

วิธีการฝึกสอนโครงข่ายประสาทเทียมแบบป้อนผลการคำนวณไปข้างหน้า

โดยใช้หลักการสหสัมพันธ์ไขว้น้อยสุด

AN ALGORITHM FOR TRAINING THE ARTIFICIAL FEEDFORWARD NEURAL NETWORKS USING THE LEAST CROSS-CORRELATION PRINCIPLE

อักรพันธ์ วงศ์กังแห¹ คลหทัย กันนัย^{2*}

¹ ภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

² นักศึกษา หลักสูตรวิศวกรรมศาสตรมหาบัณฑิต สาขาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

บทคัดย่อ - บทความนี้ นำเสนอ วิธีการประยุกต์ใช้หลักการสหสัมพันธ์ไขว้น้อยสุดในการฝึกสอนโครงข่ายประสาทเทียมแบบป้อนผลการคำนวณไปข้างหน้า ในการฝึกสอนนั้นใช้หลักการ การรบกวนค่าน้ำหนัก โดยกำหนดให้ค่าการรบกวนค่าน้ำหนัก มีลักษณะการกระจายของตัวแปรสุ่มแบบเกาส์เซียน ทำการปรับค่าน้ำหนักและหาจุดเหมาะสมของค่าน้ำหนักในโครงข่ายโดยประยุกต์ใช้หลักการสหสัมพันธ์ไขว้น้อยสุดระหว่างผลต่างค่าความผิดพลาดกำลังสองกับค่าการรบกวน เราเรียกวิธีการที่เสนอนี้ว่า LCC อัลกอริทึม และได้ทำการทดสอบกับการฝึกสอนให้โครงข่าย สามารถทำงาน เป็นฟังก์ชัน เอ็กสพลูสิฟอว์ และ สามิตพาริตี พบว่า วิธีการที่เสนอสามารถใช้ฝึกสอนโครงข่าย เมื่อจำนวนชั้นโครงข่ายมีไม่เกินสี่ชั้น จำนวนเปอร์เซ็นต์ผลการฝึกที่ประสบผลสำเร็จ นั้นมากกว่าอัลกอริทึมฝึกสอนโครงข่ายประสาทเทียมที่รู้จักกันโดยทั่วไป แต่เปอร์เซ็นต์ผลการฝึกที่ประสบผลสำเร็จนั้นเป็นรอง อัลกอริทึม ลาเวนเบิร์กมาร์ควอด แต่ทั้งนี้ผลของคำตอบ จุดที่ไม่มีข้อมูลฝึกสอน วิธีการประยุกต์ใช้หลักการสหสัมพันธ์ไขว้น้อยสุด จะให้ผลของคำตอบจากการฝึกสอนที่สำเร็จในแต่ละครั้งของการฝึกสอนที่มีค่าใกล้เคียงกัน และแนวโน้มของคำตอบไปในทิศทางเดียวกัน มากกว่า การใช้ อัลกอริทึม ลาเวนเบิร์กมาร์ควอด ซึ่งการได้คำตอบไปในทิศทางเดียวกันนั้น เป็นคุณสมบัติที่ดี ที่ต้องการ เมื่อต้องการนำโครงข่ายประสาทเทียมมาประยุกต์ใช้กับระบบที่ต้องการการทำนาย คำตอบที่อยู่ ณ จุดที่ไม่มีข้อมูลฝึกสอน อีกทั้งวิธีการฝึกสอนที่เสนอนี้ใช้ทรัพยากรหน่วยความจำในการประมวลผลน้อยมากเมื่อเทียบกับวิธีการอื่นๆที่มีมาก่อน

คำสำคัญ - โครงข่ายประสาทเทียมแบบป้อนผลการคำนวณไปข้างหน้า, การรบกวนค่าน้ำหนัก, สหสัมพันธ์ไขว้น้อยสุด

ABSTRACT - This paper proposes an algorithm for training the artificial feedforward neural networks (ANN). The least cross-correlation (LCC) principle is applied for optimizing the weights of the neural networks. The weights of the network are perturbed with the Gaussian distribution random vectors. The cross-correlation matrix between squared-error difference and the perturbation vector is used for weight adjustment. The Exclusive-Or and 3 Bit Parity problems are selected for a primary evaluation of the algorithm. The results show that the LCC can outperform the most of well-known algorithms but it is inferior to the Levenberg-Marquardt algorithm (LM) for the case of training success which the networks are having 2 to 4 layers. The consistency and generalization properties of the LCC trained networks are much better than that of the LM trained ones. Although they are trained with different initial conditions, the LCC trained networks still produce similar output values when the input vectors are not in the training set. The LCC algorithm can be a promised one for the application of ANN on VLSI circuits including the small memory microcontrollers because it uses very little memory resource in the order of kilobytes.

Keywords - Artificial Feedforward Neural Network, Weight Perturbation, Least Cross- Correlation

1. บทนำ

ในการฝึกสอนโครงข่ายประสาทเทียม อัลกอริทึมเลเวนเบิร์ก มาร์ควอด (Levenberg-Marquardt Algorithm) [1] จัดว่าเป็นอัลกอริทึมที่ได้รับความนิยมเลือกใช้ฝึกสอนโครงข่ายประสาทเทียม เนื่องจากเป็นอัลกอริทึมที่สามารถประมวลผลได้รวดเร็วและมีความแม่นยำสูง [2] ซึ่งทำการปรับค่าน้ำหนักโดยอาศัยวิธีการของนิวตัน (Newton's Method) โดยคำนวณอนุพันธ์อันดับหนึ่งและเมตริกซ์อนุพันธ์อันดับสองของค่าความผิดพลาดกำลังสองเรียกว่า เกรเดียนท์ (Gradient) และเฮสเซียนเมตริกซ์ (Hessian Matrix) ตามลำดับ การฝึกสอนโครงข่าย ด้วยอัลกอริทึมเลเวนเบิร์กมาร์ควอดนั้น มีขั้นตอนการประมวลผลที่ซับซ้อนและใช้ทรัพยากรในการประมวลผลมาก เมื่อเปรียบเทียบกับวิธีการฝึกสอนโครงข่ายด้วยอัลกอริทึมอื่นๆ ที่อยู่บนพื้นฐานการคำนวณค่าเกรเดียนท์ เช่น Gradient descent backpropagation (GD), Gradient descent with adaptive learning rule backpropagation (GDA) และ Gradient descent with momentum backpropagation (GDM) เป็นต้น การฝึกสอนโครงข่ายโดยการคำนวณค่าเกรเดียนท์ จะอาศัยการเปลี่ยนแปลงค่าความผิดพลาดกำลังสองซึ่ง พบว่า เมื่อค่าความผิดพลาดกำลังสองไม่มีการเปลี่ยนแปลง วิธีการปรับค่าน้ำหนักโดยการคำนวณค่าเกรเดียนท์จะไม่สามารถปรับค่าน้ำหนักได้ ดังนั้น เพื่อให้โครงข่ายยังคงสามารถปรับค่าน้ำหนัก เพื่อให้ได้ค่าน้ำหนักที่เหมาะสม การปรับค่าน้ำหนักโดยหลักการ การรบกวนค่าน้ำหนักแบบสุ่ม ได้แก่ วิธี Linear Random Search Algorithm [3] วิธี Weight Perturbation [4][5][6][7][8][9][10][11] วิธีการรบกวนค่าน้ำหนักแบบสุ่มที่ถูกนำเสนอเหล่านี้ มีหลักการทำงานโดยทั่วไป คือ การสุ่มค่าน้อยๆ รบกวนค่าน้ำหนักของโครงข่าย ส่งผลให้ค่าความผิดพลาดกำลังสองมีค่าเปลี่ยนแปลง จากนั้นจึงประมาณค่าเกรเดียนท์ ได้จากการเปลี่ยนแปลงของค่าความผิดพลาดกำลังสอง การปรับค่าน้ำหนักด้วยวิธีนี้ มีขั้นตอนการประมวลผลในแต่ละรอบ (epoch) น้อยมาก ดังนั้น การปรับค่าน้ำหนักด้วยหลักการการรบกวนค่าน้ำหนัก จึงถูกนำไปประยุกต์ใช้ในการสร้างโครงข่ายประสาทเทียมด้วยอุปกรณ์อนาล็อก (analog hardware) เนื่องจากอุปกรณ์เหล่านี้มีหน่วยความจำน้อย แต่สามารถประมวลผลได้ด้วยความเร็วสูง จากรายงาน

ทางวิชาการที่ผ่านมา พบว่า การปรับค่าน้ำหนักด้วยวิธีการรบกวนค่าน้ำหนัก จะกำหนดให้ค่าการรบกวนเป็นค่าน้อยๆ หรือกำหนดขนาด (amplitude) ของค่าการรบกวนเป็นค่าคงที่ ซึ่งเป็นข้อจำกัดในการหาค่าน้ำหนักที่เหมาะสมของอัลกอริทึมเหล่านั้น

ดังนั้น ในบทความนี้ จึงนำเสนออัลกอริทึมฝึกสอนโครงข่ายประสาทเทียม ซึ่งปรับค่าน้ำหนักด้วยหลักการ การรบกวนค่าน้ำหนัก โดยกำหนดให้ขนาดของค่าการรบกวนมีลักษณะการกระจายตัวแปรสุ่มแบบเกาส์เซียน มีพารามิเตอร์ควบคุมการกระจายตัวของขนาดการรบกวน คือ ค่าเบี่ยงเบนมาตรฐาน (Standard Deviation) ของขนาดการรบกวน และใช้วิธีสหสัมพันธ์ไขว้น้อยสุด (Least Cross-Correlation, LCC) ของผลต่างระหว่างค่าความผิดพลาดกำลังสองกับค่าการรบกวน เพื่อปรับค่าน้ำหนักให้ได้ค่าน้ำหนักเหมาะสมที่สุดที่แท้จริง

แนวคิดและทฤษฎีพื้นฐานการใช้วิธีสหสัมพันธ์ไขว้น้อยสุดในการปรับค่าน้ำหนักได้มีรายงานการศึกษาไว้แล้วในบทความ A Global Optimization Method Using Gaussian Distribution Perturbation [12] หรือกล่าวได้ว่าบทความฉบับนี้เป็น การนำวิธีการที่ศึกษาไว้ในบทความ [12] มาประยุกต์ใช้เพื่อฝึกสอนโครงข่ายประสาทเทียม

จากการทดลอง พบว่า การกำหนดค่าเบี่ยงเบนมาตรฐานที่เหมาะสมให้กับค่าการรบกวน จะทำให้โครงข่ายสามารถหาค่าน้ำหนักเหมาะสมที่สุดได้รวดเร็วและแม่นยำ แต่ทั้งนี้เนื่องจากข้อจำกัดของความยาวของเนื้อเรื่องในบทความนี้ เราจะนำเสนอหลักการของวิธีการและข้อแตกต่างโดยทั่วไปของผลลัพธ์การฝึกสอนเทียบกับอัลกอริทึมที่เป็นที่รู้จักดีก่อน ส่วนการศึกษาผลของการแปรผันพารามิเตอร์การเรียนรู้และการกระจายตัวของขนาดของค่าการรบกวน ที่มีผลต่อประสิทธิภาพและประสิทธิผลที่ดีของโครงข่ายจะรายงานในบทความฉบับหนึ่งข้างหน้า

2. วิธีการฝึกสอนโดยใช้หลักการสหสัมพันธ์ไขว้น้อยสุด

วิธีการฝึกสอนโดยใช้หลักการสหสัมพันธ์ไขว้น้อยสุด LCC ที่เสนอในที่นี้มีขั้นตอนตามลำดับสามารถเขียนได้เป็นขั้นตอนที่ 1 ถึง 7 และสามารถแสดงเป็นสมการที่ (1) ถึง (7) ตามที่แสดงไว้ดังต่อไปนี้

ขั้นที่ 1 ทำการคำนวณค่าความผิดพลาด กำลังสองของ
โครงข่าย ดังสมการที่(1) เพื่อเก็บไว้ใช้ตรวจสอบ
การลู่เข้าสู่ค่าตอบ และใช้ในเงื่อนไขที่จะหยุดฝึก
สอน เมื่อค่าความผิดพลาดกำลังสองเฉลี่ยเป็นที่
ยอมรับ

$$\varepsilon(\mathbf{w}_k) = \sum_{p=1}^N (t_p - y_{p \mathbf{w}_k})^2 \quad (1)$$

ขั้นที่ 2 ทำการสุ่มค่าเวกเตอร์ $\Delta \mathbf{P}$

ขั้นที่ 3 ทำการรบกวนค่าน้ำหนักของโครงข่ายด้วย $v \cdot \Delta \mathbf{P}$ ทั้ง
ด้านบวกและด้านลบดังแสดง ในสมการ (2) และ (3)

$$\mathbf{w}_+ = \mathbf{w}_k + v \cdot \Delta \mathbf{P} = [w_{+1} \ w_{+2} \ \dots \ w_{+i}]^T \quad (2)$$

$$\mathbf{w}_- = \mathbf{w}_k - v \cdot \Delta \mathbf{P} = [w_{-1} \ w_{-2} \ \dots \ w_{-i}]^T \quad (3)$$

ขั้นที่ 4 ทำการคำนวณ ค่าความผิดพลาดกำลังสองของ
โครงข่ายที่ถูกรบกวนค่าน้ำหนักด้วย $v \cdot \Delta \mathbf{P}$ ทั้งด้าน
บวกและด้านลบดังแสดง ในสมการ (4) และ (5)

$$\varepsilon(\mathbf{w}_+) = \sum_{p=1}^N (t_p - y_{p \mathbf{w}_+})^2 \quad (4)$$

$$\varepsilon(\mathbf{w}_-) = \sum_{p=1}^N (t_p - y_{p \mathbf{w}_-})^2 \quad (5)$$

ขั้นที่ 5 ทำการคำนวณค่าผลต่างของค่าความผิดพลาดกำลัง
สอง ($\Delta \varepsilon$), คำนวณได้ดังสมการที่ (6)

$$\Delta \varepsilon = \varepsilon(\mathbf{w}_+) - \varepsilon(\mathbf{w}_-) \quad (6)$$

ขั้นที่ 6 ทำการปรับค่าน้ำหนักโครงข่าย โดยใช้หลักการ
สหสัมพันธ์ไขว้น้อยสุด ระหว่างผลต่างค่าความ
ผิดพลาดกำลังสอง ($\Delta \varepsilon$) กับค่าการรบกวน ($v \cdot \Delta \mathbf{P}$) ดัง
สมการที่ (7)

$$\mathbf{w}_{k+1} = \mathbf{w}_k - \mu \cdot \Delta \varepsilon \cdot v \Delta \mathbf{P} \quad (7)$$

ขั้นที่ 7 ย้อนกลับไปทำขั้นที่ 1

โดยที่สัญลักษณ์ในสมการมีความหมายดังที่แสดงไว้
ดังต่อไปนี้

$\varepsilon(\mathbf{w}_k)$ คือ ผลรวมความผิดพลาดกำลังสอง เมื่อค่าน้ำหนัก คือ $\mathbf{w}_k$

$\varepsilon(\mathbf{w}_+)$ คือ ผลรวมความผิดพลาดกำลังสอง เมื่อค่าน้ำหนัก คือ $\mathbf{w}_+$

$\varepsilon(\mathbf{w}_-)$ คือ ผลรวมความผิดพลาดกำลังสอง เมื่อค่าน้ำหนัก คือ $\mathbf{w}_-$

i คือ จำนวนสมาชิกในเวกเตอร์ค่าน้ำหนัก

K คือ ครั้งที่ปรับค่าน้ำหนัก 1,2,3,...

$\mathbf{w}_k$ คือ เวกเตอร์ค่าน้ำหนักก่อนปรับค่า

$$\mathbf{w}_k = [w_{k1} \ w_{k2} \ \dots \ w_{ki}]^T$$

$\mathbf{w}_{k+1}$ คือ เวกเตอร์ค่าน้ำหนักหลังถูกปรับค่า

$\mathbf{w}_+$ คือ เวกเตอร์ค่าน้ำหนักที่ถูกรบกวนด้านบวก

$$\mathbf{w}_+ = \mathbf{w}_k + v \cdot \Delta \mathbf{P} = [w_{+1} \ w_{+2} \ \dots \ w_{+i}]^T$$

$\mathbf{w}_-$ คือ เวกเตอร์ค่าน้ำหนักที่ถูกรบกวนด้านลบ

$$\mathbf{w}_- = \mathbf{w}_k - v \cdot \Delta \mathbf{P} = [w_{-1} \ w_{-2} \ \dots \ w_{-i}]^T$$

t_p คือ ค่าเอาต์พุตที่ต้องการ ของข้อมูลอินพุตฝึกสอน ลำดับที่ p
ค่าน้ำหนัก คือ $\mathbf{w}_{+k}$ ฝึกสอนโครงข่าย

$y_{p \mathbf{w}_+}$ คือ ค่าเอาต์พุตโครงข่ายของข้อมูลอินพุตฝึกสอน ลำดับที่ p
เมื่อ ค่าน้ำหนัก คือ $\mathbf{w}_+$

$y_{p \mathbf{w}_-}$ คือ ค่าเอาต์พุตโครงข่ายของข้อมูลอินพุตฝึกสอน ลำดับที่ p
เมื่อ ค่าน้ำหนัก คือ $\mathbf{w}_-$

$y_{p \mathbf{w}_k}$ คือ ค่าเอาต์พุตโครงข่ายของข้อมูลอินพุตฝึกสอน ลำดับที่ p
เมื่อ ค่าน้ำหนัก คือ $\mathbf{w}_k$

N คือ จำนวนอินพุตฝึกสอนทั้งหมด

v คือ ค่าคงที่ควบคุมลักษณะการกระจายตัวแปรสุ่ม

$\Delta \mathbf{P}$ คือ เวกเตอร์การรบกวน มีลักษณะการกระจายตัวแปรสุ่ม
แบบเกาส์เซียนปกติ (Normal Gaussian Distribution) มีค่า
เบี่ยงเบนมาตรฐานเท่ากับ 1.0

$v \cdot \Delta \mathbf{P}$ คือ เวกเตอร์ค่าการรบกวน มีลักษณะการกระจายตัวแปรสุ่ม
เกาส์เซียน (Gaussian Distribution) มีค่า
เบี่ยงเบนมาตรฐาน เท่ากับ v

$\Delta \varepsilon$ คือ ผลต่างระหว่างค่าความผิดพลาดกำลังสองของค่าน้ำหนัก
ที่ถูกรบกวนด้านบวก ($\mathbf{w} + v \cdot \Delta \mathbf{P}$) และค่าความ
ผิดพลาดกำลังสองของค่าน้ำหนักที่ถูกรบกวนด้านลบ
($\mathbf{w} - v \cdot \Delta \mathbf{P}$)

μ คือ ค่าอัตราการเรียนรู้

จากสมการ (7) เป็นสมการปรับค่าน้ำหนัก โดยใช้วิธี
สหสัมพันธ์ไขว้น้อยสุดระหว่างผลต่างค่าความผิดพลาดกำลัง
สอง ($\Delta \varepsilon$) กับค่าการรบกวน ($v \cdot \Delta \mathbf{P}$) จะเห็นว่าในทุกขั้นตอน
ก่อนการปรับค่าน้ำหนักโครงข่าย จะไม่มีการคำนวณค่า
อนุพันธ์ใดๆ ซึ่งทำให้ขั้นตอนการประมวลผลไม่ซับซ้อนเมื่อ
เปรียบเทียบกับอัลกอริทึมอื่นๆ ที่ปรับค่าน้ำหนักโดยการ
อนุพันธ์เพื่อคำนวณค่าเกรเดียนท์ จึงทำให้ประหยัดทรัพยากร
ในการคำนวณ เมื่อนำสมการที่ (7) มาหาค่าเฉลี่ยหรือค่า
คาดหวัง (Expect Value) ดังสมการที่ (8)

$$\mathbf{E}[\mathbf{w}_{k+1}] = \mathbf{E}[\mathbf{w}_k - \mu \cdot \Delta \varepsilon \cdot v \Delta \mathbf{P}] \quad (8)$$

จากสมการที่ (8) ในขณะที่ทำการพิจารณานี้ เรากำลัง
พิจารณาที่จุด $\mathbf{w}_k$ จุดใดจุดหนึ่ง ดังนั้นเราจะเห็นว่าจุด $\mathbf{w}_k$ นั้น
เป็นตัวแปรคงที่อิสระไม่ขึ้นอยู่กับ $\mu \cdot \Delta \varepsilon \cdot v \Delta \mathbf{P}$ และเมื่อ

พิจารณาพจน์ $\mu \cdot \Delta \varepsilon \cdot v \Delta P$ จะเห็นว่า ตัวแปร μ นั้น มีค่าคงที่ไม่เปลี่ยนแปลง ดังนั้น สมการที่ (8) จึงถูกเขียนใหม่ดังสมการที่ (9)

$$E[w_{k+1}] = E[w_k] - \mu \cdot E[\Delta \varepsilon \cdot v \Delta P] \quad (9)$$

เมื่อพิจารณาสมการที่ (9) เราจะสังเกตว่าเมื่อสหสัมพันธ์ระหว่าง $\Delta \varepsilon$ และ $v \Delta P$ นั้นมีค่าเป็นศูนย์สามารถเขียนเป็นสมการที่ (10)

$$E[\Delta \varepsilon \cdot v \Delta P] = 0 \quad (10)$$

นำสมการที่ (10) แทนลงใน (9) จะเป็นผลให้ การปรับค่าน้ำหนักนั้นไม่ขึ้นอยู่กับกระบวนการบวกรวม แสดงในสมการที่ (11)

$$E[w_{k+1}] = E[w_k] \quad (11)$$

ที่จุดนี้เราค่าหวังว่า w_{k+1} จะมีค่าเท่ากับ w_k และเราคาดหวังว่า $E[\Delta \varepsilon \cdot v \Delta P]$ จะมีค่าเป็นศูนย์ ซึ่งนั่น คือ ผลการบวกรวมค่าน้ำหนักทั้งที่เป็นทางด้านบวกและทางด้านลบให้ผลของค่าความผิดพลาดกำลังสองมีค่าเท่ากัน หรือ $\varepsilon(w_+) = \varepsilon(w_-)$ เป็นผลทำให้ค่า $\Delta \varepsilon$ มีค่าเท่ากับศูนย์

ถ้าหากว่าเราพิจารณาในกรณีนี้ $\Delta \varepsilon \cdot v \Delta P$ มีค่าเป็นบวกย่อมแสดงว่า การบวกรวมค่าน้ำหนักในด้านบวก เป็นผลให้เกิด $\varepsilon(w_+)$ มีค่ามากกว่า $\varepsilon(w_-)$ ที่เกิดจากการบวกรวมค่าน้ำหนักในด้านลบ จากสมการที่ (7) จะเห็นว่า ในกรณีนี้ ค่าน้ำหนักจะถูกปรับให้ลดค่าลงเท่ากับ $\mu \cdot \Delta \varepsilon \cdot v \Delta P$ เพราะเราทราบว่า เมื่อ $\Delta \varepsilon \cdot v \Delta P$ เป็นบวก นั้นเกิดจากการบวกรวมค่าน้ำหนักด้านบวก เป็นผลให้เกิดค่าความผิดพลาดกำลังสองมากกว่าการบวกรวมค่าน้ำหนักด้านลบ $\varepsilon(w_+) > \varepsilon(w_-)$ จะเห็นว่าการทำเช่นนี้ เป็นวิธีการหาค่าเหมาะสมอีกวิธีการหนึ่ง โดยที่ค่าน้ำหนักนี้ จะถูกปรับเข้าหาค่า ๆ หนึ่ง ซึ่งเป็นผลให้ค่าสหสัมพันธ์ไขว้ระหว่าง $\Delta \varepsilon$ และ $v \Delta P$ มีค่าเป็นศูนย์ ด้วยเหตุนี้เองเรา จึงเรียกวิธีการที่ใช้ในการปรับค่าน้ำหนักให้กับโครงข่ายประสาทเทียม แบบนี้ว่า วิธีการสหสัมพันธ์ไขว้น้อยสุดหรือเรียกย่อ ๆ ได้เป็น LCC (Least Cross-Correlation) ขอให้สังเกตว่า การลู่เข้าสู่ค่าที่เหมาะสมที่สุด (Optimum Point) ซึ่งเป็นคำตอบของการหาค่าเหมาะสมที่สุดนั้น ไม่ใช่คำตอบเดียวกับการใช้วิธีการที่อยู่บนพื้นฐานของการคำนวณค่าอนุพันธ์ คำตอบของวิธีการที่เสนอนี้จะอยู่ ณ จุดที่ $E[\Delta \varepsilon \cdot v \Delta P] = 0$ ไม่ใช่จุดที่ ค่าเกรเดียนต์ หรือ $\nabla = \frac{\partial \varepsilon}{\partial w} = 0$ คำตอบที่แตกต่างกันนี้ จะแสดงให้เห็นเพิ่มเติมในหัวข้อถัดไป

3. วิธีการทดลองและผลการทดลอง

เพื่อเปรียบเทียบประสิทธิภาพของอัลกอริทึม LCC จึงทำการทดสอบฝึกสอนโครงข่ายด้วยอัลกอริทึม LCC เปรียบเทียบกับอัลกอริทึม ดังต่อไปนี้

- LM (Levenberg-Marquardt backpropagation)
- GDA (Gradient descent with adaptive learning rule backpropagation)
- GD (Gradient descent backpropagation)

อัลกอริทึม LM, GDA และ GD นั้น เป็นเครื่องมือ (Toolbox) ในโปรแกรม MATLAB [13] โดยในการทดสอบ จะทำการฝึกสอนโครงข่ายจากทั้ง 4 อัลกอริทึม โดยในแต่ละอัลกอริทึมที่ทดสอบจะมีสถาปัตยกรรมที่เหมือนกัน โดยที่แต่ละสถาปัตยกรรมของแต่ละอัลกอริทึมจะถูกฝึกสอนจำนวน 10 รอบ โดยกำหนดเงื่อนไขการฝึกสอนโครงข่ายที่ประสบความสำเร็จ คือ มีค่าเฉลี่ยความผิดพลาดกำลังสอง (Mean Square Error:MSE) น้อยกว่าหรือเท่ากับ 0.001 และมีจำนวนรอบการฝึกสอนไม่เกิน 60,000 รอบ (epoch) และถือว่า การฝึกสอนโครงข่ายที่มีค่าเฉลี่ยความผิดพลาดกำลังสองมากกว่า 0.001 และค่ารอบการฝึกสอนโครงข่ายมากกว่า 60,000 รอบ เป็นการฝึกสอนที่ไม่ประสบความสำเร็จ

สถาปัตยกรรมของโครงข่ายหรือเรียกย่อ ๆ ว่า โครงข่าย โดยโครงข่ายที่จะใช้ทดสอบการฝึกสอนของทั้ง 4 อัลกอริทึมอธิบาย ดังนี้ โครงข่ายแบบ 3 ชั้น เขียนแทนได้เป็น (a,b,c) และโครงข่ายแบบ 4 ชั้น เขียนแทนได้เป็น (a,b,c,d) เมื่อ a,b,c และ d เป็นจำนวนโนดในแต่ละชั้น อธิบายได้ดังรูปที่ 1(a) และ 1(b) ตามลำดับ

สำหรับปัญหาและ โครงข่ายที่จะทดสอบการฝึกสอนของทั้ง 4 อัลกอริทึม เป็นดังนี้

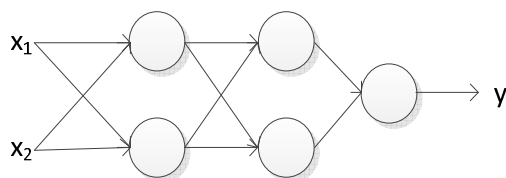
- ปัญหา Exclusive-OR โครงข่ายที่ใช้ทดสอบ ได้แก่ (4,1), (4,4,1), (4,4,4,1), (4,4,4,4,1)
- ปัญหา 3-bit parity โครงข่ายที่ใช้ทดสอบ ได้แก่ (3,1), (5,3,1), (5,3,3,1), (5,3,3,3,1)

ผลการทดสอบจากการฝึกสอนโครงข่ายของทั้ง 4 อัลกอริทึม แสดงเป็นตาราง แบ่งเป็นคอลัมน์ที่ 1 – คอลัมน์ที่ 7 ซึ่งแต่ละคอลัมน์สามารถอธิบาย ได้ดังนี้ คอลัมน์ 1 แสดงชื่ออัลกอริทึมที่ใช้ฝึกสอนโครงข่าย คอลัมน์ 2 แสดงสถาปัตยกรรมโครงข่ายที่ใช้ทดสอบการฝึกสอนโครงข่าย คอลัมน์ 3 แสดงค่าเวลาเฉลี่ยการฝึกสอนโครงข่ายของแต่ละอัลกอริทึมที่ประสบความสำเร็จ คอลัมน์ 4 แสดงค่ารอบการ

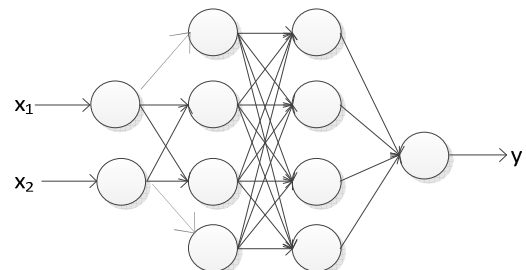
ฝึกสอนโครงข่ายโดยเฉลี่ยของแต่ละอัลกอริทึมที่ประสบความสำเร็จ คอลัมน์ 5 แสดงค่าความสำเร็จการฝึกสอนโครงข่ายของแต่ละอัลกอริทึม คอลัมน์ 6 แสดงจำนวนเวลาที่ใช้ฝึกสอนแต่ละรอบ (Epoch) และคอลัมน์ที่ 7 แสดงหน่วยความจำที่แต่ละอัลกอริทึมใช้ในการฝึกสอนโครงข่าย

ผลการทดสอบการฝึกสอนโครงข่ายด้วยอัลกอริทึม LM, LCC, GDA และ GD แสดงดังตารางที่ 1 และ 2 ตามลำดับ จากผลการทดสอบการฝึกสอนโครงข่ายทั้งสองตาราง จะเห็นว่า ค่าความสำเร็จของการฝึกสอนโครงข่ายด้วยอัลกอริทึม LCC มีค่าเท่ากับ 100 % ในการฝึกสอนโครงข่ายด้วยปัญหาเอ็กสคลูสิฟออร์ (Exclusive – Or) และในปัญหา สามบิตพาริตี (3 Bit Parity) การฝึกสอนโครงข่ายด้วยอัลกอริทึม LCC จะสามารถฝึกสอนโครงข่ายได้ดีก็ต่อเมื่อโครงข่ายมีจำนวนชั้นในโครงข่าย ตั้งแต่ 2-4 ชั้น ส่วนโครงข่ายที่มีชั้นใน

โครงข่ายมากกว่า 4 ชั้น พบว่า ความสำเร็จของการฝึกสอนโครงข่ายด้วย อัลกอริทึม LCC นั้น มีค่าลดลงเป็น 60 % ส่วนการฝึกสอนโครงข่ายด้วยอัลกอริทึม LM พบว่า ค่าความสำเร็จของการฝึกสอนโครงข่ายเท่ากับ 100 % ทุกโครงข่ายทั้งในกรณีของปัญหา Exclusive – Or และ 3 Bit Parity และการฝึกสอนโครงข่ายด้วยอัลกอริทึม GDA และ GD นั้น พบว่า ค่าความสำเร็จการฝึกสอนโครงข่ายมีค่ามากขึ้นเมื่อจำนวนชั้นในโครงข่ายเพิ่มขึ้น นอกจากนั้น จากผลการทดสอบการฝึกสอนโครงข่ายด้วยทั้ง 4 อัลกอริทึม ที่แสดงในตารางที่ 1 และ 2 พบว่า เวลาโดยเฉลี่ยที่ใช้ฝึกสอนโครงข่ายของอัลกอริทึม LM นั้นมีค่าน้อยที่สุด รองลงมาคือ GDA , LCC และ GD ตามลำดับ ซึ่งสอดคล้องกับค่ารอบเฉลี่ยการฝึกสอนโครงข่าย ซึ่งพบว่า ค่ารอบเฉลี่ยของ LM นั้นยังคงมีค่าน้อยที่สุดเช่นกัน รองลงมา คือ GDA , LCC และ GD ตามลำดับ



(a) (a=2,b=2,c=1)



(b) (a=2,b=4,c=4,d=1)

รูปที่ 1 แสดงตัวอย่างของสถาปัตยกรรมของโครงข่าย

(a) โครงข่ายแบบ 3 ชั้น

(b) โครงข่ายแบบ 4 ชั้น

ตารางที่ 1 ผลการทดสอบการทำงานกับปัญหา Exclusive-or

1	2	3	4	5	6	7
Algorithm Name	Network #Node	Ave. Time(sec)	Ave. Epoch	Time/Epoch (sec)	Success. (%)	Used Memory. (Byte)
LM	4,1_[5]	0.3	3	0.100	100	41647
GDA	4,1_[5]	6.5	895	0.007	100	41465
GD	4,1_[5]	4.1	974	0.004	90	28499
LCC	4,1_[5]	0.7	597	0.001	100	4880
LM	4,4,1_[9]	0.4	4	0.100	100	51741
GDA	4,4,1_[9]	4.7	539	0.009	100	51532
GD	4,4,1_[9]	5.2	1127	0.005	90	35434
LCC	4,4,1_[9]	1.6	788	0.002	100	7192
LM	4,4,4,4_[13]	0.4	4	0.100	100	61901
GDA	4,4,4,4_[13]	1.3	115	0.011	100	61719
GD	4,4,4,4_[13]	35.7	6240	0.006	100	42441
LCC	4,4,4,4_[13]	3.6	1309	0.003	100	9496
LM	4,4,4,4,1_[17]	0.4	4	0.100	100	72208
GDA	4,4,4,4,1_[17]	1.1	80	0.013	100	72026
GD	4,4,4,4,1_[17]	42.9	6836	0.006	100	49441
LCC	4,4,4,4,1_[17]	23.6	6814	0.003	100	11800

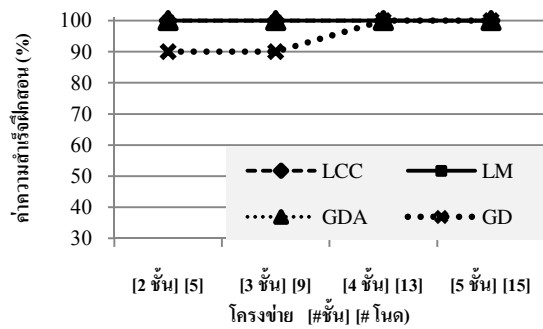
ตารางที่ 2 ผลการทดสอบการทำงานกับปัญหา 3 bit parity

1	2	3	4	5	6	7
Algorithm Name	Network #Node	Ave. Time(sec)	Ave. Epoch	Time/Epoch (sec)	Success. (%)	Used Memory (Byte)
LM	3,1_[4]	2.7	238	0.011	80	41663
GDA	3,1_[4]	51.3	3778	0.013	100	41481
GD	3,1_[4]	100.0	21103	0.005	20	28499
LCC	3,1_[4]	18	9195	0.002	100	3592
LM	5,3,1_[9]	2.2	144	0.015	100	51882
GDA	5,3,1_[9]	11.2	1434	0.012	100	51700
GD	5,3,1_[9]	105	18676	0.006	80	35602
LCC	5,3,1_[9]	45.0	12069	0.004	100	5560
LM	5,3,3,1_[12]	1.0	48	0.021	100	61869
GDA	5,3,3,1_[12]	9.3	719	0.011	100	61687
GD	5,3,3,1_[12]	94.0	13093	0.007	90	42409
LCC	5,3,3,1_[12]	80.0	16377	0.005	100	6856
LM	5,3,3,3,1_[15]	0.4	21	0.033	100	71976
GDA	5,3,3,3,1_[15]	7.8	696	0.010	100	71749
GD	5,3,3,3,1_[15]	0.7	9301	0.008	90	49288
LCC	5,3,3,3,1_[15]	472.0	18892	0.024	60	8152

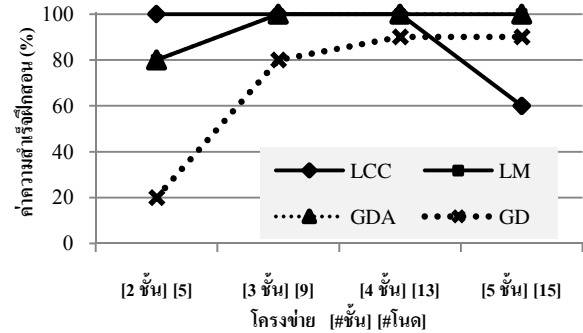
จากข้อมูลในตารางที่ 1 และ 2 ได้นำค่าเวลาที่ใช้ฝึกสอนโครงข่ายและจำนวนรอบเฉลี่ยการฝึกสอนโครงข่าย มาคำนวณเวลาเฉลี่ยที่ใช้ฝึกสอนโครงข่ายในแต่ละรอบ ของทั้ง 4 อัลกอริทึม ถึงแม้ว่า อัลกอริทึม LM จะมีค่ารอบเฉลี่ยการฝึกสอนโครงข่ายน้อยที่สุด แต่เมื่อคำนวณค่าเวลาที่ใช้ฝึกสอนโครงข่ายในแต่ละรอบ แล้ว พบว่า LM นั้น ใช้เวลาฝึกสอนโครงข่ายในแต่ละรอบ มากที่สุด รองลงคือ GDA , GD และ LCC ตามลำดับ ที่เป็นเช่นนี้ เนื่องจาก ในการฝึกสอนโครงข่ายด้วยอัลกอริทึม LM นั้น จำเป็นต้องมีการคำนวณค่าอนุพันธ์อันดับหนึ่งและเมตริกซ์อนุพันธ์อันดับสองของค่าความผิดพลาดกำลังสอง เรียกว่า เกรเดียนท์ (Gradient) และ เฮเซียนเมตริกซ์ (Hessian Matrix) ตามลำดับ มีขั้นตอนการประมวลผลที่ซับซ้อนกว่า GDA, GD และ LCC ส่วนการฝึกสอนโครงข่ายด้วยอัลกอริทึม GDA และ GD นั้น เป็นอัลกอริทึมที่อยู่บนพื้นฐานการคำนวณค่าเกรเดียนท์ แต่ไม่มีการคำนวณค่าเมตริกอนุพันธ์อันดับสอง ทำให้ใช้เวลาที่ใช้ฝึกสอนโครงข่ายในแต่ละรอบ น้อยกว่า LM และเมื่อ

เปรียบเทียบเวลาที่ใช้ฝึกสอนโครงข่ายในแต่ละรอบ ระหว่าง GDA และ GD แล้ว พบว่า GDA นั้นใช้เวลาฝึกสอนโครงข่ายในแต่ละรอบ มากกว่า GD เนื่องจากการฝึกโครงข่ายด้วยอัลกอริทึม GDA เป็นอัลกอริทึมฝึกสอนโครงข่าย ที่มีการกำหนดให้ค่าอัตราการเรียนรู้สามารถปรับตัวเองได้ขณะที่ฝึกสอนโครงข่าย ทำให้ใช้เวลาในการฝึกสอนโครงข่ายในแต่ละรอบเพิ่มมากขึ้น ซึ่งค่าอัตราการเรียนรู้ของอัลกอริทึม GD นั้น ถูกกำหนดเป็นค่าคงที่ และไม่มีการปรับตัวเองในขณะที่ฝึกสอนโครงข่าย เช่นเดียวกับ LCC และ LM และจากตารางที่ 1 และ 2 พบว่า การฝึกสอนโครงข่ายด้วย LCC นั้น ใช้เวลาฝึกสอนในแต่ละรอบน้อยที่สุด เนื่องมาจาก ในขั้นตอนการประมวลผลนั้นไม่มีการคำนวณอนุพันธ์ใด ๆ แต่ใช้วิธีสหสัมพันธ์ไขว้น้อยสุด (LCC) ในการฝึกสอนโครงข่าย ดังที่กล่าวมาแล้วในหัวข้อที่ 2

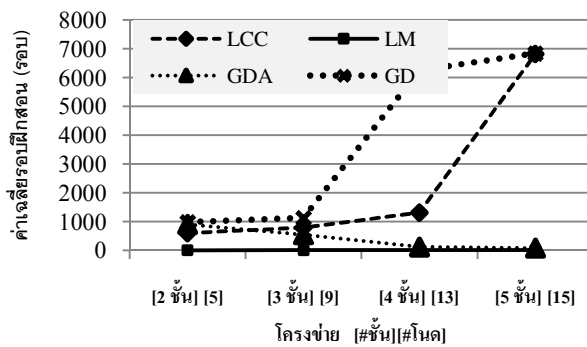
เพื่อให้มีความเข้าใจการฝึกสอนโครงข่ายทั้ง 4 อัลกอริทึมมากขึ้น ได้นำข้อมูลในตารางที่ 1 และ 2 มาแสดงเป็นกราฟในรูปที่ 2 ถึง รูปที่ 9 ดังนี้



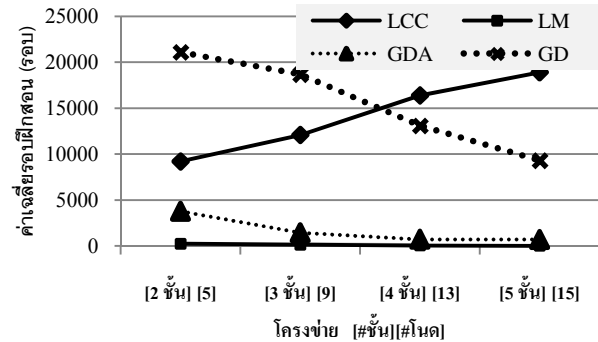
รูปที่ 2 ความสัมพันธ์ระหว่างความสำเร็จการฝึกสอนและโครงข่าย ปัญหา Exclusive – Or



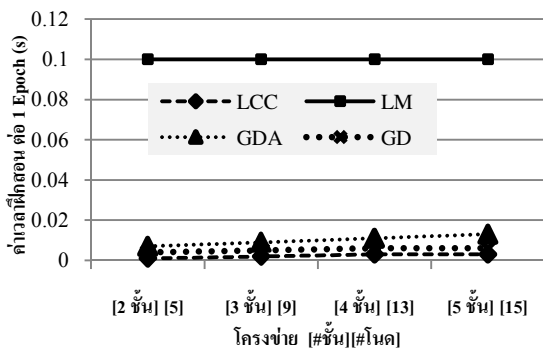
รูปที่ 3 ความสัมพันธ์ระหว่างความสำเร็จการฝึกสอนและโครงข่าย ปัญหา 3-bit parity



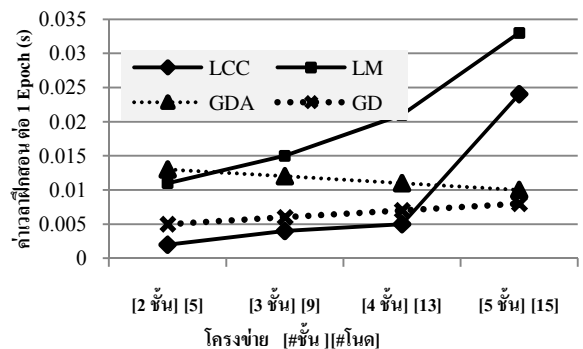
รูปที่ 4 ความสัมพันธ์ระหว่างจำนวนรอบเฉลี่ยการฝึกสอนและโครงข่าย ปัญหา Exclusive – Or



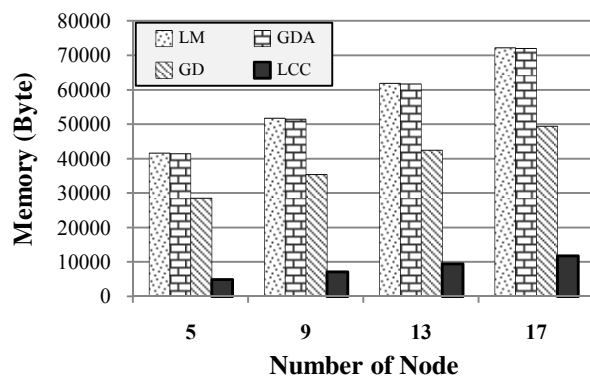
รูปที่ 5 ความสัมพันธ์ระหว่างจำนวนรอบเฉลี่ยการฝึกสอนและโครงข่าย ปัญหา 3-bit parity



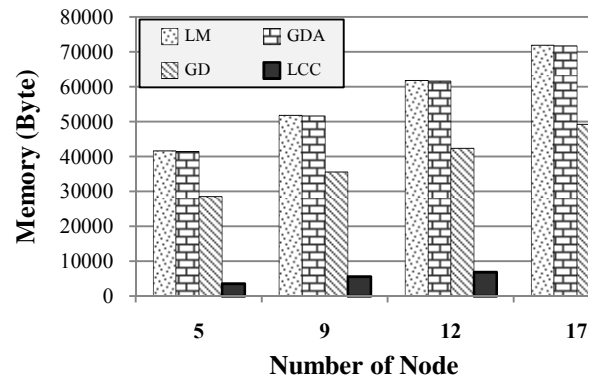
รูปที่ 6 ความสัมพันธ์ระหว่าง ค่าเวลาฝึกสอน ต่อ 1 Epoch และโครงข่าย ปัญหา Exclusive – Or



รูปที่ 7 ความสัมพันธ์ระหว่าง ค่าเวลาฝึกสอน ต่อ 1 Epoch และโครงข่าย ปัญหา 3-bit parity



รูปที่ 8 ความสัมพันธ์ระหว่าง หน่วยความจำที่ใช้ในการประมวลผลของแต่ละอัลกอริทึม และจำนวนโนดในปัญหา Exclusive-Or



รูปที่ 9 ความสัมพันธ์ระหว่าง หน่วยความจำที่ใช้ในการประมวลผลของแต่ละอัลกอริทึม และจำนวนโนดในปัญหา 3 bit parity

จากกราฟรูปที่ 2 และ 3 จะเห็นว่า ค่าความสำเร็จการฝึกสอนโครงข่ายด้วยอัลกอริทึม LCC มีค่า 100 % ในปัญหา Exclusive-Or และมีค่า 100 % ในปัญหาของ 3 Bit Parity เฉพาะโครงข่ายมีชั้นในโครงข่ายไม่เกิน 4 ชั้น แต่เมื่อโครงข่ายมีชั้นในโครงข่ายมากกว่า 4 ชั้น พบว่า ค่าความสำเร็จการฝึกสอนโครงข่ายด้วยอัลกอริทึม LCC มีค่าลดลงเป็น 60% ส่วนการฝึกสอนโครงข่ายด้วยอัลกอริทึม LM นั้น ค่าความสำเร็จการฝึกสอนโครงข่ายมีค่าเป็น 100 % ทั้งในปัญหาของ Exclusive-Or และ 3 Bit Parity และสังเกตได้ชัดเจน กราฟรูปที่ 2 และ 3 ค่าการฝึกสอนโครงข่ายด้วยอัลกอริทึม GDA และ GD นั้น จะมีค่าความสำเร็จการฝึกโครงข่ายเพิ่มขึ้นจนถึง 100 % เมื่อจำนวนชั้นในโครงข่ายเพิ่มขึ้น ในทำนองเดียวกัน กราฟรูปที่ 4 แสดงจำนวนรอบเฉลี่ยการฝึกสอนโครงข่าย ของปัญหา Exclusive-Or เห็นได้ว่า การฝึกสอนโครงข่ายด้วยอัลกอริทึม LCC และ GD นั้น มีจำนวนรอบเฉลี่ยการฝึกโครงข่ายเพิ่มขึ้นเมื่อจำนวนชั้นในโครงข่ายเพิ่มขึ้น แต่พบว่า การฝึกโครงข่ายด้วยอัลกอริทึม LM และ GDA นั้น ยังคงมีจำนวนรอบเฉลี่ยการฝึกโครงข่ายที่น้อย และมีค่าลดลงเมื่อจำนวนชั้นในโครงข่ายเพิ่มขึ้น

จากกราฟรูปที่ 5 แสดงจำนวนรอบเฉลี่ยการฝึกสอนโครงข่ายด้วยปัญหา 3 Bit Parity จะเห็นว่า การฝึกสอนโครงข่ายด้วยอัลกอริทึม LCC นั้น ยังคงมีจำนวนรอบเฉลี่ยการฝึกสอนโครงข่ายเพิ่มขึ้นเมื่อจำนวนชั้นในโครงข่ายเพิ่มขึ้น แต่พบว่า การฝึกสอนโครงข่ายด้วยอัลกอริทึม GD นั้น ให้ผลลัพธ์ที่แตกต่างจากผลลัพธ์ที่ได้จากกราฟรูปที่ 4 คือ จำนวนรอบเฉลี่ยการฝึกสอนโครงข่ายลดลงเมื่อจำนวนชั้นในโครงข่ายเพิ่มขึ้น ซึ่งตรงกันข้ามกับการฝึกสอนโครงข่ายด้วยปัญหา Exclusive-Or ส่วนการฝึกสอนโครงข่ายด้วยอัลกอริทึม LM และ GDA กับปัญหา 3 Bit Parity พบว่า จำนวนของรอบเฉลี่ยการฝึกสอนยังคงสอดคล้องกับผลการฝึกสอนโครงข่ายที่ได้จากกราฟรูปที่ 4 ก่อนหน้านั้น

จากกราฟรูปที่ 6 แสดงค่าเวลาฝึกสอนโครงข่ายในแต่ละรอบ เมื่อฝึกสอนโครงข่ายด้วยปัญหา Exclusive-Or พบว่า การฝึกสอนโครงข่ายในแต่ละรอบนั้น อัลกอริทึม LM ใช้เวลาฝึกสอนมากที่สุด ถัดมา คือ GDA, GD และ LCC ตามลำดับ โดยค่าเวลาการฝึกสอนโครงข่ายของทั้ง 4 อัลกอริทึมนี้ มีค่าเพิ่มขึ้นเมื่อจำนวนชั้นในโครงข่ายเพิ่มขึ้น

จากกราฟรูปที่ 7 จะเห็นได้อย่างชัดเจน ว่า ทั้งการฝึกสอน

โครงข่ายด้วยอัลกอริทึม LM, GDA และ LCC นั้น ใช้เวลาการฝึกสอนโครงข่ายในแต่ละรอบเพิ่มขึ้น เมื่อจำนวนชั้นในโครงข่ายเพิ่มขึ้น แต่ตรงข้ามกับอัลกอริทึม GD ที่มีค่าลดลงเมื่อจำนวนชั้นในโครงข่ายเพิ่มขึ้น

จากกราฟรูปที่ 8 ถึง 9 แสดงหน่วยความจำที่ใช้ในการฝึกสอนโครงข่ายของแต่ละอัลกอริทึม พบว่า การฝึกสอนโครงข่ายด้วยอัลกอริทึม LCC นั้น ใช้หน่วยความจำน้อยที่สุด และการฝึกสอนโครงข่ายด้วย LM ใช้หน่วยความจำมากที่สุด รองลงมา คือ GDA และ GD ตามลำดับ ทั้งการฝึกสอนโครงข่ายด้วยปัญหา Exclusive-Or และ 3 Bit Parity และยังพบว่า หน่วยความจำที่ใช้ในการฝึกสอนโครงข่ายจะมีค่าแปรผันตามจำนวนชั้นและจำนวนโนดในโครงข่าย อังมีจำนวนชั้นและจำนวนโนดในโครงข่ายเพิ่มขึ้น ย่อมทำให้การฝึกสอนโครงข่ายใช้หน่วยความจำในการฝึกสอนเพิ่มขึ้น

จากผลการทดลองกราฟรูปที่ 2 ถึง 7 นั้น สามารถสรุปได้ว่าการฝึกสอนโครงข่ายด้วยอัลกอริทึม LCC นั้นใช้หน่วยความจำน้อยมาก และการฝึกสอนโครงข่ายจะมีค่าความสำเร็จในการฝึกโครงข่ายที่สูง เมื่อจำนวนชั้นในโครงข่ายมีจำนวนชั้นในโครงข่ายไม่เกิน 4 ชั้น ส่วนการฝึกสอนโครงข่ายด้วยอัลกอริทึม LM จะใช้หน่วยความจำมาก แต่มีความรวดเร็วในการฝึกสอนโครงข่ายสูง ส่วนการฝึกสอนโครงข่ายด้วยอัลกอริทึม GDA และ GD นั้น เหมาะสมกับการฝึกสอนโครงข่ายที่มีจำนวนชั้นในโครงข่ายมาก เมื่อเปรียบเทียบกับ การฝึกสอนโครงข่ายที่มีชั้นในโครงข่ายเพียง 1-2 ชั้น และการฝึกสอนโครงข่ายด้วยอัลกอริทึม GDA นั้นจะใช้หน่วยความจำน้อยกว่าการฝึกสอนโครงข่ายด้วยอัลกอริทึม LM และการฝึกสอนโครงข่ายด้วยอัลกอริทึม GD นั้นจะใช้หน่วยความจำในการฝึกสอนโครงข่ายน้อยกว่า GDA แต่ยังคงมีค่าสูงกว่าอัลกอริทึม LCC

จากการวิเคราะห์ผลการทดสอบการฝึกสอนโครงข่ายของทั้ง 4 อัลกอริทึมที่ผ่านมา โดยทั่วไปแล้ว พบว่า อัลกอริทึม LCC นั้น มีค่าความสำเร็จการฝึกสอนโครงข่ายต่ำกว่าอัลกอริทึม LM และค่าความสำเร็จการฝึกสอนโครงข่ายมีค่าลดลง เมื่อจำนวนชั้นและโนดในโครงข่ายเพิ่มขึ้น เพื่อพิจารณาปัญหาดังกล่าว ได้ทดลองพิจารณาเอาท์พุทที่ได้จากการฝึกสอนโครงข่ายด้วยอัลกอริทึม LCC และ LM ในการแก้ปัญหา Exclusive-Or ที่มีค่าอินพุทฝึกสอนและเอาท์พุทฝึกสอน ดังตารางที่ 3

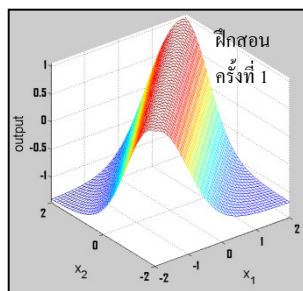
ตารางที่ 3 แสดงค่าอินพุตและเอาต์พุตของปัญหา Exclusive-Or

Input X_1	Input X_2	Output y
-1	-1	1
-1	1	-1
1	-1	-1
1	1	1

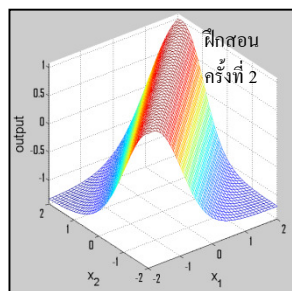
ตารางที่ 3 แสดงความสัมพันธ์ระหว่างกันของค่าเอาต์พุตที่ต้องการ และ อินพุตฝึกสอนโครงข่าย ซึ่งมีค่าเท่ากับ -1 และ 1 เท่านั้น ตามที่แสดงในตารางที่ 3 เพื่อทดสอบความถูกต้อง และ พิจารณาพฤติกรรม หลังจากโครงข่ายถูกฝึกสอนจาก อัลกอริทึม LM และ LCC แล้ว จะกำหนดให้อินพุตทดสอบ คือ x_1 และ x_2 มีค่าเท่ากับ $[-2,2]$ เข้าสู่โครงข่ายที่ถูกฝึกสอนด้วย อัลกอริทึม LCC และ LM มาก่อนหน้านั้นแล้ว โดยใช้ แก่ปัญหา Exclusive-Or ผลการทดสอบได้แสดงในรูปที่ 10 มี

ลักษณะเป็น 3 มิติ โดยแกนแนวระนาบ คือค่าอินพุตทดสอบ x_1 และ x_2 แกนแนวตั้งเป็นค่าเอาต์พุต y ซึ่งได้จากค่าอินพุตทดสอบ ดังแสดงรูปที่ 10 (a), (b), (c), (d), (e) และ (f) ตามลำดับ ในแต่ละรูป จะแสดงผลของการฝึกสอน จำนวน 2 ครั้ง โดยที่ โครงข่ายนั้นมีค่าน้ำหนักเริ่มต้นในโครงข่ายต่างกัน

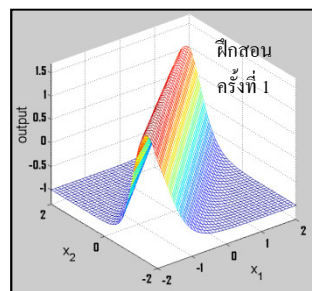
จากกราฟรูปที่ 10 (a) และ (b) เป็นค่าเอาต์พุตที่ได้จากการฝึกสอนโครงข่ายด้วยอัลกอริทึม LM และ LCC ของโครงข่ายแบบสองชั้น $([2,1])$ ที่มีโนดในชั้นซ่อนเท่ากับ 2 โนด และ โนดในชั้นเอาต์พุตเท่ากับ 1 โนด พบว่า โครงข่ายที่ผ่านการฝึกสอนจากทั้งสองอัลกอริทึมให้ค่าเอาต์พุตสอดคล้องกับค่าเอาต์พุตฝึกสอน ดังแสดงในตารางที่ 3 และพบว่าการฝึกสอนโครงข่ายด้วย อัลกอริทึม LM และ LCC ที่มีโครงข่ายแบบสองชั้นนั้น มีเสถียรภาพของความสม่ำเสมอสูง เห็นได้จาก ผลการทดลองทั้งสองครั้งให้ค่าเอาต์พุตแน่นอนและสอดคล้องกันทั้งสองครั้งการของการทดสอบ



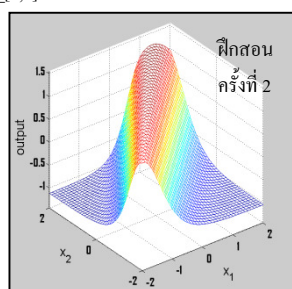
(a) LCC_[2,1]



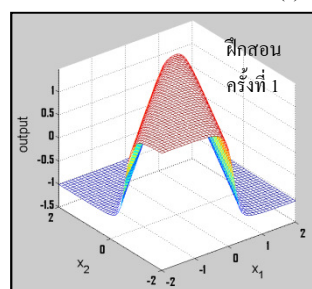
(b) LM_[2,1]



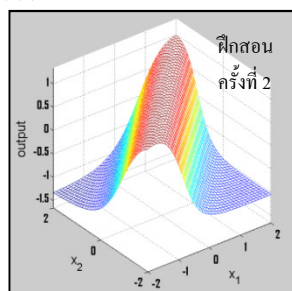
(c) LCC_[2,2,1]



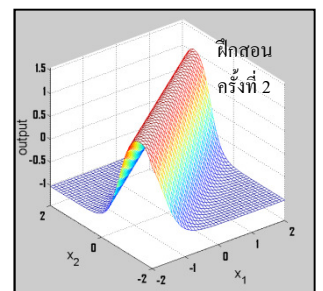
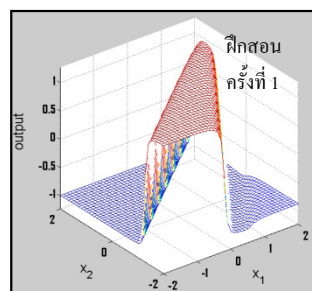
(d) LM_[2,2,1]



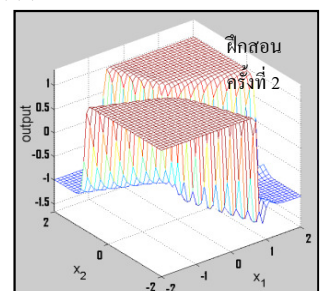
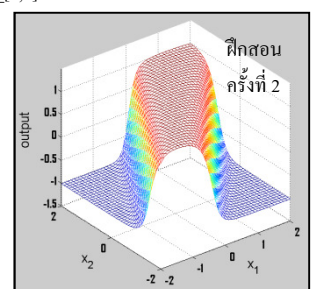
(e) LCC_[2,4,4,1]



(f) LM_[2,4,4,1]



(h) LM_[2,2,2,1]



กราฟรูปที่ 10 (c) และ (d) เป็นค่าเอาต์พุตของโครงข่ายที่ถูกฝึกสอนโดยอัลกอริทึม LCC และ LM ที่มีโครงข่ายแบบสี่ชั้น (2,2,2,1) มีโนดในแต่ละชั้นซ่อนเท่ากับ 2 โนด และ โนดชั้นเอาต์พุตเท่ากับ 1 โนด จากกราฟรูปที่ 10 (c) พบว่า เอาต์พุตที่ได้จากโครงข่ายที่ถูกฝึกสอนโดยอัลกอริทึม LCC สามารถสร้างค่าเอาต์พุตได้สอดคล้องกับเอาต์พุตฝึกสอนในตารางที่ 3 เมื่ออินพุตทดสอบมีค่าเท่ากับอินพุตฝึกสอน และเมื่ออินพุตทดสอบมีค่าเปลี่ยนแปลงไปจากอินพุตฝึกสอนเล็กน้อย พบว่าโครงข่ายที่ถูกฝึกสอนด้วยอัลกอริทึม LCC ยังคงสามารถสร้างเอาต์พุตที่มีแนวโน้มของสอดคล้องกับเอาต์พุตฝึกสอนตามตารางที่ 3 ได้ และจากกราฟรูปที่ 10 (d) แสดงเอาต์พุตที่ได้จากโครงข่ายที่ถูกฝึกสอนโดยอัลกอริทึม LM พบว่า สร้างค่าเอาต์พุตได้ตรงตามเอาต์พุตฝึกสอนในตารางที่ 3 เมื่ออินพุตทดสอบมีค่าตรงกับอินพุตฝึกสอนเท่านั้น แต่เมื่อ อินพุตทดสอบมีค่าเปลี่ยนแปลงไปจากอินพุตฝึกสอน พบว่าโครงข่ายที่ถูกฝึกสอนด้วยอัลกอริทึม LM ไม่สามารถสร้างเอาต์พุตให้มีแนวโน้มของสอดคล้องกับเอาต์พุตฝึกสอนตามตารางที่ 3 ได้ ค่าเอาต์พุตที่ได้จะไม่สม่ำเสมอ ในทำนองเดียวกันเมื่อทำการฝึกสอนในแต่ละครั้ง กราฟรูปที่ 10 (e) และ (f) เป็นค่าเอาต์พุตของโครงข่ายที่ถูกฝึกโดยอัลกอริทึม LCC และ LM ที่มีโครงข่ายแบบห้าชั้น (2,4,4,4,1) มีโนดในชั้นซ่อนแรกเท่ากับ 2 โนด ในแต่ละชั้นซ่อนถัดมา จำนวนโนดในชั้นซ่อนเท่ากับ 4 โนด และ จำนวนโนดในชั้นเอาต์พุตเท่ากับ 1 โนด จากกราฟรูปที่ 10 (e) พบว่า เอาต์พุตที่ได้จากโครงข่ายที่ถูกฝึกสอนโดยอัลกอริทึม LCC สามารถ สร้างค่าเอาต์พุตได้ตรงตามเอาต์พุตฝึกสอนดังแสดงในตารางที่ 3 เมื่ออินพุตทดสอบมีค่าตรงกับอินพุตฝึกสอน อีกทั้งรูปทรงพื้นผิวของเอาต์พุตตอบสนองต่ออินพุตนั้นคล้ายกับ รูป 10(a) และ (c) ซึ่งเกิดจาก อัลกอริทึม LCC

เมื่อพิจารณากราฟรูปที่ 10 (a) ถึง(f) พบว่า เสถียรภาพของความสม่ำเสมอ (Consistency) ของการให้คำตอบของโครงข่ายที่ถูกฝึกสอนด้วยอัลกอริทึม LCC นั้น มีแนวโน้มที่ดีกว่าโครงข่ายที่ถูกฝึกสอนด้วยอัลกอริทึม LM สังเกตได้จาก ค่าเอาต์พุตที่ได้ในแต่ละครั้งของการฝึกสอนโครงข่ายด้วยอัลกอริทึม LCC มีแนวโน้มที่สอดคล้องกัน แสดงรูปแบบพื้นผิวเอาต์พุตตอบสนองในลักษณะเดียวกันแม้ว่าโครงข่ายนั้นมีโครงสร้างแตกต่างกันทำการฝึกสอนด้วยค่าเริ่มต้นที่ต่างกัน ซึ่งผลลัพธ์นี้แตกต่างจากค่าเอาต์พุตที่ได้ในแต่ละครั้งของการฝึกสอนโครงข่ายด้วยอัลกอริทึม LM ซึ่งมีความ

แตกต่างกันอย่างมากและไม่สอดคล้องกันในการทดสอบแต่ละครั้ง แสดงถึงความไม่มีเสถียรภาพของความสม่ำเสมอ ในการทำนาย คำตอบที่อยู่ ณ จุดที่ไม่มีข้อมูลฝึกสอน

ผลต่อเนื่องจากความสม่ำเสมอของแนวโน้มของเอาต์พุตที่ใกล้เคียงกับข้อมูลที่ใช้ฝึกสอนเมื่ออินพุตเบี่ยงเบนไปจากค่าฝึกสอน เราอาจสามารถกล่าวได้ว่า โครงข่ายที่ถูกฝึกสอนด้วยอัลกอริทึม LCC นั้น มีความเป็นนัยทั่วไป (Generalization) สูงกว่า โครงข่ายที่ถูกฝึกสอนด้วยอัลกอริทึม LM ซึ่งสามารถเรียนรู้จากข้อมูลฝึกสอนโครงข่ายได้ดี และเมื่อนำมาโครงข่ายที่ถูกฝึกสอนด้วยอัลกอริทึม LCC ไปใช้กับข้อมูลทดสอบพบว่า ผลลัพธ์ที่ได้จากข้อมูลทดสอบมีความสอดคล้องตามข้อมูลฝึกสอน ทำให้สามารถทำนายความน่าจะเป็นของผลลัพธ์จากโครงข่ายได้ การได้คำตอบไปในทิศทางเดียวกันนั้น เป็นคุณสมบัติที่ดี ที่ต้องการ เมื่อต้องการนำโครงข่ายประสาทเทียมมาประยุกต์ใช้กับระบบที่ต้องการการทำนาย คำตอบที่อยู่ ณ จุดที่ไม่มีข้อมูลฝึกสอน แตกต่างจากโครงข่ายที่ถูกฝึกสอนด้วยอัลกอริทึม LM ซึ่งจะประมวลผลได้อย่างแม่นยำ เฉพาะข้อมูลทดสอบที่ตรงกับข้อมูลฝึกสอนเท่านั้น ส่วนข้อมูลทดสอบที่มีค่าต่างไปจากข้อมูลฝึกสอนแล้ว อัลกอริทึม LM จะไม่สามารถสร้างผลลัพธ์ ให้ค่าที่มีแนวโน้มสอดคล้องกับข้อมูลฝึกสอนโครงข่ายได้ ซึ่งยากต่อการทำนายผลลัพธ์ที่จะได้จากโครงข่าย

4. สรุปโดยรวม

อัลกอริทึมฝึกโครงข่าย LCC ที่นำเสนอนี้ ให้ผลลัพธ์ของคำตอบที่มีความเป็นนัยทั่วไปที่ดี และ มีเสถียรภาพของความสม่ำเสมอในการให้คำตอบที่ดี สามารถเรียนรู้ข้อมูลฝึกสอนได้ดี เมื่อนำโครงข่ายที่ถูกฝึกสอนด้วยอัลกอริทึม LCC ไปใช้กับข้อมูลทดสอบ ที่มีค่าข้อมูลทดสอบที่แตกต่างไปจากข้อมูลฝึกสอน พบว่า โครงข่ายสามารถสร้างเอาต์พุตที่มีแนวโน้มสอดคล้องกับข้อมูลฝึกสอน และมีความสม่ำเสมอในการให้คำตอบ หน่วยความจำที่ใช้ในการประมวลผลนั้นมีค่าน้อยมาก เมื่อเปรียบเทียบกับหน่วยความจำที่ใช้ในการประมวลผลของอัลกอริทึมของ LM, GDA และ GD ผลการทดลอง แสดงให้เห็นว่า อัลกอริทึม LCC นี้เหมาะสมกับการฝึกสอน โครงข่ายที่มีจำนวนชั้นในโครงข่ายที่ไม่ควรเกิน 4 ชั้น (ในขอบเขตการทดลองที่แสดงไว้ในบทความนี้) ซึ่งอัลกอริทึม LCC จะประมวลผลได้รวดเร็วและมีความแม่นยำมากขึ้น

5. การนำไปประยุกต์ใช้งาน

นำไปประยุกต์ใช้ในการประมวลผล เพื่อหาค่าที่เหมาะสมที่สุดของระบบการเรียนรู้แบบมีผู้ฝึกสอน (Supervisor Training) ซึ่งประมวลผลด้วยหน่วยประมวลผลที่มีหน่วยความจำสำรองอย่างจำกัด อย่างเช่น ไมโครคอนโทรลเลอร์ ที่มีความง่ายกว่าในการออกแบบเพื่อประยุกต์ใช้งานในสภาพแวดล้อมไม่คงที่ เช่น มีอุณหภูมิสูง-ต่ำหรือมีสัญญาณรบกวนจากอุปกรณ์ชนิดอื่นๆ ตลอดเวลา แทนที่จะใช้หน่วยประมวลผลที่เป็นดิจิทัลที่ซับซ้อนมาก เช่น คอมพิวเตอร์ตั้งโต๊ะ ที่มีหน่วยความจำสำรองมากและความเร็วสูง แต่ไม่เหมาะสมต่อการนำไปใช้ในสภาพแวดล้อม เช่น ในโรงงานอุตสาหกรรม เนื่องจากไม่สามารถทนต่อสัญญาณรบกวน และสภาพแวดล้อม เมื่อเทียบกับการใช้วงจรไมโครคอนโทรลเลอร์ที่ออกแบบเป็นอย่างดี

6. เอกสารอ้างอิง

- [1] Amir Abolfazl Suratgar, Mohammad Bagher Tavakoli & Abbas Hoseinabad, "Modified Levenberg-Marquardt Method for Neural Networks Training", World Academy of Science, Engineering and Technology, yr. 2005.
- [2] A. Vongkungha and A. Chumthong " The Performance comparisons of Backpropagation Algorithm's Family on a Set of Logical Functions", Transactions on Electrical Eng., Electronics, and Communications (ECTI-EEC) Volume 5, No. 2, August 2007
- [3] Bernard Widrow and S. D. Stearns. "Adaptive Signal Processing". Englewood Cliffs, NJ, Printice-Hall, 1985
- [4] Bernard Widrow and Michael A. Lehr, "30 Years of Adaptive Neural Networks : Perceptron, Madaline and Backpropagation", Proceedings of the IEEE, Volume 78, Issue 9, Sept 1990 pp. 1415-144.
- [5] Y. Maeda, H. Yamashita, and Y. Kanata, "Learning rules for multilayer neural networks using difference approximation". In Proc. IJCNN, Singapore, vol.1 1990 pp. 628-633
- [6] M.Jabri and B.Flower, "Weight perturbation : an optimal architecture and learning technique for analog VLSI feedforward and recurrent multilayer networks". Neural Network, IEEE Transaction on, vol.3 1992 pp.154-157
- [7] K. Hirotsu and M. Brooke, "An analog neural network chip with random weight change learning algorithm". in Proc. Int. Joint Conf. Neural Networks, Oct 1993 pp. 3031-3034
- [8] G. Cauwenberghs, "Fast Stochastic Error-descent algorithm for supervised learning and optimization". in Advances in Neural Information Processing Systems 5, S. J. Hanson, J. D. Cowan, and C. L. Giles, Eds. San Mateo, CA: Morgan Kaufmann, 1993 pp. 244-25
- [9] P.W. Hollis and J. J. Paulos, "A neural network learning algorithm tailored for VLSI implementation". IEEE Trans. Neural Networks, vol. 5 Sept 1994 pp. 784-791
- [10] G. Cauwenberghs, "A learning analog neural network chip with continuous-time recurrent dynamics", In Advances in Neural Information Processing Systems. New York: Morgan Kaufmann, vol. 6 Sept 1994 pp. 77-86
- [11] G.Cauwenberghs, "Analog VLSI stochastic perturbative learning architectures". Analog Intregret. Circuits and Signal Process, vol. 13 Sept 1994 pp. 195-209
- [12] A. Vongkunghae, " A Global Optimization Method Using Gaussian Distribution Perturbation", Naresuan University Engineering Journal, Naresuan University , Phitsanulok, Thailand, Jan – April 2007 Vol. 2 No.1 pp. 39 – 43
- [13] MATLAB. Version 2010a, Service Pack 2, March 2010, Mathwork. Inc., US