

บทที่ 4

โครงข่ายประสาทเทียม

4.1 ความเป็นมาของโครงข่ายประสาทเทียม [14]

การพัฒนาโครงข่ายประสาทเทียมเกิดขึ้นเมื่อมีการเสนอแบบจำลองทางตรรกะของโครงข่ายประสาทเทียมเป็นครั้งแรกในปีค.ศ.1943 โดย Warren McCulloch และ Walter Pitts แนวคิดสำคัญคือการคำนวณค่าสัญญาณเอาต์พุตโดยพิจารณาจากระดับกระตุ้น กล่าวคือ นิวรอนจะให้สัญญาณเอาต์พุตก็ต่อเมื่อ ผลรวมของสัญญาณอินพุตมีค่ามากกว่าระดับกระตุ้นซึ่งแนวคิดนี้ยังคงเป็นส่วนสำคัญในการพัฒนาโครงข่ายประสาทเทียมมาจนทุกวันนี้

ในปี 1949 Dinal Hebb จากคณะฟิสิกส์มหาวิทยาลัยแมคกริลได้สร้างกฎการเรียนรู้ของโครงข่ายประสาทเทียมขึ้นเป็นครั้งแรก ทฤษฎีของ Hebb กล่าวว่าถ้านิวรอน 2 หน่วยทำงานพร้อมกันค่าน้ำหนักระหว่างสายการเชื่อมโยงของทั้งสองนิวรอนต้องลดลง

ในช่วงปี ค.ศ.1950 ถึง 1960 ถือว่าเป็นช่วงที่มีการพัฒนาโครงข่ายประสาทเทียมกันอย่างกว้างขวาง เมื่อ John van Neumann บิดาแห่งวงการคอมพิวเตอร์ยุคใหม่ได้นำเอาแบบจำลองของสมองที่เสนอโดยนักวิจัยที่ทำงานเกี่ยวกับโครงข่ายประสาทเทียมไม่ว่าจะเป็น Warren McCulloch และคนอื่น ๆ มาเป็นแนวทางในการพัฒนาเครื่องคอมพิวเตอร์ยุคต่อมา จนในปี ค.ศ.1956 ก็ได้มีการสร้างแบบจำลองโดยใช้คอมพิวเตอร์ขึ้นสำเร็จ

ในปี ค.ศ.1962 Frank Rosenblatt ได้เสนอทฤษฎีการเรียนรู้ของโครงข่ายที่เรียกว่า Perceptrons การเรียนรู้ใช้การปรับค่าน้ำหนักภายในโครงข่ายจนกระทั่งได้ค่าสัญญาณเอาต์พุตตรงกับสัญญาณเอาต์พุตตัวอย่าง

ในปี ค.ศ.1972 Kohonen ได้พัฒนาการเรียนรู้ของโครงข่ายด้วยการแก้ไขค่าน้ำหนักแบบเมตริกซ์ โดยอาศัยพื้นฐานจากทฤษฎีของ Hebb และในปีเดียวกันเขาก็ได้เสนอทฤษฎีการเรียนรู้ของโครงข่ายประสาทเทียมแบบ Self-organizing โดยการจัดรวมนิวรอนเป็นกลุ่มเพื่อใช้ในการวิเคราะห์เสียงพูดและต่อมา James Anderson ก็ได้พัฒนาโครงข่ายแบบ Brain-State-in-a-box เพื่อใช้ในการวินิจฉัยโรคและเรียนรู้สูตรคูณต่าง ๆ

ในช่วงปี ค.ศ.1985-1990 Carpenter และ Grossberg ได้เสนอทฤษฎี Adaptive resonance สำหรับวิเคราะห์ข้อมูลแบบไบนารี (ART1) และข้อมูลอินพุตแบบต่อเนื่อง (ART2) David Parker ได้เสนอทฤษฎีการเรียนรู้แบบแพร่ค่าย้อนกลับ ทฤษฎีการเรียนรู้ของเขาใช้การนำค่าผิดพลาดของสัญญาณเอาต์พุตที่คำนวณได้กับสัญญาณเอาต์พุตตัวอย่างกลับมาเป็นค่าแก้ไขน้ำหนัก

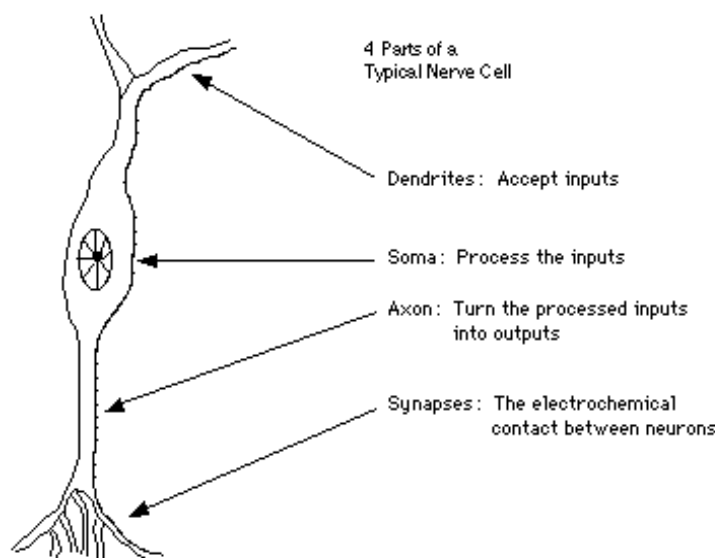
4.2 โครงข่ายประสาทตามทัศนคติชีววิทยา

แบบจำลองโครงข่ายประสาทเทียม ที่มีรายละเอียดไม่เหมือนกันกับระบบประสาทตามทัศนคติชีววิทยา ซึ่งก็แล้วแต่ให้ความสำคัญใดเป็นหลัก แต่ส่วนใหญ่แล้วจะเน้นความสามารถของโครงข่ายที่นำคุณสมบัติมาใช้งาน มากกว่าเหตุผลตามทัศนคติชีววิทยาของโครงข่าย ถึงแม้ว่าเกือบทั้งหมดจะให้ความสนใจเป็นอย่างมากกับความสามารถในการคำนวณของโครงข่ายประสาทเทียมก็ตามต่อไปนี้จะได้อธิบายถึงลักษณะบางอย่างของโครงข่ายประสาทเทียมตามทัศนคติชีววิทยา ซึ่งอาจจะช่วยให้เข้าใจอย่างชัดเจนถึงลักษณะพิเศษที่สำคัญของโครงข่ายประสาทเทียม เพื่อเข้าถึงจุดแรกเริ่มสำหรับโครงข่ายประสาทเทียม ในระบบโครงข่ายประสาทเทียมตามทัศนคติชีววิทยาได้ให้แนวคิดลักษณะสำคัญถึงสิ่งที่เป็ประโยชน์ต่อการคำนวณอย่างพิเศษ อุปมาเหมือนมีความใกล้เคียงกันมากระหว่างโครงสร้างของโครงข่ายประสาทเทียมตามทัศนคติชีววิทยา (สมอง หรือ เซลล์ประสาท) และส่วนประมวลผล (หรือโครงข่ายประสาทเทียม) ในความเป็นจริงแล้วโครงสร้างของโครงข่ายประสาทเทียมแต่ละเซลล์จะเปลี่ยนแปลงน้อยมาก

โครงข่ายประสาทเทียมตามทัศนคติชีววิทยา มีส่วนประกอบที่สำคัญและให้ความสนใจเป็นพิเศษ เพื่อที่จะทำความเข้าใจถึงโครงข่ายประสาทเทียม มีด้วยกัน 3 ส่วน คือ เดนไดรต์ (Dendrite) โซมา (Soma) และแอกซอน (Axon) สำหรับ เดนไดรต์จะรับสัญญาณมาจากนิวรอนอื่น สัญญาณก็คือแรงกระตุ้นประจุไฟฟ้า ที่ถูกส่งผ่านเข้ามายังช่องไซแนปติก (Synaptic) นั้นหมายถึงการส่งผ่านตามกระบวนการทางเคมี พฤติการณ์ของการแปรเปลี่ยนส่งผ่านทางเคมี สัญญาณที่เข้ามา มีวิธีการคล้ายกับพฤติกรรมของน้ำหนัในโครงข่ายประสาทเทียม

โซมา หรือลำตัวเซลล์ จะทำหน้าที่รวมสัญญาณที่ส่งผ่านเข้ามา ภายในตัวของโซมาจะทำให้เกิดแรงที่ต้องการส่งสัญญาณข้ามผ่านบนแอกซอนไปยังเซลล์อื่น การส่งสัญญาณจากนิวรอนได้เป็นผลสำเร็จนั้นก็ด้วยพฤติกรรมที่เป็นผลระดับความดันของไฟฟ้าจากการเชื่อมต่อที่ต่างกันของไอออน (Ion) บนแอกซอนของนิวรอนอีกด้านหนึ่ง ในรูปที่ 4.1 แสดงโครงข่ายประสาทเทียมตามทัศนคติชีววิทยาแบบทั่วไป เงื่อนไขที่พิเศษหลายอย่างของการประมวลผลในโครงข่ายประสาทเทียม ที่สำเร็จได้ด้วย คุณสมบัติของโครงข่ายประสาทเทียมตามทัศนคติชีววิทยา มีดังนี้

1. ส่วนประมวลผล ที่รับสัญญาณเข้ามาหลายส่วน
2. สัญญาณอาจถูกแปรเปลี่ยนโดยน้ำหนัก ณ ที่การรับไซแนป
3. ส่วนประมวลผล ทำการรวมอินพุตน้ำหนัก
4. ภายใต้พฤติกรรมที่เหมาะสม นิวรอนจะส่งสัญญาณเอาต์พุต
5. เอาท์พุตจากนิวรอนที่พิเศษอาจจะส่งไปยังนิวรอนอื่นได้หลายเซลล์ (ตามกิ่งแขนงแอกซอน)



รูปที่ 4.1 แสดงนิวรอนตามทัศนคติชีววิทยาแบบทั่วไป

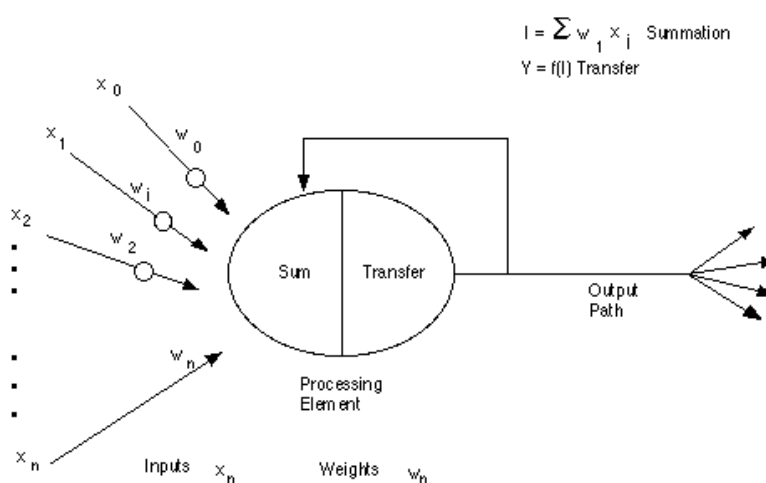
4.3 โครงข่ายประสาทเทียมตามทัศนคติของคอมพิวเตอร์

ปัจจุบันวิศวกรและนักวิทยาศาสตร์แขนงต่างๆ ได้พยายามพัฒนาเครื่องคอมพิวเตอร์ให้มีความฉลาดมากยิ่งขึ้น โดยพยายามให้เครื่องคอมพิวเตอร์มีการเรียนรู้และสามารถทำการตัดสินใจได้เอง หรือเป็นส่วนหนึ่งที่จะช่วยในการตัดสินใจในขั้นต้น ระบบการจำลองโครงข่ายประสาทเทียมหรือระบบจำลองนิวรอนเน็ตเวิร์ค (Artificial Neural Network System: ANNS) โครงข่ายประสาทเทียมในทางทัศนคติของคอมพิวเตอร์แล้ว ก็จะประกอบด้วยส่วนประมวลผล ที่มีการเชื่อมต่อถึงกันหลาย ๆ ตัว ทำงานกันในลักษณะคู่ขนานคล้ายกับนิวรอนในสมองมนุษย์ เพื่อแปลงข้อมูลจากรูปหนึ่งเป็นอีกรูปหนึ่ง การใช้งานโครงข่ายประสาทเทียมจะเป็นไปในรูปแบบของการสอนแทนการป้อนโปรแกรม การสอนโดยให้ตัวเลขที่แสดงถึงน้ำหนักความสำคัญกำกับไว้ ณ จุดเชื่อมต่อต่างๆ ที่อยู่ในสถาปัตยกรรมของโครงข่าย ในขณะที่โครงข่ายประสาทเทียมทำงานอยู่ จะมีการปรับค่าตัวเลขที่แสดงน้ำหนักความสำคัญ โดยการใช้เกณฑ์ต่าง ๆ เข้ามาช่วยปรับเปลี่ยน จนทำให้ผลของเอาต์พุตนั้น ๆ ถูกต้องเป็นไปตามกลไกของโครงข่ายประสาทเทียมจนสามารถเรียนรู้และจดจำได้

4.4 โครงข่ายประสาทเทียม (Artificial Neural Networks)

โครงข่ายประสาทเทียม (Artificial Neural Networks) เป็นระบบประมวลผลข้อมูลที่มีคุณลักษณะการทำงานเทียบได้กับโครงข่ายประสาทจริงทางชีววิทยา โครงข่ายประสาทเทียมนี้ได้พัฒนาการรู้คิดหรือระลึกจำของมนุษย์ให้เป็นแบบจำลองทางคณิตศาสตร์ ภายใต้สมมติฐานดังนี้

1. โครงข่ายประสาทเทียม ประกอบด้วยหน่วยประมวลผลจำนวนมากมาย เรียกว่า นิวรอน (Neuron, Unit, Cell หรือ Nodes) การประมวลผลข้อมูลจะเกิดขึ้นได้ที่นิวรอนนี้
2. สัญญาณต่าง ๆ ส่งผ่านนิวรอนได้ด้วยการเชื่อมต่อกันเป็นลิงค์ (Link)
3. แต่ละลิงค์จะมีความสัมพันธ์กับค่าน้ำหนัก (Weight) ซึ่งเปรียบเสมือนเป็นค่าที่ให้โครงข่ายในการตัดสินใจแก้ปัญหา โดยทั่วไป ค่าน้ำหนักนี้จะถูกคูณเข้ากับสัญญาณอินพุตเพื่อส่งออกไปยังนิวรอนอื่น ๆ
4. แต่ละนิวรอนจะมีค่าสถานะภายใน (Internal State) เรียกว่าระดับการกระตุ้น (Activation หรือ Activity Level) ที่จะให้ฟังก์ชันการกระตุ้น (Activation Function) ของสัญญาณอินพุตผลลัพธ์ เพื่อส่งเป็นสัญญาณเอาต์พุตออกไปให้นิวรอนตัวอื่น ๆ ฟังก์ชันการกระตุ้นนี้



รูปที่ 4.2 แสดงโครงสร้างพื้นฐานของโครงข่ายประสาทเทียม

4.4.1 คุณลักษณะสมบัติของโครงข่ายประสาทเทียม

จากหลักการเบื้องต้นของโครงข่ายประสาทเทียม จะเห็นได้ว่าคุณลักษณะของโครงข่ายประสาทเทียมจะถูกกำหนดโดย

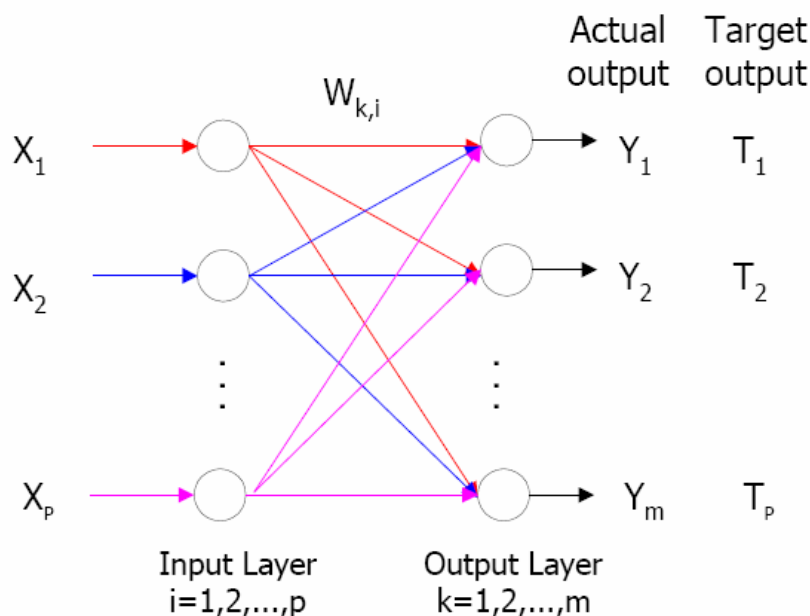
1. รูปแบบของการเชื่อมต่อ (Connection) ระหว่างนิวรอน เรียกว่า สถาปัตยกรรมโครงข่าย (Net Architecture)
2. วิธีการหาค่าน้ำหนักบนการเชื่อมต่อ เรียกว่าการเรียนรู้ (Training, Learning หรือ Algorithm)
3. ฟังก์ชันการกระตุ้น (Activation Function)

4.4.2 สถาปัตยกรรมของโครงข่ายประสาทเทียม

สถาปัตยกรรมของโครงข่ายประสาทเทียม สามารถจำแนกได้ตามลักษณะโครงสร้างของโครงข่าย หรือจำแนกตามกระบวนการเรียนรู้ ซึ่งในหัวข้อนี้จะจำแนกตามลักษณะโครงสร้างที่ประกอบกันเป็นโครงข่ายประสาทเทียม สามารถจำแนกออกได้เป็นสามประเภท ดังนี้

4.4.2.1 โครงข่ายประสาทเทียมชั้นเดียว (Single Layer)

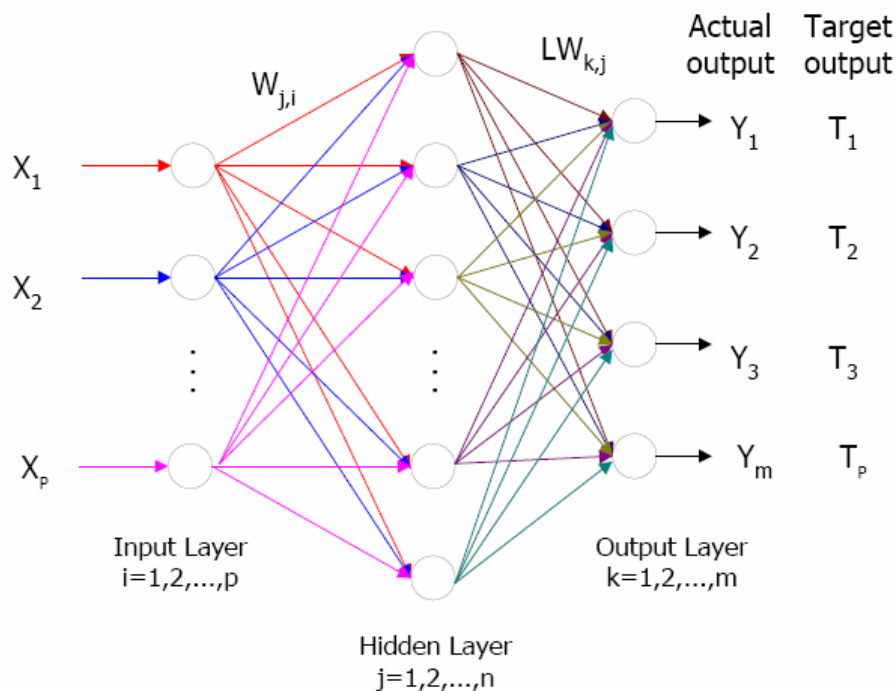
สถาปัตยกรรมโครงข่ายประสาทเทียมชนิดนี้ ประกอบด้วยอินพุตนิวรอน ($X_1, X_2, \dots, X_p$) เชื่อมต่ออยู่กับเอาต์พุตนิวรอน ($Y_1, Y_2, \dots, Y_m$) โดยมีค่าน้ำหนัก ($W_{k,i}$) เป็นตัวปรับระดับสัญญาณอินพุต ดังรูปที่ 4.3 แสดงโครงสร้างของโครงข่ายประสาทเทียมแบบชั้นเดียว ส่วนค่าน้ำหนักที่คำนวณได้จากเอาต์พุต ถือว่าเป็นคำตอบของโครงข่ายประสาทเทียม จะสังเกตได้ว่าโครงข่ายประสาทเทียมชนิดชั้นเดียวนี้มีค่าน้ำหนักเชื่อมอยู่ระหว่างอินพุตและเอาต์พุตเพียงชั้นเดียว ค่าน้ำหนักแต่ละค่าจะเป็นอิสระต่อกัน ไม่ส่งผลกระทบต่อค่าการปรับค่าน้ำหนักตัวอื่น จึงมักใช้กับการประมวลผลข้อมูลที่ไม่ซับซ้อนมากนัก



รูปที่ 4.3 แสดงโครงสร้างของโครงข่ายประสาทเทียมแบบชั้นเดียว

4.4.2.2 โครงข่ายประสาทเทียมชนิดหลายชั้น (Multi Layer)

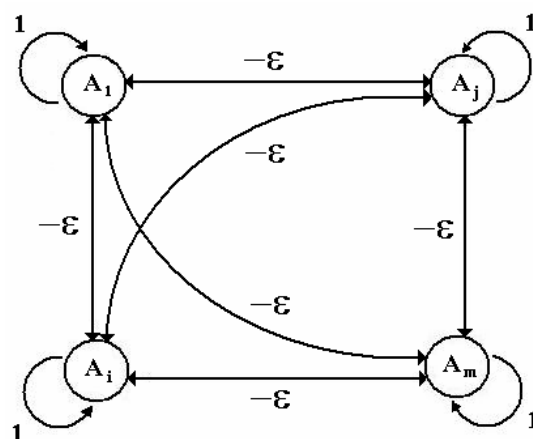
สถาปัตยกรรมโครงข่ายประสาทเทียมชนิดนี้ ประกอบด้วยค่าน้ำหนักมากกว่าหนึ่งชั้นขึ้นไป โครงข่ายประสาทเทียมชนิดนี้มีชั้นซ่อนเป็นตัวเชื่อมระหว่างชั้นอินพุตและชั้นเอาต์พุต โครงข่ายประเภทนี้สามารถใช้วิเคราะห์ปัญหาที่มีความซับซ้อนได้ดีกว่าชนิดชั้นเดียว แต่กระบวนการเรียนรู้ก็จะยุ่งยากมากกว่า รูปที่ 4.4 แสดงโครงสร้างของโครงข่ายประสาทเทียมแบบหลายชั้น



รูปที่ 4.4 แสดงโครงสร้างของโครงข่ายประสาทเทียมแบบหลายชั้น

4.4.2.3 โครงข่ายประสาทเทียมชนิดคอมเพทิทีฟ (Competitive Layer)

สถาปัตยกรรมโครงข่ายประสาทเทียมชนิดนี้ เป็นการเชื่อมโยงโครงข่ายหลาย ๆ โครงข่ายเข้าด้วยกัน โครงข่ายย่อย (A_1, A_i, A_j, A_k) ที่ประกอบกันเป็นโครงข่ายใหญ่ จะเชื่อมต่อกันด้วยน้ำหนักซึ่งเป็นค่าคงที่ ($-\epsilon$) แต่ละโครงข่ายย่อยมีสถาปัตยกรรมภายในของตัวเอง ไม่จำเป็นต้องเหมือนกันและเมื่อโครงข่ายย่อยประมวลผลภายในแล้ว จะส่งสัญญาณเอาต์พุตไปเป็นสัญญาณอินพุตของโครงข่ายอื่นผ่านค่าน้ำหนัก $-\epsilon$ จนกระทั่งเหลือเพียงโครงข่ายย่อยโครงข่ายเดียวที่ให้ค่าสัญญาณเอาต์พุตไม่เท่ากับศูนย์



รูปที่ 4.5 แสดงโครงสร้างของโครงข่ายประสาทเทียมชนิดคอมเพทิทีฟ

4.4.3 การเรียนรู้ของโครงข่าย

การกำหนดน้ำหนักของโครงข่าย มีวิธีการกำหนดค่าเริ่มต้นของน้ำหนัก (ในช่วงการฝึกสอน) เป็นการหาค่าน้ำหนักที่เหมาะสมให้แก่โครงข่าย ที่สามารถฝึกสอนให้โครงข่ายประสาทเทียมเกิดการเรียนรู้ได้ โดยการป้อนชุดข้อมูลอินพุตและเอาต์พุตตัวอย่างที่ถูกต้องให้กับนิวรอนแต่ละเซลล์และนิวรอนจะมีกระบวนการปรับค่าน้ำหนักในระบบให้สอดคล้องกับกลุ่มตัวอย่างของชุดข้อมูลอินพุตที่จุดเชื่อมโยงค่าน้ำหนักจะถูกปรับแต่งเป็นไปตามฟังก์ชันการทำงานของระบบ ตามลักษณะพิเศษของกลุ่มตัวอย่างที่ป้อนให้ ขั้นตอนนี้อาจเป็นมากที่จะแบ่งแยกคุณลักษณะของโครงข่ายประสาทเทียม กระบวนการเรียนรู้จะจำแนกออกเป็นกลุ่มใหญ่ ๆ ดังนี้

4.4.3.1 การเรียนรู้แบบควบคุม (Supervised Learning)

ในการกำหนดแบบโครงข่ายประสาทเทียมส่วนมาก ได้รับความสำเร็จจากการฝึกสอน ก็โดยการนำเสนออย่างเป็นลำดับของเวกเตอร์การฝึกสอนหรือรูปแบบที่ดี (Pattern) ในแต่ละเวกเตอร์เอาต์พุตเป้าหมายที่มีความสัมพันธ์กัน และได้ค่าน้ำหนักที่ผ่านการปรับแต่งได้อย่างเหมาะสมดี ในช่วงขั้นตอนวิธีการเรียนรู้ กระบวนการนี้เป็นที่รู้จักอย่างเดียวกัน คือ การเรียนรู้แบบควบคุม

4.4.3.2 การเรียนรู้แบบอิสระ (Unsupervised Learning)

การฝึกสอนวิธีนี้ใช้หลักการปรับน้ำหนักด้วยตัวเอง ที่จุดเชื่อมต่อต่างๆ จะมีการเปรียบเทียบระหว่างค่าจากการคำนวณเอาต์พุตที่ถูกต้อง นั้นหมายถึงการปรับน้ำหนักนั้นจะไม่มี การกำหนดค่าสัญญาณที่ถูกต้องไว้ ว่าควรตั้งน้ำหนักในแต่ละจุดอย่างไร แต่อาศัยการป้อนกลับของสัญญาณแล้วนำมาวิเคราะห์หาค่าของเซลล์นิวรอนเอง

4.4.4 ชนิดของโครงข่ายประสาทเทียม

จากที่ได้กล่าวมาแล้วเกี่ยวกับโครงข่ายประสาท ทั้งที่เป็นแบบจำลองและเป็นโครงข่ายของเซลล์ประสาทจริงของมนุษย์ จะมีการเชื่อมต่อกันของโหนดในลักษณะของโครงข่ายอย่างหนาแน่น เพื่อให้โครงข่ายสามารถเรียนรู้และจดจำสิ่งที่เรียนรู้มาแล้วได้ จากการเชื่อมโยงของโครงข่ายจะมีสองลักษณะด้วยกัน คือ

1) โครงข่ายที่ส่งสัญญาณแบบป้อนไปข้างหน้า (Feedforward Network) โครงข่ายชนิดนี้จะประกอบด้วยชั้นต่าง ๆ ของโครงข่าย โดยชั้นแรกจะเป็นชั้นอินพุต และชั้นสุดท้ายเป็นชั้นของเอาต์พุต ส่วนระหว่างทั้งสองชั้นจะมีหรือไม่มีก็ได้ จะมีชั้นที่แทรกอยู่ภายในอีกก็ได้ ซึ่งจะขึ้นอยู่กับขั้นตอนที่ใช้ในการฝึกสอนโครงข่าย เช่น ถ้าเป็นโครงข่ายเปอร์เซพตรอนแบบหลายชั้น (Multilayer Perceptron) ก็จะมีชั้นที่อยู่ระหว่างอินพุตกับเอาต์พุตอีก ซึ่งอาจจะมีมากกว่าหนึ่งชั้นก็ได้

ได้ ส่วนโครงข่าย Self – Organizing Map ของโคโฮเนนจะมีเพียงชั้นอินพุตกับเอาต์พุตเท่านั้น การเชื่อมต่อระหว่างชั้นของโครงข่ายแบบส่งสัญญาณป้อนไปข้างหน้า จะมีค่าน้ำหนักเป็นตัวเชื่อม และสัญญาณอินพุตที่เข้ามาจะถูกส่งไปตามทิศทางไปข้างหน้าจนถึงชั้นเอาต์พุต โดยไม่มีการป้อนกลับ

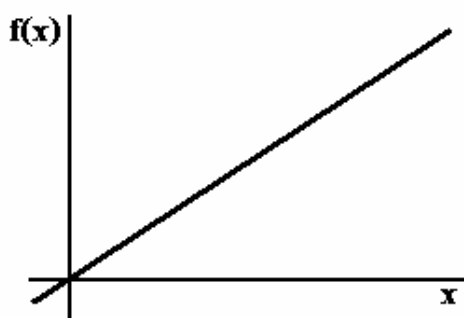
2) โครงข่ายที่ส่งสัญญาณแบบป้อนกลับ (Feedback Network) ในส่วนแรกของโครงข่ายนี้จะเป็นโครงข่ายแบบส่งสัญญาณไปข้างหน้าเหมือนกับแบบแรก แต่มีส่วนที่เพิ่มเข้ามา คือ ส่วนของการป้อนกลับ (Feedback)

4.5 ฟังก์ชันกระตุ้น(Activation function)

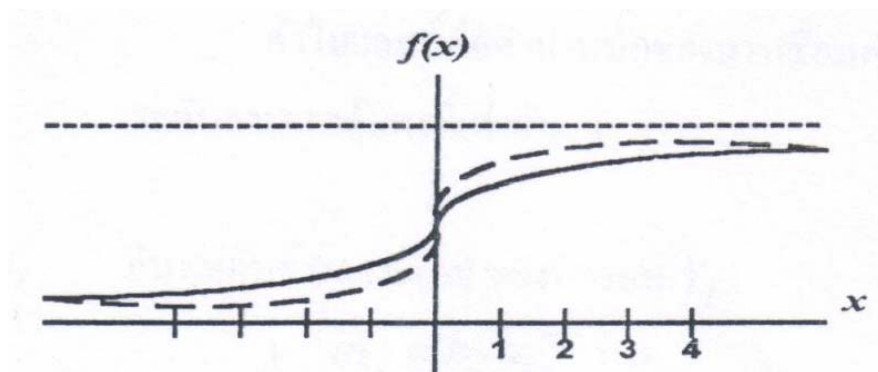
ฟังก์ชันกระตุ้นคือ ฟังก์ชันคำนวณค่าสัญญาณเอาต์พุตของนิวรอนโดยพิจารณาระดับผลรวมของสัญญาณอินพุตที่ส่งมายังนิวรอนแต่ละหน่วยสำหรับอินพุตนิวรอน ฟังก์ชันกระตุ้นจะมีลักษณะเป็น Identity function ดังรูปที่ 4.6

สมการของ Identity function : $f(x)=x$ for all x

ฟังก์ชันกระตุ้นที่ใช้กันมากอีกฟังก์ชันหนึ่งคือ ฟังก์ชันของซิกมอยด์ ฟังก์ชันนี้ให้ความละเอียดในการพิจารณาค่าเอาต์พุตสูงเพราะใช้ความสัมพันธ์ระหว่างสัญญาณอินพุตและค่าอนุพันธ์ของสัญญาณอินพุต ในกรณีที่ฟังก์ชันกระตุ้นของโครงข่ายมีขอบเขตของข้อมูลในการวิเคราะห์อยู่ระหว่าง 0 ถึง 1 หรือสัญญาณไบนารีเรียกว่า ฟังก์ชันไบนารีซิกมอยด์ (Binary sigmoid function) ดังรูปที่ 4.7

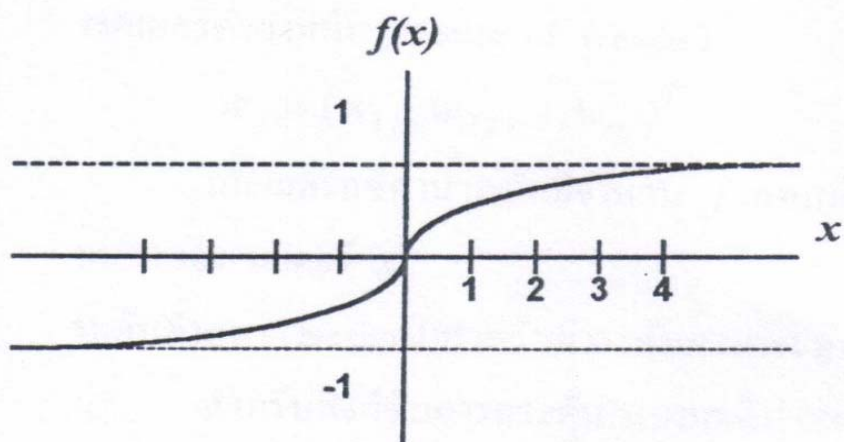


รูปที่ 4.6 แสดงไอเดนติตี้ฟังก์ชัน (Identity function)



รูปที่ 4.7 แสดงฟังก์ชันไบนารีซิกมอยด์

ในกรณีที่ต้องวิเคราะห์ข้อมูลซึ่งมีความซับซ้อนมาก การใช้ขอบเขตข้อมูลในช่วง 0 ถึง 1 อาจให้คำตอบที่ไม่ละเอียดพอ ซึ่งอาจทำให้เกิดความผิดพลาดต่อการคำนวณค่าสัญญาณเอาต์พุตได้จึงจำเป็นต้องขยายขอบเขตของข้อมูลเป็น -1 ถึง 1 เพื่อขยายช่วงความชันของกราฟให้มีความละเอียดมากขึ้น ฟังก์ชันของซิกมอยด์ที่ใช้วิเคราะห์ข้อมูลโดยมีขอบเขตอยู่ในช่วง -1 ถึง 1 เรียกว่า ฟังก์ชันไบโพลาร์ซิกมอยด์ (Bipolar sigmoid function) ดังรูปที่ 4.8



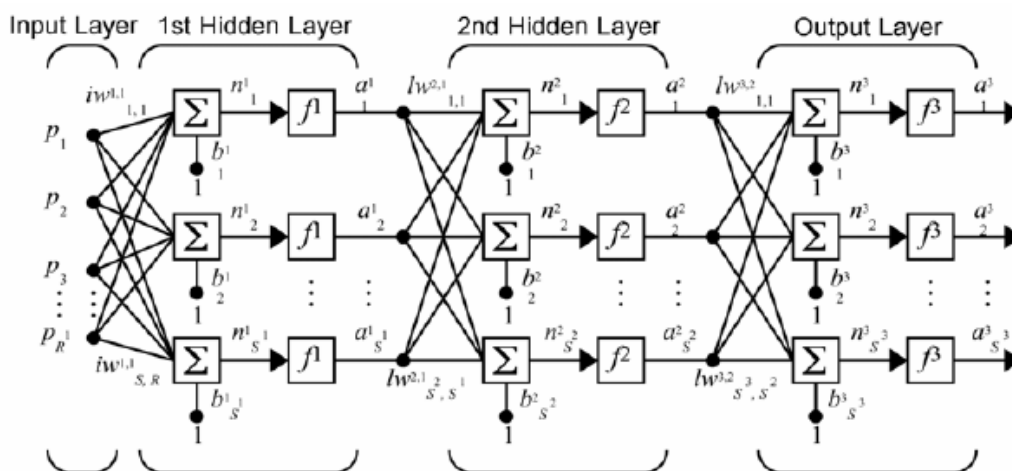
รูปที่ 4.8 แสดงฟังก์ชันไบโพลาร์ซิกมอยด์

ฟังก์ชันกระตุ้นของโครงข่ายประสาทเทียมมีคุณสมบัติสำคัญหลายประการคือ ต้องเป็นฟังก์ชันต่อเนื่อง (Continuous function) ให้ค่าผลลัพธ์จากการคำนวณเป็นค่าเฉพาะเพียงค่าเดียวและสามารถคำนวณค่าอนุพันธ์ของข้อมูลได้ ยิ่งไปกว่านั้นถ้าต้องการให้การคำนวณมีประสิทธิภาพควรเลือกสมการของฟังก์ชันกระตุ้นที่สามารถคำนวณค่าอนุพันธ์ได้ง่ายและในการปรับค่าน้ำหนักหรือไบอัสจะต้องคำนวณค่าอนุพันธ์ด้วยทุกครั้ง

4.6 โครงข่ายประสาทเทียมชนิดแพร่ค่าย้อนกลับ (Back-propagation NN)[15]

4.6.1 โครงสร้าง

โครงสร้างพื้นฐานของโครงข่ายประสาทเทียมชนิดแพร่ค่าย้อนกลับ (Back-propagation neural network : BP) ประกอบด้วยนิวรอนเซลล์ที่เรียงกันอยู่มากกว่า 2 ชั้น กล่าวคือมีได้ตั้งแต่ 3 ชั้นขึ้นไป ซึ่งประกอบด้วยชั้นอินพุท (Input layer), ชั้นซ่อน (Hidden layer) และ ชั้นเอาต์พุท (Output layer) ลักษณะของโครงข่ายชนิดนี้เป็นโครงข่ายที่ป้อนไปข้างหน้า โดยมีการเชื่อมโยงในแต่ละชั้นแบบต่อถึงกันหมด นั่นคือ ทุก ๆ นิวรอนในชั้นอินพุทจะส่งสัญญาณไปยังทุก ๆ นิวรอนในชั้นซ่อนชั้นแรกและทุก ๆ นิวรอนในชั้นซ่อนชั้นแรกจะส่งสัญญาณไปยังทุก ๆ นิวรอนในชั้นถัดไปจนในที่สุดทุก ๆ นิวรอนในชั้นซ่อนสุดท้ายจะส่งสัญญาณไปยังทุก ๆ นิวรอนในชั้นเอาต์พุท จากรูปที่ 4.9 แสดงโครงสร้างพื้นฐานของโครงข่ายที่มี 3 ชั้น



รูปที่ 4.9 แสดงแบบจำลองของโครงข่ายประสาทเทียมที่มี 2 ชั้นซ่อน [16]

รูปที่ 4.9 แสดงโครงข่ายประสาทเทียมที่มี 2 ชั้นซ่อน มีอินพุท R ชุด, นิวรอนในชั้นซ่อนที่ 1 และ 2 เท่ากับ S^1 และ S^2 และมีเอาต์พุท S^3 ชุด สมาชิกของอินพุทเวกเตอร์ P แต่ละตัวจะถูกเชื่อมเข้ากับนิวรอนแต่ละนิวรอนในชั้นซ่อนที่ 1 ด้วยค่าถ่วงน้ำหนัก W^1 (เรียกว่า Input weight, $IW^{1,1}$) ซึ่งเป็นเมทริกซ์ขนาด $S^1 \times R$ และเวกเตอร์ a^1 ซึ่งเป็นผลลัพธ์ของชั้นซ่อนที่ 1 และมีขนาด S^1 จะเป็นอินพุทของชั้นซ่อนที่ 2 สมาชิกแต่ละตัวของเวกเตอร์ a^1 ก็จะถูกเชื่อมเข้ากับนิวรอนแต่ละนิวรอนในชั้นซ่อนที่ 2 ด้วยค่าถ่วงน้ำหนัก W^2 (เรียกว่า Layer weight, $LW^{2,1}$) ซึ่งเป็นเมทริกซ์ขนาด $S^2 \times S^1$ อินพุทของชั้นข้อมูลเอาต์พุท สมาชิกแต่ละตัวของเวกเตอร์ a^2 จะถูกเชื่อมเข้ากับนิวรอนแต่ละตัวในชั้นข้อมูลเอาต์พุทด้วยค่าถ่วงน้ำหนัก W^3 ($LW^{3,2}$) ซึ่งเป็น

เมตริกซ์ขนาด $S^3 \times S^2$ โดยที่เวกเตอร์ a^3 ซึ่งเป็นเมตริกซ์ขนาด S^3 จะเป็นผลลัพธ์ของโครงข่ายประสาทเทียม

4.6.2 แนวคิด [14]

ขั้นตอนการฝึกสอนของโครงข่ายประสาทเทียมชนิดแพร่ค่าย้อนกลับ จะสามารถแบ่งออกเป็น 3 ขั้นตอนด้วยกัน คือ ป้อนข้อมูลอินพุตไปข้างหน้า, แพร่ค่าย้อนกลับของค่าผิดพลาดที่เกี่ยวข้องและปรับค่าถ่วงน้ำหนัก

ระหว่างป้อนไปข้างหน้า จะเริ่มต้นด้วยการนำค่าอินพุตที่ได้รับจากชั้นอินพุตมาคำนวณหาผลรวมที่ได้รับในแต่ละนิวรอน แล้วจึงส่งผลลัพธ์ได้ส่งต่อไปยังชั้นซ่อนที่ 1 แต่ละนิวรอนในชั้นซ่อนที่ 1 จะทำการคำนวณด้วยฟังก์ชันกระตุ้นในชั้นนั้น แล้วจึงส่งผลลัพธ์ที่ได้ส่งต่อไปยังชั้นซ่อนที่อยู่ถัดไป ในชั้นซ่อนที่ 2 แต่ละนิวรอนในชั้นซ่อนที่ 2 จะทำการคำนวณด้วยฟังก์ชันกระตุ้นในชั้นนั้น จากนั้นส่งผลลัพธ์ที่ได้ส่งต่อไปยังชั้นถัดไป สุดท้าย แต่ละนิวรอนในชั้นเอาต์พุตจะทำการคำนวณด้วยฟังก์ชันกระตุ้นในชั้นเอาต์พุต เพื่อหาคำตอบหรือผลตอบสนองของโครงข่ายสำหรับแต่ละรูปแบบข้อมูลที่ป้อนให้ ทิศทางการไหลของข้อมูลเหล่านี้จะเป็นลักษณะเคลื่อนไปข้างหน้าจากชั้นของอินพุต ผ่านชั้นของนิวรอนเซลล์ภายในไปสู่ชั้นเอาต์พุต วิธีการคำนวณของโครงข่ายลักษณะนี้ จึงมีชื่อตามลักษณะดังกล่าวว่า forward-propagation

ขั้นต่อไปจะเป็นการคำนวณแบบ backward-propagation สำหรับขั้นตอนนี้จะเป็นการนำผลลัพธ์ที่ได้จากเอาต์พุตของโครงข่ายมาเปรียบเทียบกับเอาต์พุตเป้าหมายเพื่อคำนวณค่าผิดพลาดที่เกี่ยวข้อง แล้วค่าผลต่างระหว่างเอาต์พุตของโครงข่ายกับเอาต์พุตเป้าหมายที่ได้จากโครงข่ายจะถูกส่งถอยหลังกลับมายังชั้นซ่อนที่ 2 และส่งต่อไปยังนิวรอนต่างๆ ภายในชั้นซ่อนที่ 2 แต่ละโนดจะได้รับค่าผิดพลาดเพียงบางส่วนขึ้นอยู่กับนิวรอนเซลล์นั้นเป็นตัวที่ส่งผลมากหรือน้อยไปสู่เอาต์พุตนั้น เพื่อปรับค่าถ่วงน้ำหนักระหว่างชั้นเอาต์พุตกับชั้นซ่อนที่ 2 กระบวนการของการส่งค่าความผิดพลาดกลับมานั้นจะทำซ้ำกับชั้นถัดลงมาอีกจนกระทั่งทุกโนดในโครงข่ายได้รับส่วนแบ่งค่าความผิดพลาดนั้น

ขั้นตอนสุดท้ายจะนำค่าผิดพลาดที่ได้มาใช้ในการปรับค่าถ่วงน้ำหนัก ซึ่งค่าถ่วงน้ำหนักจะเปลี่ยนไปมากหรือน้อยขึ้นอยู่กับปริมาณของค่าผิดพลาดที่ได้รับ กระบวนการจะกลับไปทำซ้ำในขั้นตอนแรกจนกระทั่งค่าผิดพลาดที่ได้มีค่าน้อยกว่าที่ค่าผิดพลาดต่ำสุดที่กำหนดไว้ต่อไปเป็นรายละเอียดของการทำงานในแต่ละขั้นตอน

1. แพร่ค่าป้อนไปข้างหน้า (Forward-propagation)

เมื่อข้อมูลอินพุตปรากฏบนชั้นอินพุต ($P_R, R=1, \dots, R$) สัญญาณจากชั้นอินพุตทั้งหมดจะถูกส่งไปตามสายเชื่อมโยงไปยังนิวรอนเซลล์ในชั้นถัดไป แต่ละนิวรอนในชั้นซ่อนที่ 1 ถูก

เชื่อมโยงด้วยค่าถ่วงน้ำหนัก ($iw_{S^1,R}^{1,1} = S^1 = 1, \dots, S^1$) จะทำการคำนวณสัญญาณอินพุตของ นิวรอนเซลล์ในชั้นซ่อนที่ 1 ได้จากสมการ

$$net_{S^1} = b_{S^1}^1 + \sum_{R=1}^R iw_{S^1,R}^{1,1} P_R \quad (4.1)$$

ใช้ฟังก์ชันกระตุ้นเพื่อคำนวณสัญญาณเอาต์พุต (a^1) ในชั้นซ่อนที่ 1

$$a^1 = f(net_{S^1}) \quad (4.2)$$

เมื่อสัญญาณเอาต์พุต (a^1) ในชั้นซ่อนที่ 1 เอาต์พุตที่ได้ส่งต่อไปยังชั้นซ่อนที่ 2 โดยแต่ละ นิวรอนในชั้นซ่อนที่ 2 ถูกเชื่อมโยงด้วยค่าถ่วงน้ำหนัก ($lw_{S^2,S^1}^{2,1} = S^2 = 1, \dots, S^2$) จะทำการ คำนวณสัญญาณอินพุตของนิวรอนเซลล์ในชั้นซ่อนที่ 2 ได้จากสมการ

$$net_{S^2} = b_{S^2}^2 + \sum_{S^1=1}^{S^1} lw_{S^2,S^1}^{2,1} a^1 \quad (4.3)$$

ใช้ฟังก์ชันกระตุ้นเพื่อคำนวณสัญญาณเอาต์พุต (a^2) ในชั้นซ่อนที่ 2

$$a^2 = f(net_{S^2}) \quad (4.4)$$

เมื่อสัญญาณเอาต์พุต (a^2) ในชั้นซ่อนที่ 2 เอาต์พุตที่ได้ส่งต่อไปยังชั้นเอาต์พุต โดยแต่ละ นิวรอนในชั้นเอาต์พุต ถูกเชื่อมโยงด้วยค่าถ่วงน้ำหนัก ($lw_{S^3,S^2}^{3,2} = S^3 = 1, \dots, S^3$) จะทำการ คำนวณสัญญาณอินพุตของนิวรอนเซลล์ในชั้นเอาต์พุตได้จากสมการ

$$net_{S^3} = b_{S^3}^3 + \sum_{S^2=1}^{S^2} lw_{S^3,S^2}^{3,2} a^2 \quad (4.5)$$

ใช้ฟังก์ชันกระตุ้นเพื่อคำนวณสัญญาณเอาต์พุต (a^3) ในชั้นเอาต์พุต

$$a^3 = f(net_{S^3}) \quad (4.6)$$

2) แพร่ค่าย้อนกลับของค่าผิดพลาด (Back-propagation of error)

ในแต่ละนิวรอนของชั้นเอาต์พุต ค่าผิดพลาดของโครงข่ายจะเป็นผลต่างระหว่างเอาต์พุตเป้าหมายกับเอาต์พุตของโครงข่าย ($t^3 - a^3$) ค่าผิดพลาดที่ได้จะถูกส่งกลับเข้ามายังโครงข่ายในรูปของสัญญาณค่าผิดพลาด (error signal) การหาสัญญาณค่าผิดพลาดที่ชั้นเอาต์พุต หาได้จากสมการ

$$\delta^3 = (t^3 - a^3) f'(net_{s_3}) \quad (4.7)$$

คำนวณค่าถ่วงน้ำหนักที่เกี่ยวข้อง (ใช้เพื่อปรับปรุ้ค่าถ่วงน้ำหนัก $lw_{s^3,s^2}^{3,2}$ หลังจากนี้) โดยที่ α คือ อัตราการเรียนรู้ (Learning rate)

$$\Delta lw_{s^3,s^2}^{3,2} = \alpha \delta^3 a^2 \quad (4.8)$$

คำนวณค่าไบอัส (bias) ที่เกี่ยวข้อง (ใช้เพื่อปรับปรุ้ $b_{s_3}^3$ หลังจากนี้)

$$\Delta b_{s_3}^3 = \alpha \delta^3 \quad (4.9)$$

และส่ง δ^3 ไปที่ชั้นซ่อนที่ 2

ในแต่ละนิวรอนของชั้นซ่อนที่ 2 การหาสัญญาณค่าผิดพลาดที่ชั้นซ่อนที่ 2 หาได้จากสมการ

$$\delta^2 = \sum_{s^3=1}^{s^3=3} \delta^3 lw_{s^3,s^2}^{3,2} f'(net_{s^2}) \quad (4.10)$$

คำนวณค่าถ่วงน้ำหนักที่เกี่ยวข้อง (ใช้เพื่อปรับปรุ้ค่าถ่วงน้ำหนัก $lw_{s^2,s^1}^{2,1}$ หลังจากนี้)

$$\Delta lw_{s^2,s^1}^{2,1} = \alpha \delta^2 a^1 \quad (4.11)$$

คำนวณค่าไบอัส (bias) ที่เกี่ยวข้อง (ใช้เพื่อปรับปรุ้ $b_{s^2}^2$ หลังจากนี้)

$$\Delta b_{s^2}^2 = \alpha \delta^2 \quad (4.12)$$

และส่ง δ^2 ไปยังชั้นช่อนที่ 1

ในแต่ละนิวรอนของชั้นช่อนที่ 1 การหาสัญญาณค่าผิดพลาดที่ชั้นช่อนที่ 1 หาได้จากสมการ

$$\delta^1 = \sum_{s^2=1}^{s^2} \delta^2 lw_{s^2,s^1}^{2,1} f'(net_{s^1}) \quad (4.13)$$

คำนวณค่าถ่วงน้ำหนักที่เกี่ยวข้อง (ใช้เพื่อปรับปรุงค่าถ่วงน้ำหนัก $\Delta lw_{s^1,R}^{1,1}$ หลังจากนี้)

$$\Delta lw_{s^1,R}^{1,1} = \alpha \delta^1 P_R \quad (4.14)$$

คำนวณค่าไบอัส (bias) ที่เกี่ยวข้อง (ใช้เพื่อปรับปรุง $b_{s^1}^1$ หลังจากนี้)

$$\Delta b_{s^1}^1 = \alpha \delta^1 \quad (4.15)$$

3. ขั้นตอนการปรับค่าถ่วงน้ำหนักและค่าไบอัส

ถ้าหากมีข้อผิดพลาดขึ้น ค่าผิดพลาดที่คำนวณได้จะถูกนำมาใช้ในการปรับน้ำหนักเพื่อให้โครงข่ายได้ผลลัพธ์ที่ถูกต้อง การปรับน้ำหนักสำหรับแต่ละนิวรอนในชั้นเอาต์พุต คำนวณได้จาก

$$lw_{s^3,s^2}^{3,2} = lw_{s^3,s^2}^{3,2} + \Delta lw_{s^3,s^2}^{3,2} \quad (4.16)$$

ส่วนการปรับน้ำหนักของชั้นช่อนที่ 2 การทำงานจะเป็นเช่นเดียวกับสมการที่ 4.16 แต่ตัวแปรต่าง ๆ จะเป็นของชั้นช่อนอยู่ สำหรับแต่ละนิวรอนในชั้นช่อนที่ 2 ซึ่งสามารถหาได้จาก

$$lw_{s^2,s^1}^{2,1} = lw_{s^2,s^1}^{2,1} + \Delta lw_{s^2,s^1}^{2,1} \quad (4.17)$$

สำหรับแต่ละนิวรอนในชั้นช่อนที่ 1

$$lw_{s^1,R}^{1,1} = lw_{s^1,R}^{1,1} + \Delta lw_{s^1,R}^{1,1} \quad (4.18)$$

การปรับค่าถ่วงน้ำหนักจะปรับทีละชั้นเอาต์พุตกลับไปยังชั้นอินพุต เมื่อเสร็จสิ้นก็จะกลับสู่ส่วนของป้อนไปข้างหน้าอีกครั้ง เพื่อหาค่าเอาต์พุตของโครงข่ายใหม่มาเปรียบเทียบกับค่าเอาต์พุต

เป้าหมาย และดำเนินการปรับค่าถ่วงน้ำหนักในส่วนของแพร่ค่าย้อนกลับเช่นนี้สลับกันไป โดยวัฏจักรของการคำนวณหาค่าความผิดพลาดและการปรับน้ำหนักจะคงดำเนินต่อไปจนกระทั่งค่าความผิดพลาดต่ำกว่าที่กำหนดไว้ค่าหนึ่งแล้วจึงหยุดกระบวนการ ค่าถ่วงน้ำหนักที่ได้จะเป็นค่าที่เหมาะสมสำหรับทุกคู่ของอินพุตและค่าเป้าหมาย ในการฝึกสอนโครงข่ายประสาทเทียมชนิดแพร่ค่าย้อนกลับ การคำนวณหาความผิดพลาดรวมของตัวอย่าง หาได้จากค่าเฉลี่ยของค่าผิดพลาดยกกำลังสอง (Mean-square error, MSE) ระหว่างคำตอบที่แท้จริง (t_i^3) กับผลลัพธ์ของโครงข่ายประสาทเทียม (a_i^3) ในสมการ (4.19)

$$MSE = \frac{1}{N} \sum_{i=1}^N (t_i^3 - a_i^3)^2 \quad (4.19)$$

โดยที่ N = จำนวนชุดของข้อมูลฝึกสอน

ในขั้นตอนการปรับน้ำหนักที่กล่าวมาข้างต้น เป็นหลักการพื้นฐานแต่ลำดับขั้นตอนการปรับค่าถ่วงน้ำหนักและค่าไบแอสของการฝึกสอนโครงข่ายประสาทเทียมชนิด Feed-forward Neural Networks โดยใช้ Neural Network Toolbox นั้นมีอยู่หลายวิธี อาทิเช่น Levenberg-Maquardt algorithm, Resilient Backpropagation, Conjugate Gradient เป็นต้น แต่ละวิธีจะมีประสิทธิภาพและความรวดเร็วในการฝึกสอนแตกต่างกัน ตามที่ระบุไว้ใน Neural Network Toolbox User's Guide นั้น Levenberg-Maquardt algorithm มีประสิทธิภาพและความรวดเร็วในการฝึกสอนสูงที่สุดในการศึกษานี้จึงได้นำเอาลำดับขั้นตอนการปรับค่าถ่วงน้ำหนักและค่าไบแอสวิธีนี้มาใช้ โดยมีลำดับขั้นตอนการปรับค่าถ่วงน้ำหนักและค่าไบแอส ดังสมการที่ 4.20

$$X_{(new)} = X_{(old)} + \Delta X = X_{(old)} + [J^T \cdot J + \alpha I]^{-1} \cdot J^T \cdot e \quad (4.20)$$

โดยที่ X คือค่าถ่วงน้ำหนัก (w) และค่าไบแอส (b)
 J คือ จาโคเบียนเมตริกซ์ (Jacobian matrix) ของอนุพันธ์ของค่าความผิดพลาด (แต่ละค่า) ต่อค่า X (แต่ละค่า)
 J^T คือ Transpose matrix ของ J
 e คือ เมตริกซ์ของค่าผิดพลาด (Error) ระหว่างคำตอบ (Target, t^3) กับค่าที่โครงข่ายประสาทเทียมคำนวณได้ (Answer, a^3)
 I คือ Identity matrix

$$\text{จาโคเบียนเมตริกซ์ของ (W)} \quad J(W) = \begin{bmatrix} \frac{\partial e_1}{\partial w_{1,1}} & \frac{\partial e_1}{\partial w_{1,2}} & \dots & \frac{\partial e_1}{\partial w_{1,S^2}} \\ \frac{\partial e_2}{\partial w_{2,1}} & \frac{\partial e_2}{\partial w_{2,2}} & \dots & \frac{\partial e_2}{\partial w_{2,S^2}} \\ \vdots & \vdots & \dots & \vdots \\ \frac{\partial e_{S^3}}{\partial w_{S^3,1}} & \frac{\partial e_{S^3}}{\partial w_{S^3,2}} & \dots & \frac{\partial e_{S^3}}{\partial w_{S^3,S^2}} \end{bmatrix} \quad (4.21)$$

$$\text{จาโคเบียนเมตริกซ์ของค่า (b)} \quad J(b) = \begin{bmatrix} \frac{\partial e_1}{\partial b_1} \\ \frac{\partial e_2}{\partial b_2} \\ \vdots \\ \frac{\partial e_{S^3}}{\partial b_{S^3}} \end{bmatrix} \quad (4.22)$$

$$\text{เมตริกซ์ของค่าผิดพลาด} \quad \begin{bmatrix} e_1 \\ e_2 \\ \vdots \\ e_{S^3} \end{bmatrix} = \begin{bmatrix} t_1^3 - a_1^3 \\ t_2^3 - a_2^3 \\ \vdots \\ t_{S^3}^3 - a_{S^3}^3 \end{bmatrix} \quad (4.23)$$

โครงข่ายประสาทเทียมที่ฝึกสอนด้วยกฎการเรียนรู้แบบแพร่ค่าย้อนกลับนั้น จะมีจำนวนนิวรอนในชั้นข้อมูลอินพุต และชั้นข้อมูลเอาต์พุตคงที่ ซึ่งเท่ากับจำนวนของอินพุตและเอาต์พุตของปัญหานั้นเอง ส่วนจำนวนนิวรอนในชั้นซ่อนนั้นสามารถปรับเปลี่ยนได้ตามความเหมาะสม และไม่มีจำนวนที่เป็นสูตรสำเร็จ ดังนั้นในการฝึกสอนโครงข่ายประสาทเทียมจึงต้องมีการปรับเปลี่ยนจำนวนนิวรอนในชั้นซ่อน เพื่อหาโครงสร้างของโครงข่ายประสาทเทียมที่ให้ค่าผิดพลาดระหว่างคำตอบที่แท้จริงกับผลลัพธ์ของโครงข่ายประสาทเทียมต่ำที่สุด (มีประสิทธิภาพสูงที่สุด) จำนวนนิวรอนในชั้นซ่อนนอกจากจะมีผลต่อประสิทธิภาพของโครงข่ายประสาทเทียมแล้วยังมีผลต่อระยะเวลาที่ใช้ในการฝึกสอนอีกด้วย โครงข่ายที่มีจำนวนนิวรอนในชั้นซ่อนน้อย จะใช้เวลาในการฝึกสอนน้อยกว่าโครงข่ายที่มีจำนวนนิวรอนในชั้นซ่อนมาก เมื่อฝึกสอนด้วยเงื่อนไขเดียวกัน แต่อย่างไรก็ตาม ไม่สามารถสรุปได้ว่าโครงข่ายประสาทเทียมที่มีจำนวนนิวรอนในชั้นซ่อนมากจะมีประสิทธิภาพสูงกว่าในโครงข่ายประสาทเทียมที่มีจำนวนนิวรอนในชั้นซ่อนน้อย

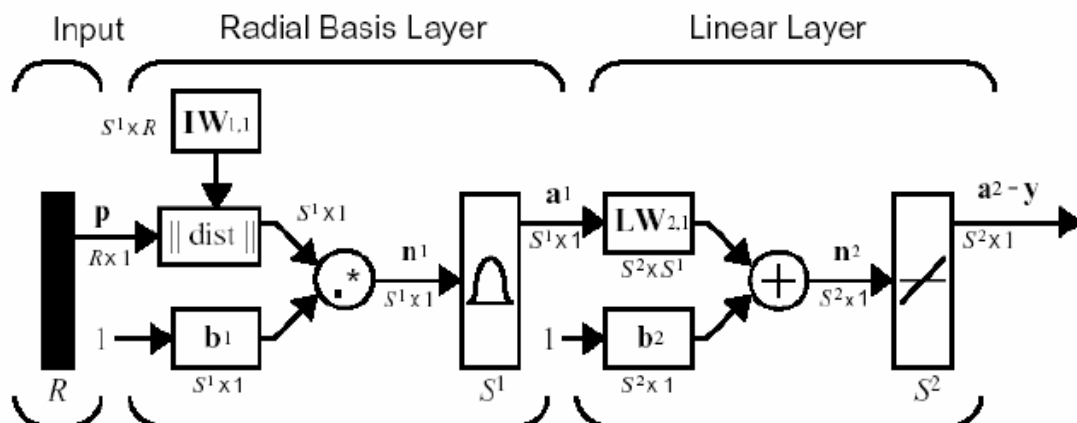
4.7 โครงข่ายประสาทเทียมชนิดฟังก์ชันฐานรัศมี (Radial Basis Function NN)[15]

โครงข่ายประสาทเทียมชนิดแพร่ค่าย้อนกลับและชนิดฟังก์ชันฐานรัศมีเป็นสองชนิดของโครงข่ายประสาทเทียมป้อนไปข้างหน้าที่นิยมใช้กัน โดยทั้งสองชนิดนี้มีลักษณะโครงสร้างพื้นฐานและการประมวลผลที่แตกต่างกัน กล่าวคือ โครงข่ายประสาทเทียมชนิดแพร่ค่าย้อนกลับมีจำนวนชั้นซ่อนมากกว่าหรือเท่ากับหนึ่งชั้น และฟังก์ชันกระตุ้นในชั้นซ่อนสามารถใช้ได้ทั้ง Log-sigmoid หรือ Tangent-sigmoid นอกจากนี้ในชั้นซ่อนของโครงข่ายประสาทเทียมชนิดแพร่ค่าย้อนกลับจะทำการประมวลผลระหว่างค่าอินพุตกับค่าถ่วงน้ำหนักโดยใช้ผลคูณภายใน (Inner products) ในขณะที่โครงข่ายประสาทเทียมชนิดฟังก์ชันฐานรัศมีมีจำนวนชั้นซ่อนเพียงแค่ชั้นเดียว และใช้ฟังก์ชันกระตุ้นชนิด Radial basis เท่านั้น ส่วนการประมวลผลระหว่างอินพุตกับค่าถ่วงน้ำหนักจะใช้การคำนวณระยะห่างยูคลิดีียน (Euclidean distance)

4.7.1 โครงสร้าง

โครงสร้างพื้นฐานของโครงข่ายประสาทเทียมชนิดฟังก์ชันฐานรัศมี (Radial basis function neural network : RBF) ประกอบด้วยนิวรอนเซลล์ที่เรียงกันอยู่มากกว่า 2 ชั้น กล่าวคือ มีได้ตั้งแต่ 3 ชั้นขึ้นไป ซึ่งประกอบด้วย ชั้นอินพุต (Input layer), ชั้นซ่อนหรือบางครั้งเรียกว่า ชั้นฐานรัศมี (Radial basis layer) และ ชั้นเอาต์พุตหรือบางครั้งเรียกว่าชั้นเชิงเส้น (Linear layer) ลักษณะของโครงข่ายชนิดนี้เป็นโครงข่ายที่ป้อนไปข้างหน้า โดยมีการเชื่อมโยงในแต่ละชั้นแบบต่อเนื่องกันหมดนั่นคือ ทุก ๆ นิวรอนในชั้นอินพุตจะส่งสัญญาณไปยังทุก ๆ นิวรอนในชั้นฐานรัศมี และทุก ๆ นิวรอนในชั้นฐานรัศมีจะส่งสัญญาณไปยังทุก ๆ นิวรอนในชั้นเอาต์พุต จากรูปที่ 4. แสดงโครงสร้างพื้นฐานของโครงข่ายประสาทเทียมชนิดฟังก์ชันฐานรัศมี

รูปที่ 4.10 แสดงโครงข่ายประสาทเทียมชนิดฟังก์ชันฐานรัศมี มีอินพุต R ชุด, นิวรอนในชั้นฐานรัศมี เท่ากัน S^1 และมีเอาต์พุต S^2 ชุด สมาชิกของอินพุตเวกเตอร์ P แต่ละตัวจะถูกเชื่อมเข้ากับนิวรอนแต่ละนิวรอนในชั้นฐานรัศมีด้วยค่าถ่วงน้ำหนัก W^1 (เรียกว่า Input weight, $IW^{1,1}$) ซึ่งเป็นเมตริกซ์ขนาด $S^1 \times R$ และเวกเตอร์ a^1 ซึ่งเป็นผลลัพธ์ของชั้นฐานรัศมี โดยมีขนาด S^1 จะเป็นอินพุตของชั้นเชิงเส้นหรือชั้นเอาต์พุต สมาชิกแต่ละตัวของเวกเตอร์ a^1 ก็จะถูกเชื่อมเข้ากับนิวรอนแต่ละนิวรอนในชั้นเชิงเส้น ด้วยค่าถ่วงน้ำหนัก W^2 (เรียกว่า Layer weight, $IW^{2,1}$) ซึ่งเป็นเมตริกซ์ขนาด $S^2 \times S^1$ เช่นเดียวกันกับชั้นฐานรัศมี โดยที่ เวกเตอร์ a^2 ซึ่งเป็นเมตริกซ์ขนาด S^2 จะเป็นผลลัพธ์ของโครงข่ายประสาทเทียม



รูปที่ 4.10 แสดงแบบจำลองของโครงข่ายประสาทเทียมชนิดฟังก์ชันฐานรัศมี [16]

4.7.2 แนวคิด [14]

ขั้นตอนการฝึกสอนของโครงข่ายประสาทเทียมชนิดฟังก์ชันฐานรัศมี จะสามารถแบ่งออกเป็น 2 ขั้นตอนด้วยกัน คือ ป้อนข้อมูลอินพุตไปข้างหน้าและกระบวนการปรับค่าถ่วงน้ำหนักด้วย OLS

ระหว่างป้อนไปข้างหน้า จะเริ่มต้นด้วยการนำค่าอินพุตที่ได้รับจากชั้นอินพุตมาคำนวณหาระยะห่างกับค่าถ่วงน้ำหนักในแต่ละนิวรอน แล้วจึงส่งผลลัพธ์ที่ได้ส่งต่อไปคำนวณฟังก์ชันกระตุ้นชนิดฐานรัศมี จากนั้นส่งผลลัพธ์ที่ได้ส่งต่อไปยังชั้นเอาต์พุตซึ่งอยู่ถัดไป ในชั้นเอาต์พุต แต่ละนิวรอนในชั้นเอาต์พุต จะทำการคำนวณหาผลรวมที่ได้รับในแต่ละนิวรอนกับค่าไบอัส แล้วจึงส่งผลลัพธ์ที่ได้ส่งไปคำนวณกับฟังก์ชันกระตุ้นชนิดเชิงเส้น เพื่อหาคำตอบหรือผลตอบสนองของโครงข่ายสำหรับแต่ละรูปแบบข้อมูลที่ป้อนให้ ทิศทางการไหลของข้อมูลเหล่านี้จะเป็นลักษณะเคลื่อนไปข้างหน้า

ขั้นตอนสุดท้ายจะนำค่าผิดพลาดที่ได้มาใช้ในการปรับค่าถ่วงน้ำหนักด้วยกระบวนการเรียนรู้ Orthogonal Least Squares (OLS) โดยในกระบวนการนี้ค่าถ่วงน้ำหนักในชั้นฐานรัศมีจะเปลี่ยนไปตามจำนวนนิวรอนในชั้นฐานรัศมี ($MN = 1, \dots, N$) และจะทำซ้ำกระบวนการนี้ไปจนกระทั่งจำนวนนิวรอนในชั้นฐานรัศมีมีค่าเท่ากับจำนวนชุดฝึกสอน หรือจนกว่าค่าผิดพลาดที่ได้มีค่าน้อยกว่าค่าผิดพลาดผลรวมยกกำลังสอง (Sum-square error, SSE) ที่ได้กำหนดไว้ ในสมการ (4.24) จึงไม่มีการปรับค่าถ่วงน้ำหนักใด ๆ ทั้งสิ้น

$$SSE = \sum_{i=1}^N (t_i^2 - a_i^2)^2 \quad (4.24)$$

ต่อไปเป็นรายละเอียดของการทำงานในแต่ละขั้นตอน

เมื่อข้อมูลอินพุตปรากฏบนชั้นอินพุต ($P_R, R=1,...,R$) สัญญาณจากชั้นอินพุตทั้งหมดจะถูกส่งมาตามสายเชื่อมโยงไปยังนิวรอนเซลล์ในชั้นถัดไป ในช่วงรอบแรกๆ แต่ละนิวรอนในชั้นฐานรัศมี ถูกเชื่อมโยงด้วยค่าถ่วงน้ำหนักที่อยู่ในชั้นฐานรัศมี ($iw_{MN,R}^{1,1} = MN = 1,...,N$) (โดยจะกำหนดให้มีขนาดของค่าถ่วงน้ำหนักในชั้นฐานรัศมีมีขนาดเล็กแล้วจึงค่อย ๆ เพิ่มขนาดของค่าถ่วงน้ำหนักในชั้นฐานรัศมีเพิ่มขึ้นตามจำนวนรอบที่คำนวณ) ขณะนั้น จะทำการคำนวณสัญญาณเอาต์พุตของนิวรอนเซลล์ในชั้นฐานรัศมี (a^1) ได้จากสมการ

$$a_{MN}^1 = \exp\left(-\frac{\|P_R - iw_{MN,R}^{1,1}\|^2}{2\sigma_{MN}^2}\right) \quad (4.25)$$

จากสมการที่ 4.25 สามารถนำมาเขียนสมการใหม่ได้เป็น

$$a^1 = \exp\left(-\sqrt{\sum_{R=1}^R (iw_{MN,R}^{1,1} - P_R)^2 * b_{MN}^1}\right)^2 \quad (4.26)$$

โดยที่ $b_{MN}^1 = \frac{\sqrt{-\ln(0.5)}}{spread} = \frac{0.8326}{spread}$

Spread คือ ค่ากระจายของฟังก์ชันกระตุ้นชนิดฐานรัศมี

เมื่อสัญญาณเอาต์พุต (a^1) ในชั้นฐานรัศมี เอาต์พุตที่ได้ส่งต่อไปยังชั้นเอาต์พุต โดยแต่ละนิวรอนในชั้นเอาต์พุต ถูกเชื่อมโยงด้วยค่าถ่วงน้ำหนักที่อยู่ในชั้นเอาต์พุต ($lw_{S^2,MN}^{2,1} = S^2 = 1,...,S^2$) จะทำการคำนวณสัญญาณอินพุตของนิวรอนเซลล์ในชั้นเอาต์พุตได้จากสมการ

$$net_{S^2} = b_{S^2}^2 + \sum_{S^1=1}^{S^1} lw_{S^1=1,MN}^{2,1} a_{MN}^1 \quad (4.27)$$

ใช้ฟังก์ชันกระตุ้นเพื่อคำนวณสัญญาณเอาต์พุต (a^2) ในชั้นเอาต์พุต

$$a^2 = f(net_{S^2}) \quad (4.28)$$

การปรับค่าถ่วงน้ำหนักจะปรับโดยเพิ่มจำนวนรอบที่ละรอบ และดำเนินการปรับค่าถ่วงน้ำหนักเช่นนี้ ไปเรื่อย ๆ จนกระทั่งค่าความผิดพลาดต่ำกว่าที่กำหนดไว้ค่าหนึ่งแล้วจึงหยุดกระบวนการเรียนรู้ OLS ในการฝึกสอนโครงข่ายประสาทเทียมชนิดฟังก์ชันฐานรัศมี นอกจากจะปรับค่าถ่วงน้ำหนักโดยเพิ่มจำนวนรอบแล้ว ยังสามารถทำการเพิ่มค่ากระจายของฟังก์ชันกระตุ้นชนิดรัศมี (*spread*) ควบคู่กันไป เพื่อค่าถ่วงน้ำหนักและค่าไบอัสที่ได้จะเป็นค่าที่เหมาะสมสำหรับทุกคู่ของอินพุตและค่าเป้าหมาย