

พื้นฐานระบบประสาทเทียม

Principles of Artificial Neural Systems

รองศาสตราจารย์ ดร.ทวิชัย อวยพรภักขกร



พื้นฐานระบบประสาทเทียม

Principles of Artificial Neural Systems

รองศาสตราจารย์ ดร. ทวีชัย อวยพรกชกร

พื้นฐานระบบประสาทเทียม

Principles of Artificial Neural Systems

โดย รองศาสตราจารย์ ดร.ทวีชัย อวยพรกชกร

ราคา 150 บาท

สงวนลิขสิทธิ์ตามพระราชบัญญัติลิขสิทธิ์ พ.ศ.2537 โดย รองศาสตราจารย์ ดร.ทวีชัย อวยพรกชกร

ห้ามคัดลอก ลอกเลียน ดัดแปลง ทำซ้ำ จัดพิมพ์ หรือกระทำการอื่นใด โดยวิธีการใดๆ ในรูปแบบใดๆ ไม่ว่าส่วน
หนึ่งส่วนใดของหนังสือเล่มนี้ เพื่อเผยแพร่ทุกประเภท หรือเพื่อวัตถุประสงค์ใดๆ

พิมพ์ครั้งที่ 1 พ.ศ. 2568

ข้อมูลทางบรรณานุกรมของหอสมุดแห่งชาติ

ทวีชัย อวยพรกชกร.

พื้นฐานระบบประสาทเทียม = Principles of artificial neural systems.-- กรุงเทพฯ : [ม.ป.พ.],
2568.

256 หน้า.

1. ระบบประสาท. 2. นิวรัลเน็ตเวิร์ค (วิทยาการคอมพิวเตอร์). I. ชื่อเรื่อง.

612.8

ISBN 978-616-623-428-2

ISBN (e-book) : 978-616-623-428-2

จัดพิมพ์และจำหน่ายโดย

รองศาสตราจารย์ ดร.ทวีชัย อวยพรกชกร

คำนำ

ในปัจจุบันปัญญาประดิษฐ์ได้กลายมามีบทบาทสำคัญในชีวิตประจำวันมากขึ้น ปัญญาประดิษฐ์ไม่เพียงถูกนำมาใช้เพียงเพื่อสร้างความสะดวกสบายในชีวิต แต่กลายมาเป็นเครื่องมือในการประกอบธุรกิจหรือเป็นเครื่องมือที่ถูกใช้ในชีวิตประจำวันมากขึ้น เครือข่ายประสาทเทียมเป็นปัญญาประดิษฐ์ตัวหนึ่งที่ถูกพูดถึงเป็นอย่างมากในปัจจุบัน เนื่องจากมีความสามารถที่สูงกว่าปัญญาประดิษฐ์หลายๆ ตัวในอดีตเป็นอย่างมาก สามารถถูกนำไปประยุกต์ใช้ในงานประยุกต์ที่หลากหลาย เป็นที่น่ายินดีที่ ในปัจจุบันการสร้างปัญญาประดิษฐ์ประเภทเครือข่ายประสาทเทียมสามารถทำได้ง่ายมาก มีเครื่องมือที่ช่วยพัฒนามากมาย รวมถึงมีตัวอย่างที่มาพร้อมกับฐานข้อมูลให้สามารถทำตามได้อย่างง่ายดาย นักพัฒนายังสามารถสร้างโดยใช้เวลาไม่มากโดยที่ไม่จำเป็นต้องมีความรู้ทางทฤษฎีก็สามารถสร้างได้ อย่างไรก็ตามความง่ายนี้ก็มีข้อเสียเช่นกัน มีนักพัฒนาจำนวนมากสามารถสร้างเครือข่ายประสาทเทียมได้ แต่อาจมีนักพัฒนาจำนวนไม่มากนักที่มีความเข้าใจในหลักการหรือทฤษฎีที่เกี่ยวข้องที่มีอยู่จำนวนมาก รวมถึงสามารถอธิบายการทำงาน ปรับตั้งการทำงานให้เหมาะสมมากกว่าใช้การลองผิดลองถูก หรือสามารถประยุกต์ต่อไปในงานประยุกต์อื่นๆ ที่แตกต่างไปหรือที่ซับซ้อนมากขึ้นได้ หนังสือเล่มนี้จึง “ไม่” เน้นสอนวิธีการสร้างเครือข่ายประสาทเทียม หรืออธิบายคำสั่งในการสร้าง แต่เน้นที่การสร้างความเข้าใจถึงทฤษฎีที่อยู่เบื้องหลัง ด้วยตัวอย่างง่ายๆ ที่เชื่อมโยงกับทฤษฎี เพื่อให้ความเข้าใจการทำงานของเครือข่ายประสาทเทียม เพื่อเติมเต็มความรู้ของนักพัฒนาให้สมบูรณ์มากยิ่งขึ้น ผู้เขียนมีความคาดหวังว่า ผู้อ่านจะสามารถมีความรู้พื้นฐานที่มากพอที่จะเข้าใจหลักการและเหตุผลที่อยู่เบื้องหลังชุดคำสั่งในการเครือข่ายประสาทเทียมที่มีอยู่มากมายในอินเทอร์เน็ต มีความรู้มากพอในการความเข้าใจในสิ่งที่นักพัฒนาท่านอื่นกำลังทำหรือกำลังให้เหตุผล (จากแหล่งเรียนรู้ทั่วโลก) และท้ายที่สุดจะสามารถทำให้เกิดปัญญาประดิษฐ์ที่มีคุณภาพของคนไทยต่อไปในอนาคต

ทวีชัย อวยพรกชกร

สารบัญ

บทที่ 1. ระบบประสาทเทียม.....	8
1.1 ความหมายของระบบประสาทเทียม	8
1.2 ความเข้าใจผิดเกี่ยวกับระบบประสาทเทียม.....	11
1.3 โมเดลของนิวรอลในระบบประสาทเทียม.....	11
1.4 โมเดลของนิวรอลแบบหลายอินพุต (Multiple-input neuron).....	14
1.5 โมเดลของนิวรอลแบบหลายอินพุตหลายเอาต์พุต (Multiple-input multiple-output neuron)	15
1.6 เลเยอร์ของนิวรอล (Layers of neuron network).....	18
1.7 Perceptron	26
1.8 การทดลองสร้างเครือข่ายประสาทเทียม.....	27
1.8.1 การทดลองสร้างเครือข่ายประสาทเทียมแบบนิวรอลเดียวหลายอินพุต.....	27
1.8.2 การทดลองสร้างเครือข่ายประสาทเทียมแบบหลายนิวรอลหลายอินพุต.....	29
1.8.3 การทดลองสร้างเครือข่ายประสาทเทียมแบบหลายเลเยอร์.....	32
1.8.4 การทดลอง sensitivity ของเครือข่ายประสาทเทียม.....	36
คำถามท้ายบท.....	40
บทที่ 2. การสอนระบบประสาทเทียมแบบเลเยอร์เดียว.....	44
2.1 ความหมายของการสอนระบบประสาทเทียม	44

2.2	การสอนด้วยวิธี Unified Learning Rule	46
2.3	การสอนด้วยวิธี Hebb Rule	53
2.4	การสอนด้วยวิธี Psuedo Inverse Rule	57
2.5	การสอนด้วยวิธี Mean Square Error.....	62
2.6	การสอนด้วยวิธี Least Square Error (LMS).....	71
	คำถามท้ายบท.....	83
บทที่ 3.	การสอนระบบประสาทเทียมแบบหลายเลเยอร์.....	84
3.1	การสอนเครือข่ายประสาทเทียมแบบหลายเลเยอร์ด้วยวิธี Backpropagation	84
3.2	การสอนเครือข่ายประสาทเทียมแบบหลายเลเยอร์ด้วยวิธี Levenberg-Marquardt (LM)....	108
3.3	การสอนเครือข่ายประสาทเทียมแบบหลายเลเยอร์ด้วยวิธี Bayesian Regularization.....	136
3.4	ตัวอย่างงานประยุกต์ในการสร้างเครือข่ายประสาทเทียมแบบหลายเลเยอร์.....	154
	คำถามท้ายบท.....	167
บทที่ 4.	Deep learning.....	168
4.1	ความหมายของ Deep learning และการนำไปใช้งาน.....	168
4.2	Convolutional neural network (CNN).....	171
4.3	การสร้าง Deep Learning Network.....	184
	คำถามท้ายบท.....	198
บทที่ 5.	เกร็ดเพิ่มเติมเกี่ยวกับการสร้างเครือข่ายประสาทเทียม	200
5.1	Data augmentation	200
5.2	การสร้างเครือข่ายประสาทเทียมสำหรับแยกกลุ่มประเภทข้อมูล.....	201

5.3	การสอนที่ดีและการใช้ประโยชน์โมเดลที่ผ่านการสอนมาแล้ว	206
5.4	การกำหนดค่า Padding	208
5.5	การกำหนดค่าเริ่มต้นของค่าน้ำหนักและ bias	209
5.6	การใช้ batch ในการสอน	210
5.7	การเลี่ยงการใช้ Full connected layer ของ CNN	213
5.8	อัลกอริทึมอื่นๆ ที่อยู่บนพื้นฐานของ gradient descent ถูกนำมาใช้ในปัจจุบัน	214
5.8.1	การกำหนด Learning Rate Decay	215
5.8.2	การใช้ Momentum-Based Learning	215
5.8.3	การใช้ Nesterov momentum	218
5.8.4	AdaGrad Algorithm	220
5.8.5	RMSProp Algorithm	220
5.8.6	AdaDelta Algorithm	221
5.8.7	Adam Algorithm	221
	คำถามท้ายบท	222
บทที่ 6.	การประยุกต์ใช้งานระบบประสาทเทียมในงานวิศวกรรมชีวการแพทย์	223
	คำถามท้ายบท	228
	เฉลยคำถามท้ายบท	229
	บรรณานุกรม	253
	ดัชนี	254

บทที่ 1. ระบบประสาทเทียม

1.1 ความหมายของระบบประสาทเทียม

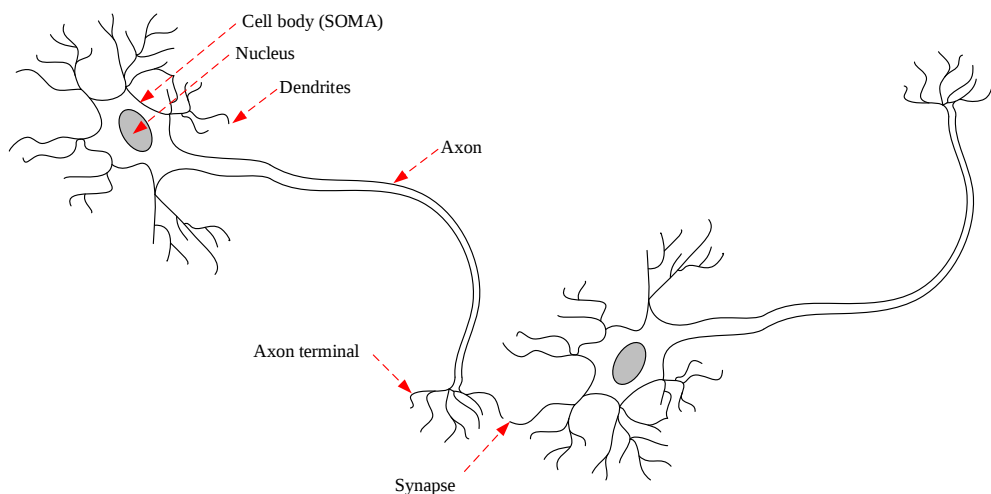
ระบบประสาทเทียม (artificial neural network) เป็นคำเรียกของการนำระบบการคำนวณโดยใช้หน่วยประมวลผล ซึ่งอาจจะเป็นคอมพิวเตอร์หรือไม่ก็ได้ มาใช้ในการประมวลผลในลักษณะเดียวกับสมองของมนุษย์ สมองของมนุษย์นั้นใช้นิวรอล (neural cell) จำนวนมากที่ทำงานร่วมกันจนเกิดเป็น “ความจำ”, “ประสบการณ์”, และ “ความฉลาด” โดยอาศัยการ “เรียนรู้” ทั้งนี้ในปัจจุบัน นักวิทยาศาสตร์ทราบแต่เพียงหน้าที่การทำงานคร่าวๆ ของสมองเท่านั้น เช่น สมองส่วนด้านข้าง (temporal lobe) ใช้สำหรับการประมวลผลด้านการได้ยิน สมองส่วนหลัง (occipital lobe) ใช้สำหรับการประมวลผลด้านการมองเห็น แต่นักวิทยาศาสตร์ไม่สามารถระบุได้ว่านิวรอลเซลล์ส่วนใด ทำหน้าที่ในการจำเสียงพูดของพ่อแม่ ใช้แยกแยะเสียงสัตว์ ใช้จดจำใบหน้าของเพื่อน เนื่องจากนิวรอลเซลล์ทั้งหมดทำงานประสานกัน มีการรายงานไว้ว่ามนุษย์ 1 คนมีจำนวนนิวรอลเซลล์ถึงประมาณ 10^{11} เซลล์ (แสนล้านเซลล์) โดยแต่ละเซลล์จะเชื่อมต่อกับนิวรอลเซลล์อื่นถึง 10,000 นิวรอล

นิวรอลเซลล์แต่ละเซลล์นั้นประกอบด้วยส่วนของ cell body ที่ทำหน้าที่เป็นหน่วยประมวลผลหลัก มี axon ที่มีลักษณะเป็นเป็นทางสำหรับโยงไปเชื่อมกับนิวรอลเซลล์อื่น ที่ปลายของ axon จะมี axon terminal จำนวนมากที่จะโยงไปเชื่อมกับนิวรอลเซลล์อื่นผ่านทาง dendrite ที่อยู่ใกล้กับ cell body (รูปที่ 1-1) เมื่อนิวรอลเซลล์ต้องการส่งสัญญาณไปยังร่างกายส่วนอื่นๆ เช่น สมองส่วน motor (ที่ motor lobe อยู่ในส่วนบนของสมอง) ต้องการสั่งให้มือมีการเคลื่อนไหว นิวเคลียสของ cell body จะผลิต neurotransmitter ซึ่งเป็นสัญญาณทางเคมีจากสมองส่งไปทางทาง axon และเกิดการ synapse ในการส่ง neurotransmission จากนิวรอลเซลล์หนึ่งไปอีกนิวรอลเซลล์หนึ่ง (ในที่นี้จะส่งจากนิวรอลเซลล์หนึ่งไปอีกหลายๆ เซลล์) นิวรอลเซลล์ที่ได้รับ neurotransmission ผ่านทาง dendrite จากประมวลผลและส่ง neurotransmission ต่อไปเรื่อยๆ สัญญาณจากสมองจึงจะแรงขึ้นเรื่อยๆ จนมากพอที่จะสั่งการให้เกิดการเคลื่อนไหวของมือ การ

เคลื่อนไหวของมือสามารถถูกพัฒนาให้สามารถใช้งานได้คล่องแคล่วขึ้นจากการฝึกฝน เนื่องจากทุกครั้งที่มีการสั่งการให้มือเคลื่อนไหว (หรือกล่าวได้ว่าเป็นการฝึกฝน) นิวรอลเซลล์จะได้รับออกซิเจนและสารอาหารเพื่อใช้ในการผลิต neurotransmission ซึ่งทำให้นิวรอลเซลล์นั้นๆ มีความแข็งแรงขึ้นเรื่อยๆ ตามจำนวนครั้งที่ฝึกฝนหรือเรียนรู้ แต่หากนิวรอลเซลล์ใดๆ ไม่ได้ถูกใช้งาน เช่น การท่องจำเรื่องราวหนึ่งๆ แล้วไม่ได้นำเรื่องที่ท่องจำนั้นมาใช้เลย นิวรอลเซลล์ที่เกี่ยวข้องกับการจำเรื่องราวนั้นๆ ก็จะไม่ได้รับออกซิเจนหรือสารอาหารมากนัก จนท้ายสุดแล้วสมองก็จะลืมเรื่องราวนั้นๆ ไปเอง ดังนั้นจะสังเกตได้ว่าสมองมีกระบวนการจำจากการเรียนรู้โดยอาศัยเครือข่ายของนิวรอลเซลล์

ด้วยลักษณะการทำงานเช่นนี้นักวิทยาศาสตร์จึงสร้างโมเดลการคำนวณให้มีลักษณะเดียวกับเครือข่ายของนิวรอล กลายเป็นเครือข่ายประสาทเทียม (artificial neuron network หรือ ANN) ความฉลาดของของ ANN นั้นไม่ได้เกิดจากการโปรแกรมความฉลาดให้ตั้งแต่เริ่มต้น แต่เกิดจากการสอนด้วยข้อมูลที่มีอยู่หรือให้เรียนรู้จากประสบการณ์เอง อย่างไรก็ตาม ANN นั้นมีจำนวนนิวรอลที่น้อยกว่าจำนวนนิวรอลที่มีในสมองมนุษย์มากๆ โดยอาจมีเพียงประมาณ 1,000 – 2,000 เซลล์เท่านั้น (เทียบกับจำนวนนิวรอลของสมองที่ 10^{11} เซลล์) หากพิจารณาจากจำนวนนิวรอลเซลล์ ความฉลาดที่ ANN ทำได้นั้นจึงเทียบไม่ได้เลยกับสมองมนุษย์ แต่ลักษณะงานที่ ANN เอาไปใช้นั้นมักจะกำหนดให้ทำงานเฉพาะอย่าง เช่น การอ่านตัวหนังสือจากรูปภาพ การค้นหาหน้าบุคคลจากภาพ การกรองสัญญาณรบกวนออก งานประยุกต์ของ ANN จึงมีความเฉพาะเจาะจงมากกว่า ซึ่งทำให้จำนวนนิวรอลเซลล์ในระดับพันตัวสามารถทำงานได้ ทั้งนี้ ANN จะถูกประมวลผลบนหน่วยประมวลผล ซึ่งอาจเป็นเครื่องคอมพิวเตอร์ โทรศัพท์มือถือ ซึ่งในปัจจุบันเทคโนโลยีการประมวลผลเหล่านี้สามารถทำได้ด้วยความเร็วที่สูงมาก ด้วยความที่หน่วยประมวลผลทำงานได้เร็วและจำนวนนิวรอลไม่มาก ทำให้ ANN สามารถคิดและประมวลผลให้เสร็จได้เร็วในระดับนาโนหรือไมโครวินาที ผิดกับสมองมนุษย์ที่มีการส่งสัญญาณด้วยข้อมูลทางเคมีและมีจำนวนนิวรอลเซลล์มาก สมองมนุษย์สามารถตอบสนองต่อสิ่งกระตุ้นด้วยความเร็วประมาณ 100-500 มิลลิวินาที ซึ่งถือว่าช้ากว่าคอมพิวเตอร์เป็นอย่างมาก การนำ ANN ไปใช้ในทางปฏิบัติจึงมักนำไปใช้กับงานประยุกต์ที่มีลักษณะการทำงานเฉพาะจง ทำงานซ้ำๆ และต้องการความเร็วสูง เช่น การนำ ANN ในการหาตำแหน่งของวัตถุและนำไปควบคุมการหยิบวัตถุให้แม่นยำ เป็นต้น

ระบบประสาทเทียมจึงเป็นเทคนิคในการพัฒนากระบวนการเรียนรู้ของเครื่องจักร (machine learning) วิธีการหนึ่งในการนำไปพัฒนาปัญญาประดิษฐ์ (artificial intelligence) ซึ่งแท้จริงแล้วเทคนิคการพัฒนากระบวนการเรียนรู้ให้กับเครื่องจักรยังมีอีกหลายวิธี แต่ด้วยความแม่นยำของเทคนิค ANN จึงทำให้เป็นที่กล่าวถึงมากที่สุดในปัจจุบัน สิ่งที่เราควรทราบหนึ่งเกี่ยวกับระบบประสาทเทียมคือชื่อเรียกที่เปลี่ยนไปตามยุคสมัย แม้ว่าคำว่า “ระบบประสาทเทียม” จะถูกกล่าวถึงและถูกเรียกเป็นเวลานาน แต่ในปัจจุบันนักวิทยาศาสตร์เชื่อว่าระบบประสาทเทียมยังห่างไกลกับระบบประสาทของมนุษย์ รวมถึงมีกระบวนการทำงานที่ซับซ้อนของนิวรอลเซลล์หลายอย่าง ที่ระบบประสาทเทียมไม่ได้นำมาพิจารณาหรือยังไม่สามารถทำได้ ประกอบกับในปัจจุบันมีเทคนิคของระบบประสาทเทียมใหม่ๆ ที่มีโครงสร้างของเครือข่ายที่แตกต่างไปจากการทำงานของระบบประสาทของมนุษย์ด้วย ในปัจจุบันจึงมักไม่ถูกนิยามเรียกว่าระบบประสาทเทียมอีกต่อไป แต่จะเรียกว่า “ระบบการเรียนรู้เชิงลึก (deep learning)” แทน สำหรับโครงสร้างของ ANN ที่นิยมใช้ในอดีต จะกลายเป็นเพียงส่วนประกอบหนึ่งของโครงสร้างของเครือข่าย deep learning สมัยใหม่ ซึ่งในปัจจุบันจะเรียกว่า “full-connected network” แทน (ซึ่งจะกล่าวถึงต่อไปในภายหลัง) อย่างไรก็ดีเนื้อหาในเอกสารนี้ทั้งหมดจะยังคงใช้คำว่าระบบประสาทเทียม (หรือ ANN) เพื่อความง่ายในการอธิบาย



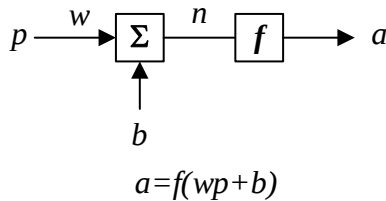
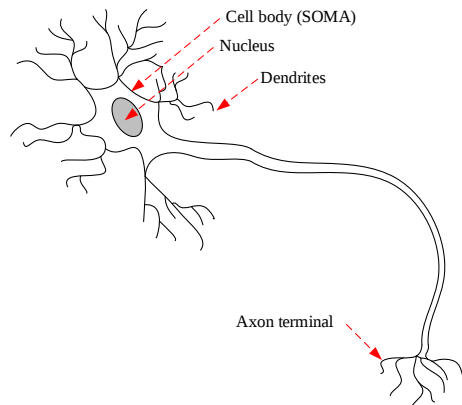
รูปที่ 1-1 ลักษณะของนิวรอลเซลล์และการเชื่อมต่อ

1.2 ความเข้าใจผิดเกี่ยวกับระบบประสาทเทียม

ระบบประสาทเทียม มักจะถูกเข้าใจว่าเป็นระบบที่สามารถเรียนรู้ได้ทุกอย่าง สามารถที่จะทำงานแทนมนุษย์ได้ในลักษณะของปัญญาประดิษฐ์ที่คิด เรียนรู้ และตัดสินใจได้เอง ความเข้าใจนี้อาจเป็นจริงในอนาคต แต่ถือว่าไม่ถูกต้องในปัจจุบัน เนื่องจากโมเดลการคิดตัดสินใจ (หรืออาจเรียกว่าสมองกล) ของปัญญาประดิษฐ์ที่ใช้ระบบประสาทเทียม ยังมีความซับซ้อนน้อยกว่าระบบสมองของมนุษย์อย่างเปรียบเทียบไม่ได้ (อาจพิจารณาได้จากจำนวนนิวรอนเซลล์ก็ได้) ดังนั้นความฉลาดของปัญญาประดิษฐ์จึงอยู่ในขอบเขตที่จำกัด ที่สำคัญคือ ความฉลาดของหรือความแม่นยำของระบบประสาทเทียมนั้นขึ้นขึ้นกับการสอนเป็นสำคัญ ซึ่งปกติแล้วมนุษย์จะต้องเป็นผู้เตรียมข้อมูลให้ ซึ่งข้อมูลที่นำมาสอนอาจมีอยู่อย่างจำกัด หรือข้อมูลที่นำมาสอนอาจนำมาจากสถานะแวดล้อมหนึ่งที่เหมาะสมกับบริบทหนึ่งแต่อาจไม่เหมาะสมกับอีกสถานะแวดล้อมอื่นก็ได้ ตัวอย่างเช่น การทำลายพฤติกรรมกรรมการซื้อของของคนไทย อาจไม่เหมือนหรือแตกต่างโดยสิ้นเชิงกับคนฝรั่งเศส เป็นต้น ดังนั้นความฉลาดของระบบประสาทเทียมจึงมีอยู่จำกัด อย่างไรก็ดี แม้ว่าความฉลาดของระบบประสาทเทียมจะสู้กับความฉลาดของมนุษย์ไม่ได้ (กล่าวอีกนัยหนึ่งระบบประสาทเทียมต้องการเลียนแบบสมองมนุษย์ที่มีความฉลาดมากกว่า) แต่ก็สามารถฉลาดกว่ามนุษย์ได้เช่นกันถ้าข้อมูลที่นำมาสอนมีความหลากหลายมากกว่า เช่น หากสอนระบบประสาทเทียมด้วยข้อมูลคนเป็นโรคเบาหวานด้วยการนำข้อมูลสถิติมาจากทั่วโลก ความถูกต้องในการตรวจคัดคนเป็นเบาหวานด้วยระบบประสาทเทียมก็จะมีแม่นยำมากกว่า หรือในกรณีของคอมพิวเตอร์ที่เล่นหมากรุกชนะมนุษย์ ก็เกิดจากการป้อนข้อมูลการเล่นของนักเล่นทั่วโลกและให้คอมพิวเตอร์ลองเล่นกับคอมพิวเตอร์ด้วยตัวเองด้วย

1.3 โมเดลของนิวรอนในระบบประสาทเทียม

นักวิทยาศาสตร์ได้พัฒนาโมเดลทางคณิตศาสตร์ที่มาเทียบเคียงกับนิวรอน 1 เซลล์ได้ดังรูปที่ 1-2



รูปที่ 1-2 โมเดลทางคณิตศาสตร์ของนิวรอน 1 เซลล์

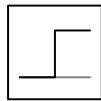
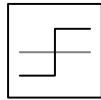
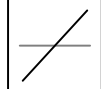
Dendrite ของนิวรอนจะแทนด้วยอินพุต (input) ของโมเดลนิวรอน (p) ส่วน axon terminal ของนิวรอนจะแทนด้วยเอาต์พุต (output) ของโมเดลนิวรอน (a) ในส่วนของ cell body ที่เป็นตัวประมวลผลหลัก ในโมเดลของนิวรอนจะใช้สมการง่ายคือให้นำผลรวมของการคูณอินพุตแต่ละตัวด้วยค่าน้ำหนัก (weight) (w) บวกกับค่าถ่วงน้ำหนัก (bias) (b) โดยผลรวมนี้จะเรียกว่า summer output (n) ซึ่งจะถูกนำไปคำนวณต่อใน activation function (f) เพื่อให้ได้ output ต่อไปด้วยสมการ (1-1) ดังนั้นการเขียนแทนนิวรอน 1 เซลล์จึงอาจเขียนแบบรูปที่ 1-2 หรือสมการ (1-1) ก็ได้

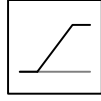
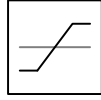
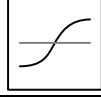
$$a = f(wb + b) \quad (1-1)$$

ทั้งนี้ในส่วน transfer function หรือ activation function (ปัจจุบันนิยมเรียกในชื่อนี้มากกว่า) นั้นสามารถเลือกใช้ได้ ดังแสดงตัวอย่างในตาราง 1-1 จะเห็นว่ามีเลือกใช้ได้หลายอย่าง ซึ่งสมการเหล่านี้เป็นสมการพื้นฐานที่พบได้ในโปรแกรมที่ช่วยคำนวณทางคณิตศาสตร์ทั่วไป อย่างเช่น matlab เป็นต้น ลักษณะของความสัมพันธ์ของสมการนั้นได้แสดงในตารางแล้ว ซึ่งแต่ละฟังก์ชันมีความเหมาะสมในการนำไปใช้งานที่แตกต่างและมีรายละเอียดที่แตกต่างกัน ยกตัวอย่างเช่น ในกรณีที่ output เป็นการทำนายโรคว่าจะเป็นหรือไม่เป็น (มีได้เพียง 2 สถานะ ในที่นี้กำหนดให้เป็น 0 และ 1) จะสามารถใช้ฟังก์ชัน hard limit ก็ได้ ซึ่งกรณีที่ summer output (n) มีค่าน้อยกว่าศูนย์ (ไม่จำกัดว่าน้อยกว่าเป็นค่าเท่าไร) ผลลัพธ์จะเป็น 0 แต่หากมากกว่าหรือเท่ากับ 0 ผลลัพธ์จะมีค่าเป็น 1 ซึ่งคุณสมบัติเช่นนี้อาจไม่เหมาะสมในบางลักษณะงาน เช่น หาก n มีค่าเป็น -0.000001 กับ 0 ผลลัพธ์ (a) จะแตกต่างกันโดยสิ้นเชิง คือ 0 กับ 1 ซึ่งการใช้ฟังก์ชัน log-sigmoid อาจจะได้ผลลัพธ์ที่ดีกว่า เนื่องจาก a จะมีค่าที่ไม่แตกต่างกันมากเกินไป

อนึ่ง เป็นที่สังเกตว่าในโมเดลนิวรอล 1 โมเดล ใน 1 นิวรอลจะมีเอาต์พุตได้เพียง 1 เอาต์พุตเท่านั้น ซึ่งคุณลักษณะนี้เป็นตัวอย่างหนึ่งที่ทำให้โมเดลนิวรอลแตกต่างจากนิวรอลเซลล์จริงๆ

ตาราง 1-1 Activation function ที่ใช้ในการสร้างโมเดลของนิวรอล

Name	Formula	Symbol	MATLAB function
Hard Limit	$a = 0 \quad n < 0$ $a = 1 \quad n \geq 0$		hardlim
Symmetrical Hard Limit	$a = -1 \quad n < 0$ $a = +1 \quad n \geq 0$		hardlims
Linear	$a = n$		purelin

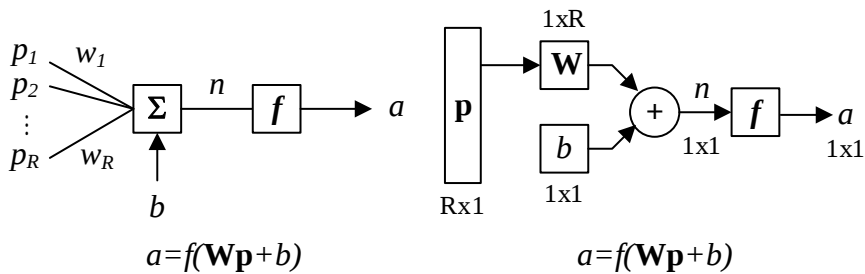
Name	Formula	Symbol	MATLAB function
Saturating Linear	$a=0 \quad n<0$ $a=n \quad 0 \leq n \leq 1$ $a=1 \quad n>1$		satlin
Symmetric Saturating Linear	$a=-1 \quad n<-1$ $a=n \quad -1 \leq n \leq 1$ $a=1 \quad n>1$		satlins
Log-Sigmoid	$a = \frac{1}{1+e^{-n}}$		logsig
Hyperbolic Tangent Sigmoid	$a = \frac{e^n - e^{-n}}{e^n + e^{-n}}$		tansig
Positive Linear (Rectified Linear Unit - RELU)	$a=0 \quad n<0$ $a=n \quad n \geq 0$		poslin

1.4 โมเดลของนิวรอลแบบหลายอินพุต (Multiple-input neuron)

ในกรณีที่มีหลายอินพุตนั้น การเขียนแทนด้วยภาพจะมีความยุ่งยากมากขึ้นดังแสดงในรูปที่ 1-3 (ซ้าย) เพราะเขียนแสดงด้วยเส้นตามจำนวนของอินพุต การเขียนที่ดูง่ายขึ้นและมีความสอดคล้องกับการเขียนในลักษณะของสมการคณิตศาสตร์จะแสดงในรูปที่ 1-3 (ขวา) ซึ่งจะนำเมตริกซ์มาช่วยในการแสดงจำนวนอินพุต ในที่นี้จะเห็นว่าจำนวนอินพุตจะเท่ากับ R หรืออาจเขียนด้วยเมตริกซ์ $\mathbb{R}^{R \times 1}$ หรือ $\mathbb{R}^R$ ซึ่งเป็นเวกเตอร์ของอินพุต ในส่วนของ W นั้นจะเขียนแทนด้วยเมตริกซ์ขนาด $\mathbb{R}^{1 \times R}$ และ b จะเป็น scalar (หรือเป็นเมตริกซ์ขนาด $\mathbb{R}^{1 \times 1}$) ผลลัพธ์จากการคูณจะกลายเป็น scalar เสมอ ทั้ง n และ a ด้วยลักษณะนี้การ

เขียนสมการจะเปลี่ยนไปเล็กน้อยดังแสดงในสมการ (1-2) ตัว W จะกลายเป็นพื้มพีใหญ่และหนา แสดงถึงความเป็นเมตริกซ์ส่วน p จะเป็นตัวพื้มพีเล็กและหนา แสดงถึงความเป็นเวกเตอร์ ส่วน a และ b จะเป็นตัวเอียงพื้มพีเล็กแสดงถึงความเป็น scalar

$$a=f(W\mathbf{p}+b) \quad (1-2)$$



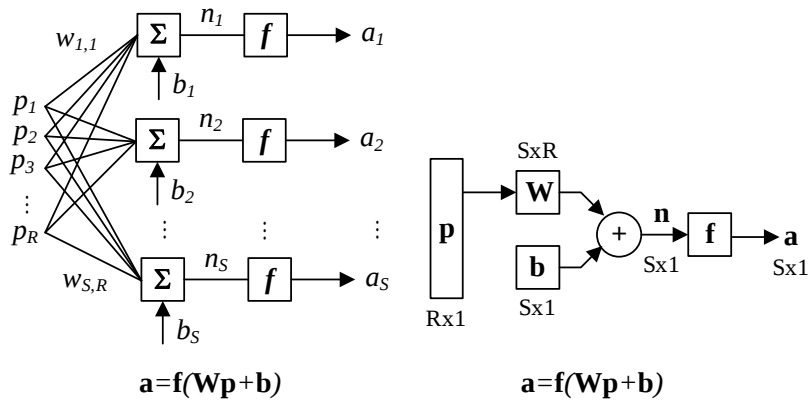
รูปที่ 1-3 ตัวอย่างการเขียนนิรูลแบบหลายอินพุต

1.5 โมเดลของนิรูลแบบหลายอินพุตหลายเอาต์พุต (Multiple-input multiple-output neuron)

เนื่องจากโมเดลของนิรูลนั้น 1 นิรูลมีได้เพียง 1 เอาต์พุต ในกรณีที่มีหลายเอาต์พุต จะต้องมียุหลายนิรูลในโมเดลเช่นกัน (รูปที่ 1-4) ซึ่งในตัวอย่างในรูปมี S เอาต์พุต จะต้องมียุ S นิรูลในโมเดล ซึ่งจะสังเกตเห็นว่าทุกอินพุตจะต้องเป็นอินพุตของทุกนิรูลด้วย ซึ่งทำให้มีจำนวน w เพิ่มขึ้น ด้วยลักษณะโครงสร้างเช่นนี้หากเขียนแทนด้วยลักษณะของเมตริกซ์แล้วจะแสดงในรูปที่ 1-4 (ขวา) โดย a จะเป็นเมตริกซ์ขนาด $\mathbb{R}^{S \times 1}$ หรือ $\mathbb{R}^S$ ส่วน b จะมีเท่ากับจำนวนของนิรูลหรือเป็นเมตริกซ์ที่ขนาดเท่ากับ a ในส่วนของ W นั้นจะกลายเป็นเมตริกซ์ขนาด $\mathbb{R}^{S \times R}$ ซึ่งจะสังเกตได้ว่า จำนวนแถวของ W จะเท่ากับจำนวนนิรูล ส่วนจำนวนหลักของ W จะเท่ากับจำนวนอินพุต รูปแบบของเมตริกซ์ที่เกี่ยวข้องจะแสดงในรูปที่ 1-5 และสมการ (1-3)

จะสังเกตได้ว่า b และ a จะกลายเป็นเวกเตอร์ ส่วน activation function จะเขียนด้วยตัวหนา เนื่องจากเกี่ยวข้องกับหลายสมการ (แต่ละนิวรอนแทน 1 สมการ) รูปแบบการเขียนของสมการ (1-3) เป็นที่นิยมมากที่สุด เนื่องจากสามารถแทนที่อินพุตหรือที่เอาต์พุตก็ได้

$$\mathbf{a} = \mathbf{f}(\mathbf{W}\mathbf{p} + \mathbf{b}) \quad (1-3)$$



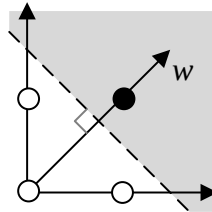
รูปที่ 1-4 ตัวอย่างการเขียนนิวรอนแบบหลายอินพุตและหลายเอาต์พุต

$$\mathbf{p} = \begin{bmatrix} p_1 \\ p_2 \\ \vdots \\ p_R \end{bmatrix} \quad \mathbf{W} = \begin{bmatrix} w_{1,1} & w_{1,2} & \cdots & w_{1,R} \\ w_{2,1} & w_{2,2} & \cdots & w_{2,R} \\ \vdots & \vdots & & \vdots \\ w_{S,1} & w_{S,2} & \cdots & w_{S,R} \end{bmatrix} \quad \mathbf{b} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_S \end{bmatrix} \quad \mathbf{a} = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_S \end{bmatrix}$$

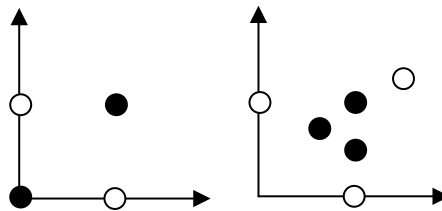
รูปที่ 1-5 ตัวแปรที่เกี่ยวข้องในสมการโมเดลของระบบประสาทเทียม

อย่างไรก็ดี รูปแบบของเครือข่ายของประสาทเทียมที่แสดงนี้นั้นจะเป็นแบบเลเยอร์เดียว ซึ่งหากพิจารณาจากลักษณะรูปแบบการสมการที่นำมาทำเป็นโมเดล (ดังที่แสดงในสมการ (1-3)) จะพบว่าอยู่บน

พื้นฐานของสมการเส้นตรง เครือข่ายประสาทเทียมแบบเลเยอร์เดียว (single-layer neuron network) นั้น จึงสามารถแก้ปัญหาที่อยู่ในลักษณะสมการเส้นตรงได้เท่านั้น ไม่สามารถแก้ปัญหาที่มีความซับซ้อนใน ลักษณะไม่เป็นเส้นตรงได้ หากพิจารณาปัญหาที่ไม่ซับซ้อนดังเช่นใน รูปที่ 1-6 หากสามารถหาเส้นตรงที่แบ่ง ข้อมูลสีขาวและสีดำออกจากกันได้ (linear separable) จะถือว่าเป็นปัญหาที่เครือข่ายประสาทเทียมแบบเลเยอร์เดียวทำได้ แต่หากปัญหาเป็นไปในลักษณะของรูปที่ 1-7 จะไม่สามารถหาเส้นตรงใดๆ ได้เลย ซึ่ง หมายความว่าไม่เหมาะสมกับการเอาเครือข่ายประสาทเทียมแบบเลเยอร์เดียวมาใช้ หรือหากนำมาใช้ จะมีความผิดพลาดเกิดขึ้นได้ ซึ่งจะต้องพิจารณาอีกทีหนึ่งว่าความผิดพลาดข้างต้นมากน้อยเพียงไร และเป็นที่ ยอมรับได้หรือไม่ (ในบางครั้ง ผู้พัฒนาอาจยอมให้มีความผิดพลาดได้บ้างแลกกับความไม่ซับซ้อนในการ แก้ปัญหา)



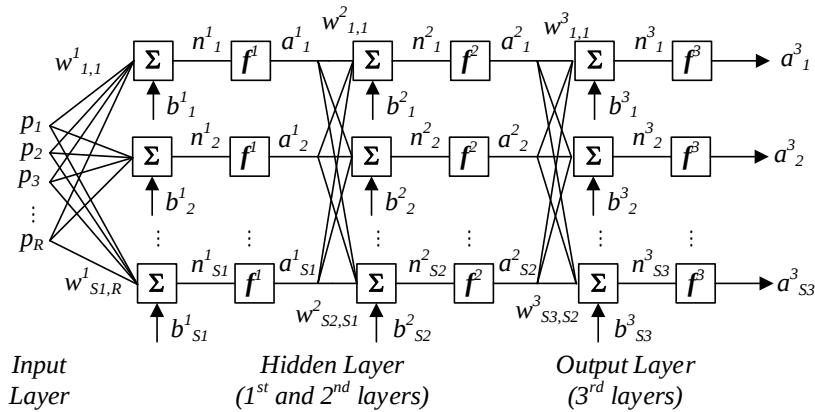
รูปที่ 1-6 การแก้ปัญหาด้วยเครือข่ายประสาทเทียมแบบเลเยอร์เดียวที่เป็นเส้นตรง



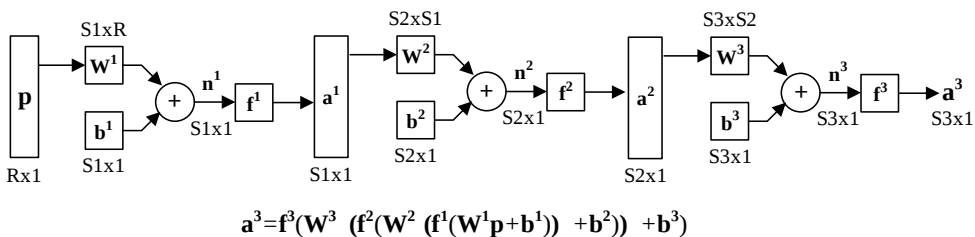
รูปที่ 1-7 ปัญหาของการแก้ปัญหาด้วยเครือข่ายประสาทเทียมแบบเลเยอร์เดียวที่ไม่เป็นเส้นตรง

1.6 เลเยอร์ของนิวรอล (Layers of neuron network)

ปัญหาที่มีความซับซ้อนจำเป็นต้องออกแบบให้เครือข่ายของประสาทเทียม มีความสามารถในการรองรับความซับซ้อนนั้นๆ ได้ การเพิ่มความสามารถในการแก้ปัญหาซับซ้อนทำได้โดยการเพิ่มเลเยอร์ของนิวรอลดังแสดงในรูปที่ 1-8 ซึ่งโดยปกติจะใช้งานที่ 2-3 เลเยอร์เท่านั้น ซึ่งหากพิจารณาจะเห็นว่า ผลลัพธ์จากเลเยอร์แรกจะกลายเป็นอินพุตของเลเยอร์ถัดมา หรืออาจกล่าวได้ว่าผลลัพธ์ของเลเยอร์ที่สองจะกลายเป็นสมการกำลังสอง (สมมติว่า activation function เป็น pure linear a_2 จะมีค่า $W_2 (W_1 p_1 + b_1) + b_2$ ซึ่งเป็นสมการกำลังสอง) เช่นเดียวกับเลเยอร์ที่สามจะเป็นสมการกำลังสาม ด้วย order ของสมการที่สูงขึ้น จึงทำให้สามารถแก้ปัญหาที่ซับซ้อนได้



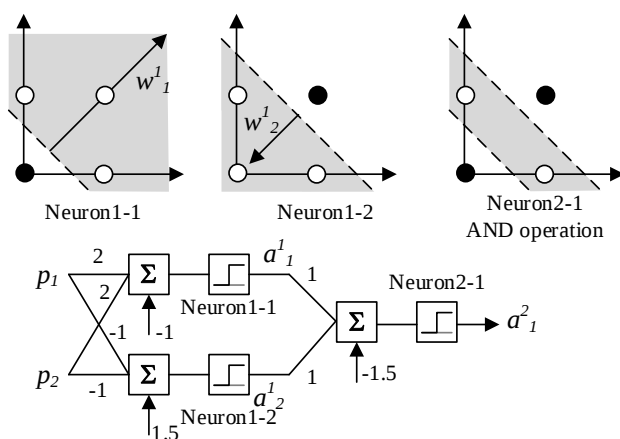
R-S1-S2-S3



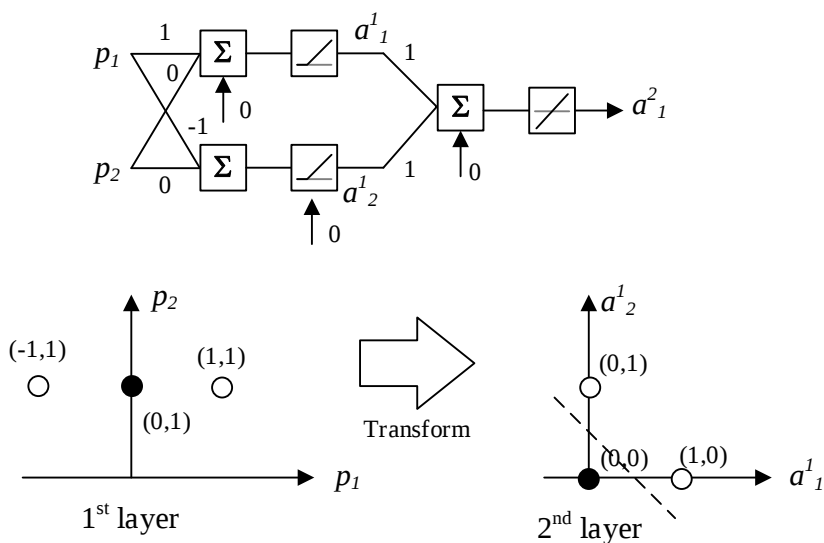
รูปที่ 1-8 ตัวอย่างการเขียนนิวรอลแบบหลายเลเยอร์

ทั้งนี้ เลเยอร์สุดท้ายจะมีจำนวนนิวรอนเท่ากับจำนวนของเอาต์พุต ซึ่งมักถูกเรียกว่าเอาต์พุตเลเยอร์ (output layer) ส่วนเลเยอร์ที่เหลือจะถูกเรียกรวมๆ ว่า hidden layer โดยอาจมีการเรียกเลเยอร์แรกว่าอินพุตเลเยอร์ (input layer) ในบางครั้ง hidden layer นั้นจะมีจำนวนนิวรอนก็ได้ ไม่เกี่ยวกับทั้งจำนวนอินพุตและจำนวนเอาต์พุต

จำนวนเลเยอร์ที่เพิ่มขึ้นมา ทำให้ความซับซ้อนของสมการในการแก้ปัญหามีมากขึ้น หากพิจารณาตัวอย่างอย่างง่ายในรูปที่ 1-9 ซึ่งมี 2 นิวรอนในเลเยอร์แรก นิวรอนตัวแรกในเลเยอร์แรกจะเปลี่ยนเหมือนการแก้ปัญหาด้วยการหาสมการเส้นตรงเส้นหนึ่ง ส่วนนิวรอนตัวที่สองในเลเยอร์แรกก็จะเป็นการแก้ปัญหาด้วยการหาสมการอีกเส้นหนึ่ง ผลของนิวรอนทั้งสองตัวในเลเยอร์แรกจะถูกนำมารวมกัน หรือกล่าวคือเป็นการ “and” กัน ซึ่งจะทำให้สามารถกำหนดรูปแบบการแบ่งข้อมูลจุดสีขาวและสีดำได้ซับซ้อนมากขึ้น (ในที่นี้แทนที่จะใช้เส้นตรงเส้นเดียวในกรณีที่ใช้เลเยอร์เดียว จะกลายเป็นให้หาเส้นตรง 2 เส้นที่สามารถแบ่งข้อมูลได้ในกรณีที่ใช้สองเลเยอร์ตามรูป) ในลักษณะเดียวกับตัวอย่างในรูปที่ 1-10 ที่แม้ข้อมูลดิบจะไม่สามารถหาสมการที่แบ่งข้อมูลสองชุดที่แตกต่างกันได้อย่างถูกต้อง แต่เมื่อข้อมูลถูกส่งผ่านเข้าเลเยอร์แรกจะถูกแปลงให้อยู่ในรูปแบบที่ง่ายในการแบ่ง ทำให้นิวรอนในเลเยอร์ถัดมาสามารถแก้ปัญหาให้ถูกต้องได้โดยง่าย

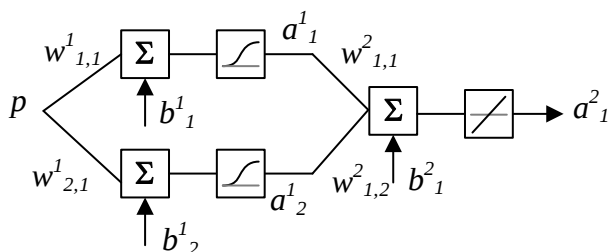


รูปที่ 1-9 ตัวอย่างการแก้ปัญหาด้วยเครือข่ายประสาทเทียมแบบ 2 เลเยอร์ โดยเลเยอร์แรกมีเพียง 2 นิวรอนและเลเยอร์สุดท้ายมีเพียงเลเยอร์เดียว

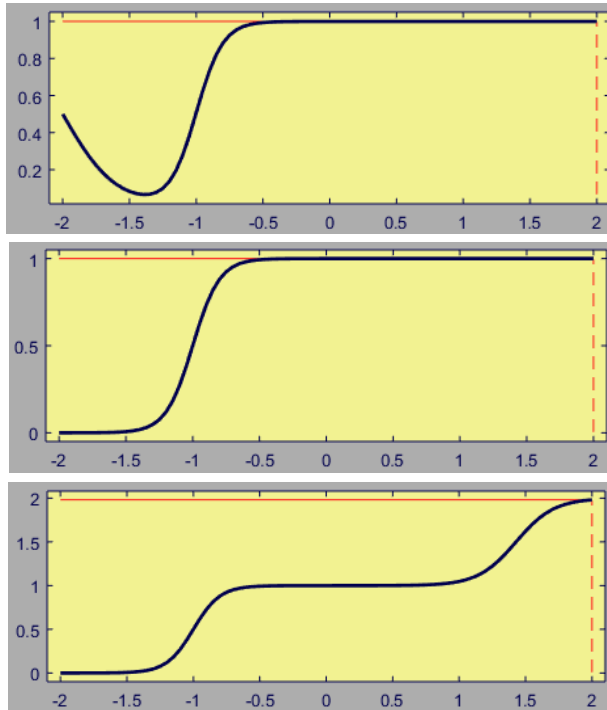


รูปที่ 1-10 ตัวอย่างการแก้ปัญหาด้วยเครือข่ายประสาทเทียมแบบ 2 เลเยอร์ เลเยอร์แรกจะแปลงข้อมูลให้
ง่ายสำหรับเลเยอร์ถัดไปในการแก้ปัญหา

หาก activation function ของนิวรอลมีความซับซ้อนมากขึ้น เช่น เป็น non-linear function (เช่น sigmoid) ดังแสดงในรูปที่ 1-11 การปรับเปลี่ยนค่าน้ำหนัก (w) และค่า bias (b) ของแต่ละนิวรอลก็จะทำให้การตอบสนองของเอาต์พุตที่มีผลมาจากอินพุตเปลี่ยนแปลงไปด้วย ดังตัวอย่างแสดงในรูปที่ 1-12



รูปที่ 1-11 ตัวอย่างนิวรอลแบบหลายเลเยอร์



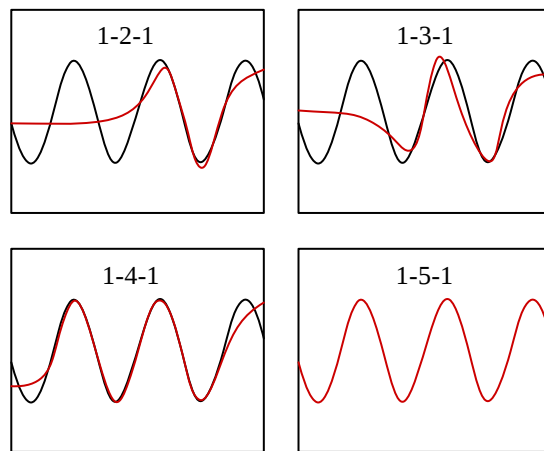
รูปที่ 1-12 ผลจากการปรับพารามิเตอร์ ค่าน้ำหนัก ($w^1_{1,1}$) ของรูปที่ 1-11 ทั้งนี้แกน x จะแทนอินพุต (p) ส่วนแกน y จะแทนเอาต์พุต (a)

ด้วยรูปแบบโครงสร้างของเครือข่ายประสาทเทียมแบบหลายเลเยอร์ที่มีจำนวนเลเยอร์ จำนวน นิวรอล ที่ออกแบบได้หลายหลาย การเขียนแทนจึงอาจเขียนแทนในลักษณะต่อไปนี้

จำนวนอินพุต-จำนวนนิวรอลในเลเยอร์ที่ 1-จำนวนนิวรอลในเลเยอร์ที่ 2-...

ยกตัวอย่างเช่น 1-2-1 จะหมายถึงมีจำนวนอินพุตเท่ากับ 1 อินพุต โดยมี 2 เลเยอร์ โดยเลเยอร์แรก จะมีจำนวนนิวรอลเท่ากับ 2 และเลเยอร์ที่สองจะมีจำนวนนิวรอลเท่ากับ 1 และเนื่องจากจำนวนเอาต์พุตจะ เท่ากับจำนวนนิวรอลของเลเยอร์สุดท้ายอยู่แล้ว จึงทำให้เครือข่ายนี้มีเอาต์พุตเท่ากับ 1 เอาต์พุต

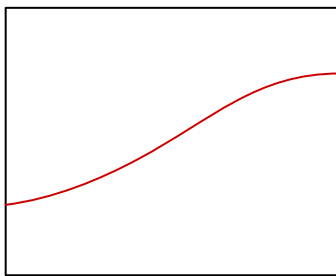
จำนวนนิวรอนที่มากขึ้นจะส่งผลให้ความสามารถในการแก้ปัญหาซับซ้อนได้ดีมากขึ้น ยกตัวอย่างดังรูปที่ 1-13 ในการสร้างเครือข่ายประสาทเทียมที่จะนำมาทดแทนการสร้างสมการ sine ที่มี 3 คาบเวลา (โดยใช้ transfer function แบบ sigmoid) โดยให้อินพุตเป็นข้อมูลในแกน x ส่วนเอาต์พุตเป็นข้อมูลในแกน y หากใช้เครือข่ายแบบ 1-2-1 จะทำให้อาต์พุตมีความสอดคล้องกับรูปคลื่นเอาต์พุต (target) ที่ต้องการเพียงบางส่วน แต่เมื่อเพิ่มจำนวนนิวรอนมากขึ้น จะทำให้เครือข่ายมีความสามารถในการคิดคำนวณให้ทดแทนสมการ sine ได้ใกล้เคียงมากขึ้น ซึ่งจากตัวอย่างจะเห็นว่า เครือข่ายแบบ 1-5-1 (มีนิวรอนในเลเยอร์แรก 5 นิวรอน) สามารถให้ผลลัพธ์ตรงกับเอาต์พุตที่ต้องการ



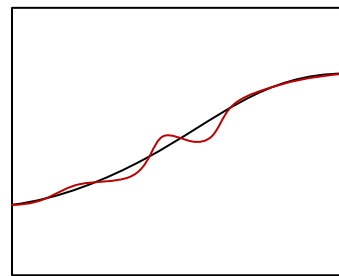
รูปที่ 1-13 ผลจากการกำหนดจำนวนนิวรอนที่แตกต่างกัน

อย่างไรก็ดี การเพิ่มจำนวนนิวรอนอาจไม่ได้ส่งผลในแง่ดีเสมอไป (กล่าวอีกนัยหนึ่งคือ ไม่ได้ทำให้ความถูกต้องสูงขึ้นทุกครั้งไป) หากพิจารณาจากตัวอย่างใน รูปที่ 1-14 ซึ่งเป็นการประมาณคำตอบของส่วนหนึ่งของสัญญาณ sine จะพบว่า จำนวนนิวรอนในเลเยอร์แรกที่มีเพียง 2 ตัวกลับให้ผลที่ดีกว่าจำนวนนิวรอน 9 ตัว ทั้งที่ผลจากการประมาณโดยใช้นิวรอน 9 ตัวให้ error ที่น้อยกว่าแบบ 2 ตัว (การพิจารณา error นั้นพิจารณาจากผลรวมของความแตกต่างระหว่างข้อมูลเอาต์พุตที่ได้จากเครือข่ายประสาทเทียม เทียบกับเอาต์พุตที่คาดหวังจริงๆ ซึ่งแม้ดูด้วยตาแล้วผลอาจไม่ได้ดีกว่า แต่เมื่อรวมผลต่างแล้วอาจให้ผลที่น้อยกว่าก็ได้) หรือปัญหาอาจเกิดเนื่องมาจากข้อมูลที่นำไปสอนเพื่อสร้างเครือข่ายประสาทเทียม อาจมีความผิดพลาด

อยู่บ้าง (ซึ่งเป็นเรื่องปกติในทางปฏิบัติ) การที่เครือข่ายมีจำนวนนิวรอนมาก เครือข่ายจะเรียนรู้ความผิดพลาดนั้นไปด้วย การแก้ปัญหาในลักษณะนี้จะใช้การทำ generalization กล่าวคือ เป็นการทำให้เครือข่ายประสาทเทียมให้มีความซับซ้อนให้น้อยที่สุด โดยให้เกิด error ในระดับที่ยอมรับได้ แม้ว่า error นั้นอาจไม่น้อยที่สุดก็ตาม ซึ่งการทำในลักษณะนี้จะทำให้เครือข่ายมีความซับซ้อนน้อย ใช้เวลาในการสอนลดลงมาก มีความทนต่อข้อมูลไม่พึงประสงค์ แต่ก็ทำให้ความถูกต้องอาจต่ำกว่าค่าที่ดีที่สุด ดังนั้นในทางปฏิบัตินั้น จำนวนเลเยอร์ของเครือข่ายจะมีไม่มาก (ปกติจะมีเพียง 2 เลเยอร์หรือไม่เกิน 3 เลเยอร์) รวมถึงมีจำนวนนิวรอนไม่มากนัก (อย่างไรก็ดีขึ้นกับลักษณะของปัญหาที่นำเครือข่ายประสาทเทียมไปใช้ด้วย)



1-2-1 network

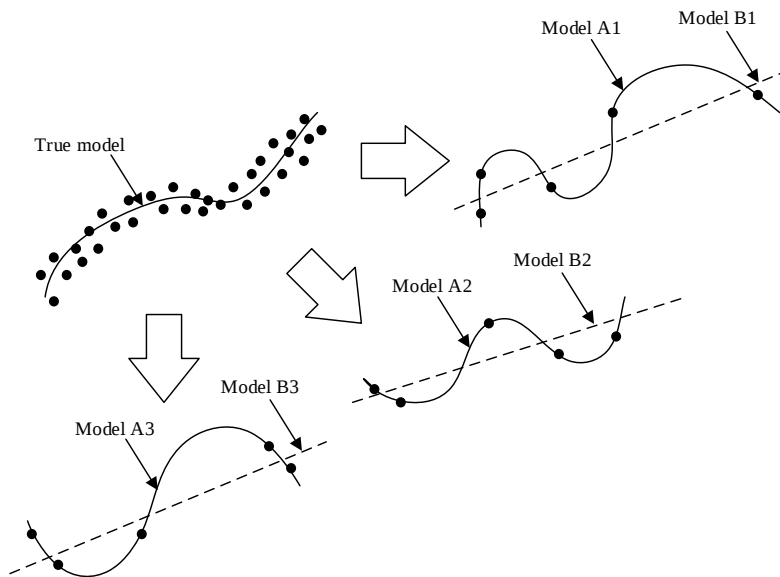


1-9-1 network

รูปที่ 1-14 ตัวอย่างผลกระทบจากการไม่ทำ generalization

การเลือกใช้งานเครือข่ายประสาทเทียมที่มีหลายเลเยอร์ยังมีข้อเสียอันเนื่องมาจากความสามารถในการแก้ปัญหาที่ซับซ้อนได้อีกประการหนึ่งคือ การสอนเพื่อประมาณหาค่าน้ำหนัก (w) และ bias (b) นั้นทำได้ยาก ด้วยความซับซ้อนของเครือข่ายประสาทเทียมอาจพบจุดที่มีค่าความผิดพลาดในการประมาณต่ำสุดมากกว่า 1 จุด ซึ่งแต่ละจุดนั้นมีค่าน้ำหนักที่แตกต่างกันโดยสิ้นเชิง ปัญหานี้ยังรวมไปถึงปัญหาเรื่อง local minimum ที่อัลกอริทึมในการสอน ไปตรวจพบค่าน้ำหนักที่ไม่ได้ดีที่สุด (จะกล่าวถึงต่อไปอีกครั้งหนึ่ง) ดังนั้นการสอนเครือข่ายประสาทเทียมแบบหลายเลเยอร์เพื่อให้ได้มาซึ่งค่าน้ำหนักหรือ bias นั้นจึงมีความยากลำบากมากกว่า ผู้พัฒนาอาจจำเป็นต้องตรวจสอบหรือทดลองซ้ำ หรือแม้แต่อาจจำเป็นต้องหาข้อมูลเพิ่มเติม เพื่อให้มั่นใจได้ว่าจะได้ค่าน้ำหนักและ bias ที่เหมาะสมแล้วจริงๆ

อีกตัวอย่างหนึ่งของการเลือกโมเดลของเครือข่ายประสาทเทียมแสดงดังรูปที่ 1-15 หากในการพัฒนาเครือข่ายประสาทเทียม ไม่สามารถได้ข้อมูลที่เกี่ยวข้องได้ทั้งหมด ในขณะที่ข้อมูลก็อาจไม่ได้มีความสมบูรณ์ อาจมีความคลาดเคลื่อนจากข้อมูลที่ควรจะเป็นอยู่ด้วย โมเดลที่พัฒนาขึ้นจะไม่สามารถเป็นโมเดลที่ให้ผลที่ตรงกับโมเดลที่แท้จริง (true model) ได้ พิจารณาในเหตุการณ์ที่มีการสุ่มข้อมูลเพียงบางส่วนจากข้อมูลทั้งหมดไปพัฒนาเครือข่ายประสาทเทียม (ซึ่งตรงกับสถานการณ์จริงที่ผู้พัฒนาสามารถได้ข้อมูลเพียงบางส่วนมาพัฒนาอยู่แล้ว) หากเครือข่ายประสาทเทียมมีความซับซ้อน ผลที่ได้อาจมีความแตกต่างกันมาก และแตกต่างกับโมเดลที่แท้จริงด้วย ทั้งนี้ขึ้นกับข้อมูลที่สุ่มไปได้ (model A1, A2, และ A3) ทั้งที่ผลการสอน (train) เครือข่ายประสาทเทียมให้ผลที่ดีมาก (เส้นกราฟผ่านทุกจุดที่สุ่มมาทั้งหมด) ในทางกลับกันหากเครือข่ายประสาทเทียมไม่มีความซับซ้อน กลับกลายเป็นว่าโมเดลกลับมีรูปร่างที่คล้ายกัน (model B1, B2, และ B3) แต่ผลการสอนกลับมีค่าความผิดพลาดสูง (เส้นกราฟแทบไม่ผ่านทุกจุดที่สุ่มมา) ในตัวอย่างนี้จึงอธิบายได้เป็นอย่างดีว่า การใช้เครือข่ายประสาทเทียมที่ซับซ้อน แม้ว่าอาจจะได้ผลการสอนที่ดีมาก แต่ไม่ได้หมายความว่าจะสามารถใช้งานได้ดีกับข้อมูลอื่นที่ไม่ได้นำมาสอน ทั้งที่ใช้เวลาสอนมากกว่าและมีความสามารถในการเรียนรู้มากกว่า ในทางกลับกันเครือข่ายประสาทเทียมแบบง่ายๆ กลับมีความสอดคล้องในโมเดลที่ได้มากกว่า แต่ก็อาจให้ผลการสอนและผลการนำไปใช้งานจริงที่แย่กว่าเช่นกัน การเลือกความซับซ้อนของเครือข่ายประสาทเทียมจึงจำเป็นต้องมีความเหมาะสม ไม่ควรที่จะซับซ้อนมากเกินไปหรือน้อยไป



รูปที่ 1-15 สถานการณ์ตัวอย่าง ที่นำข้อมูลเพียงบางส่วนไปสร้างโมเดลเครือข่ายประสาทเทียม ที่จะให้ผลที่แตกต่างกันหากโมเดลมีความซับซ้อนมากเกินไป

ความซับซ้อนของเครือข่ายประสาทเทียม (model complexity) มีความเกี่ยวข้องกับความผิดพลาดในการเอาเครือข่ายประสาทเทียมไปใช้จริง (prediction error) (รูปที่ 1-16) โมเดลที่ไม่ซับซ้อน จะมีรูปแบบของโมเดลที่เป็นไปได้ต่ำกว่า (variance) เนื่องจากมีจำนวนน้ำหนัก (w) และ bias (b) ที่น้อยกว่านั่นเอง ซึ่งมักให้ค่าความผิดพลาดสูง หากผู้พัฒนาเลือกที่จะใช้โมเดลที่ไม่ซับซ้อนนี้ นักพัฒนามักจะใช้ความคิดเห็นส่วนตัว (bias) ของตนเองในการให้เหตุผลว่าผลความผิดพลาดที่ได้น่าจะ “ยอมรับได้” แต่ในโมเดลที่มีความซับซ้อน จะมีรูปแบบของโมเดลที่เป็นไปได้สูงมาก ทำให้ผลการสอนให้ค่าความผิดพลาดที่ต่ำมาก ผู้พัฒนาไม่จำเป็นต้องอิงความคิดเห็นส่วนตัวในกรณีนี้ แต่โมเดลนี้มักจะใช้ได้เฉพาะกับข้อมูลที่นำมาสอนเท่านั้น เมื่อนำไปใช้กับข้อมูลอื่นที่ไม่ได้สอน มักให้ค่าความผิดพลาดที่สูง ในกรณีนี้ผู้พัฒนาจึงควรหาจุดที่สมดุลของระดับความซับซ้อนของเครือข่ายประสาทเทียม (optimal complexity) ซึ่งอาจใช้เวลานาน แต่ก็จำเป็น