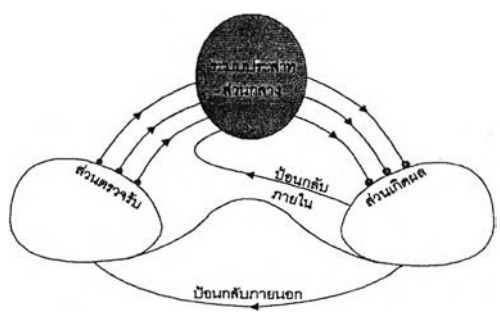
บทที่ 3

วิธีการพัฒนาแบบจำลองของโครงข่ายประสาทเทียม

อาทิฟิเชียลนิวรอลเน็ตเวิร์ก (Aritificial neural network) หรือโครงข่ายประสาทเทียมเป็นระบบประมวลผลที่เลียนแบบและพัฒนามาจากลักษณะทั่วไปทางชีววิทยาของระบบสมอง ซึ่งได้รับความสนใจในการนำมาใช้แก้ปัญหาความไม่เป็นเชิงเส้นที่ยุ่งยากและสลับซับซ้อน

3.1 เครือข่ายประสาทธรรมชาติ [15,16]

สมองมนุษย์ประกอบด้วยหน่วยการประมวลผล (Computing or processing elements or elementary nerve cell) หรือเรียกว่านิวรอล (Neuron) ประมาณ 10^{11} หน่วย แต่ละนิวรอลเชื่อมโยง (Synapse or Connection) กันเป็นเครือข่าย (Network) จำนวนการเชื่อมโยงมีประมาณ $6x10^{15}$ นั่นคือ มีความหนาแน่นของการเชื่อมโยงประมาณ 10^{4} ต่อหนึ่งนิวรอล การทำงานของสมองมนุษย์มีความซับซ้อนไม่เป็นเชิงเส้นและเป็นแบบขนาน (Parallel) การทำงานแต่ละครั้งของสมองมนุษย์จะใช้เวลาอยู่ในช่วงมิลลิวินาที (10^{-3} s) ระบบประสาทของมนุษย์มีการส่งผ่านข้อมูลที่มีการเปลี่ยนรูประหว่างสัญญาณไฟฟ้ากับสัญญาณเคมี อาจพิจารณาว่าสมองเป็นเสมือนเครือข่ายการปิดเปิดทางไฟฟ้า (Electrical switching network) ที่ถูกควบคุมด้วยกระบวนการทางชีวเคมี การทำงานของร่างกายมนุษย์ประกอบด้วย 3 ส่วนหลัก คือ ส่วนตรวจรับ (Receptors or sensory units), ระบบประสาทส่วนกลาง (Central nerve system) และส่วนเกิดผล (Effector) ดังแสดงในรูปที่ 3.1

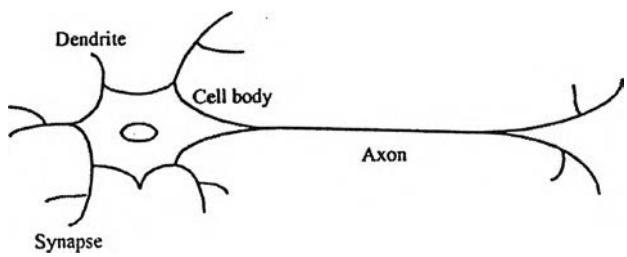


รูปที่ 3.1 องค์ประกอบของเครือข่ายประสาทของสมองมนุษย์

จากรูปที่ 3.1 จะพบว่าส่วนตรวจรับจะรับข้อมูลป้อนเข้าทั้งจากภายนอกและภายในระบบ จากนั้นจะส่งข้อมูลในรูปสัญญาณไฟฟ้าแบบอิมพัลส์ (Electrical impulse) เข้าระบบของประสาทส่วนกลางต่อไป ผลลัพธ์ที่ได้จากการประมวลผลของระบบประสาทส่วนกลางจะส่งไปควบคุมการทำงานของส่วนเกิดผล

3.2 **เซลล์ประสาทธรรมชาติ** [15,16]

เซลล์ประสาท (Neuron) ประกอบด้วยองค์ประกอบหลัก 3 ส่วน คือ โซมา (Soma) หรือตัวเซลล์ (Cell body), เดนไดรท์ (Dendrite), แอกซอน (Axon) แสดงดังรูปที่ 3.2 โดยที่เดนไดรท์คือส่วนที่มีกิ่งก้านคล้ายต้นไม้ทำหน้าที่เป็นส่วนรับข้อมูลและส่งให้กับโซมา ซึ่งแต่ละกิ่งจะรับรู้ได้ด้วยค่าถ่วงน้ำหนัก (Weight) ที่แตกต่างกัน แอกซอนคือส่วนที่มีรูปร่างเป็นทรงกระบอกยาวทำหน้าที่นำข้อมูลออกจากโซมา โซมาคือส่วนที่รวมข้อมูลป้อนเข้าทั้งหมดเข้าด้วยกันและประมวลผลในการเชื่อมนิวรอลเข้าด้วยกันเป็นระบบเครือข่ายยังมีองค์ประกอบหนึ่งที่อยู่ระหว่างแอกซอนของนิวรอลหนึ่งกับเดนไดรท์ของนิวรอลที่อยู่ข้างเคียง (Neightboring neural) เรียกว่าไซแนปส์ (Synapse)



รูปที่ 3.2 เซลล์ประสาทอย่างง่าย

คุณสมบัติของนิวรอลธรรมชาติอาจจะสรุปได้ดังนี้

1. แต่ละนิวรอลจะรับข้อมูลป้อนเข้าเป็นจำนวนมาก
2. ข้อมูลป้อนเข้าอาจถูกปรับให้เหมาะสมด้วยค่าการเชื่อมโยง (Connection value or weight value) ที่ไซแนป
3. โซมาจะรวมข้อมูลป้อนเข้าที่ผ่านการปรับด้วยค่าการเชื่อมโยงเข้าด้วยกันและทำการประมวลผล
4. ภายใต้ภาวะที่เหมาะสม เช่น ข้อมูลป้อนเข้ามีจำนวนเพียงพอ นิวรอลจะส่งข้อมูลออกมาหนึ่งค่า

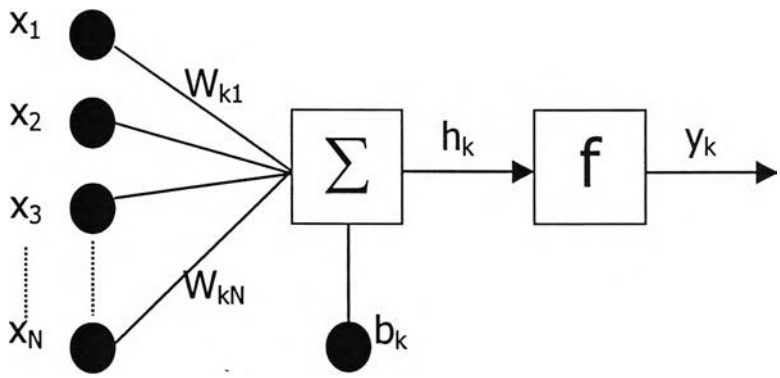
5. ข้อมูลส่งออกจากนิวรอลหนึ่งอาจจะถูกส่งไปให้นิวรอลอื่น ๆ หลายนิวรอลผ่านทางแอกซอน
6. นิวรอลแต่ละนิวรอลมีความเป็นเฉพาะตำแหน่งหรือเฉพาะที่ (Local)
7. ความจำแบ่งออกเป็น 2 ส่วน คือ
    - ความจำระยะยาว (Long-term) จะเก็บที่ไซแนปของนิวรอลหรือเก็บในค่าการเชื่อมโยง
    - ความจำระยะสั้น (Short-term) จะสัมพันธ์กับข้อมูลส่งออกจากนิวรอล
8. ความถูกต้องของค่าการเชื่อมโยงจะขึ้นกับประสบการณ์ในการเรียนรู้ที่ผ่านมา
9. การควบคุมการถ่ายโอนของไซแนปอาจจะเป็นการกระตุ้นแบบเสริมหรือแบบหักล้าง

3.3 เครือข่ายประสาทเทียม

เครือข่ายประสาทเทียม อาจพิจารณาให้เป็นขั้นตอนวิธีทางคณิตศาสตร์ (Mathematical algorithm) ของเครือข่ายธรรมชาติ โดยตั้งอยู่บนพื้นฐานดังนี้

1. หน่วยประมวลผลเรียกว่า นิวรอล (neuron) การส่งผ่านข้อมูลระหว่างนิวรอลทำได้โดยการเชื่อมโยงระหว่างนิวรอล
2. การเชื่อมโยงแต่ละส่วนจะขึ้นอยู่กับค่าการเชื่อมโยงหรือค่าน้ำหนัก (weight)
3. ข้อมูลส่งออกถูกกำหนดด้วยฟังก์ชันแอคติเวชัน (Activation function หรือ Transfer function) ซึ่งปกติความสัมพันธ์ระหว่างผลรวมของข้อมูลป้อนเข้าที่คูณด้วยค่าการเชื่อมโยงกับข้อมูลส่งออกเป็นความสัมพันธ์แบบไม่เป็นเชิงเส้น (Non-linear)

จากกลไกการทำงานของเซลล์ประสาทธรรมชาติ สามารถสร้างแบบจำลองของเซลล์ประสาทได้ ซึ่งเป็นแบบจำลองทางคณิตศาสตร์ที่ประดิษฐ์ขึ้นมาโดยทั่วไปเรียกว่าเซลล์ประสาทเทียม (Artificial neuron) ได้ดังรูปที่ 3.3



รูปที่ 3.3 เซลล์ประสาทเทียมอย่างง่าย

ลักษณะของเซลล์ประสาทธรรมชาติและเซลล์ประสาทเทียม สามารถเปรียบเทียบส่วนต่าง ๆ ได้ดังตารางที่ 3.1

ตารางที่ 3.1 เปรียบเทียบระหว่างนิวรอลธรรมชาติและแบบจำลองนิวรอล

ลำดับ	นิวรอลธรรมชาติ	แบบจำลองนิวรอล
1.	โซมา	ส่วนประมวลผลผล (ตัวรวมและฟังก์ชันแอคติเวชัน)
2.	เดนไดรท์	ข้อมูลป้อนเข้า
3.	แอกซอน	ข้อมูลส่งออก
4.	ไซแนป	ค่าการเชื่อมโยง
5.	ความเร็วในการประมวลผลต่ำ	ความเร็วในการประมวลผลสูง
6.	เครือข่ายประกอบด้วยนิวรอลจำนวนมาก	เครือข่ายประกอบด้วยนิวรอลจำนวนน้อย

จากรูปที่ 3.3 ตัวแปรด้านเข้า x_i โดยที่ $i = 1,\ 2,\ 3,...,\ N$ จะถูกส่งผ่านไปรวมกันด้วยค่าถ่วงน้ำหนักที่ไม่เท่ากัน จะได้ผลรวมเป็น

$$\begin{aligned} h_k &= \sum_{i=1}^{N} w_{ki} x_i + b_k \\ &= w_{k1}x_1 + w_{k2}x_2 + ... + w_{kN}x_N + b_k \end{aligned} \qquad (3.1)$$

จากนั้นส่งสัญญาณ h_k ผ่านแอคติเวชันฟังก์ชัน (Activation function) เพื่อคำนวณค่าตัวแปรด้านออก y_k

$$y_k = f\left(\sum_{i=1}^{N} w_{ki} x_i + b_k\right) \tag{3.2}$$

โดยที่ h_k คือ ค่าผลรวมของตัวแปรด้านเข้าของนิวรอลที่ k

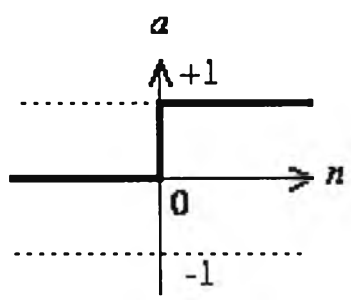
w_{ki} คือ ค่าถ่วงน้ำหนักระหว่างนิวรอลที่ i และ k

b_k คือ ค่าไบอัส (bias) ของนิวรอลที่ k

3.4 แอคติเวชันฟังก์ชัน

แอคติเวชันฟังก์ชันเป็นตัวกำหนดค่าข้อมูลส่งออก แบ่งได้หลายชนิด เช่น

3.4.1 **ฟังก์ชันเทรสโฮลด์** (Threshold function **หรือ** Hard limit function **หรือ** Step function) ตัวอย่างของฟังก์ชันนี้แสดงดังรูปที่ 3.4

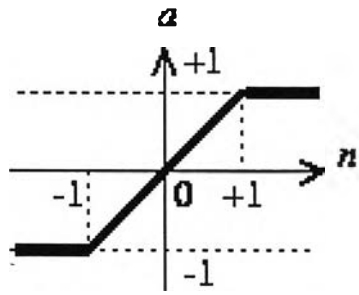


รูปที่ 3.4 ฟังก์ชันเทรสโฮลด์

โดยมีความสัมพันธ์ดังสมการ

$$g(v) = \begin{cases} 1; v \geq 0 \\ 0; v < 0 \end{cases} \tag{3.3}$$

3.4.2 **ฟังก์ชันเชิงเส้นอิ่มตัว** (Saturating Linear function) ตัวอย่างของฟังก์ชันนี้แสดงดังรูปที่ 3.5



รูปที่ 3.5 ฟังก์ชันเชิงเส้นอิ่มตัว

โดยมีความสัมพันธ์ดังสมการ

$$g(v)=\begin{cases}1; v \geq 0\\ av; -b < v < b\\ 0; v \leq 0\end{cases} \qquad (3.4)$$

โดยที่ a คือ ค่าความชัน (Slope) ในช่วงความเป็นเชิงเส้น

b คือ ช่วงที่เป็นเชิงเส้นของฟังก์ชันเชิงเส้นอิ่มตัว

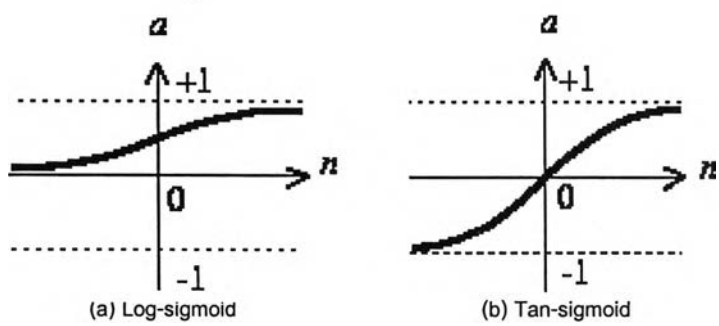
จากฟังก์ชันเชิงเส้นอิ่มตัวจะสังเกตเห็นได้ว่า เมื่อไม่มีช่วงอิ่มตัว ($b=\infty$) จะได้ฟังก์ชันเชิงเส้น นั่นคือ $g(v)=av$ และฟังก์ชันเชิงเส้นอิ่มตัวจะเปลี่ยนรูปไปเป็นฟังก์ชันเทรสโฮลด์ได้ถ้าค่าความชันเท่ากับอนันต $(a=\infty)$

3.4.3 **ฟังก์ชันซิกมอยด์** (Sigmoid function) เป็นฟังก์ชันที่ได้รับความนิยมนำมาใช้เป็นแอคติเวชันฟังก์ชันของเครือข่ายประสาทเทียมกันมากเนื่องจากเป็นฟังก์ชันที่มีความต่อเนื่องมีลักษณะสอดคล้องกับธรรมชาติของการตอบสนองของหลายระบบ ฟังก์ชันซิกมอยด์จัดเป็นแอคติเวชันฟังก์ชันที่ไม่เป็นเชิงเส้น ตัวอย่างของฟังก์ชันซิกมอยด์ คือ ฟังก์ชันลอจิสติก (Logistic function) และฟังก์ชันแทนเจนท์ไฮเพอร์โบลิก (Hyperbolic tangent function) ตัวอย่างของฟังก์ชันนี้แสดงดังรูปที่ 3.6 ซึ่งมีความสัมพันธ์ดังสมการที่ 3.5 และสมการที่ 3.6

ฟังก์ชันลอจิสติก $$g(v)=\frac{1}{1+\exp(-\lambda v)} \quad (3.5)$$

โดยที่ λ คือ ค่าความชัน

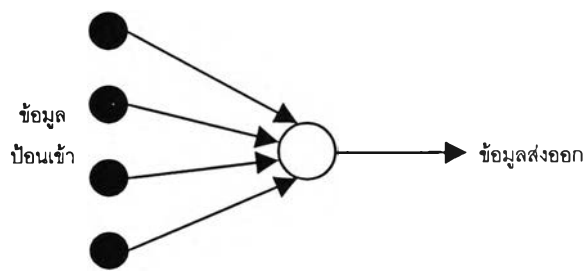
ฟังก์ชันแทนเจนท์ไฮเพอร์โบลิก $$g(v)=\tanh\left(\frac{v}{2}\right)=\frac{1-\exp(-v)}{1+\exp(-v)} \quad (3.6)$$



รูปที่ 3.6 ฟังก์ชันลอจิสติกที่ค่าความชันต่าง ๆ

3.5 สถาปัตยกรรมของเครือข่ายประสาท (Neural network architecture)

การออกแบบโครงสร้างของเครือข่ายประสาทจะเกี่ยวกับเส้นทางการเชื่อมโยง ทิศทางการส่งผ่านข้อมูลในเส้นทาง จำนวนชั้นทั้งหมด จำนวนนิวรอลทั้งหมดที่ประกอบเป็นเครือข่ายประสาท และจำนวนนิวรอลต่อชั้น เป็นต้น เพื่อให้โครงสร้างของเครือข่ายประสาทดูได้ง่ายขึ้นจึงใช้แบบจำลองนิวรอลอย่างย่อดังรูปที่ 3.7 แทนแบบจำลองในรูปที่ 3.3

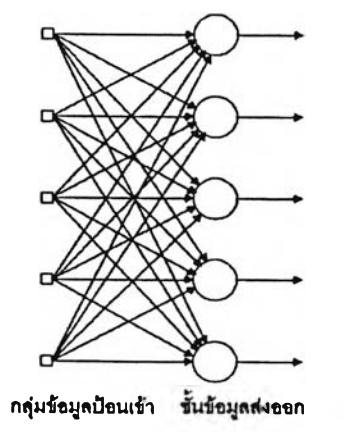


รูปที่ 3.7 แบบจำลองนิวรอลอย่างง่าย

จากรูปแบบจำลองนิวรอลขนาดย่อจะเห็นว่ารวมเอาส่วนแอคติเวชันฟังก์ชันเข้ากับส่วนตัวบวกซึ่งเรียกว่าส่วนประมวลผล (Processing part or Computation part) และในแบบจำลองขนาดย่อนี้จะไม่แสดงค่าเทรสโฮลด์ อาจจะแบ่งโครงสร้างพื้นฐานของเครือข่ายประสาทออกได้ คือ

3.5.1 **เครือข่ายป้อนสู่ด้านหน้าชั้นเดียว** (Single-layer feedforward networks)

เครือข่ายป้อนสู่ด้านหน้าชั้นเดียว จัดเป็นแบบของโครงสร้างเครือข่ายประสาทที่ซับซ้อนน้อยที่สุด ชั้นของเครือข่ายประสาทเทียมจะประกอบด้วยกลุ่มของนิวรอล การส่งผ่านข้อมูลกระทำจากกลุ่มข้อมูลป้อนเข้าส่งไปยังชั้นข้อมูลส่งออก และไม่มีการส่งข้อมูลย้อนกลับ จึงเรียกรูปแบบการส่งผ่านข้อมูลแบบนี้ว่า ป้อนสู่ด้านหน้า (Feedforward) รูปที่ 3.8 เรียกว่า เครือข่ายชั้นเดียว (Single-layer network) การนับจำนวนชั้นของเครือข่ายในปัจจุบันจะนับเป็นชั้นเมื่อ ชั้นนั้นมีโหนดที่เป็นส่วนประมวลผล กล่าวคือ ไม่นับกลุ่มข้อมูลป้อนเข้าภายนอกของเครือข่ายเป็นชั้น เนื่องจากไม่มีส่วนประมวลผลอยู่

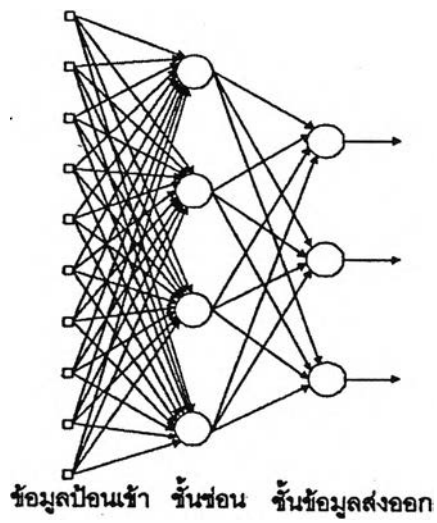


รูปที่ 3.8 เครือข่ายป้อนสู่ด้านหน้าชั้นเดียว

3.5.2 **เครือข่ายป้อนสู่ด้านหน้าหลายชั้น** (Multilayer feedforward networks)

โครงสร้างของเครือข่ายแบบที่สองเป็นแบบป้อนสู่ด้านหน้าเช่นกัน แต่แตกต่างจากแบบแรก เนื่องจากมีชั้นซ่อนอย่างน้อยหนึ่งชั้น ชั้นซ่อนหรือนิวรอลที่ซ่อน (Hidden neural) จะมีส่วนประมวลผลอยู่ด้วย นิวรอลที่ซ่อนอยู่นี้จะแทรกอยู่ระหว่างส่วนข้อมูลป้อนเข้าจากภายนอกของเครือข่ายและส่วนข้อมูลส่งออกจากเครือข่าย ข้อมูลป้อนเข้าโหนดต้นทางภายในกลุ่มข้อมูลป้อนเข้าของเครือข่ายจะถูกป้อนให้ส่วนประมวลผลของชั้นต่อไปหรือชั้นซ่อนชั้นแรกนั่นเอง ข้อมูล

ส่งออกที่ได้จากชั้นซ่อนชั้นแรกนี้จะถูกส่งให้กับชั้นที่อยู่ต่อไป ในฐานะข้อมูลป้อนเข้าของชั้นที่รับข้อมูลนั้น หากยังมีชั้นซ่อนเหลืออยู่ก็จะรับข้อมูลที่ส่งออกจากชั้นที่อยู่ก่อนหน้านี้ สำหรับข้อมูลส่งออกของเครือข่ายได้จากชั้นข้อมูลส่งออก ตัวอย่าง ของเครือข่ายป้อนสู่ด้านหน้าหลายชั้นดังรูปที่ 3.9



รูปที่ 3.9 เครือข่ายป้อนสู่ด้านหน้าหลายชั้น

เครือข่ายประสาทแบบหลายชั้นได้นำมาใช้ในการแก้ปัญหาที่ซับซ้อนและปัญหาที่ไม่ลู่เข้า (Diverse problems) ด้วยการปรับสอน (Training) ของเครือข่ายประสาทแบบชั้นเดียวได้ กระบวนวิธีที่นิยมใช้กันอย่างกว้างขวางคือ กระบวนวิธีแพร่กระจายความผิดพลาดกลับ (Error back-propagation algorithm) กระบวนวิธีนี้ขึ้นอยู่กับกฎการเรียนรู้เพื่อปรับค่าความผิดพลาด (Error-correction learning rule)

การที่จะทำให้เครือข่ายประสาทมีความสามารถในการแก้ปัญหาได้ต้องปรับสอนเครือข่ายประสาทหรือกล่าวอีกอย่างก็คือ เครือข่ายประสาทต้องมีความสามารถที่จะเรียนรู้ กระบวนการปรับสอนหรือกระบวนการเรียนรู้จะเกี่ยวกับความรู้ ความรู้ที่กล่าวถึงนี้คือข้อมูลข่าวสารแบบจำลองหรือสิ่งอื่นใดที่คนหรือเครื่องจักรเก็บไว้ใช้ในกระบวนการพยากรณ์ ทำนาย ประมาณการ คาดคะเน และตอบสนองอย่างเหมาะสมต่อสถานการณ์หรือสิ่งที่กระตุ้นภายนอก นั่นคือ สิ่งที่เกี่ยวกับความรู้จะมี 2 ประเด็น ประเด็นที่หนึ่งเกี่ยวกับความรู้ว่าเป็นข่าวสารข้อมูลอะไร ประเด็นที่สองคือ วิธีการเก็บข่าวสารข้อมูลเพื่อนำมาใช้ประโยชน์นั้นทำอย่างไร เนื่องจากผลลัพธ์ที่ดีจะขึ้นอยู่กับความถูกต้องแม่นยำของตัวผลลัพธ์เองและความรวดเร็วที่จะได้ผลลัพธ์นั้นมา เครือข่ายประสาทที่ดีจึงควรเป็นแบบจำลองภาวะในการเก็บความรู้พร้อมทั้งสามารถประมวลผลได้ด้วย การเรียนรู้

ของเครือข่ายประสาทเป็นกระบวนการทำให้เกิดความรู้ สามารถกล่าวได้ว่า การเรียนรู้ของเครือข่ายประสาทเป็นกระบวนการปรับค่าตัวแปรอิสระ (Free parameters) ของเครือข่ายเพื่อปรับพฤติกรรมให้สอดคล้องกับการกระตุ้นจากสิ่งแวดล้อม กระบวนการเรียนรู้ของเครือข่ายประสาทจึงเป็นกระบวนการทำซ้ำ (Iterative) ลำดับของกระบวนการเรียนรู้เป็นดังต่อไปนี้

1. เครือข่ายประสาทถูกกระตุ้นโดยสิ่งแวดล้อม
2. เครือข่ายจะปรับตัวเพื่อตอบสนองต่อสิ่งที่มากระตุ้น
3. เครือข่ายจะตอบสนองต่อสิ่งแวดล้อมด้วยภาวะใหม่หลังจากการปรับตัว

3.6 การเรียนรู้ของเครือข่ายประสาทเทียม (Neural network learning)

การเรียนรู้ของเครือข่ายประสาทจะใช้กฎการเรียนรู้ (Learning rule) เพื่อสอนให้เครือข่ายทำการคำนวณหาผลลัพธ์ในการทำงาน โดยการปรับเปลี่ยนค่าถ่วงน้ำหนักของเครือข่าย การเรียนรู้ของเครือข่ายประสาทเทียมมีหลายแบบ แต่ที่ใช้กันทั่วไปมี 2 แบบได้แก่

3.6.1 การเรียนรู้แบบมีการควบคุม (Supervised learning)

การเรียนรู้แบบนี้ต้องมีชุดตัวอย่างในการปรับสอน (Training set) เพื่อแสดงวัตถุประสงค์ที่แท้จริงในการทำงานของเครือข่ายนั้น ๆ ตัวอย่าง เช่น

$$\{x_1, t_1\}, \{x_2, t_2\}, \{x_3, t_3\}, \ldots, \{x_p, t_p\}$$

โดยที่ x_i คือ ตัวแปรด้านเข้าของเครือข่ายประสาท (Input Variable)

t_i คือ ผลลัพธ์ที่ต้องการ (Target)

p คือ จำนวนรูปแบบ (Pattern) ที่ใช้ในการปรับสอนทั้งหมด

เมื่อป้อนตัวแปรด้านเข้าให้แก่เครือข่ายประสาท เครือข่ายจะคำนวณหาผลลัพธ์คือตัวแปรด้านออก (Output variable) แล้วนำผลลัพธ์จากเครือข่ายประสาทนี้ไปเปรียบเทียบกับผลลัพธ์ที่ต้องการ จากนั้นนำค่าผิดพลาดที่เกิดขึ้นไปใช้ปรับค่าถ่วงน้ำหนักของเครือข่ายโดยอาศัยกฎการเรียนรู้ และวิธีการทำซ้ำ (Iterative method) เพื่อให้ตัวแปรด้านออกมีค่าเข้าใกล้ผลลัพธ์ที่ต้องการ

3.6.2 **การเรียนรู้แบบไม่มีการควบคุม** (Unsupervised learning)

การเรียนรู้แบบนี้ค่าถ่วงน้ำหนักจะถูกปรับเพื่อให้สอดคล้องกับตัวแปรด้านเข้าเท่านั้น ไม่ต้องใช้ค่าผลลัพธ์ที่ต้องการ ส่วนใหญ่การเรียนรู้แบบนี้จะใช้ในงานจำแนกประเภทของข้อมูลด้านเข้า (Clustering operation or network classifier) ถ้าข้อมูลด้านเข้าเป็นประเภทเดียวกัน เครือข่ายจะให้ผลลัพธ์อย่างเดียวกัน

สำหรับในงานวิจัยนี้ จะใช้วิธีการเรียนรู้แบบมีการควบคุมในการปรับสอนเครือข่ายประสาทเพื่อให้ได้ผลลัพธ์ที่ต้องการ

3.7 **เครือข่ายเพอร์เซฟตรอน** (Perceptron network)

เพอร์เซฟตรอนเป็นรูปแบบของเครือข่ายประสาทเทียมอย่างง่ายที่ใช้ฟังก์ชันฮาร์ดลิมิตเป็นแอคติเวชันฟังก์ชันหลัก ซึ่งมีคุณสมบัติหลักสำหรับใช้เพื่อให้เกิดการแยกออกจากกันได้เชิงเส้น (Linearity separable) ของตัวแปรด้านเข้า

3.7.1 **โครงสร้างและการทำงานของเครือข่ายเพอร์เซฟตรอน**

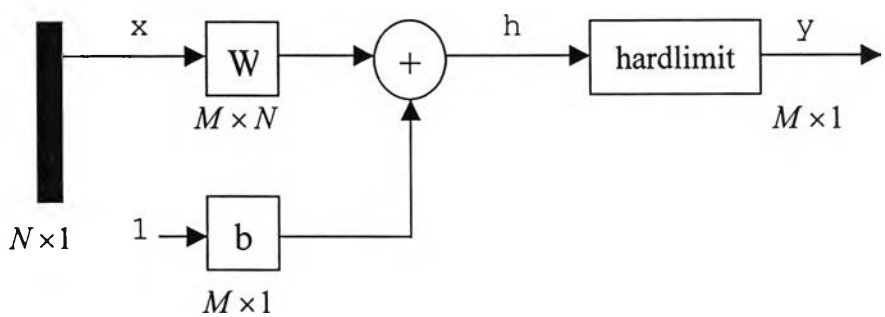
รูปแบบของเครือข่ายเพอร์เซฟตรอนแสดงในรูปที่ 3.10 ประกอบด้วยเวคเตอร์ของตัวแปรด้านเข้า x_i มีมิติ $N \times 1$ เมตริกซ์ของค่าถ่วงน้ำหนัก w มีมิติ $M \times N$ เวคเตอร์ของตัวแปรด้านออก y_i มีมิติ $M \times 1$ และค่าไบอัส b_i มีมิติ $M \times 1$ โดย M คือจำนวนปมประสาท ค่าของตัวแปรด้านออกสามารถคำนวณได้จาก

$$y_k = \text{hardlim}\left(\sum_{i=1}^{N} w_{ki} x_i + b_k\right) \tag{3.7}$$

โดยที่เมตริกซ์ค่าถ่วงน้ำหนักสามารถเขียนได้เป็น

$$w_{11} = \begin{bmatrix} w_{11} & w_{12} & \cdots & w_{1i} & \cdots & w_{1N} \\ w_{21} & w_{22} & \cdots & w_{2i} & \cdots & w_{2N} \\ \bullet & \bullet & & \bullet & & \bullet \\ w_{k1} & w_{k2} & \cdots & w_{ki} & \cdots & w_{kN} \\ \bullet & \bullet & & \bullet & & \bullet \\ w_{M1} & w_{M2} & \cdots & w_{Mi} & \cdots & w_{MN} \end{bmatrix} \quad (3.8)$$

จากสมการ (3.7) จะเห็นว่าถ้า Inner product ของตัวแปรด้านเข้ากับหลักที่ i^{th} ของเมตริกซ์ค่าถ่วงน้ำหนัก มีค่ามากกว่าหรือเท่ากับ $-b_k$ ตัวแปรด้านออกจะมีค่าเป็น 1 นอกนั้นจะมีค่าเป็น 0 ดังนั้น ในแต่ละนิวรอลของเครือข่ายเพอร์เซฟตรอนจะแบ่งชุดตัวแปรด้านเข้าออกเป็นสองส่วน โดยมีเส้นแบ่งหรือเส้นแสดงขอบเขตการตัดสินใจคือ $(w \cdot x) + b = 0$ ซึ่งเป็นลักษณะสมบัติของเครือข่ายแบบนี้



รูปที่ 3.10 เครือข่ายเพอร์เซฟตรอน

3.7.2 กฎการเรียนรู้ของเพอร์เซฟตรอน (Perceptron learning rule)

กฎการเรียนรู้ของเพอร์เซฟตรอนนี้เป็นวิธีการเรียนรู้แบบมีการควบคุมวิธีหนึ่งที่ใช้ในการแก้ปัญหาที่สามารถแยกออกจากกันได้เชิงเส้น (Linearly separable) ซึ่งมีคำตอบและวิธีการคำนวณค่าถ่วงน้ำหนักที่เหมาะสม โดยจะปรับค่าถ่วงน้ำหนักไปทีละนิวรอลตามตัวแปรด้านออกที่เกิดขึ้น ถ้าตัวแปรด้านออกมีเครื่องหมายเหมือนกับผลลัพธ์ที่ต้องการแล้วจะไม่ปรับค่าถ่วงน้ำหนัก ถ้าตัวแปรด้านออกมีเครื่องหมายตรงกันข้ามจะต้องทำการปรับค่าถ่วงน้ำหนัก โดยเป็นสัดส่วนเดียวกับผลคูณของตัวแปรด้านเข้ากับผลลัพธ์ที่ต้องการ ดังสมการ

$$w_{ki}^{new} = w_{ki}^{old} + \Delta w_{ki} \quad (3.9)$$

$$\Delta w_{ki} = (t_k - y_k)x_k \tag{3.10}$$

$$b_k^{new} = b_k^{old} + (t_k - y_k) \tag{3.11}$$

โดยที่ t_k = คือค่าผลลัพธ์ที่ต้องการจากนิวรอลที่ k

y_k = คือค่าตัวแปรด้านออกของนิวรอลที่ k

3.7.3 กฎการเรียนรู้แบบเกรเดียนต์เดสเซนต์ (Gradient descent learning) [15]

การเรียนรู้แบบแบบเกรเดียนต์เดสเซนต์เป็นการเรียนรู้พื้นฐานของเครือข่ายที่ประกอบด้วยนิวรอลที่ใช้ฟังก์ชันเชิงเส้น หรือฟังก์ชันที่ไม่เป็นเชิงเส้น เช่น ฟังก์ชันซิกมอยด์ต่าง ๆ ที่มีค่าต่อเนื่อง (Continuous functions) เป็นแอคติเวชันฟังก์ชัน นิยามของค่าความผิดพลาดของตัวแปรด้านออกด้วยฟังก์ชันที่แสดงดังสมการ

$$\begin{aligned} E &= \frac{1}{2}\sum_k (t_k - y_k)^2 \\ &= \frac{1}{2}\sum_k \left[t_k - f\left(\sum_i w_{ki} x_i + b_k \right) \right]^2 \end{aligned} \tag{3.12}$$

ฟังก์ชันค่าผิดพลาด (E) เป็นสมการกำลังสองในรูปของค่าถ่วงน้ำหนักกับค่าผิดพลาด กฎเกรเดียนต์เดสเซนต์จะปรับเวกเตอร์ถ่วงน้ำหนักในทิศทางของเวกเตอร์ตัวแปรด้านเข้า โดย E จะมีค่าเป็นบวกเสมอ ถ้า w_{ki} และ b_k ถูกปรับเข้าสู่ค่าที่ต้องการ ค่าของฟังก์ชันจะน้อยลงในลักษณะที่ลู่เข้าศูนย์เมื่อผ่านกระบวนการทำซ้ำไปเรื่อย ๆ วิธีเกรเดียนต์เดสเซนต์จะปรับเปลี่ยนค่า w_{ki} และค่า b_k แต่ละตัวด้วยค่า Δw_{ki} และ Δb_k ซึ่งเป็นสัดส่วนกับค่าเกรเดียนต์ของฟังก์ชันค่าผิดพลาดที่นิยามไว้ข้างต้น ดังสมการ

$$\begin{aligned} \Delta w_{ki} &= -\alpha \frac{\partial E}{\partial w_{ki}} \\ &= \alpha \sum_p (t_k - y_k) x_i \end{aligned} \tag{3.13}$$

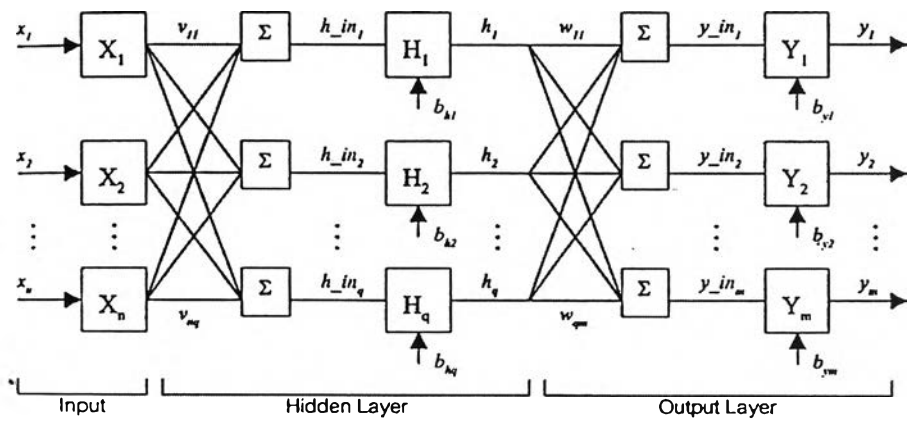
$$\Delta b_k = -\alpha \frac{\partial E}{\partial b_k}$$
$$= \alpha \sum_p (t_k - y_k) \quad (3.14)$$

โดยที่ p คือ จำนวนรูปแบบทั้งหมดที่ใช้ในการปรับสอน

α คือ ค่าอัตราการเรียนรู้

3.7.4 กฎการเรียนรู้แบบการแพร่กระจายความผิดพลาดกลับ (Error back-propagation learning rule)

พื้นฐานกระบวนวิธีการแพร่กระจายความผิดพลาดกลับประกอบด้วย การส่งผ่านสองแบบระหว่างชั้นของเครือข่ายประสาท คือ การส่งผ่านไปข้างหน้า (Forward pass) และการส่งผ่านย้อนกลับ (Backward pass) การส่งผ่านไปข้างหน้าข้อมูลจะถูกป้อนเข้าสู่เครือข่ายประสาทให้แก่ชั้นซ่อนเพื่อทำการคำนวณ และได้ผลลัพธ์ส่งออกมายังชั้นตัวแปรด้านออก โดยค่าถ่วงน้ำหนักที่เชื่อมต่อระหว่างชั้นของตัวแปรในเครือข่ายประสาททั้งหมดยังไม่เปลี่ยนแปลง ส่วนการส่งย้อนกลับจะตรงกันข้ามคือค่าถ่วงน้ำหนักระหว่างชั้นของเครือข่ายประสาทจะถูกปรับเปลี่ยนค่าไปตามกฎการปรับค่าความผิดพลาด (Error-correction rule) โดยค่าความแตกต่างระหว่างผลลัพธ์ที่ได้จากเครือข่ายประสาทกับค่าผลลัพธ์ที่ต้องการ (Desired or target) คือค่าความผิดพลาด (Error) ซึ่งค่าความผิดพลาดนี้จะแพร่กระจายกลับไปยังเครือข่ายตามทิศทางของการเชื่อมต่อ เพื่อนำไปใช้ปรับค่าถ่วงน้ำหนักและค่าไบอัสเพื่อให้ได้ค่าผลลัพธ์ที่ได้จากเครือข่ายเข้าใกล้ผลลัพธ์ที่ต้องการ



รูปที่ 3.11 เครือข่ายประสาทแบบป้อนไปข้างหน้าที่มี 2 ชั้น

I22170753

โดยกำหนดให้

X_i	คือ นิวรอลของตัวแปรด้านเข้าที่ i โดยที่ $i = 1,\ 2,\ 3,...,\ n$
H_j	คือ นิวรอลซ่อนที่ j โดยที่ $j = 1,\ 2,\ 3,...,\ q$
Y_k	คือ นิวรอลของตัวแปรด้านออกที่ k โดยที่ $k = 1,\ 2,\ 3,...,\ m$
h_in_j	คือ ผลรวมของข้อมูลด้านเข้าของนิวรอลซ่อนที่ j
h_j	คือ ข้อมูลด้านออกของนิวรอลซ่อนที่ j
y_in_k	คือ ผลรวมของข้อมูลด้านเข้าของนิวรอลตัวแปรด้านออกที่ k
y_k	คือ ข้อมูลด้านออกของนิวรอลตัวแปรด้านออกที่ k
v_{ij}	คือ ค่าถ่วงน้ำหนักระหว่างนิวรอลตัวแปรด้านเข้ากับนิวรอลซ่อน
w_{jk}	คือ ค่าถ่วงน้ำหนักระหว่างนิวรอลซ่อนกับนิวรอลตัวแปรด้านออก
b_{hj}	คือ ค่าไบอัสที่ป้อนให้แก่นิวรอลซ่อนที่ j
b_{yk}	คือ ค่าไบอัสที่ป้อนให้แก่นิวรอลของตัวแปรด้านออกที่ k

จะได้ผลรวมของข้อมูลด้านเข้าของนิวรอลซ่อน H_j คือ

$$h_in_j = \sum_i x_i v_{ij} + b_{hj} \tag{3.15}$$

ข้อมูลด้านออกจากการผ่านแอคติเวชันฟังก์ชันของนิวรอลซ่อน H_j คือ

$$h_j = f_h\left(h_in_j\right) \tag{3.16}$$

ผลรวมของข้อมูลด้านเข้าของนิวรอลของตัวแปรด้านออก Y_k คือ

$$y_in_k = \sum_j h_j w_{jk} + b_{yk} \tag{3.17}$$

ข้อมูลด้านออกจากการผ่านแอคติเวชันฟังก์ชันของนิวรอลของตัวแปรด้านออก Y_k คือ

$$y_k = f_y\left(y_in_k\right) \tag{3.18}$$

ดังนั้นสมการ (3.12) สามารถจัดรูปแบบใหม่ได้ดังนี้

$$E=\frac{1}{2}\sum_{k}\left[t_k-f_y\left(\sum_j w_{jk}f_h\left(\sum_i v_{ij}x_i+b_{hj}\right)+b_{yk}\right)\right]^2 \tag{3.19}$$

ด้วยการใช้กฎลูกโซ่ในการหาเกรเดียนต์จะได้

$$\frac{\partial E}{\partial w_{jk}}=-(t_k-y_k)f_y^{'}(y_in_k)h_j \tag{3.20}$$

$$\frac{\partial E}{\partial b_{yk}}=-(t_k-y_k)f_y^{'}(y_in_k) \tag{3.21}$$

$$\frac{\partial E}{\partial v_{ij}}=-\sum_k\left[(t_k-y_k)f_y^{'}(y_in_k)w_{jk}\right]f_h^{'}(h_in_j)x \tag{3.22}$$

$$\frac{\partial E}{\partial b_{hj}}=-\sum_k\left[(t_k-y_k)f_y^{'}(y_in_k)w_{jk}\right]f_h^{'}(h_in_j) \tag{3.23}$$

กำหนดให้

$$\delta_k=(t_k-y_k)f_y^{'}(y_in_k) \tag{3.24}$$

$$\delta_j=f_h^{'}(h_in_j)\sum_k(\delta_k w_{jk}) \tag{3.25}$$

ถ้า E เป็นฟังก์ชันที่สามารถหาอนุพันธ์ได้อย่างต่อเนื่องทุกค่าของค่าถ่วงน้ำหนักและค่าไบอัสจะสามารถคำนวณหาค่าถ่วงน้ำหนักและค่าไบอัสที่เหมาะสมโดยใช้กฎเกรเดียนต์เดสเซนต์ตามสมการ (3.13) และสมการ (3.14) ได้ดังนี้

สำหรับการปรับค่าถ่วงน้ำหนักและค่าไบอัสของนิวรอลตัวแปรด้านออก จะได้

$$\Delta w_{jk}=\alpha\delta_k h_j \tag{3.26}$$

$$\Delta b_{yk}=\alpha\delta_k \tag{3.27}$$

และสำหรับการปรับค่าถ่วงน้ำหนักและค่าไบอัสของนิวรอลซ่อน จะได้

$$\Delta v_{ij} = \alpha \delta_j x_i \tag{3.28}$$

$$\Delta b_{hj} = \alpha \delta_j \tag{3.29}$$

ค่าถ่วงน้ำหนักและค่าไบอัสของแต่ละนิวรอลจะถูกปรับด้วยค่า Δw และ Δb ตามลำดับ จนกระทั่งฟังก์ชันค่าผิดพลาดมีค่าน้อยลงถึงจุดที่ยอมรับได้ การที่กระบวนการนี้นำค่าผิดพลาด $(t_k - y_k)$ ในชั้นของตัวแปรด้านออกกลับมาคำนวณหาค่าที่ใช้ปรับค่าถ่วงน้ำหนักและค่าไบอัสของแต่ละนิวรอล เริ่มจากนิวรอลในชั้นของตัวแปรด้านออก จนถึงนิวรอลในชั้นซ่อนชั้นแรกตามลำดับ จึงเรียกกระบวนการนี้ว่า กระบวนการแพร่กระจายความผิดพลาดกลับ (Error back-propagation algorithm)

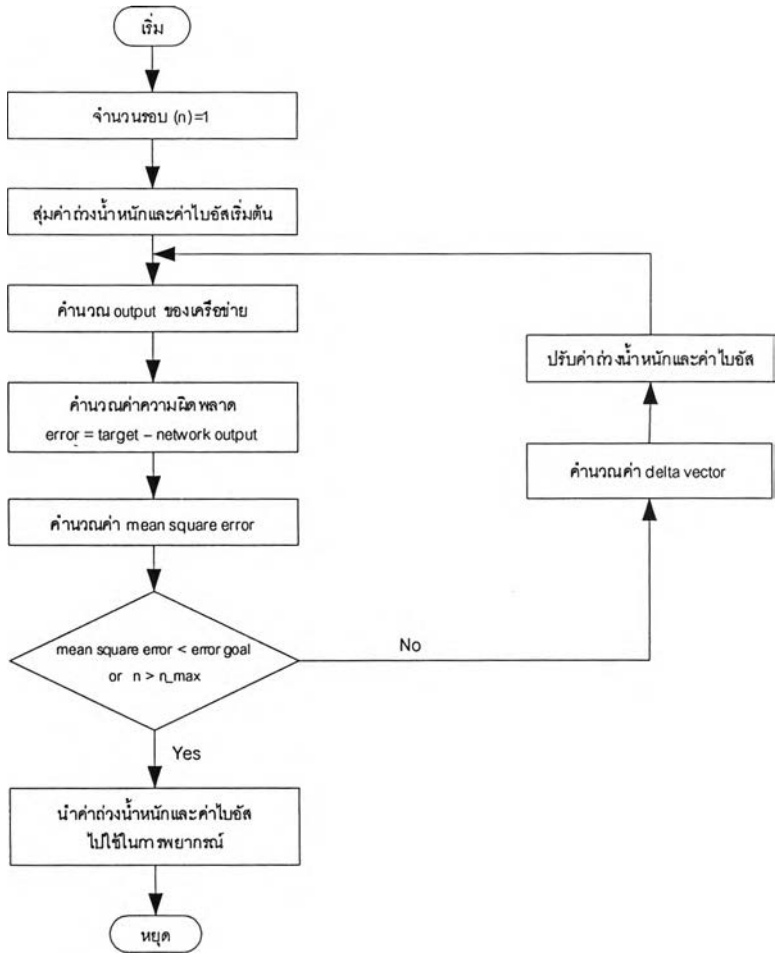
การนำกฎการเรียนรู้แบบแพร่กระจายความผิดพลาดกลับไปใช้ในการปรับสอนเครือข่ายประสาทเทียมสามารถสรุปเป็นขั้นตอนได้ดังนี้

ขั้นตอนที่ 1 คำนวณหาเวคเตอร์ตัวแปรด้านออกจากเวคเตอร์ตัวแปรด้านเข้าที่กำหนด ทั้งนี้หากเป็นการคำนวณรอบแรกจะต้องมีการสุ่มค่าน้อย ๆ ให้แก่ค่าถ่วงน้ำหนักและค่าไบอัสก่อน

ขั้นตอนที่ 2 ตรวจสอบเงื่อนไขการหยุดทำงาน โดยคำนวณค่าเฉลี่ยของค่าผิดพลาดกำลังสอง (Mean square error) หรือค่าผลรวมของค่าผิดพลาดกำลังสอง (Sum square error) ที่เกิดขึ้นจากเวคเตอร์ที่ใช้ในการปรับสอนทั้งหมด ถ้ามีค่าน้อยกว่าค่าผิดพลาดที่ยอมรับได้หรือรอบการปรับสอนเกินจำนวนรอบที่กำหนดไว้ ก็ให้หยุดการปรับสอน ไม่เช่นนั้นให้ทำการคำนวณต่อไปในขั้นตอนที่ 3

ขั้นตอนที่ 3 คำนวณค่า δ (Delta vector) ตามกฎการเรียนรู้แบบแพร่กระจายความผิดพลาดกลับโดยเริ่มจากชั้นตัวแปรด้านออกก่อน แล้วใช้กฎการเรียนรู้แบบแพร่กระจายความผิดพลาดกลับคำนวณค่า δ สำหรับชั้นถัดเข้าไป

ขั้นตอนที่ 4 คำนวณค่าถ่วงน้ำหนักและค่าไบอัสทั้งหมดใหม่ แล้วกลับไปทำขั้นที่ 1 ใหม่ ขั้นตอนทั้งหมดสามารถแสดงในรูปแผนภูมิการทำงานได้ดังนี้



รูปที่ 3.12 แผนภูมิแสดงการทำงานของเครือข่ายประสาทเทียมที่ใช้กระบวนการเรียนรู้แบบแพร่กระจายความผิดพลาดกลับ

3.7.5 วิธีการฝึกเครือข่ายแบบ Levenberg-Marquadt

วิธีการฝึกเครือข่ายแบบ Levenberg-Marquadt จะเป็นวิธีการของ Newton ซึ่งใช้อินเวิร์สเมตริกซ์ของ Hessian ในการทำหน้าที่เป็นผลของสัดส่วนการเรียนรู้คูณด้วยทิศทางการลดลงของค่าผิดพลาดแบบเกรเดียนส์ ซึ่ง Hessian เมตริกซ์จะเป็นดัชนีค่าความผิดพลาด (error index) สำหรับอนุพันธ์อันดับที่สองของค่าน้ำหนักและฟังก์ชันควอดราติก (quadratic function) เพื่อให้ได้ค่าที่จุดถัดไปต่ำที่สุดตามต้องการ ดังนั้นอัตราการลู่เข้าด้วยวิธีการแบบ Newton จะเร็วกว่าวิธีการแบบแพร่กระจายความผิดพลาดกลับแบบธรรมดา สมการการปรับค่าน้ำหนักด้วยวิธีของ Levenberg-Marquadt แสดงดังสมการที่ 3.30

$$x^{k+1} = x^k + \left(J^T J + \mu I\right)^{-1} J^T e \qquad (3.30)$$

เมื่อ J คือเมตริกซ์ Jacobian ซึ่งจะเป็นค่าผลต่างของค่าผิดพลาดของน้ำหนัก และ μ คือ ค่าน้ำหนักสำหรับปรับอัตราการเรียนรู้

3.8 ปัจจัยที่มีผลต่อการปรับสอน [15]

การปรับสอนให้แก่เครือข่ายโดยใช้กฎการเรียนรู้นั้นเป็นการปรับค่าถ่วงน้ำหนักให้ลู่เข้า การลู่เข้าจะเร็วหรือช้าขึ้นอยู่กับว่ากฎการเรียนรู้นั้นจะปรับค่าถ่วงน้ำหนักในทิศทางที่ถูกต้องหรือไม่ ปัจจัยอื่นนอกเหนือจากกฎการเรียนรู้ที่จะทำให้การลู่เข้าเร็วขึ้น คือ

3.8.1 จำนวนชั้นซ่อน

ขึ้นอยู่กับความซับซ้อนของเครือข่ายว่ามีความซับซ้อนมากน้อยเพียงใด ถ้ามีความซับซ้อนมากจะต้องใช้จำนวนชั้นซ่อนหลายชั้น แต่สำหรับงานโดยทั่วไป จำนวนชั้นซ่อน 1 ชั้นก็เพียงพอ

3.8.2 จำนวนหน่วยของชั้นซ่อน

ปัจจุบันยังไม่มีวิธีใดที่สามารถระบุจำนวนนิวรอลที่แน่นอนเพื่อที่จะบอกว่าจะต้องใช้กี่หน่วยจึงจะเหมาะสมที่สุด ซึ่งในการทำงานจริง ๆ จะทดลองกำหนดค่าหน่วยของชั้นซ่อน แล้วพิจารณาอัตราการลู่เข้า จำนวนหน่วยของชั้นซ่อนที่เหมาะสมควรเป็นค่าที่น้อยที่สุดโดยที่ยังคงอัตราการลู่เข้าที่ดี เพราะถ้าใช้จำนวนหน่วยซ่อนมากจะทำให้เวลาในการคำนวณระหว่างการปรับสอนช้ามากขึ้น และจำนวนหน่วยชั้นซ่อนที่ใช้ยังขึ้นอยู่กับชนิดของงานที่จะนำเครือข่ายประสาทไปใช้อีกด้วย

3.8.3 จำนวนข้อมูลที่ใช้ในการปรับสอน

ยิ่งมีข้อมูลมาก การเรียนรู้จะทำให้คำตอบถูกต้องยิ่งขึ้นแต่จะเสียเวลาในการปรับสอนเพิ่มขึ้นด้วย แต่ถ้าใช้จำนวนน้อยเกินไปจะทำให้ผลลัพธ์มีความคลาดเคลื่อนสูง ในงานทั่วไปจำนวนรูปแบบที่เพียงพอในการปรับสอนสามารถคำนวณได้จากสมการดังนี้

$$p = \frac{W}{e} \tag{3.31}$$

โดยที่ w คือ จำนวนค่าถ่วงน้ำหนักที่ใช้ในเครือข่ายทั้งหมด

e คือ ค่าความผิดพลาดที่ต้องการ

p คือ จำนวนรูปแบบในการปรับสอน

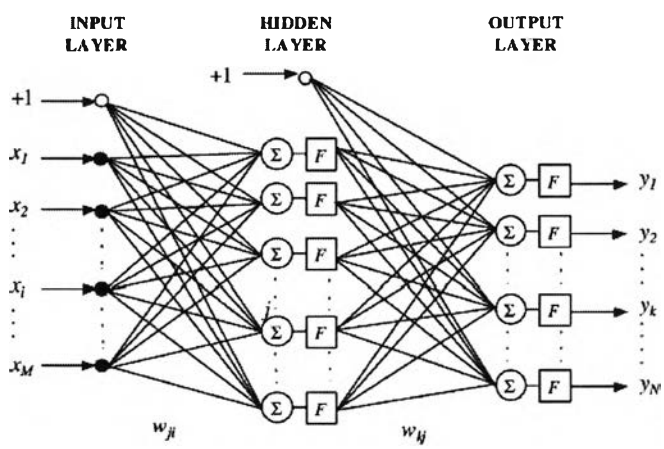
3.8.4 ค่าถ่วงน้ำหนักเริ่มต้น

หากใช้ค่าเริ่มต้นที่ดีจะทำให้ใช้จำนวนรอบในการปรับสอนลดลงหรือการปรับสอนเร็วขึ้นในทางปฏิบัติจะใช้การสุ่มค่าถ่วงน้ำหนักที่เป็นค่าบวกหรือค่าลบน้อย ๆ ระหว่าง -1 ถึง 1

3.9 แบบจำลองของโครงข่ายประสาทเทียม

3.9.1 แบบจำลองทางคณิตศาสตร์ของนิวรอลเน็ตเวิร์ก (Neural network modeling)

จากลักษณะของนิวรอลธรรมชาติสามารถนำมาสร้างเป็นแบบจำลองทางคณิตศาสตร์หรือเซลล์ประสาทเทียม ซึ่งมีข้อดีคือ สามารถใช้ในการคำนวณแบบจำลองที่สลับซับซ้อนและไม่เป็นเชิงเส้น ดังนั้นในงานวิจัยนี้จึงได้นำเซลล์ประสาทเทียมมาใช้ในการหาแบบจำลองของหน่วยทดสอบเซลล์เชื้อเพลิงแบบเยื่อแผ่นแลกเปลี่ยนโปรตอนแบบเซลล์เดียว

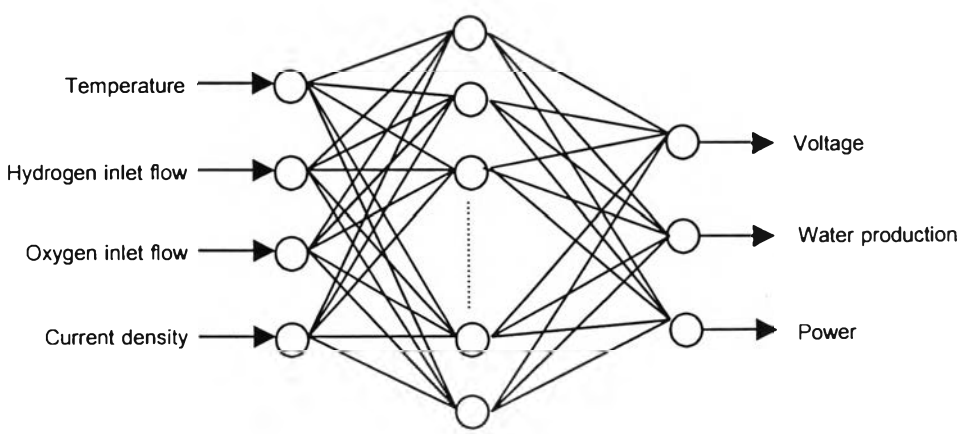


รูปที่ 3.13 โครงข่ายประสาทเทียมป้อนไปข้างหน้าแบบหลายชั้น [9]

รูปที่ 3.13 แสดงโครงสร้างของโครงข่ายประสาทเทียมแบบป้อนไปข้างหน้าแบบหลายชั้น ซึ่งจะประกอบด้วยชั้นอินพุต, ชั้นซ่อนและชั้นเอาท์พุต โดยที่ตัวแปรอินพุตแต่ละค่าจะเป็นลักษณะของโหนดในชั้นของตัวแปรขาเข้าและจะไม่เกิดกระบวนการใด ๆ ขึ้นที่ชั้นนี้ นั่นคือ โหนดของอินพุตจะมีลักษณะเช่นเดียวกับตัวสะสมสัญญาณขาเข้า $(x_1,...,x_M)$ หลังจากนั้นข้อมูลที่มาจากทุก ๆ อินพุตโหนดจะเกิดการกระจายตัวไปยังหน่วยในชั้นซ่อน และจะเกิดการขยายหรือลดค่าลงโดยรอยต่อระหว่างชั้นทั้งสอง นั่นคือ ค่าน้ำหนัก (w_{ji}) ข้อมูลแต่ละค่าที่มาถึงหน่วยของชั้นซ่อนจะเกิดการรวมเข้าด้วยกันโดยฟังก์ชันการรวมตัว (Σ) และเกิดการเปลี่ยนแปลงโดยฟังก์ชันการเปลี่ยนแปลง (F) จากชั้นซ่อนข้อมูลจะเกิดการกระจายตัวและค่าน้ำหนักอีกครั้งโดยการกำหนดค่าน้ำหนักใหม่ (w_{kj}) และจากนั้นข้อมูลก็จะผ่านไปยังหน่วยกระบวนการในชั้นเอาท์พุต เมื่อข้อมูลเกิดการรวมตัวและเกิดการเปลี่ยนอีกครั้งจะทำให้เกิดสัญญาณขาออก $(y_1,..., y_N)$มีอินพุตพิเศษอีกหนึ่งค่าซึ่งมีค่าเท่ากับ 1 จะถูกป้อนเข้าไปทั้งในหน่วยของชั้นซ่อนและหน่วยของชั้นเอาท์พุต ซึ่งนั่นคือค่าไบอัส

3.9.2 **เซลล์เชื้อเพลิงแบบพีอีเอ็มตามลักษณะโครงสร้างนิวรอลเน็ตเวิร์ก**

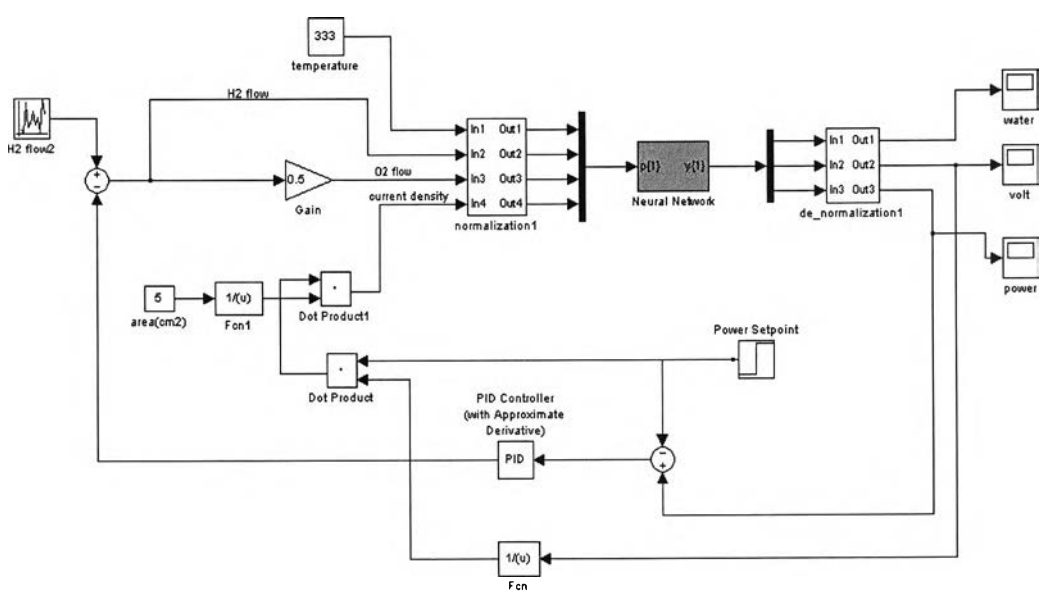
จากลักษณะโครงสร้างของนิวรอลเน็ตเวิร์กเทียมและความสัมพันธ์ของตัวแปรในระบบเซลล์เชื้อเพลิงแบบพีอีเอ็มสามารถนำมาสร้างแบบจำลองเป็นโครงสร้างของนิวรอลเน็ตเวิร์กเทียมได้ โดยกำหนดให้ อุณหภูมิของเซลล์ อัตราการไหลขาเข้าของแก๊สไฮโดรเจน อัตราการไหลขาเข้าของแก๊สออกซิเจนและค่าความหนาแน่นกระแสเป็นตัวแปรทางด้านอินพุต ค่าศักย์ไฟฟ้า อัตราการเกิดของน้ำและค่ากำลังไฟฟ้าที่เกิดขึ้นเป็นตัวแปรทางด้านเอาท์พุต ดังแสดงในรูปที่ 3.14



รูปที่ 3.14 โครงสร้างนิวรอลเน็ตเวิร์กของเซลล์เชื้อเพลิงแบบพีอีเอ็ม

3.10 การสร้างระบบควบคุมของหน่วยทดสอบเซลล์เชื้อเพลิงแบบพีอีเอ็ม

หน้าที่ของระบบควบคุมคือทำการปรับเปลี่ยนปัจจัยภายนอกที่มีผลกระทบต่อกระบวนการ (Input) ในทิศทางที่จะนำตัวแปรควบคุมไปสู่ค่าเป้าหมาย หรือไปในทิศทางที่หักล้างกับความเบี่ยงเบนหรือความแปรปรวนที่เกิดขึ้นจากปัจจัยที่เข้ามารบกวนกระบวนการ (Disturbances) [17] สำหรับในงานวิจัยนี้การสร้างระบบควบคุมเพื่อใช้เป็นแนวทางในการพัฒนาหน่วยทดสอบเซลล์เชื้อเพลิง โดยจะกำหนดให้ค่าอุณหภูมิ, อัตราการไหลของแก๊สไฮโดรเจน, อัตราการไหลของแก๊สออกซิเจนและค่าความหนาแน่นกระแสเป็นตัวแปรขาเข้า (Input variable) มีตัวแปรขาออก (Output variable) เป็นค่าศักย์ไฟฟ้า, ค่ากำลังไฟฟ้าและอัตราการเกิดน้ำ ทำการศึกษาผลการตอบสนองของระบบเมื่อเกิดการเปลี่ยนแปลงของตัวแปรขาเข้าที่สามารถปรับค่าได้ ดังนั้นงานวิจัยนี้จึงได้พัฒนาระบบควบคุมของหน่วยทดสอบเซลล์เชื้อเพลิงหลังจากได้แบบจำลองของนิวรอลเน็ตเวิร์กแล้ว การสร้างแบบจำลองจะใช้โปรแกรม Simulink ซึ่งเป็นโปรแกรมย่อยภายในโปรแกรม MATLAB ที่สามารถใช้ในการสร้างแบบจำลองของระบบต่าง ๆ ได้ โดยกำหนดให้แบบจำลองนิวรอลเน็ตเวิร์กที่ผ่านการฝึกและตรวจสอบความถูกต้องของแบบจำลองแล้วทำหน้าที่เสมือนเป็นหน่วยทดสอบเซลล์เชื้อเพลิงพีอีเอ็มแบบเซลล์เดียว ดังแสดงการควบคุมหน่วยทดสอบเซลล์เชื้อเพลิงแบบพีอีเอ็มแบบเซลล์เดียวแสดงได้ดังรูปที่ 3.15



รูปที่ 3.15 ระบบควบคุมหน่วยทดสอบเซลล์เชื้อเพลิง