การพัฒนาโครงข่ายประสาทเทียมด้วยวิธีระบบอาณานิคมมด

Evolving Neural Networks Topology using Ant Colony System

นิกร โภคอุดม[*] เอื้อน ปิ่นเงิน[**] และ บุญวัฒน์ อัตชู[*]

[*]ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

**ภาควิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ มหาวิทยาลัยรามคำแหง

บทคัดย่อ

บทความนี้นำเสนอการใช้วิธีระบบอาณานิคมมด (Ant Colony System : ACS) ทำงานร่วมกับการเรียนรู้แบบ Back-Propagation (BP) มาพัฒนาหาโครงข่ายประสาทเทียมที่เหมาะสมทั้งรูปแบบการเชื่อมต่อระหว่างโหนดและค่าน้ำหนักการเชื่อมภายในไปพร้อมๆ กัน วิธีที่นำเสนอนี้ถูกพัฒนาขึ้นเพื่อหลบเลี่ยงปัญหาซึ่งเกิดจากการฝึกสอนด้วย BP เพียงอย่างเดียว ได้แก่การติดกับดักปัญหาจุดเหมาะสมเฉพาะบริเวณ (local optimum) และปัญหาการเข้ากันกับชุดข้อมูลฝึกสอนมากเกินไป (overfitting) โดยวิธี ACS จะใช้สำหรับเลือกโครงสร้างของโครงข่ายที่เหมาะสมส่วน BP จะใช้ฝึกสอนแบบบางส่วนกับโครงข่ายที่ได้มาจากมดแต่ละตัวด้วยจำนวนรอบน้อยๆ เพื่อปรับค่าน้ำหนักและวัดค่าความเหมาะสมจากค่าความผิดพลาดของโครงข่ายนั้นเพื่อนำไปใช้ในการปรับปรุงค่าพารามิเตอร์ต่างๆ และค่าฟีโรโมนในกระบวนการของ ACS รอบถัดไป โครงข่ายที่ได้จากวิธีที่นำเสนอนี้จะถูกนำไปเปรียบเทียบกับโครงข่ายแบบมาตรฐานที่ฝึกสอนด้วยวิธี BP และโครงข่ายที่ได้จากงานวิจัยซึ่งใช้วิธี ACO-BP [1] โดยวัดจากประสิทธิภาพของจำแนกชนิดกลุ่มข้อมูลจากชุดข้อมูลมาตรฐาน PROBEN1 [2] จำนวน 3 ชุดข้อมูลได้แก่ Cancer1, Diabetes1 และ Glass1 ผลการทดลองพบว่าวิธีที่นำเสนอมีผลลัพธ์ที่ดีกว่าในทุกชุดข้อมูล

คำสำคัญ: วิธีระบบอาณานิคมมด, การพัฒนาโครงข่ายประสาทเทียม, โครงข่ายประสาทเทียม.

Abstract

This paper proposed a new method using Ant Colony System (ACS) to evolving the NN structure both link between neurons and connections weights simultaneously and partially train the network of ant by BP learning for adjust parameter and pheromone value in ACS process. Due to training difficulties in general NN structure, this method can cope with trapping, local optimum and overfitting problem. For evaluating our algorithm, we apply it to pattern classification problems from PROBEN1 benchmark data set [2], we chose Cancer1, Biabetes1 and Glass1 for our experimentation. The proposed method is compare with the standard ANN and ACO-BP method in [1]. The computing results show that the precision and efficiency of the proposed method are all better.

Key words: Ant Colony System, evolving neural networks, neural networks.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. บทนำ

การฝึกสอนโครงข่ายประสาทเทียมแบบแพร่ย้อนกลับ (Back-Propagation: BP) เป็นการปรับปรุงค่าน้ำหนักการเชื่อมต่อระหว่างโหนดด้วยวิธี Gradient descent ที่มีเป้าหมายเพื่อลดความผิดพลาดของโครงข่ายในแต่ละรอบการทำงานนั้น พบว่าบ่อยครั้งที่ค่าน้ำหนักที่ได้จากการปรับจะตกอยู่ในช่วงของความเหมาะสมเฉพาะบริเวณ (local optimum) [4,5,6,7] อันเป็นสาเหตุทำให้ไม่สามารถทำการฝึกสอนโครงข่ายได้สำเร็จ ปัญหาดังกล่าวนี้อาจเกิดจากการกำหนดรูปแบบการเชื่อมต่อภายในและค่าน้ำหนักเริ่มต้นที่ไม่เหมาะสมให้โครงข่าย ซึ่งผู้ฝึกสอนมักแก้ไขโดยกำหนดโครงข่ายเริ่มต้นใหม่อีกครั้ง(trial-and-error) ส่วนปัญหาการเข้ากันกับชุดข้อมูลฝึกสอนมากเกินไป (overfitting) [3] เป็นปัญหาที่เกิดจากการที่โครงข่ายประสาทเทียมนั้นมีประสิทธิภาพในการเรียนรู้กับชุดข้อมูลฝึกสอนดีมากเกินไป โครงข่ายที่ได้จึงมีความสามารถในการระบุกลุ่มของข้อมูลฝึกสอนได้สูงแต่กลับมีความถูกต้องในการระบุกลุ่มข้อมูลทดสอบต่ำ

จึงมีงานวิจัยต่างๆ ได้นำเสนอวิธีการแก้ไขปัญหาดังกล่าวได้แก่ K.S. Tang [4] ใช้วิธีการทางพันธุศาสตร์ (genetic algorithms : GA) มาฝึกสอนหาค่าน้ำหนักของโครงข่ายประสาทเทียมที่เหมาะสมแทนการเรียนรู้แบบ BP, V. Maniezzo [5] ได้ใช้ GA พัฒนาหาโครงข่ายที่เหมาะสมทั้งการเชื่อมต่อระหว่างโหนดและค่าน้ำหนักไปพร้อมกันและโหนดนั้นสามารถเชื่อมต่อกันข้ามชั้นได้ ซึ่งวิธีการเข้ารหัสปัญหาเป็นค่าบิต 0 และ 1, S.J. Han [7] นำเสนอการพัฒนาโครงข่ายประสาทเทียมสำหรับปัญหา anomaly detection โดยออกแบบให้โครงข่ายไม่มีชั้นซ่อนภายในแต่จะกำหนดเป็นกลุ่มของโหนดซ่อน (hidden nodes)ไว้แทนและภายในสามารถเชื่อมต่อข้ามโหนดถัดไปได้และใช้ GA พัฒนาค้นหารูปแบบการเชื่อมต่อระหว่างโหนดซ่อนภายในและค่าน้ำหนักที่เหมาะสมไปพร้อมๆ กันได้, งานวิจัย Y.P. Liu [1] ได้นำเสนอวิธี ACO-BP โดยเป็นการนำวิธีการหาคำตอบที่เหมาะสมด้วยอาณานิคมมด (Ant Colony Optimization: ACO) [8,9] มาค้นหาเฉพาะค่าน้ำหนักการเชื่อมต่อเริ่มต้นที่เหมาะสมของโครงข่ายประสาทเทียมแบบมาตรฐาน (ANN) จากนั้นจะนำโครงข่ายที่ได้ไปฝึกสอนด้วย BP อีกครั้ง ซึ่งวิธี ACO เป็นขั้นตอนวิธีที่มีลักษณะในการค้นหาคำตอบที่เหมาะสมอีกแบบหนึ่ง โดยมีพื้นฐานมาจากการร่วมมือกันค้นหาแหล่งอาหารของมดภายในฝูง และในการค้นหาเส้นทางจากรังไปยังแหล่งอาหารนั้นจะพยายามหาเส้นทางที่มีระยะทางที่สั้นที่สุดโดยให้สารเคมีที่เรียกว่า ฟีโรโมน (Pheromone) เป็นข้อมูลในการค้นหา ซึ่งจุดเด่นของ ACO ที่แตกต่างจาก GA นั้นคือ ACO จะไม่มีการตัดคำตอบที่ไม่ดีทิ้งไปดังนั้นในรอบการทำงานท้ายๆ จึงยังคงมีตัวเลือกของคำตอบให้เลือกเท่าเดิมซึ่งต่างจาก GA ที่มีจุดอ่อน [6] ในกระบวนการคัดเลือกโครโมโซมพ่อและแม่โครโมโซมใดที่ไม่ได้ถูกคัดเลือกจะต้องถูกคัดทิ้งไปซึ่งบางครั้งโครโมโซมที่ถูกคัดทิ้งอาจจะมีคำตอบที่สามารถนำมาสร้างเป็นโครงข่ายที่เหมาะสมได้

ดังนั้นบทความนี้จึงได้นำเสนอเทคนิควิธีระบบอาณานิคมมด (Ant Colony System: ACS) [8,9] ซึ่งเป็นอีกวิธีการหนึ่งของ ACO มาใช้ในการพัฒนาโครงข่ายประสาทเทียมซึ่งจะแตกต่างจากวิธี ACO-BP ของงานวิจัย [1] โดยโครงข่ายที่ออกแบบนั้นโหนดภายในสามารถเชื่อมต่อกันข้ามชั้นได้ ซึ่งจะใช้ACS ทำหน้าที่เลือกรูปแบบการเชื่อมต่อของโหนดภายในและค่าน้ำหนักที่เหมาะสมไปพร้อมๆ กัน และจะใช้การฝึกสอนแบบ BP ทำงานร่วมกันในกระบวนการของ ACS ด้วย

2. การพัฒนาโครงข่ายประสาทเทียมโดยใช้วิธีระบบอาณานิคมมด

วิธีที่นำเสนอจะใช้วิธีระบบอาณานิคมมด (ACS) ในการค้นหาโครงสร้างของโครงข่ายประสาทเทียมที่เหมาะสมทั้งในส่วนของการเชื่อมต่อภายในและค่าน้ำหนักจากโครงข่ายที่ได้กำหนดจำนวนชั้นซ่อน จำนวนโหนดในชั้นอินพุท ชั้นซ่อน และชั้นเอาท์พุทไว้แล้ว ซึ่งโหนดภายในโครงข่ายสามารถเชื่อมต่อกันข้ามชั้นได้แต่จะไม่เชื่อมต่อย้อนกลับไปยังโหนดในชั้นก่อนหน้า และโหนดที่อยู่ในชั้นติดกันก็ไม่จำเป็นต้องเชื่อมต่อกัน และภายในกระบวนการ ACS สามารถลบโหนดทิ้งได้ ซึ่งโหนดที่ถูก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ลบจะเป็นโหนดที่ไม่มีการเชื่อมต่อเข้ามาจากโหนดอื่น หรืออาจเป็นโหนดที่ไม่มีการเชื่อมต่อออกไปยังโหนดอื่น ส่วน BP นั้นจะนำมาใช้ฝึกสอนโครงข่ายแบบบางส่วน (partial learning) ด้วยจำนวนรอบน้อยๆ เพื่อปรับค่าน้ำหนักของโครงข่ายของมดแต่ละตัวและดูแนวโน้มความเหมาะสมของโครงข่ายโดยวัดจากค่าความผิดพลาดของโครงข่ายและนำไปใช้ในการปรับปรุงค่าฟีโรโมนของพารามิเตอร์ที่เป็นตัวเลือกของ ACS ในรอบถัดไป

2.1 การคำนวณของโครงข่ายประสาทเทียม

ด้วยวิธีการนี้ได้ออกแบบให้โหนดภายในสามารถเชื่อมต่อกับโหนดอื่นแบบข้ามชั้นได้ จึงมีการปรับปรุงสมการให้สอดคล้องกับรูปแบบของโครงข่ายดังนี้

1) การคำนวณค่าเอาท์พุตของโหนดในชั้นซ่อนและชั้นเอาท์พุตดังสมการที่ (1) และ (2)

$$y_h = f\left(\sum_l x_l w_{lh} - \theta_h\right) \quad (1)$$

$$y_g = f\left(\sum_l x_l w_{lg} - \theta_g\right) \quad (2)$$

โดย y_h คือค่าเอาท์พุตของโหนด h ในชั้นซ่อน, y_g คือค่าเอาท์พุตของโหนด g ในชั้นเอาท์พุต, f เป็นฟังก์ชันการกระตุ้น, x_l คือค่าอินพุตจากโหนด l ซึ่ง l คือหมายเลขโหนดใดๆ ในชั้นก่อนหน้า (รวมชั้นอินพุต) ที่ได้เชื่อมต่อมายังโหนด h หรือ g , w_{lh} คือค่าน้ำหนักการเชื่อมต่อระหว่างโหนด l กับโหนด h, w_{lg} คือค่าน้ำหนักการเชื่อมต่อระหว่างโหนด l กับโหนด g และ θ_h กับ θ_g คือค่า threshold ของโหนด h และ g ตามลำดับ

2) การปรับปรุงค่าน้ำหนักการเชื่อมต่อของโหนดในชั้นเอาท์พุตและโหนดชั้นซ่อน

- การคำนวณค่า error gradient สำหรับโหนด g ในชั้นเอาท์พุต ($\delta_g(p)$) ดังสมการที่ (3) และ (4)

$$\delta_g(p) = y_g(p).[1 - y_g(p)].e_g(p) \quad (3)$$

$$e_g(p) = y_{d,g} - y_g(p) \quad (4)$$

ซึ่ง $e_g(p)$ คือค่าความผิดพลาดของโหนด g ในรอบการทำงานที่ p ระหว่างค่าเอาท์พุตเป้าหมาย $y_{d,g}$ กับค่าเอาท์พุตจริง $y_g(p)$ ที่ได้จากการคำนวณ

- การคำนวณค่า weight corrections ($\Delta w_{lg}(p)$) เพื่อใช้ในการปรับค่าน้ำหนักการเชื่อมต่อระหว่างโหนด l กับ g ดังสมการที่ (5)

$$\Delta w_{lg}(p) = \alpha . y_l(p).\delta_g(p) \quad (5)$$

ซึ่ง α คืออัตราการเรียนรู้ของโครงข่ายประสาทเทียม

- การปรับค่าน้ำหนักการเชื่อมต่อระหว่างโหนด l กับ g ดังสมการที่ (6)

$$w_{lg}(p+1) = w_{lg}(p) + \Delta w_{lg}(p) \quad (6)$$

- การคำนวณค่า error gradient สำหรับโหนด g ในชั้นซ่อน ($\delta_g(p)$) ดังสมการที่ (7)

$$\delta_h(p) = y_h(p).[1 - y_h(p)].\sum_r \delta_r(p).w_{hr}(p) \quad (7)$$

ซึ่ง r คือหมายเลขโหนดใดๆ ชั้นถัดไปที่โหนด h ได้เชื่อมต่ออยู่ด้วย

- การคำนวณค่า weight corrections ($\Delta w_{hr}(p)$) เพื่อใช้ในการปรับค่าน้ำหนักการเชื่อมต่อระหว่างโหนด h กับ r

$$\Delta w_{hr}(p) = \alpha . x_h(p).\delta_r(p) \quad (8)$$

- การปรับค่าน้ำหนักการเชื่อมต่อระหว่างโหนด h กับ r

$$w_{hr}(p+1) = w_{hr}(p) + \Delta w_{hr}(p) \quad (9)$$

2.2 ตารางพารามิเตอร์สำหรับ ACS

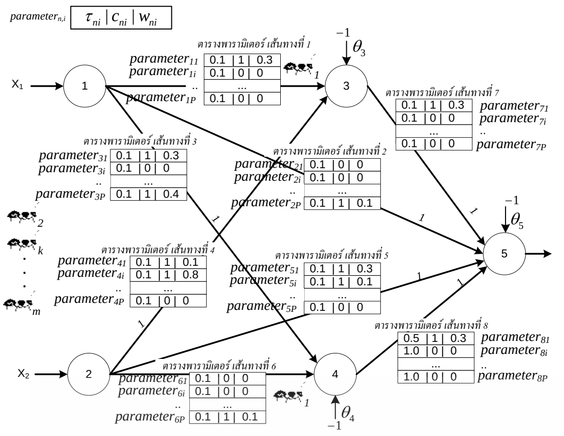
จากโครงข่ายที่ได้กำหนดจำนวนโหนดในชั้นอินพุท จำนวนชั้นซ่อน โหนดในชั้นซ่อน และโหนดในชั้นเอาท์พุทไว้แล้ว จะได้การเชื่อมต่อหรือเส้นทาง (*route*) ระหว่างโหนดภายในที่เป็นไปได้ ซึ่งแต่ละเส้นทางจะกำหนดให้มีตารางพารามิเตอร์ โดยให้แต่ละพารามิเตอร์ในเส้นทาง n แทนด้วย $parameter_{ni}$ ($i = 1, 2, \ldots, P$) กำหนดให้ภายใน $parameter_{ni}$ ประกอบด้วยค่าฟีโรโมน (τ_{ni}) การเชื่อมต่อ(c_{ni}) และค่าน้ำหนักการเชื่อมต่อ (w_{ni}) ตัวอย่างแสดงดังรูปที่ 1 โดยมดทุกตัวใน ACS แทนด้วย k ซึ่ง $k = 1,..,m$ จะทำการเลือกพารามิเตอร์จากตารางพารามิเตอร์ของแต่ละเส้นทาง เพื่อนำมากำหนดลักษณะของโครงข่ายสำหรับการทำงาน

2.3 ขั้นตอนการทำงานของวิธีการที่นำเสนอ

<u>ขั้นตอนที่ 1</u> กำหนดค่าเริ่มต้นให้กับ $parameter_{ni}$ ในตารางพารามิเตอร์ในทุกๆ เส้นทางดังนี้

- ค่าฟีโรโมน (τ_{ni}) มีค่าเท่ากับค่าฟีโรโมนเริ่มต้น (τ_0)
- สุ่มค่าการเชื่อมต่อ(c_{ni}), $c_{ni} \in \{0,1\}$ ถ้า $c_{ni} = 1$ หมายถึงมีการเชื่อมต่อระหว่างโหนดจะต้องสุ่มค่าน้ำหนัก (w_{ni})

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 1 ตารางพารามิเตอร์ในแต่ละเส้นทาง

ซึ่งมีค่าอยู่ในช่วง $[W_{min}, W_{max}]$ แต่ถ้าสุ่มได้ $c_{ni} = 0$ ซึ่งหมายถึงไม่มีการเชื่อมต่อระหว่างโหนด จะกำหนดให้ค่าน้ำหนักการเชื่อมต่อ(w_{ni}) = 0

<u>ขั้นตอนที่ 2</u> ในเส้นทางที่ n มดตัวที่ k จะพิจารณาเลือกพารามิเตอร์ลำดับที่ i ($parameter_{ni}$) โดยใช้สมการที่ (10) และเลือกจนครบทุกเส้นทาง

$$i = \begin{cases} \arg\max_{1 \le i \le P}\{\tau_{ni}\} & if\ q \le q_0; \\ I & if\ q > q_0; \end{cases} \tag{10}$$

โดย q_0 คือค่าคงที่ ($0 \le q_0 \le 1$), q คือค่าสุ่มที่อยู่ในช่วง [0,1] (แต่ละเส้นทางจะสุ่มค่า q ใหม่ทุกครั้ง) ถ้า $q \le q_0$ มดตัวที่ k จะเลือกพารามิเตอร์ลำดับที่ i ที่มีค่าฟีโรโมนมากที่สุดในตารางพารามิเตอร์นั้น แต่ถ้า $q > q_0$ จะเลือกพารามิเตอร์ลำดับที่ I ซึ่ง I คือหมายเลขพารามิเตอร์ที่ได้จากการสุ่มโดยอาศัยค่าความน่าจะเป็นของการถูกเลือกของแต่ละพารามิเตอร์(p_{ni}) ซึ่งค่า p_{ni} ในตารางพารามิเตอร์ของเส้นทางที่ n คำนวณได้ดังสมการ (11)

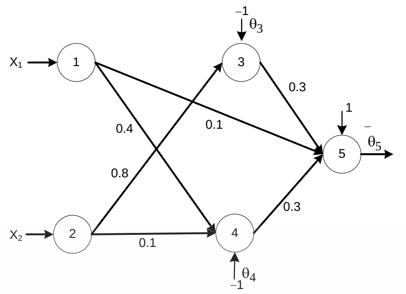
$$p_{ni} = \frac{\tau_{ni}}{\sum_{i=1}^{P} \tau_{ni}} \tag{11}$$

<u>ขั้นตอนที่ 3</u> มดตัวที่ k จะทำการปรับปรุงค่าฟีโรโมนแบบเฉพาะบริเวณ(local pheromone update) เฉพาะพารามิเตอร์ลำดับที่ i ที่ได้เลือกมาด้วยสมการที่ (12) ในตารางพารามิเตอร์ทุกเส้นทางที่ n

$$\tau_{ni} = (1-\varphi).\tau_{ni} + \varphi.\tau_0 \tag{12}$$

โดย $\varphi \in (0,1]$ คือสัมประสิทธิ์การเสื่อมสลายของฟีโรโมนแบบเฉพาะบริเวณ

<u>ขั้นตอนที่ 4</u> มดตัวที่ k นำการเชื่อมต่อ c_{ni} ,ค่าน้ำหนัก w_{ni} จากพารามิเตอร์ที่เลือกมาและค่า threshold มาประกอบกันสร้างเป็นโครงข่ายประสาทเทียมดังรูปที่ 2



รูปที่ 2 โครงข่ายของมดตัวมดตัวที่ k

นำโครงข่ายที่ได้ไปทำการฝึกสอนด้วย BP แบบบางส่วนด้วยจำนวนรอบค่าหนึ่งกับชุดข้อมูล training set จากนั้นวัดค่าความผิดพลาด MSE_k (Mean of Square Error) ของโครงข่ายกับชุดข้อมูล validation set (ป้องกันการเกิดปัญหา overfitting)

<u>ขั้นตอนที่ 5</u> ทำซ้ำขั้นตอนที่ 2 – 4 จนครบมดทุกตัว (m ตัว)

<u>ขั้นตอนที่ 6</u> มดตัวที่มีค่า MSE ที่น้อยที่สุด (MSE_{best}) และทำการปรับปรุงค่าฟีโรโมนแบบครอบคลุมทั้งระบบ (global pheromone update) ในพารามิเตอร์ลำดับที่ i ในทุกเส้นทางที่ n ที่มดตัวนั้นได้เลือกมาโดยใช้สมการที่ (13)

$$\tau_{ni} \leftarrow \begin{cases} (1-\rho).\tau_{ni} + \rho.\Delta\tau_{ni} & \text{กรณีที่ } i \text{ คือพารามิเตอร์ที่ถูกเลือกโดยมดตัวที่มีค่า} MSE_{best}; \\ \tau_{ni} & \text{กรณีอื่นๆ;} \end{cases} \tag{13}$$

โดย $\rho \in (0,1]$ คือค่าสัมประสิทธิ์การเสื่อมสลายของฟีโรโมนแบบครอบคลุมทั้งระบบ ซึ่ง $\Delta\tau_{ni} = 1/MSE_{best}$ และมดตัวที่มีค่าความผิดพลาดที่ดีที่สุดจะนำค่าน้ำหนักที่ผ่านการฝึกสอนด้วย BP มาแทนที่ค่าเดิมในพารามิเตอร์ลำดับที่ i ที่ได้เลือกมาและค่า threshold ของมดนั้นก็จะนำไปใช้เป็นค่า threshold ใหม่ของโครงข่ายในรอบถัดไป

<u>ขั้นตอนที่ 7</u> ทำซ้ำขั้นตอนที่ 2 – 6 จนครบตามจำนวนรอบการทำงานของ ACS ที่ตั้งไว้

<u>ขั้นตอนที่ 8</u> เลือกใช้ค่าพารามิเตอร์และค่า threshold ของมดตัวที่มีค่าความผิดพลาดที่ดีสุด (MSE_{best}) ในรอบสุดท้ายของวิธี ACS มาสร้างเป็นโครงข่ายและนำไปทดสอบประสิทธิภาพกับชุดข้อมูลทดสอบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. การทดลอง

ในหัวข้อนี้จะเป็นการทดสอบประสิทธิภาพของวิธีที่นำเสนอเปรียบเทียบวิธี ANN มาตรฐานที่ฝึกสอนด้วย BP และวิธี ACO-BP ของงานวิจัย [1] ทั้งนี้ในงานวิจัย [1] ชุดข้อมูลที่นำมาทดสอบในส่วนฟังก์ชันของ Mackay ไม่มีการระบุถึงช่วงของข้อมูลที่นำมาใช้ในการฝึกสอนและทดสอบโครงข่ายที่ชัดเจน ส่วนชุดข้อมูล Herbicides ไม่ใช่ชุดข้อมูลมาตรฐานที่นิยมนำมาใช้ตามที่เคยปรากฏในงานวิจัยที่ผ่านมา ด้วยเหตุนี้ในงานวิจัยนี้จึงเลือกใช้ชุดข้อมูลมาตรฐาน PROBEN1 [2] ที่มีการนำมาใช้อย่างหลากหลายในงานวิจัยที่ผ่านมา [11,12,13] สำหรับงานวิจัยนี้ได้เลือกใช้ 3 ชุดข้อมูล คือ Cancer1, Diabetes1 และ Glass1 ทั้งนี้ทุกชุดข้อมูลจะแบ่งออกเป็นชุดข้อมูลสำหรับการฝึกสอน (training set) ชุดข้อมูลที่ใช้วัดประสิทธิภาพของโครงข่ายระหว่างทำการฝึกสอน (validation set) และข้อมูลที่ใช้สำหรับทดสอบโครงข่าย (test set) รายละเอียดของแต่ละชุดข้อมูลแสดงดังตารางที่ 1

ตารางที่ 1 รายละเอียดของชุดข้อมูลที่ใช้ทดลอง

ชุดข้อมูล	จำนวนโหนด		จำนวนรูปแบบข้อมูล			
	input	output	training	valida-tion	test	รวม
Cancer1	9	2	350	175	174	699
Diabetes1	8	2	384	192	192	768
Glass1	9	6	107	54	53	214

3.1 จำนวนรอบการทำงานของโครงข่าย

ในการทดลองของแต่ละวิธีจะกำหนดให้โครงข่ายประสาทเทียมมีจำนวนรอบการทำงานที่เท่ากัน เรียกว่า *TotalPasses* [10] สำหรับ ANN แบบมาตรฐานและวิธี ACO-BP [1] ค่าของ *TotalPasses* จะขึ้นอยู่กับจำนวนรอบการฝึกสอนโครงข่ายด้วย BP แต่สำหรับวิธีที่นำเสนอ ค่า *TotalPasses* จะคำนวณได้จากสมการ (14)

$$TotalPasses = m * T * epPartial \qquad (14)$$

ซึ่ง *epPartial* คือจำนวนรอบการใช้ BP ฝึกสอนโครงข่ายที่ได้มาจากมดแต่ละตัวแบบบางส่วน

3.2 การกำหนดค่าตัวแปรสำหรับการทดลอง

ในการทดลองของทุกชุดข้อมูล กำหนดให้โครงข่ายประสาทเทียมมี 1 ชั้นซ่อน และมีจำนวนโหนดในชั้นจำนวน 10 โหนด ใช้ฟังก์ชันกระตุ้นแบบ sigmoid มีอัตราการเรียนรู้เท่ากับ 0.01 ในส่วนของ ACO-BP และ ACS จะเลือกใช้ค่าโดยอ้างอิงกับงานวิจัย [1] ซึ่งกำหนดให้มีจำนวนมด (m) จำนวน 30 ตัว จำนวนรอบการทำงาน (T) 100 รอบ จำนวนพารามิเตอร์ในตารางพารามิเตอร์ (*parameter table*) 60 พารามิเตอร์, ช่วงของ [W_{min}, W_{max}] เท่ากับ [-2, 2], ค่า φ , ρ และ q_0 เท่ากับ 0.1, 0.1 และ 0.9 ตามลำดับ [8] สำหรับโครงข่าย ANN แบบมาตรฐานและโครงข่ายจากวิธี ACO-BP [1] จะกำหนดให้ฝึกสอนด้วย BP จำนวน 60,000 รอบ และวิธีที่นำเสนอจะใช้ BP ฝึกสอนโครงข่ายที่ได้มาจากมดแต่ละตัวแบบบางส่วน (*epPartial*) จำนวนรอบ 20 รอบ ดังนั้นวิธีที่นำเสนอจะมีค่า $TotalPasses = 30 \times 100 \times 20 = 60{,}000$ รอบเท่ากับอีกสองวิธี

3.3 การเลือกขนาดของตารางพารามิเตอร์และจำนวนมด

การเลือกขนาดของพารามิเตอร์ในตารางพารามิเตอร์หากมีน้อยเกินไปจะทำให้มีตัวเลือกของคำตอบไม่มากแต่ถ้ามีจำนวนมากก็จะต้องใช้พื้นที่ในการเก็บข้อมูลมากขึ้น ในส่วนจำนวนของมดหากมีน้อยเกินไปก็จะไม่สามารถค้นหาคำตอบได้หลากหลายแต่ถ้ามีมากเกินไปก็จะทำให้มีจำนวนของโครงข่ายในระบบมากขึ้นทำให้ต้องใช้เวลาในการทำงานเพิ่มขึ้นตามไปด้วย

3.4 ผลการทดลอง

ผลการทดลองเปรียบเทียบประสิทธิภาพในการจำแนกข้อมูลของชุดข้อมูลมาตรฐานทั้ง 3 ชุดข้อมูล ระหว่างโครงข่ายประสาทเทียมของทั้งสามวิธี ซึ่งทำการทดลองทั้งหมด 10 ครั้ง พบว่าโครงข่ายประสาทเทียมที่ได้จากวิธีที่นำเสนอให้เมื่อทดสอบกับข้อมูลทดสอบ (test set) พบว่ามีค่าความผิดพลาดเฉลี่ยน้อยกว่า ANN แบบมาตรฐานที่ฝึกสอนด้วย BP และวิธี ACO-BP ของ Y.P. Liu [1] ในทุกชุดข้อมูล ดังแสดงในตารางที่ 2

4. สรุป

จุดเด่นของวิธีที่นำเสนอนี้คือได้โครงข่ายที่มีความยืดหยุ่นคือการเชื่อมต่อของโหนดภายในสามารถเชื่อมต่อกันข้ามชั้นได้และสามารถลดโหนดที่ไม่ได้ใช้ทิ้งไปได้ วิธีที่นำเสนอได้ใช้ BP ในการฝึกสอนโครงข่ายแบบบางส่วนคือจำนวนรอบน้อยๆ เท่านั้น และใช้วิธี ACS ทำการค้นหา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2 ผลการทดสอบประสิทธิภาพกับข้อมูลทดสอบ (test set) ของชุดข้อมูล Cancer1 , Diabetes1และ Glass1

วิธี	ชุดข้อมูล Cancer1 The test set Classifier Error (%)				ชุดข้อมูล Diabetes1 The test set Classifier Error (%)				ชุดข้อมูล Glass1 The test set Classifier Error (%)			
	Max	Min	Mean	Stdev	Max	Min	Mean	Stdev	Max	Min	Mean	Stdev
ANN แบบมาตรฐาน	5.747	1.149	3.506	1.195	32.292	26.563	28.854	1.889	41.509	30.189	37.359	4.153
ACS-BP	5.747	1.149	3.678	1.489	29.167	23.958	26.615	1.812	39.623	30.189	35.094	2.977
วิธีที่นำเสนอ	2.874	0.575	1.609	0.757	28.646	21.875	25.834	1.774	35.849	28.302	32.076	2.950

ในส่วนที่เป็นคำตอบครอมคลุมทั้งระบบ ซึ่ง ACS สามารถปรับเปลี่ยนการเชื่อมต่อภายในและค่าน้ำหนักของโครงข่ายได้อย่างอัตโนมัติ จึงสามารถแก้ไขปัญหาของการติดปัญหาจุดเหมาะสมเฉพาะบริเวณ (local optimum) และปัญหาการเข้ากันกับชุดข้อมูลฝึกสอนมากเกินไป (overfitting) จึงทำให้โครงข่ายประสาทเทียมที่ได้ให้มีประสิทธิภาพมากขึ้น

5. เอกสารอ้างอิง

[1] Y.P. Liu, M.G. Wu, and J.X. QianH, ”Evolving Neural Networks Using the Hybrid of Ant Colony Optimization and BP Algorithms”, Advances in Neural Networks, vol. 3971, 2006, pp 714-722.

[2] Prechelt L, “Proben1 – a set of neural network benchmark problems and benchmarking rules”. Technical Report 21, Fakultät für Informatik, Umiversität Karlsruhe, Karlsruhe, Germany, 1994.

[3] A. K. Jain, J. Mao, and K. M. Mohiuddin. “Artificial neural networks: A tutorial, IEEE Computer, 29(3) :31-44, 1996.

[4] K.S. Tang, C.Y. Chan, K.F Man and S.Kwong “Genetic Structure for NN Topology and Weights Optimization” Genetic Algorithms in Engineering Systems: Innovations and Applications, Conference Publication No. 414, September 1995.

[5] V. Maniezzo, “Genetic Evolution of the Topology and Weight Distribution of Neural Networks”, IEEE Trans. on NN, 1994, pp. 39-53.

[6] W. Gao, “Study on new evolutionary neural network”,Proc. of ICLMC2003, IEEE, New York, 2003, pp. 1287-1293.

[7] S.J. Han, S.B. Cho, “Evolutionary Neural Networks for Anomaly Detection Based on the Behavior of a Program” IEEE Trans. On System, Man, And Cybernetics—Part B: Cybernetis, vol. 36, No. 3, June 2006, pp, 559-570

[8] M. Dorigo and L. M. Gambardella, “Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem” IEEE Trans. On Evolutionary Computation, vol. 1, no. 1, 1997, pp. 53-66.

[9] M. Dorigo, M. Birattari, and T. Stützle, “Ant Colony Optimization” IEEE Computational Intelligence Magazine, November 2006.

[10] T.Whitfort, D. Foster, and J. McCullagh, “Evolution versus training: An investigation into combining genetic algorithms and neural networks”, Proceedings ICONIP'99 6th International Conference on Neural Information Processing, 3:848-854, 1999.

[11] E. Alba and F. J. Chicano, “Training Neural Networks with GA Hybrid algorithms”, Int. Conf. GECCO, 2004.

[12] M. Matteucci, “ELeaRNT: Evolutionary Learning of Rich Neural Network Topologies”, Technical Report N CMU-CALD-02-103, Carnegie Mellon University, Pittsburgh PA, 2002.

[13] X. Yao and Y. Liu, “A new evolutionary system for evolving artificial neural networks,” IEEE Trans. Neural Networks, vol.8, no.3, pp.694-713,May 1997.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้