บทที่ 2

ทฤษฎีที่ใช้ในการวิจัย

ทฤษฎีที่ใช้ในการวิจัยนี้ ประกอบด้วย ทฤษฎีที่เกี่ยวกับการประมวลผลข้อมูลตัวอักษรเบื้องต้น (Data pre-processing) และทฤษฎีนิวรอลเน็ตเวิร์ค รายละเอียดของทฤษฎีทั้งสองมีดังต่อไปนี้

2.1 การประมวลผลข้อมูลตัวอักษรเบื้องต้น (Data pre-processing) [4]

การเขียนตัวอักษรหนึ่งครั้งหรือหนึ่งสโตรค (Stroke) จะหมายถึงจุดที่สไตลัสสัมผัสกับพื้นที่เขียนเรื่อยไปจนกระทั่งยกสไตลัสขึ้น ดังนั้นจึงอธิบายลำดับของจุดบนระนาบ XY ของแต่ละสโตรคได้เป็น

$$S = p_1 p_2 \ldots p_L \tag{1}$$

เมื่อ $p_l = (x_l, y_l)\ \ 1 \le l \le L$

p_1 คือจุดแรกที่ปลายสไตลัสเริ่มสัมผัสกับพื้นที่เขียน

p_L คือจุดสุดท้ายก่อนที่จะยกสไตลัสออกจากพื้นที่เขียน

L คือจำนวนของจุดในการเขียนหนึ่งสโตรค

เพราะฉะนั้น ตัวอักษรที่เป็นลายมือเขียนสามารถอธิบายได้เป็นลำดับของสโตรคคือ

$$C = S_1 S_2 \ldots S_N \tag{2}$$

เมื่อ N คือจำนวนของสโตรคในการเขียนตัวอักษรนั้น ๆ

2.1.1 การทำนอร์มอลไลซ์ (Normalization) [4]

เนื่องจากว่าขนาดของตัวอักษรจากลายมือเขียน มีขนาดและตำแหน่งที่ไม่แน่นอน ดังนั้นจึงต้องมีการทำนอร์มอลไลซ์ (Normalize) เพื่อให้ง่ายต่อการประมวลผลต่อไป ในที่นี้การปรับอัตราส่วนทั้งในแนวระดับและแนวตั้งของตัวอักษรแต่ละตัวทำได้โดย

หอสมุดกลาง สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

$$x_i = \frac{x_i^o - x_{min}}{x_{max} - x_{min}} W \tag{3}$$

$$y_i = \frac{y_i^o - y_{min}}{y_{max} - y_{min}} H \tag{4}$$

โดยที่

(x_i^o, y_i^o) คือจุดที่เป็นข้อมูลจริง

(x_i, y_i) คือจุดที่เกิดจากการแปลงค่าแล้ว

$x_{\min}$ คือ พิกัด x ที่มีค่าน้อยที่สุดในข้อมูลจริงที่ได้จากการเขียนด้วยสไตลัส

$y_{\min}$ คือ พิกัด y ที่มีค่าน้อยที่สุดในข้อมูลจริงที่ได้จากการเขียนด้วยสไตลัส

$x_{\max}$ คือ พิกัด x ที่มีค่ามากที่สุดในข้อมูลจริงที่ได้จากการเขียนด้วยสไตลัส

$y_{\max}$ คือ พิกัด y ที่มีค่ามากที่สุดในข้อมูลจริงที่ได้จากการเขียนด้วยสไตลัส

W และ H คือความกว้างและความสูงของตัวอักษรที่ปรับแล้วตามลำดับ

ขั้นตอนต่อไปของการทำนอร์มอลไลซ์คือการทำให้แต่ละสโตรคมีความราบเรียบ ดังนั้นลำดับของจุดที่ใช้แทนสโตรคจะเป็นไปตามความสัมพันธ์ดังต่อไปนี้

$$p_{l+1} \neq p_l \tag{5}$$

$$|x_{l+1} - x_l| \leq 1 \tag{6}$$

$$|y_{l+1} - y_l| \leq 1 \tag{7}$$

ใช้เงื่อนไขข้างบนสำหรับแต่ละสโตรคเพื่อที่จะให้ได้รหัสลูกโซ่คือ

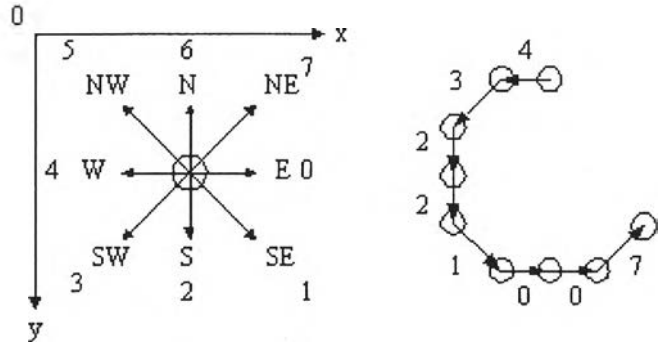
$$D = d_1 d_2 ... d_{L-1} \tag{8}$$

เมื่อ D คือลำดับของรหัสลูกโซ่จากจุด p_l ไปยังจุด p_{l+1} ถ้ามีจำนวนของจุดในสโตรคทั้งหมด L จุด จะมีจำนวนของลำดับ $L-1$ จุด

$d_l \in \{E, SE, S, SW, W, NW, N, NE\}$ และ $1 \leq l \leq L-1$

ซึ่งเป็นตัวบอกทิศทางของจุดจากจุดหนึ่งไปยังจุดถัดไป ทิศทางเหล่านี้แสดงในรูปที่ 2.1 (ก) และมีรหัสเป็น 0 ถึง 7

ค่าของ d_l สามารถหาได้จากค่าผลต่างของคู่ลำดับดังแสดงในตารางที่ 2.1 ตัวอย่างของรหัสลูกโซ่แสดงในรูปที่ 2.1 (ข) ซึ่งมีการแปลงค่าของการเขียนในสโตรคนี้แล้ว และรหัสลูกโซ่ที่ได้คือ 43221007



(ก) รหัสทิศทาง (ข) ตัวอย่างของรหัสลูกโซ่

รูปที่ 2.1 แสดงรหัสทิศทางและตัวอย่างของรหัสลูกโซ่ [4]

ตารางที่ 2.1 แสดงค่าของรหัสทิศทางจากผลต่างของคู่ลำดับ [4]

d_l	0	1	2	3	4	5	6	7
Δx	1	1	0	-1	-1	-1	0	1
Δy	0	1	1	1	0	-1	-1	-1

2.1.2 จุดเด่นในสโตรค (Dominant points) [4]

จุดเด่นในสโตรค หมายถึงจุดที่อยู่ในเงื่อนไขดังต่อไปนี้

(ก) เป็นจุดแรกในสโตรคนั่นคือจุดแรกที่ปลายปากกาสัมผัสกระดานเขียน และจุดสุดท้ายในสโตรค นั่นคือจุดที่ยกปลายปากกาขึ้น และ

(ข) จุดที่อยู่บนส่วนโค้งที่มีความโค้งมากที่สุด (Extrema of curvature) และ

(ค) จุดกึ่งกลางระหว่างจุดที่เป็นไปตามเงื่อนไขข้อที่ 1 หรือข้อที่ 2

2.1.3 จุดที่อยู่บนส่วนโค้งที่มีความโค้งมากที่สุด (Extrema of curvature) [7]

ในดิฟเฟอเรนเชียลแคลคูลัส ส่วนโค้งของ c ที่จุด p บนระนาบส่วนโค้งที่ต่อเนื่อง C ถูกนิยามโดย

$$c = \lim_{\Delta s \to 0} \frac{\Delta \alpha}{\Delta s} \tag{9}$$

เมื่อ s คือระยะทางไปยังจุด p ตามส่วนโค้ง และ $\Delta\alpha$ คือค่ามุมที่เปลี่ยนแปลงระหว่างจุด s และ $s + \Delta s$ พบว่าทิศทางของเส้นตรงแทนเจนท์จะเป็นไปตามทิศทางของความเร็ว (velocity) เนื่องจากค่าของส่วนโค้งจะเป็นค่าบวกหรือลบจะมีความสัมพันธ์กับทิศทางของส่วนโค้ง และใช้ในการกำหนดว่าส่วนโค้งนั้นเป็นส่วนโค้งนูน (convex) หรือโค้งเว้า (concave) ตามความสัมพันธ์ต่อไปนี้

- c เป็นส่วนโค้งนูน ก็ต่อเมื่อค่า $c < 0$
- c เป็นส่วนโค้งเว้า ก็ต่อเมื่อค่า $c > 0$

จากนิยามข้างบน ทิศทางของเส้นตรงแทนเจนท์จะหมุนตามเข็มนาฬิกาเมื่อเป็นส่วนโค้งนูน และทิศทางของเส้นแทนเจนท์จะหมุนทวนเข็มนาฬิกาเมื่อเป็นส่วนโค้งเว้า

ในทางปฏิบัติ เป็นการยากที่จะคำนวณหาค่าของลิมิตตามสมการข้างบน อย่างไรก็ตามเมื่อให้ $\Delta s = 1$ ตามแนวของส่วนโค้งแล้ว ความโค้งสามารถประเมินได้เป็น $c = \Delta\alpha$ จากแนวคิดนี้สามารถคำนวณได้จากการทำการอินเตอร์โพเลท (Interpolate) นั่นคือ $C = p_1 p_2 ... p_L$ เมื่อ $d(p_l, p_{l+1}) = 1,\ 1 \le l < L$ ในกรณีนี้ ส่วนโค้ง c_l ที่จุด p_l สามารถคำนวณได้จาก

$$c_l = \Delta\alpha_l \tag{10}$$

เมื่อให้ค่า $\Delta s = 1$

เพื่อประเมินค่าของส่วนโค้งข้างบน จะทำการคำนวณค่าลำดับของมุมจากจุดหนึ่งไปยังอีกจุดหนึ่ง

$$A = \alpha_1 \alpha_2 ... \alpha_L \tag{11}$$

เมื่อ $\alpha_l \in [-180°, 180°]$ ซึ่งเป็นมุมของเส้นตรงแทนเจนท์ที่จุด p_l และประเมินค่าได้เป็น

$$\alpha_l = \tan^{-1}\left[\frac{y_{l+1} - y_l}{x_{l+1} - x_i}\right] \tag{12}$$

จากค่าของ A สามารถหาลำดับค่าของมุมที่เปลี่ยนแปลงได้ดังนี้

$$\Delta A = \Delta\alpha_1 \Delta\alpha_2 ... \Delta\alpha_L \quad (13)$$

เมื่อ

$$\Delta\alpha_l = (\alpha_l - \alpha_{l-1}) \bmod 360° \quad (14)$$

จากสมการ (14) รับประกันได้ว่า $\Delta\alpha_l \in [-180°, 180°]$ ในกรณีที่ $\alpha_l - \alpha_{l-1} = \pm 180°$ ค่าสัญลักษณ์ของ $\Delta\alpha_l$ จะเหมือนกับของ $\Delta\alpha_{l-1}$ เพราะสัญญาณดั้งเดิมจะต้องมีความต่อเนื่อง

เป็นที่ทราบอยู่แล้วว่าสัญญาณของส่วนโค้งจะถูกปะปนอยู่กับสัญญาณรบกวนอื่น ๆ ซึ่งสามารถลดสัญญาณรบกวนนี้ด้วยการคอนโวลว์ (convolving) ลำดับของ ΔA ด้วยฟิลเตอร์ G ในสปาเซียลโดเมน (spatial domain) นั่นคือ $\Delta A^* = \Delta A * G$ เพราะฉะนั้นค่าของส่วนโค้งจึงประเมินได้บนพื้นฐานของสัญญาณที่ถูกกรองเรียบร้อยแล้วคือ

$$c_l = \Delta\alpha_l^* \quad (15)$$

เมื่อได้รับสัญญาณข้างบนนี้แล้ว เราจะกำหนดจุดที่อยู่บนส่วนโค้งที่มีความโค้งมากที่สุดได้จาก

(ก) ความเข้มของสัญญาณ

$$I = \sqrt{\frac{1}{L}\sum_{l=1}^{L} \Delta\alpha_l^* \times \Delta\alpha_l^*} \quad (16)$$

(ข) ค่าขีดแบ่ง (threshold value) ของความเข้มของสัญญาณ

$$T = k_S I + k_L \quad (17)$$

เมื่อ k_S คือ ค่าสัมประสิทธิ์ของความชัน และ k_L เป็นค่าต่ำสุดของค่าขีดแบ่ง

โดยใช้การคำนวณข้างบน เราจะกำหนดจุดที่อยู่บนส่วนโค้งที่มีความโค้งมากที่สุดได้คือ

ถ้า $\Delta\alpha_l^*$ เป็นค่าที่น้อยที่สุดนั่นคือ $\Delta\alpha_l^* \le -T$ ฉะนั้น p_l เป็นจุดต่ำสุดของส่วนโค้ง

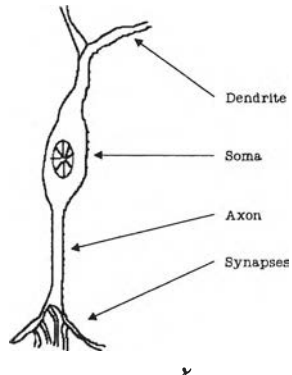
ถ้า $\Delta\alpha_l^*$ เป็นค่าที่มากที่สุดนั่นคือ $T \le \Delta\alpha_l^*$ ฉะนั้น p_l เป็นจุดสูงสุดของส่วนโค้ง

เพราะฉะนั้นทุกจุดที่เป็นไปตามเงื่อนไขข้างบนนี้ จะใช้แทนจุดของส่วนโค้งที่มีความโค้งมากที่สุด (Extrema of curvature) นั่นเอง

2.2 ทฤษฎีนิวรอลเน็ตเวิร์ค (NEURAL NETWORK) [8]

นิวรอลเน็ตเวิร์ค (Neural Network) คือ การจำลองการทำงานเน็ตเวิร์คของเซลสมองมนุษย์ โดยพื้นฐานแล้ว สมองของมนุษย์เรียนรู้จากประสบการณ์ที่ผ่านมา ซึ่งได้พิสูจน์แล้วว่าปัญหาบางประการที่ไม่สามารถแก้ได้โดยใช้คอมพิวเตอร์ในปัจจุบัน สามารถแก้ได้โดยสมองของมนุษย์ คอมพิวเตอร์ได้นำมาใช้งานทางด้านต่าง ๆ ในองค์กร เช่น ทางด้านบัญชี ทางด้านวิศวกรรม หรือที่ต้องใช้ในการคำนวณทางคณิตศาสตร์ที่ซับซ้อน แต่สิ่งที่คอมพิวเตอร์ทำไม่ได้ก็คือการรู้จำรูปแบบต่างต่าง แม้ว่าจะเป็นรูปแบบที่ง่าย ๆ ก็ตาม

การวิจัยทางด้านชีววิทยาในปัจจุบันทำให้เข้าใจถึงกลไกการทำงานของการคิด พบว่าสมองมีการเก็บข้อมูลอย่างมีรูปแบบ รูปแบบนี้บางส่วนมีความซับซ้อน และมีความสามารถในการรู้จำรูปร่างต่าง ๆ ดังนั้นแนวทางใหม่ในวงการคอมพิวเตอร์จึงเป็นการเลียนแบบจากสมองของมนุษย์คือ อาศัยกระบวนการของการเก็บข้อมูลอย่างมีรูปแบบ มีการใช้งานรูปแบบต่าง ๆ อย่างเหมาะสม และการแก้ไขปัญหาที่อยู่รอบข้าง ซึ่งแนวทางใหม่นี้ไม่ได้ใช้ประโยชน์จากการเขียนโปรแกรมแบบดั้งเดิม แต่จะเกี่ยวข้องกับการสร้างเครือข่ายขนาดใหญ่แบบคู่ขนาน และมีการสอนให้เครือข่ายนี้รู้การแก้ปัญหาเฉพาะทาง



รูปที่ 2.2 แสดงส่วนประกอบพื้นฐานของเซลประสาท [8]

จากรูปที่ 2.2 เป็นการแสดงส่วนประกอบพื้นฐานของเซลประสาทของมนุษย์ซึ่งมีอยู่ 4 ประเภทด้วยกัน คือ

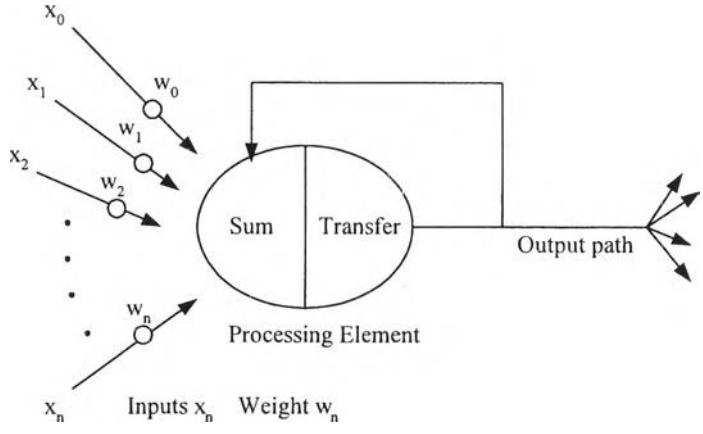
(ก) เดนไดรท์ (Dendrite) ทำหน้าที่รับข้อมูลเข้า

(ข) โซมา (Soma) ทำหน้าที่ประมวลผลข้อมูลที่เข้ามา

(ค) แอ็กซอน (Axon) ทำหน้าที่เปลี่ยนข้อมูลเข้าที่ผ่านการประมวลผลแล้วเป็นเอาท์พุต

(ง) ไซแน็ปส์ (Synapses) เป็นเคมีไฟฟ้า (Electrochemical) เชื่อมต่อระหว่างเซลประสาท

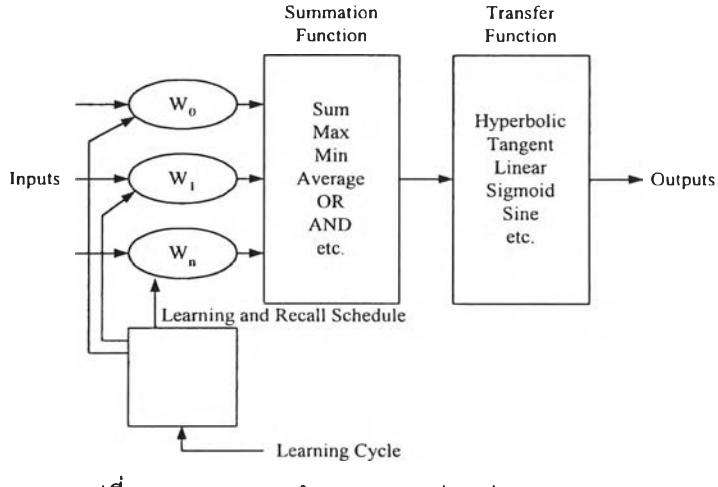
หน่วยพื้นฐานของนิวรอลเน็ตเวิร์คแสดงดังรูปที่ 2.3



รูปที่ 2.3 แสดงหน่วยพื้นฐานของเซลประสาท [8]

รูปที่ 2.3 ข้อมูลที่เข้าสู่เน็ตเวิร์คซึ่งแทนด้วยสัญลักษณ์ x_n จะนำมาคูณด้วยค่าน้ำหนัก (Weight) ของแต่ละตัว ค่าน้ำหนักนี้จะแทนด้วย w_n ผลลัพธ์ที่ได้จะนำมารวมกันแล้วส่งต่อไปยังฟังก์ชันการแปลงค่า (Transfer function) เพื่อสร้างผลลัพธ์ หรือเอาท์พุต

ซอฟต์แวร์ทางด้านนิวรอลเน็ตเวิร์คจะเรียกเซลประสาทว่า หน่วยประมวลผล (Processing element)

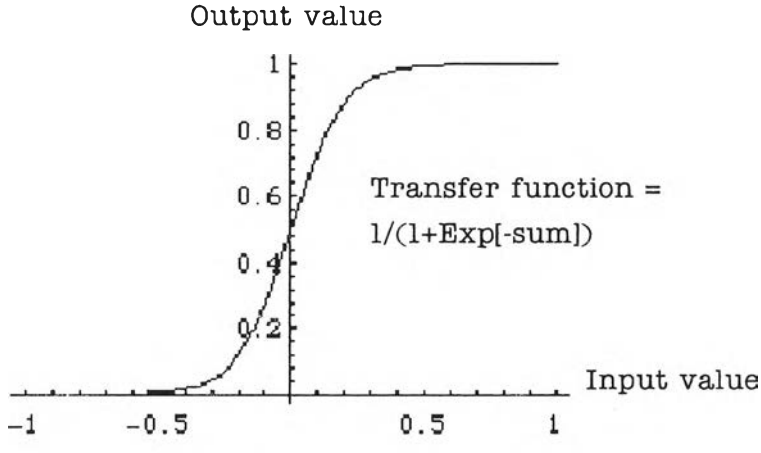


รูปที่ 2.4 แสดงแบบจำลองของหน่วยประมวลผล [8]

จากรูปที่ 2.4 อินพุต (Inputs) จะเป็นข้อมูลเข้าที่ป้อนสู่หน่วยประมวลผล ข้อมูลนี้จะนำมาคูณกับค่าน้ำหนัก w_n แล้วจึงป้อนสู่ฟังก์ชันผลรวม (Summing function) ฟังก์ชันผลรวมอาจเป็นฟังก์ชัน

ของ ค่าเฉลี่ย ค่าที่มากที่สุด ค่าที่น้อยที่สุด เป็นต้น ผลลัพธ์จากฟังก์ชันผลรวมจะถูกส่งไปยังฟังก์ชันการแปลงค่า (Transfer function) ซึ่งฟังก์ชันนี้จะเปลี่ยนข้อมูลที่รับเข้ามานี้ให้เป็นผลลัพธ์จริงโดยใช้อัลกอริธึมบางอย่าง เช่น การใช้ฟังก์ชันซิกมอยด์ (Sigmoid function) ฟังก์ชันไซน์ (Sine function) เป็นต้น ฟังก์ชันการแปลงค่าสามารถปรับค่าของผลลัพธ์ หรือควบคุมค่าของผลลัพธ์ได้โดยใช้ค่าขีดแบ่ง

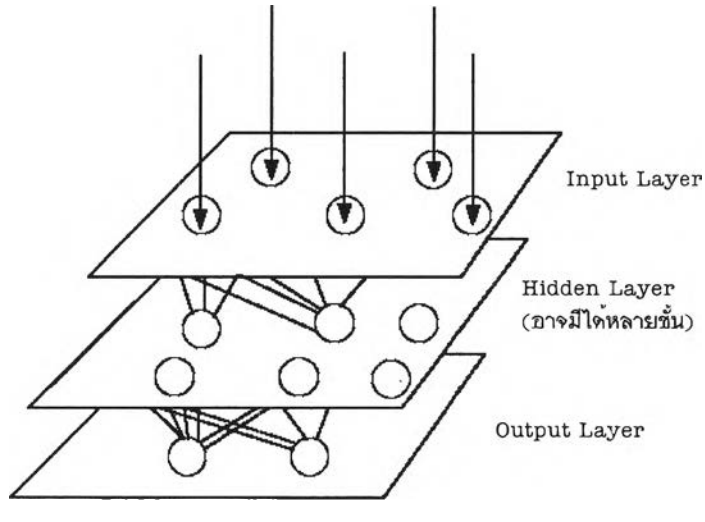
ตัวอย่างของฟังก์ชันการแปลงค่า



รูปที่ 2.5 ตัวอย่างของฟังก์ชันซิกมอยด์ [8]

ฟังก์ชันซิกมอยด์จะรับค่าจากฟังก์ชันผลรวมแล้วเปลี่ยนข้อมูลนั้นให้มีค่าระหว่าง 0 ถึง 1 ดังรูปที่ 2.5

ผลลัพธ์ที่ได้จากหน่วยประมวลผลหนึ่ง จะถูกส่งต่อไปยังหน่วยประมวลผลอื่น หรือติดต่อกับภายนอก ตามโครงสร้างของเน็ตเวิร์คที่กำหนดขึ้น



รูปที่ 2.6 แสดงไดอะแกรมอย่างง่ายของนิวรอลเน็ตเวิร์ค [8]

โดยพื้นฐานแล้ว นิวรอลเน็ตเวิร์คจะมีโครงสร้างที่คล้ายคลึงกัน ดังรูปที่ 2.6 แต่ละชั้น (Layer) จะมีหน่วยประมวลผลจำนวนต่าง ๆ กัน และแบ่งออกเป็น 3 ประเภทคือ

(ก) ชั้นข้อมูลเข้า (Inputer layer) ทำหน้าที่รับข้อมูลเข้ามา
(ข) ชั้นแอบแฝง (Hidden layer) ทำหน้าที่ประมวลผลข้อมูล
(ค) ชั้นผลลัพธ์ (Output layer) ทำหน้าที่แสดงผลลัพธ์จากชั้นแอบแฝง

2.2.1 การสอนนิวรอลเน็ตเวิร์ค [9]

แบ่งออกเป็น 2 ประเภทคือ

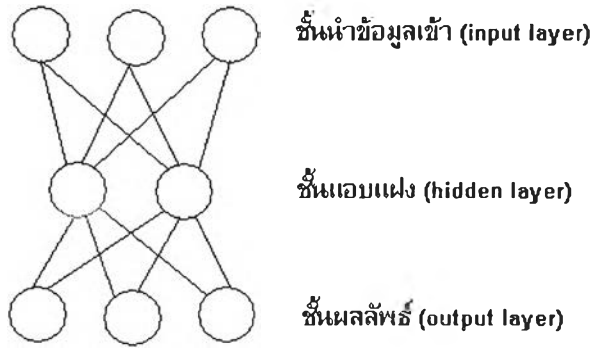
(ก) การสอนแบบมีผู้แนะนำ (Supervised training) ต้องการชุดข้อมูลเข้า และชุดข้อมูลเป้าหมายหรือข้อมูลที่ต้องการ ทั้งชุดข้อมูลเข้าและชุดข้อมูลเป้าหมายจะเรียกว่า ชุดการสอนควบคู่ (Training pair) โดยปกติ การสอนเน็ตเวิร์คจะใช้ชุดการสอนควบคู่หลายชุด ในระหว่างการสอนจะเกิดผลลัพธ์จริงขึ้น โดยที่ผลต่างระหว่างผลลัพธ์จริงกับผลลัพธ์เป้าหมายคือค่าความคลาดเคลื่อนหรือค่าความผิดพลาด

(ข) การสอนแบบไม่มีผู้แนะนำ (Un-supervised training) ถูกพัฒนาโดยโคโฮเนน (ค.ศ. 1984) และท่านอื่น ๆ เพื่อให้ใกล้เคียงกับระบบการเรียนรู้ของสมองมนุษย์มากยิ่งขึ้น โดยไม่ต้องมีชุดข้อมูลเป้าหมาย มีเพียงชุดข้อมูลเข้า โดยกระบวนการเรียนรู้เน็ตเวิร์คนั้น จะใช้หลักการทางสถิติ โดยหาค่าทางสถิติของชุดการสอน และจัดกลุ่มข้อมูลออกเป็นระดับต่าง ๆ และนิวรอลเน็ตเวิร์คจะหาค่าผลลัพธ์เอง โดยจะเป็นความสัมพันธ์ระหว่างข้อมูลเข้าและผลลัพธ์

2.2.2 นิวรอลเน็ตเวิร์คแบบแพร่ย้อนกลับ (Back-propagation)

นิวรอลเน็ตเวิร์คประเภทนี้ได้ถูกพัฒนาขึ้นในราวปี ค.ศ. 1970 ลักษณะพื้นฐานของเน็ตเวิร์คประเภทนี้จะประกอบไปด้วยชั้นนำข้อมูลเข้า (input layer) จำนวน 1 ชั้น ชั้นผลลัพธ์ (output layer) จำนวน 1 ชั้น และมีชั้นแอบแฝง (hidden layer) อย่างน้อยหนึ่งชั้น โดยทั่วไปจะมีจำนวนของชั้นแอบแฝงหนึ่งหรือสองชั้น โหนดแต่ละโหนดในชั้นที่ติดกันจะเชื่อมโยงถึงกันแบบสมบูรณ์ (fully-connected) ดังรูปที่ 2.7 ในช่วงระหว่างขั้นตอนของการรู้จำ ข้อมูลนำเข้าจะถูกป้อนเข้าสู่ชั้นนำข้อมูลเข้า แล้วผ่าน

ไปยังชั้นผลลัพธ์ในนิวรอลเน็ตเวิร์ค ที่ได้รับการสอนแล้ว (trained neural network) ส่วนในช่วงของการสอนและทดสอบนิวรอลเน็ตเวิร์คเท่านั้นที่มีการใช้โครงสร้างแบบแพร่กระจายย้อนกลับ



รูปที่ 2.7 นิวรอลเน็ตเวิร์คแบบแพร่กระจายย้อนกลับ

2.2.3 การนำนิวรอลเน็ตเวิร์คไปใช้งาน

โดยทั่วไปแล้วภายหลังจากการสอนและทดสอบนิวรอลเน็ตเวิร์คผลลัพธ์ที่ได้ก็คือนิวรอล เน็ตเวิร์คที่ได้รับการสอนแล้วและสามารถนำไปใช้งานตามวัตถุประสงค์ที่ต้องการได้ เช่น งานที่เกี่ยวข้องกับการพยากรณ์ การรู้จำรูปแบบหรือรู้จำเสียง เป็นต้น นั่นคือค่าน้ำหนัก (weight) จะไม่มีการเปลี่ยนแปลงอีกต่อไปในขั้นตอนของการนำไปใช้งาน นิวรอลเน็ตเวิร์คที่ได้รับการสอนแล้วนี้อาจถูกนำไปบรรจุในอุปกรณ์ฮาร์ดแวร์เพื่อให้ทำงานได้อย่างรวดเร็วก็ได้ หรือทำการแปลงให้อยู่ในรูปของโปรแกรมคอมพิวเตอร์ก็ได้ โดยที่จะมีจำนวนของข้อมูลนำเข้าเท่ากับจำนวนของข้อมูลในชั้นนำข้อมูลเข้า เมื่อทำการป้อนข้อมูลเข้าไปในอุปกรณ์ฮาร์ดแวร์หรือโปรแกรมคอมพิวเตอร์นี้แล้ว จะมีการคำนวณขึ้นภายในแล้วให้ผลลัพธ์ออกมา จำนวนของผลลัพธ์จะมีจำนวนเท่ากับจำนวนของข้อมูลในชั้นผลลัพธ์ของนิวรอลเน็ตเวิร์ค จากนั้นก็สามารถนำเอาผลลัพธ์ที่ได้นี้ไปใช้ตามวัตถุประสงค์ที่ต้องการต่อไป