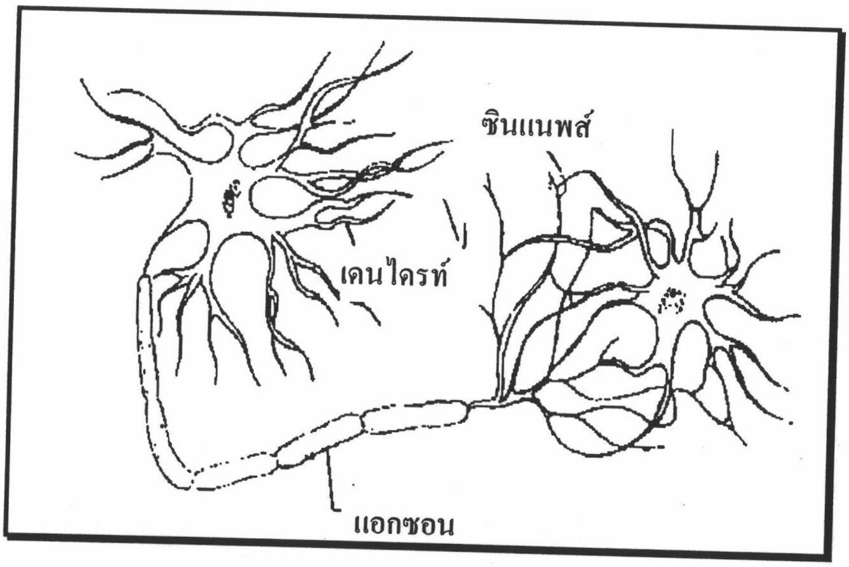
บทที่ 3

ทฤษฎีเกี่ยวกับข่ายงานนิวรัล

บทนี้กล่าวถึงการศึกษาเกี่ยวกับข่ายงานนิวรัลซึ่งได้แก่โครงสร้างของข่ายงานในยุคแรก ๆ และโครงสร้างของข่ายงานในปัจจุบัน โครงสร้างของข่ายงานโดยทั่วไปเป็นแบบจำลองข่ายงานนิวรัลที่ใช้แนวความคิดมาจากระบบประสาทโดยมีองค์ประกอบอยู่ 3 ส่วนคือ ตัวเซลนิวรอน, เดนไดรท์ และซินแนพส์ โครงสร้างของข่ายงานในยุคแรก ๆ จะมีชั้นของน้ำหนักเพียงชั้นเดียวซึ่งทำให้ข่ายงานนี้สามารถแก้ปัญหาแบบเชิงเส้นได้ แต่ไม่สามารถแก้ปัญหาแบบไม่เชิงเส้น หลังจากนั้นนักวิจัยต่าง ๆ ได้ศึกษาโครงสร้างของข่ายงานที่มีความซับซ้อนมากขึ้นโดยการเพิ่มชั้นของน้ำหนัก และสามารถแก้ปัญหาแบบไม่เชิงเส้นได้เป็นอย่างดี ในตอนท้ายกล่าวถึงอัลกอริธึมการกระจายย้อนกลับ และการเรียนรู้โดยการใช้โมเมนตัมซึ่งจะช่วยเร่งให้เกิดการเรียนรู้ได้เร็วขึ้น

3.1 ชีววิทยาของนิวรอนและแบบจำลองของข่ายงานนิวรัลในยุคแรก ๆ

ข่ายงานนิวรัลได้แนวคิดมาจากการเลียนแบบโครงสร้างระบบประสาทของมนุษย์ ซึ่งประกอบด้วยกลุ่มของนิวรอนจำนวนมาก โดยสามารถติดต่อกับนิวรอนหน่วยอื่น ๆ ด้วยแอกซอน (axon) และซินแนพส์ (synapes) ข้อสมมติฐานเกี่ยวกับแบบจำลองของระบบประสาทคือการติดต่อกันระหว่างนิวรอนที่อาศัยการกระตุ้นทางไฟฟ้า และการทำงานของนิวรอนนั้นเกิดขึ้นเนื่องจากกระบวนการทางชีวเคมี ในทางชีววิทยานิวรอนมีองค์ประกอบ 3 ส่วนซึ่งแสดงในรูปที่ 3.1 คือ



รูปที่ 3.1 โครงสร้างของเซลประสาท

1. ตัวเซลนิวรอน (cell body) ทำหน้าที่รวบรวมสัญญาณจากเดนไดรท์ และส่งไปให้แก่แอกซอน

2. เดนไดรท์ (dendrite) ทำหน้าที่รับสัญญาณจากแอกซอนของนิวรอนตัวอื่นเข้ามา และส่งต่อไปยังตัวเซลนิวรอน

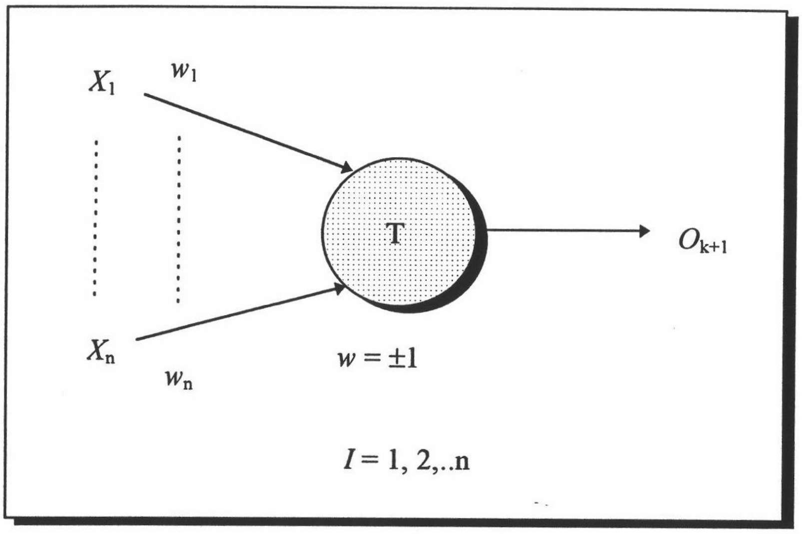
3. แอกซอน (axon) ทำหน้าที่นำสัญญาณที่ออกจากตัวเซลนิวรอน และส่งไปยังเดนไดรท์ของนิวรอนถัดไป

เดนไดรท์จะรับสัญญาณจากแอกซอนของนิวรอนอื่นเข้ามาและส่งต่อไปให้ตัวเซลนิวรอน ตัวเซลนิวรอนจะรวมสัญญาณที่ได้รับส่งต่อไปให้แอกซอนจนถึงรอยต่อซินแนพส์ ซึ่งเป็นรอยต่อระหว่างแอกซอนของนิวรอนตัวปัจจุบัน และเดนไดรท์ของนิวรอนตัวถัดไป จากแนวความคิดพื้นฐานดังกล่าวนักวิจัยจึงได้สนใจที่จะศึกษากลไก และโครงสร้างของระบบประสาทของมนุษย์ซึ่งนำไปสู่การพัฒนาของแบบจำลองสำหรับการแก้ปัญหาที่ซับซ้อนซึ่งได้แก่การแก้ปัญหาแบบไม่เชิงเส้น

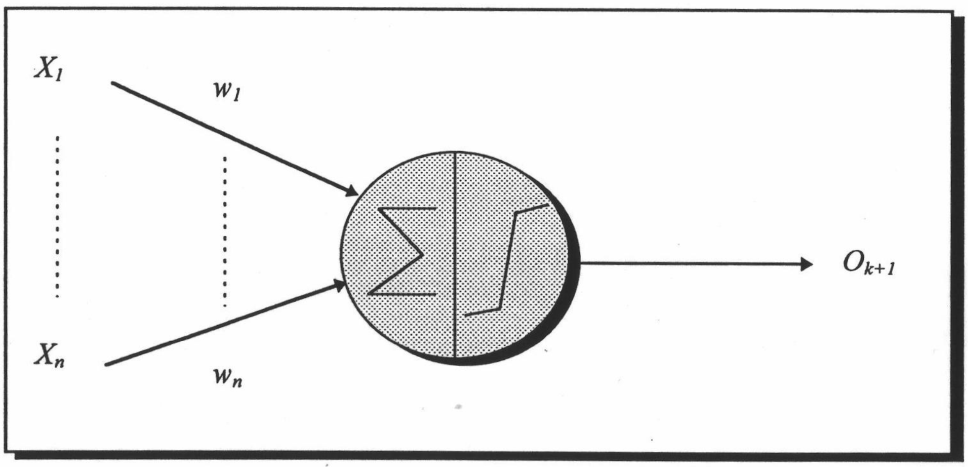
McCulloch และ Pitts (1943) ซึ่งเป็นผู้บุกเบิกของข่ายงานนิวรัล ได้เสนอตรรกเทรีสโฮลด์ (threshold logic) ซึ่งแสดงในรูปที่ 3.2 แบบจำลองของ McCulloch และ Pitts มีค่าอินพุทและเอาท์พุทเป็น 0 กับ 1 เท่านั้น และค่าน้ำหนักมีค่าคงที่ อีกทั้งไม่มีการติดต่อถึงกันระหว่างนิวรอน แบบจำลองชนิดนี้ใช้ได้กับระบบเชิงเส้น เนื่องจากมีเพียงนิวรอนเพียงหน่วยเดียวเท่านั้นโดยไม่มีการเชื่อมกับนิวรอนหน่วยอื่น สัญญาณที่ออกจากนิวรอนกำหนดด้วยเงื่อนไขดังสมการ (3.1)

$$O_{k+1} = \begin{cases} 1 \;\; if \;\; \sum_{i=1}^{n} w_i x_{ik} \;\; \geq T \\ 0 \;\; if \;\; \sum_{i=1}^{n} w_i x_{ik} \;\; < T \end{cases} \tag{3.1}$$

Rosenblatt (1958) เป็นผู้เสนอแบบจำลองเปอร์เซปตรอนซึ่งเป็นแบบจำลองข่ายงานแบบตัดสินใจเพื่อใช้ในการแยกข้อมูล 2 คลาสออกจากกัน Minsky และ Papert (1969) พิสูจน์



รูปที่ 3.2 แบบจำลองการทำงานของนิวรอนเสนอโดย McCulloch และ Pitts



รูปที่ 3.3 หน้าที่และการทำงานของนิวรอน

คุณสมบัติของเปอร์เซปตรอน และชี้ให้เห็นถึงข้อจำกัดของแบบจำลองนี้ซึ่งสามารถแก้ปัญหาเฉพาะระบบที่เป็นเชิงเส้นเท่านั้น

3.2 แบบจำลองของข่ายงานนิวรัลในปัจจุบัน

แบบจำลองของข่ายงานนิวรัลในปัจจุบันประกอบด้วยชั้นของนิวรอน และส่วนเชื่อมโยงระหว่างอินพุทและเอาท์พุทที่เรียกว่าน้ำหนัก (weight) อินพุทของนิวรอนซึ่งจะถูกนำมารวมกันโดยใช้ฟังก์ชันมูลฐาน (basis function) เอาท์พุทที่ออกจากฟังก์ชันมูลฐานจะถูกแปลงด้วยฟังก์ชันกระตุ้น (activation function) ซึ่งจะได้เอาท์พุทของนิวรอน และส่งเอาท์พุทของนิวรอนนี้ไปเป็นอินพุทของนิวรอนถัดไปเป็นทอด ๆ

หน้าที่ของนิวรอนแต่ละหน่วยซึ่งแสดงในรูปที่ 3.3 มีดังนี้

1. รับสัญญาณจากนิวรอนหน่วยอื่น
2. รวมสัญญาณจากนิวรอนหน่วยอื่นเข้าด้วยกันโดยใช้ฟังก์ชันมูลฐาน
3. แปลงสัญญาณที่รวมได้โดยใช้ฟังก์ชันกระตุ้น
4. ส่งผลลัพธ์ที่ได้จากฟังก์ชันกระตุ้นไปยังนิวรอนถัดไป

น้ำหนัก (weight) หรือที่เรียกว่าสัญญาณที่เชื่อมโยงระหว่างนิวรอนจะถูกปรับเปลี่ยนค่าอยู่ตลอดเวลา ในระหว่างการเรียนรู้ข่ายงานสร้างแบบจำลองภายในขึ้นมาเพื่อให้มีค่าใกล้เคียงกับระบบแบบจำลองของข่ายงานนิวรัลซึ่งสามารถใช้แทนแบบจำลองของกระบวนการได้ก็ต่อเมื่อค่าผลลัพท์ที่ออกจากระบบมีค่าใกล้เคียงกับค่าผลลัพท์ที่ออกจากข่ายงาน โครงสร้าง

ของข่ายงานประกอบไปด้วยชั้นของน้ำหนักอย่างน้อย 1 ชั้น สำหรับจำนวนชั้นของนิวรอนขึ้นอยู่กับจำนวนชั้นของน้ำหนัก ถ้าข่ายงานมีชั้นของน้ำหนัก 1 ชั้นข่ายงานจะมีชั้นของนิวรอน 2 ชั้นซึ่งได้แก่ชั้นอินพุท และชั้นเอาท์พุท ถ้าข่ายงานมีชั้นของน้ำหนักตั้งแต่ 2 ชั้นขึ้นไป ข่ายงานจะมีชั้นของนิวรอน 3 ชั้นซึ่งได้แก่ชั้นอินพุท ชั้นซ่อน และชั้นเอาท์พุท ในข่ายงานนิวรัลชั้นของนิวรอนสองชั้นจะติดต่อถึงกันโดยใช้โครงสร้างเชื่อมโยง

ในข่ายงานนิวรัลหนึ่ง ๆ จะมีฟังก์ชันมูลฐาน ฟังก์ชันกระตุ้น และโครงสร้างการเชื่อมโยงที่แตกต่างกันกันไป โดยสามารถแบ่งประเภทของฟังก์ชันมูลฐาน ฟังก์ชันกระตุ้น และโครงสร้างการเชื่อมโยงได้ดังนี้

3.2.1 ฟังก์ชันมูลฐาน (Basis function)

ฟังก์ชันมูลฐานคือการแทนการแมพพิงก์ (mapping) ทางคณิตศาสตร์ด้วยฟังก์ชัน, $u(w,x)$ เมื่อ w คือ แมทริกซ์ของน้ำหนักและ x คืออินพุทเวคเตอร์ ฟังก์ชันมูลฐานทำหน้าที่ในการรวมสัญญาณที่ได้รับมาจากนิวรอนหน่วยอื่น ๆ เข้าด้วยกัน และฟังก์ชันกระตุ้นจึงแปลงสัญญาณนี้เป็นเอาท์พุทของนิวรอน ฟังก์ชันมูลฐานสามารถแบ่งเป็น 2 ประเภท (ดูรูปที่ 3.4) คือ

ก. ฟังก์ชันมูลฐานเชิงเส้น (linear basis function)

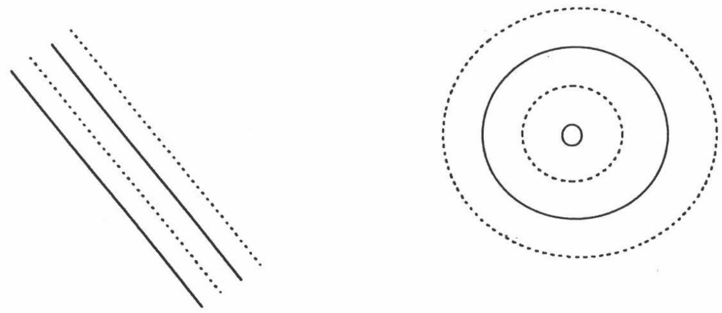
เป็นฟังก์ชันชนิดไฮเปอร์แพลน (hyperplane) โดยเป็นฟังก์ชันมูลฐานอันดับที่ 1 ค่า net ที่ได้เป็นผลรวมเชิงเส้นของค่าอินพุทซึ่งแสดงในสมการ (3.3)

$$u_i(w,x) = \sum_{j=1}^{n} w_{ij} x_j \qquad (3.3)$$

ข. ฟังก์ชันมูลฐานเชิงรัศมี (radial basis function)

เป็นฟังก์ชันไฮเปอร์สเฟียร์ (hypersphere) โดยเป็นฟังก์ชันมูลฐานอันดับที่ 2 ค่า net ที่ได้จะแทนระยะทางไปยังชุดข้อมูลอ้างอิง ซึ่งแสดงในสมการ (3.4)

$$u_i(w,x) = \sqrt{\sum_{j=1}^{n} (x_i - w_{ij})^2} \tag{3.4}$$



(a) (b)

รูปที่ 3.4 ฟังก์ชันมูลฐาน (a) linear (hyperplane) (b) radial (hypersphere)

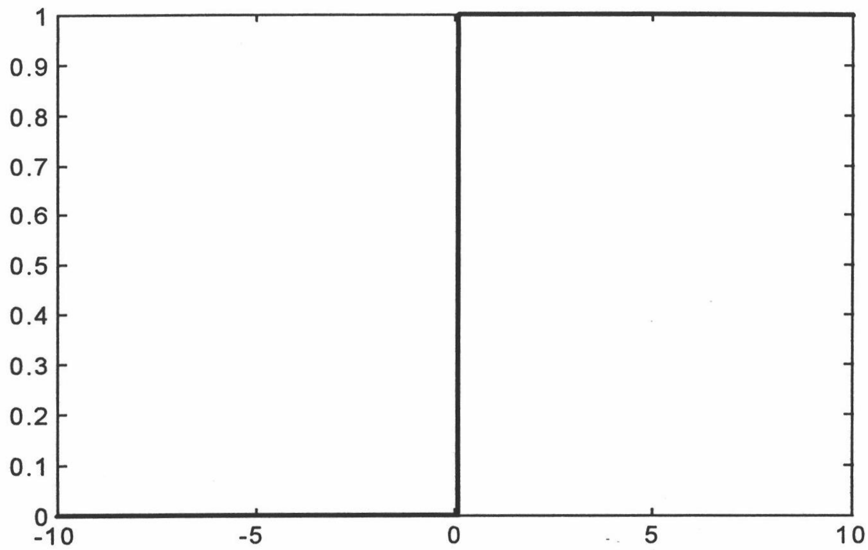
3.2.2 ฟังก์ชันกระตุ้น (Activation function)

ฟังก์ชันกระตุ้นทำหน้าที่แปลงค่า net ที่อยู่ในรูปของฟังก์ชันมูลฐานให้เป็นเอาท์พุทของนิวรอน ฟังก์ชันกระตุ้นแบ่งเป็น 2 ประเภทคือ ฟังก์ชันกระตุ้นแบบเชิงเส้น และฟังก์ชันกระตุ้นแบบไม่เชิงเส้น

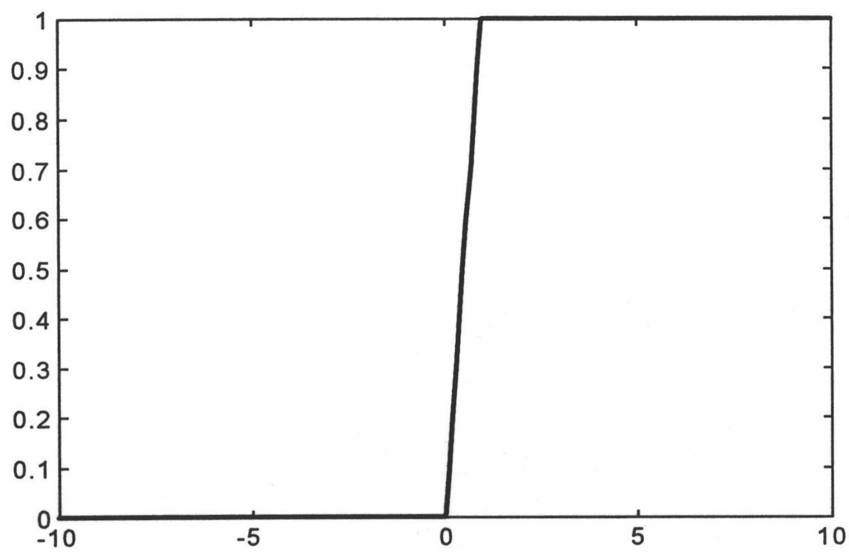
3.2.2.1 ฟังก์ชันกระตุ้นแบบเชิงเส้น (Linear activation function)

ฟังก์ชันกระตุ้นแบบเชิงเส้นได้แก่ฟังก์ชันแบบสเต็พ (step function) ฟังก์ชันแรมพ์ (ramp function) ซึ่งแสดงในรูปที่ 3.5 (a) และ 3.5 (b) ตามลำดับโดยมีสมการคณิตศาสตร์ดังนี้คือ

ก. ฟังก์ชันแบบสเต็พ



(a)



(b)

รูปที่ 3.5 ฟังก์ชันกระตุ้นแบบเชิงเส้น (a) ฟังก์ชันแบบสเต็พ (step function) (b) ฟังก์ชันแรมพ์ (ramp function)

$$f(net) = \begin{cases} 1 \ \ if \ \ \ net > 0 \\ 0 \ \ otherwise \end{cases} \tag{3.5}$$

ข. ฟังก์ชันแรมพ์ (ramp function)

$$f(net) = \begin{cases} 1 \ \ if \ \ \ net \geq 1 \\ net \ \ if \ \ \ |net| < 1 \\ -1 \ \ if \ \ \ net \leq -1 \end{cases} \tag{3.6}$$

3.2.2.2 ฟังก์ชันกระตุ้นแบบไม่เชิงเส้น (Nonlinear activation function)

ฟังก์ชันกระตุ้นแบบไม่เชิงเส้นได้แก่ ฟังก์ชันซิกมอยด์ (sigmoid function) และฟังก์ชันเกาส์เสียน (guassian function) ซึ่งแสดงในรูปที่ 3.6 (a) และ 3.6 (b) ตามลำดับ ฟังก์ชันกระตุ้นที่นิยมใช้กันมากที่สุดคือ ฟังก์ชันซิกมอยด์

สมการคณิตศาสตร์ของฟังก์ชันกระตุ้นแบบไม่เชิงเส้นคือ

ก. ฟังก์ชันซิกมอยด์

$$f(net) = \frac{1}{1+e^{-net}} \tag{3.7}$$

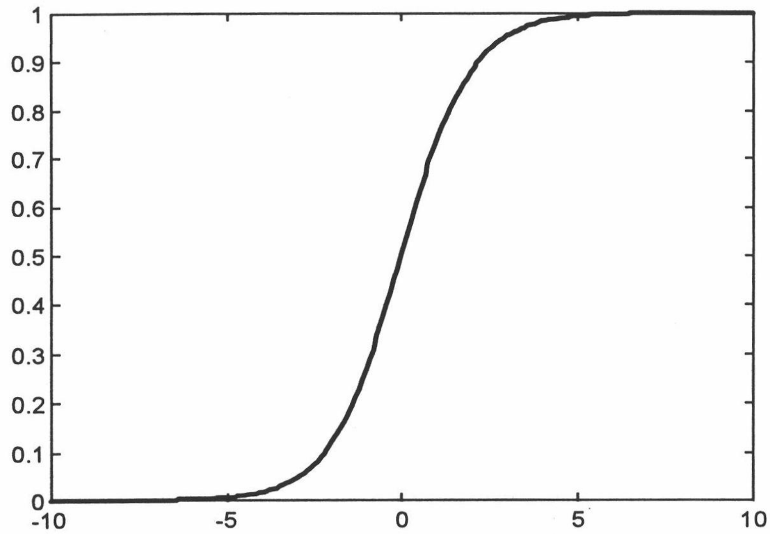
ข. ฟังก์ชันเกาส์เสียน

$$f(net) = ce^{-net^2} \tag{3.8}$$

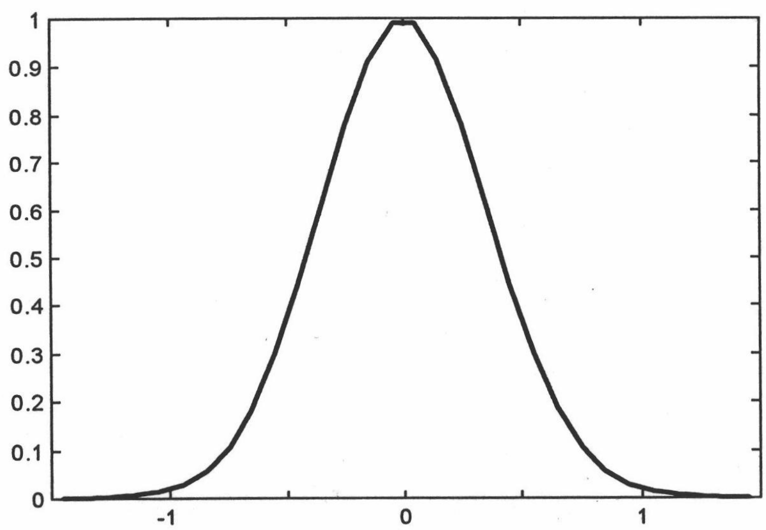
3.2.3 โครงสร้างเชื่อมโยงของน้ำหนักระหว่างชั้นของข่ายงานนิวรัล

ข่ายงานนิวรัลประกอบด้วยนิวรอน และแมทริกซ์ของน้ำหนัก ประพฤติกรรมของข่ายงานขึ้นกับการเชื่อมโยงระหว่างนิวรอน ชั้นของนิวรอนสองชั้นติดต่อถึงกันโดยผ่านข่ายงานเชื่อมต่อที่เป็นน้ำหนักแสดงในรูปที่ 3.7 ซึ่งแบ่งเป็น 4 ประเภทคือ

I 17180405



(a)



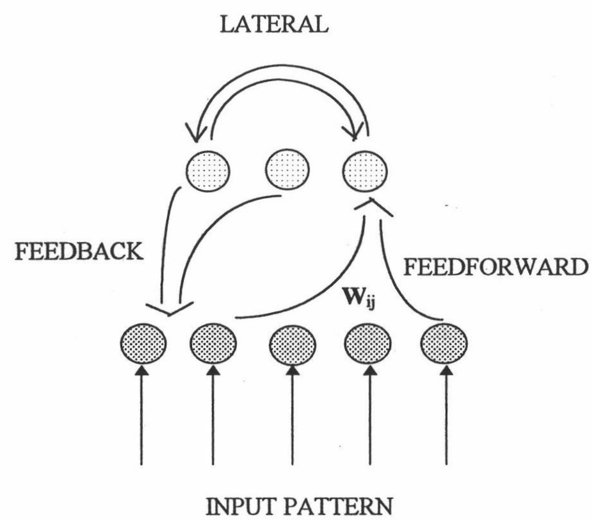
(b)

รูปที่ 3.6 ฟังก์ชันกระตุ้นแบบไม่เชิงเส้น (a) ฟังก์ชันซิกมอยด์ (sigmoid function)

(b) ฟังก์ชันเกาส์เสียน (guassian function)

ก. การเชื่อมโยงแบบป้อนไปข้างหน้า (feedforward connection) เป็นการนำข้อมูลของนิวรอนจากชั้นที่ต่ำกว่าและนำไปใช้ในชั้นที่อยู่เหนือกว่า

ข. การเชื่อมโยงแบบป้อนกลับ (feedback connection) เป็นการนำข้อมูลจากชั้นที่อยู่เหนือกว่าและส่งกลับลงมายังชั้นที่อยู่ต่ำกว่า



รูปที่ 3.7 โครงสร้างพื้นฐานของข่ายงานนิวรัลที่แสดงการเชื่อมโยงแบบต่าง ๆ

ค. การเชื่อมโยงภายในชั้น (lateral connections) เป็นการนำข้อมูลภายในชั้นเดียวกันและกระจายกลับมาใช้ในชั้นเดิม

ง. การเชื่อมโยงแบบหน่วงเวลา (time-delayed connections) เป็นการนำข้อมูลมาหน่วงเวลาแล้วนำเข้าไปรวมกับการเชื่อมโยงของข้อมูล เพื่อให้ได้แบบจำลองเป็นไดนามิกเชิงเวลา (temporal dynamics) ซึ่งนำมาประยุกต์ใช้กับการระลึกรูปแบบเชิงเวลา (temporal pattern recognition)

3.3 การจำแนกข่ายงานนิวรัล

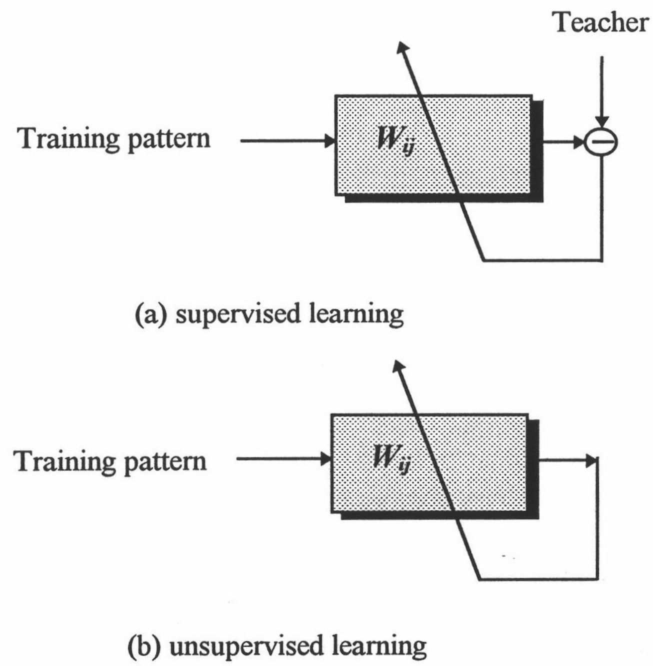
ข่ายงานนิวรอนสามารถจำแนกได้เป็นสองประเภทคือข่ายงานนิวรัลแบบมีการชี้นำ (supervised neural network) และข่ายงานนิวรัลแบบไม่มีการชี้นำ (unsupervised neural network) สุรพล คำสุภา (1995) แนะนำว่าข่ายงานนิวรัลแบบมีการชี้นำเรียนรู้ได้รวดเร็วมากกว่าข่ายงานนิวรัลแบบไม่มีการชี้นำเนื่องจากมีผู้ฝึกช่วยในการเรียนรู้ งานวิจัยนี้จะใช้ข่ายงานที่ใช้ข่ายงานนิวรัลแบบมีการชี้นำประเภทการกระจายย้อนกลับ (backpropagation)

3.3.1 ข่ายงานนิวรัลแบบมีการชี้นำ (Supervised neural network)

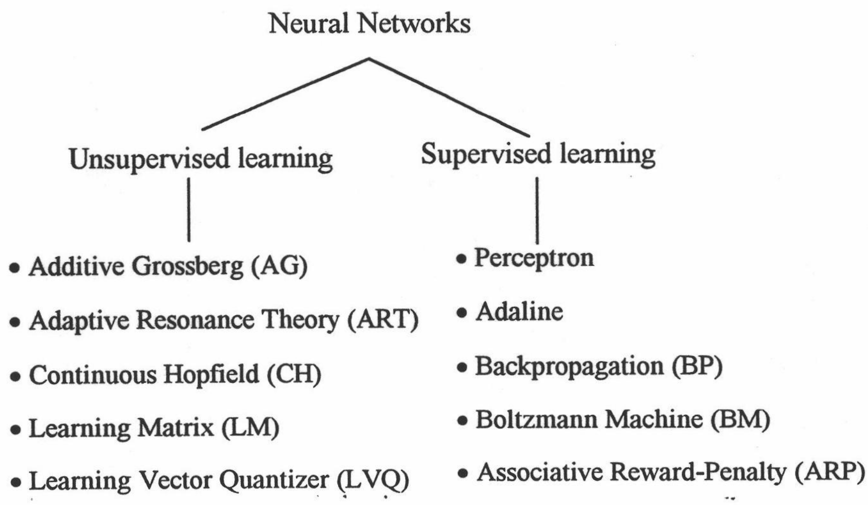
ข่ายงานนิวรัลที่มีการเรียนรู้แบบที่มีการชี้นำถูกฝึกโดยใช้คู่ลำดับของค่าอินพุท และเอาท์พุท ค่าอินพุทใช้เป็นอินพุทให้แก่ข่ายงาน และค่าเอาท์พุทใช้เป็นผู้สอน (teacher) ซึ่งแสดงในรูปที่ 3.8 (a) การฝึกจะทำโดยการปรับค่าน้ำหนักของข่ายงานเพื่อทำให้ค่าความแตกต่างระหว่างเอาท์พุทที่ต้องการและเอาท์พุทของข่ายงานให้มีค่าน้อยที่สุด กระบวนการนี้อาจทำซ้ำจน (iteration) กระทั่งค่าผิดพลาดเป็นค่าน้อย ๆ ที่ยอมรับได้ ตัวอย่างของข่ายงานประเภทนี้ได้แก่เปอร์เซปตรอน (perceptron), อะดาไลน์ (adaline), และการกระจายย้อนกลับ (backpropagation) ซึ่งแสดงในรูปที่ 3.9

3.3.2 ข่ายงานนิวรัลแบบไม่มีการชี้นำ (Unsupervised neural network)

ข่ายงานนิวรัลที่มีการเรียนรู้แบบที่ไม่มีการชี้นำต้องการเพียงค่าอินพุทในการฝึกข่ายงานโดยไม่มีผู้สอนซึ่งแสดงในรูปที่ 3.8 (b) ในระหว่างการฝึกค่าน้ำหนักของข่ายงานถูกปรับเพื่อให้อินพุทที่คล้าย ๆ กันสร้างเอาท์พุทที่คล้ายกัน การเรียนรู้ของข่ายงานประเภทนี้คล้ายกับ



รูปที่ 3.8 การเรียนรู้ของข่ายงานนิวรัลในการปรับค่าน้ำหนัก



รูปที่ 3.9 ข่ายงานนิวรัลสามารถจำแนกเป็นแบบมีการชี้นำ และแบบไม่มีการชี้นำ

สมองของมนุษย์ที่เปลี่ยนแปลงโครงสร้างของมันภายใต้อิทธิพลของประสบการณ์โดยไม่มีผู้สอน ตัวอย่างการเรียนรู้ของข่ายงานนิวรัลประเภทนี้ได้แก่ AG (additive grossberg) และ ART (adaptive resonance theory) ซึ่งแสดงในรูปที่ 3.9

3.4 ข่ายงานนิวรัลแบบมีการชี้นำ

ข่ายงานนิวรัลแบบมีการชี้นำได้แก่ ข่ายงานเปอร์เซปตรอน (perceptron) ข่ายงานแบบอะดาไลน์ (adaline) และข่ายงานแบบกระจายย้อนกลับ ข่ายงานแบบเปอร์เซปตรอน และอะดาไลน์เป็นข่ายงานที่มีชั้นของน้ำหนักเพียงชั้นเดียว ดังนั้นข่ายงานทั้งสองนี้จึงสามารถแก้ปัญหาได้เฉพาะปัญหาแบบเชิงเส้น สำหรับข่ายงานแบบกระจายย้อนกลับเป็นข่ายงานที่มีชั้นของน้ำหนักตั้งแต่ 2 ชั้นขึ้นไป ดังนั้นข่ายงานจึงสามารถแก้ปัญหาแบบไม่เชิงเส้นได้เป็นอย่างดี

ข่ายงานนิวรัลแบบมีการชี้นำจะใช้ชุดข้อมูลที่ประกอบไปด้วยค่าอินพุทและเอาท์พุทที่เป็นคู่ที่สอดคล้องกัน

$$[u,o] = \{[u_1, o_1],[u_2, o_2],\ldots,[u_n, o_n]\} \tag{3.9}$$

เมื่อ n คือจำนวนชุดของข้อมูลที่ใช้ในการฝึก วัตถุประสงค์ในการฝึกข่ายงานคือหาค่าน้ำหนักที่เหมาะสม และทำให้ความแตกต่างของเป้าหมายกับค่าที่ออกจากข่ายงานให้มีค่าน้อยที่สุด ข้อกำหนดของค่าความแตกต่างคือผลรวมของความแตกต่างระหว่างค่าเป้าหมายกับเอาท์พุทของข่ายงานกำลังสอง (sum square error, SSE) ฟังก์ชันของผลต่างกำลังสองคือ

$$E = [t - f(u,w)]^2 \tag{3.10}$$

ฟังก์ชันพลังงาน หรือฟังก์ชันต้นทุน (energy function or cost function) ของระบบ แบบจำลองของข่ายงานเป็นฟังก์ชันของอินพุท และแมทริกซ์ของน้ำหนัก, $y = f(u,w)$ แมทริกซ์ของน้ำหนักสามารถปรับค่าเพื่อให้ฟังก์ชันพลังงานมีค่าต่ำลงให้เหลือน้อยที่สุดไปในทิศทางลดลงของเกรเดียนท์ (gradient)

3.4.1 ข่ายงานแบบเปอร์เซปตรอน (Perceptron)

Rosenblatt (1958) เสนอข่ายงานเปอร์เซปตรอน ข่ายงานเปอร์เซปตรอนเป็นแบบจำลองของข่ายงานนิวรัลแบบตัดสินใจ โดยมีขอบเขตการตัดสินเป็นแบบเชิงเส้น รูปที่ 3.10 แสดงโครงสร้างของข่ายงานเปอร์เซปตรอนซึ่งมีชั้นของน้ำหนักเพียงชั้นเดียว ในการเรียนรู้ข่ายงานอาศัยสัญญาณการเรียนรู้จากค่าความแตกต่างระหว่างค่าเป้าหมายกับค่าเอาท์พุทที่ออกจากข่ายงานนิวรัล ดังนั้นวิธีการเรียนรู้แบบนี้จึงเป็นการเรียนรู้แบบชี้นำ เปอร์เซปตรอนใช้ฟังก์ชันมูลฐานเป็นฟังก์ชันมูลฐานเชิงเส้น และฟังก์ชันกระตุ้นเป็นฟังก์ชันแบบไม่ต่อเนื่องซึ่งได้แก่ฟังก์ชันแบบไบโพลาสเต็พ (bipola step function) ดังนั้นสมการคณิตศาสตร์ของฟังก์ชันกระตุ้นแบบไบโพลาสเต็พคือ

$$f(net) = \begin{cases} 1 & if \quad net > 0 \\ -1 & otherwise \end{cases} \tag{3.11}$$

สัญญาณการเรียนรู้แสดงดังสมการ (3.7)

$$r = d_i - o_i \tag{3.12}$$

เมื่อ $o_i = f(u,w)$ และ d_i คือค่าเป้าหมายโดยการปรับค่าน้ำหนักแสดงดังสมการ (3.13)

$$\Delta w_{ij} = \eta[d_i - f(w_{ij}^T u)]x_j \qquad \text{เมื่อ } j = 1, 2, 3 \ldots n \tag{3.13}$$

การเรียนรู้แบบเปอร์เซปตรอนนำไปประยุกต์ใช้ได้เฉพาะระบบตัวเลขไบนารีเท่านั้น ดังนั้นแบบจำลองนี้จึงสามารถนำไปใช้ในการแยกข้อมูล 2 คลาสออกจากกัน ซึ่งการปรับค่าน้ำหนักจะถูกกระทำก็ต่อเมื่อค่าเอาท์พุทจากข่ายงานนิวรัลไม่ตรงกับค่าเป้าหมาย

3.4.2 ข่ายงานนิวรัลแบบอะดาไลน์ (ADALINE)

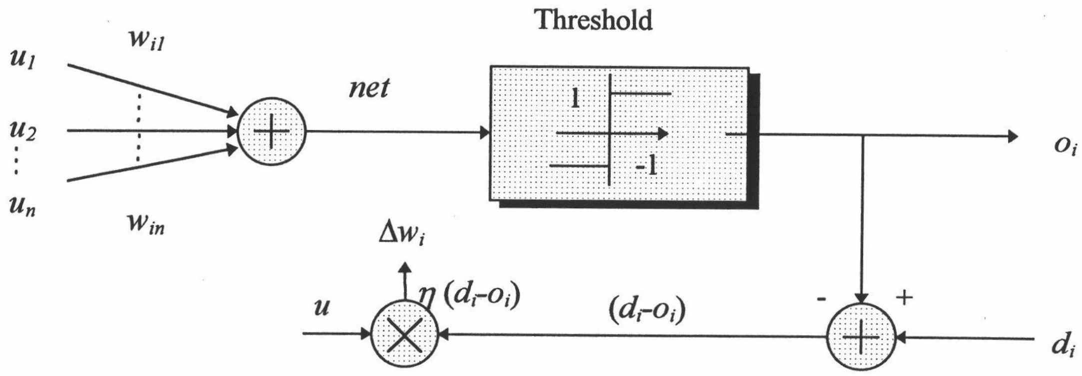
McClelland และ Rumelhart (1986) เสนอแบบจำลองของข่ายงานที่เรียกว่า ADALINE (ADAptive LINear Element) ที่ใช้กฎการเรียนรู้แบบเดลตาในการหาค่าน้ำหนัก

กฎการเรียนรู้แบบเดลตา (delta learning rule) เป็นการเรียนรู้แบบชี้นำเพียงชนิดเดียวที่ใช้กับฟังก์ชันกระตุ้นแบบต่อเนื่องซึ่งได้แก่ฟังก์ชันซิกมอยด์ และฟังก์ชันเกาส์เสียน ฟังก์ชันมูลฐานเป็นฟังก์ชันมูลฐานเชิงเส้น โครงสร้างของการเรียนรู้ของข่ายงานแบบอะดาไลน์มีชั้นของน้ำหนักเพียง 1 ชั้น (ดูรูปที่ 3.11) โดยมีนิยามของสัญญาณการเรียนรู้ (learning signal, *r*) ดังนี้

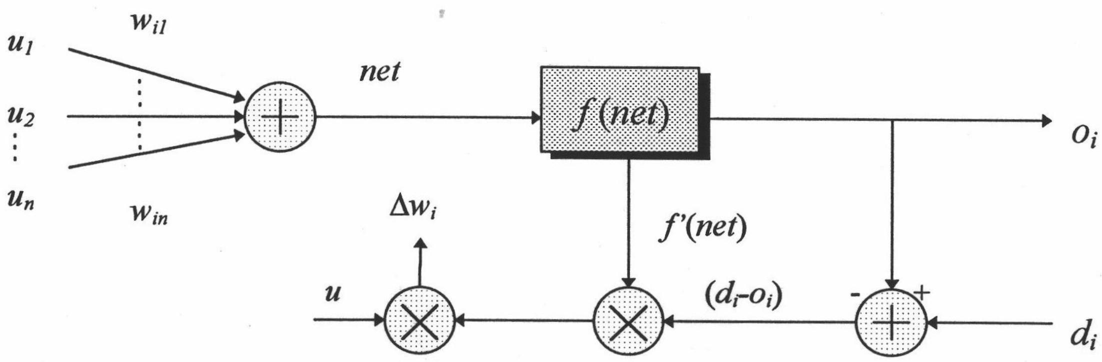
$$r = [d_i - f(net)]f'(net) \tag{3.14}$$

เมื่อ $f'(net)$ เป็นอนุพันธ์ของฟังก์ชันกระตุ้น, $f(net)$ ซึ่งคำนวณจาก $net = w_{ij}u$

กฎการเรียนรู้สามารถหาได้จากค่าผิดพลาดระหว่าง o_i และ d_i ยกกำลังสอง การคำนวณเกรเดียนท์เวคเตอร์ (gradient vector) ได้จากการหาค่าอนุพันธ์ของค่าผิดพลาดเทียบกับ w_i เพื่อปรับค่าน้ำหนักซึ่งการเรียนรู้อยู่บนพื้นฐานการลดค่ากำลังสองของค่าความแตกต่างให้มีค่าน้อยที่สุดโดยมีนิยามดังนี้



รูปที่ 3.10 ข่ายงานแบบเปอร์เซปตรอน



รูปที่ 3.11 ข่ายงานแบบอะดาไลน์

$$\text{minimize} \qquad E = \frac{1}{2}\sum(d_i - o_i)^2 \qquad (3.15)$$

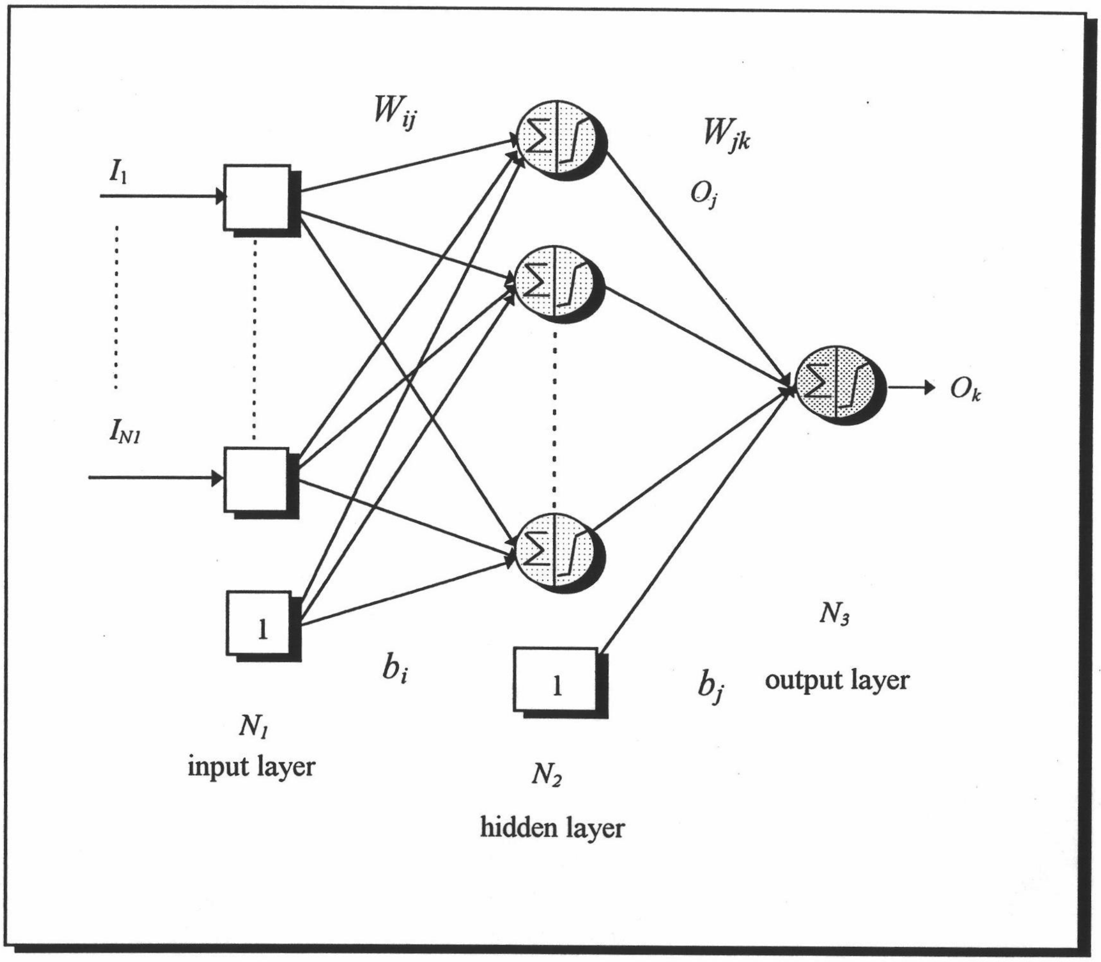
เมื่อเปรียบเทียบการเรียนรู้ของข่ายงานแบบเดลตากับเปอร์เซปตรอนพบว่ากฎการเรียนรู้แบบเดลตาใช้ค่าความแตกต่างของค่าเป้าหมาย, d_i เอาท์พุทของข่ายงาน, o_i ป้อนกลับโดยใช้การออพติไมซ์ แต่การเรียนรู้แบบเปอร์เซปตรอนใช้ค่าความแตกต่างระหว่างค่าเป้าหมายกับค่าเอาท์พุทของข่ายงานนิวรัลโดยตรงเพื่อใช้ในการปรับค่าน้ำหนัก

3. 4.3 ข่ายงานนิวรัลแบบการกระจายย้อนกลับ (Backpropagation neural network)

เนื่องจากข่ายงานนิวรัลชั้นเดียวแบบเปอร์เซปตรอนและ แบบอะดาไลน์มีชั้นของน้ำหนักชั้นเดียว ซึ่งสามารถแก้ปัญหาแบบเชิงเส้นเท่านั้น ต่อมาจึงได้มีการพัฒนาข่ายงานนิวรัลแบบกระจายย้อนกลับซึ่งเป็นข่ายงานแบบหลายชั้น เพื่อแก้ปัญหาแบบไม่เชิงเส้น โดยจำนวนชั้นในข่ายงานนิวรัลหมายถึงชั้นของน้ำหนัก (weight layers) ชั้นน้ำหนักจะเป็นตัวแทนของการเปลี่ยนแปลงแบบเชิงเส้นแต่ชั้นของนิวรอนจะเกิดการเปลี่ยนแปลงแบบไม่เชิงเส้น ในทางปฏิบัติพบว่าชั้นซ่อนเพียงสองชั้นถือว่าเพียงพอต่อการแก้ปัญหาระบบต่าง ๆ ได้ทุกระบบ (สุรพล, 1995) ข่ายงานนิวรัลนี้จะใช้อัลกอริธึมการกระจายย้อนกลับ (backpropagation algorithm) ในการฝึกข่ายงานนิวรัล และใช้ฟังก์ชันมูลฐานเป็นฟังก์ชันเชิงเส้น ฟังก์ชันกระตุ้นเป็นฟังก์ชันซิกมอยด์ และโครงสร้างการเชื่อมโยงเป็นแบบป้อนไปข้างหน้า (feedforward)

3. 4.3.1 โครงสร้างของข่ายงานนิวรัลแบบกระจายย้อนกลับ

ข่ายงานนิวรัลแบบกระจายย้อนกลับแสดงในรูปที่ 3.12 ประกอบไปด้วยหลาย ๆ นิวรอนที่มีการเชื่อมโยงแบบป้อนไปข้างหน้า



รูปที่ 3.12 ข่ายงานนิวรัลป้อนไปข้างหน้าแบบสองชั้น

โครงสร้างของข่ายงานประกอบไปด้วยชั้นของนิวรอนอยู่ 3 ชั้น ได้แก่

1. ชั้นอินพุท (input layer) ทำหน้าที่รับข้อมูลที่ใช้ในการฝึกข่ายงาน

2. ชั้นซ่อน (hidden layer) ทำหน้าที่รวมสัญญาณจากชั้นอินพุท และแปลงค่าโดยใช้ฟังก์ชันกระตุ้น หลังจากนั้นจึงส่งสัญญาณที่ได้ให้แก่นิวรอนในชั้นเอาท์พุท โดยชั้นซ่อนมีได้มากกว่าหนึ่งชั้น

3. ชั้นเอาท์พุท (output layer) ทำหน้าที่รวมสัญญาณจากชั้นซ่อนและแปลงค่าโดยใช้ฟังก์ชันกระตุ้น หลังจากนั้นจึงส่งสัญญาณที่ได้เพื่อเป็นเอาท์พุทที่ออกจากข่ายงาน

ชั้นอินพุทประกอบด้วยนิวรอนที่ไม่มีการคำนวณใด ๆ แต่จะกระจายค่าอินพุทให้แก่ทุก ๆ นิวรอนในชั้นถัดไป ชั้นซ่อนอยู่ระหว่างชั้นอินพุทและชั้นเอาท์พุทซึ่งข่ายงานนิวรัลสามารถมีชั้นซ่อนหลายชั้นแต่จะพิจารณาในที่นี้เพียงชั้นเดียวเท่านั้น เพื่อความง่ายต่อการเข้าใจ และชั้นสุดท้ายคือชั้นเอาท์พุทโดยจะทำหน้าที่ส่งค่าเอาท์พุทสุดท้ายของข่ายงานนิวรัล นิวรอนในชั้นอินพุทรับค่าอินพุทเป็น $I_1,.., I_{N1}$ และมีน้ำหนักเป็น $W_{1j},...,W_{N1j}$

ผลรวมน้ำหนักของอินพุทของนิวรอนลำดับที่ j ในชั้นซ่อนคือ

$$S_j = \sum_{i=1}^{N_1} w_{ij} I_i + b_i \qquad 1 \le j \le N_2 \tag{3.16}$$

เอาท์พุทของนิวรอนลำดับที่ j ในชั้นซ่อนคือ

$$O_j = f(S_j) \qquad 1 \le j \le N_2 \tag{3.17}$$

เอาท์พุทของนิวรอนในชั้นซ่อนใช้เป็นอินพุทให้แก่นิวรอนในชั้นเอาท์พุทอีกทอดหนึ่ง

$$S_k = \sum_{j=1}^{N_2} w_{jk} O_j + b_j \quad 1 \le k \le N_3 \tag{3.18}$$

นิวรอนในชั้นเอาท์พุทสร้างเอาท์พุทของกระบวนการที่ถูกทำนายในช่วงหนึ่งคาบการเก็บตัวอย่าง (one sampling period) ดังแสดงในสมการ (3.19)

$$O_k = f(S_k) \tag{3.19}$$

เมื่อ b_i และ b_j เป็นเทอมไบอัส (bias term) ซึ่งเป็นน้ำหนัก โดยใช้ค่าอินพุทมีค่าเป็น 1 และ O_k หมายถึงค่าเอาท์พุทจากข่ายงานนิวรัล

เมื่อ f แทนด้วยฟังก์ชันกระตุ้น ฟังก์ชันกระตุ้นมักจะเป็นฟังก์ชันซิกมอยด์ (sigmoid function) โดยมีสมการดังนี้คือ

$$f(x) = \frac{1}{1+e^{(-x)}} \tag{3.20}$$

และอนุพันธ์ของฟังก์ชันซิกมอยด์คือ

$$f'(x) = x(1-x) \tag{3.21}$$

ค่าอนุพันธ์ของฟังก์ชันซิกมอยด์จะนำไปใช้ในอัลกอริธึมการกระจายย้อนกลับ ซึ่งจะกล่าวถึงในหัวข้อถัดไป

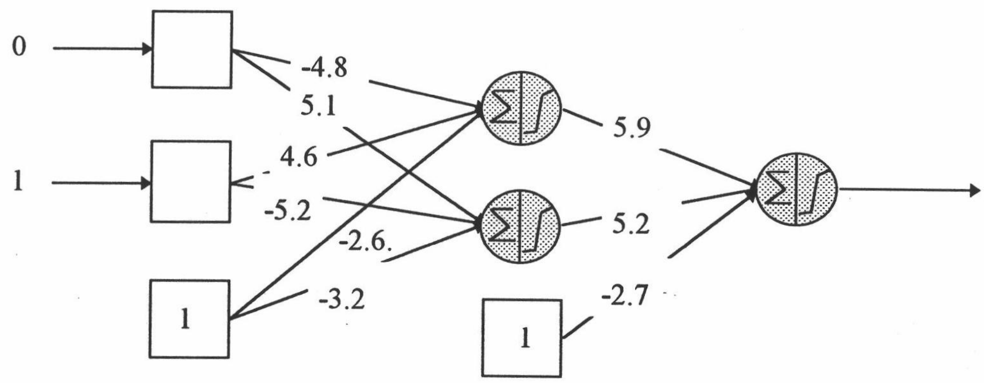
3.4.3.2 ตัวอย่างของการหาเอาท์พุทของข่ายงานแบบกระจายย้อนกลับ

สมมติว่าทราบค่าแมทริกซ์ของน้ำหนักของข่ายงานนี้ซึ่งข่ายงานนี้ได้รับการฝึกโดยใช้อัลกอริธึมการกระจายย้อนกลับเพื่อหาค่าน้ำหนักแล้ว โดยข่ายงานมีโครงสร้างดังในรูปที่ 3.13

ก. การคำนวณค่าเอาท์พุทของนิวรอนในชั้นซ่อนตัวแรก

0.0*-4.8 + 1.0*4.6 - 2.6 = 2.0

กำหนดให้ข่ายงานนี้ใช้ฟังก์ชันกระตุ้นเป็นฟังก์ชันซิกมอยด์ และฟังก์ชันมูลฐานเป็นฟังก์ชันมูลฐานเชิงเส้น จากฟังก์ชันซิกมอยด์ในรูปที่ 3.8 (a) อินพุทเท่ากับ 2.0 จะได้เอาท์พุทของนิวรอนในชั้นซ่อนตัวแรกเป็น 0.88



รูปที่ 3.13 ตัวอย่างการคำนวณค่าเอาท์พุทของข่ายงานนิวรัล

ข. การคำนวณค่าเอาท์พุทของนิวรอนในชั้นซ่อนตัวที่สอง

$$0.0*5.1 + 1.0*-5.2 - 3.2 = -8.4$$

จากฟังก์ชันซิกมอยด์ในรูปที่ 3.8 (a) อินพุทเท่ากับ -8.4 จะได้เอาท์พุทของนิวรอนในชั้นซ่อนตัวที่สองมีค่าเข้าใกล้ 0

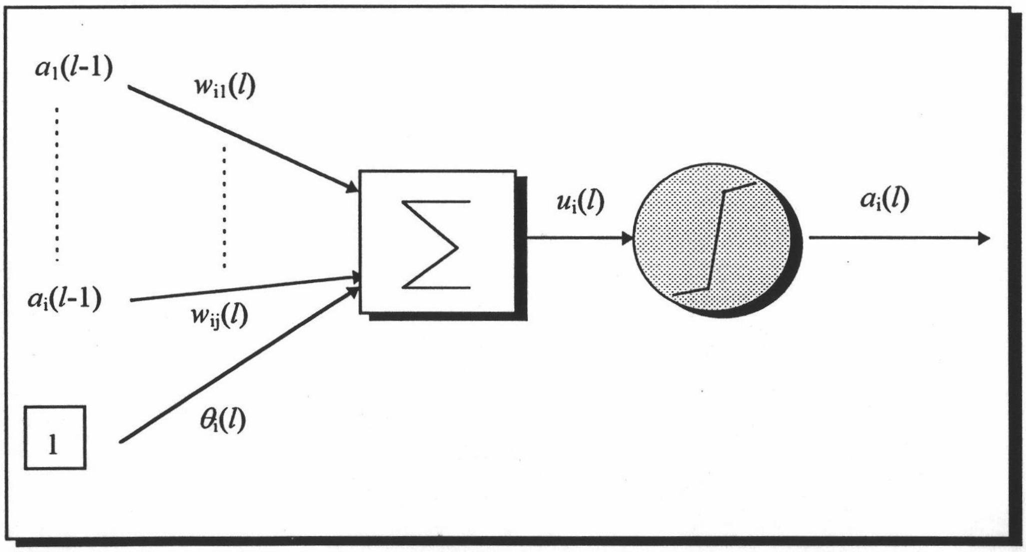
ค. การคำนวณค่าเอาท์พุทของนิวรอนในชั้นเอาท์พุท

$$0.88*5.9 + 0.0*5.2 - 2.7 = 2.5$$

จากฟังก์ชันซิกมอยด์ในรูปที่ 3.8 (a) อินพุทเท่ากับ 2.5 เอาท์พุทของนิวรอนในชั้นเอาท์พุทเป็น 0.92 ดังนั้นข่ายงานในตัวอย่างนี้ให้ค่าเอาท์พุทเป็น 0.92

3.5 อัลกอริธึมการกระจายย้อนกลับ (Backpropagation algorithm)

Werbos (1974) พัฒนาอัลกอริธึมการกระจายย้อนกลับ (backpropagation algorithm) Rumelhart (1986) เสนออัลกอริธึมนี้อีกครั้งทำให้อัลกอริธึมนี้เป็นที่รู้จักกันอย่างกว้างขวาง และเป็นอัลกอริธึมมาตราฐานที่ใช้ในการฝึกข่ายงาน อัลกอริธึมการกระจายย้อนกลับเป็นวิธีที่คำนวณเกรเดียนท์อย่างได้ผล (Kung, 1993) ข่ายงานนิวรัลแบบการกระจายย้อนกลับที่มีฟังก์ชันมูลฐานเป็นเป็นฟังก์ชันมูลฐานเชิงเส้น และมีฟังก์ชันกระตุ้นเป็นฟังก์ชันซิกมอยด์มีสมการทางไดนามิกดังสมการ (3.22) และ (3.23) และแสดงในรูปที่ 3.14

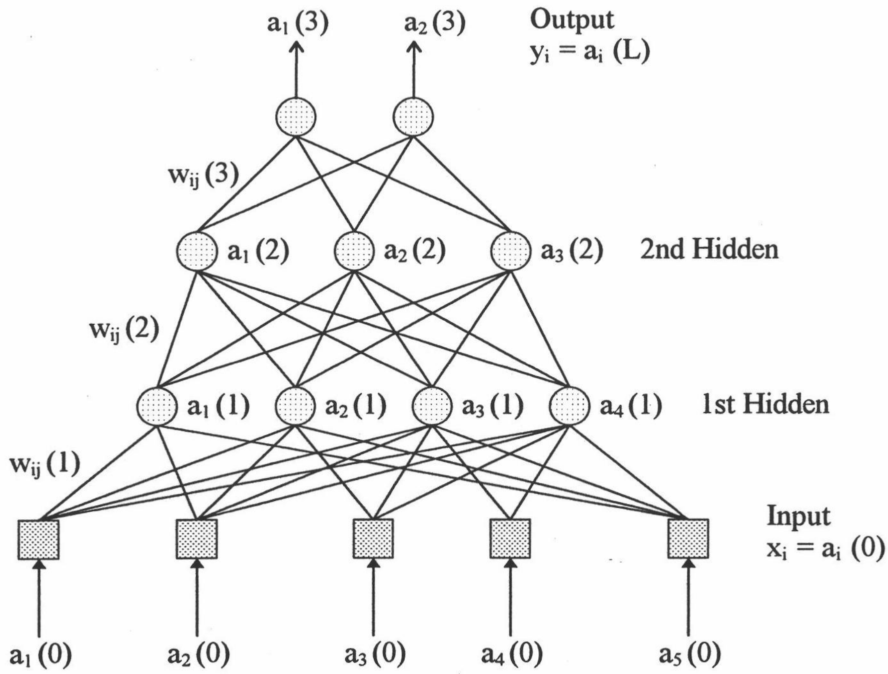


รูปที่ 3.14 แสดงแบบจำลองของนิวรอนในชั้นที่ l

$$u_i(l) = \sum_{j=1}^{N} w_{ij}(l)a_j(l-1) + \theta_i(l) \tag{3.22}$$

$$a_i(l) = f(u_i(l)) \quad 1 \le i \le N_l; \qquad 1 \le l \le L \tag{3.23}$$

รูปที่ 3.15 แสดงข่ายงานนิวรัลสามชั้นแบบไม่เชิงเส้น เมื่ออินพุทคือ $x_i \equiv a_i(0)$ และเอาท์พุทแทนด้วย $y_i \equiv a_i(L)$ เมื่อ L คือจำนวนชั้นทั้งหมดของข่ายงานนิวรัล



รูปที่ 3.15 ข่ายงานนิวรัลแบบหลายชั้น ซึ่งแสดงชั้นของน้ำหนักสามชั้น

3.5.1 วิธีการออพติไมซ์ (Optimization method)

อัลกอริธึมการกระจายย้อนกลับต้องการลดค่าผิดพลาดกำลังสองระหว่างค่าเป้าหมายกับผลลัพธ์จากข่ายงาน โดยใช้วิธีการออพติไมซ์ (optimization method) ซึ่งแสดงในสมการที่ (3.24)

$$\text{minimize } E = \frac{1}{2}\sum_{i=1}^{N_3}(t_i - a_i(L))^2 \qquad (3.24)$$

subject to: $t_{max} > t_i > t_{min}$

$a_{max} > a_i\ (L) > a_{min}$

เมื่อ t_i คือค่าเป้าหมาย $a_i(L)$ คือค่าเอาท์พุทของข่ายงาน t_{max} และ t_{min} คือค่าสูงสุด และต่ำสุดของค่าเป้าหมาย a_{max} และ a_{min} คือค่าสูงสุด และค่าต่ำสุดของเอาท์พุทของข่ายงาน ถ้า

ทราบอนุพันธ์ย่อย (partial derivative) ของค่าผิดพลาดเทียบกับค่าน้ำหนักแต่ละค่าจะทำให้รู้ทิศทางที่น้ำหนักไปในทิศทางที่ลดค่าผิดพลาด สำหรับอัลกอริธึมการกระจายย้อนกลับจะทำการปรับค่าน้ำหนัก (W_{ij}) เพื่อให้ค่า E มีค่าต่ำสุด สมการการเรียนรู้แบบเกรเดียนท์แสดงในสมการ (3.25)

$$w_{ij}^{new}(l) = w_{ij}^{old}(l) + \Delta w_{ij}(l) \tag{3.25}$$

อนุพันธ์ของค่าผิดพลาดจะใช้วิธีการตามเทคนิคของกฎลูกโซ่ (Chain rule)

$$\begin{aligned} \Delta w_i\,(l) &= -\eta \frac{\partial E}{\partial w_i\,(l)} \\ &= -\eta \frac{\partial E}{\partial a_i(l)} \frac{\partial a_i(l)}{\partial w_i\,(l)} \end{aligned} \tag{3.26}$$

จากสมการ (3.26) นำมาแยกพิจารณาทีละเทอมและจากสมการ (3.22) และ (3.23) จะได้

$$\begin{aligned} \frac{\partial a_i(l)}{\partial w_{ij}(l)} &= \frac{\partial f(u_i(l))}{\partial w_{ij}(l)} \\ &= \frac{\partial f(\sum_{j=1}^{N} w_{ij}(l) a_j(l-1) + \theta_i(l))}{\partial w_{ij}(l)} \\ &= f'(u_i(l)) a_j(l-1) \end{aligned} \tag{3.27}$$

กำหนดให้สัญญาณความผิดพลาด, $\delta_i(l)$ เป็น

$$\delta_i(l) = -\frac{\partial E}{\partial a_i(l)} \tag{3.28}$$

สมการ (3.27) และ (3.28) แทนในสมการ (3.26) จะได้สมการ (3.29)

$$\Delta w_{ij}(l) = \eta \delta_i(l) f'(u_i(l)\ a_i(l-1) \tag{3.29}$$

ก. ค่าสัญญาณความแตกต่างในชั้นเอาท์พุทสามารถหาได้จากการแทนค่า E ลงในสมการ (3.28)

$$\delta_i(L) = -\frac{\partial E}{\partial a_i(L)}$$

$$= -\frac{\partial(\frac{1}{2}\sum_{i=1}^{N}(t_i - a_i(L))^2)}{\partial a_i(L)}$$

$$= -\frac{1}{2}(2)(t_i - a_i(L))(-1)$$

$$\delta_i(L) = t_i - a_i(L) \tag{3.30}$$

ข. ค่าสัญญาณความแตกต่างในชั้นซ่อนคือ

$$\delta_i(l) = -\frac{\partial E}{\partial a_i(l)}$$

$$= -\sum_{j=1}^{N_{L+1}} \frac{\partial E}{\partial u_i(l+1)} \frac{\partial u_i(l+1)}{\partial a_i(l)} \tag{3.31}$$

สมการ (3.31) นำมาแยกพิจารณาทีละเทอมโดยพิจารณาเทอมแรกจะได้

$$-\frac{\partial E}{\partial u_i(l+1)} = -\frac{\partial(\frac{1}{2}(t_i - a_i(L))^2)}{\partial u_i(l+1)}$$

$$= (t_i - a_i(L))\frac{\partial a_i(L)}{\partial u_i(l+1)}$$

สมการ (3.23) แทนค่าลงใน $a_i(L)$ จะได้

$$-\frac{\partial E}{\partial u_i(l+1)} = (t_i - a_i(L))\frac{\partial f(u_i(L))}{\partial u_i(l+1)}$$

ในกรณีนี้ $L = l+1$ ดังนั้น

$$-\frac{\partial E}{\partial u_i(l+1)} = (t_i - a_i(L)) f'(u_i(l+1))$$

สมการ (3.30) แทนค่า (t_i - a_i (L)) จะได้

$$-\frac{\partial E}{\partial u_i(l+1)} = \delta_i(l+1) f'(u_i(l+1)) \tag{3.32}$$

สมการ (3.31) เทอมที่ 2 ที่ได้จากการแทนสมการ (3.22) คือ

$$\frac{\partial u_i(l+1)}{\partial a_i(l)} = \frac{\partial(w_{ij}(l+1)a_i(l) + \theta_i(l+1))}{\partial a_i(l)}$$

$$= w_{ij}(l+1) \tag{3.33}$$

สมการ (3.32) และ (3.33) แทนในสมการ (3.31) โดย $l = L$ - 1,.., 1 จะได้สัญญาณความแตกต่างในชั้นซ่อนคือ

$$\delta_i(l) = \sum_{j=1}^{N_{l+1}} \delta_i(l+1) f'(u_i(l+1)) w_{ij}(l+1) \tag{3.34}$$

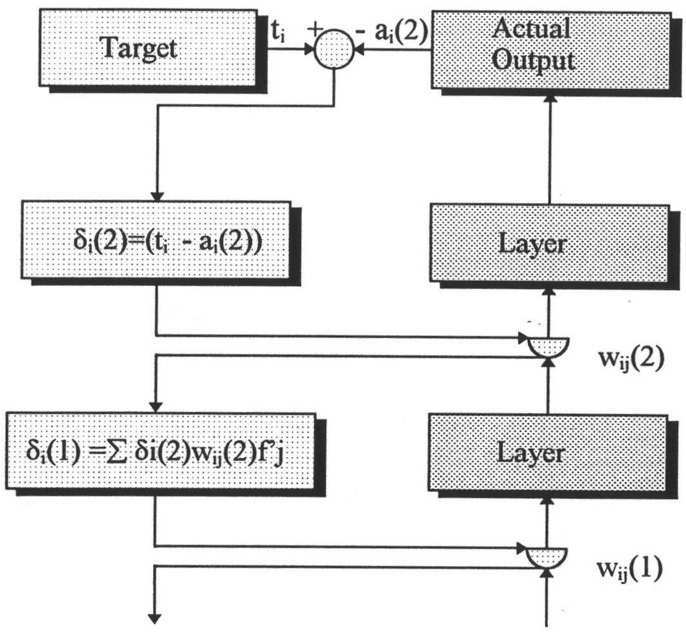
ถ้ามีชั้นซ่อนจำนวน n ชั้นต้องคำนวณหาค่าสัญญาณความแตกต่างในสมการ (3.34) จำนวน n ครั้งเพื่อนำไปใช้เป็นแฟคเตอร์ในการปรับค่าน้ำหนัก

แทนค่าสมการ (3.29 ลงในสมการ (3.25) ค่าน้ำหนักเชื่อมโยงระหว่างชั้นที่ l กับชั้นที่ l - 1 สามารถปรับค่าโดยความสัมพันธ์ดังนี้ (ตามลำดับของ $l = L$-1,..,1)

$$w_{ij}^{new}(l) = w_{ij}^{old}(l) + \eta\delta_i(l) f'(u_i(l)) a_i(l-1) \tag{3.35}$$

สำหรับการเรียนรู้ที่ใช้อัลกอริธึมการกระจายย้อนกลับสัญญาณความแตกต่างของชั้นที่ต่ำกว่า δ_i (l) สามารถคำนวณได้โดยการรวมสัญญาณความแตกต่างของชั้นที่อยู่เหนือกว่า, $\delta_i(l+1)$ ดัง

นั้นสัญญาณความแตกต่างจึงสามารถกระจายย้อนกลับผ่านชั้นทั้งหมดจากชั้นบนลงสู่ชั้นล่าง ซึ่งหมายถึงว่าชั้นอยู่เหนือกว่าจะมีอิทธิพลต่อชั้นที่อยู่ต่ำกว่าซึ่งแสดงในรูปที่ 3.16



รูปที่ 3.16 อัลกอริธึมการกระจายย้อนกลับ (backpropagation) สำหรับข่ายงานนิวรัลแบบสองชั้น

3.5.2 ขั้นตอนของอัลกอริธึมการกระจายย้อนกลับ

อัลกอริธึมการกระจายย้อนกลับสำหรับข่ายงานนิวรัลแบบสองชั้น ซึ่งแสดงในรูปที่ 3.12 ต้องกำหนดจำนวนคู่ลำดับของอินพุท, u_i และค่าเป้าหมาย, t_i จำนวน p คู่

$$\{(u_1, t_1), (u_2, t_2), (u_3, t_3), \ldots\ldots\ldots, (u_p, t_p)\} \tag{3.36}$$

ก. ขั้นตอนที่ 1 กำหนดค่าอัตราการเรียนรู้ (learning rate, η) ให้มีค่ามากกว่าศูนย์ กำหนดค่าผิดพลาดต่ำสุด (E_{min}) และกำหนดค่าเริ่มต้นให้กับน้ำหนัก (weight) ที่เชื่อมต่อระหว่างนิวรอนแต่ละชั้น W_{ij} (1), W_{ij} (2) โดยใช้ตัวเลขสุ่มค่าน้ำหนักที่มีค่าน้อย ๆ กำหนดข้อมูลคู่แรกเข้ามา $p = 1$ ค่าผิดพลาดเริ่มต้น $E = 0$ และเริ่มการเรียนรู้ $q = 1$

ข. ขั้นตอนที่ 2 ส่งอินพุทและ ค่าเป้าหมายที่สอดคล้องกันให้นิวรอนของชั้นอินพุทเพื่อรับข้อมูลซึ่งคือ $u = u_p$, $t = t_p$ และคำนวณค่าผลลัพธ์ให้นิวรอนในชั้นซ่อน และ ชั้นเอาท์พุทโดยใช้ฟังก์ชันซิกมอยด์เป็นฟังก์ชันกระตุ้น

ค. ขั้นตอนที่ 3 คำนวณค่าผิดพลาด

$$E^{new} = E^{old} + \frac{1}{2}\sum_{i=1}^{N_3}\{t_i - a_i(2)\}^2 \qquad (3.37)$$

เมื่อ N_3 คือจำนวนนิวรอนในชั้นเอาท์พุท t_i คือค่าเป้าหมายของนิวรอนในชั้นเอาท์พุทตัวที่ i และ o_i คือเอาท์พุทของนิวรอนในชั้นเอาท์พุทตัวที่ i

ง. ขั้นตอนที่ 4 คำนวณค่าเวคเตอร์สัญญาณความผิดพลาด

สัญญาณความผิดพลาดสำหรับชั้นเอาท์พุท

$$\delta_i(2) = (t_i - a_i(2)) \qquad \text{เมื่อ } i = 1, 2, \ldots, N_3 \qquad (3.38)$$

สัญญาณความผิดพลาดสำหรับชั้นซ่อน

$$\delta_i(1) = (\sum_{j=1}^{N_3}\delta_i(2)W_{ij}(2)a_i(1)(1 - a_i(1)) \qquad \text{เมื่อ } i = 1, 2, .., N_2 \qquad (3.39)$$

ในกรณีที่มีจำนวนชั้นซ่อน n ชั้นต้องหาค่าสัญญาณความผิดพลาด n ครั้งด้วย

จ. ขั้นตอนที่ 5 ปรับค่าน้ำหนักที่เชื่อมระหว่างชั้นเอาท์พุทกับชั้นซ่อน

$$W_{ij}(2)^{new} = W_{ij}(2)^{old} + \eta\delta_i(2)a_i(1) \qquad (3.40)$$

เมื่อ i=1, 2,..., N_3 และ j=1, 2,..., N_2

ฉ. ขั้นตอนที่ 6 ปรับค่าน้ำหนักที่เชื่อมระหว่างชั้นอินพุทกับชั้นซ่อน

$$W_{ij}(1)^{\text{new}} = W_{ij}(1)^{old} + \eta\delta_i(1)a_i(0) \tag{3.41}$$

เมื่อ j=1, 2,..., N_2 และ i=1, 2,..., N_1

ช.ขั้นตอนที่ 7 ตรวจสอบชุดข้อมูลการเรียนรู้

ถ้าชุดข้อมูลยังไม่ถูกทำงานจนครบรอบ ($p< P$) ให้ส่งข้อมูลของชุดถัดไป ไปยังนิวรอนในชั้นอินพุท โดยที่ $p = p + 1$ และ $q = q + 1$ แล้วจึงกลับไปทำขั้นตอนที่ 2 แต่ถ้าชุดข้อมูลได้ถูกเรียนรู้จนครบแล้ว ($p = P$) ให้ไปดำเนินการขั้นตอนที่ 8

ซ. ขั้นตอนที่ 8 ตรวจสอบค่าความผิดพลาดกับค่าต่ำสุดที่ตั้งไว้

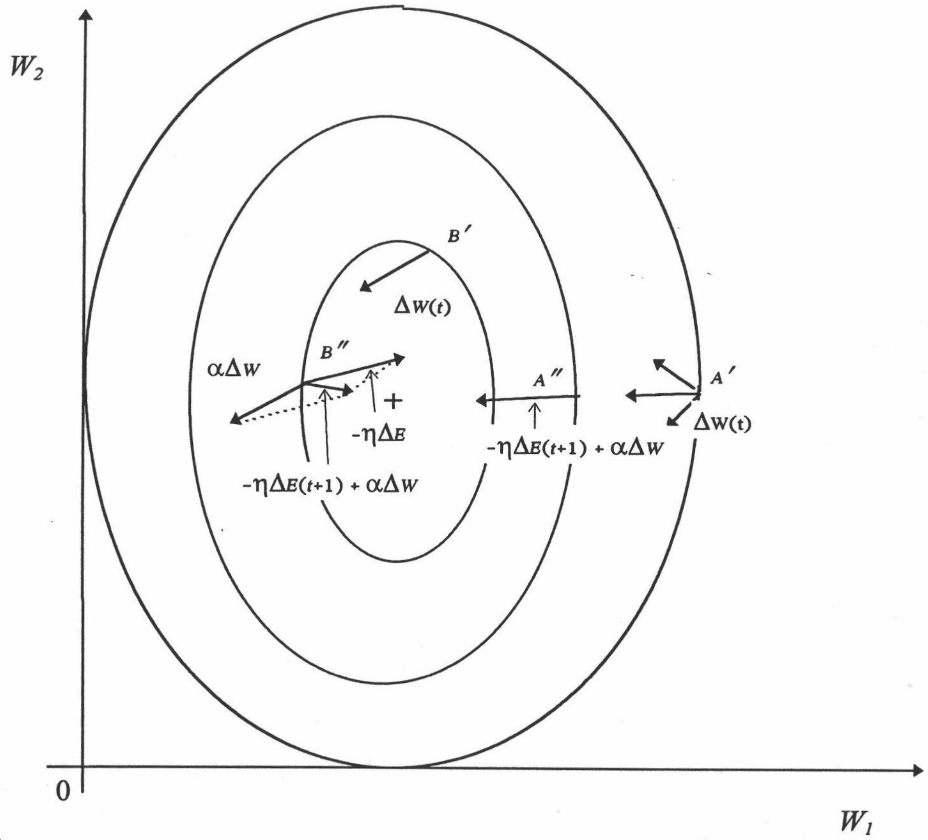
ขั้นตอนการเรียนรู้จะเสร็จสมบูรณ์เมื่อ ค่าความผิดพลาดในการเรียนรู้มีค่าน้อยกว่าค่าความผิดพลาดต่ำสุดที่ตั้งไว้ ($E < E_{min}$) ถ้า $E > E_{min}$ แล้วให้ $E = 0$ และ $p = 1$ แล้วเริ่มวงจรการเรียนรู้ใหม่ตามขั้นตอนที่ 2 ในรูปที่ 3.16 เป็นไดอะแกรมของอัลกอริธึมการกระจายย้อนกลับ

3.6 การเรียนรู้โดยการใช้โมเมนตัม

การเรียนรู้โดยการใช้โมเมนตัมช่วยเร่งให้การเรียนรู้ของข่ายงานเข้าสู่ค่าเป้าหมายได้เร็วขึ้น โดยการนำค่าน้ำหนักล่าสุดมาใช้ในการปรับหาค่าน้ำหนักในปัจจุบันตามขั้นตอนที่ 5 และขั้นตอนที่ 6 ในหัวข้อ 3.5.2 ตามสัดส่วนของค่าโมเมนตัมดังสมการ (3.42)

$$\Delta w(t) = -\eta\nabla E(t) + \alpha\Delta w(t\text{-}1) \tag{3.42}$$

เมื่อ α คือ แฟคเตอร์โมเมนตัมที่มักเลือกค่าอยู่ระหว่าง 0 - 1 รูปที่ 3.17 แสดงวิธีการเร่งการเรียนรู้ของข่ายงานโดยการใช้โมเมนตัม และการเรียนรู้เริ่มต้นที่จุด A' ค่าเกรเดียนท์ของค่า



รูปที่ 3.17 การเรียนรู้แบบการกระจายความผิดพลาดย้อนกลับที่มีการรวมค่าแฟคเตอร์โมเมนตัมมาใช้ในการปรับค่าน้ำหนัก

ความผิดพลาดเมื่อเทียบกับค่าน้ำหนัก $\partial E/\partial w_1$ และ $\partial E/\partial w_2$ ที่จุด A', และ A'' มีเครื่องหมายตรงกัน ดังนั้นองค์ประกอบเกรเดียนท์จะถูกนำมารวมกันซึ่งทำให้การเข้าสู่เป้าหมายรวดเร็วขึ้น โดยการปรับน้ำหนักที่จุด A'' ซึ่งอาศัยค่าน้ำหนักที่จุด A' เข้ามาร่วมด้วยตามสัดส่วนของค่าโมเมนตัม

3.7 บทสรุป

ในบทนี้เป็นการกล่าวถึงทฤษฎีทั่ว ๆ ไปของข่ายงานนิวรัล อัลกอริธึมการกระจายย้อนกลับ และการเรียนรู้โดยการใช้โมเมนตัม บทที่ 4 จะเป็นเนื้อหาเกี่ยวกับการนำข่ายงานนิวรัลไปประยุกต์ใช้กับวิศวกรรมเคมี โดยพิจารณาถึงโครงสร้างการเรียนรู้ โครงสร้างการควบคุม และการระบุกระบวนการ