บทที่ 2
ทฤษฎีโครงข่ายประสาทเทียม

2.1 ประวัติการพัฒนาโครงข่ายประสาทเทียม

โครงข่ายประสาทเทียม ได้มีการคิดและวิจัยกันมาตั้งแต่ปี ค.ศ. 1940 โดยในปี ค.ศ. 1943 วอร์แรน แมคคูลล็อค และ วอลเตอร์ พิทซ์ [8] ได้ทำการออกแบบโครงข่ายนิวรอล ซึ่งถือได้ว่าเป็นต้นกำเนิด และพื้นฐานของโครงข่ายประสาทเทียมในปัจจุบัน โครงข่ายที่แมคคูลล็อค และพิทซ์ ได้ให้แนวคิดไว้ประกอบด้วย โครงข่ายจำนวนสองชั้น คือชั้นอินพุต และชั้นเอาต์พุตโครงข่ายจะมีการเชื่อมโยงถึงกัน จากอินพุตกับเอาต์พุต โดยมีค่าน้ำหนัก (Weight) ส่งผ่านถึงกัน

ต่อมาหลังปี ค.ศ. 1960 เป็นต้นมา โครงข่ายประสาทเทียม ได้รับการพัฒนารูปแบบเป็นชนิดต่าง ๆ เป็นจำนวนมาก รวมทั้งกฎของการเรียนรู้

2.2 โครงข่ายประสาทตามทัศนคติชีววิทยา

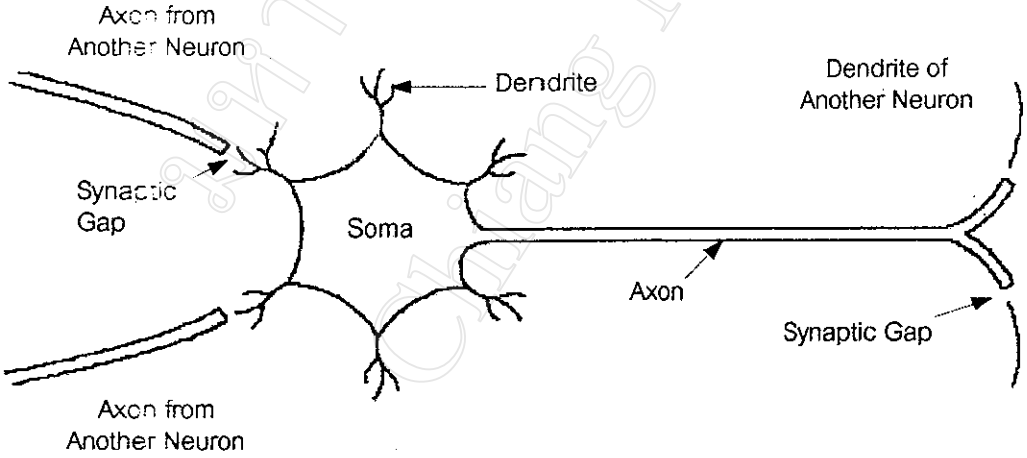
แบบจำลองโครงข่ายนิวรอล ที่มีรายละเอียดไม่เหมือนกันกับระบบประสาทตามทัศนคติชีววิทยา ซึ่งก็แล้วแต่จะให้ความสำคัญใดเป็นหลัก แต่ส่วนใหญ่แล้วจะเน้นความสามารถของโครงข่ายที่นำคุณสมบัติมาใช้งาน มากกว่าเหตุผลตามทัศนคติชีววิทยาของโครงข่าย ถึงแม้ว่าเกือบทั้งหมดจะให้ความสนใจเป็นอย่างมากกับความสามารถในการคำนวณของนิวรอลเน็ตเวอร์คก็ตาม ต่อไปนี้จะได้อธิบายถึงลักษณะบางอย่างของนิวรอนตามทัศนคติชีววิทยา ซึ่งอาจจะช่วยให้เข้าใจอย่างชัดเจนถึงลักษณะพิเศษที่สำคัญของโครงข่ายประสาทเทียม เพื่อเข้าถึงจุดแรกเริ่มสำหรับโครงข่ายประสาทเทียม ในระบบนิวรอลตามทัศนคติชีววิทยาได้ให้แนวคิดลักษณะสำคัญถึงสิ่งที่เป็นประโยชน์ต่อการคำนวณอย่างพิเศษ อุปมาเหมือนมีความใกล้เคียงกันมากระหว่างโครงสร้างของนิวรอนตามทัศนคติชีววิทยา (สมอง หรือ เซลล์ประสาท) และส่วนประมวลผล (หรือนิวรอน-เทียม) ในความเป็นจริงแล้วโครงสร้างของนิวรอนแต่ละเซลล์จะเปลี่ยนแปลงน้อยมาก

นิวรอนตามทัศนคติชีววิทยา มีส่วนประกอบที่สำคัญและให้ความสนใจเป็นพิเศษ เพื่อที่จะทำความเข้าใจถึงนิวรอนเทียม มีด้วยกัน 3 ส่วน คือ เดนไดรต์ (Dendrite) โซมา (Soma) และแอกซอน (Axon) สำหรับ เดนไดรต์จะรับสัญญ ณมาจากนิวรอนอื่น สัญญาณก็คือแรงกระตุ้นประจุไฟฟ้า ที่ถูกส่งผ่านข้ามมายังช่องไซแนบติก (Synaptic) นั้นหมายถึงการส่งผ่านตาม

กระบวนการทางเคมี พฤติการณ์ของการแปรเปลี่ยนส่งผ่านทางเคมี สัญญาณที่เข้ามา มีวิธีการคล้ายกับพฤติการณ์ของน้ำหนักในโครงข่ายประสาทเทียม

โซมา หรือลำตัวเซลล์ จะทำหน้าที่รวมสัญญาณที่ส่งผ่านเข้ามา ภายในตัวของโซมาจะทำให้เกิดแรงที่ต้องการส่งสัญญาณข้ามผ่านบนแอกซอนไปยังเซลล์อื่น การส่งสัญญาณจากนิวรอนได้เป็นผลสำเร็จนั้นก็ด้วยพฤติการณ์ที่เป็นผลระดับความดันของไฟฟ้าจากการเชื่อมต่อที่ต่างกันของไออ็อน (Ion) บนแอกซอนของนิวรอนอีกด้านหนึ่ง ในรูปที่ 2.1 แสดงนิวรอนตามทัศนคติชีววิทยาแบบทั่วไป เงื่อนไขที่พิเศษหลายอย่างของการประมวลผลในโครงข่ายประสาทเทียม ที่สำเร็จได้ด้วย คุณสมบัติของนิวรอนตามทัศนคติชีววิทยา มีดังนี้

1) ส่วนประมวลผล ที่รับสัญญาณเข้ามาหลายส่วน
2) สัญญาณอาจถูกแปรเปลี่ยนโดยน้ำหนัก ณ ที่การรับไซแนบ
3) ส่วนประมวลผล ทำการรวมอินพุตน้ำหนัก
4) ภายใต้พฤติการณ์ที่เหมาะสม นิวรอนจะส่งสัญญาณเอาต์พุต
5) เอาต์พุตจากนิวรอนที่พิเศษอาจจะส่งไปยังนิวรอนอื่นได้หลายเซลล์ (ตามกิ่งแขนงแอกซอน)



รูปที่ 2.1 นิวรอนตามทัศนคติชีววิทยา [8]

2.3 โครงข่ายประสาทเทียมตามทัศนคติของคอมพิวเตอร์

ปัจจุบันวิศวกรและนักวิทยาศาสตร์แขนงต่าง ๆ ได้พยายามพัฒนาเครื่องคอมพิวเตอร์ให้มีความฉลาดมากยิ่งขึ้น โดยพยายามให้เครื่องคอมพิวเตอร์มีการเรียนรู้และสามารถทำการตัดสินใจได้เอง หรือเป็นส่วนหนึ่งที่ช่วยในการตัดสินใจในขั้นต้น ระบบการจำลองโครงข่ายประสาทเทียมหรือระบบจำลองนิวรอลเน็ตเวอร์ค (Artificial Neural Network System : ANNS) โครงข่ายประสาทเทียมในทางทัศนคติของคอมพิวเตอร์แล้ว ก็จะประกอบด้วยส่วนประมวลผล ที่มีการเชื่อมต่อถึงกันหลาย ๆ ตัว ทำงานกันในลักษณะคู่ขนานคล้ายกับนิวรอลในสมองมนุษย์ เพื่อแปลงข้อมูลจากรูปหนึ่งเป็นอีกรูปหนึ่ง การใช้งานโครงข่ายประสาทเทียมจะเป็นไปในรูปแบบของการสอนแทนการป้อนโปรแกรม การสอนโดยให้ตัวเลขที่แสดงถึงน้ำหนักความสำคัญกำกับไว้ ณ จุดเชื่อมต่อต่าง ๆ ที่อยู่ในสถาปัตยกรรมของโครงข่าย ในขณะที่โครงข่ายประสาทเทียมทำงานอยู่ จะมีการปรับค่าตัวเลขที่แสดงน้ำหนักความสำคัญ โดยการใช้เกณฑ์ต่าง ๆ เข้ามาช่วยปรับเปลี่ยน จนทำให้ผลของเอาต์พุต นั้น ๆ ถูกต้องเป็นไปตามกลไกของโครงข่ายประสาทเทียมจนสามารถเรียนรู้และจดจำได้

2.4 โครงข่ายประสาทเทียม

โครงข่ายประสาทเทียม เป็นระบบการประมวลผลข่าว ที่มีลักษณะพิเศษที่แสดงการกระทำที่แน่นอนในพื้นฐานรวมโดยใช้โครงข่ายประสาทตามทัศนคติชีววิทยา โครงข่ายประสาทเทียมได้รับการพัฒนามาโดยลำดับ ตามวิธีการทางคณิตศาสตร์ที่มนุษย์ มีความรู้ความเข้าใจ โดยตั้งอยู่บนสมมุติฐานดังนี้

1) การประมวลผลข่าวที่เกิดขึ้นได้ จะประกอบด้วยสมาชิกจำนวนหนึ่ง ที่เรียกว่า นิวรอน
2) สัญญาณที่ถูกส่งผ่านระหว่างนิวรอน จะเกิดขึ้นได้ทั่วไป ณ ที่มีการเชื่อมต่อถึงกัน
3) ในแต่ละจุดของการเชื่อมโยง จะเกี่ยวข้องกับน้ำหนักด้วย ซึ่งขึ้นอยู่กับชนิดของโครงข่าย
4) ในแต่ละนิวรอน จะมีการปรับใช้ฟังก์ชันกระตุ้นด้วย (โดยปกติจะใช้ฟังก์ชันแบบไม่เป็นเชิงเส้น) ให้แก่อินพุตโครงข่าย ที่ส่งผลให้กำหนดเป็นสัญญาณเอาต์พุต

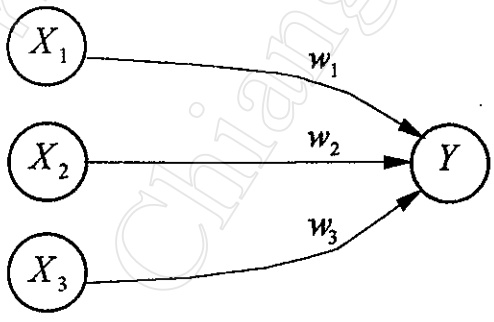
นิวรอลเน็ตเวอร์ค สามารถอธิบายลักษณะได้โดย รูปแบบของการเชื่อมต่อโครงข่ายระหว่างนิวรอน (เรียกว่า สถาปัตยกรรมของโครงข่ายประสาท) วิธีการของการกำหนดน้ำหนักบน

การเชื่อมต่อของโครงข่ายประสาท (เรียกว่า การฝึกสอน หรือ การเรียนรู้ ขั้นตอนวิธี) และฟังก์ชันกระตุ้น

โครงข่ายนิวรอล ประกอบด้วยสมาชิกประมวลผลจำนวนมาก ซึ่งเรียกว่า นิวรอน หน่วย เซลล์ หรือ โหนด และในแต่ละนิวรอนจะเชื่อมต่อกับนิวรอนอื่นด้วย นั้นหมายถึง การเชื่อมต่อสื่อสารกันโดยตรง ด้วยน้ำหนักที่สัมพันธ์กันในแต่ละส่วน น้ำหนักที่แสดงออกมาเป็นการเริ่มต้นที่ใช้ในการแก้ปัญหาในโครงข่ายแล้ว

ในแต่ละนิวรอนจะมีสถานะภายใน ถูกเรียกว่าการกระตุ้นของนิวรอน หรือระดับกิจกรรม ซึ่งเป็นฟังก์ชันของอินพุตที่มีการรับค่าเข้ามาแล้ว แบบอย่างคือ นิวรอนจะทำการส่งระดับการกระตุ้นของมัน ซึ่งคล้ายกับเป็นสัญญาณหนึ่ง ที่ส่งออกไปยังนิวรอนอื่นเป็นรายตัว โดยมีความสำคัญมากที่ทำให้นิวรอนนั้น ๆ สามารถส่งสัญญาณหนึ่งออกมาได้ในช่วงเวลาหนึ่ง ถึงแม้ว่าสัญญาณนั้นจะแพร่กระจายออกไปยังนิวรอนอื่นเป็นรายตัว ตัวอย่าง พิจารณานิวรอน Y ที่แสดงในรูปที่ 2.2 จะรับอินพุตจากนิวรอน X_1 X_2 และ X_3 การกระตุ้น (สัญญาณเอาต์พุต) ของนิวรอนเหล่านี้เป็น x_1 x_2 และ x_3 โดยลำดับ น้ำหนักบนการเชื่อมต่อจาก X_1 X_2 และ X_3 จนถึง Y คือ w_1 w_2 และ w_3 ตามลำดับ โดย y_in เป็นอินพุตโครงข่าย ให้แก่นิวรอน Y ซึ่งเป็นผลรวมของสัญญาณน้ำหนักที่ได้รับมาจากนิวรอน X_1 X_2 และ X_3

$$y_in = w_1x_1 + w_2x_2 + w_3x_3 \tag{2.1}$$

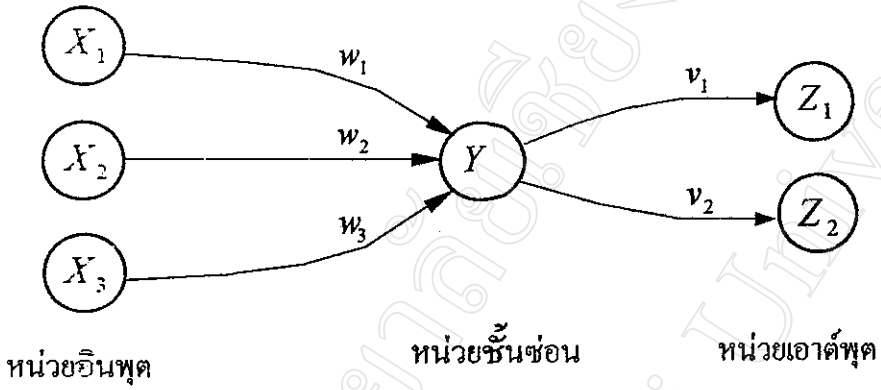


รูปที่ 2.2 ตัวอย่างนิวรอน [8]

y เป็นการกระตุ้นของนิวรอน Y ดังนั้นฟังก์ชันของอินพุตโครงข่าย คือ $y = f(y_in)$ ตัวอย่าง ถ้าเป็นฟังก์ชันซิกมอยด์โลจิสติก (Logistic Sigmoid) มีลักษณะเป็นเส้นโค้งรูป S มีรูปแบบฟังก์ชันดังนี้

$$f(x) = \frac{1}{1 + \exp(-x)} \tag{2.2}$$

ในรูปที่ 2.3 นิวรอน Y จะเชื่อมต่อกับนิวรอน Z_1 และ Z_2 โดยนิวรอน Y จะส่งสัญญาณของมันคือ y ไปยังแต่ละหน่วยที่ต่ออยู่ ปกติแล้วนิวรอน Z_1 และ Z_2 จะรับสัญญาณที่มีค่าแตกต่างกันทั้งนี้เพราะว่าแต่ละสัญญาณถูกกำหนดขนาดได้โดยการให้น้ำหนักอย่างเหมาะสมคือ v_1 และ v_2 ตามลำดับ สำหรับนิวรอน Z_1 และ Z_2 จะมีการกระตุ้นด้วย z_1 และ z_2 ตามลำดับ



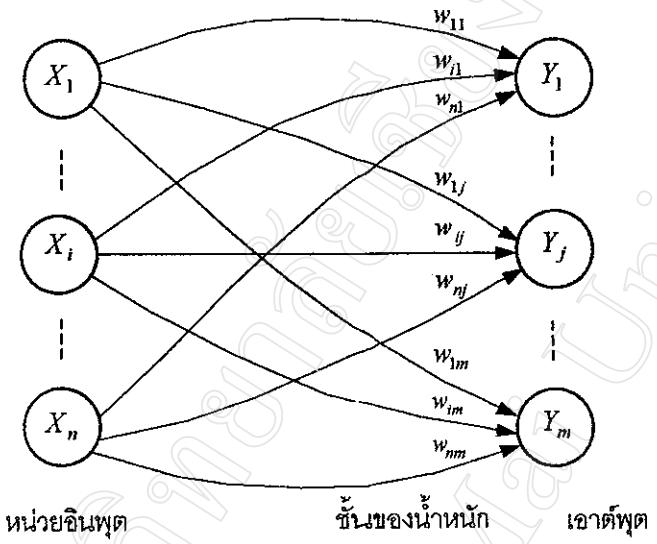
รูปที่ 2.3 ตัวอย่างนิวรอลเน็ตเวอร์ค [8]

2.5 สถาปัตยกรรมโครงข่ายประสาทเทียม

การจัดการของนิวรอนภายในชั้น รูปแบบการเชื่อมต่อภายใน และระหว่างชั้น เรียกว่า สถาปัตยกรรมโครงข่าย โครงข่ายนิวรอลทั้งหลายประกอบด้วยชั้นอินพุต ซึ่งภายในแต่ละหน่วย (Unit) จะได้รับการกระตุ้นเท่ากับสัญญาณอินพุตภายนอก โครงข่ายนิวรอลมักจะถูกจัดแบ่งเป็นโครงข่ายชั้นเดียว หรือหลายชั้น การกำหนดจำนวนของชั้น ส่วนที่เป็นหน่วยอินพุตจะไม่ถูกนับว่าเป็นหนึ่งชั้น ทั้งนี้เพราะว่าในส่วนนี้ไม่มีการคำนวณ สำหรับจำนวนของชั้นในโครงข่ายสามารถแบ่งได้เป็นจำนวนของชั้นน้ำหนักที่เชื่อมต่อระหว่างปีกของนิวรอน ดังแสดงในรูปที่ 2.3 จะมีชั้นของน้ำหนักแบ่งเป็นสองชั้น สถาปัตยกรรมของโครงข่ายนิวรอล สามารถจัดแบ่งได้ตามลักษณะโครงสร้างของโครงข่าย หรือจัดแบ่งตามกระบวนวิธีการเรียนรู้ ในหัวข้อนี้สามารถจำแนกตามลักษณะโครงสร้าง ที่ประกอบเป็นโครงข่ายนิวรอล ได้เป็นสามประเภทดังนี้

2.5.1 โครงข่ายนิวรอลชนิดชั้นเดียว (Single Layer)

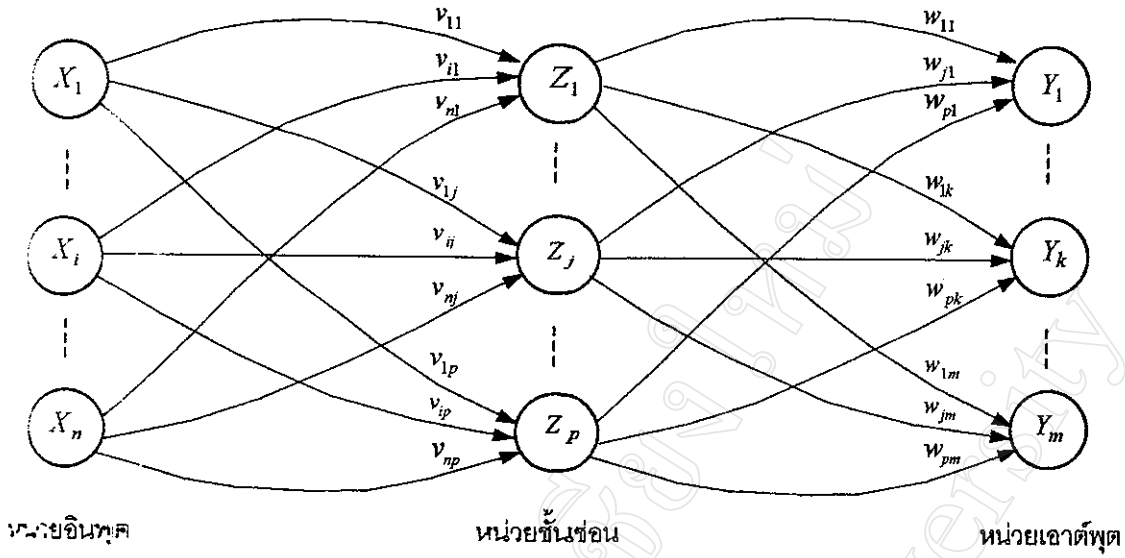
สถาปัตยกรรมของโครงข่ายนิวรอลชนิดนี้จะมีน้ำหนักเชื่อมต่อหนึ่งชั้น ดังแสดงในรูปที่ 2.4 ประกอบด้วยหน่วย ($X_1 .., X_i .., X_n$) เชื่อมต่ออยู่กับหน่วยเอาต์พุต ($Y_1 .., Y_j .., Y_m$) โดยมีน้ำหนักหนึ่งชั้น เป็นส่วนที่ปรับระดับสัญญาณอินพุต ค่าน้ำหนักที่ได้จะเป็นอิสระต่อกัน ไม่ส่งผลกระทบต่อการปรับค่าน้ำหนักตัวอื่น จึงมักใช้กับการประมวลผลข้อมูลที่ไม่ซับซ้อนมากนัก



รูปที่ 2.4 โครงข่ายนิวรอลชนิดชั้นเดียว [8]

2.5.2 โครงข่ายนิวรอลชนิดหลายชั้น (Multi Layer)

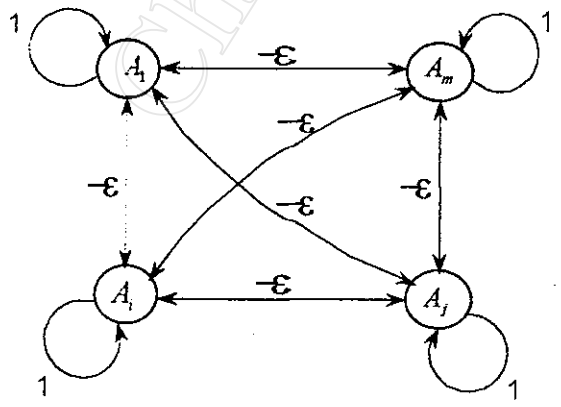
สถาปัตยกรรมของโครงข่ายนิวรอลชนิดนี้ เป็นโครงข่ายที่มีโหนดหลายชั้น เชื่อมต่อกันอยู่ระหว่างหน่วยอินพุตกับหน่วยเอาต์พุต ดังแสดงในรูปที่ 2.5 โดยมีชั้นซ่อนเป็นตัวเชื่อมระหว่างสองหน่วย โครงข่ายชนิดหลายชั้นสามารถใช้วิเคราะห์ปัญหาที่ซับซ้อนได้ดีกว่าโครงข่ายชนิดชั้นเดียว แต่การฝึกสอนอาจจะยุ่งยากกว่า ถึงอย่างไรก็ตามในการฝึกสอนอาจได้รับผลสำเร็จมากกว่า เป็นเพราะว่าในบางครั้งการแก้ปัญหาของโครงข่ายชนิดชั้นเดียว ไม่สามารถรับการฝึกสอนได้อย่างถูกต้องทุกปัญหา



รูปที่ 2.5 โครงข่ายนิวรอลชนิดหลายชั้น [8]

2.5.3 โครงข่ายนิวรอลชนิดแข่งขัน (Competitive Layer)

ลักษณะชั้นแข่งขันในส่วนนี้จะมีจำนวนของนิวรอลเน็ตเวอร์คมาก การเชื่อมต่อภายในระหว่างนิวรอนด้วยกันในชั้นแข่งขันนี้มีสถาปัตยกรรม แสดงในรูที่ 2.6 โครงข่ายย่อยที่ประกอบกันเป็นโครงข่ายใหญ่ จะเชื่อมต่อกันภายในด้วยน้ำหนัก $(-\varepsilon)$ แต่ละโครงข่ายย่อยจะมีสถาปัตยกรรมภายในของตัวเอง เมื่อโครงข่ายย่อยประมวลผลภายในแล้ว ก็จะส่งสัญญาณเอาต์พุตไปยังสัญญาณอินพุตของ โครงข่ายอื่นโดยผ่านค่าน้ำหนัก $-\varepsilon$ จนกระทั่งเหลือเพียงโครงข่ายย่อยโครงข่ายเดียวที่ให้ค่าสัญญาณเอาต์พุตไม่เท่ากับศูนย์



รูปที่ 2.6 โครงข่ายนิวร อลชนิดชั้นแข่งขัน [8]

2.6 การฝึกสอนโครงข่ายประสาทเทียม

การกำหนดน้ำหนักของโครงข่าย มีวิธีการกำหนดค่าเริ่มต้นของน้ำหนัก (ในช่วงการฝึกสอน) เป็นการหาค่าน้ำหนักที่เหมาะสมให้แก่โครงข่าย ที่สามารถฝึกสอนให้โครงข่ายนิวรอล เกิดการเรียนรู้ได้ โดยการป้อนชุดข้อมูลอินพุต และเอาต์พุตตัวอย่างที่ถูกต้องให้กับนิวรอนแต่ละเซลล์ และนิวรอนจะมีกระบวนการปรับค่าน้ำหนัก ในระบบให้สอดคล้องกับกลุ่มตัวอย่างของชุดข้อมูลอินพุตที่จุดเชื่อมโยงค่าน้ำหนักจะถูกปรับแต่ง เป็นไปตามฟังก์ชันการทำงานของระบบ ตามลักษณะพิเศษของกลุ่มตัวอย่างที่ป้อนให้ ขั้นตอนนี้จำเป็นมากที่จะแบ่งแยกคุณลักษณะของโครงข่ายนิวรอล กระบวนการฝึกสอน จำแนกออกเป็น 2 กลุ่มใหญ่ ๆ คือ การฝึกสอนแบบซุปเปอร์ไวซ์ (Supervised Training) และการฝึกสอนแบบอันซุปเปอร์ไวซ์ (Unsupervised Training)

2.6.1 การฝึกสอนแบบซุปเปอร์ไวซ์

ในการกำหนดแบบโครงข่ายนิวรอลส่วนมาก ได้รับความสำเร็จจากการฝึกสอน ก็โดยการนำเสนออย่างเป็นลำดับของเวกเตอร์การฝึกสอน หรือแบบรูปที่ดี (Pattern) ในแต่ละเวกเตอร์เอาต์พุตเป้าหมาย ที่มีความสัมพันธ์กัน และได้ค่าน้ำหนักที่ผ่านการปรับแต่งได้อย่างเหมาะสมดี ในช่วงขั้นตอนวิธีการเรียนรู้ กระบวนการนี้เป็นที่รู้จักอย่างเดียวกันคือ การฝึกสอนแบบซุปเปอร์ไวซ์

การป้อนแบบรูปที่ดี เข้าไปยังเซลล์นิวรอน และสังเกตแบบรูปเอาต์พุตที่ได้จากการคำนวณกับแบบรูปที่ต้องการ โดยมีการกำหนดสมการหรือวิธีการปรับค่าน้ำหนักแบบต่าง ๆ เพื่อให้ได้แบบรูปเอาต์พุตที่ต้องการ

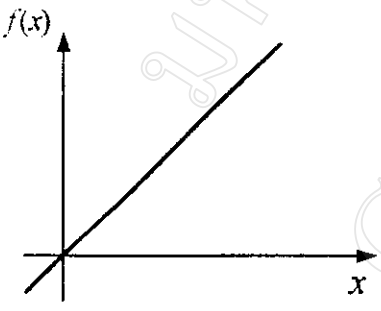
2.6.2 การฝึกสอนแบบอันซุปเปอร์ไวซ์

การฝึกสอนวิธีนี้ใช้หลักการปรับหาน้ำหนักด้วยตัวเอง ที่จุดเชื่อมต่อต่าง ๆ จะมีการเปรียบเทียบระหว่างค่าจากการคำนวณเอาต์พุตที่ถูกต้อง นั้นหมายถึงการปรับน้ำหนักนั้นจะไม่มีการกำหนดค่าสัญญาณที่ถูกต้องไว้ ว่าควรตั้งน้ำหนักในแต่ละจุดอย่างไร แต่อาศัยการป้อนกลับของสัญญาณ แล้วนำมาวิเคราะห์หาค่าของเซลล์นิวรอนเอง

2.6.3 ฟังก์ชันกระตุ้นโดยรวม

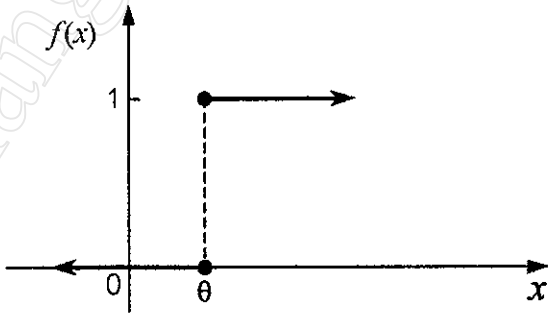
การดำเนินการ ขั้นพื้นฐานของเซลประสาทเทียม โดยรวมแล้วเกี่ยวข้องกับสัญญาณอินพุต ที่ทำให้น้ำหนักได้รับการปรับ และส่งผลไปยังเอาต์พุต สำหรับหน่วยอินพุต ในรูปที่ 2.7 (ก) ฟังก์ชันนี้ก็ถือว่าเป็นฟังก์ชันเอกลักษณ์ (Identity Function)

โครงข่ายชนิดชั้นเดียวส่วนมากจะใช้ฟังก์ชันแบบขั้น ในการแปลงอินพุตโครงข่าย ซึ่งมีค่าเปลี่ยนแปลงไปอย่างคงที่ เพื่อยังผลให้หน่วยเอาต์พุตนั้นเป็นไบนารี (Binary) ค่าที่ได้จะเป็น 1,0 หรือสัญญาณแบบไบโพลาร์ (Bipolar) มีค่าแบ่งเป็นขั้ว 1,-1 ฟังก์ชันแบบขั้นไบนารี เป็นที่รู้แล้วว่าเหมือนกันกับฟังก์ชันขีดเริ่มเปลี่ยน ให้ θ คือ ขีดเริ่มเปลี่ยน (Threshold) สำหรับฟังก์ชันซิกมอยด์ มีลักษณะเป็นเส้นโค้งรูป S ซึ่งเป็นฟังก์ชันกระตุ้นที่นำมาใช้ประโยชน์มาก สำหรับฝึกสอนให้กับโครงข่ายนิวรอล ขอบเขตของฟังก์ชันกระตุ้นที่แสดงในรูปที่ 2.7 (ค) ส่วนที่เป็นเส้นทึบนี้เรียกว่า "ซิกมอยด์ไบนารี" (Binary Sigmoid) ในรูปที่ 2.7 (ง) มีขอบเขตของฟังก์ชันอยู่ในช่วง –1,1 ซึ่งเรียกส่วนนี้ว่า "ซิกมอยด์ไบโพลาร์" (Bipolar Sigmoid) จากรูปจะเห็นได้ว่าฟังก์ชันกระตุ้นได้กำหนดให้ค่าเอาต์พุตอยู่ในช่วงระหว่าง –1 ถึง 1 และการสนองตอบระหว่างทั้งสองนี้ จะมีพารามิเตอร์ความสูงชัน $\sigma = 1$ ในการที่จะระบุว่า ฟังก์ชันกระตุ้นใดนั้น มีคุณสมบัติดีที่สุดคงเป็นไปไม่ได้ ก็เนื่องจากโครงข่ายประสาทเทียมมีโครงข่ายไม่แน่นอน นั้นก็หมายความว่า ในโครงข่ายหนึ่ง ที่ใช้กับปัญหาหนึ่งได้ดีที่สุดเท่านั้น เพราะฉะนั้นการเลือกฟังก์ชันกระตุ้นแบบใด จะต้องพิจารณาให้เหมาะสมกับปัญหานั้น ๆ ด้วย



$f(x) = x$ สำหรับทุก ๆ x

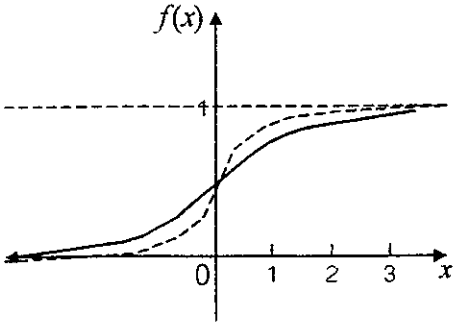
(ก)



$$f(x) = \begin{cases} 1 & \text{if } x \geq \theta \\ 0 & \text{if } x < \theta \end{cases}$$

(ข)

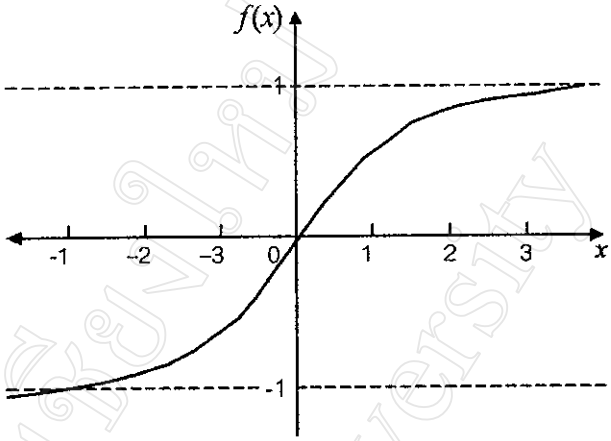
รูปที่ 2.7 (ก) ฟังก์ชันเอกลักษณ์ [10]
(ข) ฟังก์ชันแบบขั้นไบนารี [10]



$$f(x) = \frac{1}{1 + \exp(-\sigma x)}$$

พารามิเตอร์ความสูงชัน σ =1 และ σ =3

(ค)



$$g(x) = 2f(x) - 1 = \frac{2}{1 + \exp(-\sigma x)} - 1$$

(ง)

รูปที่ 2.7 (ค) Binary Sigmoid [10]
(ง) Bipolar Sigmoid [10]

2.6.4 แบบอย่างโดยรวมของการจดจำ

การจดจำต่อไปนี้จะใช้ในการพิจารณาสิ่งที่ได้ผ่านมาของลักษณะเฉพาะนิวรอลเน็ต เว้นแต่ว่าโครงข่ายเฉพาะที่ต่างไปจากแบบอื่น (ค่าที่ถือว่าเหมาะสมสำหรับพารามิเตอร์ที่ตั้งอยู่บนแบบจำลองนิวรอลเน็ตเฉพาะอย่าง ก็จะมีลักษณะที่ต่างกันไปขึ้นอยู่กับแต่ละแบบจำลอง)

x_i, y_j การกระตุ้นของหน่วย X_i, Y_j สำหรับหน่วยอินพุต X_i ก็จะมีสัญญาณอินพุตเป็น x_i ส่วนหน่วยอื่น Y_j จะมีสัญญาณอินพุตเป็น $y_j = f(y_in_j)$

w_{ij} เป็นน้ำหนักที่อยู่บนการเชื่อมต่อจากหน่วย X_i ไปยัง หน่วย Y_j

b_j ให้ไบแอส (Bias) บนหน่วย Y_j การให้ไบแอสกระทำเหมือนกับ ให้น้ำหนักบนการเชื่อมต่อกับหน่วย ด้วยการกระตุ้นค่าคงที่ มีค่าเป็น 1 (แสดงในรูปที่ 2.8)

y_in_j อินพุตโครงข่ายที่ส่งไปยังหน่วย Y_j กำหนดให้ $y_in_j = b_j + \sum_i x_i w_{ij}$

W เมตริกซ์น้ำหนัก ให้ $W = \{w_{ij}\}$

w_{1j} เป็นเวกเตอร์ของน้ำหนัก ให้ $w_{1j} = (w_{1j} + w_{2j},....,w_{nj})^T$ โดยที่ j เป็นลำดับที่ทางแนวตั้งของเมตริกซ์น้ำหนัก

$\|X\|$ ขนาดของเวกเตอร์ X

θ_j ขีดเริ่มเปลี่ยนสำหรับกระตุ้นนิวรอน Y_j เมื่อจัดให้ฟังก์ชันกระตุ้นแบบขั้นทำการกระตุ้นนิวรอน มีค่าเป็น 1 ก็ต่อเมื่ออินพุตโครงข่ายของมันมีขนาดสูงกว่าค่าขีดเริ่มเปลี่ยนที่กำหนดไว้เป็น θ_j และในทางตรงกันข้ามเมื่ออินพุตโครงข่ายของมันมีขนาดน้อยกว่า ก็จะได้ค่าการกระตุ้นเป็น 0 (แสดงในรูปที่ 2.7 ข)

s เวกเตอร์อินพุตฝึกสอน ให้ $s = (s_1,....,s_i,....,s_n)$

t เวกเตอร์เอาต์พุตฝึกสอน (หรือ เป้าหมาย) ให้ $t = (t_1,....,t_j,....,t_m)$

x เวกเตอร์อินพุต (สำหรับโครงข่ายที่มีการตอบรับ) ให้ $x = (x_1,....,x_i,....,x_n)$

Δw_{ij} การเปลี่ยนแปลงในน้ำหนัก w_{ij}

α อัตราการเรียนรู้ เป็นสิ่งที่ถูกนำมาใช้ในการควบคุมจำนวนของการปรับน้ำหนัก ในแต่ละขั้นตอนของการฝึกสอน

การคูณ เมตริกซ์สำหรับคำนวณอินพุตโครงข่าย

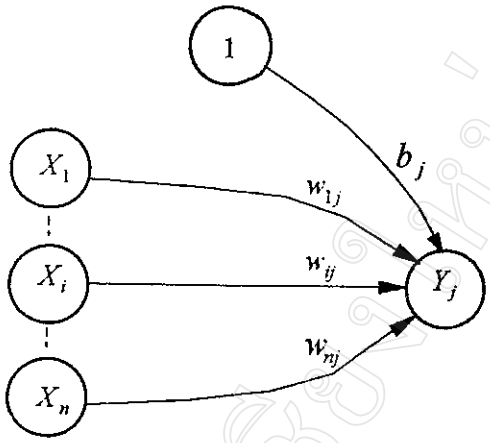
ถ้าน้ำหนักสำหรับโครงข่ายนิวรอล ได้มีการจัดเก็บในรูปเมตริกซ์ $W = (w_{ij})$ และอินพุตโครงข่ายที่เชื่อมโยงไปยังหน่วย Y_j (ไม่ได้ให้ไบแอสที่หน่วย j) จะได้จุดผลคูณของเวกเตอร์อย่างง่ายเป็น $x = (x_1,...,x_i,...,x_n)$ และน้ำหนัก w_{ij} (j เป็นลำดับที่ทางแนวตั้งของเมตริกซ์น้ำหนัก)

$$y_in_j = x.w_{ij}$$
$$= \sum_{i=1}^{n} x_i w_{ij} \qquad (2.3)$$

การให้ไบแอสสามารถทำได้โดยการเพิ่มส่วนประกอบ $x_o = 1$ เข้าไปยังเวกเตอร์ x ดังนั้น สมการจะเป็น $x = (1 + x_1,...,x_i,....,x_n)$ การให้ไบแอสเป็นการปรับเพื่อให้มีความถูกต้องเหมือนกันทุก ๆ น้ำหนัก มีสมการเป็น $w_{oj} = b_j$ อินพุตโครงข่ายที่ส่งให้หน่วย Y_j แสดงได้ดังนี้

$$y_in_j = x.w_{ij}$$
$$= \sum_{i=0}^{n} x_i w_{ij}$$
$$= w_{oj} + \sum_{i=1}^{n} x_i w_{ij}$$
$$= b_j + \sum_{i=1}^{n} x_i w_{ij} \qquad (2.4)$$

ในรูปที่ 2.8 แสดงความสัมพันธ์ระหว่าง การให้ไบแอสและจุดเริ่มเปลี่ยน



รูปที่ 2.8 นิวรอนกับการให้ไบแอส [8]

หน้าที่หนึ่งของโครงข่ายนิวรอลเมื่อได้รับการฝึกสอน คือเพื่อต้องการแบ่งแยกแบบรูป ปัญหาการแบ่งแยกแบบรูปคือ ในแต่ละเวกเตอร์อินพุต (แบบรูป) จะเป็นแบบช่วงยาว หรือไม่ยาว ทั้งนี้เพื่อต้องการแบ่งแยกลักษณะจำเพาะ หรือแบ่งแยกออกเป็นประเภท ๆ สำหรับโครงข่ายนิวรอล จะต้องกำหนดแบบรูปการฝึกสอนก่อน เพื่อจะได้แบ่งแยกสิ่งที่รู้จำได้อย่างถูกต้อง ในกรณีตัวอย่างที่พิจารณาในเรื่องของสมาชิกภายในกลุ่มหนึ่งที่หน่วยเอาต์พุตได้แสดงการเป็นสมาชิกภายในกลุ่มด้วยการสนองตอบด้วย 1 และ –1 (หรือ 0 ถ้าเป็นแบบไบนารี) ซึ่งแสดงถึงแบบรูปนั้น ๆ ไม่ได้เป็นสมาชิกของกลุ่ม ปัญหาการแบ่งแยกแบบรูปนี้สามารถแก้ไขได้โดย นิวรอลเน็ตเวอร์ค ที่มีวิธีการฝึกสอนโครงข่ายนิวรอลแบบชั้นเดียว

2.7 หลักการของโครงข่ายประสาทเทียมแบบหน่วยความจำร่วม

ในการแสดงขอบเขตของการเรียนรู้ เป็นกระบวนการของชนิดหน่วยร่วมที่ดั้นกลางด้วยแบบรูปร่วมกัน ลักษณะพิเศษที่สำคัญของหน่วยร่วม โดยสร้างการกระตุ้นให้หน่วยร่วมเข้าร่วมแสดงการตอบรับแบบรูป ตัวอย่างเช่น หน่วยร่วมคือกลุ่มของพวกเราที่ได้ผ่านการเรียนรู้ในการอ่านโน๊ตเพลง ดังนั้นพวกเราจึงเข้าร่วมกันจัดพิมพ์โน๊ตเพลงขึ้นมา เพื่อให้ระดับเสียงตรงกัน และสิ่งที่พวกเราจัดทำขึ้นมานี้ไม่ต้องการความประณีตอะไร เพียงแต่ให้มีแบบรูปคล้ายก็พอ อาจจะเขียนด้วยมือก็ได้ แต่เมื่อพวกเราเห็นแล้วก็ยังคงจำวิถีทางแบบเดิมได้ ในการนี้จะรู้ถึงความสัมพันธ์ของโครงข่ายนิวรอลที่สามารถเรียนรู้กลุ่มของคู่แบบรูป (หรือ หน่วยร่วม) โครงข่ายประสาทเทียมแบบหน่วยความจำร่วม เป็นโครงข่ายชนิดชั้นเดียว ในแต่ละหน่วยร่วมจะเป็นเวกเตอร์คู่นั้นคือเวกเตอร์

อินพุต-เอาต์พุต (s:t) ถ้าแต่ละเวกเตอร์ t เป็นเหมือนกันกับเวกเตอร์ s ก็ด้วยที่มันมีความสัมพันธ์ร่วมกัน ดังนั้นโครงข่าย จึงถูกเรียกว่าหน่วยความจำร่วมด้วยตนเอง (Autoassociative Memory) แต่ถ้าเวกเตอร์ t ต่างไปจากเวกเตอร์ s โครงข่ายก็จะถูกเรียกว่า Heteroassociative Memory ในแต่ละแบบโครงข่ายจะไม่มีวิธีการเรียนรู้เฉพาะตามชนิดคู่แบบรูป ที่ใช้ในการฝึกสอนเท่านั้น แต่ทั้งหมดนี้ก็ยังสามารถสนองตอบการเรียกแบบรูป กลับคืนมาได้ เมื่ออินพุตได้รับการกระตุ้นคล้ายกับของเดิม

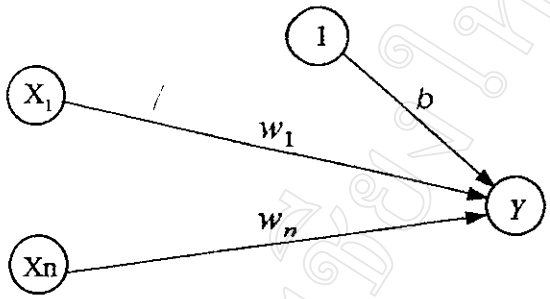
ก่อนทำการฝึกสอนโครงข่ายประสาทเทียมแบบหน่วยความจำร่วม แบบรูปต้นฉบับจะต้องทำการปรับเปลี่ยนให้เหมาะสมก่อนที่แสดงถึงการคำนวณ ตัวอย่างเช่น แบบรูปต้นฉบับ อาจประกอบด้วยสัญญาณ "เปิด" และ "ปิด" แล้วให้ทำการแปลงจาก "เปิด" $\rightarrow +1$ และ "ปิด" $\rightarrow 0$ (แสดงแบบไบนารี) หรือการแปลงจาก "เปิด" $\rightarrow +1$ และ "ปิด" $\rightarrow -1$ (แสดงแบบเชิงขั้ว) วิธีการฝึกสอนสำหรับโครงข่ายชนิดชั้นเดียว สามารถแบ่งได้ 2 วิธี คือ วิธีการฝึกสอนที่ใช้กฎแฮบบ์ (Hebb Rule) และกฎเดลต้า (Delta Rule) สำหรับสถาปัตยกรรมของโครงข่ายประสาทเทียมแบบหน่วยความจำร่วม อาจจะเป็นโครงข่ายแบบป้อนไปข้างหน้า หรือเป็นแบบปรากฏซ้ำ (วนซ้ำ) ในโครงข่ายแบบป้อนไปข้างหน้า มีรูปแบบการไหลสารสนเทศ จะเริ่มจากหน่วยอินพุต ส่งต่อไปยังหน่วยเอาต์พุต แต่สำหรับโครงข่ายแบบปรากฏซ้ำ มีการเชื่อมต่อระหว่างหน่วยทั้งหลายนั้น จะสร้างให้เป็นแบบวงวนซ้ำ

2.7.1 นิวรอน ADALINE

นิวรอนแบบหน่วยความจำร่วม ADALINE (ADAPtive Linear NEuron) มีรูปแบบการกระตุ้นเป็นแบบไบโพลาร์ (1 หรือ –1) ทั้งนี้เพื่อให้ค่าน้ำหนักบนจุดเชื่อมต่อสามารถทำการปรับค่าได้ ADALINE มีการให้ไบแอส ก็เพื่อทำการปรับน้ำหนักบนจุดเชื่อมต่อกัน ปกติให้การกระตุ้นด้วย 1 และการฝึกสอนจะใช้กฎเดลต้า หรือเป็นที่รู้กันว่า กฎ Windrow-Hoff ซึ่งใช้ได้กับโครงข่ายชนิดชั้นเดียว ในระหว่างทำการฝึกสอนการกระตุ้นของหน่วยจะเป็นการให้อินพุตโครงข่ายของมัน สำหรับหลังจากฝึกสอน ถ้านำโครงข่ายใช้เพื่อทำการแบ่งแยกแบบรูปในการรู้จำ เอาต์พุตจะได้รับการออกแบบเป็น +1 หรือ –1 ซึ่งเรียกว่าฟังก์ชันขีดเริ่มเปลี่ยน ถูกนำมาใช้กับอินพุตโครงข่าย ในกรณีที่อินพุตโครงข่าย ADALINE มีค่ามาก หรือเท่ากับ 0 การกระตุ้นกำหนดให้เป็น 1 และในทางตรงกันข้ามจะกำหนดให้เป็น –1 ปัญหาใด ๆ เพื่อต้องการให้แบบรูปอินพุตตรงกันกับค่าเอาต์พุต +1 เป็นการแบ่งแยกกันอย่างเป็นเชิงเส้นจากแบบรูปอินพุตที่ตรงกันกับค่าเอาต์พุต –1 นี้เป็นแบบอย่างที่สำเร็จได้ด้วยหน่วย ADALINE

สถาปัตยกรรม ADALINE

ADALINE แสดงนิวรอนหนึ่งเซลล์ ที่มีอินพุตรับสัญญาณมาจากหลาย ๆ หน่วย และมีการให้ไบแอสเป็น +1 แสดงในรูปที่ 2.9



รูปที่ 2.9 แสดงสถาปัตยกรรมของ ADALINE [8]

ขั้นตอนวิธี

ขั้นตอนวิธีการฝึกสอนสำหรับ ADALINE มีลำดับดังนี้

ขั้นตอนที่ 1 กำหนดน้ำหนักเริ่มต้น (ปกติใช้สุ่มค่าเริ่มต้นต่ำ ๆ)
กำหนดอัตราการเรียนรู้ α

ขั้นตอนที่ 2 ตรวจสอบเงื่อนไขการหยุดเมื่อเป็นเท็จ, ทำขั้นตอน 3-7

ขั้นตอนที่ 3 สำหรับแต่ละคู่ฝึกสอนแบบไบโพลาร์
แทนด้วย s : t ทำตามขั้นตอนที่ 4-6

ขั้นตอนที่ 4 กำหนดการกระตุ้นของหน่วยอินพุต, $(i = 1,\ldots,n)$:

$$x_i = s_i \tag{2.5}$$

ขั้นตอนที่ 5 คำนวณอินพุตโครงข่ายให้แก่หน่วยเอาต์พุต

$$y_in = b + \sum_i x_i w_i \tag{2.6}$$

ขั้นตอนที่ 6 ทำการปรับน้ำหนักและไบแอสให้ทันกาล $(i = 1,...,n)$:

$$b(\text{new}) = b(\text{old}) + \alpha(t - y_in) \tag{2.7}$$

$$w_i(\text{new}) = w_i(\text{old}) + \alpha(t - y_in)x_i \tag{2.8}$$

ขั้นตอนที่ 7 ตรวจสอบเงื่อนไขการหยุด :

ในกรณีที่น้ำหนักที่ทำการเปลี่ยนมีค่ามาก ให้ทำในขั้นตอนที่ 3 แต่ถ้ามีค่าน้อยกว่าที่กำหนดไว้ ให้หยุด

การกำหนดอัตราการเรียนรู้ให้มีค่าที่เหมาะสมนั้น ขึ้นอยู่กับความเหมาะสมซึ่งมีผลต่อความเร็วในการหาคำตอบ สำหรับนิวรอนเดียวกำหนดให้ขอบเขตสำหรับอัตราการเรียนรู้ α เป็น $0.1 \leq n\alpha \leq 1.0$ เมื่อ n เป็นจำนวนของหน่วยอินพุต

การประยุกต์ใช้

หลังจากทำการฝึกสอนแล้ว หน่วย ADALINE สามารถนำมาใช้ทำการแบ่งแยกแบบรูปอินพุตได้ โดยมีการประยุกต์ใช้ฟังก์ชันกระตุ้นสำหรับหน่วยเอาต์พุต มีลำดับขั้นตอนของฟังก์ชันสำหรับเป้าหมายแบบไบโพลาร์ ดังต่อไปนี้

ขั้นตอนที่ 1 กำหนดน้ำหนักเริ่มต้น

ขั้นตอนที่ 2 สำหรับแต่ละเวกเตอร์อินพุตไบโพลาร์ x , ทำตามขั้นตอนที่ 3-5

ขั้นตอนที่ 3 กำหนดการกระตุ้นของหน่วยอินพุต ให้กับ x

ขั้นตอนที่ 4 คำนวณอินพุตโครงข่ายให้แก่หน่วยเอาต์พุต

$$y_in = b + \sum_i x_i w_i \tag{2.9}$$

ขั้นตอนที่ 5 ปรับใช้ฟังก์ชันการกระตุ้น :

$$y = \begin{cases} 1 & \text{if } y_in \geq 0; \\ -1 & \text{if } y_in < 0 \end{cases} \tag{2.10}$$

ตัวอย่างการใช้ ADALINE กับฟังก์ชัน AND เมื่อ ให้อินพุตเป็น ไบนารี และเป้าหมายเป็นไบโพลาร์ พิจารณาฟังก์ชันที่ได้กำหนดไว้โดยมีแบบรูปฝึกสอน 4 แบบ ดังนี้

x_1	x_2	t
1	1	1
1	0	-1
0	1	-1
0	0	-1

การหาน้ำหนักที่ทำให้มีค่าผิดพลาดโดยรวมน้อยที่สุด ปรับใช้กฎเดลต้าเพื่อหาขอบเขตได้ดังนี้

$$E = \sum_{p=1}^{4} (x_1(p)w_1 + x_2(p)w_2 + w_0 - t(p))^2 \qquad (2.11)$$

เมื่อ $x_1(p)w_1 + x_2(p)w_2 + w_0$ เป็นอินพุตโครงข่ายที่ให้กับหน่วยเอาต์พุต สำรับแบบรูป p และ $t(p)$ คือ เป้าหมายร่วมสำหรับแบบรูป p

2.7.2 กฎแฮบบ์สำหรับหน่วยร่วมแบบรูป

กฎแฮบบ์เป็นวิธีการกำหนดน้ำหนักแบบธรรมดามาก เพื่อที่ใช้กับโครงข่ายประสาทเทียมแบบหน่วยความจำร่วม ซึ่งใช้ได้กับแบบรูปที่แสดงเป็นเวกเตอร์ไบนารี และเวกเตอร์แบบไบโพลาร์ สามารถแสดงขั้นตอนวิธี การฝึกสอน และวิธีดำเนินการสำหรับการหาน้ำหนัก ทำการพิจารณาในส่วนอินพุตที่ต่อไปยังโครงข่ายก่อนทำการฝึกสอนจะมีแบบรูปที่คล้ายกัน แต่ไม่เหมือนกัน มีสิ่งหนึ่งของอินพุตฝึกสอนที่ใช้แสดงคู่เวกเตอร์ฝึกสอน คือ s : t และมีเวกเตอร์อินพุตที่ใช้ทดสอบแทนด้วย x

ขั้นตอนวิธี

ขั้นตอนที่ 1 กำหนดน้ำหนักเริ่มต้นทั้งหมด เป็น $(i = 1,\ldots,n; j = 1,\ldots,m)$:

$$w_{ij} = 0 \qquad (2.12)$$

ขั้นตอนที่ 2 สำหรับแต่ละคู่เวกเตอร์อินพุตฝึกสอน-เอาต์พุตเป้าหมาย แทนด้วย s : t ทำตามขั้นตอนที่ 3-5

ขั้นตอนที่ 3 กำหนดค่ากระตุ้นสำหรับหน่วยอินพุต เพื่อจะให้เป็นอินพุตฝึกสอน $(i = 1,...,n)$:

$$x_i = s_i \tag{2.13}$$

ขั้นตอนที่ 4 กำหนดค่ากระตุ้นสำหรับหน่วยเอาต์พุต ให้เอาต์พุตเป้าหมาย $(j = 1,...,m)$:

$$y_j = t_j \tag{2.14}$$

ขั้นตอนที่ 5 ทำการปรับน้ำหนัก $(i = 1,...,n; j = 1,...,m)$:

$$w_{ij}(\text{new}) = w_{ij}(\text{old}) + x_i y_j \tag{2.15}$$

การหาน้ำหนัก

การหาน้ำหนักโดยการใช้กฎแฮบบ์ (น้ำหนักทุกค่าเริ่มต้นที่ 0) สามารถกำหนดได้ในเทอมผลคูณทั้งหมดของเวกเตอร์คู่อินพุต-เอาต์พุต ผลคูณทั้งหมดของสองเวกเตอร์ คือ $\mathbf{s} = (s_1,....,s_i,....,s_n)$ และ $\mathbf{t} = (t_1,....,t_j,....,t_m)$ ผลคูณเมตริกซ์ของ เมตริกซ์ n x 1 $\mathbf{S} = \mathbf{s}^T$ และเมตริกซ์ 1 x m กำหนดให้ $\mathbf{T} = \mathbf{t}$:

$$\mathbf{ST} = \begin{bmatrix} s_1 \\ \vdots \\ s_i \\ \vdots \\ s_n \end{bmatrix} [t_1..t_j..t_m] = \begin{bmatrix} s_1t_1 & \cdots & s_1t_j & \cdots & s_1t_m \\ \vdots & \cdot & \vdots & \cdot & \vdots \\ s_it_1 & \cdots & s_it_j & \cdots & s_it_m \\ \vdots & \cdot & \vdots & \cdot & \vdots \\ s_nt_1 & \cdots & s_nt_j & \cdots & s_nt_m \end{bmatrix} \tag{2.16}$$

เมตริกซ์น้ำหนักที่พอดีนี้เป็นการจัดเก็บของหน่วยร่วม **s:t** หาค่าได้โดยใช้กฎแฮบบ์ ในการจัดเก็บ กำหนดให้หน่วยร่วม $\mathbf{s}(p)$: $\mathbf{t}(p)$ เมื่อให้ $p = 1,....,P$ ที่ซึ่ง

$$\mathbf{s}(p) = (s_1(p),....,s_i(p),.....,s_n(p)) \text{ และ}$$
$$\mathbf{t}(p) = (t_1(p),....,t_j(p),.....,t_m(p)) \tag{2.17}$$

เมตริกซ์น้ำหนัก $W = \{w_{ij}\}$ กำหนดโดย

$$w_{ij} = \sum_{p=1}^{P} s_i(p)t_j(p) \tag{2.18}$$

ผลรวมของเมตริกซ์ผลคูณทั้งหมดนี้ ต้องการจัดเก็บในแต่ละหน่วยร่วม โดยทั่วไปแล้วพวกเราจะใช้สมการก่อนหน้านี้ หรือสร้างเมตริกซ์เวกเตอร์ที่รวบรัดกว่านี้ คือ

$$W = \sum_{p=1}^{P} s^T(p)\, t(p) \tag{2.19}$$

ทดสอบเรียกกลับคืน

ความเหมาะสมของการใช้กฎแฮบบ์ สำหรับปัญหาเฉพาะในการแสดงความเกี่ยวพันระหว่างเวกเตอร์ฝึกสอนอินพุต ถ้าให้เวกเตอร์อินพุตไม่ได้เกี่ยวพันกันเลย ตามกฎของแฮบบ์ จะยังผลให้น้ำหนักปรับให้ถูกต้องและเป็นการสนองตอบของโครงข่าย เมื่อได้รับการทดสอบกับเวกเตอร์ฝึกสอนหนึ่ง ก็จะมีการเรียกเป้าหมายที่ร่วมกันของเวกเตอร์อินพุตออกมาเทียบเคียงให้ได้ผลดีที่สุดออกมา แต่ถ้าเวกเตอร์อินพุตไม่เป็นทะแยงมุมกัน ผลการสนองตอบก็จะมีการจัดแบ่งค่าเป้าหมายออกเป็นส่วน ๆ ในการจัดเก็บเวกเตอร์ เหตุการณ์ดังกล่าวนี้เป็นการเรียกค่าเวกเตอร์สองส่วนกลับคืนมาซึ่งค่าทะแยงมุม ถ้าผลคูณกันเป็น 0 เมื่อสองเวกเตอร์เป็น $s(k)$ และ $s(p)$ เมื่อ $k \neq p$ สามารถเขียนรวมกันคือ $s(k)\, s^T(p) = 0$ หรือ

$$\sum_{i=1}^{n} s_i(k) s_i(p) = 0 \tag{2.20}$$

W เป็นเมตริกซ์น้ำหนักปัจจุบัน ซึ่งได้กำหนดค่าไว้ก่อนแล้ว ในการจัดเก็บของคู่เวกเตอร์อินพุตและเวกเตอร์เป้าหมาย การสนองตอบของโครงข่าย (กระทำด้วยฟังก์ชันเอกลักษณ์เพื่อกระตุ้นให้มีค่ามากกว่าฟังก์ชันขีดเริ่มเปลี่ยน) ให้เป็น $y = xW$ ถ้าสัญญาณอินพุต (การทดสอบ) เป็นเวกเตอร์อินพุตฝึกสอน kth ถ้า $x = s(k)$ แล้วการสนองตอบของโครงข่ายจะเป็น

$$\begin{aligned} s(k)W &= \sum_{p=1}^{P} s(k)\, s^T(p)\, t(p) \\ &= s(k)\, s^T(k)\, t(p) + \sum_{p \neq k} s(k)\, s^T(p)\, t(p) \end{aligned} \tag{2.21}$$

2.7.3 กฎเดลต้าสำหรับหน่วยร่วม

กฎเดลต้าเป็นกระบวนการเรียนรู้แบบวนซ้ำที่ถูกนำเสนอโดย Widrow และ Hoff สำหรับนิวรอน ADALINE (ADaptive Linear NEuron) กฎนี้ใช้สำหรับแบบรูปอินพุตที่เป็นแถวแนวอิสระต่อกัน ในการแก้ปัญหาเวกเตอร์อินพุตเราสามารถใช้โครงข่ายชนิดชั้นเดียว

กฎเดลต้าจะทำการเปลี่ยนน้ำหนักของการเชื่อมต่อนิวรอล ดังนั้นความแตกต่างที่ เหมาะสมค่าน้อยที่สุดระหว่างอินพุตโครงข่ายจนถึงหน่วยเอาต์พุต คือ y_in และ t เป็นค่าเป้าหมาย มีจุดมุ่งหมายก็เพื่อให้ค่าผิดพลาดทุก ๆ แบบรูปฝึกสอนมีค่าน้อยที่สุด ในลำดับการจัดการจะมีดรรชนีของน้ำหนักที่ต้องทำการปรับไปตามที่ได้กำหนด ในส่วนค่าที่ได้มาและดรรชนีของผลรวมที่ต้องการกำหนดให้ l เป็นดรรชนีน้ำหนัก และ i เป็นดรรชนีผลรวม ในกฎเดลต้าทำการปรับน้ำหนัก lth (สำหรับแต่ละแบบรูป) จะได้

$$\Delta w_l = \alpha(t - y_in)x_l \tag{2.22}$$

กำหนดให้

x เวกเตอร์กระตุ้นของหน่วยอินพุต (เวกเตอร์อินพุตฝึกสอน)

y_in อินพุตโครงข่ายเพื่อจะให้หน่วยเอาต์พุต Y เป็น $y_in = \sum_{i=1}^{n} x_i w_i$

t เอาต์พุตเป้าหมายสำหรับเวกเตอร์อินพุต x

α อัตราการเรียนรู้

ตามกฎเดลต้า เมื่อหน่วยเอาต์พุตได้แสดงถึงความสมดุลหลังจากได้รับการกระตุ้นแล้ว นั้นหมายถึงความแตกต่างระหว่างอินพุตโครงข่ายจนถึงหน่วยเอาต์พุตและค่าเป้าหมาย เหมาะสมที่สุด ดังนั้นใช้ y แสดงค่าเอาต์พุตที่คำนวณได้ สำหรับเวกเตอร์อินพุต x ดังสมการ

$$y_j = \sum_i x_i w_{ij} \tag{2.23}$$

และปรับน้ำหนักให้ทันกาล เป็น $(i = 1,\ldots,n\,;\, j = 1,\ldots,m)$

$$w_{ij}(\text{new}) = w_{ij}(\text{old}) + \alpha(t_i - y_j)x_i \tag{2.24}$$

ส่วนที่ได้มาคือค่าผิดพลาดสำหรับแบบรูปฝึกสอนเฉพาะ เป็น

$$E = (t - y_in)^2 \tag{2.25}$$

E เป็นฟังก์ชันของทุก ๆ น้ำหนัก w_i $(i = 1,\ldots,n)$ ความลาดชันของ E เกิดขึ้นจากเวกเตอร์บางส่วนที่ได้มาของ E ในแต่ละส่วนของน้ำหนัก ในความลาดชันนี้บ่งบอกถึงความผิดพลาดที่

เพิ่มขึ้น หรือลดลงอย่างรวดเร็วได้ ความผิดพลาดสามารถลดลงได้โดยการปรับน้ำหนัก w_i ไปในทิศทางของ $-\frac{\partial E}{\partial w_i}$

เมื่อ $y_in = \sum_{i=1}^{n} x_i w_i$,

$$\frac{\partial E}{\partial w_1} = -2(t - y_in)\frac{\partial y_in}{\partial w_1}$$

$$= -2(t - y_in)x_1 \qquad (2.26)$$

ดังนั้น ค่าผิดพลาดขณะนั้นจะถูกปรับลดลงอย่างรวดเร็วมาก (สำหรับกำหนดอัตราการเรียนรู้) โดยการปรับน้ำหนักตามกฎเดลต้า

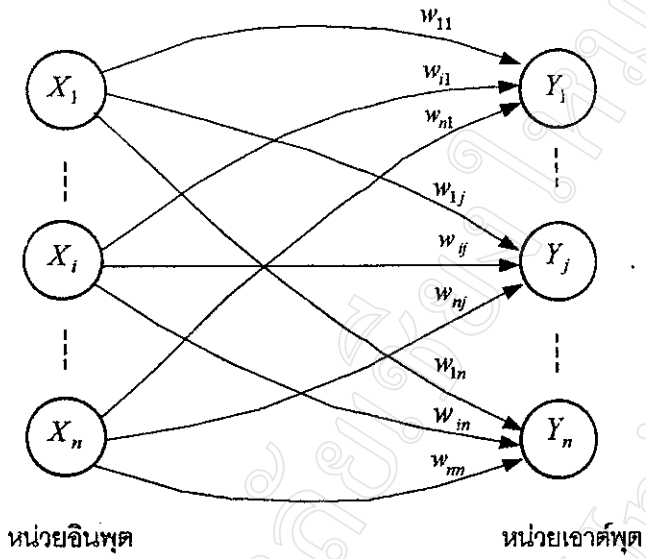
$$\Delta w_1 = \alpha(t - y_in)x_1 \qquad (2.27)$$

2.7.4 โครงข่ายหน่วยร่วมด้วยตัวเอง (Auto-Associative Net)

โครงข่ายหน่วยร่วมด้วยตัวเองแบบป้อนไปข้างหน้า ประกอบด้วยเวกเตอร์อินพุต ฝึกสอน และเวกเตอร์เป้าหมาย ถือว่าเป็นส่วนเดียวกัน กระบวนการฝึกสอนบ่อยครั้งถูกเรียกว่า การจัดเก็บเวกเตอร์ ซึ่งอาจจะเป็นแบบไบนารี หรือ แบบไบโพลาร์ เวกเตอร์ที่ถูกจัดเก็บไว้สามารถกู้กลับคืนมาได้จากส่วนที่ผิดเพี้ยนไป หรือ อินพุตบางส่วนที่ผิดรูปไป (สิ่งรบกวน) การกระทำของโครงข่ายจะตัดสินว่าถูกต้องหรือไม่ ก็ขึ้นอยู่กับความฉลาดของมันเองในการจำลองถอดแบบรูปที่จัดเก็บไว้แล้วออกมาจากอินพุตรบกวน โดยปกติในทางปฏิบัติเวกเตอร์ไบโพลาร์จะให้คุณสมบัติที่ดีกว่าเวกเตอร์แบบไบนารี สำหรับโครงข่ายหน่วยร่วมด้วยตัวเอง ค่าน้ำหนักจะถูกกำหนดให้เป็นศูนย์ในแนวทะแยงมุมของเมตริกซ์ (ความสัมพันธ์กันในองค์ประกอบแบบรูปอินพุตที่ตรงกันกับส่วนประกอบที่มีลักษณะเช่นเดียวกันในองค์ประกอบแบบรูปเอาต์พุต)

สถาปัตยกรรมโครงข่ายนิวรอล

สถาปัตยกรรมของโครงข่ายนิวรอลแบบหน่วยร่วมด้วยตัวเอง แสดงในรูปที่ 2.10



รูปที่ 2.10 โครงข่ายนิวรอลแบบหน่วยร่วมด้วยตัวเอง [8]

ขั้นตอนวิธี

กฎแฮบบ์ถูกนำมาใช้เพื่อทำการกำหนดค่าน้ำหนัก ให้กับโครงข่ายแบบหน่วยร่วมด้วยตัวเอง ก็เพราะว่าเวกเตอร์อินพุต และเวกเตอร์เอาต์พุต มีความเกี่ยวพันกันดี ในด้านของส่วนประกอบกับส่วนประกอบที่เหมือนกัน นั้นก็หมายถึงจำนวนของหน่วยเอาต์พุตเหมือนกันกับหน่วยอินพุต กำหนดลำดับขั้นตอนดังนี้

ขั้นตอนที่ 1 กำหนดน้ำหนักเริ่มต้นทั้งหมด เป็น $(i = 1,....,n; j = 1,....,m)$:

$$w_{ij} = 0 \tag{2.28}$$

ขั้นตอนที่ 2 สำหรับแต่ละเวกเตอร์ ที่ถูกจัดเก็บ ให้กระทำในขั้นตอนที่ 3-5

ขั้นตอนที่ 3 กำหนดค่ากระตุ้นสำหรับแต่ละหน่วยอินพุต $(i = 1,...,n)$:

$$x_i = s_i \tag{2.29}$$

ขั้นตอนที่ 4 กำหนดค่ากระตุ้นสำหรับแต่ละหน่วยเอาต์พุต $(j = 1,...,m)$:

$$y_j = s_j \tag{2.30}$$

ขั้นตอนที่ 5 ทำการปรับน้ำหนัก $(i = 1,...,n; j = 1,...,m)$:

$$w_{ij}(\text{new}) = w_{ij}(\text{old}) + x_i y_j \tag{2.31}$$

ในทางปฏิบัติค่าน้ำหนัก ที่หาได้จะใช้สมการดังนี้

$$\mathrm{W} = \sum_{p=1}^{P} s^T(p) t(p) \tag{2.32}$$

การประยุกต์ใช้

โครงข่ายนิวรอลแบบหน่วยร่วมด้วยตัวเอง ได้วางการตัดสินใจของโครงข่ายไว้แล้วด้วยเวกเตอร์อินพุต ให้เป็น "สิ่งที่รู้" (ซึ่งถูกจัดเก็บอยู่ในโครงข่าย) หรือ "สิ่งที่ไม่รู้" การรู้จำของโครงข่าย คือเวกเตอร์ "สิ่งที่รู้" รู้ได้โดยผลคูณแบบรูปของการกระตุ้นบนหน่วยเอาต์พุตที่โครงข่ายนั้น และภายในโครงข่าย มีเวกเตอร์หนึ่งที่จัดเก็บค่าไว้ก่อนแล้ว ซึ่งยังผลให้มีค่าที่เหมือนกัน วิธีดำเนินการประยุกต์ใช้ มีลำดับดังนี้ (ให้อินพุตแบบไบโพลาร์ และ การกระตุ้น)

ขั้นตอนที่ 1 กำหนดน้ำหนัก (โดยใช้กฎของแฮบบ์)
ขั้นตอนที่ 2 ในแต่ละเวกเตอร์อินพุตฝึกสอน ให้ทำตามขั้นตอนที่ 3-5
ขั้นตอนที่ 3 กำหนดการกระตุ้นของหน่วยอินพุตให้เท่ากัน ซึ่งต่อกับเวกเตอร์อินพุต
ขั้นตอนที่ 4 อินพุตโครงข่ายทำการคำนวณ เพื่อยังผลให้แต่ละหน่วยเอาต์พุต $(j = 1,...,n)$:

$$y_in_j = \sum_i x_i w_{ij} \tag{2.33}$$

ขั้นตอนที่ 5 ปรับใช้ฟังก์ชันกระตุ้น $(j = 1,...,n)$:

$$y_j = f(y_in_j) = \begin{cases} 1 & \text{if } y_in_j > 0; \\ -1 & \text{if } y_in_j \le 0. \end{cases} \tag{2.34}$$