บทที่ 4

การเรียนรู้แบบย้อนกลับ

4.1 ประวัติของการเรียนรู้แบบย้อนกลับ

ความคิดพื้นฐานแรกเริ่มของการเรียนรู้แบบย้อนกลับ (Backpropagation หรือ BP ซึ่งนิยมเรียกกันโดยทั่วไป) ถูกนำเสนอในปี ค.ศ. 1974 โดย Paul Werbos จากนั้นก็มีการนำเสนอใหม่ โดย David Parker ในปี ค.ศ. 1982 และ ถูกนำเสนอต่อผู้อ่านอย่างกว้างขวาง โดย Rumelhart And McClelland ในปี ค.ศ.1986 ในหนังสือที่ชื่อ Parallel Distributed Processing ซึ่งกล่าวถึง ศักยภาพของนิวรัล เน็ทเวิร์ค และ Back-Error Propagation Network

งานประยุกต์แรกเริ่มของการเรียนรู้แบบย้อนกลับ สร้างโดย Terry Sejnowski และคณะในมหาวิทยาลัย Johns Hopkins โดยเป็นโปรแกรม NETTALK สร้างโดย Terry Sejnowski และ Rosenberg

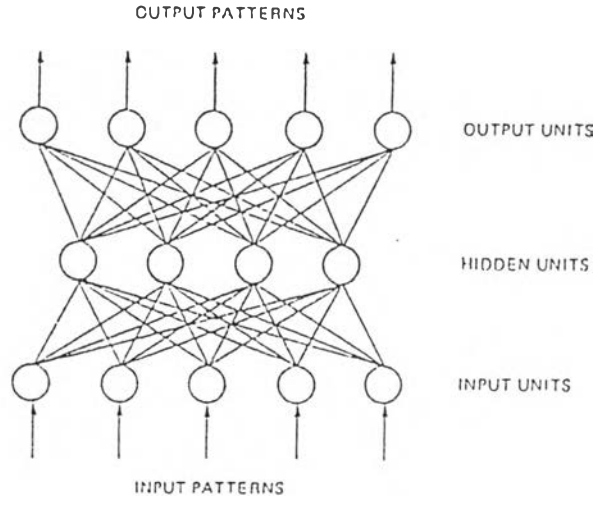
การเรียนรู้แบบย้อนกลับ เป็นรูปแบบหนึ่งของนิวรัล เน็ทเวิร์ค ที่ใช้กันอย่างกว้างขวาง และถูกนำมาประยุกต์ใช้กับการศึกษางานประเภทต่างๆ เช่น ทางการทหาร การช่วยในการวินิจฉัยทางการแพทย์ การจดจำคำพูด การจดจำตัวอักษร หุ่นยนต์

การเรียนรู้แบบย้อนกลับ สามารถแก้ปัญหาที่ต้องการรูปแบบ โดยการป้อนรูปแบบเข้าไป นิวรัล เน็ทเวิร์ค ก็จะผลิตรูปแบบผลลัพธ์ที่เกี่ยวข้องกันออกมา (Judith E. Dayhoff,1990)

การเรียนรู้แบบย้อนกลับ เป็นวิธีการหนึ่งของ นิวรัล เน็ทเวิร์ค ที่ง่ายต่อการเข้าใจ เนื่องจากกระบวนการเรียนรู้และปรับปรุงแก้ไขนั้นเป็นไปด้วยตัวมันเอง ถ้าเน็ทเวิร์คให้คำตอบที่ผิด ดังนั้น ค่าน้ำหนัก จะถูกแก้ไขจนกว่า ค่าความผิดพลาดจะลดน้อยลง หรืออยู่ในเกณฑ์ที่ยอมรับได้ นั่นก็คือค่าได้ในครั้งต่อๆไปจะมีความถูกต้องมากยิ่งขึ้น

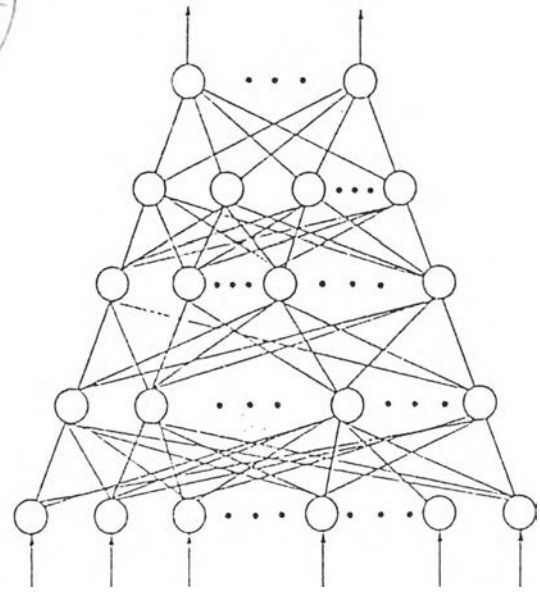
4.2 โครงสร้างของการเรียนรู้แบบย้อนกลับ

การเรียนรู้แบบย้อนกลับ มีโครงสร้างเป็นชั้นๆ โดยมีโครงสร้างอย่างง่าย 3 ชั้น คือ ชั้นข้อมูลเข้า ชั้นแอบแฝง ชั้นแสดงผลลัพธ์ แต่ละชั้น จะติดต่อกันอย่างสมบูรณ์ ดังรูป 4-1



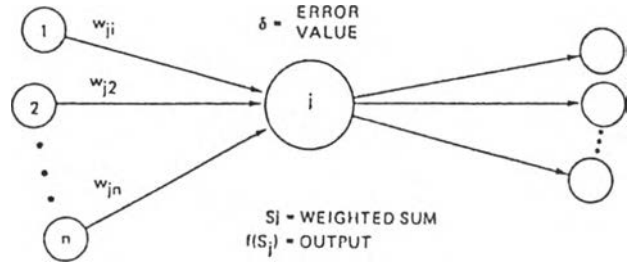
รูป 4-1 แสดงการเรียนรู้แบบย้อนกลับ ชนิด 3 ชั้น

จากรูป 4-1 แสดงรูปแบบ(Topology) ของการเรียนรู้แบบย้อนกลับ ชนิด 3 ชั้น ชั้นล่างคือชั้นข้อมูลเข้า ซึ่งนำเข้าข้อมูลจากภายนอก ชั้นถัดมาคือชั้นแอบแฝง ซึ่งจะติดต่อกับทั้ง ชั้นข้อมูลเข้า และ ชั้นผลลัพธ์ ที่อยู่ชั้นบน



รูป 4-2 แสดงการเรียนรู้แบบย้อนกลับ ชนิด 5 ชั้น

เป็นตัวอย่างของการเรียนรู้แบบย้อนกลับ ชนิด 5 ชั้น ที่ติดต่อกันอย่างสมบูรณ์ โดยมีชั้นแอบแฝง อยู่ 3 ชั้น



รูป 4-3 แสดงแต่ละโนดภายในชั้นแอบแฝง

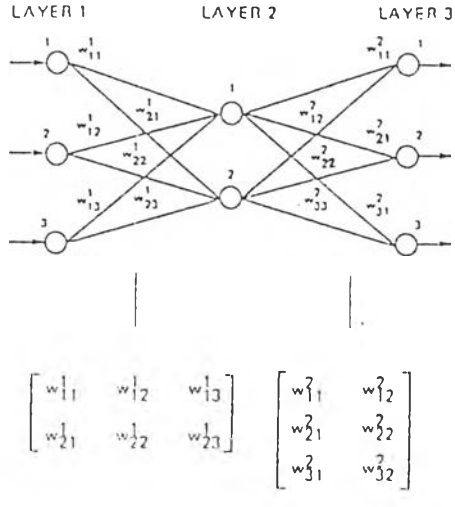
จะเห็นว่าข้อมูลนำเข้าอยู่ทางด้านซ้าย ส่วนทางด้านขวาคือหน่วยที่รับผลลัพธ์จากโนด j

หน่วยประมวลผลกลาง(Processing Unit) ประกอบด้วย

- ผลรวมถ่วงน้ำหนักของข้อมูลนำเข้า (S_j)
- ผลลัพธ์ (X_j)
- ค่าความผิดพลาด ($\mathcal{E}_j$) ซึ่งถูกใช้ระหว่างการการปรับค่าน้ำหนัก

ค่าน้ำหนักที่เกี่ยวเนื่องกับการติดต่อแต่ละชั้น จะถูกปรับแต่งตลอดเวลาระหว่างการเรียนรู้ เพื่อให้ลดความแตกต่างระหว่างผลลัพธ์ที่ได้กับผลลัพธ์ที่ต้องการ

กำหนดให้ W_{ji} หมายถึง ค่าน้ำหนักจากหน่วยที่ i ไปยัง หน่วยที่ j



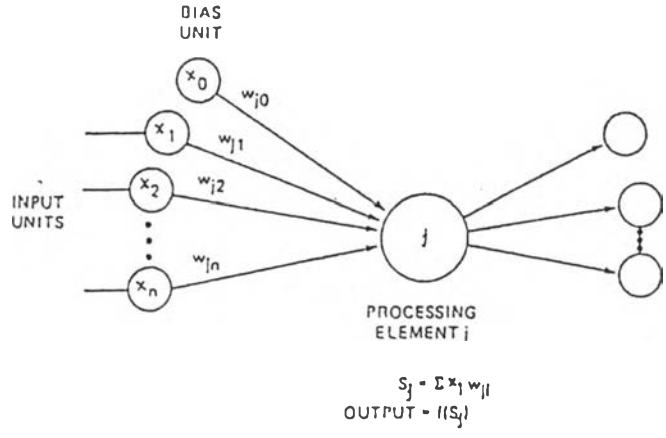
รูป 4-4 แสดงการแทนค่าน้ำหนักด้วยเมตริกซ์

จากรูป เป็นการแทนค่าน้ำหนักด้วย เมตริกซ์ โดยมีค่ากำกับบน (Superscript) บ่งบอกถึงชั้นที่ต่างๆกัน เช่น ${}^{1}W_{21}$ หมายถึงค่าน้ำหนักจากหน่วยประมวลผลที่ 2 ในระดับชั้นที่ 1 ไปยัง หน่วยประมวลผลที่ 1 ในระดับที่ 2

4.3 ขั้นตอนการเรียนรู้ของการเรียนรู้แบบย้อนกลับ

ประกอบด้วย 2 ขั้นตอน

4.3.1 การแพร่เดินหน้า (Forward-Propagation)



รูป 4-5 แสดงขั้นตอนของการแพร่เดินหน้า

จากรูปเป็นการคำนวณผลรวมของผลลัพธ์ที่เข้ามายังหน่วยที่ j ดังสมการ

$$S_j = \sum_i X_i W_{ji} \qquad \text{----- (4.1)}$$

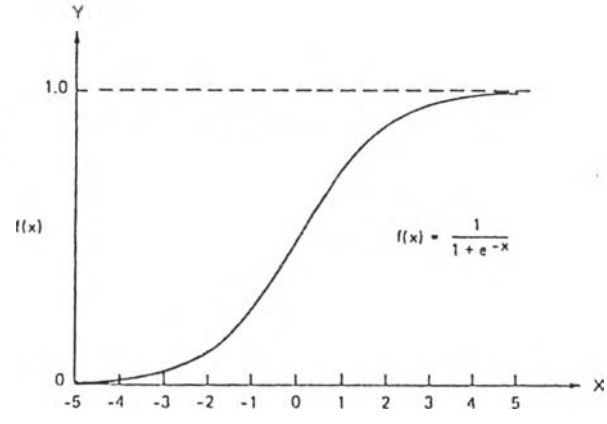
โดยที่ X_i = ข้อมูลจากหน่วยที่ i

W_{ji} = ค่าน้ำหนักจากหน่วยที่ i ไปยังหน่วยที่ j

เมื่อคำนวณจนได้ค่า S_j ก็ทำการคำนวณหาค่า $f(S_j)$ อีกครั้ง

โดย $f(x) = \dfrac{1}{1 + e^{-x}}$ ซึ่งเป็นสมการของฟังก์ชันซิกมอยด์

ดังนั้น $f(S_j) = \dfrac{1}{1 + e^{-S_j}}$ ----- (4.2)



รูป 4-6 กราฟของฟังก์ชันซิกมอยด์

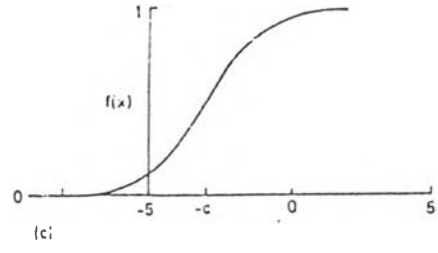
เมื่อได้ค่า $f(S_j)$ แล้ว ค่า $f(S_j)$ ก็จะกลายเป็นผลลัพธ์ของหน่วยที่ j ซึ่งก็คือค่า X_j ดังรูป 4-5 โดยจะส่งออกไปทางหน่วยอื่นๆ ด้วยค่า X_j ที่เท่ากัน

จากรูป 4-5 ค่าเอนเอียง (Bias Unit) เป็นค่าคงที่ที่เราใส่เข้าไป เพื่อให้การเรียนรู้ ของเน็ทเวิร์คเร็วขึ้น หรือที่เรียกว่า Convergence Time เร็วขึ้น กล่าวคือ ค่าเอนเอียง ยังมีผลต่อระดับการกระตุ้น (Threshold) ด้วย เช่น ให้ค่า C = 5 ทำให้กราฟขยับไปทางซ้าย 5 หน่วย ดังรูป 4-7

$$C = W_{jo}$$

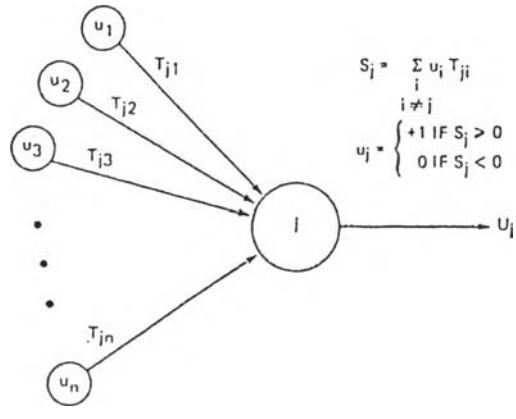
ให้ $S = \sum X_i W_{ji}$

ดังนั้น ผลรวม $= S + C$ ----- (4.3)



รูป 4-7 แสดงผลของค่าเอนเอียงที่มีต่อการเรียนรู้

สาเหตุที่ต้องใช้ฟังก์ชันซิกมอยด์ ก็เนื่องจาก ต้องการให้เป็นระดับการกระตุ้นแบบอ่อน (Soft Threshold) มากกว่าการกระตุ้นแบบรุนแรง (Hard Threshold) (ดังเช่น ฟังก์ชันขั้นบันได ดังรูป 4-8) นั่นคือ ฟังก์ชันซิกมอยด์ให้ค่าที่ต่อเนื่องกัน



รูป 4-8 แสดงนิวรัล เน็ทเวิร์คแบบที่ใช้ฟังก์ชันขั้นบันได

จากรูป 4-8 ฟังก์ชันขั้นบันได คือ

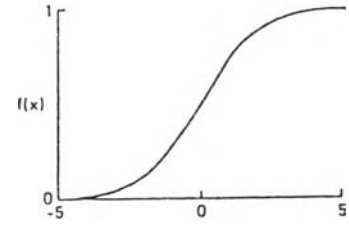
$S_j >= 0$ ถ้า $U_j = 1$

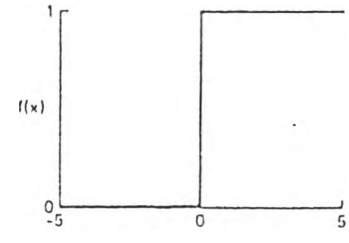
$S_j < 0$ ถ้า $U_j = 0$

$$S_j = \sum_{i} U_i T_{ji}$$

$i != j$

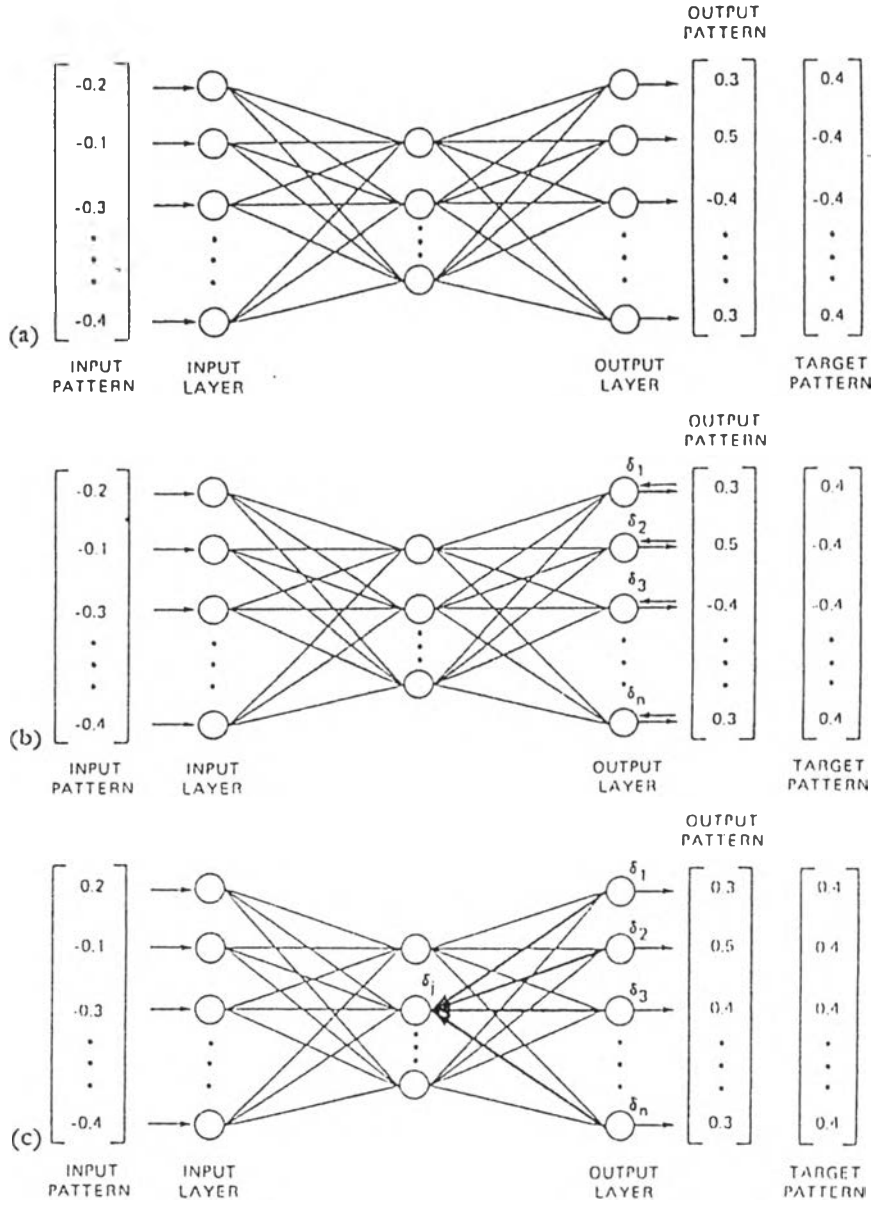
ซึ่งให้ผลลัพธ์ U_j 2 ค่า เท่านั้น คือ 0 , 1



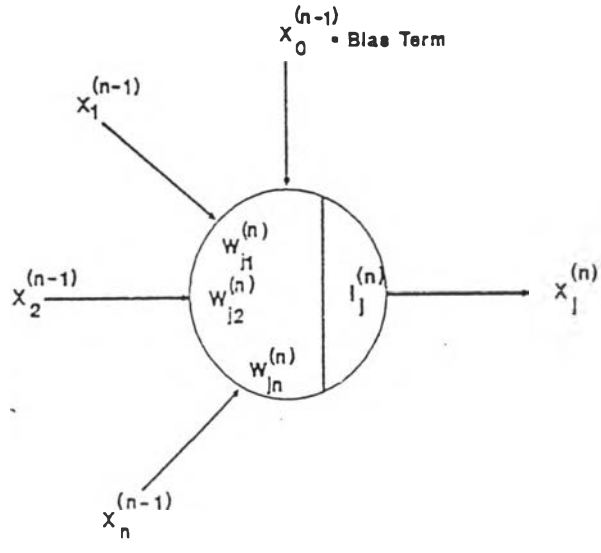


รูป 4-9 กราฟเปรียบเทียบ ฟังก์ชันซิกมอยด์ และ ฟังก์ชันขั้นบันได

4.3.2 การแพร่ย้อนกลับ (Backward-Propagation)



รูป 4-10 แสดง การแพร่ย้อนกลับ



รูป 4-11 รายละเอียดในแต่ละโนด

ในขั้นตอนของการแพร่ขยายเดินหน้า สามารถพิจารณาแต่ละโนดหรือหน่วยประมวลผลโดยละเอียดได้ดังรูป และอธิบายได้ดังนี้

ให้ X_t = ชุดข้อมูลเข้าลำดับที่ t

$$X_t = \{ x_{t,1} , x_{t,2} , \cdots , x_{t,N} \} \qquad \text{----- (4.4)}$$

Y_t = ผลลัพธ์ที่เกิดขึ้นจริงจากการใส่ชุดข้อมูลเข้า X_t

$$Y_t = \{ y_{t,1} , y_{t,2} , \cdots , y_{t,N} \} \qquad \text{----- (4.5)}$$

D_t = ผลลัพธ์ที่ต้องการจากการใส่ชุดข้อมูลเข้า X_t

$$D_t = \{ d_{t,1} , d_{t,2.} , \cdots , d_{t,N} \} \qquad \text{----- (4.6)}$$

$X^{(n)}_{t,j}$ คือ ผลลัพธ์จากโนดที่ j จากการใส่ชุดข้อมูลเข้าลำดับที่ t ในชั้นที่ n

$W^{(n)}_{j,i}$ คือ ค่าน้ำหนักจากโนดที่ i ในชั้นที่ (n-1) ไปยังโนดที่ j ในชั้นที่ n

$I^{(n)}_{t,j}$ หรือ $S^{(n)}_{t,j}$ คือ ผลรวมของโนดที่ j ในชั้นที่ n จากการใส่ชุดข้อมูลเข้าลำดับที่ t

<u>หมายเหตุ</u>

การที่ใช้สัญลักษณ์ $I^{(n)}_{t,j}$ แทน $S^{(n)}_{t,j}$ เนื่องจากค่าผลรวมที่ได้นั้นจะกลายเป็นข้อมูลเข้า (Input) ของโนดในชั้นถัดไป และต่อไปนี้จะใช้ $I^{(n)}_{t,j}$ แทน $S^{(n)}_{t,j}$ ในการพิสูจน์ในตอนต่อไป

ดังนั้น ค่าผลลัพธ์ของโนดที่ j ในชั้นที่ n จากการใส่ชุดข้อมูลเข้าลำดับที่ t สามารถเขียนได้ดังนี้

$$X^{(n)}_{t,j} = f\left[\sum_{i=1}^{N_{n-1}} W^{(n)}_{j,i} X^{(n)}_{t,i}\right] = f(S^{(n)}_{t,j}) = f(I^{(n)}_{t,j}) \quad \text{----- (4.7)}$$

ซึ่ง N_{n-1} คือ จำนวนโนดในชั้นที่ (n-1)

4.3.2.1 ค่าความผิดพลาดท้องถิ่นของการแพร่ย้อนกลับ

สมมติว่า เน็ทเวิร์ค มีค่าความผิดพลาดโดยรวม (Global Error) คือ E ซึ่งสามารถหาอนุพันธ์ได้ทุกค่าน้ำหนักในเน็ทเวิร์ค และให้ $\delta^{(n)}_{t,j}$ คือ ค่าความผิดพลาดท้องถิ่น (Local Error) ที่โนด j ในชั้นที่ n ดังนั้นค่า $\delta^{(n)}_{t,j}$ สามารถเขียนได้ดังนี้

ให้ $$\delta^{(n)}_{t,j} = -\frac{\partial E}{\partial I^{(n)}_{t,j}} \qquad \text{----- (4.8)}$$

โดยกฎลูกโซ่จะได้ $$\delta^{(n)}_{t,j} = -\frac{\partial E}{\partial X^{(n)}_{t,j}} \frac{\partial X^{(n)}_{t,j}}{\partial I^{(n)}_{t,j}} \qquad \text{----- (4.9)}$$

ดังนั้น ค่าความผิดพลาดท้องถิ่นของโนดที่ j คือผลลัพธ์ของ 2 เทอมคือ

เทอมที่ 1 $\frac{\partial E}{\partial X^{(n)}_{t,j}}$ คือ ค่าการเปลี่ยนแปลงค่า E เปรียบเทียบกับค่า $X^{(n)}_{t,j}$

เทอมที่ 2 $\frac{\partial X^{(n)}_{t,j}}{\partial I^{(n)}_{t,j}}$ คือ ค่าการเปลี่ยนแปลงค่า $X^{(n)}_{t,j}$ เปรียบเทียบกับค่า $I^{(n)}_{t,j}$

ดังนั้นจากสมการ (4.7) จะได้ $$\frac{\partial X^{(n)}_{t,j}}{\partial I^{(n)}_{t,j}} = f'_j\left(I^{(n)}_{t,j}\right) \qquad \text{----- (4.10)}$$

ซึ่งคือค่าอนุพันธ์ของ f_j สำหรับโนดที่ j

เทอมที่ 1 มี 2 กรณีคือ

<u>กรณีที่ 1</u> สมมติว่าโนดที่ j คือ โนดผลลัพธ์ของผลลัพธ์ของเน็ทเวิร์ค และให้ค่าความผิดพลาดโดยรวม E คือ ค่า Mean Square Error (MSE) ของผลลัพธ์ที่ต้องการและผลลัพธ์ที่เกิดขึ้นจริง สามารถแสดงได้ดังนี้

$$E = 1/2 \sum_{j=1}^{N_y} (d_{t,j} - y_{t,j})^2 \qquad \text{----- (4.11)}$$

N_y คือ จำนวนโนดในชั้นผลลัพธ์ ดังนั้นค่าอนุพันธ์ที่ได้เมื่อเทียบกับ $y_{t,j}$ คือ

$$\frac{\partial E}{\partial y_{t,j}} = - (d_{t,j} - y_{t,j}) \qquad \text{----- (4.12)}$$

จากสมการ (4.9) เมื่อรวมสมการ (4.10) และ (4.12) จะได้ ค่า δ สำหรับโนดที่ j ใดๆ คือ

$$\delta_{t,j} = (d_{t,j} - y_{t,j}) f_j' (I_{t,j}^{(n)}) \qquad \text{----- (4.13)}$$

<u>กรณีที่ 2</u> ถ้าโนดที่ j ไม่ใช่โนดผลลัพธ์ของเน็ทเวิร์ค ค่าความผิดพลาดท้องถิ่นสามารถแทนด้วย

$$\sum_k \frac{\partial E}{\partial I_{t,k}^{(n+1)}} \frac{\partial I_{t,k}^{(n+1)}}{\partial X_{t,j}^{(n)}} = \sum_k \frac{\partial E}{\partial I_{t,k}^{(n+1)}} \frac{\partial (\sum_i W_{k,i}^{(n+1)} X_{t,i}^{(n)})}{\partial X_{t,j}^{(n)}}$$

$$= \sum_k \frac{\partial E}{\partial I_{t,k}^{(n+1)}} W_{k,j}^{(n+1)} = -\sum_k \delta_{t,k} W_{k,j}^{(n+1)} \qquad \text{---- (4.14)}$$

ดังนั้น จากสมการ (4.9) (4.10) จะได้

$$\delta^{(n)}_{t,j} = f'_j\left(I^{(n)}_{t,j}\right)\sum_k \delta^{(n+1)}_{t,k}\, W^{(n+1)}_{k,j} \qquad \text{----- (4.15)}$$

ซึ่งเรียกว่า สมการ Generalized δ Rule (Rumelhart & McClelland ,1986)

เนื่องจาก f_j คือ ฟังก์ชันซิกมอยด์

จาก ฟังก์ชันซิกมอยด์ $f(z) = 1 / (1 + e^{-z})$ ----- (4.16)

$$f'(z) = e^{-z} / (1 + e^{-z})^2$$

$$= [1 / (1 + e^{-z})] - [1 / (1 + e^{-z})^2]$$

$$= 1 / (1 + e^{-z}) [1 - 1 / (1 + e^{-z})]$$

ดังนั้น $f'(z) = f(z) [1 - f(z)]$ ----- (4.17)

จากสมการ (4.15) สามารถเขียนได้ดังนี้

$$\delta^{(n)}_{t,j} = X^{(n)}_{t,j}\left(1 - X^{(n)}_{t,j}\right)\sum_k \left(\delta^{(n+1)}_{k}\, W^{(n+1)}_{k,j}\right) \qquad \text{----- (4.18)}$$

ดังนั้น จุดประสงค์สำคัญของการเรียนรู้แบบย้อนกลับ คือ การแพร่เดินหน้าของ X_t จากชั้นข้อมูลเข้าไปยังชั้นผลลัพธ์ จากนั้นก็ทำการคำนวณค่าความผิดพลาด โดยเปรียบเทียบค่าผลลัพธ์ที่เกิดขึ้นจริงกับผลลัพธ์ที่ต้องการ และแพร่ค่าความผิดพลาดย้อนกลับจากชั้นผลลัพธ์ย้อนกลับมายังชั้นข้อมูลเข้า

4.3.2.2 การทำค่าความผิดพลาดโดยรวมให้น้อยที่สุด

จากแนวความคิดของ ค่าความผิดพลาดท้องถิ่นที่แต่ละโนด หรือแต่ละหน่วยประมวลผล ค่าความผิดพลาดโดยรวม E สามารถทำให้ลดน้อยที่สุด โดยการปรับปรุงชุดของน้ำหนัก โดยใช้กฎของ Gradient Descent ดังนี้

$$\Delta W_{j,i}^{(n)} = -\eta \left(\frac{\partial E}{\partial W_{j,i}^{(n)}}\right) \qquad \text{----- (4.19)}$$

η อ่านว่า อีต้า (Eta) คือ ค่าสัมประสิทธิ์การเรียนรู้

จากกฎลูกโซ่จะได้ $\dfrac{\partial E}{\partial W_{j,i}^{(n)}} = \dfrac{\partial E}{\partial I_{t,j}^{(n)}} \dfrac{\partial I_{t,j}^{(n)}}{\partial W_{j,i}^{(n)}}$ ----- (4.20)

พิจารณาเทอมที่ 2

$$\frac{\partial I_{t,j}^{(n)}}{\partial W_{j,i}^{(n)}} = \frac{\partial}{\partial W_{j,i}} \left[\sum_k W_{j,k}^{(n)} X_{t,k}^{(n-1)}\right]$$

$$= X_{t,i}^{(n-1)} \qquad \text{----- (4.21)}$$

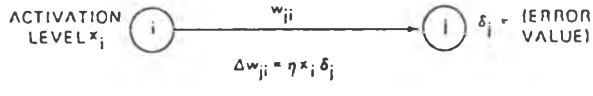
แทนค่า สมการ (4.21) ลงใน (4.20) จะได้

$$\frac{\partial E}{\partial W_{j,i}^{(n)}} = -\delta_{t,j}^{(n)} X_{t,i}^{(n-1)} \qquad \text{----- (4.22)}$$

แทนค่า สมการ (4.22) ลงใน (4.19) จะได้

$$\Delta_t W_{j,i}^{(n)} = \eta \, \delta_{t,j}^{(n)} \, X_{t,j}^{(n-1)} \qquad \text{----- (4.23)}$$

เป็นสมการการปรับปรุงค่าน้ำหนัก



รูป 4-12 แสดงการปรับปรุงค่าน้ำหนัก

4.3.2.3 ฟังก์ชันความผิดพลาดโดยรวม (The Global Error Function)

ให้ X_t คือ เวกเตอร์ข้อมูลเข้า (Input Vector)

D_t คือ เวกเตอร์ข้อมูลผลลัพธ์ที่ต้องการ (Desired Output Vector)

Y_t คือ เวกเตอร์ข้อมูลผลลัพธ์ที่เกิดขึ้นจริง (Actual Output Vector)

ค่าความผิดพลาดโดยรวมสามารถแสดงได้ดังเช่นสมการ (4.11) คือ

$$E = 1/2 \sum_{j=1}^{N_y} (d_j - y_j)^2 \qquad \text{----- (4.24)}$$

จากสมการ (4.8) $\delta_j = -\frac{\partial E}{\partial I_j}$

$$= -\frac{\partial E}{\partial y_j}\frac{\partial y_j}{\partial I_j}$$

$$= (d_j - y_j)\, f'(I_j)$$

ซึ่งมีค่าเท่ากับสมการ (4.13) คือมีค่าเท่ากับความผิดพลาดท้องถิ่น

4.3.2.4 การใช้ค่าโนดเอนเอียง

จุดประสงค์ก็เพื่อให้การเรียนรู้ของเน็ทเวิร์คเร็วขึ้น และมีความยืดหยุ่นมากขึ้น

$$y_j = f\left[\sum_{i=1}^{N_x} W_{ji} X_i + \theta_j\right] \qquad \text{------ (4.25)}$$

ถ้าเป็นสมการเชิงเส้นจะได้

$$y_j = \sum_{i=1}^{N_x} W_{ji} X_i + \theta_j \qquad \text{------ (4.26)}$$

4.3.2.5 การใช้ค่าโมเมนต้ม

นอกจากค่า η แล้ว ยังมีค่าโมเมนต้ม ใช้สัญญลักษณ์ α อ่านว่า อัลฟ่า (Alpha) ที่ช่วยให้การเรียนรู้เร็วขึ้น

ค่าโมเมนต้มนั้น คิดค้นโดย Rumelhart Hinton และ William (1986) โดยได้สร้างเทคนิคช่วยให้เวลาในการสอนเน็ทเวิร์คของการเรียนรู้แบบย้อนกลับเร็วขึ้น โดยการใส่ค่าโมเมนต้ม เข้าไป ทำให้เน็ทเวิร์คมีความคงตัวมากขึ้น และ ได้สมการดังนี้

$$\Delta W_{ji}^{(n)}(k+1) = \eta \delta_j^{(n)} X_i^{(n-1)} + \alpha \Delta W_{ji}^{(n)}(k) \qquad \text{----- (4.30)}$$

ดังนั้นจะได้

$$\Delta W_{ji}^{(n)}(k+1) = \sum_{t=1}^{I} \Delta_t W_{ij}^{(n)}(k) \qquad \text{----- (4.31)}$$

ค่า k คือ ดัชนีเวลา (Time Index) หรือ จำนวนรอบการปรับปรุงค่าน้ำหนัก

I คือ จำนวนเวกเตอร์ข้อมูลเข้า หรือ จำนวนเวกเตอร์ผลลัพธ์

4.4 <u>สรุปขั้นตอนของการเรียนรู้แบบย้อนกลับมาตรฐาน</u>

สามารถสรุปเป็นขั้นตอนได้ดังนี้

ขั้นตอนที่ 1 กำหนดค่าเริ่มต้นให้กับค่าน้ำหนักและค่าความเอนเอียง

โดยการกำหนดค่าน้ำหนักทั้งหมดและค่าความเอนเอียงให้เป็นค่าเล็กๆ โดยปกติอยู่ในช่วง [-0.1,0.1]

ขั้นตอนที่ 2 กำหนดชุดข้อมูลเข้าและผลลัพธ์ที่ต้องการ

โดยการกำหนดชุดข้อมูลเข้าและระบุผลลัพธ์ที่ต้องการ ชุดข้อมูลเข้าสามารถกำหนดขึ้นใหม่ในการสอนแต่ละรอบ หรือ เลือกจากชุดการสอน โดยจะมีการกำหนดเป็นรอบๆ จนกระทั่งได้ค่าน้ำหนักที่คงที่

ขั้นตอนที่ 3 คำนวณผลลัพธ์ที่เกิดขึ้นจริง

โดยใช้ ฟังก์ชัน ซิกมอยด์ $f(z) = 1 / (1 + e^{-z})$ ในการคำนวณผลลัพธ์

<u>ชั้นแอบแฝงที่ 1</u>

$$X_j^{(1)} = f\left[\sum_{i=1}^{N} W_{ij}^{1}\, X_i^{(1)}\, X_i^{(0)} - \theta_j^{(1)}\right] \quad ; \quad 1 <= j <= N_1$$

ชั้นแอบแฝงที่ 2

$$X_k^{(2)} = f\left[\sum_{j=1}^{N_2} W_{kj}^{(2)(1)} X_j - \theta_k^{(2)}\right] \quad ; \quad 1 <= k <= N_2$$

ชั้นแอบแฝงที่ n

$$X_k^{(n)} = f\left[\sum_{j=1}^{N_n} W_{kj}^{(n)(n-1)} X_j - \theta_k^{(n)}\right] \quad ; \quad 1 <= k <= N_n$$

ชั้นผลลัพธ์

$$Y_j = f\left[\sum_{k=1}^{N_y} W_{jk}^{(n)(n-1)} X_k - \theta_j^{(n)}\right] \quad ; \quad 1 <= j <= N_y$$

โดย N_x = จำนวนโนดของข้อมูลเข้า

N_y = จำนวนโนดผลลัพธ์

$N_1, N_2, \ldots, N_n$ = จำนวนโนดของชั้นแอบแฝงที่ $1, 2, \ldots, n$

$\theta_j^{(n)}$ = ค่าความเอนเอียง

ขั้นตอนที่ 4 ปรับปรุงค่าน้ำหนัก

ก. ใช้ขั้นตอนย้อนกลับ (Recursive Algorithm) โดยเริ่มต้นจากชั้นผลลัพธ์ย้อนกลับมาชั้นแอบแฝงที่อยู่ถัดมา และปรับปรุงค่าน้ำหนัก

โดย $W_{ji}^{(n)}(k+1) = W_{ji}^{(n)}(k) + \eta \, \delta_j^{(n)(n-1)} X_i$

ซึ่ง $W_{ji}(k)$ คือ ค่าน้ำหนักของโนดที่ i ในชั้นที่ (n-1) ไปยังโนดที่ j ในชั้นที่ n ณ เวลา k

$[.]$

X_i = ค่าผลลัพธ์ของโนด i หรือเป็นข้อมูลของโนด j

η = ค่าสัมประสิทธิ์การเรียนรู้

$\delta_j^{(n)}$ = ค่าความผิดพลาดของโนด j

โดยถ้า <u>กรณี 1</u> โนด j เป็นโนดผลลัพธ์จะได้

$$\delta_j = y_j\ (1 - y_j)\ (d_j - y_j)$$

d_j คือ ผลลัพธ์ที่ต้องการของโนด j

y_j คือ ผลลัพธ์ที่เกิดขึ้นจริงของโนด j

<u>กรณี 2</u> โนด i เป็นโนดในชั้นแอบแฝง จะได้

$$\delta_j^{(n)} = X_j^{(n)}\ (1 - X_j^{(n)}) \sum_k \delta_k^{(n)}\ W_{kj}^{(n)}$$

k คือ จำนวนโนดทั้งหมดในชั้นถัดจาก j

ข. หยุดเมื่อค่าน้ำหนักทั้งหมดมีค่าคงที่ ซึ่งหมายความว่า กระบวนการเรียนรู้ได้สิ้นสุดแล้ว

ขั้นตอนที่ 5 วนซ้ำย้อนกลับไปขั้นตอนที่ 2

ค่า η โดยปกติ มีค่าอยู่ระหว่าง 0.25-0.75 โดยถูกกำหนดโดยผู้ใช้ ถ้ากำหนดค่า η มากเกินไปจะทำให้เกิดความไม่แน่นอนขึ้นในเน็ทเวิร์ค และทำให้การเรียนรู้ของเน็ทเวิร์คไม่ดีเท่าที่ควร หรือถ้ากำหนดค่าน้อยเกินไปจะทำให้การเรียนรู้ช้ามาก นอกจากค่า η แล้ว ยังมีค่าโมเมนตัม α ที่ช่วยให้การเรียนรู้เร็วขึ้น และทำให้เน็ทเวิร์คมีความคงตัวมากขึ้น นอกจากนี้ ยังช่วยไม่ให้เน็ทเวิร์คตกอยู่ภายใต้จุดต่ำสุดท้องถิ่น (Local Minima)

$$\Delta W_{ji}(k+1) = \eta \delta_j X_i + \alpha [\Delta W_{ji}(k)] \quad \text{----- (4.32)}$$

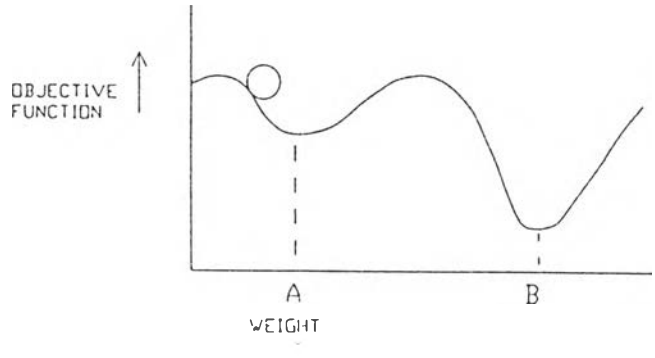
ดังนั้น $$W_{ji}(k+1) = W_{ji}(k) + \Delta W_{ji}(k+1) \quad \text{----- (4.33)}$$

ค่า α ที่เหมาะสมอยู่ในช่วง 0.00 - 1.0 (Phillip D.Wasserman,1989) และมีค่าเท่ากับ 0.9 (John Herz,Anders Krogh,Richard G. Palmer,1991)

ค่า η ที่เหมาะสมอยู่ในช่วง 0.01 - 1.0 (Phillip D.Wasserman,1989) และ 0.25 - 0.75 (Judith E.Dayhoff,1990)

ซึ่งการจะกำหนดค่าของ η และ α นั้นขึ้นกับแต่ละงานประยุกต์ Rumelhart และ McClelland กำหนดให้ตอนเริ่มการสอนเน็ทเวิร์ค มีค่า $\eta = 0.5$ และ $\alpha = 0.9$

จะเห็นว่าการกำหนดค่า η และ α นั้นควรเริ่มจากค่ามากๆ แล้วค่อยลดลงเรื่อยๆ จะทำให้เน็ทเวิร์คสามารถเข้าสู่จุดต่ำสุดโดยรวม (Global Minimum) เร็วขึ้น (แทนที่จะไปตกอยู่ที่จุดต่ำสุดท้องถิ่น) และ มีความคงตัว ดังรูป 4-13



รูป 4-13 ปัญหาของจุดต่ำสุดท้องถิ่น

4.4 <u>การสอนเน็ทเวิร์ค</u>

การสอนเน็ทเวิร์คชนิดการเรียนรู้แบบย้อนกลับ เรียกว่า การสอนแบบมีครู โดยมีการกำหนดรูปแบบข้อมูลเข้าควบคู่ กับรูปแบบเป้าหมาย โดยมีการกำหนดชุดการสอนหลายๆ รูปแบบ เพื่อให้นิวรัล เน็ทเวิร์คสามารถ เรียนรู้ นิวรัล เน็ทเวิร์คสามารถแยกแยะได้ว่า ลักษณะ สัญญาณความถี่นั้นผิดปกติจากความถี่ธรรมชาติ (Natural Frequency) หรือไม่

ในกระบวนการสอน เพื่อให้นิวรัล เน็ทเวิร์คสามารถเรียนรู้ได้ดีนั้น จะต้องใช้การวนซ้ำ การสอนหลายๆรอบ ซึ่งอาจเป็น 1,000 - 30,000 รอบ ก็ได้ เพื่อให้ค่า RMS. ลดลงต่ำกว่า 0.1 ซึ่งก็หมายความว่า นิวรัล เน็ทเวิร์คได้เกิดการเรียนรู้แล้ว ดังนั้นจำต้องใช้ ฮาร์แวร์ที่มีประสิทธิภาพสูงเข้าช่วย เช่นต้องมี ตัวช่วยประมวลคณิตศาสตร์ หรือ อาจต้องใช้เครื่องเวิร์คสเตชันประมวลผล เพื่อต้องการผลลัพธ์ที่เร็วขึ้น

แต่ถ้าเน็ทเวิร์ค ตกอยู่ภายใต้จุดต่ำสุดท้องถิ่น นั่นคือ นิวรัล เน็ดเวิร์คได้หยุดการเรียนรู้แล้ว หรือ ไม่สามารถระลึก ชุดการทดสอบได้เลย แม้ว่าจะระลึก ชุดการสอนได้

รูป 4-14 แสดงจุดต่ำสุดโดยรวม และ จุดต่ำสุดท้องถิ่น

ถ้าเน็ทเวิร์ค ตกอยู่ที่จุดต่ำสุดโดยรวม นั่นคือ ค่า RMS. ของเน็ทเวร์คได้ถึงจุดต่ำสุดแล้ว ซึ่งมันสามารถระลึก ได้ทั้งชุดการสอน และ ชุดการทดสอบ ได้ในระดับที่พอใจ

มีวิธีการแก้ปัญหาไม่ให้เน็ทเวร์ค ตกอยู่ภายใต้จุดต่ำสุดท้องถิ่น คือ

4.4.1 การปรับค่า η และ α โดยเริ่มต้นจากค่ามากๆ ซึ่งน้อยกว่า 1 แล้วค่อยๆ ลดลงเรื่อยๆ ตามจำนวนรอบที่ใช้สอนเน็คเวร์ค

4.4.2 การเพิ่มจำนวนโนดแอบแฝง

4.4.3 การเปลี่ยนฟังก์ชันการแปลงค่า รูปแบบหรือโครงสร้าง (Model)

4.4.4 การเปลี่ยนชุดการสอนเสียใหม่ เนื่องจากชุดการสอนเดิมไม่ได้เป็นตัวแทนที่ดีของประเภทอาการนั้นๆ

4.5 การประเมินผลการสอนนิวรัล เน็ทเวิร์ค

การสอนนิวรัล เน็ทเวิร์ค จะสำเร็จหรือไม่ เราวัดจากค่า RMS. ดังสมการ

$$RMS = \sqrt{\frac{\sum_p \sum_j (t_{jp} - x_{jp})^2}{n_p \, n_o}} \qquad \text{---------- (4.34)}$$

โดย n_p = จำนวนของรูปแบบในชุดการสอน

n_o = จำนวนของหน่วยในชั้นผลลัพธ์

t_{jp} = ค่าเป้าหมายของหน่วยที่ j หลังจากที่เสนอรูปแบบ p

x_{jp} = ค่าผลลัพธ์ที่ออกมาโดยหน่วยที่ j หลังจากที่เสนอรูปแบบ p

โดยปกติ ถ้าค่า RMS. ที่ได้ < 0.1 แสดงว่านิวรัล เน็ทเวิร์ค ได้เกิดการเรียนรู้แล้ว

4.6 <u>ข้อดีและข้อจำกัดของการเรียนรู้แบบย้อนกลับ</u> (Judith E. Dayhoff,1990)

4.6.1 ข้อดี

คือ มีความสามารถในการจดจำรูปแบบของปัญหา(Pattern-Mapping) ซึ่งการเรียนรู้แบบย้อนกลับ สามารถที่จะเรียนรู้ความสัมพันธ์ของรูปแบบได้มากมาย โดยการเรียนรู้แบบย้อนกลับ ต้องการตัวอย่างรูปแบบที่จะเรียนรู้ ความยืดหยุ่นของการเรียนรู้แบบย้อนกลับ อยู่ที่การมีความหลากหลายในการออกแบบทางเลือกต่างๆ เช่น จำนวนชั้น เส้นเชื่อมโยง จำนวนของโนด ค่า ส.ป.ส.การเรียนรู้ ค่าโมเมนตัม ที่เรากำหนดขึ้น และการแทนรูปแบบของข้อมูล

ความยืดหยุ่นดังกล่าว ทำให้การเรียนรู้แบบย้อนกลับ สามารถแก้ปัญหางานประยุกต์ได้อย่างมากมาย

4.6.2 ข้อจำกัด

คือ การใช้เวลาอย่างมากเพื่อให้การสอนนิวรัล เน็ทเวิร์ค สามารถเรียนรู้ได้ หรือที่เรียกว่า Convergence Time ซึ่งบางครั้งอาจต้องการการวนซ้ำถึง 100-1,000 รอบ สำหรับการเรียนรู้การ แก้ปัญหานั้นๆ ทำให้บางครั้งต้องใช้เวลา 1 วัน หรือมากกว่านั้นในการสอนนิวรัล เน็ทเวิร์ค

อุปสรรคอีกข้อหนึ่ง คือ การต้องใช้ชุดการสอนอย่างมากมาย ซึ่งบางครั้งอาจต้องใช้ถึง 1,000 ตัวอย่าง ด้วยเหตุผลดังกล่าว เพื่อที่จะลดเวลาการเรียนรู้ ทำให้ต้องใช้ฮาร์ดแวร์ ที่มีกำลังและประสิทธิภาพสูง กล่าวคือ จำเป็นต้องใช้ บอร์ดตัวเร่งชนิดพิเศษ เครื่องประมวลแบบคู่ขนาน

นอกจากความสามารถทางด้านฮาร์ดแวร์แล้ว อาจมีการเปลี่ยนชุดการสอนใหม่ การปรับเปลี่ยนค่า ส.ป.ส.การเรียนรู้ ก็มีส่วนช่วยให้การเรียนรู้เร็วขึ้น โดยปกติค่า η จะเริ่มจากค่ามากและลดค่าลงเรื่อยๆเพื่อให้การเรียนรู้เป็นไปอย่างมีประสิทธิภาพ ค่า η ที่เหมาะสมจะมีค่าระหว่าง 0.25-0.75