บทที่ 3
วิธีการปรับอัตราการเรียนรู้แบบพลวัต

ในการลดระดับตามแนวเกรเดียนต์ กรณีที่สัญญาณขาเข้าเป็นกระบวนการสโตคาสติกส์และไม่ทราบค่าทางสถิติ เราจำเป็นต้องใช้วิธีเกรเดียนต์แบบสโตคาสติกส์ (Stochastic Gradient) เพื่อหาค่าเหมาสมที่สุด ดังแสดงไว้ในบทที่ 2 ในวิธีนี้ พารามิเตอร์ที่ปรับมีลักษณะเป็นกระบวนการสโตคาสติกส์ และไม่สามารถเข้าสู่ค่าเหมาะสมที่สุดได้อย่างแท้จริง ค่าที่ได้จะเป็นค่าประมาณที่ใกล้เคียงเท่านั้น และอัตราการเรียนรู้จะมีผลต่อการประมาณค่านี้อย่างมาก ถ้าอัตราการเรียนรู้ยิ่งมากเท่าใด ค่าประมาณจะยิ่งห่างจากค่าเหมาะสมที่สุดมากขึ้น แต่ถ้ากำหนดอัตราการเรียนรู้น้อยจะประมาณได้ใกล้เคียงมากขึ้น แต่ก็ทำให้ลู่เข้าช้ากว่า ปัญหาที่กล่าวนี้เกิดจากการใช้อัตราการเรียนรู้เป็นค่าคงตัว ดังนั้น การหลีกเลี่ยงปัญหานี้ จึงใช้อัตราการเรียนรู้ที่มีค่าเปลี่ยนค่าตามเวลาในการปรับพารามิเตอร์ เพื่อให้สามารถลู่เข้าได้เร็วและประมาณค่าเหมาะสมที่สุดได้ใกล้เคียง

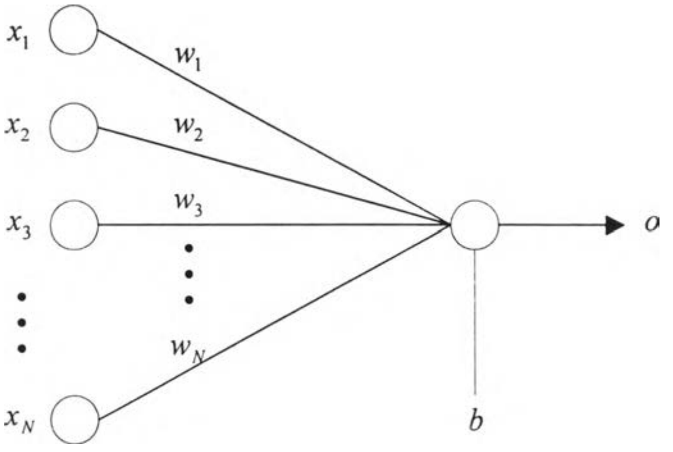
การใช้ข่ายงานระบบประสาทสามารถใช้งานได้ 2 แบบ คือ แบบกลุ่ม (Batch) และแบบเชื่อมตรง (on-line) การใช้งานแบบกลุ่มจะปรับพารามิเตอร์หลังจากได้รับข้อมูลทั้งหมด ส่วนแบบเชื่อมตรงจะปรับพารามิเตอร์ทุกครั้งหลังจากที่ข้อมูลถูกป้อนเข้าระบบโดยไม่จำเป็นต้องรอข้อมูลทั้งหมดก่อน ซึ่งมีข้อได้เปรียบหลายประการ เช่น เราไม่จำเป็นต้องเก็บข้อมูลเข้าทั้งหมดไว้, ระบบสามารถปรับพารามิเตอร์ไปพร้อมกับการใช้งานได้ และเนื่องจากระบบได้ถูกปรับพารามิเตอร์อยู่ตลอดเวลา ทำให้สามารถติดตามการเปลี่ยนแปลงของข้อมูลเข้าได้เร็วกว่า นอกจากนี้ การใช้งานแบบนี้ยังมีแนวโน้มที่จะเรียนรู้ได้เร็วกว่า โดยเฉพาะในกรณีที่มีข้อมูลเข้าจำนวนมาก ข้อได้เปรียบเหล่านี้เป็นจุดน่าสนใจในการนำไปประยุกต์ใช้งาน ซึ่งทำให้ใช้งานได้กว้างขวางกว่า

เนื่องจาก การปรับอัตราการเรียนรู้ให้เปลี่ยนค่าตามเวลาสามารถให้ความเร็วในการลู่เข้าและการประมาณค่าที่ใกล้เคียงได้ ดังนั้น เมื่อนำมาใช้ในระบบที่ทำงานแบบเชื่อมตรง จะทำให้สามารถนำไปประยุกต์ใช้งานได้กว้างขวางมากขึ้น จึงมีผู้เสนอวิธีการปรับอัตราการเรียนรู้ไว้หลายวิธีด้วยกันดังแสดงไว้ในบทที่ 1 ซึ่งได้ทดสอบแล้วว่าสามารถให้ประสิทธิภาพได้ดีกว่าการใช้อัตราการเรียนรู้เป็นค่าคงตัว แต่วิธีที่เหมาะสมกับการใช้งานแบบเชื่อมตรงควรจะมีการคำนวณน้อยเพื่อให้ใช้เวลาน้อยเมื่อระบบเรียนรู้ไปพร้อมกับการใช้งาน ทำให้สามารถปรับตัวได้เร็ว ในวิทยานิพนธ์นี้ จึงได้เลือกวิธีที่มีคุณสมบัติข้างต้นมาทดสอบ ซึ่งได้แก่ วิธี Dbd, Idbd, Entropy และ cosine direction ซึ่งได้ปรับปรุงให้อัตราการเรียนรู้มีค่าเป็นบวกเสมอ

ในบทนี้ กล่าวถึงวิธีการปรับอัตราการเรียนรู้แบบพลวัต (การปรับอัตราการเรียนรู้ให้เปลี่ยนค่าตามเวลา) ที่ได้เลือกมาทดสอบในวิทยานิพนธ์ รวมไปถึงแนวทางและเหตุผลที่ใช้เพื่อสร้างขั้นตอนของแต่ละวิธี ในตอนท้ายของบทได้กล่าวถึงความซับซ้อน (complexity) และการคำนวณ (computational) ของแต่ละวิธี ความซับซ้อนเป็นหน่วยความจำที่จำเป็นต้องใช้เพื่อทำตามขั้นตอนวิธี ส่วนการคำนวณที่แสดงเป็นการคำนวณที่ต้องเกิดขึ้นในแต่ละรอบการเรียนรู้ ทั้ง 2 อย่างซึ่งแสดงในรูปของจำนวนพารามิเตอร์ของระบบ (n)

3.1 โครงสร้างของข่ายงานระบบประสาท

โครงสร้างของข่ายงานระบบประสาท ปกติจะมีหลายชั้นและแต่ละชั้นจะประกอบด้วยจำนวนปม (node) หลายปม โดยแต่ละปมมีลักษณะเป็นแบบเดียวกัน ดังนั้น เพื่อความเข้าใจและสามารถพิจารณาได้ง่าย เราจะพิจารณาข่ายงานระบบประสาทที่มีโครงสร้างเป็นแบบชั้นเดียวและประกอบด้วยปมเพียง 1 ปม ซึ่งจะมีลักษณะโครงสร้างและความสัมพันธ์ ตลอดจนการปรับพารามิเตอร์ของระบบ เช่นเดียวกัน โดยสามารถขยายการใช้งานออกไปใช้กับโครงสร้างที่มีหลายปม หลายชั้น ได้โดยพิจารณาให้แต่ละปมให้มีการทำงานที่เป็นอิสระต่อกัน



รูปที่ 3.1 โครงสร้างข่ายงานระบบประสาทแบบชั้นเดียว และมีปม 1 ปม

ในรูปที่ 3.1 แสดงโครงสร้างที่ประกอบด้วยชั้นสัญญาณขาออก และมีปม 1 ปม สัญญาณขาเข้า N สัญญาณ ซึ่งแสดงด้วยสัญญลักษณ์ $x_1, x_2, x_3, \ldots, x_N$ จะถูกส่งผ่านมายังปมที่ใช้แสดงสัญญาณขาออก o นี้ โดยมีค่าน้ำหนัก $w_1, w_2, w_3, \ldots, w_N$ คูณอยู่กับค่าสัญญาณแต่ละสัญญาณ ทำให้ได้ผลรวมถ่วงน้ำหนักของสัญญาณขาเข้าเป็นข้อมูลเข้าของปมแสดงสัญญาณขาออก โดยมีค่าไบแอสประจำปม ซึ่งจะเป็นพารามิเตอร์สำหรับปรับข้อมูลเข้าอีกที่ ค่าไบแอสนี้ไม่จำเป็นต้องกำหนดไว้ในโครงสร้างก็ได้ ซึ่งจะเหมือนกับใช้ค่านี้เป็นศูนย์ เราจะใช้ค่าน้ำหนักและค่าไบแอสเป็น

พารามิเตอร์ของระบบเพื่อทำการปรับให้สัญญาณขาออกของระบบ o เข้าใกล้สัญญาณอ้างอิง d โดยใช้กำหนดสัญญลักษณ์เป็นเวกเตอร์ของสัญญาณขาเข้า $\underline{x}$ และเวกเตอร์พารามิเตอร์ $\underline{w}$ ซึ่งมีขนาดมิติเท่ากับจำนวนที่มีอยู่ในระบบคือ N ดังนี้

$$\underline{x} = \begin{bmatrix} x_1 & x_2 & \dots & x_N \end{bmatrix}^T \qquad (3.1.1)$$
$$\underline{w} = \begin{bmatrix} w_1 & w_2 & \dots & w_N \end{bmatrix}^T$$

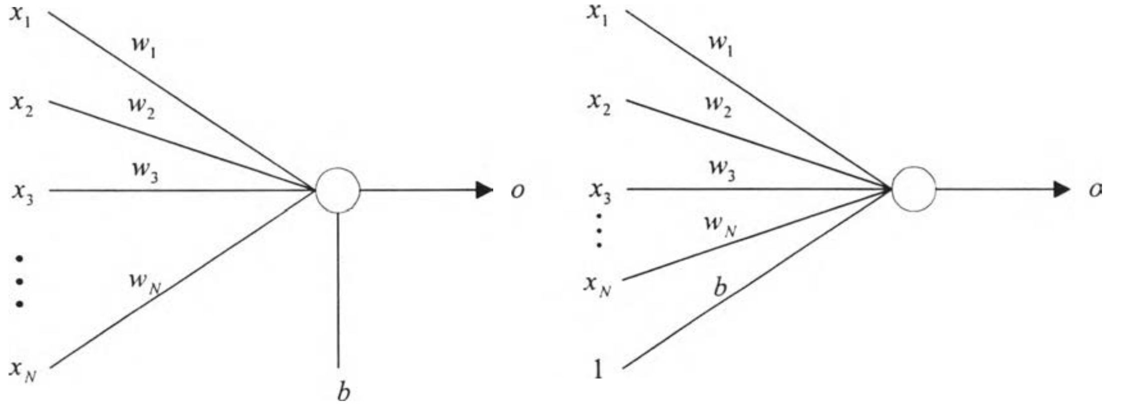
โดยที่สัญญาณขาออก o มีความสัมพันธ์กับสัญญาณขาเข้าและพารามิเตอร์ของระบบตามฟังก์ชัน activation $g(.)$ ของปมที่เลือก ดังนี้

$$o = g\left(\underline{w}^T \underline{x}\right) \qquad (3.1.2)$$

เมื่อเลือกฟังก์ชัน activation นี้เป็นแบบเชิงเส้น สัญญาณขาออกจะเท่ากับผลรวมถ่วงน้ำหนักของสัญญาณขาเข้าและค่าไบแอส ซึ่งแสดงได้เป็น

$$o = \underline{w}^T \underline{x} + b \qquad (3.1.3)$$

เราสามารถพิจารณาให้ค่าไบแอสรวมอยู่ในเวกเตอร์พารามิเตอร์โดยให้เป็นค่าน้ำหนักคูณกับสัญญาณขาเข้าอีกสัญญาณที่มีค่าเป็น 1 ดังแสดงในรูปที่ 3.2



รูปที่ 3.2 การรวมค่าไบแอสไว้กับเวกเตอร์พารามิเตอร์โดยพิจารณาให้เป็นค่าน้ำหนักของสัญญาณขาเข้าที่เป็น 1

เวกเตอร์สัญญาณขาเข้า $\underline{x}$ และเวกเตอร์ค่าน้ำหนัก $\underline{w}$ จะมีขนาดเพิ่มขึ้น ดังนี้

$$\underline{x}_{new} = \begin{bmatrix} \underline{x}_{old} \\ 1 \end{bmatrix} \;;\quad \underline{w}_{new} = \begin{bmatrix} \underline{w}_{old} \\ b \end{bmatrix} \tag{3.1.4}$$

โดยที่ $\underline{x}_{new}$ เป็นเวกเตอร์สัญญาณขาเข้าใหม่ที่ได้จากการรวมสัญญาณขาเข้าเดิม $\underline{x}_{old}$ กับสัญญาณขาเข้าของค่าไบแอสคือ 1 และเวกเตอร์พารามิเตอร์ใหม่ $\underline{w}_{new}$ เป็นเวกเตอร์ที่ได้จากการรวมเวกเตอร์ค่าน้ำหนัก $\underline{w}_{old}$ และค่าไบแอสไว้ด้วยกัน

3.2 การแผ่ย้อนกลับโดยค่าผิดพลาด (Error Backpropagation)

การแผ่ย้อนกลับโดยค่าผิดพลาด [1,11] เป็นการปรับพารามิเตอร์ที่ใช้กับข่ายงานระบบประสาทที่มีโครงสร้างแบบหลายชั้น โดยใช้การลดระดับตามแนวเกรเดียนต์ โดยใช้ฟังก์ชันเป้าหมาย $\varepsilon(t)$ เป็นความผิดพลาดกำลังสองเฉลี่ย ตามสมการ

$$\varepsilon(t) = E\left[\left(d(t) - o(t)\right)^2\right] \tag{3.2.1}$$

ซึ่งเป็นความผิดพลาดกำลังสองเฉลี่ย (mean-square error) สัญญลักษณ์ E ใช้แทนการหาค่าคาดหวังของความผิดพลาดกำลังสองของสัญญาณเป้าหมายที่เวลา t , $d(t)$ และสัญญาณขาออกที่เวลา t , $o(t)$ การใช้สัญญลักษณ์เวลาในสมการ เป็นเพื่อเป็นการแสดงถึงการใช้งานแบบเชื่อมตรงที่มีการปรับพารามิเตอร์ทุกครั้งที่ได้รับข้อมูล และค่าฟังก์ชันเป้าหมายจะถูกคำนวณเพื่อปรับพารามิเตอร์ทุกครั้ง

การแผ่ย้อนกลับโดยค่าผิดพลาด จะปรับพารามิเตอร์ไปในทิศทางตรงข้ามของเกรเดียนต์ตามสมการ

$$\underline{w}(t+1) = \underline{w}(t) - \frac{\beta}{2}\nabla_{\underline{w}}\varepsilon(t) \tag{3.2.2}$$

โดยที่ β เป็นอัตราการเรียนรู้ เราสามารถหาเกรเดียนต์ในสมการ (3.2.2) ได้ดังนี้

$$\nabla_{\underline{w}}\varepsilon(t) = -\left(d(t) - o(t)\right)\frac{\partial g\left(\underline{w}^T(t)\underline{x}(t)\right)}{\partial\left(\underline{w}^T(t)\underline{x}(t)\right)}\underline{x}(t) \tag{3.2.3}$$

นำมาแทนในสมการ (3.2.2) จะได้เป็น

$$\underline{w}(t+1) = \underline{w}(t) + \beta\big(d(t) - o(t)\big)\ \frac{\partial g\big(\underline{w}^T(t)\underline{x}(t)\big)}{\partial\big(\underline{w}^T(t)\underline{x}(t)\big)}\underline{x}(t) \tag{3.2.4}$$

ถ้าเลือกฟังก์ชัน activation $g(.)$ เป็นแบบเชิงเส้น ซึ่งทำให้อนุพันธ์ของฟังก์ชันมีค่าเป็น 1 จะได้สมการสำหรับการปรับพารามิเตอร์เป็น

$$\underline{w}(t+1) = \underline{w}(t) + \beta e(t)\underline{x}(t) \tag{3.2.5}$$

ซึ่งเป็นสมการเดียวกับที่แสดงไว้ในหัวข้อที่ 2.4 โดยที่ $e(t) = d(t) - o(t)$ เป็นความผิดพลาดระหว่างสัญญาณอ้างอิงที่เวลา t , $d(t)$ และสัญญาณขาออกที่เวลา t , $o(t)$

สมการ (3.2.4) เป็นสมการปรับค่าที่ใช้อัตราการเรียนรู้เพียงค่าเดียวกับพารามิเตอร์ทุกตัว ในกรณีที่ใช้อัตราการเรียนรู้ของแต่ละพารามิเตอร์แยกกัน จะสามารถเขียนสมการสำหรับปรับพารามิเตอร์แต่ละตัว โดยเขียนเป็นส่วนประกอบเดียวของเวกเตอร์พารามิเตอร์ได้ดังนี้

$$w_i(t+1) = w_i(t) - \beta_i\big(d(t) - o(t)\big)\ \frac{\partial g\big(\underline{w}^T(t)\underline{x}(t)\big)}{\partial\big(\underline{w}^T(t)\underline{x}(t)\big)}x(t),\ \ 1 \le i \le N \tag{3.1.9}$$

โดย β_i เป็นอัตราการเรียนรู้สำหรับการปรับพารามิเตอร์ $w_i(t)$ ซึ่งเป็นส่วนประกอบที่ i ของเวกเตอร์พารามิเตอร์ $\underline{w}(t)$ หรือในกรณีที่ใช้ฟังก์ชัน activation เป็นแบบเชิงเส้น สมการ (3.1.9) สามารถเขียนใหม่ ได้เป็น

$$w_i(t+1) = w_i(t) - \beta_i e(t)x_i(t),\ \ 1 \le i \le N \tag{3.1.9}$$

โดยที่ $e(t)$ เป็นความผิดพลาดระหว่างสัญญาณอ้างอิงและสัญญาณขาออกของระบบ

การแผ่ย้อนกลับโดยค่าผิดพลาดสำหรับข่ายงานระบบประสาทแบบชั้นเดียว เป็นการลดระดับตามแนวเกรเดียนต์ เหมือนที่แสดงไว้ในบทที่ 2 ในกรณีที่ข่ายงานมีโครงสร้างหลายชั้น การปรับพารามิเตอร์ในแต่ละปมของข่ายงานระบบประสาทจะเป็นอิสระต่อกัน ดังนั้น การปรับพารามิเตอร์ในแบบชั้นเดียวก็สามารถขยายไปใช้ได้กับแบบหลายชั้นที่มีหลายปม โดยให้แต่ละปมมีการปรับพารามิเตอร์แยกจากกัน

3.3 จำนวนอัตราการเรียนรู้ที่ใช้ในขั้นตอนวิธี

วิธีการปรับอัตราการเรียนรู้ที่จะกล่าวถึงในบทนี้ มีลักษณะการใช้จำนวนอัตราการเรียนรู้อยู่ 2 แบบคือ ใช้เพียงค่าเดียวกับทุกพารามิเตอร์ และใช้แยกกันสำหรับแต่ละพารามิเตอร์ ซึ่งจะสรุปไว้ในตารางที่ 3.1

ตารางที่ 3.1 จำนวนอัตราการเรียนรู้ที่ใช้ในแต่ละขั้นตอนวิธี

ขั้นตอนวิธี	จำนวนอัตราการเรียนรู้ (ตัว)
ปกติ	1
Dbd	N
Idbd	N
Entropy	1
direction cosine	1

โดยที่ N เป็นจำนวนพารามิเตอร์ในระบบ และขั้นตอนวิธีปกติคือ การใช้อัตราการเรียนรู้เป็นค่าคงตัว

3.4 การปรับอัตราการเรียนรู้แบบ Delta-bar-delta (Dbd)

วิธีการปรับอัตราการเรียนรู้ Delta-bar-delta (Dbd) นี้ [2,13] เป็นวิธีแบบ heuristic ซึ่งกำหนดให้มีอัตราการเรียนรู้ของแต่ละพารามิเตอร์แยกกัน เพื่อให้สามารถปรับค่าได้อย่างอิสระโดยไม่มีผลต่อกัน อัตราการเรียนรู้จะถูกปรับให้เพิ่มขึ้นหรือลดลงบนการเปลี่ยนแปลงของพารามิเตอร์ที่ใช้อัตราการเรียนรู้นั้น

การปรับค่าอัตราการเรียนรู้ จะใช้เครื่องหมายของการเปลี่ยนแปลงพารามิเตอร์ในรอบการเรียนรู้ในอดีต โดยกำหนดเป็นการเปลี่ยนแปลงเฉลี่ย (delta-bar) เทียบกับเครื่องหมายการเปลี่ยนแปลงในรอบปัจจุบัน (delta) ถ้าการเปลี่ยนแปลงทั้งสองมีเครื่องหมายเดียวกันจะเพิ่มอัตราการเรียนรู้ แต่ถ้าต่างกันก็จะลดอัตราการเรียนรู้ลง การเพิ่มจะใช้การบวกเพื่อให้เพิ่มในลักษณะเชิงเส้น แต่การลดจะใช้การคูณเพื่อให้ลดแบบเอ็กโปเนนเชียล (exponential) การตั้งกฎเช่นนี้เพื่อให้การปรับค่ามีเสถียรภาพและลู่เข้าใกล้ค่าเหมาะสมที่สุด

การปรับอัตราการเรียนรู้แบบ Dbd สามารถแสดงเป็นขั้นตอน ได้ดังนี้

ขั้นที่ 1 กำหนดค่าเดลต้า (delta) ดังนี้

$$\delta_i(t) = \frac{\partial \varepsilon(t)}{\partial w_i} \; ; \; 1 \le i \le N \tag{3.4.1}$$

โดยที่ $\varepsilon(t)$ เป็นฟังก์ชันเป้าหมายซึ่งก็คือความผิดพลาดกำลังสองเฉลี่ย w_i เป็นพารามิเตอร์น้ำหนักที่เชื่อมระหว่างสัญญาณขาเข้า i และปมแสดงสัญญาณขาออก ดังนั้น ค่า $\delta_i(t)$ นี้ ก็คือส่วนประกอบที่ i ของเวกเตอร์เกรเดียนต์ที่เวลา t ซึ่งจำเป็นต้องคำนวณเพื่อใช้ปรับพารามิเตอร์ตามปกติอยู่แล้ว

ขั้นที่ 2 ใช้ $\delta_i(t)$ ในขั้นที่ 1 คำนวณหาค่าเฉลี่ยเดลต้า (delta bar) $\bar{\delta}_i(T)$ จากสมการหาค่าเฉลี่ยตามเวลาของเดลต้า ดังนี้

$$\bar{\delta}_i(t) = (1-\theta)\delta_i(t) + \theta\bar{\delta}_i(t-1) \tag{3.3.2}$$

โดยที่ θ เป็นค่าคงที่สำหรับการถ่วงน้ำหนักให้กับค่าในอดีตมีค่าระหว่าง 0 กับ 1 และกำหนดค่าเริ่มต้น $\bar{\delta}_i(0) = \delta_i(0) = 0$ เพื่อให้ไม่มีผลจากค่าเริ่มต้น ค่าเฉลี่ยของเดลต้านี้เป็นตัวแทนของค่าเดลต้าในอดีตที่เกิดขึ้น ซึ่งก็คือ ค่าเฉลี่ยตามเวลาของอนุพันธ์ของฟังก์ชันเป้าหมายเทียบกับพารามิเตอร์ที่ i ที่ผ่านมาตั้งแต่เริ่มการปรับ โดยมี θ เป็นค่าแสดงน้ำหนักของอดีต ซึ่งทำให้ค่าเฉลี่ยเดลต้านี้เป็นผลรวมของเดลต้าในอดีตที่มี θ^k คูณอยู่ โดยที่ k ของแต่ละเทอมเป็นจำนวนรอบที่เดลต้าในอดีตนั้นเกิดห่างจากรอบปัจจุบัน ดังนั้น ค่าเดลต้าในอดีตที่อยู่ห่างจากรอบปัจจุบันจะยิ่งมีตัวคูณน้อยและทำให้มีผลต่อค่าเฉลี่ยเดลต้านี้น้อยลง ส่วนเทอมที่มีผลมากที่สุดก็คือเทอมของเดลต้าในรอบปัจจุบัน

ขั้นที่ 3 ปรับอัตราการเรียนรู้ของพารามิเตอร์แต่ละตัว โดยเปรียบเทียบค่าเดลต้าของรอบปัจจุบันและค่าเฉลี่ยเดลต้าในรอบก่อนปัจจุบัน ดังนี้

$$\Delta\beta_i(t) = \begin{cases} \kappa, & \delta_i(t)\delta_i(t-1) > 0 \\ -\varphi\beta_i(t-1), & \delta_i(t)\delta_i(t-1) < 0 \\ 0, & \textit{otherwise} \end{cases} \tag{3.3.3}$$

$$\beta_i(t) = \beta_i(t-1) + \Delta\beta_i(t) \tag{3.3.4}$$

โดยที่ k เป็นค่าคงตัวที่มีค่ามากกว่า 0 สำหรับการเพิ่มอัตราการเรียนรู้ (โดยการบวก) ϕ เป็นค่าคงตัวสำหรับการลดอัตราการเรียนรู้ มีค่าระหว่าง 0 กับ 1 ซึ่งจะถูกนำไปคูณกับอัตราการเรียนรู้ในรอบก่อนหน้า ดังนั้น การเพิ่มจะเป็นแบบเชิงเส้น คือเพิ่มด้วยอัตราคงที่ แต่ลดแบบเอ็กซ์โปเนนเชียล คือลดด้วยอัตรายกกำลังของค่าคงตัว จึงทำให้การลดลงทำได้เร็วกว่าการเพิ่ม ซึ่งขึ้นกับผลคูณของเดลต้าและค่าเฉลี่ยเดลต้า ถ้ารอบที่ติดกันมีการเปลี่ยนแปลงไปในทางเดียวกัน ผลคูณนี้ก็จะเป็นบวก อัตราการเรียนรู้จะเพิ่มขึ้น แต่ถ้ามีการเปลี่ยนแปลงต่างกัน อัตราการเรียนรู้ก็จะลดลง

ขั้นที่ 4 ปรับค่าพารามิเตอร์ $w_i(t)$ ของระบบ ตามการแผ่ย้อนกลับโดยค่าผิดพลาดโดยใช้อัตราการเรียนรู้ที่คำนวณได้ โดยใช้สมการ (3.1.9) ซึ่งเป็นสมการปรับพารามิเตอร์ในกรณีที่ใช้อัตราการเรียนรู้แยกกัน

ขั้นที่ 5 ทำตามขั้นตอนการแผ่ย้อนกลับโดยค่าผิดพลาดต่อไป จนจบรอบการเรียนรู้ และเมื่อเริ่มต้นรอบการเรียนรู้ใหม่ ให้ทำการคำนวณเดลต้าและค่าเฉลี่ยเดลต้าเพื่อคำนวณอัตราการเรียนรู้ตามขั้นตอนที่ 1 ถึง 3

วิธีการปรับอัตราการเรียนรู้ Dbd นี้ ต้องเลือกค่าพารามิเตอร์ของขั้นตอนวิธี 3 ตัว และประสิทธิภาพของขั้นตอนวิธีก็จะขึ้นกับพารามิเตอร์เหล่านี้อย่างมาก จึงจำเป็นต้องเลือกให้ค่าทั้งสามมีความเหมาะสมกัน สิ่งที่ควรระวังในการเลือกค่าพารามิเตอร์นี้คือ การเลือกค่า k ซึ่งเป็นค่าคงตัวสำหรับการเพิ่ม เพราะว่า ถ้าอัตราการเรียนรู้มีค่ามากเกินไปจะเป็นผลให้ระบบไม่มีเสถียรภาพ ดังที่กล่าวไว้ในบทที่ 2

โดยปกติ ค่า k จะถูกกำหนดให้มีค่าน้อยเพื่อให้ระบบสามารถเข้าสู่จุดเหมาะสมที่สุดได้ แต่ถ้าอัตราการเรียนรู้ถูกเพิ่มจนมีค่ามากเกินไป ระบบก็จะเสียเสถียรภาพไป ถึงแม้จะกำหนดให้มีการลดที่เร็วกว่าก็ตาม ดังนั้น สำหรับกรณีที่ระบบสามารถลู่เข้าสู่จุดเหมาะสมที่สุดได้โดยใช้วิธีนี้จะทำให้การลู่เข้าเร็วขึ้นได้ดีมาก [13] แต่ว่าในบางกรณีระบบก็อาจไม่ลู่เข้า ซึ่งจะต้องปรับให้ค่าพารามิเตอร์เหมาะสมมากขึ้น

3.5 การปรับอัตราการเรียนรู้แบบ Incremental Delta-Bar-Delta (Idbd)

การปรับอัตราการเรียนรู้แบบ Incremental Delta-bar-delta (Idbd) [3,4] จะกำหนดขั้นตอนขึ้นเพื่อให้การปรับอัตราการเรียนรู้ มีผลให้ฟังก์ชันเป้าหมาย $\varepsilon(t)$ เข้าสู่จุดเหมาะสมที่สุดเช่น

เดียวกับการปรับพารามิเตอร์ของระบบ โดยให้แต่ละพารามิเตอร์มีอัตราการเรียนรู้แยกกัน เพื่อให้การปรับเป็นอิสระต่อกัน เช่นเดียวกับในวิธี Dbd

การปรับอัตราการเรียนรู้จะใช้ความยาวก้าวเป็นค่าคงตัว ความยาวก้าวสำหรับการปรับของวิธีนี้ เรียกว่า meta step size ซึ่งเป็นพารามิเตอร์ที่ต้องกำหนดค่าให้ของวิธี ปกติจะกำหนดให้มีค่าน้อย เพื่อให้อัตราการเรียนรู้มีค่าอยู่ภายในขอบเขตที่ทำให้ลู่เข้าได้ ส่วนภายในขั้นตอนวิธี ยังได้กำหนดพารามิเตอร์เสริมเพื่อการคำนวณอัตราการเรียนรู้ในแต่ละรอบโดยเฉพาะอีก โดยกำหนดสัญญลักษณ์เป็น α และ h ตามลำดับ

วิธีการปรับอัตราการเรียนรู้แบบ Idbd สามารถแสดงเป็นขั้นตอน ได้ดังนี้

ขั้นที่ 1 กำหนดค่าเริ่มต้นให้กับพารามิเตอร์เสริมและอัตราการเรียนรู้ทุกตัว พารามิเตอร์ α และ h จะถูกกำหนดขึ้นตามจำนวนอัตราการเรียนรู้ β ซึ่งมีจำนวนเท่ากับจำนวนพารามิเตอร์ในระบบ N ดังนั้นจำนวนพารามิเตอร์เสริม α และ h จะมีอย่างละ N ตัวเช่นกัน การกำหนดค่าเริ่มต้นให้กับพารามิเตอร์เสริมและอัตราการเรียนรู้เหล่านี้ สามารถทำได้โดยใช้จำนวนพารามิเตอร์ของระบบมากำหนด ดังนี้

$$\alpha_i(0) = \log\frac{1}{N}, \quad 1 \le i \le N$$

และ

$$h_i(0) = 0, \qquad 1 \le i \le N \tag{3.5.1}$$

โดยที่ N เป็นจำนวนพารามิเตอร์ในระบบ และกำหนดค่าอัตราการเรียนรู้เริ่มต้นจากพารามิเตอร์เสริม α ดังนี้

$$\beta_i(0) = \exp(\alpha_i(0)), \quad 1 \le i \le N \tag{3.5.2}$$

ขั้นที่ 2 คำนวณความผิดพลาดระหว่างสัญญาณขาออก $o(t)$ และสัญญาณอ้างอิง $d(t)$

$$e(t) = d(t) - o(t) \tag{3.5.3}$$

เพื่อใช้ในการคำนวณค่าพารามิเตอร์เสริม $h_i(t)$ ตามสมการ

$$h_i(t) = h_i(t-1)\left[1 - \beta_i(t-1)x_i^2(t-1)\right]^+ + \beta_i(t-1)e(t-1)x_i(t-1), \qquad 1 \le i \le N \tag{3.5.4}$$

เทอมในวงเล็บที่มีเครื่องหมายบวก กำหนดให้มีค่าดังนี้

$$[z]^+ = \begin{cases} z, & z > 0 \\ 0, & otherwise \end{cases} \tag{3.5.5}$$

ขั้นที่ 3 ใช้ค่า $h_i(t)$ ในขั้นที่ 2 คำนวณค่า $\alpha_i(t)$ ตามสมการ

$$\alpha_i(t) = \alpha_i(t-1) + \frac{\eta e(t-1)x_i(t-1)h_i(t-1)}{\sqrt{\beta_i(t-1)}} \tag{3.5.6}$$

โดยที่ η เป็นค่า meta step size สำหรับการปรับอัตราการเรียนรู้

ขั้นที่ 4 ปรับอัตราการเรียนรู้ด้วยค่า $\alpha_i(t)$ ที่ได้จากขั้นที่ 3 ตามสมการ

$$\beta_i(t) = \exp\left(\alpha_i(0)\right) \tag{3.5.7}$$

. โดยที่ $\beta_i(t)$ เป็นอัตราการเรียนรู้ของพารามิเตอร์ $w_i(t)$

ขั้นที่ 5 ปรับค่าพารามิเตอร์ $w_i(t)$ ของระบบ ตามการแผ่ย้อนกลับโดยค่าผิดพลาดโดยใช้อัตราการเรียนรู้ที่คำนวณได้ โดยใช้สมการ (3.1.9) ซึ่งเป็นสมการปรับพารามิเตอร์ในกรณีที่ใช้อัตราการเรียนรู้แยกกัน

ขั้นที่ 6 ทำตามขั้นตอนการแผ่ย้อนกลับโดยค่าผิดพลาดต่อไป จนจบรอบการเรียนรู้ และเมื่อเริ่มต้นรอบการเรียนรู้ใหม่ ให้ทำการคำนวณเดลต้าและค่าเฉลี่ยเดลต้าเพื่อคำนวณอัตราการเรียนรู้ตามขั้นตอนที่ 1 ถึง 4

ลักษณะการใช้อัตราการเรียนรู้ของวิธีนี้ใช้อัตราการเรียนรู้แยกกันเช่นเดียวกับวิธี Dbd แต่ใช้การหาจุดเหมาะสมที่สุด แทนการปรับให้เพิ่มหรือลดตามการเปลี่ยนแปลงของพารามิเตอร์ ขั้นตอนวิธีจะลู่เข้าเมื่อเลือกค่า meta step size ไม่มากเกินขอบเขตที่ทำให้ลู่เข้า และทำให้อัตราการเรียนรู้จะมีการเปลี่ยนแปลงน้อย การเลือก meta step size จะต้องเลือกให้อัตราการเรียนรู้ที่ทำ

ให้การปรับค่าของทุกพารามิเตอร์ลู่เข้า ถ้าเลือกให้มีค่ามากเกินไปอาจทำให้ระบบขาดเสถียรภาพได้

การรักษาเสถียรภาพของระบบวิธีหนึ่งที่ใช้ในขั้นตอนวิธีคือ การควบคุมค่า h การปรับค่านี้จะถูกกำหนดไว้ว่า ถ้าผลคูณ $\beta_i(t-1)x_i^2(t-1)$ ในสมการ (3.5.4) มีค่ามากกว่า 1 แล้วค่า h ในรอบถัดไปจะไม่เพิ่มขึ้นจากรอบก่อน แต่เป็นผลคูณของสัญญาณขาเข้า $x_i(t-1)$, ค่าผิดพลาด $e(t-1)$ และอัตราการเรียนรู้ $\beta_i(t-1)$ แทน เพื่อไม่ให้ค่าของ h เพิ่มขึ้นเมื่อ $\beta_i(t-1)$ มีค่าสูง และทำให้ $\beta_i(t)$ มีค่าต่ำลง

3.5 การปรับอัตราการเรียนรู้แบบ Entropy-based Learning Rate

การปรับอัตราการเรียนรู้โดยวิธีนี้ จะอาศัยทฤษฎี

"เมื่อขั้นตอนวิธีการลดระดับตามแนวเกรเดียนต์ ปรับค่าพารามิเตอร์ในแบบเชื่อมตรงให้กับข่ายงานระบบประสาทแบบชั้นเดียว (Single Layer Perceptron) เมื่อปรับค่าพารามิเตอร์มาถึงค่าเหมาะสมที่สุดของฟังก์ชันเป้าหมายและอยู่ในสถานะอยู่ตัว (steady state) แล้ว ค่าพารามิเตอร์ที่ได้จากขั้นตอนวิธีจะกระเพื่อมอยู่รอบจุดเหมาะสมที่สุด โดยมีค่าความแปรปรวน (variance) ที่เท่ากัน แต่จะไม่มีสหสัมพันธ์ (Uncorrelate) กัน" [8]

จากทฤษฎีข้างต้น เราสามารถใช้ระดับความเหมือนกัน (degree of uniformity) ของความแปรปรวนของค่าพารามิเตอร์ $w_i(t)$ เพื่อแสดงความใกล้กับค่าเหมาะสมที่สุด ถ้าระดับความเหมือนกันของค่าความแปรปรวน (variance) มากเท่าไร ก็แสดงว่าค่าอัตราการเรียนรู้จากขั้นตอนวิธีอยู่ใกล้กับค่าเหมาะสมที่สุดเท่านั้น

การกำหนดให้อัตราการเรียนรู้มีค่ามากเมื่ออยู่ไกลและมีค่าน้อยเมื่ออยู่ใกล้จุดเหมาะสมที่สุด สามารถทำได้โดยใช้ฟังก์ชันเอนโทรปี (Entropy Function) มากำหนดค่าของอัตราการเรียนรู้นี้ ตามสมการ [8,9]

$$\beta(t) = 1 - \frac{H(t-1)}{\ln(N)} \tag{3.6.1}$$

โดยที่ $H(t)$ เป็นฟังก์ชันเอนโทรปี กำหนดโดย

$$H(t) = \sum_{i=1}^{N} -(\upsilon_i)_n \ln(\upsilon_i)_n \qquad (3.6.2)$$

โดยที่ $$(\upsilon_i)_n = \upsilon_i \Big/ \sum_{i=1}^{N} \upsilon_i \qquad (3.6.3)$$

เราสามารถประมาณค่าความแปรปรวนของพารามิเตอร์ $\upsilon_i(t)$ ที่เวลาใด ๆ ได้จากสมการสำหรับการหาค่าเฉลี่ยตามเวลา ดังนี้

$$\upsilon_i(t) = (1-\theta)\upsilon_i(t) + \theta\left[w_i(t) - \overline{w}_i(t)\right]^2$$
$$\overline{w}_i(t) = (1-\theta)\overline{w}_i(t-1) + \theta w_i(t) \qquad (3.5.4)$$

โดยที่ $\upsilon_i(t)$ เป็นค่าความแปรปรวนของค่าพารามิเตอร์ $w_i(t)$ ที่เวลา t, $w_i(t)$ เป็นค่าพารามิเตอร์ที่เวลา t, $\overline{w}_i(t)$ เป็นค่าเฉลี่ยของค่าพารามิเตอร์ $w_i(t)$ ที่เวลา t, θ เป็นค่าถ่วงน้ำหนักสำหรับค่าในอดีตของการหาค่าเฉลี่ย ซึ่งเป็นค่าคงตัวที่มีค่าอยู่ระหว่าง 0 กับ 1

วิธีการปรับอัตราการเรียนรู้แบบ Entropy-based learning rate นี้ จะใช้อัตราการเรียนรู้เพียงค่าเดียวกับพารามิเตอร์ทุกตัว เราสามารถแสดงเป็นขั้นตอน ได้ดังนี้

ขั้นที่ 1 กำหนดค่าเริ่มต้น สำหรับค่าเฉลี่ยของพารามิเตอร์และความแปรปรวน ซึ่งแทนด้วยสัญญลักษณ์ $\overline{w}_i(0)$, $v_i(0)$ ตามลำดับ

ขั้นที่ 2 หาค่าเฉลี่ยของพารามิเตอร์และความแปรปรวนของพารามิเตอร์ โดยใช้สมการหาค่าเฉลี่ยตามเวลา ดังแสดงในสมการ (3.6.4)

ขั้นที่ 3 คำนวณค่าผลรวมของความแปรปรวน เพื่อสร้างค่าปกติของความแปรปรวนของพารามิเตอร์ ดังแสดงในสมการ (3.6.3)

ขั้นที่ 4 คำนวณค่าฟังก์ชันเอนโทรปีจากค่าปกติของความแปรปรวนของพารามิเตอร์ ดังแสดงในสมการ (3.6.2)

ขั้นที่ 5 คำนวณค่าอัตราการเรียนรู้ โดยใช้ฟังก์ชันเอนโทรปี ดังแสดงในสมการ (3.6.1)

วิธีการปรับอัตราการเรียนรู้นี้ ใช้ระดับความเหมือนกันของความแปรปรวนมาวัดความใกล้จุดเหมาะสมที่สุด หมายความว่า ความแปรปรวนของพารามิเตอร์แต่ละตัวจะมีค่าเข้าใกล้กัน

มากขึ้นเมื่อค่าพารามิเตอร์ลู่เข้าใกล้จุดเหมาะสมที่สุดมากขึ้น ดังนั้น เราสามารถวัดความใกล้จุดเหมาะสมที่สุดได้จากการกระจายของความแปรปรวนของพารามิเตอร์ ซึ่งถูกเรียกว่า ระดับความเหมือนกันของความแปรปรวน

การวัดระดับความเหมือนกันของความแปรปรวน จะใช้ฟังก์ชันเอนโทรปี ดังแสดงในสมการ (3.5.2) ซึ่งมีค่าสูงสุดเท่ากับ $\ln(N)$ โดยเกิดขึ้นเมื่อความแปรปรวนของพารามิเตอร์ทุกตัวมีค่าเท่ากัน $\left(\upsilon_1 = \upsilon_2 = \ldots = \upsilon_n\right)$ และกำหนดอัตราการเรียนรู้ดังแสดงในสมการ (3.6.1) อัตราการเรียนรู้จะมีค่าตั้งแต่ 0 ถึง 1 ทำให้อัตราการเรียนรู้เข้าใกล้ 1 เมื่ออยู่ไกลจากจุดเหมาะสมที่สุดและลดลงใกล้ศูนย์มากขึ้นเมื่อเข้าใกล้จุดเหมาะสมที่สุด

3.7 การปรับอัตราการเรียนรู้แบบ direction cosine และ modified direction cosine

การปรับอัตราการเรียนรู้แบบ direction cosine ให้อัตราการเรียนรู้ที่มีค่าลบได้ ดังนั้น จึงควรมีการปรับปรุงเพื่อให้อัตราการเรียนรู้เป็นบวกเสมอและสามารถใช้งานได้ วิธีที่ปรับปรุงใหม่เรียกว่า modified direction cosine ซึ่งจะกล่าวถึงแยกกัน

3.7.1 การปรับอัตราการเรียนรู้แบบ direction cosine

การปรับอัตราการเรียนรู้แบบ direction cosine ซึ่งเสนอโดย Franzini [14] โดยอาศัยโคซายน์แสดงทิศทาง (direction cosine) ซึ่งเป็นผลคูณภายในของ $\Delta\underline{w}_n(t)$ และ $\Delta\underline{w}_n(t-1)$ ซึ่งเป็นเวกเตอร์หน่วย (unit vector) ของการเปลี่ยนแปลงพารามิเตอร์ของรอบการคำนวณที่อยู่ติดกัน ดังนี้

$$\beta(t) = \frac{\Delta\underline{w}^T(t)\Delta\underline{w}(t-1)}{\|\Delta\underline{w}(t)\|\,\|\Delta\underline{w}(t-1)\|} \qquad (3.7.1)$$

แต่เนื่องจากโคซายน์แสดงทิศทางมีค่าอยู่ในช่วง -1 ถึง 1 ทำให้อัตราการเรียนรู้ที่ได้จากวิธีทั้งสองมีโอกาสเป็นลบ ซึ่งทำให้ฟังก์ชันเป้าหมาย $\varepsilon(t)$ มีค่าเพิ่มขึ้น และค่าพารามิเตอร์อาจไม่ลู่เข้าได้ ในการใช้งานจริงจึงจำเป็นต้องปรับปรุงเพื่อให้อัตราการเรียนรู้เป็นบวกเสมอ นอกจากนี้ การใช้เวกเตอร์พารามิเตอร์ที่เปลี่ยนแปลง ทำให้ไม่สามารถใช้เวกเตอร์การปรับเปลี่ยนพารามิเตอร์ในรอบปัจจุบันได้ เพราะว่า การคำนวณหาค่าอัตราการเรียนรู้เกิดก่อนการปรับพารามิเตอร์ จึงจำเป็นต้องใช้การปรับพารามิเตอร์ของรอบก่อนปัจจุบัน ซึ่งทำให้อัตราการเรียนรู้ที่ได้ไม่เหมาะสมกับการเปลี่ยนแปลงในรอบปัจจุบัน เพราะค่าที่นำมาคำนวณเป็นรอบในอดีต

3.7.2 การปรับอัตราการเรียนรู้แบบ modified direction cosine

สำหรับในวิทยานิพนธ์ฉบับนี้ ได้นำวิธี direction cosine มาปรับปรุงโดยใช้การเปลี่ยนแปลงพารามิเตอร์เป็นเกรเดียนต์ เพราะว่า การคำนวณ direction cosine เพียงต้องการทิศทางของการปรับพารามิเตอร์ ซึ่งเป็นทิศตรงข้ามของเกรเดียนต์ ดังนี้

$$\rho(t) = \frac{\left(\nabla_{\underline{w}}\varepsilon(t)\right)^T\left(\nabla_{\underline{w}}\varepsilon(t-1)\right)}{\left\|\nabla_{\underline{w}}\varepsilon(t)\right\|\left\|\nabla_{\underline{w}}\varepsilon(t-1)\right\|} \qquad (3.7.2)$$

ทำให้สามารถรับรู้ถึงทิศทางการเปลี่ยนแปลงพารามิเตอร์ของรอบปัจจุบัน และใช้เงื่อนไขเพื่อให้อัตราการเรียนรู้ที่ได้มีค่าเป็นบวกเสมอ ดังนี้

$$\beta(t) = \begin{cases} \rho(t) & , \rho(t) \geq 0 \\ 1+\rho(t) & , \rho(t) < 0 \end{cases} \qquad (3.7.3)$$

ซึ่งเป็นเงื่อนไขที่เปลี่ยนค่าอัตราการเรียนรู้โดยอาศัยค่าเดิม โดยที่ เมื่อ $\rho(t)$ มีค่ามากกว่าหรือเท่ากับ 0 จะใช้เป็นค่าเดิม แต่เมื่อมีค่าน้อยกว่า 0 จะให้ลบออกจาก 1 เพราะว่า เมื่อ $\rho(t)$ มีค่าเป็นลบมาก แสดงว่า ทิศทางการปรับตัวในรอบก่อนหน้าและรอบปัจจุบันต่างกันมาก จึงควรให้อัตราการเรียนรู้มีค่าน้อย วิธีที่มีการปรับปรุงจากการปรับอัตราการเรียนรู้แบบ direction cosine นี้ จะเรียกว่า modified direction cosine

การปรับอัตราการเรียนรู้แบบ modified direction cosine แสดงเป็นขั้นตอนได้ดังนี้

ขั้นที่ 1 กำหนดค่าเริ่มต้นของอัตราการเรียนรู้ $\beta(0)$ ให้มีค่าเป็นบวกน้อยๆ เพื่อใช้ในการปรับพารามิเตอร์ในรอบแรก โดยไม่ทำให้ระบบเสียเสถียรภาพ

ขั้นที่ 2 หาเกรเดียนต์สำหรับการปรับพารามิเตอร์ในรอบปัจจุบัน และเก็บเวกเตอร์เกรเดียนต์ที่หาได้ในรอบแรกเพื่อใช้หาค่าอัตราการเรียนรู้ในรอบต่อไป

ขั้นที่ 3 ปรับพารามิเตอร์ในรอบแรกด้วยอัตราการเรียนรู้เริ่มต้นที่กำหนดไว้ ตามสมการ (3.1.7) ซึ่งเป็นการปรับพารามิเตอร์ด้วยอัตราการเรียนรู้เพียงค่าเดียว

ขั้นที่ 4 ทำตามขั้นตอนการแผ่ย้อนกลับโดยค่าผิดพลาด จนจบรอบการคำนวณ และขึ้นรอบการคำนวณใหม่

ขั้นที่ 5 ก่อนปรับพารามิเตอร์ของรอบการคำนวณให้คำนวณ $\rho(t)$ ของรอบปัจจุบัน โดยใช้เกรเดียนต์รอบก่อนหน้าที่เก็บไว้ และเกรเดียนต์ที่หาได้ในรอบปัจจุบัน ตามสมการ (3.7.2)

ขั้นที่ 6 คำนวณอัตราการเรียนรู้ $\beta(t)$ จาก $\rho(t)$ ที่คำนวณได้ในขั้นที่ 5 ตามเงื่อนไขและการหาค่าอัตราการเรียนรู้ในสมการ (3.7.3)

ขั้นที่ 7 ปรับพารามิเตอร์ด้วยอัตราการเรียนรู้ที่ได้และทำจนจบรอบการเรียนรู้ เมื่อขึ้นรอบการเรียนรู้ใหม่ ให้คำนวณหาค่าอัตราการเรียนรู้ตามขั้นที่ 5 และ 6 เพื่อคำนวณอัตราการเรียนรู้ จนครบจำนวนรอบที่กำหนดไว้

วิธีนี้เป็นการใช้ทิศทางของการปรับพารามิเตอร์ เพื่อใช้ปรับอัตราการเรียนรู้ให้เหมาะสม โดยมีหลักการว่า ในรอบการคำนวณที่ติดกัน ถ้าการปรับพารามิเตอร์เป็นไปในทิศเดียวกัน แสดงว่า พื้นผิวฟังก์ชันเป้าหมายที่จุดปัจจุบันมีความโค้ง (curvature) น้อย สามารถก้าวเข้าใกล้จุดเหมาะสมที่สุดโดยง่ายจึงควรให้อัตราการเรียนรู้ที่มีค่าสูง ในขณะที่ถ้าการปรับพารามิเตอร์มีทิศทางที่แตกต่างกันมาก แสดงว่าความโค้งของพื้นผิวที่จุดปัจจุบันมีมาก อัตราการเรียนรู้ควรจะน้อย เพื่อไม่ให้ค่าฟังก์ชันเป้าหมายเพิ่มขึ้น หรือข้ามจุดเหมาะสมที่สุดไป

3.8 การคำนวณและความซับซ้อนของขั้นตอนวิธี

จากการปรับอัตราการเรียนรู้ที่แสดงไว้ในหัวข้อที่ผ่านมา แต่ละวิธีจะมีความซับซ้อน (complexity) และการคำนวณ (computation) ต่างกัน ความซับซ้อนจะแสดงถึงหน่วยความจำที่จำเป็นต้องใช้เพื่อทำตามขั้นตอนวิธี ส่วนการคำนวณที่จะต้องเกิดขึ้นในแต่ละรอบ แยกเป็นการบวก การคูณ และการใช้ฟังก์ชันไม่เชิงเส้น โดยถือว่าการเปรียบเทียบและการบวกจะเป็นการคำนวณที่เทียบเท่ากัน

ตารางที่ 3.2 สรุปความซับซ้อนและการคำนวณของแต่ละขั้นตอนวิธี โดยแสดงสัมพันธ์กับจำนวนพารามิเตอร์ในระบบ (N) ในตารางแสดงความซับซ้อนไว้ในหัวข้อ “หน่วยความจำ”

ตารางที่ 3.2 ความซับซ้อนและการคำนวณของขั้นตอนวิธี

วิธีการ	หน่วยความจำ	การคำนวณ		
		การบวกและเปรียบเทียบ	การคูณ	การคำนวณฟังก์ชันไม่เชิงเส้น
Dbd	$2N$	$4N$	$4N$	ไม่มี
Idbd	$3N$	$4N$	$9N$	$2N+1$
Entropy	$2N$	$5N+1$	$6N+1$	$1N$
direction cosine	$2N$	$3N-3$	$3N+1$	2
modified direction cosine	$2N+1$	$3N-1$	$3N+1$	2

ข้อมูลในตาราง ได้ใช้สมการดังที่แสดงไว้ด้านบนและทำการนับ และเทียบกับเอกสารที่มีอยู่ เช่นวิธี Idbd ได้เทียบกับเอกสารอ้างอิง [3] ซึ่งได้แสดงการบวกและการคูณรวมกันไว้เป็นจำนวน $13n$ และหน่วยความจำเป็น $3n$ ส่วนวิธีอื่น ๆ ได้นับจากสมการที่มีอยู่ในเอกสารอ้างอิง [2, 7, 9] ซึ่งเป็นที่มาของสมการที่แสดงไว้ในบทนี้

จากตาราง จะเห็นว่า วิธี Dbd จะมีความซับซ้อนและการคำนวณน้อยกว่าวิธีอื่น และไม่มีการใช้ฟังก์ชันไม่เชิงเส้น เพราะว่าเป็นวิธีที่ใช้การเปรียบเทียบเพื่อปรับอัตราการเรียนรู้ ในขณะที่วิธีอื่นใช้การคำนวณ แต่ว่าประสิทธิภาพของวิธี Dbd ขึ้นกับค่าพารามิเตอร์ของขั้นตอน ทำให้ผู้ใช้งานต้องทำการเลือกค่าที่เหมาะสมโดยวิธีทดสอบดู

ขั้นตอนวิธีที่นำมาทดสอบในวิทยานิพนธ์ฉบับนี้ ได้เลือกขั้นตอนวิธีที่มีความซับซ้อนและการคำนวณในอันดับหนึ่ง ($O(n)$) มาใช้ทดสอบร่วมกัน โดยละวิธีอื่น ๆ ที่มีความซับซ้อนและการคำนวณในอันดับสอง ($O(n^2)$) หรือมากกว่าไป เนื่องจาก ในวิทยานิพนธ์ฉบับนี้เน้นการใช้งานแบบเชื่อมตรง (on-line) ซึ่งต้องการขั้นตอนวิธีที่มีการคำนวณและความซับซ้อนน้อย เพื่อลดช่วงเวลาที่ใช้เพื่อปรับพารามิเตอร์ของระบบ

3.9 สรุป

ในบทนี้ ได้แสดงถึงขั้นตอนและรายละเอียด รวมทั้งแนวความคิดในการสร้างของวิธี ซึ่งจะประกอบด้วยการใช้อัตราการเรียนรู้เป็นค่าคงตัวและการปรับอัตราการเรียนรู้แบบพลวัต 4 วิธี คือ Delta-bar-delta (Dbd), Incremental Delta-bar-delta (Idbd), Entropy-based learning rate (Entropy) และ modified direction cosine แต่ละวิธีมีแนวความคิดในการสร้างแตกต่างกัน

ไป วิธี Dbd ใช้การเปลี่ยนแปลงของพารามิเตอร์, วิธี Idbd ใช้การหาจุดเหมาะสมที่สุด ส่วนวิธี Entropy ใช้การวัดระดับความเหมือนกันของความแปรปรวน และ modified direction cosine ใช้ทิศทางของการปรับพารามิเตอร์ในรอบติดกัน ข้อมูลที่กล่าวในบทนี้ เป็นแนวทางสำหรับวิเคราะห์ผลการทดสอบในบทต่อไป