บทที่ 2

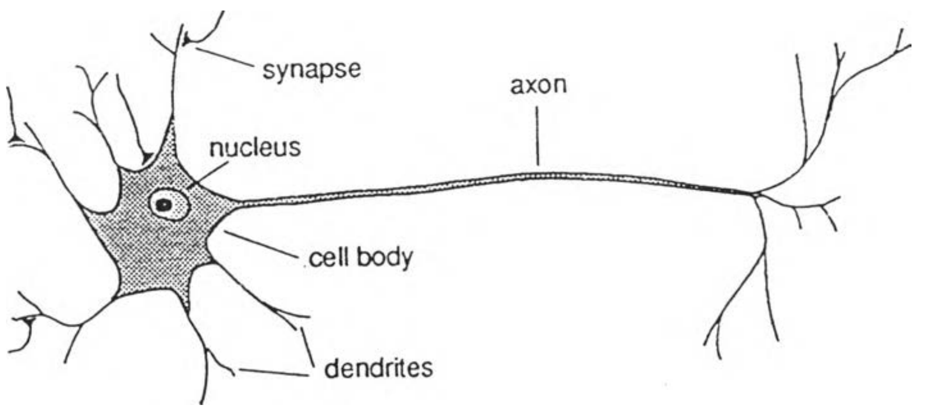
ทฤษฎีเกี่ยวกับข่ายงานนิวรัล

2.1 ชีววิทยาของนิวรอนและแบบจำลองประดิษฐ์ (biological neurons and their artificial models)

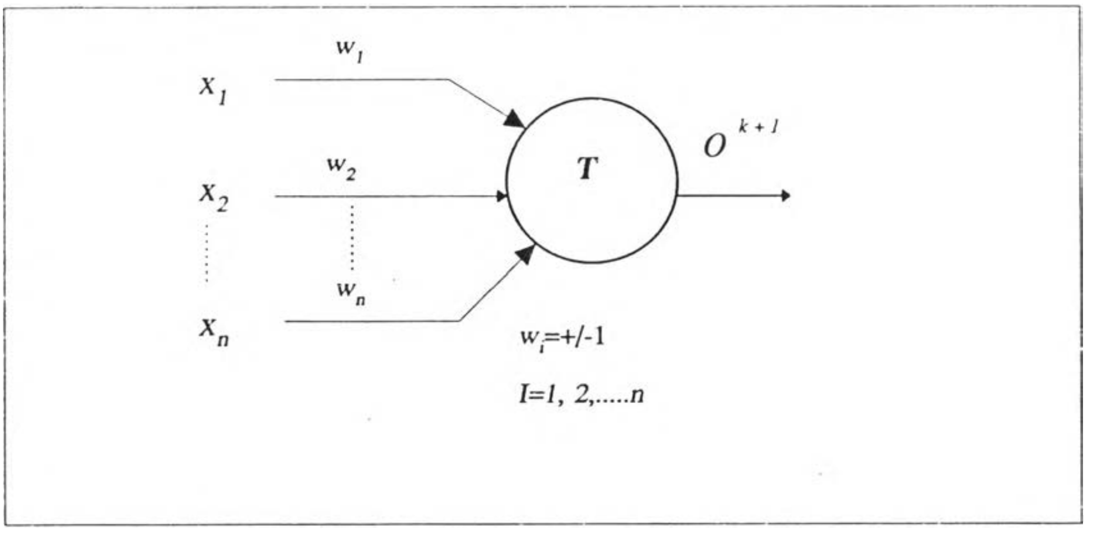
ระบบประสาทของมนุษย์ (human neurons system) ประกอบด้วยกลุ่มของนิวรอน (neurons) ประมาณ 10^{11} เซล โดยสามารถติดต่อสื่อสารกับเซลหน่วยอื่นๆ ด้วยแอกซอน (axon) และซินแนพส์ (synapses) ซึ่งมีความหนาแน่นของซินแนพส์ประมาณ 10^4 หน่วยต่อนิวรอน จากข้อสมมุติฐานเกี่ยวกับแบบจำลองของระบบประสาทของมนุษย์นั้น ติดต่อกันระหว่างนิวรอนโดยอาศัยการกระตุ้นทางไฟฟ้า (electrical impulse) แต่การทำงานของนิวรอนนั้นเกิดขึ้นเนื่องจากกระบวนการทางชีวเคมี แม้ว่าช่วงเวลาในการประมวลผลของนิวรอน (neuron switch time) จะมีค่าน้อยกว่าหนึ่งในพันวินาที (milliseconds) ซึ่งน้อยกว่า ช่วงเวลาที่กระแสไฟฟ้า วิ่งผ่านคอมพิวเตอร์ถึงหนึ่งล้านเท่า อย่างไรก็ตาม นิวรอนมีความสามารถ เชื่อมโยงถึงกันได้ไวกว่าซูเปอร์คอมพิวเตอร์ ในปัจจุบันถึงหนึ่งพันเท่า ในรูปที่ 2.1 แสดงถึงส่วนประกอบของนิวรอนในทางชีววิทยา อันได้แก่ 1) ตัวเซลนิวรอน (cell body) 2) เดนไดรท์ (dendrite) ที่แตกแขนงออกจากตัวเซลเพื่อรับข้อมูล และ 3) แอกซอน เป็นตัวนำสัญญานที่ออกจากนิวรอนไปยังเดนไดรท์ของนิวรอนอื่นๆ สำหรับซินแนพส์นั้นเป็นรอยต่อระหว่างแอกซอน กับ เดนไดรท์ การทำงาน

ร่วมกันของนิวรอนยังคงเป็นสิ่งที่ลึกลับและซับซ้อนเป็นอย่างมาก โดยทั่วไปแล้วสัญญานจะถูกส่งออกไปยังนิวรอนหน่วยอื่นโดยแอกซอน ซึ่งแอกซอนจะนำสัญญานข่าวสารผ่านไปตามลำดับของกระแสคลื่นกระแสไฟฟ้า อันขึ้นกับค่าแรงดันต่างศักย์ของนิวรอนและเนื้อเยื่อของนิวรอน (membrane) ที่สร้างแรงดันต่างศักย์กระจาย (propagate) ผ่านแอกซอนไปจนถึงรอยต่อซินแนพส์ (synaptic junction) จากแนวคิดพื้นฐานดังกล่าวนี้ McCulloch และ Pitts จึงได้เสนอโครงสร้างข่ายงานนิวรัลโดยอยู่ในรูปของแบบจำลองการคำนวณ ในปี ค.ศ. 1943 โดยการสร้างตรรกการเทร็สโฮลด์ (threshold logic) อย่างง่ายๆ ได้ ดังแสดงในรูปที่ 2.2 แต่เนื่องจากแบบจำลองของ McCulloch-Pitts นั้นมีค่าจำกัดของค่าอินพุท และเอาท์พุท เฉพาะ 0 , 1 เท่านั้น และค่าน้ำหนัก (weight) ในแบบจำลองนั้นกำหนดให้มีค่าคงที่ (fixed) และไม่มีการติดต่อ (interaction) ซึ่งกันระหว่างแต่ละนิวรอน สัญญานออกจากนิวรอนนั้นถูกกำหนดด้วยเงื่อนไขดังนี้

$$o^{k+1} = \begin{cases} 1 & \text{if } \sum_{i=1}^{n} w_i x_{ik} \geq T \\ 0 & \text{if } \sum_{i=1}^{n} w_i x_{ik} < T \end{cases} \tag{2.1}$$



รูปที่ 2.1 โครงสร้างของเซลประสาท (neuron)



รูปที่ 2.2 แบบจำลองการทำงานของนิวรอนโดย McCulloch-Pitts

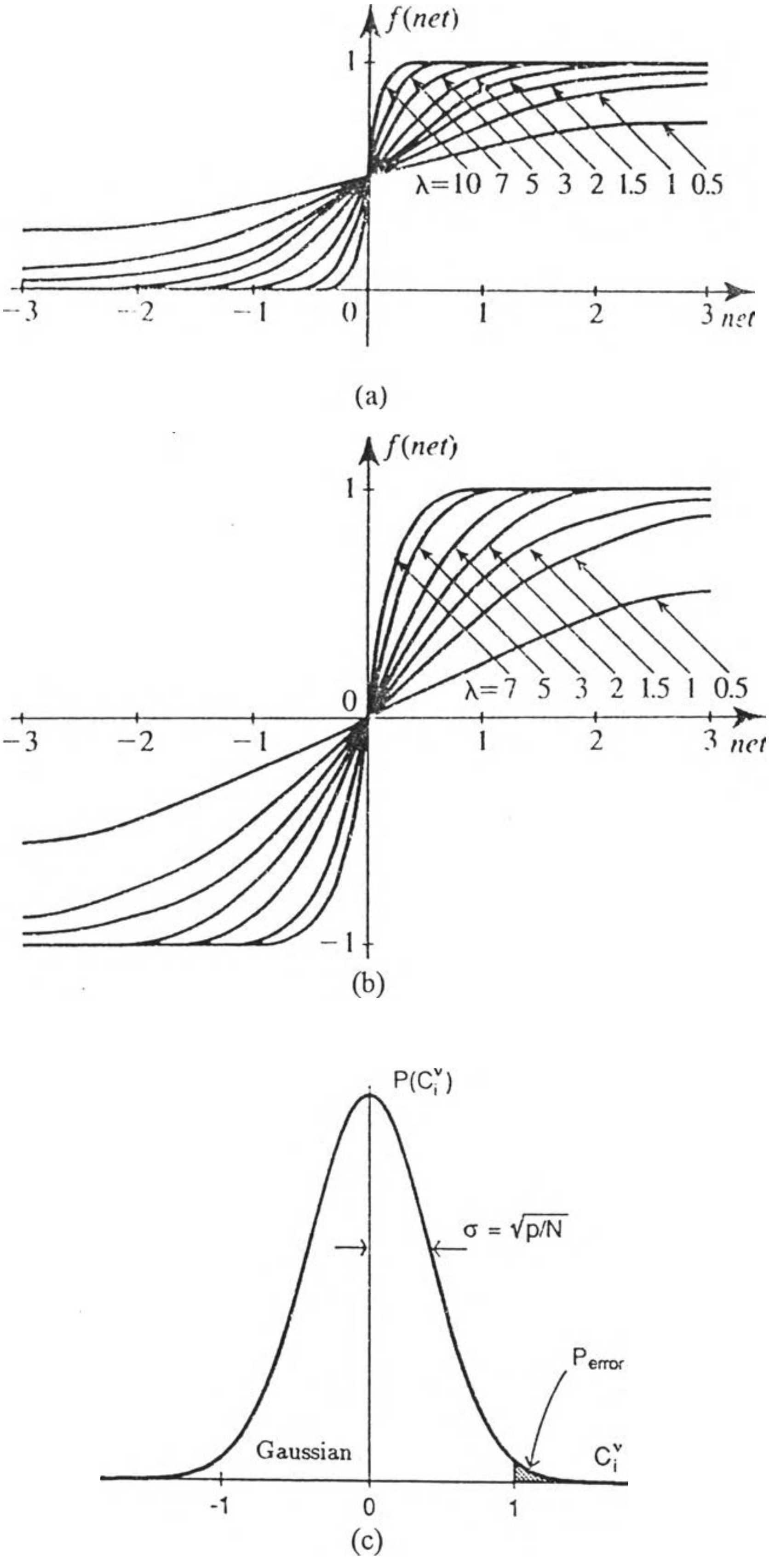
2.2 แบบจำลองของนิวรัลในข่ายงานนิวรัลในปัจจุบัน

แบบจำลองส่วนใหญ่ของนิวรอนในปัจจุบัน ประกอบด้วย หน่วย (processing element , node) กับส่วน เชื่อมโยง (synaptic connection) ระหว่างอินพุทกับ เอาท์พุท ซึ่งเรียกว่า น้ำหนัก (weights) โดยสัญญาน ที่ออกจากเอาท์พุท ของนิวรอน จะถูกแปลงค่าด้วย ฟังก์ชันกระตุ้น (activation function) ดังความสัมพันธ์ต่อไปนี้

$$o = f(w^T x) = f(\sum_{i=1} w_i x_i) \qquad \text{โดยที่} \quad net \stackrel{\Delta}{=} w^t x \qquad (2.2)$$

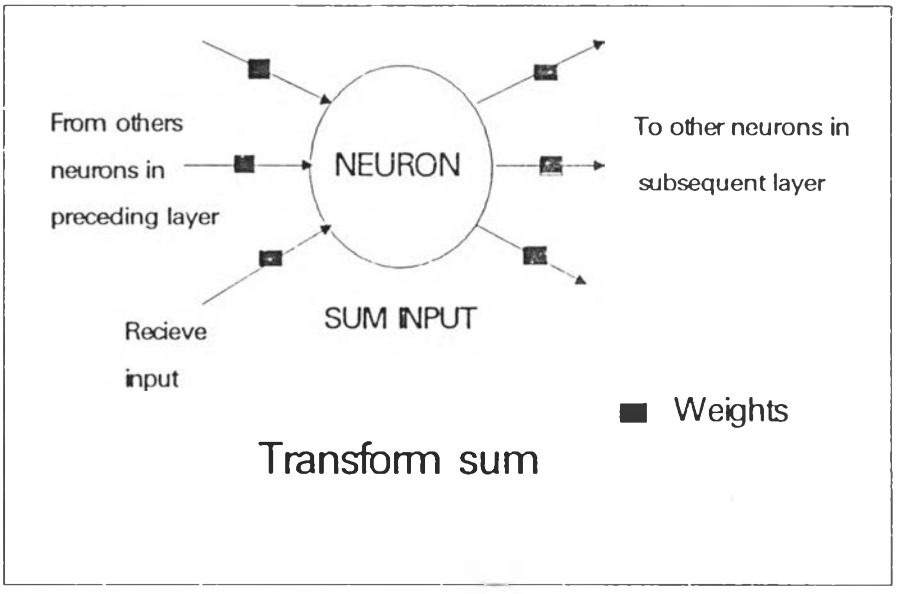
โดยที่ค่าฟังก์ชันกระตุ้น นั้นนิยมใช้ฟังก์ชันซิกมอยด์ (sigmoid) มีค่าอยู่ระหว่าง 0 - 1 หรือ ฟังก์ชัน ไบโพลาร์ (bipolar) มีค่าอยู่ระหว่าง - 1 ถึง 1 และ ฟังก์ชันเกาส์เซียน (guassian function) ซึ่งจะได้อธิบายรายละเอียดในการเลือกใช้ต่อไป ในรูปที่ 2.3 แสดงรูปร่างของฟังก์ชันกระตุ้น สำหรับรูปที่ 2.4 แสดงการทำงานของนิวรอน หน้าที่ของนิวรอนแต่ละหน่วยมีดังนี้

1. รับสัญญาณจากนิวรอนหน่วยอื่นๆ
2. รวมสัญญาณจากนิวรอนหน่วยอื่นเข้าด้วยกัน
3. แปลงสัญญาณโดยใช้ฟังก์ชันกระตุ้น ดังแสดงในรูปที่ 2..3
4. ส่งผลลัพธ์ไปยังนิวรอนหน่วยอื่นๆ



รูปที่ 2.3 ฟังก์ชันกระตุ้นที่ใช้เป็นฟังก์ชันการแปลง (transfer function)

(a) ฟังก์ชันซิกมอยด์ (b) ฟังก์ชันไบโพลาร์ (c) ฟังก์ชันเกาส์เสียน



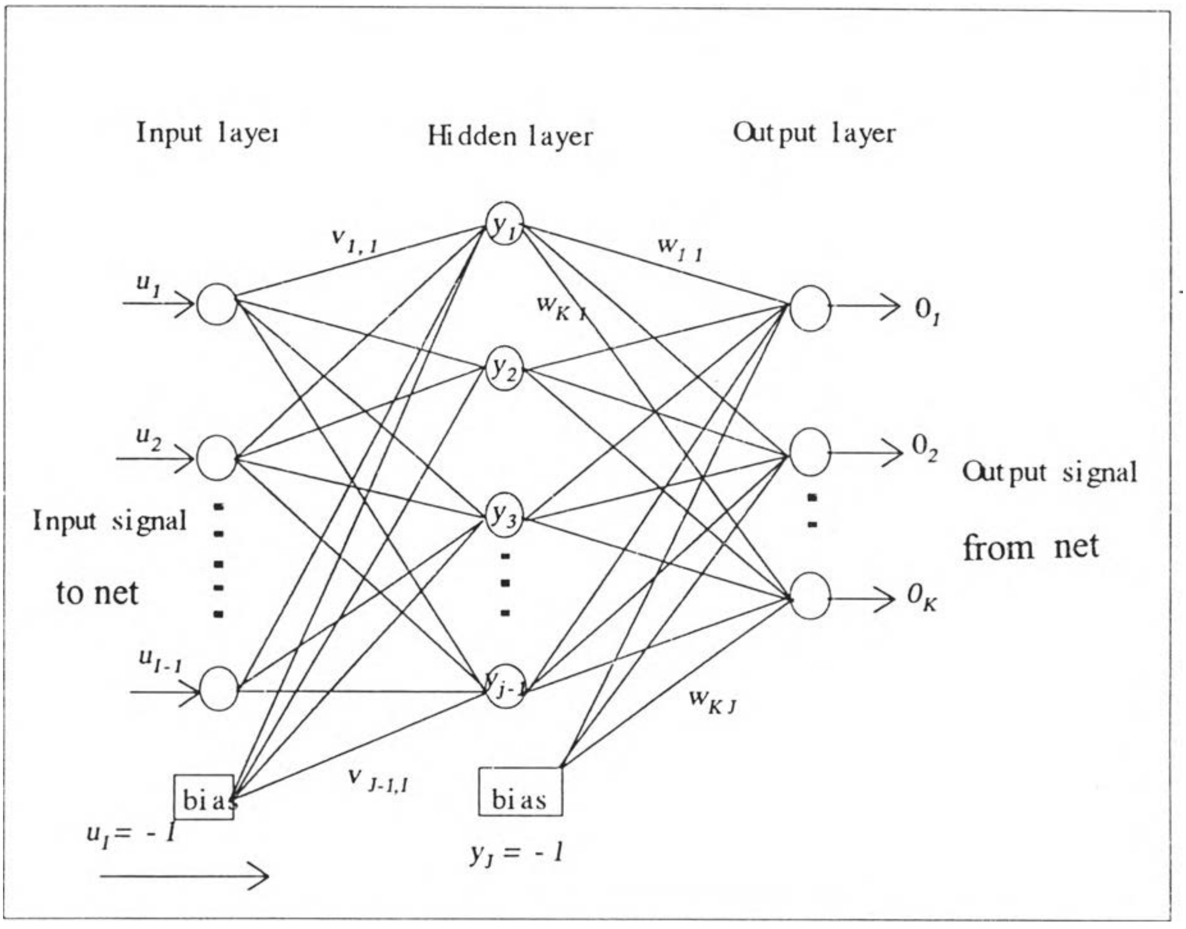
รูปที่ 2.4 หน้าที่และการทำงานของนิวรอน

สัญญาณที่เชื่อมโยงระหว่างนิวรอน คือ ค่าน้ำหนัก ซึ่งจะถูกปรับเปลี่ยนค่า strength ตลอดเวลา ในระหว่างการเรียนรู้ เพื่อสร้างแบบจำลองภายใน ขึ้นมาให้ใกล้เคียงกับระบบ เมื่อค่าผลลัพธ์ที่ออกจากระบบ (y^p) มีค่าใกล้เคียงกับ ค่าผลลัพธ์จากข่ายงาน (y^m) ถือว่าแบบจำลองที่ได้สามารถใช้แทนแบบจำลองของกระบวนการได้ โครงสร้างของข่ายงานนิวรัลที่แสดง ดังรูปที่ 2.5 ประกอบด้วย หน่วยย่อยๆ ที่เรียกว่า นิวรอน (neuron) โครงสร้างของข่ายงานนิวรัลจะเป็นชั้น (layer) 3 ชั้น เพื่อใช้ในการรับและส่งสัญญาณ คือ

1. ชั้นอินพุท (input layer) : รับข้อมูลที่ใช้ในการฝึกข่ายงานนิวรัล

2. ชั้นภายใน (hidden layer) : รวมสัญญาณจาก ชั้นรับข้อมูลไปแปลงค่า (transfrom) โดย ใช้ฟังก์ชันกระตุ้น เพื่อใช้สร้างแบบจำลองภายในของระบบ ชั้นภายในสามารถ มีได้มากกว่า 1 ชั้น

3. ชั้นเอาท์พุท (output layer) : รวมสัญญาณจากชั้นภายในไปแปลงค่าเป็นผลลัพธ์ออกจากข่ายงานนิวรัล



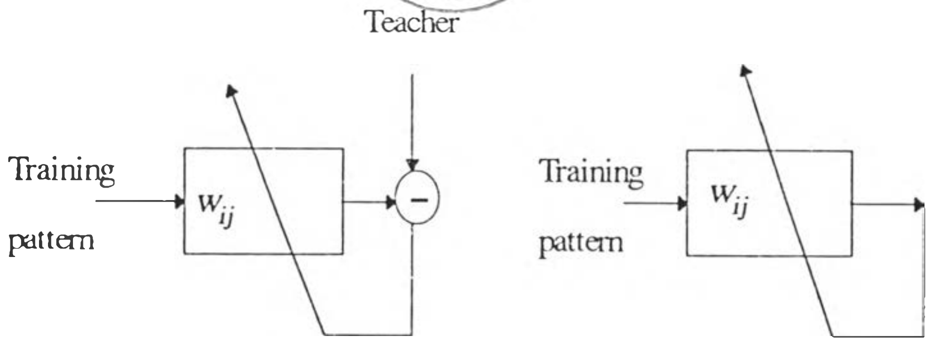
รูปที่ 2.5 โครงสร้างของข่ายงานนิวรัล

2.3 การจำแนกข่ายงานนิวรัล (Taxonomy of neural networks)

ในการออกแบบการประยุกต์ใช้ของข่ายงานนิวรัล ขึ้นอยู่กับ การเลือก ฟังก์ชันพลังงาน (energy function หรือ error function) กลไกการฝึกข่ายงานนิวรัล หลายวิธีนิยมใช้การลดค่าฟังก์ชันพลังงานให้ต่ำที่สุด ปัจจัยที่มีผลต่อการออกแบบข่ายงานนิวรัล ขึ้นกับคุณลักษณะ (characterized) ดังนี้

2.3.1 แบบจำลองที่มีการชี้นำ (Supervised model)

จุดหลักของการพัฒนาแบบจำลองข่ายงานนิวรัล ในการวิจัยครั้งนี้นั้น มุ่งเน้นในส่วนของข่ายงานที่มีการเรียนรู้แบบทมีการชี้นำ ซึ่งชุดของข้อมูล ประกอบด้วย คู่ของค่าอินพุท-เอาท์พุท ที่สอดคล้องกัน นั่นคือ ข้อได้เปรียบของการเรียนรู้ด้วยวิธีนี้ เพราะมีเป้าหมาย (target) ที่ต้องการเป็นผู้สอน (teacher) โดยค่าที่แตกต่างระหว่าง ค่าเอาท์พุทที่ออกจากข่ายงาน กับ ค่าเป้าหมาย ที่ต้องการจะเป็นข้อมูลที่ใช้ในการเรียนรู้ ในขณะที่ การเรียนรู้แบบไม่มีการชี้นำ (unsupervised) นั้นชุดข้อมูลจะมีเพียงชุดข้อมูลอินพุทการฝึกเท่านั้น โดยการเรียนรู้จะเป็นการปรับตัวบนพื้นฐานของประสบการณ์เพื่อจัดกลุ่ม (cluster) ให้เข้ากับชุดข้อมูลที่ใช้เรียนรู้ที่ผ่านมา ในทางปฏิบัตินั้นถือว่าการเรียนรู้แบบชี้นำนั้น สอดคล้องกับ วิธีการออพติไมเซชันแบบสโตเชสติคเกรเดียนท์เดสเซ็นท์ (stochastic gradient descent optimization)



(a) supervised learning **(b) unsupervised learning**

รูปที่ 2.6 การเรียนรู้ของข่ายงานนิวรัลในการปรับค่าน้ำหนักซินแนพส์ (synaptic weights)

2.3.2 ฟังก์ชันมูลฐาน (basis functions)

สำหรับการศึกษาเชิงวิเคราะห์นั้น ข่ายงานเชื่อมโยง (connection networks) ก็คือ การแทนการ แมพพิงก์ (mapping) เชิงคณิตศาสตร์ ด้วยฟังก์ชันมูลฐาน, $u(w,x)$ นั่นเอง เมื่อ w คือ เมตริกน้ำหนัก ส่วน x คือ เวคเตอร์อินพุท ฟังก์ชันมูลฐานมีอยู่ 2 รูปแบบ คือ

ก. ฟังก์ชันมูลฐานเชิงเส้น (linear basis function, LBF)

เป็นฟังก์ชันชนิดไฮเปอร์แพลน (hyperplane) โดยเป็นฟังก์ชันมูลฐานอันดับที่ 1 (เชิงเส้น) ค่า *net* ที่ได้เป็นผลรวมเชิงเส้นของค่าอินพุท ดังสมการที่ 2.3

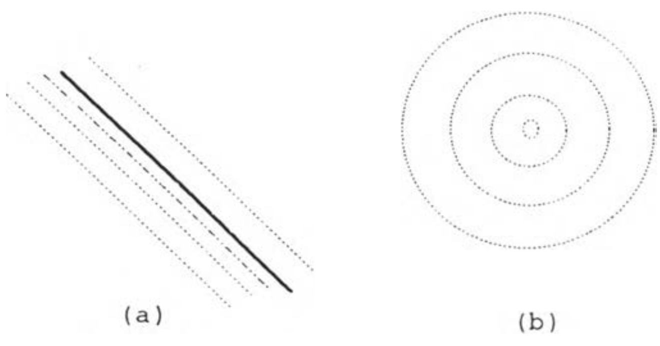
$$u_i(w,x) = \sum_{j=1}^{n} w_{ij} x_j \qquad (2.3)$$

ข. ฟังก์ชันมูลฐานเชิงรัศมี (radial basis function , RBF)

เป็นฟังก์ชันชนิดไฮเปอร์สเฟียร์ (hypersphere) โดยเป็นฟังก์ชันมูลฐานอันดับที่ 2 (ไม่เชิงเส้น) ค่า *net* ที่ได้จะแทนระยะทางไปยังชุดข้อมูลอ้างอิง (reference pattern)

$$u_i(w,x) = \sqrt{\sum_{j=1}^{n}(x_i - w_{ij})^2} \tag{2.4}$$

นอกจากนี้ฟังก์ชันอันดับที่ 2 ยังสามารถขยายไปเป็นฟังก์ชันมูลฐานอีลิปติค (elliptic basis function) ได้อีกด้วย



รูปที่ 2.7 ฟังก์ชันมูลฐาน (a) linear (hyperplane) (b) radial (hypersphere)

2.3.3 ฟังก์ชันกระตุ้น (activation function)

ค่า *net* ที่อยู่ในรูปของฟังก์ชันมูลฐาน, *u(w,x)* จะถูกแปลงค่าโดยใช้ฟังก์ชันกระตุ้นไม่เชิงเส้น (nonlinear activation function) ฟังก์ชันที่ส่วนใหญ่นิยมใช้ทั่วไป คือฟังก์ชันขั้นบันได, แรมป์ (ramp), ซิกมอยด์ (sigmoid), ไบโพลาร์ (Bipolar) และ ฟังก์ชันเกาส์เสียน (guassian function) สมการคณิตศาสตร์ของฟังก์ชันกระตุ้น มีดังนี้

ก. <u>ฟังก์ชันซิกมอยด์</u>

$$f(net) = \frac{1}{1 + e^{-net/\lambda}} \tag{2.5}$$

ข. ฟังก์ชันเกาส์เสียน

$$f(net) = ce^{-net^2/\lambda^2} \tag{2.6}$$

ค. ฟังก์ชันไบโพลาร์

$$f(net) = \frac{2}{1+exp(-\lambda net)} - 1 \tag{2.7}$$

2.4 โครงสร้างของข่ายงานนิวรัล (Neural networks structures)

โครงสร้างหลักของข่ายงานนิวรัล ประกอบด้วย โครงสร้างเชื่อมโยงระหว่างนิวรอน, ขนาดของข่ายงาน, กลวิธีในการฝึกข่ายงานที่มีโครงสร้างซูเปอร์เน็ต (supernet, All-Class-in-One-Network, ACON) กับ โครงสร้างซับเน็ต (subnet, One-Class-in-One-Network, OCON) และ โครงสร้างเชื่อมโยงของน้ำหนักระหว่างชั้น (interlayer) กับ ภายในชั้นเดียวกัน (intralayer) พฤติกรรมของข่ายงานจึงขึ้นอยู่กับการกระทำต่อกันระหว่างนิวรอน กับ น้ำหนัก ชั้นของนิวรอนมีอยู่ 3 ชนิด คือ อินพุท, ฮิดเดน และ เอาท์พุท การติดต่อสื่อสารกันระหว่างชั้นของนิวรอนผ่าน น้ำหนักเชื่อมโยงซึ่งนั้นมีอยู่ 4 ชนิด ตามแสดงในรูปที่ 2.8

2.4.1 การเชื่อมโยงป้อนไปข้างหน้า (feedforward connection)

ข้อมูลของนิวรอนจากชั้นที่ต่ำกว่าจะกระจายไปยังชั้นที่เหนือกว่า

2.4.2 การเชื่อมโยงแบบป้อนกลับ (feedback connection)

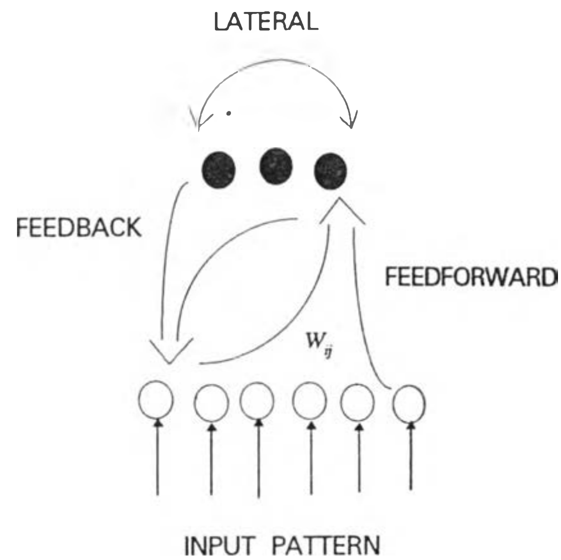
เป็นการนำข้อมูลจากชั้นที่อยู่เหนือกว่า กลับลงมายังชั้นที่อยู่ต่ำกว่า

2.4.3 การเชื่อมโยงภายในชั้น (lateral connections)

เป็นการนำข้อมูลภายในชั้นเดียวกัน กระจายกลับมาใช้ในชั้นเดิม

2.4.4 การเชื่อมโยงแบบหน่วงเวลา (time-delayed connections)

เป็นการนำข้อมูลมาหน่วงเวลาแล้วนำเข้าไปรวมกับ การเชื่อมโยงของข้อมูล เพื่อให้ได้แบบจำลองเป็น temporal dynamic ซึ่งนำมาประยุกต์ใช้กับ temporal pattern recognition



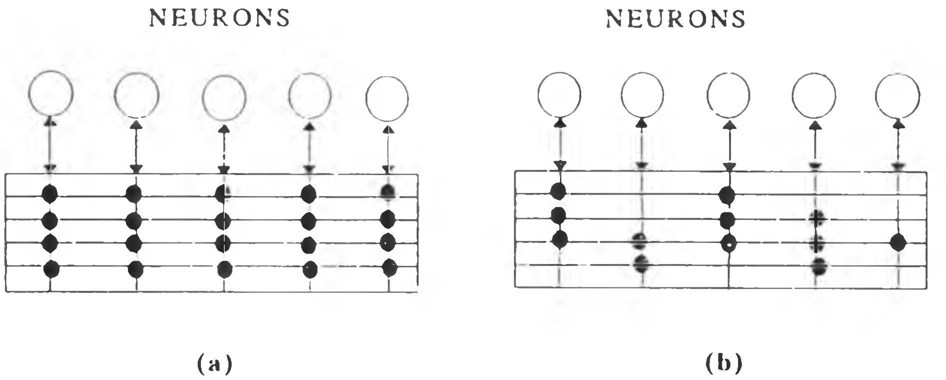
รูปที่ 2.8 โครงสร้างพื้นฐานของข่ายงานนิวรัลที่แสดงการเชื่อมโยงของน้ำหนัก

นอกจากนี้ การเชื่อมโยงระหว่างนิวรอน อาจเชื่อมโยงระหว่างกันแบบทั้งหมด (fully) หรือ เฉพาะบริเวณ (locally) ก็ได้ ดังแสดงในรูปที่ 2.9

2.5 กลวิธีในการฝึกข่ายงานนิวรัลแบบร่วมกัน และ แบบแต่ละส่วน (mutual and individual training strategies)

2.5.1 กลวิธีในการฝึกข่ายงานนิวรัลแบบร่วมกัน (mutual training)

กลวิธีในการฝึกซึ่งใช้การอ้างอิงแบบข้าม (cross-reference) ระหว่าง ค่าเอาท์พุท ทั้งหมด



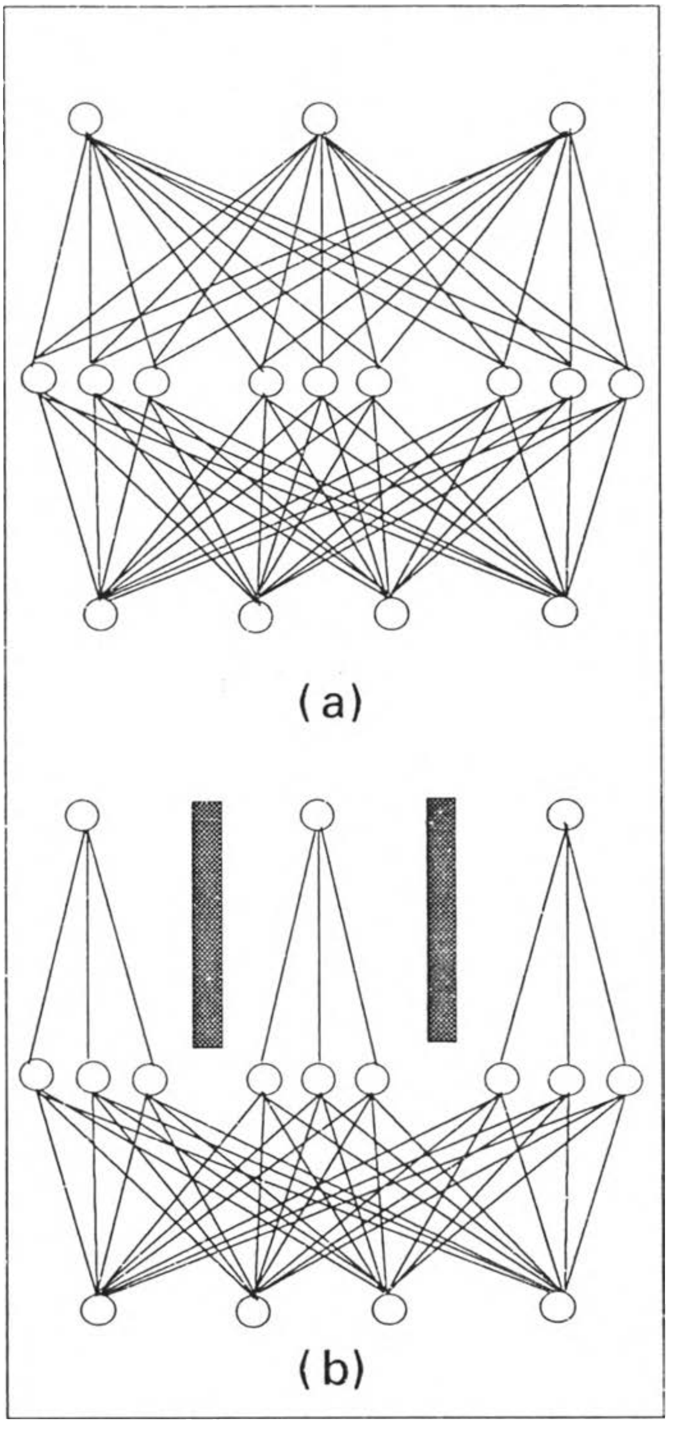
รูปที่ 2.9 การเชื่อมโยงของนิวรอนในข่ายงานนิวรัลป้อนกลับชั้นเดียว

(a) การเชื่อมโยงทั้งหมด (b) การเชื่อมโยงบางส่วน

สามารถนำมาใช้เพื่อช่วยในการเรียนรู้โดยที่ น้ำหนักทั้งหมดจะมีอิทธิพล มาจากค่าเอาท์พุท ทั้งหมด แต่ข้อเสียก็คือ จะต้องใช้เวลานาน (จำนวนรอบมาก) เพื่อให้ถึงจุดหมายร่วมกัน (mutual agreement) ระหว่าง node ทั้งหมดโครงสร้างที่ใช้ในการฝึกร่วมกันมี 2 แบบ คือ เชื่อมโยงทุกหน่วยของเอาท์พุท ในหนึ่งข่ายงาน (ACON) กับ เชื่อมโยง หนึ่งหน่วยเอาท์พุท (OCON) โดยให้ เอาท์พุทนิวรอน 1 หน่วย แทน 1 คลาส

2.5.2 กลวิธีในการฝึกข่ายงานนิวรัลแบบแต่ละส่วน (individual training)

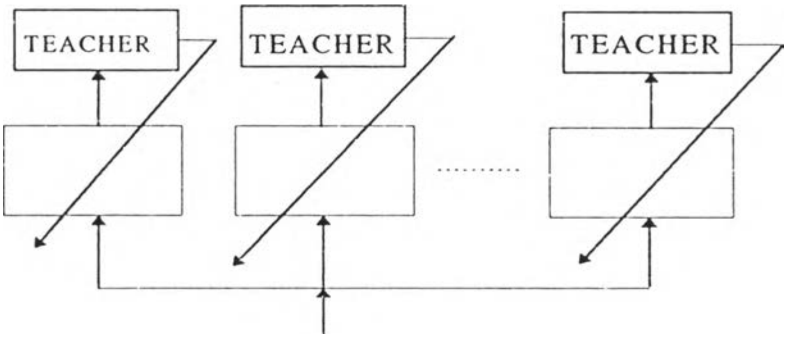
ข้อมูลอ้างอิงข้ามระหว่างค่าเอาท์พุทไม่มีอิทธิพลต่อการเรียนรู้ นั่นคือ การฝึกของข่ายงานย่อย เป็นอิสระต่อกัน ซึ่งมีข้อดี ในแง่ความง่ายในการฝึกด้วยตัวมันเอง แต่จะมีข้อจำกัดในความสามารถที่จะจัดการกับระบบที่ซับซ้อนได้ ตัวอย่างกลวิธีการฝึกแบบนี้เช่น การฝึกแบบดิสไครมิเมทิพ (discriminative training) ดังในรูปที่ 2.10 กลวิธีนี้จะใช้ชุดข้อมูลการฝึกทั้งหมด (ทั้งข้อมูลค่าบวก และ ค่าลบ) เพื่อฝึกข่ายงานย่อย



รูปที่ 2.10 กลวิธีในการฝึกข่ายงาน (Training strategies)

(a) โครงสร้างซูเปอร์เน็ต (supernet), ACON

(b) โครงสร้างที่มีการแบ่ง ซูเปอร์เน็ต เป็น ซับเน็ต (subnet), OCON



รูปที่ 2.11 การฝึกข่ายงานย่อยที่เป็นอิสระต่อกัน

2.6 ข่ายงานเชิงเส้นบนพื้นฐานการประมาณค่า (Approximation Base Formulation)

ข่ายงานเชิงเส้นมีความง่ายและสะดวกต่อการวิเคราะห์ แต่ข่ายงานไม่เชิงเส้นจะมีประโยชน์ในเชิงประยุกต์ปฏิบัติมากกว่า ชุดข้อมูลที่ใช้ฝึกประกอบด้วยค่า อินพุท/เอาท์พุท เป็นคู่ที่สอดคล้องกัน

$$[u,o] = \{[u_1,o_1],[u_2,o_2],\ldots\ldots,[u_n,o_n]\} \qquad (2.8)$$

เมื่อ n เป็นจำนวนคู่ของชุดข้อมูลที่ใช้ในการฝึก วัตถุประสงค์ในการฝึกข่ายงานก็เพื่อหาค่าน้ำหนักที่เหมาะสมเพื่อทำให้เกิดค่าความแตกต่างของค่าเป้าหมาย กับ ค่าที่ออกจากข่ายงาน โดยมีข้อกำหนดที่นิยม คือ ให้ค่ารากเฉลี่ยกำลังสองของความแตกต่างระหว่างผลลัพธ์ของระบบ กับข่ายงาน (root mean sqaure, RMS) มีค่าต่ำที่สุด ฟังก์ชันของรากเฉลี่ยกำลังสองของความแตกต่างได้แก่ $E = (u,w) = [t\text{-}f(u,w)]^2$ และใช้เป็นฟังก์ชันพลังงานของระบบ แบบจำลองของฟังก์ชัน คือ ฟังก์ชันของอินพุทและ น้ำหนัก $y = f(x,w)$ เวคเตอร์ของน้ำหนัก สามารถปรับค่า เพื่อให้ ฟังก์ชันพลังงาน มีค่าต่ำลงให้

เหลือน้อยที่สุดไปในทิศทางลดลงของเกรเดียนท์ (gradient-descent)

$$\Delta w \ \alpha \ -\frac{\partial E(u,w)}{\partial w} \ = \ (t-\phi(u,w))\frac{\partial \phi(u,w)}{\partial w} \tag{2.9}$$

2.6.1 ข่ายงานเปอร์เซปตรอนเชิงเส้น (linear perceptron networks)

ข่ายงานเปอร์เซปตรอนเป็นการบุกเบิกของแบบจำลองข่ายงานนิวรัลแบบตัดสิน (decision- based neural model) เสนอโดย F. Rosenblatt ในปี ค.ศ. 1958 ซึ่งออกแบบเพื่อใช้ในการแยกข้อมูล 2 คลาส (classes) ออกจากกัน โดยใช้ขอบเขตการตัดสินเชิงเส้น (linear decision) ดังแสดงในรูปที่ 2.12 และ 2.13 ในการเรียนรู้อาศัยสัญญานการเรียนรู้จากค่าความแตกต่างระหว่างค่าเป้าหมายกับค่าเอาท์พุทที่ออกจากข่ายงานนิวรอน ดังนั้นวิธีการเรียนรู้แบบนี้จึงเป็นการเรียนรู้แบบชี้นำ สัญญาณการเรียนรู้แสดงดังสมการที่ (2.10)

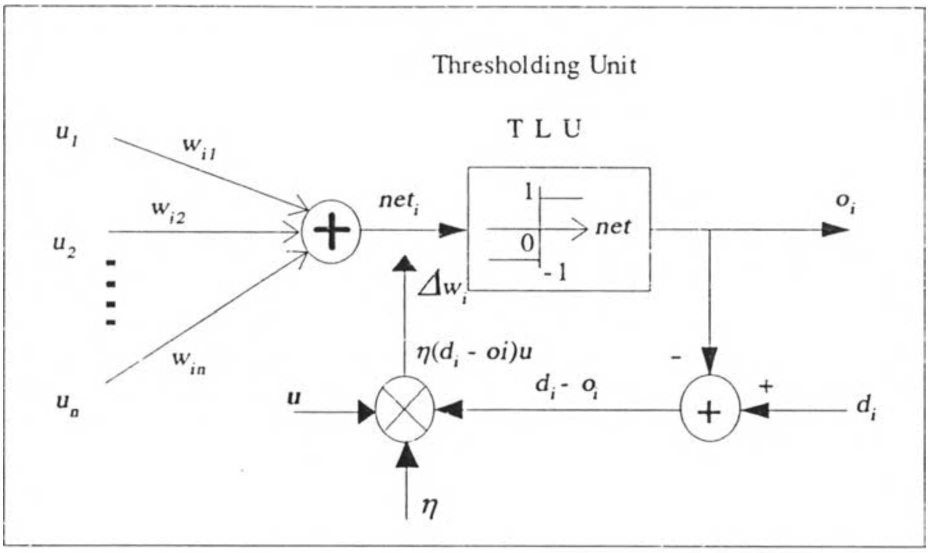
$$r = d_i - o_i \tag{2.10}$$

เมื่อ o_i = TLU($w_i u$) และ d_i คือ ค่าเป้าหมาย โดยการปรับค่าน้ำหนักแสดงดังสมการ (2.11)

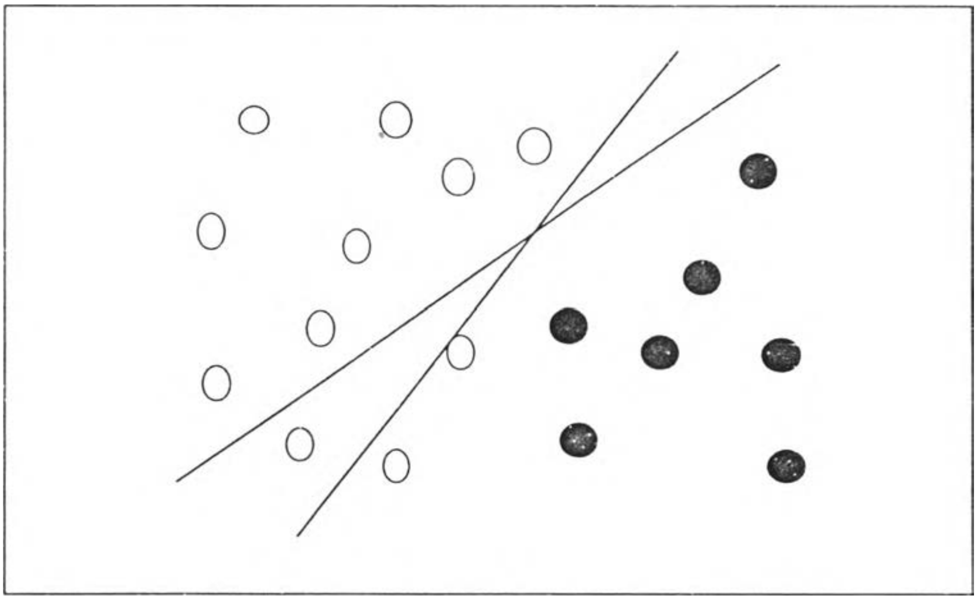
| เมื่อ $j = 1, 2, 3, \ldots\ldots\ldots\ldots n$ (2.11)

การเรียนรู้แบบเปอร์เซปตรอนนำไปประยุกต์ใช้ได้เฉพาะระบบตัวเลขไบนารีเท่านั้น ในการปรับค่าน้ำหนักจะถูกกระทำเมื่อค่าเอาท์พุทจากข่ายงานนิวรัลไม่ตรงกับค่าเป้าหมายเท่านั้น ในกรณีที่เลือกใช้ฟังก์ชันแบบไบโพลาร์เป็นฟังก์ชันกระตุ้น ดังนั้นค่าเป้าหมายคือ -1 กับ 1 กาปรับค่าน้ำหนักจะลดลงเหลือ

$$\Delta w_i = \pm 2\eta U \tag{2.12}$$



รูปที่ 2.12 กฎการเรียนรู้แบบเปอร์เซปตรอน



รูปที่ 2.13 การปรับหาขอบเขตแยกข้อมูล 2 คลาส โดยข่ายงานปอร์เซปตรอนเชิงเส้น

2.6.2 การเรียนรู้แบบเดลตา (delta learning rule : ADALINE)

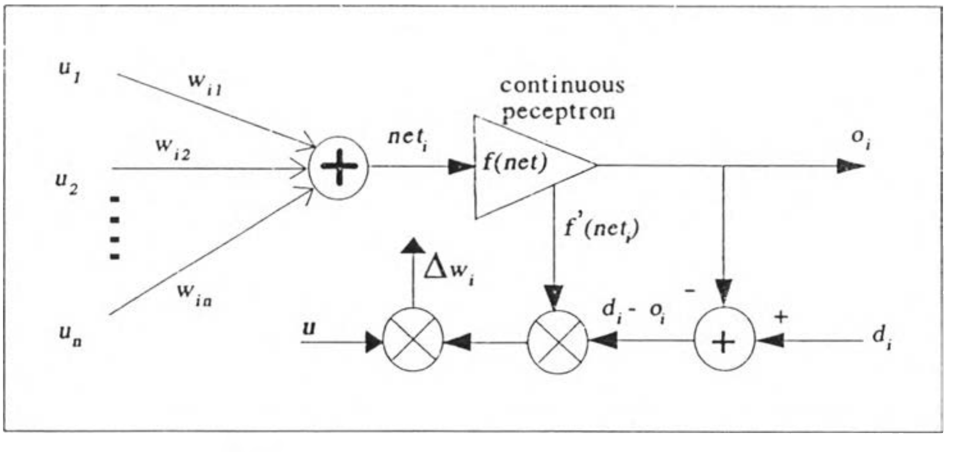
กฎการเรียนรู้แบบเดลตา ถือว่าเป็นการเรียนรู้แบบชี้นำ เพียงชนิดเดียวที่ใช้ได้กับฟังก์ชันกระตุ้นอย่างต่อเนื่อง โดยมีนิยามของสัญญาณการเรียนรู้ดังนี้

$$\text{learning signal} = [d_i - f(net)]\, f'(net) \tag{2.13}$$

เมื่อ $f'(net)$ เป็น อนุพันธุ์ของฟังก์ชันกระตุ้น $f(net)$ ซึ่งคำนวณจาก $net = w_i u$ แบบจำลองข่ายงานรุ่นบุกเบิกเป็นข่ายงาน ชื่อ ADALINE (ADAaptive LINear Element) ที่เสนอโดย McClelland กับ Rumelhart ในปี ค.ศ. 1986

เมื่อเปรียบเทียบการเรียนรู้ของข่ายงานแบบเดลตา กับ เปอร์เซปตรอน พบว่ากฎการเรียนแบบเดลตา ใช้ค่าที่แตกต่างของ $(d_i - o_i)$ ป้อนกลับนำมาเพื่อปรับค่าน้ำหนัก โดยการเรียนรู้จะอยู่บนพื้นฐานการลดค่ากำลังสองของค่าความแตกต่างให้มีค่าน้อยที่สุด โดยมีนิยาม

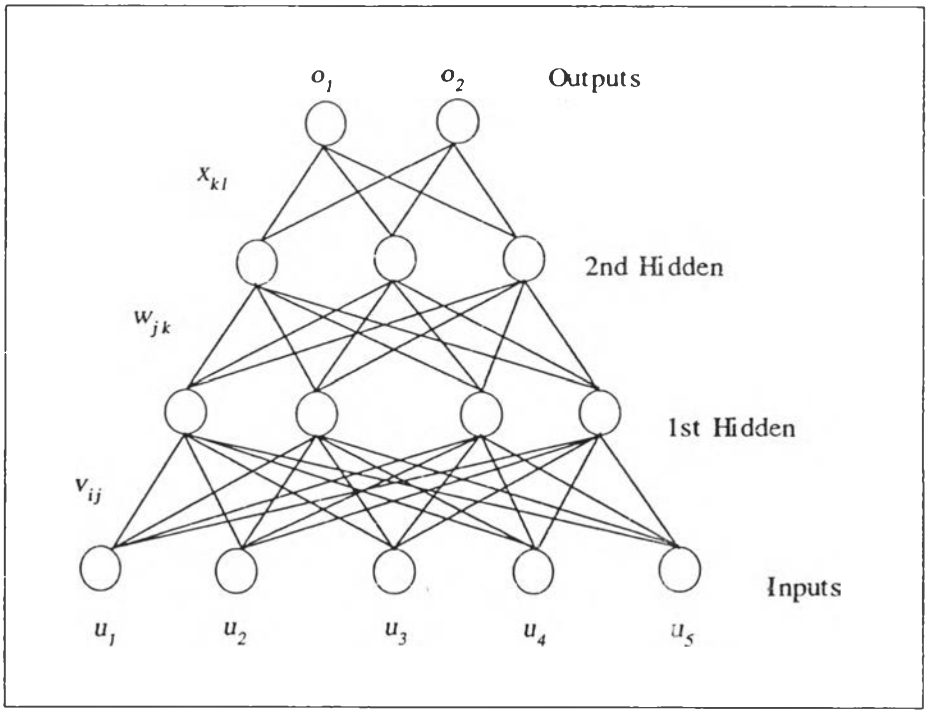
$minimize \;\; E = \frac{1}{2}\sum_{m=1}^{M}(y^m - t^m)^2$ ซึ่งต่างจากการเรียนรู้แบบเปอร์เซปตรอน ซึ่งใช้ค่าความแตกต่างระหว่าง ค่าเป้าหมาย กับ ค่าที่ได้จากข่ายงานนิวรัลโดยตรง ในการปรับค่าน้ำหนัก



รูปที่ 2.14 การเรียนรู้แบบเดลตา (delta learning rule)

2.7 ข่ายงานนิวรัลหลายชั้นไม่เชิงเส้น (nonlinear multilayer neural networks)

เนื่องจากข่ายงานชั้นเดียวที่มีฟังก์ชันเชิงเส้น มีข้อจำกัดมากในการจำแนกและ ประมาณค่าของระบบ ต่อมาจึงได้มีการพัฒนาข่ายงานเป็นแบบหลายชั้น โดยชั้นในข่ายงานนิวรัลนั้นหมายถึง ชั้นของน้ำหนัก (weight layers) ดังรูปที่ 2.15 นั้นถือว่าเป็นข่ายงานนิวรัล 3 ชั้น เนื่องจากมีชั้นน้ำหนัก 3 ชั้น อันจะเป็นตัวแทนของการแปลงเชิงเส้น (linear transformations) ส่วนชั้นของนิวรอนนั้น จะให้ค่าการแปลงแบบไม่เชิงเส้น ดูเหมือนว่าถ้าใช้ข่ายงานนิวรัลที่มีชั้นมากกว่าน่าจะให้ผลที่จะทำให้ข่ายงานมีศักยภาพมากกว่า อย่างไรก็ตามในทางปฏิบัติ พบว่าจำนวนชั้นที่มากเกินไปมักทำให้สมรรถนะของข่ายงานลดลง เนื่องจากต้องใช้เวลานานในการเข้าสู่จุดต่ำสุด นอกจากนี้สัญญาณความผิดพลาดอาจยุ่งเหยิง เมื่อการกระจาย (propagation) ของการปรับน้ำหนักข้ามจำนวนชั้นมากเกินไป อาจทำให้เกิดจุดต่ำสุดในบริเวณ (local minima) เพิ่มขึ้นได้ ดังนั้นจึงมักนิยมใช้ ข่ายงานนิวรัล เพียง 2 ชั้น ในการนำมาใช้เป็นตัวประมาณค่าให้กับฟังก์ชันไม่เชิงเส้น และข่ายงาน 3 ชั้น นั้นมีความสามารถเพียงพอที่จะจำแนกขอบเขตของโพลีฮีดรัล (polyhedral) ว่าเป็นโค้งเว้า (convex) หรือไม่ ชั้นของน้ำหนักจะทำให้เกิดการเปลี่ยนแปลงเชิงเส้น (linear transformations) ส่วนชั้นของนิวรอนจะเกิดการเปลี่ยนแปลงแบบไม่เชิงเส้น(nonlinear) ที่นิวรอน ในทางปฏิบัติโดยทั่วไปแล้วพบว่าชั้นของฮิดเดนเพียง 2 ชั้น ก็ถือว่าเพียงพอต่อการแก้ปัญหาระบบต่างๆ ได้สมบูรณ์ทุกระบบแล้ว



รูปที่ 2.15 ข่ายงานนิวรัลที่มีชั้นน้ำหนัก 3 ชั้น

2.7.1 อัลกอริธึมการเรียนรู้ของข่ายงานจากการกระจายความแตกต่างย้อนกลับ (error back propagation training algorithm, EBPT)

EBPT สามารถนำไปประยุกต์ในการหาจุดเหมาะสมสำหรับ ฟังก์ชันพลังงาน เพื่อการคำนวณหาค่าเกรเดียนท์ได้อย่างมีประสิทธิภาพ สำหรับข่ายงานหลายชั้นที่เป็นฟังก์ชันมูลฐานเชิงเส้น มีสมการไดนามิกดังนี้

$$u_i = \sum_{j=1}^{N_{l-1}} w_{ij}(l) a_j(l-1) + \theta_i(l) \tag{2.14}$$

$$\boldsymbol{a_i(l) = f(u_i(l))} \qquad 1 \le \boldsymbol{l} \le \boldsymbol{L} \quad , \quad 1 \le \boldsymbol{i} \le N_l$$

เมื่อ $a_i(0)$ = หน่วยอินพุท และ $a_i(L)$ = หน่วยเอาท์พุท

L = ชั้นสูงสุดของข่ายงาน (ชั้นเอาท์พุท)

โดยมีฟังก์ชันกระตุ้นที่นิยมทั่วไป คือ ฟังก์ชันซิกมอยด์ ที่มีสมการ

$$f(u_l) = \frac{1}{1+e^{-u_l/\lambda}}$$

สำหรับสมการไดนามิกอื่นๆ ที่น่าสนใจคือ ฟังก์ชันมูลฐานเชิงรัศมี มักใช้ฟังก์ชันเกาส์เสียน

สำหรับอัลกอริธึม แบบ EBPT นั้นเมื่อปรับน้ำหนัก w_{ij} เพื่อให้ค่า E มีค่าต่ำสุด

โดยมีการเรียนรู้พื้นฐานแบบเกรเดียนท์ เป็น

$$w_{ij}^{m+1}(l) = w_{ij}^{m}(l)+\Delta w_{ij}^{m}(l) \tag{2.15}$$

โดยการหาอนุพันธ์ของ EBPT จะใช้วิธีการตามเทคนิคของกฎลูกโซ่ (chain rule)

$$\begin{aligned}\Delta w_{ij}^{m}(l) &= -\eta \frac{\partial E}{\partial w_{ij}^{m}(l)} \\ &= -\eta \frac{\partial E}{\partial a_i^{m}(l)} \frac{\partial a_i^{m}(l)}{\partial w_{ij}^{m}(l)} \\ &= \eta \delta_i^{m}(l) f'(u_i^{m}(l)) a_j^{m}(l-1)\end{aligned} \tag{2.16}$$

เมื่อ สัญญาณความผิดพลาด ได้แก่ $\delta_i^{m}(l) = -\dfrac{\partial E}{\partial a_i^{m}(l)}$ (2.17)

2.7.2 กฎการกระจายย้อนกลับ (Back-propagation Rule) สำหรับข่ายงาน

ในการฝึกข่ายงานเพื่อปรับค่าน้ำหนักให้ทำการลดค่าความแตกต่างกำลังสอง ระหว่างค่าเป้าหมาย กับ ผลลัพธ์ จากข่ายงาน ดังนั้นจะได้สมการของความแตกต่างเป็น

$$E = \frac{1}{2}\sum_{m=1}^{M}\sum_{i=1}^{N}[t_i^{m} - a_i^{m}(L)]^2 \tag{2.18}$$

เมื่อ M คือ จำนวนชุดข้อมูลการฝึก และ N คือ จำนวนมิติของเอาท์พุท

อย่างไรก็ตาม อัลกอริธึมแบบ **EBPT** สามารถสรุปได้เป็น 2 ขั้นตอน คือ

ขั้นตอนที่ 1 หาค่าสัญญานความแตกต่าง δ_i โดยการปรับค่าการกระจายของน้ำหนักย้อนกลับ (back propagation) แบ่งเป็นค่าเริ่มแรกในชั้นบนสุด (ชั้นเอาท์พุท) นั้นค่าสัญญาณความแตกต่างหาได้จาก

$$\delta_i^m(L) = -\frac{\partial E}{\partial a_i^m(L)} = t_i^m - a_i^m(L) \tag{2.19}$$

และ หาค่าสัญญาณความแตกต่าง สำหรับค่าเริ่มแรกในชั้นป้อนกลับ (ชั้นฮิดเดนที่อยู่ถัดลงมาจากชั้นเอาท์พุท)

$$\delta_i^m(l) = \frac{\partial E}{\partial a_i^m(l)} = -\sum_{j=1}^{N_{l+1}} \frac{\partial E}{\partial u_j^m(l+1)} \frac{\partial u_j^m(l+1)}{\partial a_i^m(l)} \tag{2.20}$$

ตามลำดับของ $l = L$-1,...,1 จะได้สมการรวบยอดเป็นดังนี้

$$\delta_i^m(l) = \sum_{j=1}^{N_{l+1}} \delta_j^m(l+1) f'(u_i^m(l+1) w_{ji}^m(l+1) \tag{2.21}$$

ดังนั้นจะเห็นว่า ในกรณีที่ชั้นฮิดเดนจำนวน n ชั้น ก็ต้องคำนวณหาค่าในสมการ (2.21) จำนวน n ครั้งด้วยเช่นกัน เพื่อนำไปใช้ในการเป็นแฟคเตอร์ในการปรับค่าน้ำหนัก ในขั้นตอนที่ 2

ขั้นตอนที่ 2 จากสมการที่ 2.16 และ 2.18 ค่าน้ำหนักเชื่อมโยง ระหว่างชั้นที่ l กับชั้นที่ l-1 สามารถปรับค่าโดยความสัมพันธ์ดังนี้ (ตามลำดับของ $i = L$-$1,...,1$)

$$w_{ij}^{m+1}(l) = w_{ij}^m(l) + \eta \sigma_i^m f'(u_i^m(l) a_j^m(l-1) \tag{2.22}$$

การเรียนรู้แบบการกระจายย้อนกลับนั้น สัญญาณความแตกต่างของชั้นที่ต่ำกว่า $\delta_j^m(l)$ สามารถคำนวณได้เช่นเดียวกับการรวมเชิงเส้น (linear combination) ของสัญญาณความแตกต่าง ของชั้นที่อยู่เหนือกว่า $\delta_j^m(l+1)$ ดังนั้นสัญญาณความแตกต่างจึงสามารถกระจายย้อนกลับผ่านชั้นทั้ง

หมดจากบนสู่ล่าง นั่นหมายถึงว่า อิทธิพลจากชั้นที่อยู่เหนือกว่าไปยังชั้นที่อยู่ต่ำกว่า (หรือในทางกลับกัน) ก็จะมีผลกระทบถึงกัน ตามค่าของ สัญญาณความแตกต่าง

2.8 อัลกอริธึมการเรียนรู้ของข่ายงานนิวรัลแบบการกระจายความแตกต่างย้อนกลับ

2.8.1 ขั้นตอนการเรียนรู้

อัลกอริธึมการเรียนรู้ของข่ายงานนิวรัลแสดงเป็นผังงานดังในรูปที่ 2.14 เริ่มด้วยการกำหนดจำนวนรูปแบบ (pattern) ข้อมูลการเรียนรู้ระหว่างกลุ่มข้อมูลอินพุต, u_i กับค่าเป้าหมายที่สอดคล้องกัน, d_i เป็นจำนวน P รูปแบบ (examplar)

$$\{ u_1,\ d_1, u_2, d_2, \ldots\ldots\ldots\ldots\ldots, u_p, d_p \}$$

เมื่อ u_i เป็นเวคเตอร์มีขนาด $(I$ *1) และ d_i มีขนาด (K * 1) ซึ่ง $i = 1, 2, 3, \ldots, P$ มีค่าเป็น -1 สำหรับขั้นฮิดเดน (hidden layer) มีขนาด $(J * 1)$ *โดยที่*สมาชิกตัวที่ J *ของ* y *มีค่าคงที่* -1 ขั้นตอนการเรียนรู้ของเครือข่ายนิวรอลเป็นดังนี้

ก. **ขั้นตอนที่ 1** : กำหนดค่าอัตราการเรียนรู้ **(learning rate, η)** ให้มีค่ามากกว่าศูนย์ กำหนดค่าผิดพลาดต่ำสุด (E_{min}) และกำหนดค่าเริ่มต้นให้กับน้ำหนัก (weight) ที่เชื่อมต่อระหว่างนิวรอนแต่ละชั้น, W, V โดยใช้ตัวเลขสุ่มที่มีค่าน้อย ๆให้กับน้ำหนักที่มีขนาด $W(K*J)$, $V(J*I)$ เริ่มรูปแบบการเรียนรู้ $p = 1$ กำหนดความผิดพลาด $E = 0$ และเริ่มการเรียนรู้ $q = 1$

ข. **ขั้นตอนที่ 2** :ส่งค่าอินพุดและค่าเป้าหมายที่สอดคล้องกันให้นิวรอนของชั้นรับข้อมูล

นั่นคือ $u = u_p,\quad d = d_p$

คำนวณค่าผลลัพธ์ให้นิวรอนในขั้นฮิดเดน และ ชั้นแสดงผล (output layer) โดยใช้ฟังก์ชัน

ซิกมอยด์ เป็น ฟังก์ชันกระตุ้น

ค. ขั้นตอนที่ 3 : คำนวณค่าความผิดพลาด

$E = 0.5 * (d_k - o_k)^2 + E$ เมื่อ $k = 1, 2, \ldots, K$

ง. ขั้นตอนที่ 4 : คำนวณค่าเวคเตอร์สัญญาณความผิดพลาด , δ_o, δ_y

- สัญญาณความผิดพลาดสำหรับชั้นแสดงผล

$\delta_{ok} = (d_k - o_k)(1 - o_k)o_k$ เมื่อ $k = 1, 2, \ldots, K$

- สัญญาณความผิดพลาดสำหรับขั้นฮิดเดน

$\delta_{yj} = y_j(1 - y_j)\sum_{k=1}^{K}\delta_{ok}w_{kj}$ เมื่อ $j = 1, 2, \ldots, J$

ในกรณีที่มีจำนวนชั้นฮิดเดน n ชั้น ก็ต้องหาค่าสัญญาณความผิดพลาด n ครั้งด้วย ดังที่ได้อธิบายไว้ในหัวข้อที่ 2.7.2

จ. ขั้นตอนที่ 5 : ปรับค่าน้ำหนักที่เชื่อมระหว่างชั้นแสดงผลกับขั้นฮิดเดน

โดยอาศัยวิธี เกรเดียนท์ค่าเวกเตอร์ของค่าความแตกต่าง แล้วนำมาปรับค่าน้ำหนักเชื่อมโยงระหว่างชั้นแสดงผลกับขั้นฮิดเดน ดังแสดงในรูปที่ 2.15

$w_{kj} = w_{kj} + \eta d_{ok} y_j$

เมื่อ $k = 1, 2, \ldots, K$ และ $j = 1, .2, \ldots, J$

ฉ. ขั้นตอนที่ 6 : ปรับค่าน้ำหนักที่เชื่อมระหว่างชั้นรับข้อมูลกับขั้นฮิดเดน

$v_{kj} = v_{kj} + \eta d_{yj} z_i$ เมื่อ $j = 1, 2, \ldots, J$ และ $i = 1, 2, \ldots, I$

ช. ขั้นตอนที่ 7 : ตรวจสอบชุดข้อมูลการเรียนรู้

ถ้ารูปแบบ (pattern) การเรียนรู้ยังไม่ถูกทำงานจนครบรอบ ($p < P$) ให้ส่งข้อมูลของรูปแบบถัดไป ไปยังนิวรอนส่วนรับข้อมูล โดยที่ $p = p + 1$ และ q = $q + 1$ แล้วจึงกลับไปดำเนินการขั้นที่ 2 แต่ถ้ารูปแบบการเรียนรู้ได้ถูกปฏิบัติการจนครบรอบหมดแล้ว ให้ไปดำเนินการขั้นที่ 8 ถัดไป

ซ. ขั้นตอนที่ 8 : ตรวจสอบค่าความผิดพลาดกับค่าต่ำสุดที่ตั้งไว้

ขั้นตอนการเรียนรู้จะเสร็จสมบูรณ์เมื่อ ค่าความผิดพลาดในการเรียนรู้ มีค่าน้อยกว่าค่าความผิดพลาดต่ำสุด ที่ตั้งไว้ ($E < E_{mim}$) ถ้า $E > Emin$ แล้วให้ $E = 0$ และ $p = 1$ แล้วเริ่มวงจรการเรียนรู้ใหม่ตามขั้นตอนที่ 2

2.8.2 การเรียนรู้โดยวิธีโมเมนตัม

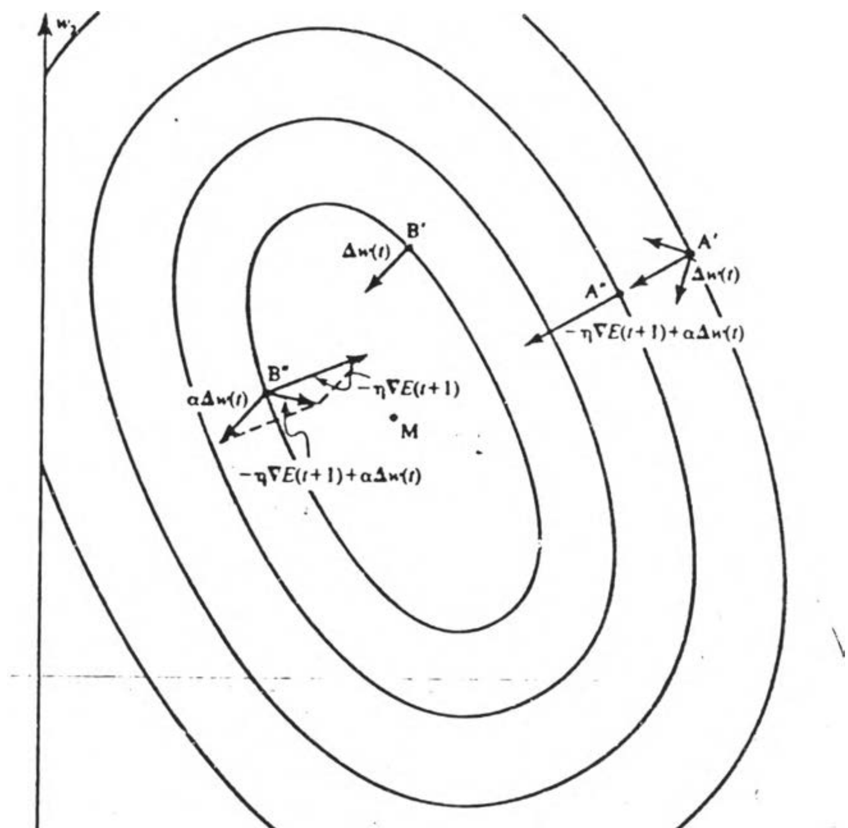
วิธีโมเมนตัมช่วยเร่งให้การเรียนรู้ของข่ายงานนิวรัลที่อาศัยการกระจายความผิดพลาดย้อนกลับเข้าสู่ (converge) ค่าเป้าหมายที่ต้องการ ทั้งนี้โดยการนำค่าน้ำหนักจากครั้งล่าสุด มาใช้ในการปรับหาค่าน้ำหนักในปัจจุบันตามขั้นตอนที่ 5 และ ขั้นตอนที่ 6 ในอัลกอริธึมการเรียนรู้ของข่ายงานนิวรัล ตามสัดส่วนของค่าโมเมนตัม ดังสมการ (2.23)

เขียนใหม่

$$\Delta w(t) = -\eta \nabla E(t) + \alpha \Delta w(t-1) \qquad (2.23)$$

เมื่อ α คือ แฟคเตอร์โมเมนตัมที่มักเลือกค่าอยู่ระหว่าง 0.1 - 0.8 รูปที่ 2.16 แสดงวิธีการในการเร่งการเรียนรู้ ของข่ายงานนิวรัลโดยการใช้วิธีโมเมนตัม ถ้าการเรียนรู้เริ่มต้นที่จุด A' เมื่อหาค่าเกรเดียนท์ของค่าความผิดพลาดเทียบกับค่าน้ำหนัก $\partial E/\partial w_1$ และ $\partial E/\partial w_2$ ที่จุด A', A'' พบว่ามีเครื่องหมายตรงกัน ดังนั้นองค์ประกอบเกรเดียนท์จะถูกนำมารวมกัน ทำให้การเข้าสู่เป้า

หมายรวดเร็วขึ้น โดยการปรับน้ำหนักที่จุด A'' โดยอาศัยค่าน้ำหนักที่จุด A' เข้ามาร่วมด้วย ตามสัดส่วนของค่าโมเมนตัม



รูปที่ 2.16 การเรียนรู้แบบการกระจายความผิดพลาดย้อนกลับ ที่มีการรวมค่าแฟคเตอร์โมเมนตัม มาใช้ในการปรับค่าน้ำหนัก