดพลาดก็จะลดลงเช่นเดียวกัน

บทที่ 3
วิธีดำเนินงานวิจัย

งานวิจัยนี้สนใจการปรับปรุงวิธีการเรียนรู้ของนิวรอลเน็ตเวิร์กบนข้อมูลขนาดใหญ่โดยใช้แนวคิดที่ว่าลดเวลาการเรียนรู้ของนิวรอลเน็ตเวิร์กลงโดยการแบ่งข้อมูลออกเป็นกลุ่มข้อมูลย่อยๆ จากนั้นทำการเรียนรู้ให้กับข้อมูลย่อยเหล่านั้น โดยใช้เน็ตเวิร์กที่มีโครงสร้างเดียวกัน เหมือนกัน ในการเรียนรู้ จากนั้น เอาค่าน้ำหนักที่ได้หลังจากการเรียนรู้ มาทำการวิเคราะห์เพื่อรวมโหนดที่เหมาะสม เพื่อให้ได้นิวรอลเน็ตเวิร์กสุดท้ายที่มีค่าเหมาะสมที่สุด (Optimized Network) ดังภาพที่ 2



ภาพที่ 2 แนวคิดในงานวิจัยในการรวมนิวรอลเน็ตเวิร์กชั้นแฝงเดี่ยว

การวิจัยนี้ได้มุ่งเน้นในการพัฒนาอัลกอริทึมในการเรียนรู้ให้กับเซตข้อมูลทดสอบโดยการแบ่งเซตข้อมูลออกเป็นเซตย่อยๆ เท่าๆกัน ซึ่งในการเรียนรู้ในขนาดข้อมูลที่น้อยกว่าจะใช้เวลาที่น้อยกว่าเนื่องจากอัตราการเรียนรู้ที่รวดเร็วกว่า บนจำนวนรอบการเรียนรู้ที่เท่ากัน ในงานวิจัยจะกำหนดที่ 10,000 รอบ เป็นต้น แต่ค่าของความผิดพลาดก็จะไม่แน่นอน บางชุดจะให้ค่าความผิดพลาดที่ต่ำกว่าหรือสูงกว่าเซตข้อมูลหลัก (ข้อมูลก่อนการแบ่ง) แต่เราสามารถเอาชุดที่มีค่าความผิดพลาดที่ต่ำที่สุดมาเป็นเน็ตเวิร์กตั้งต้นในการทดลอง โดยการนำเอาค่าน้ำหนัก ในเน็ตเวิร์กดังกล่าวนี้ไปเปรียบเทียบกับโหนดในเน็ตเวิร์กย่อยอื่นๆ ก่อนที่จะทำการรวมโหนดในชั้นแฝงเพื่อให้ได้เน็ตเวิร์กใหม่

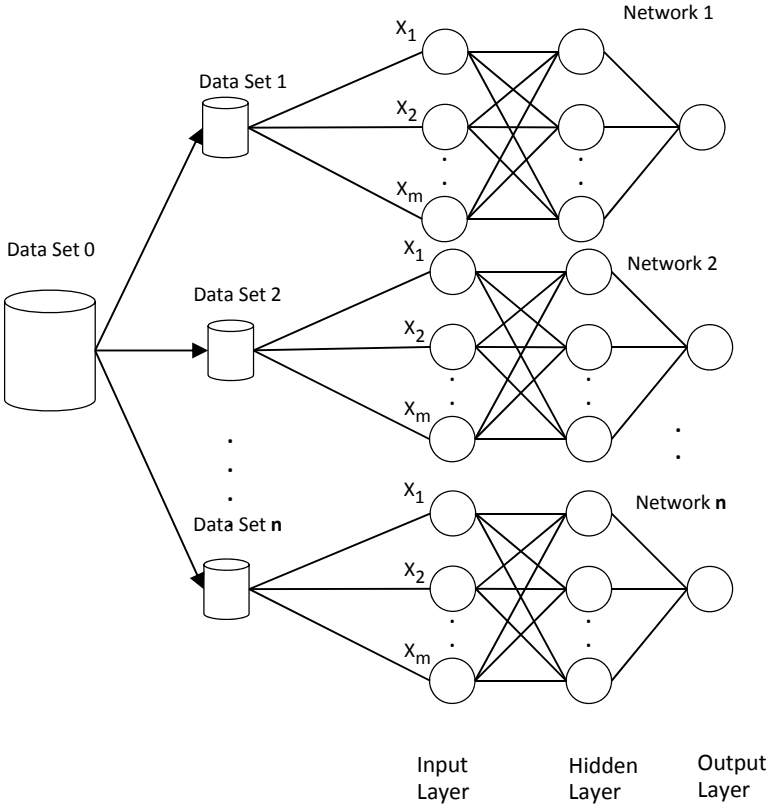
สมมติให้เซตข้อมูลมีจำนวนตัวอย่างทั้งหมด X ตัวอย่าง และให้เซตข้อมูลหลักเป็น Data Set 0 (D_0) และเซตข้อมูลที่แบ่งย่อยออกเป็น n กลุ่ม และให้ N คือ นิวรอลเน็ตเวิร์ก ตั้งแต่ N_1 จนไปถึง N_n โดยแต่ละกลุ่มจะมีข้อมูลเท่าๆกันคือ X/n ตัวอย่าง จะทำให้มี Data Set 1 ถึง N แทนเป็น D_1, D_2.., D_n โดยแต่ละเซตจะถูกเรียนรู้ในนิวรอลเน็ตเวิร์กที่มีโครงสร้างเดียวกันและเป็นแบบนิวรอลเน็ตเวิร์กแบบมีชั้นแฝงชั้นเดียว อย่างเช่น การทดสอบข้อมูล 10,000 เร็คคอร์ด แบ่งออกเป็น 10 เน็ตเวิร์ก เน็ตเวิร์กละ 800 เร็คคอร์ด เหลืออีก 2,000 เร็คคอร์ดเก็บไว้เป็นข้อมูลตรวจสอบเพื่อใช้ในการตรวจสอบ เป็นต้น ดังภาพที่ 3

เมื่อทำการเรียนรู้จนครบ n เซตแล้ว เก็บค่าน้ำหนักของแต่ละเน็ตเวิร์กเพื่อนำมาวิเคราะห์เปรียบเทียบค่า โดยลองเอาชุดค่าน้ำหนักของชุดที่ทำการแบ่งแยกชุดตัวอย่างได้ดีที่สุด คือสามารถเรียนรู้โดยมีความผิดพลาดที่น้อยที่สุด กำหนดให้เป็น Network Best (N_{best}) และเมื่อนำเอาค่าน้ำหนักที่เน็ตเวิร์ก N_{best} ในชั้นแฝง-ชั้นเอาต์พุตไปเปรียบเทียบกับชุดน้ำหนักของโหนดอื่นๆในชั้นเดียวกัน และหาค่าเฉลี่ยและทดสอบทีละโหนดเพื่อหาโหนดที่ดีที่สุด คือเมื่อเฉลี่ยค่าน้ำหนักและรวมโหนดแล้วนำเอาน้ำหนักไปทดสอบในข้อมูลตรวจสอบ แล้วดูค่าความผิดพลาดว่าน้อยลงกว่าเดิมหรือไม่

จากนั้นทำการเปรียบเทียบในชั้นอินพุต-ชั้นแฝงโดยใช้การเปรียบเทียบกลุ่มของน้ำหนักด้วยระยะยุคลิดเพื่อดูความใกล้ ถ้ามีค่ายุคลิดน้อย จะบ่งบอกได้ว่าเน็ตเวิร์กที่ทำการเปรียบเทียบทั้งสองนั้นมีความใกล้เคียงกัน สามารถนำมาเฉลี่ยรวมโหนดเข้าด้วยกันได้ โดยเปรียบเทียบไปจนครบทุกโหนด

แนวคิดนี้ทำไปประยุกต์กับการเรียนรู้ของเซตข้อมูลขนาดใหญ่ได้ เพราะในทางปฏิบัติแล้วการที่จะเรียนรู้ข้อมูลขนาดใหญ่ตรงๆนั้น เป็นเรื่องที่ทำได้ยากมาก เราใช้แนวคิดในการแบ่งข้อมูล

ออกเป็นเซตข้อมูลย่อยนี้เพื่อให้การวิเคราะห์ข้อมูลขนาดใหญ่ทำได้ง่ายขึ้น ในขณะที่ความผิดพลาดไม่ได้ต่างหรือใกล้เคียงกับของเดิมไม่มากนัก

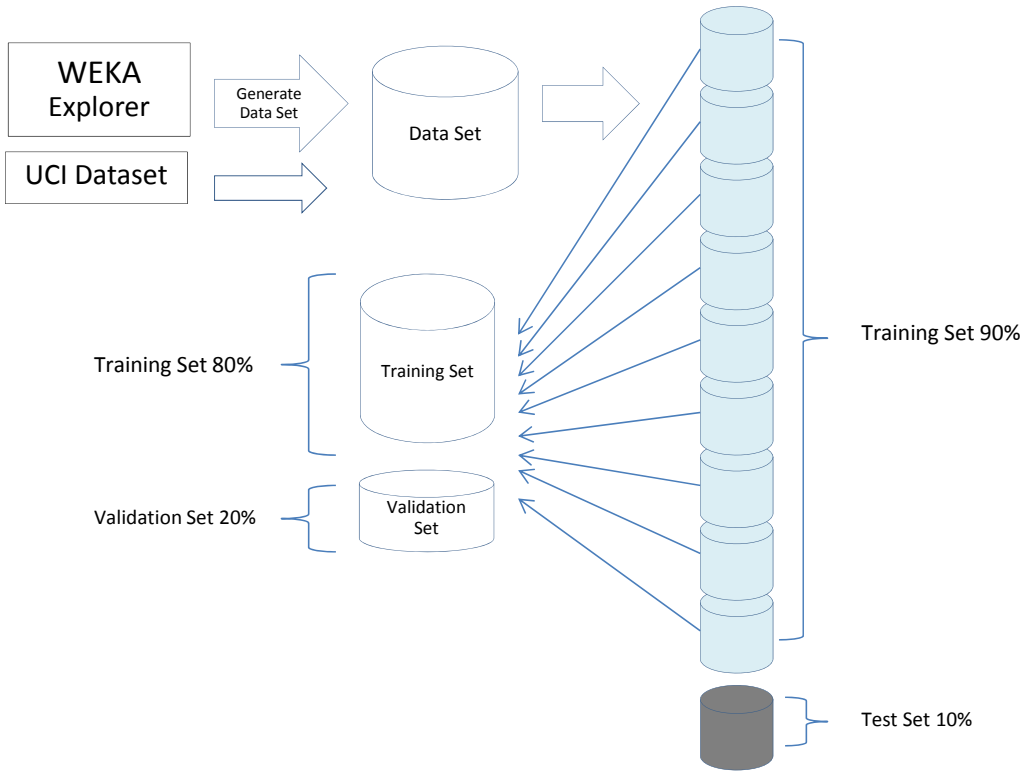


ภาพที่ 3 แบบจำลองของนิวรอลเน็ตเวิร์กที่ใช้ในงานวิจัย

3.1 การเตรียมข้อมูลสำหรับการเรียนรู้

การเตรียมข้อมูลเพื่อทำการทดสอบในงานวิจัย จะใช้ข้อมูล 3 ระดับ โดยมีข้อมูลขนาดเล็ก ขนาดกลาง และขนาดใหญ่ โดยชุดข้อมูล Iris และ Pima Diabetes เป็นตัวแทนชุดข้อมูลขนาดเล็ก เพื่อเริ่มทำการสร้างแบบจำลอง ส่วนข้อมูลขนาดกลางและขนาดใหญ่ จะสร้างบนโปรแกรมประยุกต์เวกา โดยจะนำเอาชุดข้อมูลนำออกจากโปรแกรมเวกาให้อยู่ในรูปแบบไฟล์ arff ซึ่งเป็นไฟล์ในรูปแบบข้อความ โดยทำการแปลงไฟล์ arff ให้อยู่ในรูปแบบไฟล์ CSV

หลังจากนั้นทำการนำข้อมูลเข้าจากไฟล์ CSV ไปในฐานข้อมูลเอสคิวแอล เพื่อใช้คำสั่งเอสคิวแอลช่วยในการแบ่งข้อมูลเป็นข้อมูลย่อยๆ 10 กลุ่ม เพื่อจะแบ่งเป็นชุดข้อมูลฝึกสอน ข้อมูลทดสอบ และข้อมูลตรวจสอบ และทำการส่งออกข้อมูลออกไปเป็นไฟล์ CSV เพื่อเตรียมให้โปรแกรมเอ็มบีพี ทำการฝึกฝนข้อมูลต่อไป



ภาพที่ 4 แบบจำลองการเตรียมข้อมูลสำหรับการฝึกสอน, การทดสอบ, และการตรวจสอบ

ในงานวิจัยนี้ ได้ใช้การแบ่งข้อมูลโดยใช้เทคนิค เคโฟลด์ ครอส วาลิเดชัน (K-Fold Cross Validation) [17] ซึ่งในการทำการแยกแยะข้อมูลเพื่อที่จะวิเคราะห์ข้อมูลโดยเทคนิคที่ใช้กันมาก เช่น ต้นไม้ตัดสินใจ (Decision Tree) หรือนิวรอลเน็ตเวิร์กนั้น จะแบ่งข้อมูลออกเป็นชุดฝึกสอนและชุดทดสอบ แต่บางครั้งอาจจะเกิดปัญหาในการเลือกข้อมูลที่ดีและง่ายเกินไปมาเป็นชุดข้อมูลทดสอบ ผลทำให้ข้อมูลนั้นดีเกินจริงดังนั้นจะมีการคิดวิธีเคโฟลด์ ครอส วาลิเดชัน มาช่วยแก้ปัญหาดังกล่าวได้

เคโฟลด์ ครอส วาลิเดชัน จะแบ่งข้อมูลออกเป็นชุดย่อยจำนวน K ชุด (K-Fold) ซึ่งในขั้นตอนแรกเลือกข้อมูลชุดที่ 1 เป็นข้อมูลชุดทดสอบ และข้อมูลชุดที่เหลือจะเป็นข้อมูลที่ใช้เป็นชุดฝึกสอน หลักจากฝึกสอนแล้ว จากนั้นก็จะสลับข้อมูลชุดที่ 2 มาเป็นชุดทดสอบและข้อมูลกลุ่มอื่นๆ ที่เหลือเป็นชุดฝึกสอน สลับอย่างนี้ไปเรื่อยๆจนครบ K ชุด แลในขั้นตอนสุดท้ายจะหาค่าเฉลี่ยของค่าความถูกต้องในแต่ละชุด วิธีการนี้ชุดข้อมูลย่อยทุกชุดจะมีโอกาสได้เป็นทั้งชุดฝึกสอนและชุดทดสอบ

จากภาพที่ 4 ข้อมูลที่ได้จะมาจากชุดข้อมูลจากยูซีไอ (UCI Repository) โดยใช้ข้อมูล Iris และ Pima Diabetes และอีกส่วนมาจากการสร้างขึ้นจากโปรแกรมเวกาเอ็กพลอเรอร์ (WEKA Exlplorer) ซึ่งวิธีการสร้างข้อมูลจะแสดงไว้ตามภาคผนวก ก

ข้อมูลเหล่านี้นำมาแบ่งเป็นชุดฝึกสอน ชุดทดสอบและชุดตรวจสอบ เพื่อหาเน็ตเวิร์กที่เหมาะสมที่สุดในการเรียนรู้ชุดข้อมูลนี้เปรียบเทียบกับวิธีการเรียนรู้แบบปกติ

ภาพที่ 5 การแบ่งข้อมูลสำหรับการทดสอบแบบ 10-fold Cross Validation

ในการสร้างชุดข้อมูล จะกำหนดชนิดของข้อมูลเป็นแบบตัวเลข (Numeric Attribuites) เพื่อง่ายต่อการเปรียบเทียบและวิเคราะห์ จำนวนคลาสจะขึ้นกับการกระจายตัวของข้อมูลที่เราเลือก (Data Distribution) จากนั้นทำการแบ่งข้อมูลไว้ทดสอบโดยจะแบ่งเป็น 3 กลุ่มคือ ข้อมูลสำหรับการฝึกสอน ข้อมูลสำหรับการตรวจสอบ และข้อมูลสำหรับการทดสอบ โดยข้อมูลเหล่านี้จะถูกทำให้เรียนรู้ ชุดละ 10 รอบ และแต่ละชุดก็จะได้เป็นชุดทดสอบ วนไปเรื่อยๆจนครบ 10 รอบ ดังภาพที่ 5

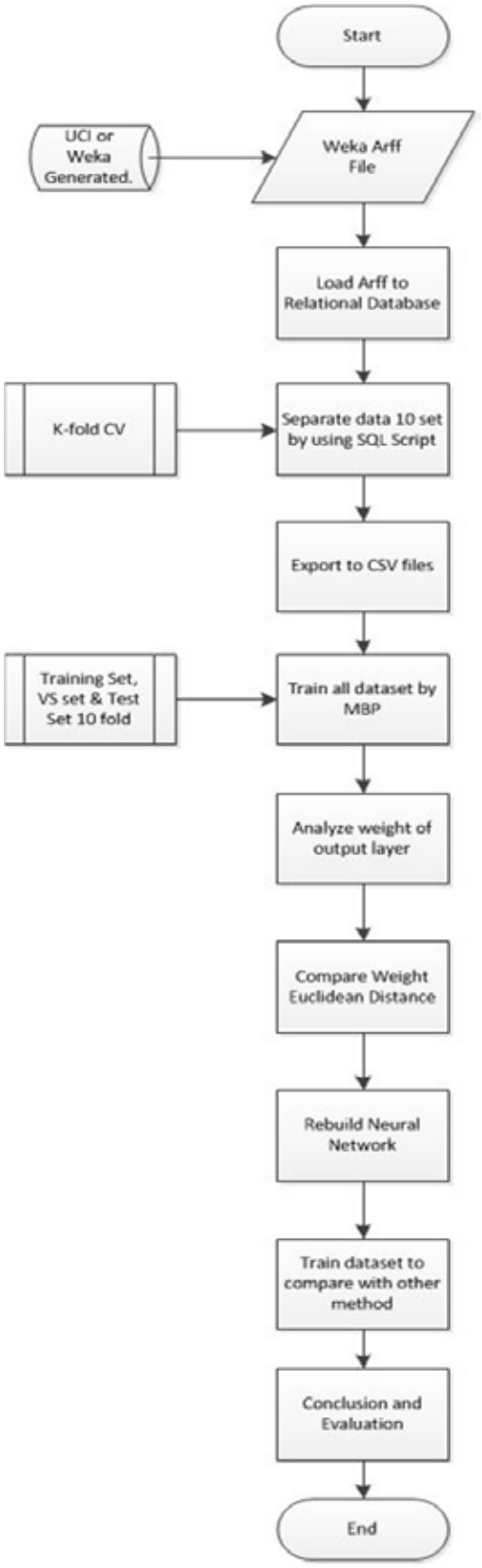
กระบวนการในการวิจัย จะเริ่มด้วยการเตรียมข้อมูลจากโปรแกรมประยุกต์เวกา ด้วยชุดข้อมูลเล็กๆก่อน จุดประสงค์คือ เพื่อง่ายต่อการวิเคราะห์และเพื่อทดสอบสร้างอัลกอริทึม โดยเริ่มจากข้อมูลจำนวนน้อยแบบง่ายๆ หลังจากนั้นนำวิธีการนี้มาประยุกต์ใช้ทดสอบกับชุดข้อมูลมาตรฐานยูซีไอ เพื่อที่จะนำไปเปรียบเทียบผลการทดลองต่อไป

เมื่อทำการคัดเลือกชุดข้อมูลแล้ว ก็นำชุดข้อมูลนำเข้าไปยังฐานข้อมูลเพื่อพักเก็บไว้ ซึ่งในงานวิจัยนี้ใช้ฐานข้อมูลไมโครซอฟต์เอสคิวแอลเซิฟเวอร์ แล้วจากนั้นก็พัฒนาชุดคำสั่งเอสคิวแอล

(SQL Script) เพื่อทำการแบ่งข้อมูลออกให้เป็นไปตามภาพที่ 4 และ 5 จากนั้นก็ทำการนำข้อมูลออกมาในรูปแบบไฟล์ CSV เป็นชุดๆ ดังตัวอย่างเอสคิวแอลที่ยกตัวอย่างไว้ในภาคผนวก ข

ทำการฝึกสอนชุดข้อมูลเหล่านั้นตามภาพที่ 5 โดยมี 1 ชุดที่เป็นชุดทดสอบ วนชุดฝึกสอนและชุดทดสอบไปเรื่อยๆโดยโปรแกรมเอ็มบีพี ทั้งนี้ก็เก็บข้อมูลของค่าน้ำหนักของแต่ละครั้งด้วยเพื่อการวิเคราะห์ จากนั้นวิเคราะห์ข้อมูลน้ำหนักในชั้นแฝงไปยังชั้นเอาต์พุตเพื่อเปรียบเทียบการแยกแยะในชั้นนี้ หาระยะยุคลิดเพื่อดูว่าโหนดในชั้นแฝงใกล้เคียงกับโหนดอื่นๆในเน็ตเวิร์กอื่นหรือไม่ ทำการสลับโหนดและรวมโหนด จากนั้นหาค่าระยะยุคลิดของน้ำหนักในชั้นอินพุตไปยังชั้นแฝงเพื่อหาความคล้าย

นำผลการวิเคราะห์ที่ได้มาสร้างนิวรอลเน็ตเวิร์กใหม่และนำไปทดสอบกับชุดข้อมูลเพื่อดูความถูกต้องและความเร็วที่ใช้ในการเรียนรู้ โดยกระบวนการทำการวิจัยจะแสดงไว้ตามภาพที่ 6



ภาพที่ 6 ขั้นตอนและกระบวนการในการทำการวิจัย

3.1.1 **ชุดข้อมูลทดสอบที่** 1 Iris

รายละเอียดของชุดข้อมูลที่ 1 ข้อมูล Iris ดังแสดงในตารางที่ 3

ตารางที่ 3 รายละเอียดของข้อมูลทดสอบชุดที่ 1

รายละเอียด	ค่า
ชื่อข้อมูล	Iris
จำนวนข้อมูล	150
จำนวนลักษณะประจำข้อมูล (Data Attribute)	4
จำนวนคลาส	3
ชนิดข้อมูล	ตัวเลข
จำนวนเน็ตเวิร์กย่อย	10

3.1.2 **ชุดข้อมูลทดสอบที่** 2 Pima Diabetes

รายละเอียดของชุดข้อมูลที่ 2 ข้อมูล Pima Diabetes ดังแสดงในตารางที่ 4

ตารางที่ 4 รายละเอียดของข้อมูลทดสอบชุดที่ 2

รายละเอียด	ค่า
ชื่อข้อมูล	Pima Diabetes
จำนวนข้อมูล	768
จำนวนลักษณะประจำข้อมูล (Data Attribute)	8
จำนวนคลาส	2
ชนิดข้อมูล	ตัวเลข
จำนวนเน็ตเวิร์กย่อย	10

3.1.3 **ชุดข้อมูลทดสอบที่ 3 ข้อมูลสร้างชุดที่ 1**

รายละเอียดของชุดข้อมูลที่ 3 Data Geberated 1 ดังแสดงในตารางที่ 5

ตารางที่ 5 รายละเอียดของข้อมูลทดสอบชุดที่ 3

รายละเอียด	ค่า
ชื่อข้อมูล	Data Generated 1
จำนวนข้อมูล	10,000
จำนวนลักษณะประจำข้อมูล (Data Attribute)	4
จำนวนคลาส	2
ชนิดข้อมูล	ตัวเลข
จำนวนเน็ตเวิร์กย่อย	10

3.1.4 **ชุดข้อมูลทดสอบที่ 4 ข้อมูลสร้างชุดที่ 2**

รายละเอียดของชุดข้อมูลที่ 4 Data Geberated 2 ดังแสดงในตารางที่ 6

ตารางที่ 6 รายละเอียดของข้อมูลทดสอบชุดที่ 4

รายละเอียด	ค่า
ชื่อข้อมูล	Data Generated 2
จำนวนข้อมูล	100,000
จำนวนลักษณะประจำข้อมูล (Data Attribute)	4
จำนวนคลาส	2
ชนิดข้อมูล	ตัวเลข
จำนวนเน็ตเวิร์กย่อย	10

3.2 การเรียนรู้ของเน็ตเวิร์กย่อย

ในการทำการเรียนรู้ให้กับชุดข้อมูลเพื่อทำการเก็บข้อมูลของน้ำหนัก จะใช้โปรแกรมประยุกต์เอ็มบีพี เพราะนอกจากจะสามารถกำหนดรูปแบบของเน็ตเวิร์กได้ง่ายแล้ว ยังสามารถที่จะเก็บข้อมูลการเรียนรู้ ความผิดพลาดรากกำลังสองเฉลี่ย (Root Mean Square Error) และค่าน้ำหนักของเน็ตเวิร์กแต่ละชั้นได้ง่าย

ในการกำหนดโครงแบบให้กับเน็ตเวิร์กจะกำหนดตามจำนวนลักษณะประจำข้อมูล จำนวนคลาส

ตารางที่ 7 รายละเอียดเน็ตเวิร์กแต่ละชุดข้อมูลที่ถูกกำหนดในโปรแกรมเอ็มบีพี

ข้อมูลชุด	อินพุต	ลักษณะประจำ	คลาส	โครงสร้างของเน็ตเวิร์ก
1	4	4	3	4-3-1
2	8	8	3	8-2-1
3	4	4	2	4-2-1
4	4	4	2	4-2-1

เมื่อทำการสร้างข้อมูลทดสอบแล้วจากข้อ 3.1 ทำการแบ่งข้อมูลออกเป็นชุดย่อยๆ ตามจำนวนเน็ตเวิร์กย่อยในตารางที่ 3, 4, 5 และ 6 และส่งออกข้อมูลในเอสคิวแอลให้อยู่ในรูปไฟล์ CSV ดังภาคผนวก ข

ในการเรียนรู้ให้กับชุดข้อมูลตามตารางที่ 3, 4, 5 และ 6 นั้น จะปฏิบัติตามขั้นตอนดังนี้

- เลือกไฟล์ CSV ที่เตรียมไว้เพื่อทำการเรียนรู้
- กำหนดโครงสร้างเน็ตเวิร์ก ให้เป็นไปตามตารางที่ 7
- กดปุ่ม Configuration เพื่อทำการกำหนดค่าจำนวนรอบการเรียนรู้ ให้กับเน็ตเวิร์ก
- ในส่วน Stopping Criteria ให้กำหนด “Stop after 10,000 epochs” เป็นการกำหนดจำนวนรอบในการเรียนรู้คงที่
- กดปุ่ม Train เพื่อทำการเรียนรู้ให้กับชุดข้อมูล
- หลังจากทำการเรียนรู้ให้กับเน็ตเวิร์ก ทำการเก็บค่าน้ำหนักของเน็ตเวิร์ก โดยการบันทึก (Save Weight) ออกไปเป็นไฟล์นามสกุล BPW

- เช่นเดียวกันกับการที่เราจะนำเอาค่าน้ำหนักที่ผ่านการรวมโหนดและปรับปรุงค่าแล้วย้อนกลับไปในเน็ตเวิร์ก ให้กดปุ่ม Load เพื่อทำการโหลดค่าน้ำหนักเข้าไปในเน็ตเวิร์ก

ในการเก็บข้อมูลการเรียนรู้ของแต่ละเรคคอร์ดจากโปรแกรมเอ็มบีพีทำได้โดยการคลิ๊กเม้าส์ขวาที่กราฟ แล้วเลือก “Copy Data” จากนั้นก็สามารถเอาข้อมูลไปวางไว้ในโปรแกรมประยุกต์เอกเซล (Excel) หรือ บนเทกต์อิดิเตอร์ (Text Editor) อื่นๆ เพื่อวิเคราะห์ต่อไป

3.3 การเปรียบเทียบค่าน้ำหนักและการรวมโหนด

การเปรียบเทียบค่าน้ำหนักที่เก็บมาได้ ในเบื้องต้นเราต้องหาชุดเน็ตเวิร์กที่มีค่าความผิดพลาดรากกำลังสองเฉลี่ยที่น้อยที่สุด จะถือว่าเป็นเน็ตเวิร์กที่ดีที่สุดบนชุดข้อมูลตรวจสอบที่เตรียมไว้แล้ว ให้เป็น N_{best} โดยการเรียนรู้ชุดข้อมูลตรวจสอบให้ครบทุกนิวรอลเน็ตเวิร์ก ซึ่งในการวิจัยได้ใช้โปรแกรมประยุกต์เอ็มบีพี ซึ่งมีฟังก์ชันในการทดแทนค่าน้ำหนักเข้าไปยังนิวรอลเน็ตเวิร์ก จากนั้นทำการวิเคราะห์และเปรียบเทียบค่าน้ำหนักของชั้นแฝงไปชั้นเอาต์พุตของ N_{best} กับค่าน้ำหนักในชั้นเดียวกันของชุดอื่นๆ โดยขั้นตอนต่างๆดังแสดงไว้ในภาพที่ 7

ในการรวมโหนดในชั้นแฝงและชั้นเอาต์พุตของงานวิจัยนี้ ได้ใช้หลักการเปรียบเทียบโหนดจาก N_{best} แต่ละโหนด ไปเปรียบเทียบกับโหนดอื่นๆที่เหลือในชั้นเดียวกันของเน็ตเวิร์กอื่น โดยแต่ละเน็ตเวิร์กต้องทำการนอร์มอลไลซ์ (Normalize) ค่าน้ำหนักโดยการเอาค่าไบแอส มาหารค่าน้ำหนักทุกค่าในโหนด และสลับโหนดจากน้อยไปหามากโดยจะต้องสลับโหนดที่ชั้นอินพุต-ชั้นแฝงด้วย จากนั้นหาระยะยุคลิด นำเอาโหนดที่ใกล้กันเอาค่าน้ำหนักมาเฉลี่ยกัน แล้วทำไปทดสอบเรียนรู้ในชุดข้อมูลตรวจสอบเพื่อดูผลการเรียนรู้ การเฉลี่ยค่าน้ำหนักจะแสดงในสมการที่ 18

ให้ $W_{N_{best}}$ แทนค่าน้ำหนักของเน็ตเวิร์กที่ดีที่สุด 1

W_{N_i} แทนค่าน้ำหนักของเน็ตเวิร์กที่ i

$$W_{N_{best}} \leftarrow \left(W_{N_{best}} + W_{N_i}\right)/2 \qquad (18)$$

เมื่อเน็ตเวิร์กทำการเฉลี่ยค่าน้ำหนักและแทนค่าเข้าไปใน N_{best} แล้วที่ละโหนด แล้วทดสอบการเรียนรู้กับชุดข้อมูลตรวจสอบจนได้ค่า $W_{N_{best}}$ ที่ดีที่สุด คือค่าน้ำหนักของเน็ตเวิร์กที่มีค่าความผิดพลาดน้อยที่สุด

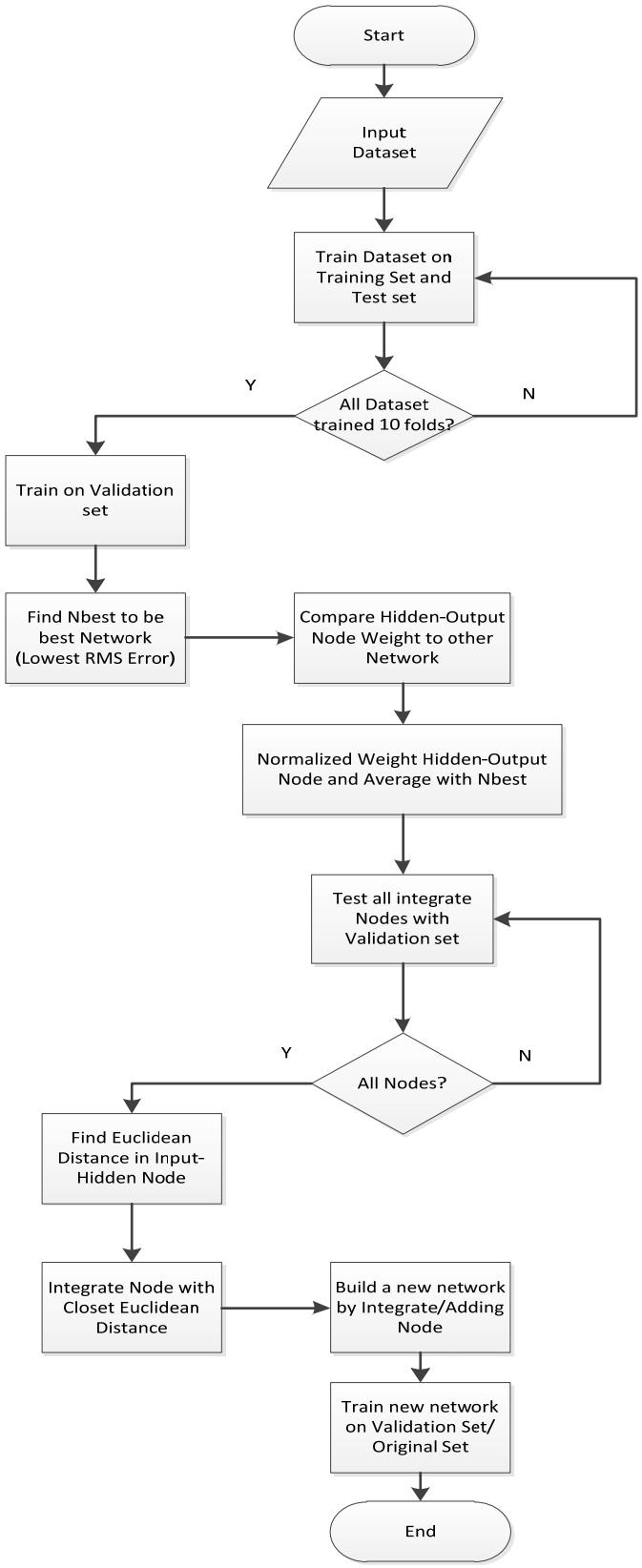
จากนั้นทำการรวมโหนดในชั้นอินพุต-ชั้นแฝง โดยมีหลักการคือในการที่จะวิเคราะห์ว่าโหนดของเน็ตเวิร์ก 2 เน็ตเวิร์ก มีความคล้ายคลึงกันนั้น เราจำเป็นต้องสร้างกฎการเรียนรู้ขึ้นมาเพื่อกำหนดขอบเขตในการวัดและตัดสินใจ

กฎการเรียนรู้ในการรวมโหนด (Node Integration Training Rule)

- ทำการนอร์มอลไลซ์ค่าน้ำหนักในโหนดโดยเอาค่าไบแอสไปหารน้ำหนักในโหนดทั้งหมด
- หาระยะยุคลิดของโหนดนิวรอน 2 โหนดใน 2 เน็ตเวิร์ก จะมีความคล้ายคลึงกันก็ต่อเมื่อ ค่า y ที่ได้ มีค่าความต่างจากกัน ไม่เกิน 10 เปอร์เซ็น เช่น 1 กับ 0.95 มีความคล้ายคลึงกัน เป็นต้น
- ถ้าวัดออกมาแล้ว 2 โหนดมีความคล้ายคลึงกัน ค่าน้ำหนักของทั้ง 2 ดังสมการที่ ดังสมการที่ 19

$$y = d(W_{1i}, W_{1j}) = \sqrt{\sum_{i=1}^{n}(W_{1i} - W_{2i})^2} \qquad (19)$$

เมื่อได้กฎการเรียนรู้นี้แล้ว เราสามารถใช้กฎดังกล่าวมาสร้างเป็นอัลกอริทึมในการรวมโหนดเพื่อให้ได้โหนดบนเน็ตเวิร์กสุดท้ายที่ดีที่สุดตามภาพที่ 7



ภาพที่ 7 ขั้นตอนการทดลองและอัลกอริทึมการรวมโหนด

บทที่ 4
การทดลองและผลการทดลอง

4.1 การทดลอง

ในการทดลองกับชุดข้อมูลดังบทที่ 3.1 และทำการเรียนรู้ด้วยโปรแกรมประยุกต์เอ็มบีพี และทำการเก็บข้อมูลเพื่อทำการวิเคราะห์น้ำหนัก

4.1.1 เครื่องมือที่ใช้ในการวิจัย

ในการวิจัยนี้ใช้เครื่องมือและโปรแกรมประยุกต์ต่างๆดังต่อไปนี้

- ฮาร์ดแวร์
  - หน่วยประมวลผลอินเทล คอร์ ไอเซเว่น 1.73 กิกะเฮิร์ต (Intel® Core™ i7 1.73 GHz)
  - หน่วยความจำ 4 กิกะไบต์ (RAM 4 GB)
  - ฮาร์ดดิสก์ 500 กิกะไบต์ (Hard Disk 500 GB)
- ซอฟต์แวร์
  - โปรแกรมเวกา เวอร์ชัน 3.6.0 เพื่อทำการนำเข้าและส่งออกข้อมูลในรูปแบบ ARFF
  - โปรแกรมไมโครซอฟต์เอสคิวแอลเซิฟเวอร์ เวอร์ชัน 2005 (Microsoft SQL Server 2005) เพื่อทำการนำเข้าข้อมูล แบ่งข้อมูลเป็นกลุ่มย่อย และส่งออกในรูปแบบ CSV
  - โปรแกรมเอ็มบีพี เวอร์ชัน 2.2.4 เพื่อทำการเรียนรู้ให้กับข้อมูลทดสอบ
  - โปรแกรมไมโครซอฟต์เอกเซล (Excel) เวอร์ชัน 2010 เพื่อทำการวิเคราะห์ข้อมูลและค่าน้ำหนักที่ได้

4.1.2 การเรียนรู้ของเน็ตเวิร์กย่อยเพื่อหาเน็ตเวิร์กที่ดีที่สุด

ตารางที่ 8-11 แสดงผลของค่าความผิดพลาดรากกำลังสองเฉลี่ยของเน็ตเวิร์กย่อยในแต่ละชุดบนชุดข้อมูลตรวจสอบ ของแต่ละชุดข้อมูลตัวอย่าง เพื่อทำการหาค่าเน็ตเวิร์กที่ดีที่สุด คือค่าความผิดพลาดรากกำลังสองเฉลี่ยที่ต่ำที่สุด เพื่อให้เป็นค่าเน็ตเวิร์กตั้งต้น N_{best}

ตารางที่ 8 ค่าความผิดพลาดรากกำลังสองเฉลี่ยชุดข้อมูลที่ 1 Iris

เน็ตเวิร์ก	ค่าความผิดพลาดรากกำลังสองเฉลี่ย (RMS Error)
N_1	0.000429
N_2	0.000433
N_3	0.000324
N_4	0.000328
N_5	0.000314
N_6	0.000485
N_7	**0.000038**
N_8	0.000397
N_9	0.000276
N_{10}	0.000397

ตารางที่ 9 ค่าความผิดพลาดรากกำลังสองเฉลี่ยชุดข้อมูลที่ 2 Pima Diabetes

เน็ตเวิร์ก	ค่าความผิดพลาดรากกำลังสองเฉลี่ย (RMS Error)
N_1	0.129101
N_2	0.105427
N_3	0.176832
N_4	0.002156
N_5	0.000393
N_6	0.111929
N_7	0.000311
N_8	**0.000011**
N_9	0.001229
N_{10}	0.001226

ตารางที่ 10 ค่าความผิดพลาดรากกำลังสองเฉลี่ยชุดข้อมูลที่ 3 Data Generated 1

เน็ตเวิร์ก	ค่าความผิดพลาดรากกำลังสองเฉลี่ย (RMS Error)
N_1	0.051776
N_2	0.051624
N_3	0.051582
N_4	0.052217
N_5	0.051728
N_6	0.051698
N_7	**0.051256**
N_8	0.051951
N_9	0.051957
N_{10}	0.051424

ตารางที่ 11 ค่าความผิดพลาดรากกำลังสองเฉลี่ยชุดข้อมูลที่ 4 Data Generated 2

เน็ตเวิร์ก	ค่าความผิดพลาดรากกำลังสองเฉลี่ย (RMS Error)
N_1	0.043746
N_2	**0.043655**
N_3	0.043821
N_4	0.044734
N_5	0.043762
N_6	0.044067
N_7	0.044295
N_8	0.043766
N_9	0.043821
N_{10}	0.043664

จากตารางที่ 8 ถึง 11 เราจะได้ค่า N_{best} คือ N_7 สำหรับข้อมูล Iris, N_8 สำหรับ Pima Diabetes, N_7 สำหรับ Data Generated 1 และ N_2 สำหรับ Data Generated 2

4.1.3 การรวมโหนดในชั้นแฝง-ชั้นเอาต์พุต

จากการทดลองที่ 4.1.2 เราจะได้เน็ตเวิร์กที่ดีที่สุด N_{best} ของแต่ละชุดข้อมูลเพื่อเป็นเน็ตเวิร์กตั้งต้นในการรวมโหนด โดยค่าน้ำหนักย่อยของแต่ละเน็ตเวิร์กที่แสดงในตารางได้จากค่าน้ำหนักชั้นแฝงไปยังชั้นเอาต์พุต ดังแสดงรายละเอียดไว้ในภาคผนวก ค

เมื่อทำการรวมโหนดและทดสอบกับข้อมูลตรวจสอบแล้วจะได้เน็ตเวิร์กที่ทำการรวมค่าในชั้นแฝง-เอาต์พุตแล้วดังแสดงในตารางที่ 12

ตารางที่ 12 ค่าค่าน้ำหนักในชั้นแฝง-ชั้นเอาต์พุตหลังทำการรวมโหนด

ชุดข้อมูล	โหนด	น้ำหนัก
Iris	Bias	1.000000
	H1	-2.183051
	H2	-1.664501
	H3	2.311932
Pima Indian Diabetes	Bias	1.000000
	H1	-3.763128
	H2	2.259591
Data Generated 1	Bias	1.000000
	H1	-2.641228
	H2	-2.200521
Data Generated 2	Bias	1.000000
	H1	-4.865601
	H2	-1.114029

โดย H คือโหนดแฝง (Hidden Node) และ ค่าไบแอสเป็น 1 คือการทำนอร์มอไลซ์เพื่อให้โหนดแต่ละโหนดสามารถเปรียบเทียบและหาค่าเฉลี่ยได้

4.1.4 การรวมโหนดในชั้นอินพุต-ชั้นแฝง

เมื่อปรับค่าน้ำหนักโหนดในชั้นแฝง-ชั้นเอาต์พุต จนได้ค่าน้ำหนักที่เหมาะสมในแล้ว คือ เมื่อทำการนำไปทดสอบในชุดข้อมูลตรวจสอบแล้ว ได้ค่าความผิดพลาดที่น้อยที่สุด จากนั้นนำเอาเน็ตเวิร์ก N_{best} ทำการรวมโหนดในชั้นอินพุต-ชั้นแฝง

ในการรวมโหนดในชั้นอินพุต-ชั้นแฝงนี้ ทำโดยการเปรียบเทียบค่าน้ำหนักของอินพุตทั้งหมดที่เข้ามาในโหนดแฝง 1 โหนด เช่น นิวรอลเน็ตเวิร์กของชุดข้อมูล Iris, Data Generated 1 และ Data Generated 2 ในโครงสร้างดังตารางที่ 7 นั้น 1 โหนดในชั้นแฝงจะมีน้ำหนักอินพุต (Input Weight) อยู่ 5 ค่า และ Pima Diabetes ใน 1 โหนดชั้นแฝงนั้น ก็จะมีน้ำหนักอินพุตอยู่ 9 ค่า

ในการเปรียบเทียบ เราจึงต้องทำการเปรียบเทียบทั้งชุด คือนำเอาชุดหนักในชั้นแฝงของโหนดแฝง N_{best} ไปเปรียบเทียบกับโหนดแฝงอื่นๆที่เหลือในเน็ตเวิร์กอื่นๆ จนครบ โดยในการเปรียบเทียบ จะทำตามสมการที่ 19 โดยการหาระยะยุคลิดเปรียบเทียบความคล้าย แล้วทำการรวมโหนด ซึ่งดังแสดงในตารางที่ 13-16

ตารางที่ 13 ค่าน้ำหนักในชั้นอินพุต-ชั้นแฝงหลังการรวมโหนดชุดข้อมูล Iris

ชั้นเน็ตเวิร์ก	โหนด	น้ำหนัก
อินพุต-ชั้นแฝง	B11	1.000000
	I12	2.821781
	I13	-0.550108
	I14	-1.887775
	I15	-1.180551
	B21	1.000000
	I22	1.475328
	I23	-0.095087
	I24	-1.604926
	I25	0.580981
	B31	1.000000
	I32	-4.671297

อินพุต-ชั้นแฝง	I33	0.988182
	I34	4.900480
	I35	3.384171
	B41	1.000000
ชั้นแฝง-เอาต์พุต	H42	-2.183051
	H43	-1.664501
	H44	2.311932

โดย B คือค่าไบแอส และ H คือโหนดแฝงและค่าความไบแอสเป็น 1 คือการทำนอร์มอไลซ์เพื่อให้โหนดแต่ละโหนดสามารถเปรียบเทียบและหาค่าเฉลี่ยได้ เมื่อทำการเปรียบเทียบโหนดแล้วพบว่าเมื่อเฉลี่ยค่าน้ำหนักและทดสอบการเรียนรู้กับชุดข้อมูลตรวจสอบแล้ว ค่าความผิดพลาดไม่ได้ลดลง ให้ทำการเพิ่มโหนดนั้นทั้งโหนดเข้ามาใน N_{best} ดังแสดงในตารางที่ 13 และเช่นเดียวกัน ในตารางที่ 14-16 แสดงค่าน้ำหนักในชั้นอินพุต-ชั้นแฝงหลังทำการรวมโหนดของชุดข้อมูล Pima Diabetes, Data Generated 1 และ Data Generated 2 ซึ่งเมื่อมีการเพิ่มโหนดเข้ามา ต้องเพิ่มในชั้นแฝง-ชั้นเอาต์พุตด้วย

ตารางที่ 14 ค่าน้ำหนักในชั้นอินพุต-ชั้นแฝงหลังการรวมโหนดชุดข้อมูล Pima Diabetes

ชั้นเน็ตเวิร์ก	โหนด	น้ำหนัก
ชั้นอินพุต-ชั้นแฝง	B11	1.000000
	I12	-0.136363
	I13	-0.282862
	I14	1.057702
	I15	-1.962678
	I16	-0.269965
	I17	-1.330596
	I18	-0.641948
	I19	2.270930
	B21	1.000000

ชั้นอินพุต-ชั้นแฝง	I22	1.208343
	I23	2.719802
	I24	0.391559
	I25	-2.005533
	I26	1.639231
	I27	2.526471
	I28	1.736348
	I29	-0.485968
ชั้นแฝง-ชั้นเอาต์พุต	B31	1.000000
	H32	-3.763128
	H33	2.259591

ตารางที่ 15 ค่าน้ำหนักในชั้นอินพุต-ชั้นแฝงหลังการรวมโหนดชุดข้อมูล Data Generated 1

ชั้นเน็ตเวิร์ก	โหนด	น้ำหนัก
ชั้นอินพุต-ชั้นแฝง	B11	1.000000
	I12	-0.066256
	I13	-0.002875
	I14	-0.002021
	I15	2.691606
	B21	1.000000
	I22	-1.877854
	I23	-0.012643
	I24	0.029055
	I25	0.287066
ชั้นแฝง-ชั้นเอาต์พุต	B31	1.000000
	I32	-2.641228

ชั้นแฝง-ชั้นเอาต์พุต	I33	-2.200521

ตารางที่ 16 ค่าน้ำหนักในชั้นอินพุต-ชั้นแฝงหลังการรวมโหนดชุดข้อมูล Data Generated 2

ชั้นเน็ตเวิร์ก	โหนด	น้ำหนัก
ชั้นอินพุต-ชั้นแฝง	B11	1.000000
	I12	-2.602518
	I13	-0.001238
	I14	0.004912
	I15	0.005639
	B21	1.000000
	I22	-0.002094
	I23	0.000315
	I24	0.000459
	I25	2.485246
ชั้นแฝง-ชั้นเอาต์พุต	B31	1.000000
	I32	-4.865601
	I33	-1.114029

โดยการรวมโหนดในตารางที่ 13, 14, 15 และ 16 จะทำให้ได้เน็ตเวิร์กโครงสร้างสุดท้าย โดยโครงสร้างนี้จะถูกนำไปทดสอบในแง่ความถูกต้องและในเชิงเวลาต่อไป

โดยโครงสร้างของเน็ตเวิร์กย่อยในแต่ละเน็ตเวิร์กจะแสดงรายละเอียดในภาคผนวก ค

4.2 ผลการทดลอง

จากการทดลอง 4.1 ผู้วิจัยได้นำเอาเน็ตเวิร์กใหม่ที่ได้จากการรวมโหนดมาทำการทดสอบกับชุดข้อมูลเพื่อเปรียบเทียบผลการทดลองดังต่อไปนี้

4.2.1 เปรียบเทียบความผิดพลาดในการเรียนรู้กับเน็ตเวิร์กตั้งต้น

เมื่อทำการรวมโหนดจนได้เน็ตเวิร์กใหม่แล้ว นำเน็ตเวิร์กใหม่มาทดสอบกับข้อมูลทดสอบแบบไม่ได้แบ่ง ดังตารางที่ 17

ตารางที่ 17 การเปรียบเทียบความผิดพลาดการเรียนรู้

ชุดข้อมูล	ค่าความผิดพลาดรากกำลังสองเฉลี่ยของเน็ตเวิร์กเดิม (N_0)	ค่าความผิดพลาดรากกำลังสองเฉลี่ยของเน็ตเวิร์กที่เลือก (N_{best})	ค่าความผิดพลาดรากกำลังสองเฉลี่ยของเน็ตเวิร์กปรับปรุง (Optimized Network N_{best})
Iris	0.002548	0.002498	0.002462
Pima Diabetes	0.194147	0.141457	0.129986
Data Generate 1	0.045129	0.038792	0.038781
Data Generate 2	0.058531	0.043775	0.043602

4.2.2 เปรียบเทียบความถูกต้องในการเรียนรู้กับเน็ตเวิร์กตั้งต้น

ในการเรียนรู้เพื่อเปรียบเทียบความถูกต้องในการเรียนรู้ของข้อมูล รายละเอียดจะแสดงในภาคผนวก ง ซึ่งผลของการเปรียบเทียบความถูกต้องดังแสดงในตารางที่ 18

ตารางที่ 18 การเปรียบเทียบความถูกต้องในการเรียนรู้

	เปอร์เซ็นความถูกต้องในการเรียนรู้			
ชุดข้อมูล	เน็ตเวิร์กเดิม (N_0) ทดสอบใน WEKA	เน็ตเวิร์กเดิม (N_0)	เน็ตเวิร์กที่เลือก (N_{best})	เน็ตเวิร์กปรับปรุง (Optimized Network N_{best})
Iris	96.93	98.32	98.22	98.76
Pima Diabetes	74.75	79.04	80.44	80.64
Data Generated 1	98.99	99.13	99.15	99.65
Data Generated 2	99.22	97.79	99.07	99.08

4.2.3 **เปรียบเทียบเวลาที่ใช้ในการเรียนรู้กับเน็ตเวิร์กตั้งต้น**

ตารางที่ 19 จะแสดงการเปรียบเทียบในแง่ของเวลาที่ใช้ในการเรียนรู้กับข้อมูลชุดข้อมูล

ตารางที่ 19 การเปรียบเทียบเวลาที่ใช้ในการเรียนรู้

	เวลาที่ใช้ในการเรียนรู้ (วินาที)	
ชุดข้อมูล	แบ็กพรอพาเกชันนิวรอลเน็ตเวิร์ก (N_0)	เน็ตเวิร์กปรับปรุงบนแบ็กพรอพาเกชัน (Optimized Network N_{best})
Iris	10	6
Pima Diabetes	23	9
Data Generated 1	227	45
Data Generated 2	1,285	314

4.2.3 **เปรียบเทียบระดับนัยสำคัญ**

ในการตรวจสอบระดับนัยสำคัญ จากข้อมูลความถูกต้องตารางที่ 18 นำมาหาระดับนัยสำคัญโดยวิธีทดสอบความแตกต่างแบบจับคู่ (Paired T-Test) โดยสมมุติฐานว่าวิธีการปรับปรุงเน็ตเวิร์กด้วยวิธีรวมโหนดชั้นแฝงบนแบ็กพรอพาเกชัน ให้ผลการเรียนรู้ที่ดีกว่าแบบแบ็กพรอพาเกชันปกติหรือไม่

ตารางที่ 20 เปอร์เซ็นความถูกต้องของข้อมูลทดสอบ

	เปอร์เซ็นความถูกต้อง	
ชุดข้อมูล	เน็ตเวิร์กที่ปรับปรุงด้วยวิธีรวมโหนด	เน็ตเวิร์กแบ็กพรอพาเกชันปกติ
Iris	98.76	98.32
Pima Diabetes	80.64	79.04
Data Generated 1	99.65	99.13
Data Generated 2	99.08	97.79

ในการวัดสมมุติฐานจากตารางที่ 20 โดยกำหนดค่าอัลฟา $\alpha = 0.05$ จากผลการทดสอบความแตกต่างแบบจับคู่ จะแสดงตามภาพที่ 10

t-Test: Paired Two Sample for Means		
	98.76	*98.32*
Mean	93.12333	91.98667
Variance	116.9564	126.161
Observations	3	3
Pearson Correlation	0.999445	
Hypothesized Mean Difference	0	
df	2	
t Stat	3.540386	
P(T<=t) one-tail	0.035674	
t Critical one-tail	2.919986	
P(T<=t) two-tail	0.071348	
t Critical two-tail	4.302653	

ภาพที่ 8 ผลการทดสอบสมมุติฐานด้วยวิธีการทดสอบความแตกต่างแบบจับคู่

ซึ่งจากภาพที่ 10 เป็นผลการทดสอบจากข้อมูลความถูกต้องในตารางที่ 20 จะได้ผลคือ พิจารณาค่า P-Value [P(T<=t) two-tail] = 0.071348 มีค่ามากกว่า 0.05 ดังนั้นจึงยอมรับสมมติฐานหลักคือวิธีการวิเคราะห์เรียนรู้ด้วยวิธีรวมโหนดชั้นแฝงบนแบ็กพรอพาเกชันดีกว่าแบ็กพรอพาเกชันแบบปกติโดยมีนัยสำคัญทางสถิติ 0.05

บทที่ 5
สรุปผลการวิจัยและข้อเสนอแนะ

5.1 สรุปผลการวิจัย

จากบทที่ 4 แสดงผลการทดลองในการเรียนรู้ของนิวรอลเน็ตเวิร์กตั้งต้น (ก่อนการแบ่งข้อมูล) และนิวรอลเน็ตเวิร์กที่ผ่านการรวมโหนดแล้ว ในตารางที่ 12 สำหรับชั้นแฝง-ชั้นเอาต์พุต และ ตารางที่ 13-16 สำหรับชันอินพุต-ชั้นแฝง ซึ่งแสดงให้เห็นว่า หลังจากการรวมโหนดนิวรอนแล้ว ค่าความผิดพลาดรากที่สองกำลังสองเฉลี่ยในการเรียนรู้ของเน็ตเวิร์กที่รวมโหนดจะมีค่าลดน้อยลงมากกว่าก่อนการรวมโหนด และเปอร์เซ็นความถูกต้องในการเรียนรู้ของเน็ตเวิร์กที่รวมโหนด มีค่าที่สูงกว่าเล็กน้อย ดังตารางที่ 17 และ 18 และให้ค่าความถูกต้องมากกว่าเมื่อเทียบกับการเรียนรู้ในโปรแกรมเวกา

ในแง่ของเวลาในการเรียนรู้ นิวรอลเน็ตเวิร์กหลังการรวมโหนดในชั้นแฝงใช้เวลาโดยรวมในการเรียนรู้น้อยกว่านิวรอลเน็ตเวิร์กที่ทำการเรียนรู้แบบเก่ามาก ดังแสดงในตารางที่ 19 ซึ่งในการเรียนรู้ของนิวรอลเน็ตเวิร์กหลังการรวมโหนดในชั้นแฝงจะทำการเรียนรู้บนข้อมูลตรวจสอบที่เป็นตัวแทนของข้อมูลทั้งชุดและค่าน้ำหนักที่ได้จากการปรับปรุงเป็นน้ำหนักที่ผ่านการฝึกฝนมาจากข้อมูลย่อยทั้ง 10 ชุด เวลาในการเรียนรู้จึงน้อยกว่าการเรียนรู้แบบปกติในขณะที่ความถูกต้องใกล้เคียงหรือดีกว่า

เมื่อทำการนำเอานิวรอลเน็ตเวิร์กมาเรียนรู้ชุดข้อมูลด้วยเทคนิคการแบ่งชุดข้อมูลออกเป็นกลุ่มย่อยๆแล้วทำการสร้างนิวรอลเน็ตเวิร์กที่มีโครงสร้างเหมือนกันมาเรียนรู้ให้กับชุดข้อมูลที่แบ่งออกมาแล้ว และคัดเลือกเน็ตเวิร์กที่ดีที่สุดมาทำการรวมโหนดกับเน็ตเวิร์กอื่นๆที่ใกล้เคียงกัน จะให้ผลความผิดพลาดที่น้อยลงเมื่อเปรียบเทียบกับการเรียนรู้ชุดข้อมูลที่ไม่ทำการแบ่ง และเมื่อเปรียบเทียบความถูกต้องในการเรียนรู้กับวิธีการอื่นเช่น เวกา จะให้ผลที่ดีกว่า และ iHWNN จากตารางที่ 1 Iris จะให้ผลความถูกต้องที่สูงกว่า และ Pima Diabetes ให้ความถูกต้องที่ใกล้เคียงกัน และในเชิงเวลาที่ใช้ก็สามารถที่มีอัตราการลู่เข้าที่ดี ซึ่งสามารถนำเอาเทคนิคนี้ไปประยุกต์ใช้กับข้อมูลขนาดใหญ่อย่างมีประสิทธิภาพมากขึ้น

ในการทดสอบความแตกต่างแบบจับคู่เพื่อวัดระดับนัยสำคัญเชิงสถิติดัง ตารางที่ 20 และผลดังภาพที่ 8 โดยตั้งระดับนัยสำคัญ 0.05 จะพบว่าการแยกแยะข้อมูลด้วยนิวรอลเน็ตเวิร์กหลังการรวมโหนดในชั้นแฝงคัดแยกได้ดีกว่าแบ็กพรอพาเกชันปกติ

5.2 ข้อเสนอแนะสำหรับการทำวิจัยต่อ

งานวิจัยนี้ เป็นการนำเสนอตัวแบบที่ใช้หลักการวิเคราะห์ในการสร้างอัลกอริทึมโดยใช้ข้อมูลตัวอย่างขนาดน้อยๆจึงสามารถฝึกฝนเพิ่มเติมข้อมูลให้กับระบบได้อีก อีกทั้งอัลกอริทึมการรวมโหนดยังสามารถใช้อัลกอริทึมอื่นๆมาผสมประสานหรือประยุกต์ได้อีกมาก ทั้งนี้สามารถเป็นประเด็นที่สามารถจะวิจัยต่อไปได้

ในการทำการวิเคราะห์ในการวิจัยนี้จะทำการวิเคราะห์โดยใช้โปรแกรมประยุกต์เอกเซลมาช่วยในการคำนวณและวิเคราะห์ ซึ่งสามารถนำไปประยุกต์และพัฒนาเป็นโปรแกรมประยุกต์เพื่อง่ายต่อการวิเคราะห์และรวมโหนดให้อัตโนมัติมากขึ้นกว่านี้

นอกจากนี้ในการวิจัยนี้ยังไม่ได้ทำการทดสอบและเปรียบเทียบอัลกอริทึมนี้กับวิธีอื่นๆ ในแง่ของการวัดความคล้ายที่นอกเหนือจากการใช้ระยะยุคลิด ทั้งนี้เพื่อทำการปรับปรุงและประยุกต์เพื่อเป็นแนวทางในการวิจัยเพื่อให้ได้อัลกอริทึมที่สามารถเรียนรู้ข้อมูลขนาดใหญ่ได้อย่างมีประสิทธิภาพ

รายการอ้างอิง

[1] Zhao, Y.,Gao J.,and Yang, X. A Survey of neural network ensembles, Proceedings 2005 International Conference on Neural Networks & Brain(ICNN&B'05), pp.438-442. IEEE Press, 2005.

[2] Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P.,and Witten, I.H. The WEKA Data Mining Software: An Update. SIGKDD Explorations, Volume 11, Issue 1, 2009.

[3] Shibata, K.,and Ikeda, Y. Effect of number of hidden neurons on learning in large-scale layered neural networks, ICROS-SICE International Joint Conference, pp.5008-5013. IEEE Press, 2009.

[4] Blake, C.,and Merz, C. UCI repository of machine learning databases [Online]. 1998. Available from : http://www.ics.uci.edu/~mlearn/MLRepository.html [2012 , May 05]

[5] Witten, I.H.,and Frank, E. Data Mining Practical Machine Learning Tools And Techniques Second Edition. San Francisco, California, USA: Morgan Kaufmann Publishers, 2005.

[6] Lopes, N.,and Ribeiro, B. GPU Implementation of the Multiple Back-Propagation Algorithm, In Proceedings of Intelligent Data Engineering and Automated Learning, pp.449-456. LNCS, Springer-Verlag, 2009.

[7] Lopes, N.,and Ribeiro, B. Fast pattern classification of ventricular arrhythmias using graphics processing units, Proceedings of 14th Iberoamerican Congress on Pattern Recognition, pp.603-610. Springer Press, 2009.

[8] Rojas, R. Neural Networks: A Systematic Introduction 1st Edition. USA:Springer, 1996.

[9] Jean, J.S. A New Distance Measure For Binary Images, ICASSP '90 - 1990 IEEE International Conference on Acoustics, Speech and Signal Processing, pp.2061-2064. IEEE Press, 1990.

[10] Owens, A.J. Empirical Modeling of Very Large Data Sets Using Neural Networks, Proceedings of the Ieee-Inns-Enns International Joint Conference on Neural Networks, pp.302-307. IEEE Press, 2000.

[11] Liang, B.,and Austin, J. A neural network for mining large volumes of time series data, IEEE International Conference on Industrial Technology (ICIT'05), pp.688-693. IEEE Press, 2005.

[12] Ivanikovas, S., Dzemyda, G.,and Medvedev, V. Large Datasets Visualization with Neural Network Using Clustered Training Data. In Hannu Jaakkola et al. (eds.),12th East European Conference ADBIS 2008, pp.143-152. Berlin:Springer-Verlag, 2008.

[13] Zhao, Y.Z., Vuong, N.K.,and Li, X. A Novel Hybrid Wavelet Neural Network and Its Performance Evaluation, The IEEE International Converence on Industrial Informatics (INDIN 2008) DCC, pp.1097-1102. Daejeon, Korea, 2008.

[14] Wang, G.J.,and Chen, C.C. A Fast Multilayer Neural-Network Training Algorithm Based on the Layer-By-Layer Optimizing Procedures. IEEE Transactions on Neural Networks Vol. 7. No.3 (1996): 768-775.

[15] Yang, B., Su, X.H.,and Wang Y.D. BP Neural Network Optimization Based On An Improved Genetic Algorithm, Proceedings of the First International Conference on Machine Learning and Cybernetics, pp.64-68. Beijing, 2002.

[16] Li, Y., Fu, Y., Li, H.,and Zhang, S.W. The Improved Training Algorithm of Back Propagation Neural Network with Self-adaptive Learning Rate, International Conference on Computational Intelligence and Natural Computing, pp.73-76. IEEE Press, 2009.

[17] ธเนศ ปานรัตน์. แบบจำลองการวิเคราะห์รูปแบบและรู้จำลำดับพันธุกรรม. วิทยานิพนธ์ปริญญามหาบัณฑิต, สาขาวิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ มหาวิทยาลัยสงขลานครินทร์, 2551.

[18] Microsoft Corperation. Microsoft SQL Server 2005 Books Online [Computer Program]. 2006. Available from : Microsoft SQL Server 2005 Documentation and Tutorials [2012, March 22]

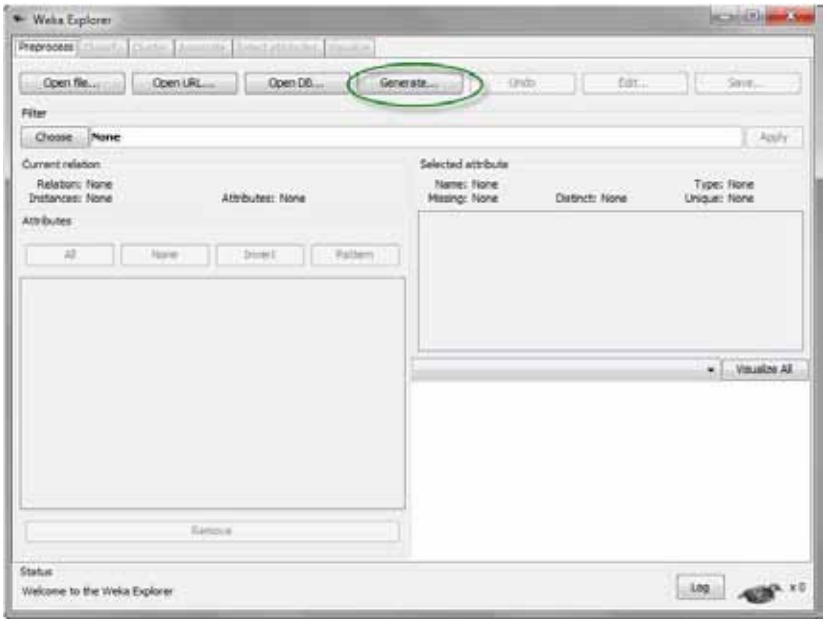
ภาคผนวก

ภาคผนวก ก
การเตรียมชุดข้อมูลจากโปรแกรมเวกา

ในการเตรียมข้อมูลทดสอบจากโปรแกรมเวกา จะทำการสร้างข้อมูลทดสอบขึ้นมาจากฟังก์ชันเวกาเอ็กพลอเรอร์ (Weka Explorer) ดังแสดงในภาพที่ 9

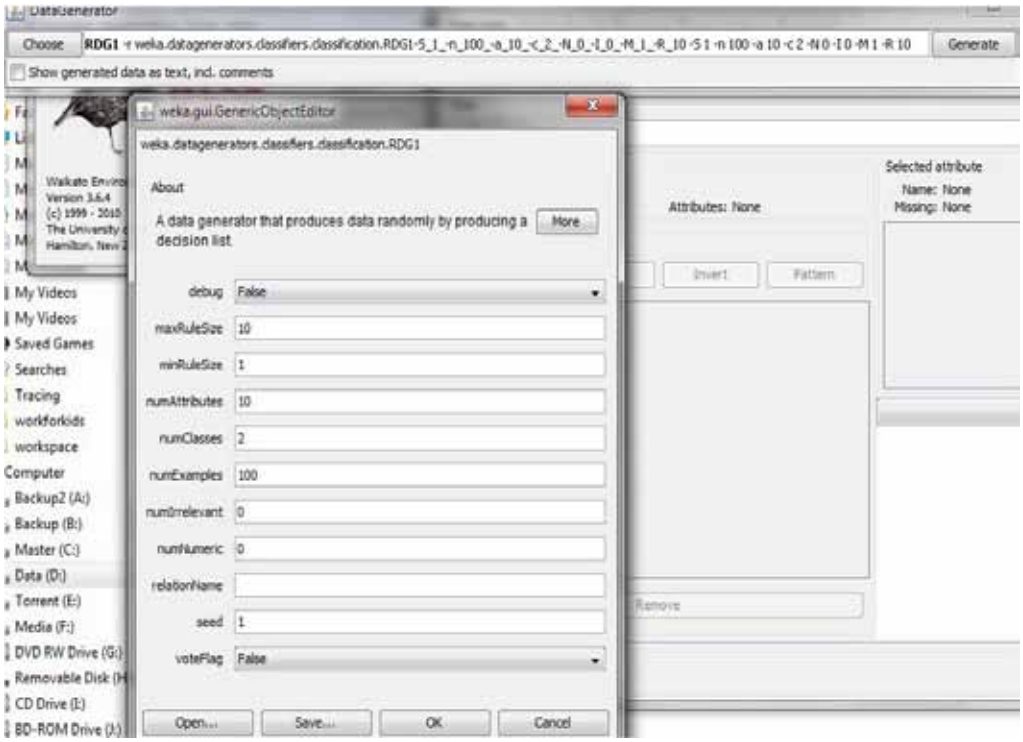
ภาพที่ 9 โปรแกรมเวกาเอ็กพลอเรอร์สำหรับเตรียมข้อมูลทดสอบ

ในโปรแกรมเวกาเอ็กพลอเรอร์มีฟังก์ชันให้ทำการสร้างข้อมูลทดสอบ โดยการกดปุ่ม “Generate…” ดังภาพที่ 10



ภาพที่ 10 ฟังก์ชันสร้างข้อมูลทดสอบในโปรแกรมเวกาเอ็กพลอเรอร์

ในการสร้างข้อมูลทดสอบในโปรแกรมเวกาเอ็กพลอเรอร์ จะใช้ฟังก์ชันการคัดแยกแบบ RDG1 ดังภาพที่ 11



ภาพที่ 11 การกำหนดค่าจำเพาะในฟังก์ชันสร้างข้อมูลทดสอบ

ในการกำหนดค่าจำเพาะในการสร้างข้อมูลทดสอบ 3 และ 4 เป็นไปตามตารางที่ 21

ตารางที่ 21 ค่าจำเพาะในการสร้างข้อมูลทดสอบ 3 และ 4

	ค่าจำเพาะในโปรแกรมเวกาเอ็กพลอเรอร์			
ชุดข้อมูล	numAttributes	numClasses	numExamples	numNumeric
Data Generated 1	4	2	10,000	4
Data Generated 2	4	2	100,000	4

ภาคผนวก ข
แอสคิวแอลในการแบ่งข้อมูลตรวจสอบและการส่งออก

คำสั่งเอสคิวแอลสำหรับการแบ่งข้อมูลเพื่อเป็นข้อมูลตรวจสอบดังภาพที่ 12 โดยชุดคำสั่งนี้จะแบ่งข้อมูลทดสอบไว้ที่ 20% ดังตัวแปร @pvs=20

```
-- Declare Percent for Validation Set
declare @pvs decimal(16,6)
set @pvs=20
set @pvs=@pvs/100
--Set Output
declare @tab table (RowID int,SepalLen decimal(8,2),SepalWid decimal(8,2),PetalLen decimal(8,2),PetalWid decimal(8,2),Class varchar(50))
--Count Record
declare @rec int
select @rec=COUNT(*) From DSIrisVS
--Validation Set Record
declare @vsnum int
select @vsnum=@pvs * @rec
--Number of Class
declare @clsnum int
select @clsnum=COUNT(distinct Class) FROM DSIrisVS
--Select data
declare @class varchar(50)
declare @i int
set @i=1

declare c cursor for
  Select distinct class from DSIrisVS
open c
fetch next from c into @class
while @@FETCH_STATUS=0
begin
  while @i<=@vsnum/@clsnum
  begin
    insert into @tab
    select top 1 * from DSIrisVS where Class=@class and RowID not in (Select RowID from @tab)
    set @i=@i+1
  end
  set @i=1
  fetch next from c into @class
end
close c
deallocate c
--Return Result Table
select * from @tab
```

ภาพที่ 12 เอสคิวแอลในการดึงข้อมูลเพื่อทำเป็นข้อมูลตรวจสอบ

เมื่อทำการแบ่งข้อมูลออกมาแล้วจะได้ผลลัพธ์ดังภาพที่ 13 ซึ่งเราสามารถนำเอาข้อมูลนี้ส่งออกไปในรูปแบบไฟล์ CSV ได้ดังแสดงในภาพที่ 14-19

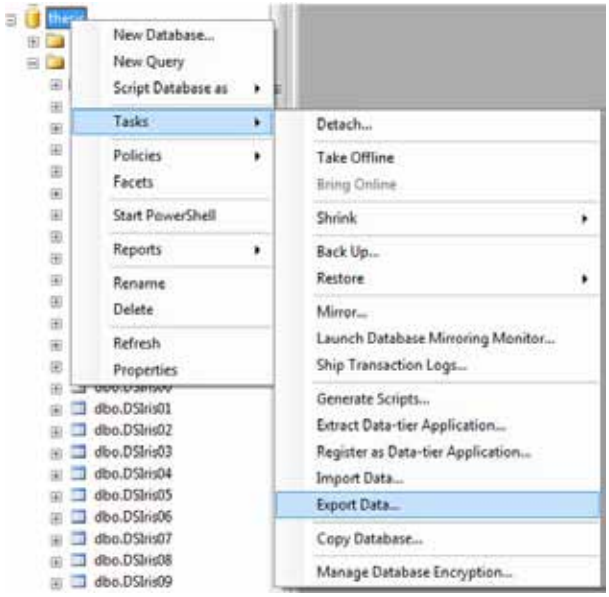
	RowID	SepalLen	SepalWid	PetalLen	PetalWid	Class
1	1	5.10	3.50	1.40	0.20	Iris-setosa
2	2	4.90	3.00	1.40	0.20	Iris-setosa
3	3	4.70	3.20	1.30	0.20	Iris-setosa
4	4	4.60	3.10	1.50	0.20	Iris-setosa
5	5	5.00	3.60	1.40	0.20	Iris-setosa
6	16	5.40	3.90	1.70	0.40	Iris-setosa
7	17	4.60	3.40	1.40	0.30	Iris-setosa
8	18	5.00	3.40	1.50	0.20	Iris-setosa
9	19	4.40	2.90	1.40	0.20	Iris-setosa
10	20	4.90	3.10	1.50	0.10	Iris-setosa
11	6	7.00	3.20	4.70	1.40	Iris-versicolor
12	7	6.40	3.20	4.50	1.50	Iris-versicolor
13	8	6.90	3.10	4.90	1.50	Iris-versicolor
14	9	5.50	2.30	4.00	1.30	Iris-versicolor
15	10	6.50	2.80	4.60	1.50	Iris-versicolor
16	21	5.70	2.80	4.50	1.30	Iris-versicolor
17	22	6.30	3.30	4.70	1.60	Iris-versicolor
18	23	4.90	2.40	3.30	1.00	Iris-versicolor
19	24	6.60	2.90	4.60	1.30	Iris-versicolor
20	25	5.20	2.70	3.90	1.40	Iris-versicolor
21	11	6.30	3.30	6.00	2.50	Iris-virginica
22	12	5.80	2.70	5.10	1.90	Iris-virginica
23	13	7.10	3.00	5.90	2.10	Iris-virginica
24	14	6.30	2.90	5.60	1.80	Iris-virginica
25	15	6.50	3.00	5.80	2.20	Iris-virginica
26	26	7.60	3.00	6.60	2.10	Iris-virginica

ภาพที่ 13 ผลลัพธ์ตัวอย่างของข้อมูลตรวจสอบ

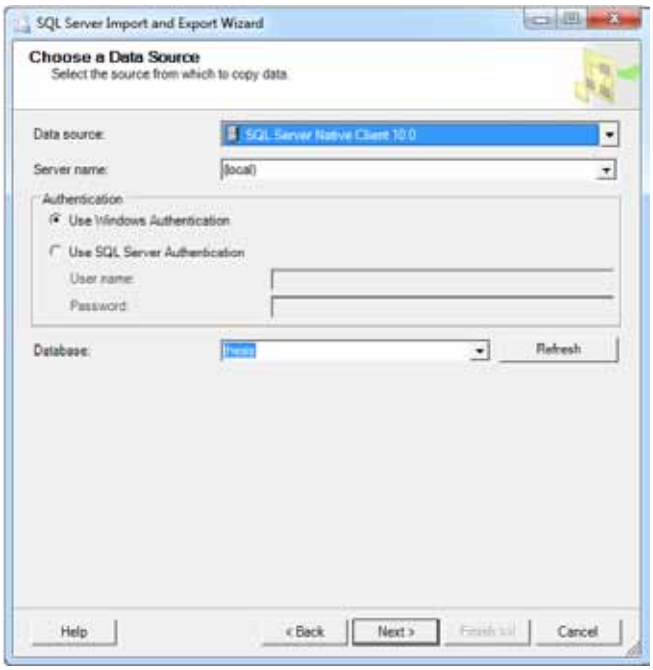
การนำข้อมูลในตารางที่แบ่งแล้วตามคำสั่งเอสคิวแอลในภาพที่ 12 ในการส่งออกข้อมูลให้อยู่ในรูปแบบไฟล์ CSV ทำได้โดยมีขั้นตอนดังต่อไปนี้

- เลือกฐานข้อมูลที่ต้องการส่งข้อมูลออกแล้วคลิ๊กขวาเลือก Task แล้วเลือก Export ดังภาพที่ 14
- เลือกแหล่งของข้อมูลดังภาพที่ 15
- เลือกชนิดของข้อมูลที่จะส่งออกเป็นไฟล์ CSV ดังภาพที่ 16
- เลือกการคัดลอกข้อมูลจากตารางดังภาพที่ 17
- เลือกตารางที่ต้องการจะส่งข้อมูลออก ดังภาพที่ 18

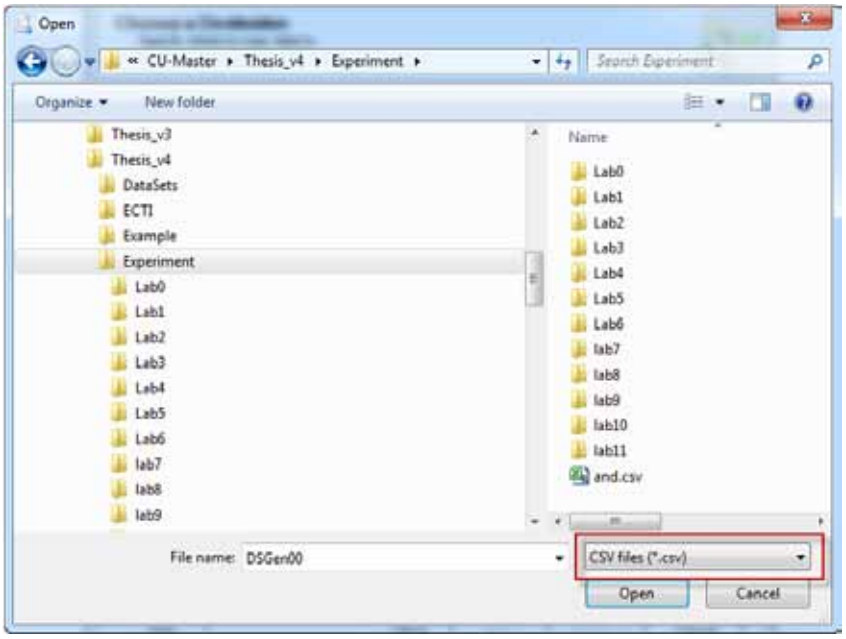
- ส่งข้อมูลออก ดังภาพที่ 19 จะแสดงผลลัพธ์การส่งข้อมูลออก



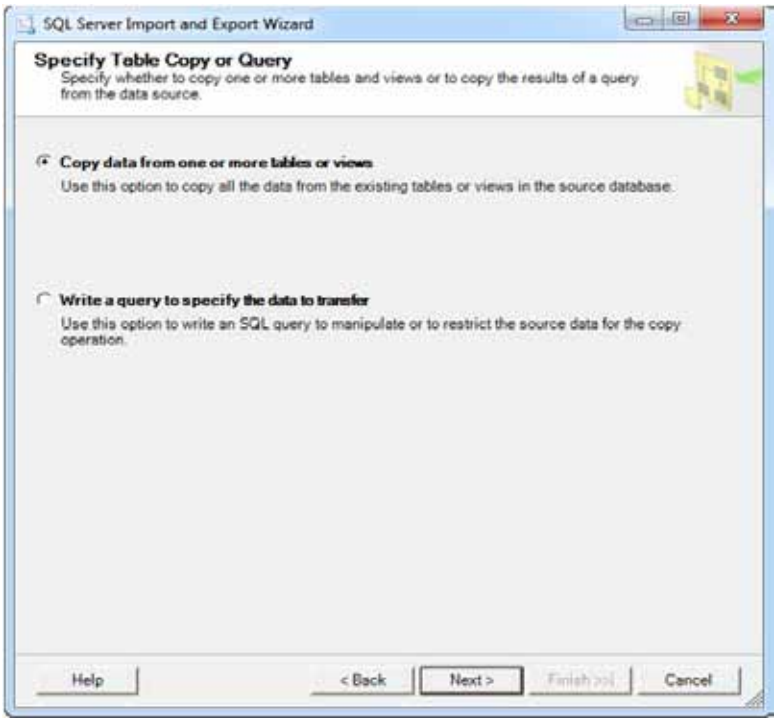
ภาพที่ 14 การส่งข้อมูลออกจากฐานข้อมูลเอสคิวแอล



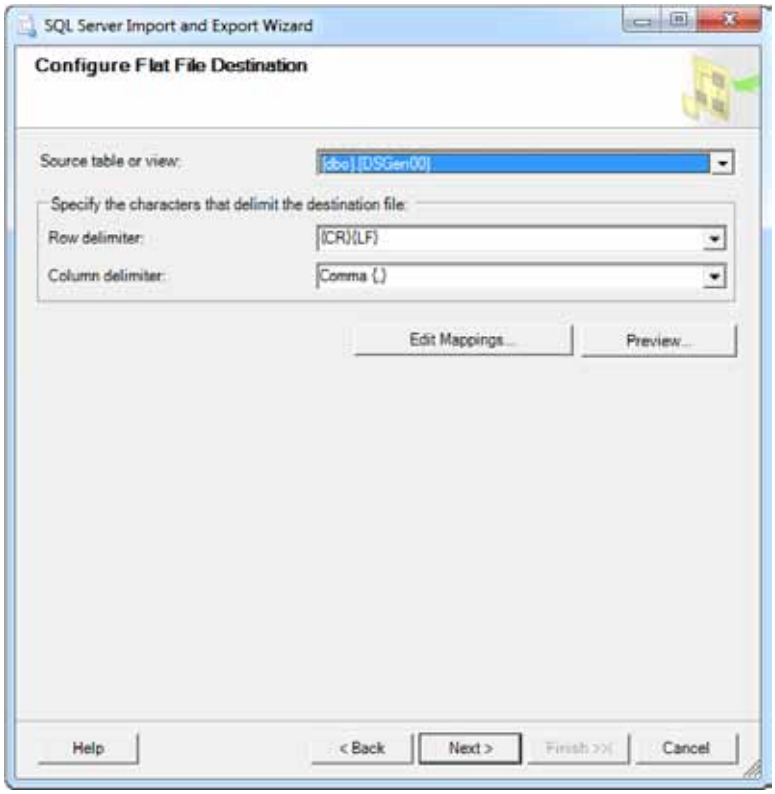
ภาพที่ 15 เลือกแหล่งข้อมูลเอสคิวแอลที่ต้องส่งออก



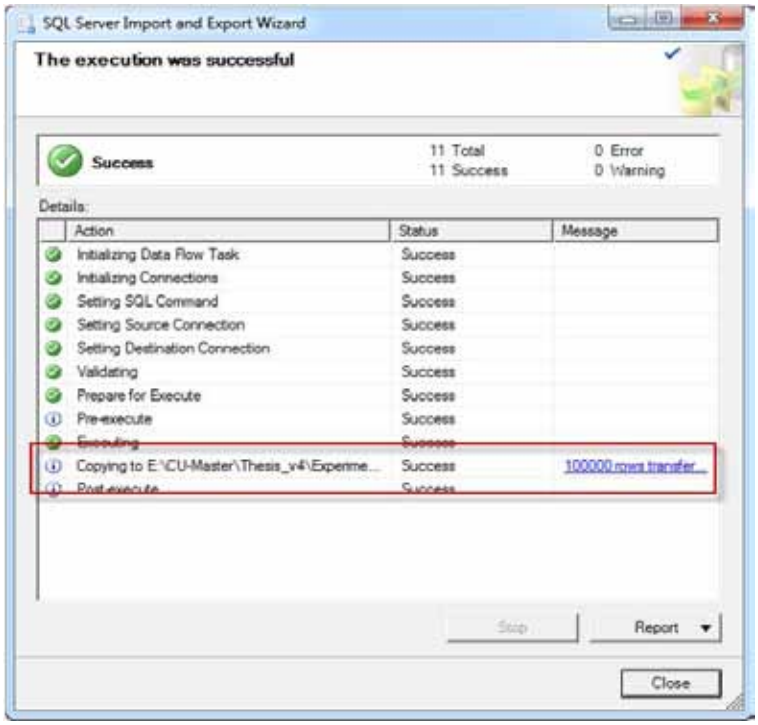
ภาพที่ 16 เลือกไฟล์ CSV ที่ต้องการส่งข้อมูลออก



ภาพที่ 17 เลือกชนิดการส่งออกเป็นแบบคัดลอกจากตารางข้อมูล



ภาพที่ 18 เลือกตารางข้อมูลที่ต้องการส่งออกไปเป็นไฟล์ CSV



ภาพที่ 19 ผลลัพธ์ของการส่งออกข้อมูลจากเอสคิวแอล

ภาคผนวก ค
ค่าน้ำหนักของเน็ตเวิร์กย่อย

		from the input layer			
to the 1th hidden layer	bias	1th neuron	2th neuron	3th neuron	4th neuron
1th neuron	-4.77776	-0.8797	2.92246	-5.89427	-4.53909
2th neuron	-3.86033	-2.46756	1.01265	-7.6656	0.561717
3th neuron	19.7036	7.41066	8.11201	-45.0559	-19.3153

		from the 1th hidden layer		
to the output layer	bias	1th neuron	2th neuron	3th neuron
1th neuron	-0.0284431	14.1011	9.13461	-16.4358

ภาพที่ 20 ค่าน้ำหนักของเน็ตเวิร์กตั้งต้น N_0 ของ Iris

		from the input layer			
to the 1th hidden layer	bias	1th neuron	2th neuron	3th neuron	4th neuron
1th neuron	-8.45514	-1.41881	-4.17331	13.5892	10.376
2th neuron	-4.50741	-3.64682	4.72265	-2.61189	-2.87295
3th neuron	3.82547	-2.28273	-0.0978376	7.34205	0.916672

		from the 1th hidden layer		
to the output layer	bias	1th neuron	2th neuron	3th neuron
1th neuron	-4.48748	6.41005	11.5079	-1.90284

ภาพที่ 21 ค่าน้ำหนักของเน็ตเวิร์กคัดเลือก N_{best} ของ Iris

		from the input layer			
to the 1th hidden layer	bias	1th neuron	2th neuron	3th neuron	4th neuron
1th neuron	3.82547	-2.28273	-0.0978376	7.34205	0.916672
2th neuron	-4.50741	-3.64682	4.72265	-2.61189	-2.87295
3th neuron	-8.45514	-1.41881	-4.17331	13.5892	10.376

		from the 1th hidden layer		
to the output layer	bias	1th neuron	2th neuron	3th neuron
1th neuron	-4.48748	-1.90284	11.5079	6.41005

ภาพที่ 22 ค่าน้ำหนักของเน็ตเวิร์กที่รวมโหนด N_{best} ของ Iris

		from the input layer							
to the 1th hidden layer	bias	1th neuron	2th neuron	3th neuron	4th neuron	5th neuron	6th neuron	7th neuron	8th neuron
1th neuron	-69.7076	-22.9877	-54.4294	2.01932	-20.4803	87.301	-17.0411	-33.5242	-115.261
2th neuron	-49.8424	-23.1781	-93.7712	39.9318	20.013	-54.3099	-109.187	-55.9403	-4.15831

		from the 1th hidden layer	
to the output layer	bias	1th neuron	2th neuron
1th neuron	-1.57046	2.53898	2.57808

ภาพที่ 23 ค่าน้ำหนักของเน็ตเวิร์กตั้งต้น N_0 ของ Pima Diabetes

		from the input layer							
to the 1th hidden layer	bias	1th neuron	2th neuron	3th neuron	4th neuron	5th neuron	6th neuron	7th neuron	8th neuron
1th neuron	23.5823	-0.924503	-50.3698	11.6616	14.6311	15.5237	-59.6469	-31.5889	0.0997795
2th neuron	-17.4925	25.8835	4.54594	19.043	-6.74557	4.65798	19.0998	-5.81798	-7.62705

		from the 1th hidden layer	
to the output layer	bias	1th neuron	2th neuron
1th neuron	-1.249	40.2643	-44.9274

ภาพที่ 24 ค่าน้ำหนักของเน็ตเวิร์กคัดเลือก N_{best} ของ Pima Diabetes

		from the input layer							
to the 1th hidden layer	bias	1th neuron	2th neuron	3th neuron	4th neuron	5th neuron	6th neuron	7th neuron	8th neuron
1th neuron	-13.4427	13.3564	59.77	-18.6211	-14.93	-16.7356	60.6629	34.2647	10.8584
2th neuron	9.33639	1.01934	24.5459	8.164	-11.414	13.0044	24.2786	22.3231	-19.9293

		from the 1th hidden layer	
to the output layer	bias	1th neuron	2th neuron
1th neuron	14.896	-4.82491	-12.9934

ภาพที่ 25 ค่าน้ำหนักของเน็ตเวิร์กที่รวมโหนด N_{best} ของ Pima Diabetes

		from the input layer			
to the 1th hidden layer	bias	1th neuron	2th neuron	3th neuron	4th neuron
1th neuron	29.5299	-0.118314	-0.0248552	-0.0479085	75.5278
2th neuron	-81.6065	210.28	-0.0635425	-0.403858	-2.57216

		from the 1th hidden layer	
to the output layer	bias	1th neuron	2th neuron
1th neuron	19.1436	-111.121	-22.3066

ภาพที่ 26 ค่าน้ำหนักของเน็ตเวิร์กตั้งต้น N_0 ของ Data Generated 1

		from the input layer			
to the 1th hidden layer	bias	1th neuron	2th neuron	3th neuron	4th neuron
1th neuron	14.2562	-0.60717	0.0226712	-0.025521	36.0803
2th neuron	-24.2386	55.9344	0.383332	-1.91563	-3.36323

		from the 1th hidden layer	
to the output layer	bias	1th neuron	2th neuron
1th neuron	33.1641	-88.5008	-62.1081

ภาพที่ 27 ค่าน้ำหนักในเน็ตเวิร์กคัดเลือก N_{best} ของ Data Generated 1

		from the input layer			
to the 1th hidden layer	bias	1th neuron	2th neuron	3th neuron	4th neuron
1th neuron	-25.0504	58.106	0.405403	-1.99299	-3.44882
2th neuron	14.5188	-0.605046	0.024261	-0.0267353	36.6502

		from the 1th hidden layer	
to the output layer	bias	1th neuron	2th neuron
1th neuron	33.1777	-59.4206	-87.1338

ภาพที่ 28 ค่าน้ำหนักของเน็ตเวิร์กที่รวมโหนด N_{best} ของ Data Generated 1

		from the input layer			
to the 1th hidden layer	bias	1th neuron	2th neuron	3th neuron	4th neuron
1th neuron	29.5299	-0.118314	-0.0248552	-0.0479085	75.5278
2th neuron	-81.6065	210.28	-0.0635425	-0.403858	-2.57216

		from the 1th hidden layer	
to the output layer	bias	1th neuron	2th neuron
1th neuron	19.1436	-111.121	-22.3066

ภาพที่ 29 ค่าน้ำหนักของเน็ตเวิร์กตั้งต้น N_0 ของ Data Generated 2

		from the input layer			
to the 1th hidden layer	bias	1th neuron	2th neuron	3th neuron	4th neuron
1th neuron	14.2562	-0.60717	0.0226712	-0.025521	36.0803
2th neuron	-24.2386	55.9344	0.383332	-1.91563	-3.36323

		from the 1th hidden layer	
to the output layer	bias	1th neuron	2th neuron
1th neuron	33.1641	-88.5008	-62.1081

ภาพที่ 30 ค่าน้ำหนักของเน็ตเวิร์กคัดเลือก N_{best} ของ Data Generated 2

		from the input layer			
to the 1th hidden layer	bias	1th neuron	2th neuron	3th neuron	4th neuron
1th neuron	-25.0504	58.106	0.405403	-1.99299	-3.44882
2th neuron	14.5188	-0.605046	0.024261	-0.0267353	36.6502

		from the 1th hidden layer	
to the output layer	bias	1th neuron	2th neuron
1th neuron	33.1777	-59.4206	-87.1338

ภาพที่ 31 ค่าน้ำหนักของเน็ตเวิร์กที่รวมโหนด N_{best} ของ Data Generated 2

ภาคผนวก ง
การเปรียบเทียบระยะยุคลิดในชั้นแฝง

ตารางที่ 22 เปรียบเทียบระยะยุคลิดในชั้นแฝงของชุดข้อมูล Iris

โหนด	ระยะยุคลิดจากค่าน้ำหนักระหว่างโหนด		
	โหนดแฝงที่ 1	โหนดแฝงที่ 2	โหนดแฝงที่ 3
1	2.508767112	3.638518249	7.077410
2	0.907406145	2.037157282	5.476049
3	9.70433698	8.574585843	5.135694
4	0.136879364	0.992871773	4.431764
5	0.961372056	0.168379081	3.607271
6	3.946435129	2.816683992	0.622208
7	2.522784865	3.652536002	7.091428
8	1.452891421	2.582642558	6.021534
9	10.14238224	9.012631098	5.573739
10	6.858462133	7.98821327	11.427105
11	0.846221772	0.283529365	3.722421
12	10.86036303	9.730611894	6.291720
13	0.889978764	2.019729901	5.458622
14	0.059427602	1.189178739	4.628071
15	6.938433179	5.808682042	2.369790
16	139.9204908	141.0502419	144.489134
17	73.91377932	75.04353045	78.482422
18	102.6694674	101.5397163	98.100825
19	0.377250984	1.507002121	4.945894
20	0.447498382	0.682252755	4.121145
21	5.222874272	4.093123135	0.654231
22	2.386589105	3.516340242	6.955232
23	0.649136943	0.480614194	3.919506

24	6.278062566	5.148311429	1.709420
25	15.40905542	16.53880655	19.977698
26	4.548575382	5.678326519	9.117218
27	18.94902186	17.81927072	14.380379

ตารางที่ 23 เปรียบเทียบระยะยุคลิดในชั้นแฝงของชุดข้อมูล Pima Diabetes

โหนด	ระยะยุคลิดจากค่าน้ำหนักระหว่างโหนด	
	โหนดแฝงที่ 1	โหนดแฝงที่ 2
1	1.529417094	4.77774307
2	2.47243273	3.834727434
3	3.270808751	3.036351413
4	3.542678171	2.764481993
5	2.811188843	3.495971321
6	4.680888662	1.626271502
7	0.393903924	6.701064088
8	6.13645608	0.170704084
9	2.591596054	3.71556411
10	2.968792794	3.33836737
11	2.337057312	3.970102852
12	2.866110372	3.441049792
13	0.334087896	5.973072268
14	7.22986641	0.922706246
15	0.73235746	5.574802704
16	1.832717933	4.474442231
17	1.75244025	4.554719914
18	6.599902143	0.292741979

ตารางที่ 24 เปรียบเทียบระยะยุคลิดในชั้นแฝงของชุดข้อมูล Data Generated 1

โหนด	ระยะยุคลิดจากค่าน้ำหนักระหว่างโหนด	
	โหนดแฝงที่ 1	โหนดแฝงที่ 2
1	0.006590	0.001797
2	0.017856	0.380346
3	0.412843	0.014641
4	0.081238	0.479439
5	0.466673	0.068471
6	0.021775	0.376427
7	0.459746	0.061545
8	0.053227	0.451428
9	0.452960	0.054759
10	0.009754	0.388447
11	0.399999	0.001797
12	0.006590	0.391612
13	0.444798	0.046597
14	0.034784	0.432985
15	0.453008	0.054807
16	0.054854	0.453055
17	0.461979609	0.063778299
18	0.063600322	0.461801632

ตารางที่ 25 เปรียบเทียบระยะยุคลิดในชั้นแฝงของชุดข้อมูล Data Generated 2

โหนด	ระยะยุคลิดจากค่าน้ำหนักระหว่างโหนด	
	โหนดแฝงที่ 1	โหนดแฝงที่ 2
1	6.880210	10.686077
2	51.623085	47.817217
3	7.472762	11.278629
4	58.963722	55.157854
5	2.072513	1.733354
6	9.648769	5.842902
7	5.278279	9.084146
8	49.468534	45.662667
9	1.561544	2.244323
10	3.880334	0.074467
11	11.394896	15.200764
12	56.320416	52.514549
13	0.155617	3.650251
14	3.852893	0.047025
15	55.849801	52.043933
16	7.573190237	11.37905756
17	60.58687314	56.78100582
18	9.982023107	13.78789043

ประวัติผู้เขียนวิทยานิพนธ์

นายกฤศณัฏฐ์ บุญเกียรติพงษ์ เกิดเมื่อวันที่ 31 ตุลาคม พ.ศ. 2518 ที่จังหวัดเชียงใหม่ สำเร็จการศึกษาหลักสูตรปริญญาวิทยาศาสตรบัณฑิต (วท.บ.) สาขาเคมี คณะวิทยาศาสตร์ มหาวิทยาลัยเชียงใหม่ ในปีการศึกษา 2540 และเข้าศึกษาต่อในหลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาวิทยาศาสตร์คอมพิวเตอร์ ที่ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย ในปีการศึกษา 2551