การรวมโหนดในชั้นแฝงเพื่อใช้เรียนรู้ข้อมูลขนาดใหญ่
Integration of Hidden Nodes to Learn Large Scale Data

กฤศณัฏฐ์ บุญเกียรติพงษ์ (Kritsanatt Boonkiatpong)* และ สุกรี สินธุภิญโญ (Sukree Sinthupinyo)*

บทคัดย่อ

การเรียนรู้ด้วยนิวรอลเน็ตเวิร์กบนเซตข้อมูลขนาดใหญ่นั้นจะเกิดปัญหาในด้านเวลาที่ใช้ในการเรียนรู้ และความสามารถในการเรียนรู้ ในงานวิจัยนี้จึงได้เสนอวิธีการเรียนรู้โดยใช้นิวรอลเน็ตเวิร์กหลายเครือข่าย เรียนรู้บนเซตตัวอย่างย่อยที่สุ่มเลือกมาจากเซตตัวอย่างทั้งหมด จากนั้นจึงนำโหนดในชั้นซ่อนจากเน็ตเวิร์กแต่ละอันมารวมกัน เพื่อหาค่าน้ำหนักประจำโหนดในชั้นซ่อนใหม่ ผลการทดลองพบว่า วิธีการที่นำเสนอสามารถลดเวลาในการเรียนรู้บนและยังคงรักษาเปอร์เซ็นต์ความถูกต้องได้เหมือนกับการใช้เซตตัวอย่างทั้งหมด

คำสำคัญ: นิวรอลเน็ตเวิร์ก, ข้อมูลขนาดใหญ่, การรวมโหนด

Abstract

Learning on large scale data set can cause problem in both learning time and learning capability. This research thus proposes a novel method that uses multiple neural networks to learn from multiple subsets extracted from the whole original data set. Then, we employ another method to integrate weight of hidden nodes from each network. The experimental results show that the proposed method can not only reduce the learning time, but also preserve the accuracy.

Keyword: Neural Network, Large Scale Dataset, Node Integration.

1. บทนำ

การศึกษาและงานวิจัยทางด้านนิวรอลเน็ตเวิร์กเป็นไปอย่างกว้างขวาง ซึ่งในปัจจุบันได้แบ่งออกไปในหลายแง่มุม [1] ไม่ว่าจะเป็นการศึกษาทางด้านโครงสร้างแบบจำลอง การออกแบบเน็ตเวิร์ก การปรับปรุงประสิทธิภาพการเรียนรู้ให้สามารถเรียนรู้ได้เร็ว จำแนกได้ถูกต้องมากขึ้น จะเห็นได้ว่ามีเทคนิคต่างๆ มากมายที่เกี่ยวกับนิวรอลเน็ตเวิร์ก รวมไปถึงการนำเอานิวรอลเน็ตเวิร์กไปประยุกต์ใช้งานจริงในรูปแบบต่างๆ เช่น การรู้จำเสียง ภาพ การพยากรณ์สภาพภูมิอากาศ หุ้น การจราจร เป็นต้น ฯลฯ ซึ่งในปัจจุบันวิชาปัญญาประดิษฐ์ได้มีส่วนสำคัญเป็นอย่างมากในเทคโนโลยีทุกแขนงและสามารถนำเอาไปประยุกต์ใช้ได้อย่างกว้างขวางและยากจะหาข้อสิ้นสุดได้ การทำวิจัยของนิวรอลเน็ตเวิร์กก็ยังมีความพยายามที่จะปรับปรุงในลักษณะต่างๆ ไม่ว่าจะเป็นการที่ทำให้เน็ตเวิร์กประมวลผลได้เร็วขึ้น มีประสิทธิภาพมากขึ้น หรือทำให้ความผิดพลาดน้อยลง ยังมีความสำคัญอยู่มากที่ทางผู้วิจัยมุ่งเน้นให้ความสำคัญ

ในงานวิจัยนี้จะมุ่งเน้นไปในการวิเคราะห์เรื่องข้อมูลขนาดใหญ่ที่ใช้ในนิวรอลเน็ตเวิร์กในการเรียนรู้ จากงานวิจัยแสดงให้เห็นว่าจำนวนนิวรอนและจำนวนชั้นในชั้นแฝงจะมีผลต่อประสิทธิภาพ [2] เนื่องจากถ้าไม่มีจำนวนชั้น (Layer) มากทำให้ประมวลผลได้รวดเร็ว และทำการวัดและประเมินผลโดยการทดสอบกับเซตข้อมูลขนาดใหญ่ในการทดสอบ งานวิจัยนี้ได้มุ่งความสนใจลงไปในระดับโครงสร้างของนิวรอลเน็ตเวิร์ก โดยทั่วไปการมีชั้นแฝงจำนวนมาก ผลลัพธ์ที่ได้จะเพิ่มความถูกต้องของการเรียนรู้มากขึ้น แต่ก็จะมีผลกับเวลาที่ใช้โดยจะใช้เวลาในการเรียนรู้มากขึ้นด้วย แต่ในทางกลับกัน ถ้าใช้จำนวนชั้นแฝงน้อยลง แม้การใช้เวลาในการเรียนรู้จะน้อยลง แต่ว่าความถูกต้องก็จะน้อยลงตามไปด้วย อีกทั้งถ้าเป็นเซตข้อมูลขนาดใหญ่ เป็นการยากที่จะเรียนรู้ได้ในครั้งเดียว จะต้องใช้ทั้งทรัพยากรและเวลาจำนวนมาก ในงานวิจัยได้นำเสนอแนวคิดที่ว่าเมื่อเราแบ่งเซตข้อมูลออกเท่าๆ กันเป็นเซตย่อยก็สามารถที่จะทำการเรียนรู้เซตข้อมูลใหญ่ได้โดยใช้ชั้นแฝงแค่ชั้นเดียว การปรับปรุงความถูกต้องจะใช้เทคนิคการ

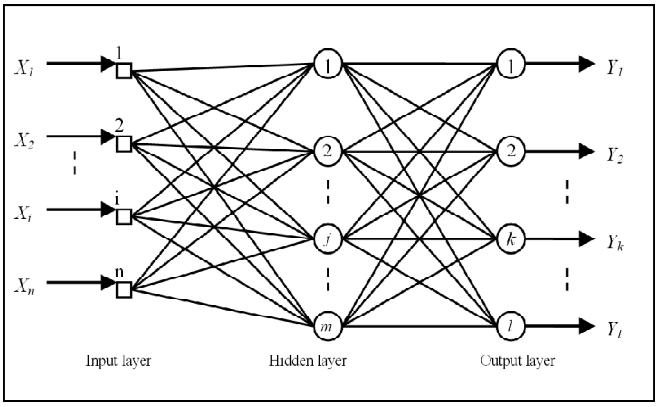
* ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย

หาค่าน้ำหนักที่เหมาะสมของแต่ละโหนดนิวรอนที่ให้ค่าความผิดพลาดที่น้อยที่สุด แล้วนำเอาค่าน้ำหนักที่ดีสุดของเซตข้อมูลย่อยมาสร้างเน็ตเวิร์กใหม่ และเมื่อผ่านการปรับค่าน้ำหนักที่เหมาะสมแล้ว จะสามารถทำให้นิวรอลเน็ตเวิร์กจะสามารถเรียนรู้เซตข้อมูลขนาดใหญ่ได้โดยที่ความถูกต้องใกล้เคียงกับค่าเดิมโดยไม่จำเป็นต้องเรียนรู้กับเซตข้อมูลขนาดใหญ่ในครั้งเดียว

2. ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

2.1 นิวรอลเน็ตเวิร์กแบบการแพร่กระจายย้อนกลับ

ในงานวิจัยนี้ ในส่วนของการเรียนรู้ได้ใช้นิวรอลเน็ตเวิร์กแบบการแพร่กระจายย้อนกลับ (Back Propagation Neural Network-BPNN) ซึ่งเป็นการเรียนรู้แบบมีผู้สอน (Supervised Learning) ซึ่งจะสร้างจากเน็ตเวิร์กที่มีโครงสร้างแบบง่ายๆ ไม่ซับซ้อน นิวรอนแต่จะหน่วยจะเชื่อมต่อกันด้วยค่าน้ำหนัก (Weight) เมื่อมีการส่งผ่านข้อมูลเข้ามาในนิวรอนโดยค่าน้ำหนักที่กำหนดและส่งต่อไปยังนิวรอนหน่วยถัดไปด้วยค่าน้ำหนักที่ต่างออกไป ดังภาพที่ 1



ภาพที่ 1 *ภาพการทำงานของนิวรอลเน็ตเวิร์กแบบแพร่กระจายย้อนกลับ*

ค่าน้ำหนักและค่าการขีดแบ่งสำหรับเน็ตเวิร์กจะถูกกำหนดในอยู่ในช่วงหนึ่งโดยจะมีฟังก์ชันการส่งผ่านโดยนิวรอนในชั้นแฝงจะถูกคำนวณค่าเอาพุตโดยผ่านสมการที่ 1

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

โดย P คือรูปแบบการเรียนรู้ในชุดข้อมูล n คือจำนวนของอินพุตของ j นิวรอนในชั้นแฝง X_i คืออินพุตที่ i ที่ส่งผ่านมายังนิวรอนและ Y_j คือผลลัพธ์เอาพุต ค่า W_{ij} คือค่าน้ำหนักที่แต่ละนิวรอนใช้ q_1 คือค่าขีดแบ่ง และฟังก์ชันซิกมอยด์ (*sigmoid*) จากสมการจะสามารถแสดงได้ดังสมการที่ 2

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

เมื่อมีการคำนวณค่าเอาต์พุตส่งไปยังชั้นเอาต์พุตจะแสดงได้ในสมการที่ 3

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

โดยค่า P คือค่ารูปแบบการเรียนรู้ *(training pattern)* ในชุดข้อมูล, m คือจำนวนของอินพุตของนิวรอนที่ k ในชั้นเอาต์พุต, X_{jk} คือค่าอินพุตเข้าไปในนิวรอนและ Y_k คือค่าผลลัพธ์ของเอาต์พุต W_{jk} คือค่าน้ำหนัก, และ q_κ คือค่าขีดแบ่งของฟังก์ชันซิกมอยด์ (sigmoid) ที่อ้างอิงในสมการที่ 2

2.2 งานวิจัยเกี่ยวกับข้อมูลขนาดใหญ่

แม้การวิจัยเกี่ยวกับนิวรอลเน็ตเวิร์กจะกว้างขวางแต่ปัญหาที่พบกันมากอันหนึ่งมาจนถึงปัจจุบันนี้คือการจัดการกับข้อมูลขนาดใหญ่ มีหลายงานวิจัยที่มุ่งเน้น ตั้งแต่ปี 2000 Aaron J. Owens [3] ได้นำเสนอรูปแบบของนิวรอลเน็ตเวิร์กโดยสร้างแบบจำลอง PLS (Partial Least Squares) MODEL เพื่อลดปัญหาคอขวดของนิวรอลเน็ตเวิร์ก และอีกทั้งเขายังได้ทดสอบการแบ่งข้อมูลออกเป็นเซ็ตย่อยเพื่อที่จะให้ลดเวลาในการเรียนรู้ของนิวรอลเน็ตเวิร์กอีกด้วย

ในปี 2005 ในโครงการ DAME มหาวิทยาลัยยอร์ค ประเทศอังกฤษ Bojian Liang และ James Austin [4] นำเสนอวิธีการลดเวลาในการเรียนรู้ของข้อมูลขนาดใหญ่ที่ใช้ในอนุกรมเวลา (Time Series) พวกเขาได้ใช้เทคโนโลยีกริด (Grid Technology) เข้ามาช่วยเพื่อลดเวลาการเรียนรู้ของข้อมูลขนาดใหญ่ที่ถูกเรียนรู้โดยอนุกรมเวลา และในปี 2008 Ivanikovas และคณะ [5] พวกเขาได้นำเสนอการจัดการข้อมูลตัวอย่างขนาดใหญ่ด้วยอัลกอริทึม SAMANN (Sammon's Mapping Neural Network) ได้ทำการลดจำนวนข้อมูลโดยการคัดแยกข้อมูลที่สนใจด้วย k-means และ SOM จากนั้นทำการเรียนรู้ด้วยนิวรอลเน็ตเวิร์ก จากผลที่แสดงให้เห็นว่าค่าความผิดพลาดลดลงจากการเทียบกับการใช้ข้อมูลทั้งหมด อีกทั้งเวลาในการเรียนรู้ก็ยังลดลงด้วย

2.3 การปรับปรุงประสิทธิภาพนิวรอลเน็ตเวิร์ก

ในการเรียนรู้ของนิวรอลเน็ตเวิร์ก อีกส่วนที่ได้รับความสนใจในการวิจัยมากคือการปรับปรุงเน็ตเวิร์กในเหมาะสมที่สุด เพื่อให้ได้เน็ตเวิร์กสุดท้ายที่จะนำไปใช้ในการเรียนรู้ในชุดข้อมูลได้อย่างมีประสิทธิภาพ ในปี 1996 Gou-Jen Wang และ Chih-Cheng Chen [6] ใช้วิธีการปรับอัลกอริทึมให้กับเน็ตเวิร์กโดยปรับที่ค่าน้ำหนักโดยใช้ค่าก่อนหน้าที่ดีที่สุดมาเป็นเกณฑ์ หลังจากที่ได้ค่าที่เหมาะสม การปรับปรุงนี้จะเกิดขึ้นที่ชั้นแฝงและชั้นเอาต์พุต อัลกอริทึมได้ถูกนำไปทดสอบปัญหา XOR, 4-4 Encoder และ SIMO ซึ่งพบว่าสามารถลดค่าผลรวมความผิดพลาดกำลังสอง (Sum Square Error) ได้อย่างน่าพอใจ

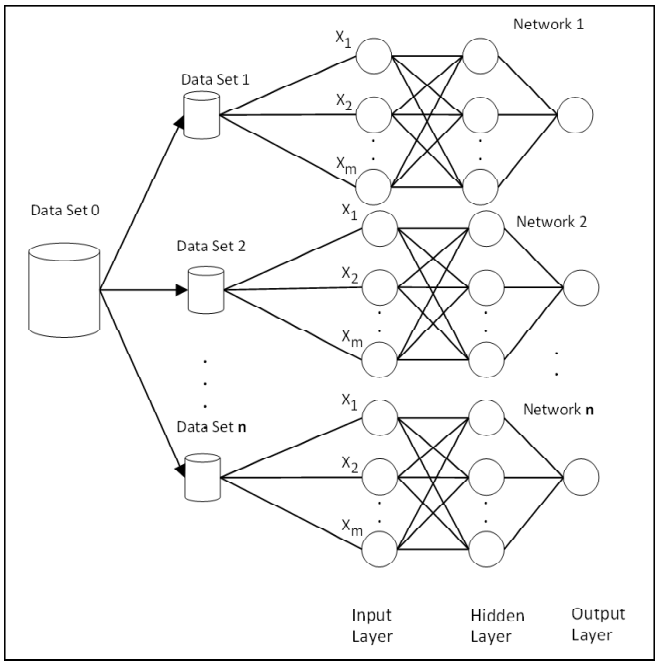
3. การดำเนินการวิจัย

3.1 แนวคิดและวิธีการที่ใช้ในการวิจัย

แนวคิดของการวิจัยเพื่อมุ่งเน้นที่จะพัฒนาอัลกอริทึมในการเรียนรู้เซตข้อมูลขนาดใหญ่โดยการแบ่งเซตข้อมูลออกเป็นเซตย่อยๆ ซึ่งในการเรียนรู้ในขนาดข้อมูลที่น้อยกว่าจะใช้เวลาที่น้อยกว่าเนื่องจากอัตราการเรียนรู้ที่รวดเร็วกว่า แต่ค่าของความผิดพลาดก็จะไม่แน่นอน บางชุดจะให้ค่าความผิดพลาดที่ต่ำกว่าหรือสูงกว่าเซตข้อมูลหลัก แต่เราสามารถเอาชุดที่มีค่าความผิดพลาดที่ต่ำที่สุดมาเป็นเน็ตเวิร์กตั้งต้นในการทดลองโดยการนำเอาค่าน้ำหนัก (Weight) ในแต่ละชั้นของเน็ตเวิร์กไปทดสอบเรียนรู้กับชุดอื่นๆ ว่าสามารถให้ผลการเรียนรู้ว่าสามารถแยกแยะข้อมูลกว่าเน็ตเวิร์กเดิมหรือไม่

สมมติให้เซตข้อมูลมีจำนวนตัวอย่างทั้งหมด X ตัวอย่าง โดยให้เซตข้อมูลหลักเป็น Data Set 0 (D_0) และเซตข้อมูลที่แบ่งย่อยออกเป็น n กลุ่ม และให้ N คือ นิวรอลเน็ตเวิร์ก ตั้งแต่ N_1 จนไปถึง N_n โดยแต่ละกลุ่มจะมีข้อมูลเท่าๆ กันคือ X/n ตัวอย่าง จะทำให้มี Data Set 1 ถึง N แทนเป็น D_1, D_2.., D_n โดยแต่ละเซตจะถูกเรียนรู้ในนิวรอลเน็ตเวิร์กที่มีโครงสร้างเดียวกันและเป็นแบบนิวรอลเน็ตเวิร์กแบบมีชั้นแฝงชั้นเดียว ดังภาพที่ 2

เมื่อทำการเรียนรู้จนครบ n เซตแล้ว เก็บค่าน้ำหนักของแต่ละเซตไว้แล้วนำมาเปรียบเทียบค่า โดยลองเอาชุดค่าน้ำหนักของชุดที่ทำการแบ่งแยกชุดตัวอย่างได้ดีที่สุด คือสามารถเรียนรู้โดยมีความผิดพลาดที่น้อยที่สุด กำหนดให้เป็น N_{best} และเมื่อนำเอาค่าน้ำหนักที่เน็ตเวิร์ก N_{best} ไปทดสอบเรียนรู้กับชุดอื่นๆเพื่อทดสอบว่าชุดค่าน้ำหนักของ N_{best} มีค่าเหมาะสมพอที่จะทำการแบ่งแยกเซตข้อมูลอื่นๆ ได้ดีกว่า หรือว่าค่า



ภาพที่ 2 แสดงแบบจำลองที่ใช้ในงานวิจัย ในการแบ่งเซตข้อมูล n กลุ่มเพื่อทำการเรียนรู้ในนิวรอลเน็ตเวิร์กชั้นแฝงเดี่ยวที่มีโครงสร้างเหมือนกัน

น้ำหนักของชุด N_i เองสามารถทำการแบ่งแยกได้ดีกว่า N_{best} เมื่อ N_i คือ นิวรอลเน็ตเวิร์กที่ของชุดข้อมูลที่ i

เมื่อเราทำอย่างนี้ไปเรื่อยๆจนครบ n กลุ่มและสลับค่าน้ำหนักที่ดีที่สุดไปเรื่อย ๆ จนสุดท้ายเราจะได้นิวรอลเน็ตเวิร์กที่โครงสร้างเดิมแต่มีค่าน้ำหนักที่เหมาะสมที่สุดเพื่อทำการแบ่งแยกเซตข้อมูลหลักที่เป็น N_0 ได้ดีกว่าเดิม

แนวคิดนี้ทำไปประยุกต์กับการเรียนรู้ของเซตข้อมูลขนาดใหญ่ได้ เพราะในทางปฏิบัติแล้วการที่จะเรียนรู้ข้อมูลขนาดใหญ่ตรงๆ นั้น เป็นเรื่องที่ทำได้ยากมาก เราใช้แนวคิดในการแบ่งข้อมูลออกเป็นเซตข้อมูลย่อยนี้เพื่อให้การวิเคราะห์ข้อมูลขนาดใหญ่ทำได้ง่ายขึ้น ในขณะที่ความผิดพลาดไม่ได้ต่างหรือใกล้เคียงกับของเดิมไม่มากนัก

3.2 การเตรียมข้อมูลในการวิจัย

ในการทดลองนี้ จะทำการทดสอบที่กลุ่มข้อมูล 2 กลุ่มซึ่งทำการสร้างจากเวกา (WEKA) [7] ซึ่งจะทำการทดสอบจากกลุ่มข้อมูลเล็กๆ ก่อน เพื่อที่จะนำไปประยุกต์กับข้อมูลใหญ่ต่อไป โดยข้อมูลจะมี 2,000 ชุด แบ่งเป็นชุดย่อย อย่างละ 1,000 ชุด และทดสอบโดยใช้นิวรอลเน็ตเวิร์กที่มีโครงสร้างแบบเดียวกันมาทำการเรียนรู้

3.3 การเปรียบเทียบค่าน้ำหนัก

เมื่อได้ชุดน้ำหนักของเซตข้อมูลย่อยที่ดีที่สุด N_{best}

ในการทดแทนค่าน้ำหนักนี้ไปยังเซตข้อมูลย่อยอื่นๆ ในการวิจัยได้ใช้โปรแกรมประยุกต์ MBP (Multiple Backpropagation) [8] ซึ่งมีฟังก์ชันในการทดแทนค่าน้ำหนักเข้าไปยังนิวรอลเน็ตเวิร์ก

ในการทดแทนค่าน้ำหนักเข้าไปในชั้นแฝงและชั้นเอาต์พุตของงานวิจัยนี้ ได้ใช้หลักการเปรียบเทียบความเหมือนของชุดน้ำหนักที่ได้มาของแต่ละเน็ตเวิร์กย่อย ซึ่งจะเป็นการแยกข้อมูล 2 ชุดออกจากกัน โดยการเปรียบเทียบค่าน้ำหนัก 2 ค่า จะเปรียบเทียบใช้เชิงของความชันบนพื้นผิวการตัดสินใจ (Decision Surface) จะแสดงในสมการที่ 4

ให้ W_{1i} แทนค่าน้ำหนักของเน็ตเวิร์กที่ 1 โหนดแฝงที่ i
W_{2i} แทนค่าน้ำหนักของเน็ตเวิร์กที่ 2 โหนดแฝงที่ i

$$y = \frac{\sum_{i=0}^{n} |W_{1i} - W_{2i}|}{n} \qquad (4)$$

3.4 การรวมโหนด

เมื่อ n คือจำนวนโหนดในชั้นแฝง โดยค่า y คือค่าผลต่างรวมเฉลี่ยของชุดน้ำหนักของเน็ตเวิร์ก 2 ชุด ถ้า y มีค่าในช่วง 0-1 ถือว่าเน็ตเวิร์ก 2 ชุดมีความใกล้เคียงกัน

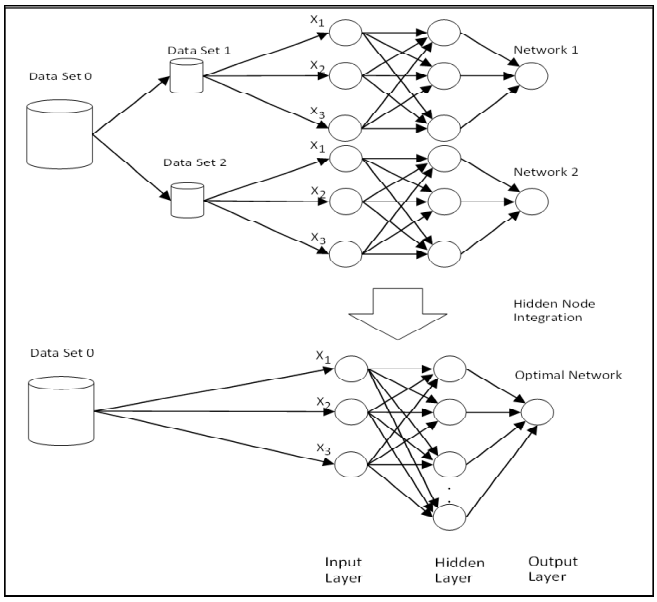
อัลกอริทึมการรวมโหนดจะมี 2 ส่วน ซึ่งวิเคราะห์จากค่าน้ำหนักของแต่ละโหนดที่ตำแหน่งเดียวกัน กำหนดให้ W_{Nbest} คือค่าน้ำหนัก ของเน็ตเวิร์กที่มีค่าความผิดพลาดน้อยที่สุด และ W_{Ni} คือค่าน้ำหนักของเน็ตเวิร์กที่ i

1) ในกรณีที่ W_{Ni} ของโหนดนั้นๆ มีค่าใกล้เคียงกันกับ W_{Nbest} ที่ตำแหน่งโหนดเดียวกัน ค่าน้ำหนักจะถูกเฉลี่ยเข้าด้วยกันกับค่า W_{Nbest} เพื่อเป็นค่าน้ำหนักใหม่ของโหนดนั้นๆ ดังแสดงในสมการที่ 5

$$W_{Nbest} \leftarrow \frac{(W_{Nbest} + W_{Ni})}{2} \qquad (5)$$

2) ในกรณีค่า W_{Ni} ของโหนดนั้นๆ มีค่าต่างจาก W_{Nbest} มาก โหนดของ N_i นั้นจะถูกเพิ่มเข้าไปในเน็ตเวิร์ก N_{best} เพื่อเพิ่มประสิทธิภาพของการแยกแยะข้อมูลในส่วนที่ N_{best} แยกแยะไม่ได้หรือได้น้อยกว่า

จากอัลกอริทึมดังกล่าว จะทำให้สามารถรวมโหนดของชั้นแฝงในนิวรอลเน็ตเวิร์ก 2 เน็ตเวิร์กเข้าด้วยกัน จะทำให้ได้เน็ตเวิร์กใหม่ที่มีชั้นแฝงเดี่ยว (Single Hidden Layer) แต่มีจำนวนโหนดที่แตกต่างออกไปจากเน็ตเวิร์กตั้งต้นดังแสดงในภาพที่ 3 ซึ่งเน็ตเวิร์กที่ผ่านการรวมโหนดนี้จะถูกนำไปใช้เรียนรู้ชุดข้อมูลเพื่อเปรียบเทียบทั้งในแง่ความผิดพลาดและเวลาที่ใช้



ภาพที่ 3 *แสดงการรวมโหนดของนิวรอลเน็ตเวิร์ก 2 เน็ตเวิร์กที่มีการรวมโหนดแล้วซึ่งจะมีจำนวนโหนดในชั้นแฝงที่ต่างออกไปจากเน็ตเวิร์กเดิม*

4. ผลการดำเนินงานวิจัย

การทดลองจะมี 3 ส่วนคือ การทำการเรียนรู้ปกติให้กับนิวรอลเน็ตเวิร์กในโครงสร้างเดียวกันให้กับแต่ละชุดข้อมูล จากนั้นก็หาน้ำหนักจากเน็ตเวิร์กที่มีค่าความผิดพลาดรากกำลังสองเฉลี่ย (Root Mean Square Error: RMS) ที่น้อยที่สุดมาเป็นตัวตั้งต้นในการสร้างเน็ตเวิร์กใหม่และในส่วนสุดท้ายจะทดลองในแง่ของเวลาที่ใช้เปรียบเทียบเน็ตเวิร์กเดิมกับเน็ตเวิร์กใหม่ที่ได้จากการรวมโหนด

4.1 เปรียบเทียบความผิดพลาดรากกำลังสองเฉลี่ย

ในการทดลองแรกจะเป็นการเรียนรู้ให้กับชุดข้อมูลแต่ละชุด ซึ่งประกอบไปด้วย ชุดข้อมูลตั้งต้น (Data Set 0: D_0) จากนั้นก็แบ่งชุดข้อมูลออกเป็น 2 ส่วน เท่าๆ กัน เป็นชุดข้อมูลย่อยที่ 1 (Data Set 1: D_1) และ ชุดข้อมูลย่อยที่ 1 (Data Set 1: D_1)

แล้วทำการเรียนรู้เพื่อหาค่าเน็ตเวิร์กที่มีค่าความผิดพลาดรากกำลังสองเฉลี่ยน้อยที่สุด กำหนดให้เป็น N_{best}

จากตารางที่ 1 จะพบว่า ค่าความผิดพลาดของชุดข้อมูลตั้งต้น (D_0) ที่เราต้องการทดสอบนั้น เมื่อนำมาแบ่งข้อมูลออกเป็น 2 ส่วน คือ ชุดข้อมูลย่อย 1 (D_1) และ ชุดข้อมูลย่อย 2 (D_2) แล้วทำการเรียนรู้บนนิวรอลเน็ตเวิร์กแล้ว จะพบว่าชุดข้อมูลย่อย 2 จะมีค่าความผิดพลาดรากกำลังสองเฉลี่ยที่น้อยกว่า (N_{best}) จึงนำเอาค่าน้ำหนักที่ได้จากชุดข้อมูลย่อยที่สองที่มาเป็นเน็ตเวิร์กตั้งต้นเพื่อสร้างรวมโหนดเพื่อให้ได้เน็ตเวิร์กใหม่ขึ้นมา

ตารางที่ 1 *แสดงค่าความผิดพลาดรากกำลังสองเฉลี่ยของแต่ละชุดข้อมูลบนนิวรอลเน็ตเวิร์กที่มีโครงสร้างเดียวกัน*

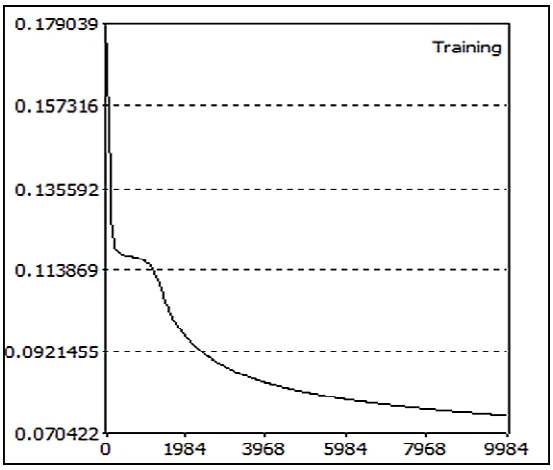
เน็ตเวิร์ก/ข้อมูล	จำนวนข้อมูล	ค่าความ
Data Set 0 (D_0)	2,000	0.0753476
Data Set 1 (D_1)	1,000	0.0762917
Data Set 2 (D_2)	1,000	0.0642485

เมื่อนำเอาค่าน้ำหนักของ N_{best} มาทำการเรียนรู้ในชุดข้อมูลย่อยที่ 1 (D_1) จากนั้นก็ทำการรวมเน็ตเวิร์กดังอัลกอริทึมที่กล่าวมาในข้อ 4.3 และภาพที่ 4 กำหนดให้เน็ตเวิร์กใหม่ที่ได้จากการรวมโหนดแล้วเป็น Data Set N (D_N) นำเอา D_N ไปเรียนรู้กับชุดข้อมูลตั้งต้น Data Set 0 ผลแสดงไว้ดังตารางที่ 2

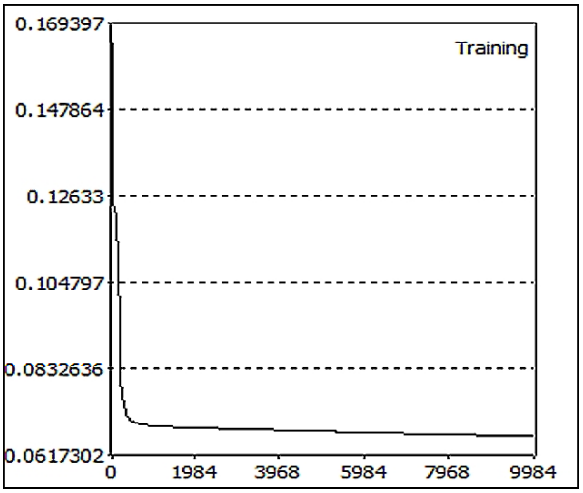
ตารางที่ 2 *แสดงค่าความผิดพลาดรากกำลังสองเฉลี่ยของเน็ตเวิร์กที่ได้จากการรวมโหนด*

เน็ตเวิร์ก/ข้อมูล	จำนวนข้อมูล	ค่าความผิดพลาด
Data Set 0 (D_0)	2,000	0.0753476
Data Set N (D_N)	2,000	0.0666208

หลังจากที่ทำการรวมโหนดโดยใช้อัลกอริทึมการแทนค่าน้ำหนักที่กล่าวมาแล้ว และนำเอาเน็ตเวิร์กใหม่ที่ได้นี้ (D_N) ไปเรียนรู้กับชุดข้อมูลเดิม จะเห็นได้ว่าค่าความผิดพลาดรากที่สองกำลังสองเฉลี่ยของนิวรอลเน็ตเวิร์กใหม่มีค่าน้อยลงกว่าเดิม ดังแสดงเพิ่มเติมใน ภาพที่ 4 และภาพที่ 5



ภาพที่ 4 *แสดงค่าความผิดพลาดรากกำลังสองเฉลี่ยของนิวรอลเน็ตเวิร์กในแบบจำลองก่อนการปรับปรุงและรวมโหนดของเน็ตเวิร์ก*



ภาพที่ 5 *แสดงค่าความผิดพลาดรากกำลังสองเฉลี่ยของนิวรอลเน็ตเวิร์กในแบบจำลองหลังจากการปรับปรุงและรวมโหนดของเน็ตเวิร์ก*

4.2 เวลาที่ใช้ในการเรียนรู้

ในแง่ของเวลา เน็ตเวิร์กใหม่ที่ได้จากการรวมโหนด (D_N) เมื่อกำหนดค่าความผิดพลาดไว้ค่าหนึ่ง (ในการทดลองกำหนดไว้ที่ 0.06 โดยใช้ผลการทดลอง 4.1 เป็นเกณฑ์) แล้วให้เน็ตเวิร์กทั้ง 2 ทำการเรียนรู้จนไปถึงค่าความผิดพลาดที่กำหนดไว้

ตารางที่ 3 *แสดงการเปรียบเทียบในเชิงของเวลาที่ใช้ของทั้งสองนิวรอลเน็ตเวิร์ก*

เน็ตเวิร์ก/ข้อมูล	ค่าความผิดพลาด	Epoch	เวลาที่ใช้ (วินาที)
Data Set 0 (D_0)	0.06	145,321	718
Data Set N (D_N)	0.06	96,153	381

จากตารางที่ 3 จะพบว่าค่าของเน็ตเวิร์กใหม่ (D_N) จะมีค่าการเรียนรู้ที่น้อยลงกว่าเน็ตเวิร์กในชุดข้อมูลตั้งต้น (D_0)

5. สรุป

จากตารางที่ 1 ในผลการทดลอง จะเห็นได้ว่าเมื่อเราทำการแบ่งข้อมูลออกเป็นกลุ่มย่อยแล้วแยกทำการเรียนรู้ในเบื้องต้นเวลาที่ใช้จะลดลง แต่ทั้งนี้เองเน็ตเวิร์กย่อยเองในแต่ละชุดข้อมูลก็ไม่ได้มีโครงสร้างสำหรับการเรียนรู้ที่ดีที่สุดจำเป็นต้องมีการปรับปรุง และเมื่อเราปรับค่าน้ำหนักให้เหมาะสมด้วยอัลกอริทึมที่กล่าวมาแล้ว จะพบว่าเมื่อนำเอาเน็ตเวิร์กที่ได้จากการรวมโหนดที่มีค่าน้ำหนักที่ดีที่สุด กลับไปเรียนรู้ในชุดข้อมูลขนาดใหญ่ที่เตรียมไว้ จะทำให้ ความผิดพลาดโดยรวมลดลงดังแสดงไว้ในตารางที่ 2 ในขณะที่ตารางที่ 3 แสดงถึงเวลาที่ใช้ในการเรียนรู้ โดยที่จะเห็นได้ว่าเมื่อเรากำหนดค่าความผิดพลาดค่าหนึ่งเป็นเกณฑ์ไว้ แล้วทำการเรียนรู้เปรียบเทียบ เวลาที่ใช้ในเน็ตเวิร์กที่มีการรวมโหนดและผ่านการปรับค่าที่เหมาะสมแล้ว (Optimized Network) จะใช้เวลาในการเรียนรู้ที่น้อยกว่าเน็ตเวิร์กตั้งต้นมาก

ในงานวิจัยนี้มุ่งเน้นไปในการปรับปรุงโครงสร้างนิวรอลเน็ตเวิร์กโดยใช้การแพร่กระจายแบบย้อนกลับเป็นเน็ตเวิร์กตั้งต้น ซึ่งอัลกอริทึมที่ได้จะถูกนำไปประยุกต์ใช้กับข้อมูลที่มีขนาดใหญ่ขึ้นเพื่อที่จะสามารถเรียนรู้และแยกแยะข้อมูลเหล่านั้นได้โดยที่ค่าความถูกต้องมากขึ้นในขณะที่ใช้เวลาในการเรียนรู้ลดลง

6. เอกสารอ้างอิง

[1] Ying Zhao, Jun Gao, and Xuezhi Yang, "A Survey of neural network ensembles," *IEEE Trans. Pattern Analysis and Machine Intelligence*, pp. 438-442, 2005.

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

[3] Aaron J. Owens, "Empirical Modeling of Very Large Data Sets Using Neural Networks," *IEEE Transactions on Neural Networks*, pp. 302-307, 2000.

[4] Bojian Liang and James Austin, "A neural network for mining large volumes of time series data," *IEEE Transactions on Neural Network*, pp. 688-693, 2005.

[5] Sergejus Ivanikovas, Gintautas Dzemyda, and Viktor Medvedev, "Large Datasets Visualization with Neural Network Using Clustered Training Data," *Springer-Verlag Berlin Heidelberg*, pp. 143-152, 2008.

[6] Gou-Jen Wang and Chih-Cheng Chen, "A Fast Multilayer Neural-Network Training Algorithm Based on the Layer-By-Layer Optimizing Procedures," *IEEE Transactions on Neural Networks*, Vol. 7, No.3, pp. 768-775, 1996.

[7] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten. "The WEKA Data Mining Software," *An Update. SIGKDD Explorations*, Vol. 11, Iss. 1, 2009.

[8] Multiple Back-Propagation, Back-Propagation Simulation Tool, Available online at http://dit.ipg.pt/MBP, 2009.