หอสมุดกลาง สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 2

นิวรอลเนตเวิร์ก

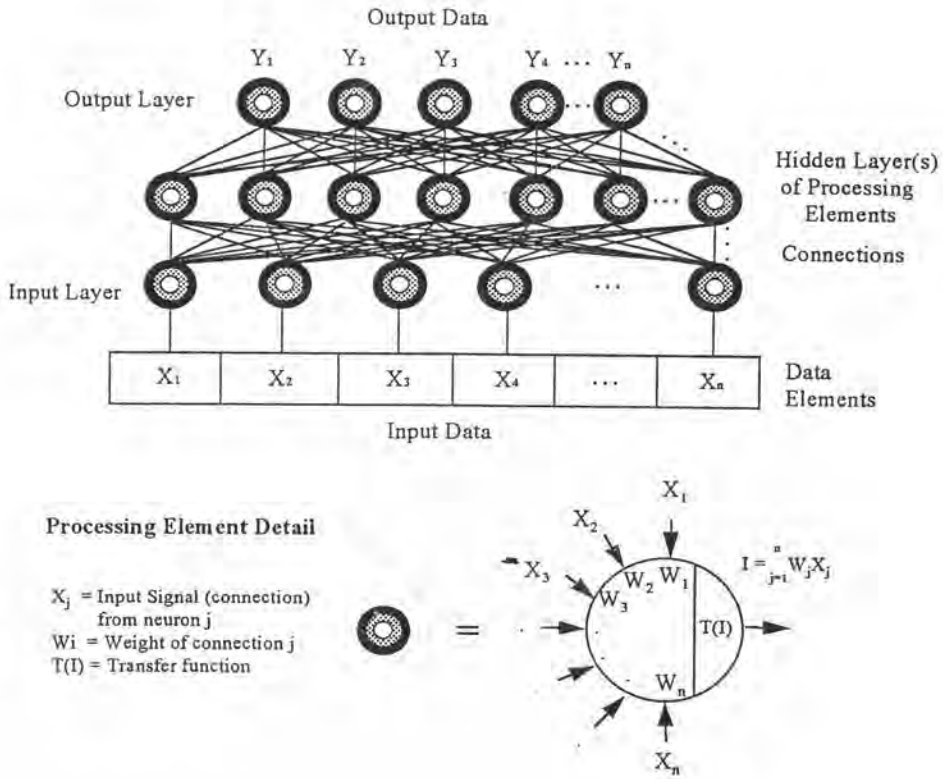
นิวรอลเนตเวิร์กคือ ระบบที่ประกอบไปด้วยหน่วยประมวลผลแบบง่ายๆหลายๆตัวที่อาจเชื่อมโยงกันอย่างทั่วถึงหรือไม่ทั่วถึงก็ได้ (Fully Connection or Partial Connection) การทำงานของระบบขึ้นกับลักษณะการเชื่อมโยงระหว่างหน่วยประมวลผลและค่าน้ำหนัก เป็นชุดสถาปัตยกรรมสำหรับการประมวลผลข้อมูลซึ่งอยู่ในรูปแบบต่างๆ อาจเป็นซอฟท์แวร์ หรือโครงสร้างของฮาร์ดแวร์ที่ถูกสอนให้รู้จักรูปแบบข้อมูลแบบต่างๆ สำหรับการทำงานนิวรอลเนตเวิร์กเรียนรู้จากการจำแนกข้อมูลจำนวนมากโดยเทียบกับข้อมูลเข้า-ออกที่กำหนด ปรับตัวแปร หรือค่าน้ำหนักที่จุดที่นิวรอลติดต่อซึ่งกันและกัน จากนั้นชั้นของหน่วยประมวลผลในแต่ละระดับจะสื่อสารกัน หน้าที่ของหน่วยประมวลผลเหล่านี้และโครงสร้างของการเชื่อมโยงได้แนวคิดจากการศึกษาระบบประสาทของมนุษย์

ระบบนี้มีความสามารถของการทำงานระดับสูง (High-Level Function) เช่น การปรับตัวหรือการเรียนรู้ได้โดยมีหรือไม่มีผู้สอน และยังมีความสามารถของการทำงานระดับล่าง (Low-Level Function) เช่น การมองเห็น และการประมวลผลคำพูด

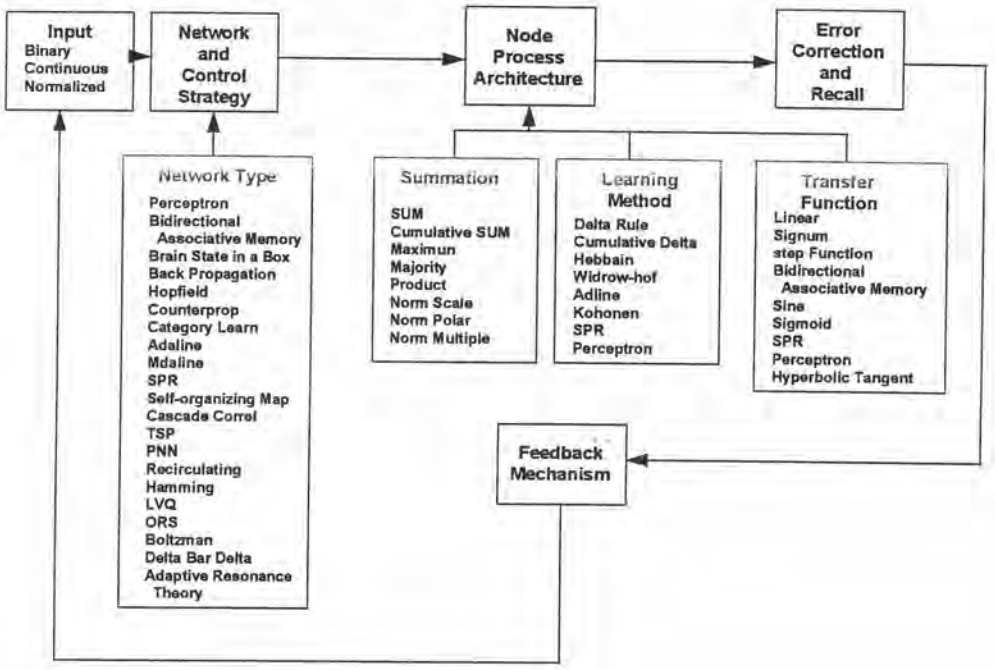
รูปที่ 2.1 แสดงถึงส่วนประกอบของนิวรอลเนตเวิร์กซึ่ง ประกอบด้วยส่วนประกอบหลักดังนี้

- หน่วยประมวลผล (Processing Elements)
- ค่าน้ำหนักและฟังก์ชันการแปลงค่าจากหน่วยความจำไปยังหน่วยประมวลผล (Weight and Transfer Function)
- ชั้นข้อมูลนำเข้า (Input Layer)
- ชั้นแอบแฝงของหน่วยประมวลผล (Hidden Layer)
- ความหนาแน่นของจุดที่นิวรอลติดต่อซึ่งกันและกัน (Connections)
- การเรียนรู้ ซึ่งเกิดจากการปรับค่าน้ำหนักระหว่างหน่วยประมวลผล
- ชั้นแสดงผลลัพธ์ (Output Layer)

รูปที่ 2.2 แสดงถึงภาพของสถาปัตยกรรมของนิวรอลเนตเวิร์ก โดยมีอย่างน้อย 50 ชนิด ที่ใช้ในการวิจัย หรือถูกนำไปพัฒนาระบบงาน และแม้ว่าทุกชนิดประกอบด้วยหน่วยประมวลผล แต่ก็มีความแตกต่างกันของอัลกอริธึมการเรียนรู้



รูปที่ 2.1 ส่วนประกอบของนิวรอลเนตเวิร์ก



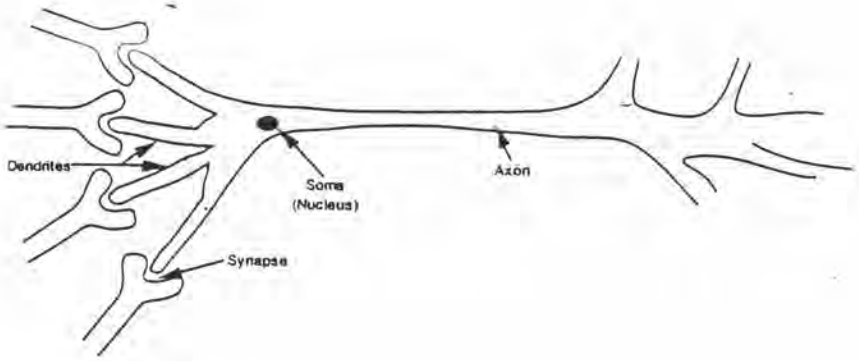
รูปที่ 2.2 สถาปัตยกรรมของนิวรอลเนตเวิร์ก

<u>ประวัติความเป็นมาและวิวัฒนาการ</u>

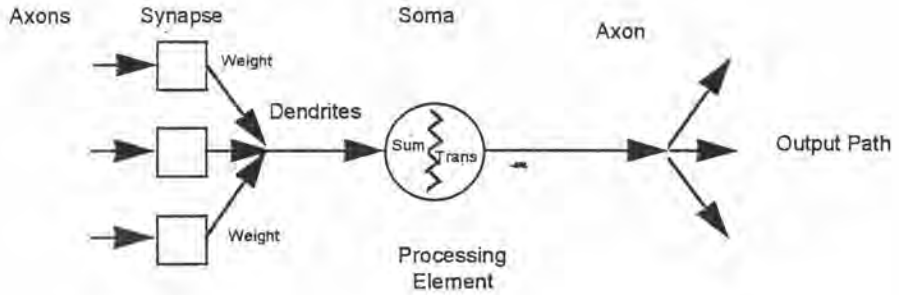
นักประสาทวิทยาได้พัฒนาโมเดลซึ่งแสดงถึงวิธีที่สมองของมนุษย์ประมวลผลข้อมูลภายในสมองเซลประสาทเชื่อมโยงกันอย่างซับซ้อนผ่านโยงใยของเส้นทางนำข้อมูลเข้าและเส้นทางผลลัพธ์ นิวเคลียสของเซลประสาทเปรียบเสมือนหน่วยประมวลผลง่ายๆที่รับสัญญาณไฟฟ้าจากนิวรอลอื่นหลายๆนิวรอลทางเส้นทางนำข้อมูลเข้า หรือที่เรียกว่าเดนไดรท์ (Dendrites) นิวรอลอื่นเป็นร้อยๆนิวรอลจะรับข้อมูลนำเข้านั้นไปเช่นกัน ถ้าผลของข้อมูลนำเข้ามากกว่าระดับการกระตุ้น (Threshold) ซึ่งขึ้นกับค่าความแข็งแกร่งและความถี่ของสัญญาณที่ได้รับจากแต่ละการเชื่อมโยง นิวรอลจะส่งสัญญาณไปยังเส้นทางของผลลัพธ์ (Axon) ดังแสดงในรูปที่ 2.3

นิวรอลที่ถูกระตุ้นอาจเป็นตัวกระตุ้นนิวรอลอื่น แล้วนิวรอลที่ถูกกระตุ้นนั้นก็อาจเป็นตัวกระตุ้นนิวรอลอื่นอีกหลายๆนิวรอลด้วยสัญญาณไฟฟ้า ทุกครั้งที่เหตุการณ์นี้เกิดขึ้นจะทำให้การเชื่อมโยงแข็งแกร่งขึ้น นั่นคือเกิดการเรียนรู้ ในสมองมนุษย์จะประกอบด้วยเซลประสาทหรือนิวรอลประมาณ 100 พันล้านนิวรอล(10^{11}) และมีการเชื่อมโยงกันถึง 10^{15} ซึ่งทำให้เกิดสัญญาณ 10^{16} ครั้งต่อวินาที

Biological Processing Element



A Neural Network Processing Element



รูปที่ 2.3 โมเดลทางชีวภาพ

นิวรอลเนตเวิร์กเป็นการคำนวณชนิดหนึ่งที่ได้รับแรงบันดาลใจจากโมเดลทางชีวภาพ นั่นคือมีความพยายามที่จะลอกเลียนความสามารถของสมองมนุษย์ ทฤษฎีของนิวรอลเนตเวิร์ก เริ่มตั้งแต่ปีทศวรรษ 1950 โดยนักฟิสิกส์ วิศวกร และนักชีววิทยา ผู้บุกเบิกเหล่านี้เริ่มต้นประยุกต์นิวรอลเนตเวิร์กให้แก้ปัญหาในการจำแนกแยกแยะ เช่นการจำแนกแยกแยะตัวอักษร

ปลายทศวรรษ 1950 Frank Rosenblatt จากมหาวิทยาลัยคอร์แนล ได้พัฒนา Perceptron ซึ่งคือระบบประสาทเทียมเลียนแบบการทำงานของระบบประสาทสมองมนุษย์ ระบบของ Rosenblatt ซึ่งสามารถแยกแยะตัวอักษร ทำให้เกิดการพัฒนา เติบโตในงานวิจัยด้านนิวรอลเนตเวิร์ก

ปีค.ศ. 1969 Marvin Minsky ซึ่งเป็นหนึ่งในผู้ก่อตั้งความคิดในเรื่องปัญญาประดิษฐ์ ได้ทำให้แนวทางจำลองแบบการทำงานของสมองต้องประสบความซะงักงันไปสิบกว่าปี โดยเขียน

หนังสือชื่อ Perceptron ร่วมกับ Papert โจมตีว่า Perceptron ปราศจากคุณค่าทางวิทยาศาสตร์ และได้พิสูจน์ให้เห็นว่า Perceptron ขณะนั้นไม่สามารถแก้ปัญหา non-linear logic (exclusive or) และปัญหาทางด้านแยกแยะรูปแบบบางอย่างได้เช่น ไม่สามารถแยกตัว T กับตัว C ออกจากกันได้ (Minsky และ Papert, 1969)

กลุ่มทางด้านนิวรอลเนตเวิร์กไม่ได้มีผลงานอะไรอีกจนกระทั่งประมาณปีค.ศ. 1978 จึงมีการรวมตัวของนักภาษาศาสตร์ นักปรัชญา นักคอมพิวเตอร์ นักชีววิทยา นักฟิสิกส์ นักจิตวิทยา โดยมีศูนย์กลางอยู่ที่มหาวิทยาลัยซานดิเอโก และเรียกตัวเองว่า PDP Group (Parallel Distributed Processing) และนี่เองทำให้นิวรอลเนตเวิร์กได้เกิดใหม่อีกครั้ง และในปี 1986 หนังสือ 2 เล่มชื่อ Parallel Distributed Processing Vol.I-II ที่ขณะนี้ถือกันว่าเป็นคัมภีร์ของนิวรอลเนตเวิร์กได้ตีพิมพ์จำหน่าย (มนตรี วงศ์ทองศรี, 2534)

ตารางที่ 2.1 แสดงถึงยุคต่างๆของการวิจัยทางนิวรอลเนตเวิร์ก

เวลา	ผู้ค้นคว้า	งานวิจัย
แนวความคิดพื้นฐาน ค.ศ.1940-1960	McCulloch & Pitts	Boolean Logic
	Farley & Clark/Hebb	Synaptic Learning Rule
	Rosenblatt	Perceptron
	Stienbuch, Taylor	Associative Memory
ทฤษฎีขั้นพื้นฐาน ค.ศ.1960-1980	Widrow & Hoff	LMS Algorithm (Adaline/Madaline)
	Albus	Cerebellum Model (CMAC)
	Anderson	Linear Associative Learning
	Von Der Malsburg	NGO-Cognitron
	Fukushima	Adaptive Resonance
	Grossberg & Carpenter	Theory (ART) and Boundary Contour System (BCS)
ยุคเฟื่องฟู ค.ศ.1980-ปัจจุบัน	Kohonen	Feature Maps
	Feldman & Ballard	Connectionist Models
	Hopfield	Associative Memory Theory
	Reily, ET. AL.	Reduced Coulomb Energy
	Hinton & Sejnowski	Boltman Machine
	Rumelhart, Et. AL.	Backpropagation and PDP Books
	Edeman, Reeke	Dadrwin III (Neural Darwinish)

ตารางที่ 2.1 ยุคของการวิจัยทางนิวรอลเนตเวิร์ก

<u>แรงจูงใจที่ทำให้มีการศึกษานิวรอลเนตเวิร์ก</u> (Herz, Krogh และ Palmer, 1991)

1. ความแข็งแกร่ง

เซลประสาทสมองมนุษย์มีการตายทุกวัน ซึ่งเป็นสิ่งปกติเช่นเดียวกับเซลผิวหนัง แต่ไม่มีผลต่อประสิทธิภาพในการทำงานของสมอง นั่นคือถ้าส่วนใดส่วนหนึ่งของนิวรอลเนตเวิร์กเสียไปก็ยังทำงานต่อไปได้

2. ความยืดหยุ่น

เนื่องจากมนุษย์มีความสามารถในการเรียนรู้ทำให้สามารถปรับสภาพให้เข้ากับสิ่งแวดล้อมต่างๆ ซึ่งเป็นสิ่งที่ต้องปรับปรุงอยู่เสมอของดิจิตัลคอมพิวเตอร์

3. ความสามารถในการบ่งชี้ข้อมูล

นิวรอลเนตเวิร์กมีความสามารถในการจัดการกับข้อมูลที่คลุมเคลือไม่สมบูรณ์หรือขัดแย้งกันได้สูง

4. ความสามารถในการทำงานคู่ขนาน

นิวรอลเนตเวิร์กมีการทำงานแบบคู่ขนาน จากชั้นหนึ่งซึ่งประกอบด้วยนิวรอลหลายนิวรอลไปอีกชั้นหนึ่งซึ่งประกอบด้วยหลายนิวรอล เช่นเดียวกับการทำงานของระบบประสาทมนุษย์

5. มีขนาดเล็กและใช้พลังงานน้อย

<u>คุณสมบัติของนิวรอลเนตเวิร์ก</u> (Wasserman,1989)

นิวรอลเนตเวิร์กมีคุณสมบัติบางอย่างคล้ายกับสมองมนุษย์ เช่นสามารถเรียนรู้จากประสบการณ์ ตัวอย่าง สามารถปรับตัวเองเข้ากับสิ่งแวดล้อมได้เป็นอย่างดี และสามารถอนุมานจากสิ่งที่เรียนรู้ไปสู่สิ่งที่ไม่เคยเรียนรู้มาก่อน หรือที่เรียกว่าความสามารถในการบ่งชี้ทั่วไป (Generalization)

นิวรอลเนตเวิร์กมีคุณสมบัติ 2 ประการคือ

- การเรียนรู้ (Learning)

นิวรอลเนตเวิร์กสามารถเรียนรู้จากชุดการสอนที่เราป้อนให้

- การระลึกหรือจดจำได้ (Recall)

นิวรอลเนตเวิร์กสามารถระลึกได้ทั้งชุดการสอนและชุดการทดสอบ หรือชุดทั่วไปได้ดีในระดับที่ยอมรับได้ โดยชุดการทดสอบจะมีความแตกต่างจากชุดการสอนอยู่บ้าง ซึ่งความแตกต่างเรียกว่ามีสิ่งปนเปื้อน หรือสิ่งบิดเบือนของข้อมูลประเภทนั้นๆ ซึ่งเป็นลักษณะของสภาพความเป็นจริงที่เกิดขึ้น

<u>ข้อเปรียบเทียบระหว่างการประมวลผลแบบดั้งเดิมกับนิวรอลเนตเวิร์ก</u>

1. การประมวลผลแบบดั้งเดิม (Traditional Approach)

เป็นการวิเคราะห์ข้อมูลโดยมนุษย์ ดังนั้นทรัพยากรมนุษย์จะถูกใช้ในการพัฒนาอัลกอริธึมและโปรแกรม ตัวอย่างเช่น การประมวลภาพในการระบุว่าเป็นแมว, สุนัข หรือกระต่าย ทำให้เกิดโปรแกรม 3 โปรแกรมในการบ่งบอกภาพแต่ละภาพ และเมื่อมีภาพชนิดใหม่ๆขึ้นมา โปรแกรมก็ไม่สามารถที่จะบอกได้ ต้องทำการพัฒนาอัลกอริธึมและโปรแกรมขึ้นมาใหม่

2. การประมวลผลแบบนิวรอลเนตเวิร์ก (Neural Network Approach)

จากตัวอย่างการประมวลภาพแรกเริ่มเราจะให้ภาพแต่ละภาพเป็นข้อมูลนำเข้า และผลลัพธ์เป็นข้อความที่ระบุชนิดของภาพนั้นเช่น แมว, สุนัข หรือกระต่าย โดยผ่านกระบวนการสอนให้นิวรอลเนตเวิร์กรู้จักจำแนกชนิดของภาพต่างๆโดยค่าน้ำหนักจะถูกปรับแต่งอยู่ตลอดเวลา

ในกระบวนการสอน จะมีการสอนวนซ้ำหลายๆรอบ เพื่อให้นิวรอลเนตเวิร์กเกิดการเรียนรู้ เมื่อสิ้นสุดการสอนแล้วนิวรอลเนตเวิร์กก็จะสามารถจำแนกชนิดของภาพได้ และเมื่อมีภาพใหม่ๆที่ต้องการให้นิวรอลเนตเวิร์กรู้จักก็สามารถทำในทำนองเดียวกัน และต้องมีการสอนใหม่ แต่ทำให้ประหยัดเวลาและแรงงานในการพัฒนาโปรแกรมขึ้นมาใหม่

การเก็บข้อมูลของนิวรอลเนตเวิร์กเป็นแบบกระจายและถูกใช้ร่วมกันโดยหลายๆนิวรอล ซึ่งต่างกับแบบดั้งเดิมคือข้อมูลจะเก็บไว้ในหน่วยความจำ

การเก็บข้อมูลของนิวรอลเนตเวิร์กแบบกระจายนั้นทำให้เกิดความซ้ำซ้อน ซึ่งเป็นการเพิ่มความปลอดภัย คือเป็นระบบสำรองทดแทน (Fault/Error Tolerance System) ได้อย่างดี

ชุดการสอนที่ใช้สอนนิวรอลเนตเวิร์กมีผลต่อการเรียนรู้ ถ้าเป็นชุดตัวอย่างที่ดีจะทำให้เกิดการเรียนรู้ได้ดีและเร็ว แต่ถ้าเป็นชุดข้อมูลการสอนที่ไม่ดีทำให้การเรียนรู้ไม่ดีเท่าที่ควร หรือไม่เกิดการเรียนรู้เลย

ตารางที่ 2.2 แสดงการเปรียบเทียบระหว่างการประมวลผลแบบดั้งเดิมกับนิวรอลเนตเวิร์ก

การประมวลผลแบบดั้งเดิม	นิวรอลเนตเวิร์ก
การพัฒนาเน้นไปที่อัลกอริธึมสำหรับทุกกรณีที่เกิดขึ้น	การพัฒนาเน้นที่ระบบการสอน
ต้องเขียนโปรแกรมล่วงหน้า - ใช้กฎและตรรกเพื่อสร้างผลลัพธ์	เรียนรู้จากตัวอย่างเพื่อผลลัพธ์ที่ถูกต้อง
ขึ้นกับกฎ/ตรรก	สถิติ ขึ้นกับความสัมพันธ์
ประมวลผลโดยอาศัยขั้นตอนต่างๆที่แน่นอนและให้ผลออกมาตามที่คาดคะเนเอาไว้	อาศัยวิธีการประมวลผลข้อมูลที่แตกต่างกันไปในแต่ละกรณีของปัญหา
การตัดสินใจต้องใช้ความรู้และข้อมูลทั้งหมด	สามารถจัดการกับข้อมูลที่ไม่สมบูรณ์ หรือข้อมูลที่มีความสับสนได้
ตัดสินใจในรูปของ ใช่หรือ ไม่ใช่ทางคณิตศาสตร์ หรือตรรกศาสตร์	ตัดสินใจโดยอาศัยตัวเลขแสดงน้ำหนัก บนฐานของข้อมูลตัวเลขที่สามารถแสดงความเอนเอียง (Fuzzy) ตลอดจนยอมรับข้อมูลที่ไม่สมบูรณ์ได้
ได้คำตอบเดิมเสมอ	บางครั้งอาจผิดพลาด - จะเตรียมคำตอบที่ยอมรับได้แต่ไม่จำเป็นต้องเป็นคำตอบที่ดีที่สุด
ให้คำตอบที่แน่นอน เมื่อให้เวลาคำนวณอย่างเพียงพอ	ให้คำตอบโดยประมาณสำหรับปัญหาที่ซับซ้อนมากๆ

การประมวลผลแบบดั้งเดิม	นิวรอลเนตเวิร์ก
เหมาะสำหรับระบบที่ต้องการความแม่นยำในการคำนวณ, มีการไหลของข้อมูลเป็นลำดับและมีตรรกที่ดี	เหมาะสมเมื่อไม่มีกฎที่แน่นอนและต้องการการตัดสิน
ประมวลผลข้อมูลในรูปไบนารี	ประมวลผลสัญญาณที่อยู่ในรูปของสัญญาณอนาล็อก ที่แปรปรวนอยู่ตลอดเวลา
อยู่บนพื้นฐานของสัญญลักษณ์	อยู่บนพื้นฐานของรูปแบบ
สามารถอธิบายและตรวจสอบได้	ไม่สามารถอธิบายการตัดสินใจได้ทั้งหมด
ไม่สามารถจัดการกับข้อยกเว้นต่างๆได้	จัดการกับข้อยกเว้นต่างๆได้ดี
ค้นหาสิ่งที่ต้องการจากฐานข้อมูลขนาดใหญ่เพื่อให้ได้สิ่งที่ตรงกับความต้องการ	ค้นหาข้อมูลจากฐานข้อมูลโดยอาศัยหลักของความใกล้เคียงที่สุดเป็นเกณฑ์
มีวิธีการเก็บข้อมูลจำเพาะแน่นอน	เก็บข้อมูลโดยอาศัยเนื้อที่ที่มีอยู่ในสื่อบันทึกข้อมูลเป็นเครื่องบอกตำแหน่งที่เก็บข้อมูลมากกว่าการจำเป็นที่จะต้องรู้ตำแหน่งที่เก็บตั้งแต่แรก
ตายตัว	ปรับได้
แบบอนุกรม	แบบขนาน

ตารางที่ 2.2 (ต่อ) เปรียบเทียบระหว่างการประมวลผลแบบดั้งเดิมกับนิวรอลเนตเวิร์ก

<u>สารสนเทศที่ใช้ในนิวรอลเนตเวิร์ก</u>

1. ข้อมูลนำเข้า (Input)

ต้องเป็นค่าตัวเลข ถ้าเป็นข้อมูลเชิงคุณภาพต้องแปลงให้อยู่ในเชิงปริมาณที่นิวรอลเนตเวิร์กยอมรับเข้าไปเพื่อเรียนรู้ได้ นั่นคือกระบวนการเบื้องต้นก่อนการประเมินผล (Preprocess)

2. ผลลัพธ์ (Output)

คือผลลัพธ์ที่เกิดขึ้นจริง (Actual Output) จากกระบวนการเรียนรู้ของนิวรอลเนตเวิร์ก

3. ค่าน้ำหนัก (Weights)

คือสิ่งที่ได้จากการเรียนรู้ของนิวรอลเนตเวิร์ก หรือเรียกอีกอย่างหนึ่งว่าค่าความรู้ (Knowledge) ค่าน้ำหนักเป็นสิ่งที่สำคัญมากของนิวรอลเนตเวิร์ก และค่าเหล่านี้จะไม่มีการเปลี่ยนแปลงอีกต่อไป ค่าเหล่านี้จะใช้ในการระลึกข้อมูลอื่นๆที่อยู่ในรูปแบบเดียวกัน

4. ฟังก์ชันผลรวม (Summation Function)

เป็นผลรวมของข้อมูลนำเข้า (a_i) และค่าน้ำหนัก

$$S = \sum_{i=1}^{n} a_i w_i$$

5. ฟังก์ชันการแปลงค่า (Transfer Function)

เป็นการคำนวณการจำลองการทำงานของนิวรอลต่างๆ ฟังก์ชันที่นิยมใช้คือฟังก์ชันซิกมอยด์ (Sigmoid Function)

$$f(x) = 1 / (1 + e^{-x})$$

จุดประสงค์ของการแปลงค่าคือ เพื่อให้ผลลัพธ์ที่ได้อยู่ในช่วง 0-1 มิเช่นนั้นค่าที่ได้จะโตมาก ดังนั้นจึงต้องมีการแปลงค่าข้อมูลที่ใช้ในการสอนและทดสอบ เพื่อให้สอดคล้องกับฟังก์ชันที่ใช้

ประเภทการเรียนรู้ของนิวรอลเนตเวิร์ก

การเรียนรู้ของนิวรอลเนตเวิร์กสามารถแบ่งออกเป็น 3 ประเภทคือ

1. การเรียนรู้แบบมีครู (Supervised Learning)

การเรียนรู้แบบมีครูต้องการชุดข้อมูลนำเข้าและผลลัพธ์เป้าหมายเป็นชุดการสอนควบคู่ (Training Pair) โดยปกติการสอนเนตเวิร์กจะใช้ชุดการสอนควบคู่หลายชุด ในระหว่างการสอนเนตเวิร์กจะเกิดผลลัพธ์จริงขึ้น ผลต่างระหว่างผลลัพธ์จริงกับผลลัพธ์เป้าหมายคือค่าความคลาดเคลื่อน หรือค่าความผิดพลาด

การเรียนรู้ประเภทนี้ได้แก่ Adaptive Linear Neural Element (ADALINE) Backpropagation Recurrent Neural Network (RNN) เป็นต้น

2. การเรียนรู้แบบไม่มีครู (Unsupervised Learning)

การเรียนรู้แบบไม่มีครูถูกพัฒนาขึ้นเพื่อให้ใกล้เคียงกับระบบการเรียนรู้ของสมองมนุษย์มากยิ่งขึ้น โดยมีเพียงชุดข้อมูลนำเข้า กระบวนการเรียนรู้จะใช้หลักการทางสถิติ โดยหาค่าทางสถิติของชุดการสอน และจัดกลุ่มข้อมูลออกเป็นระดับต่างๆ นิวรอลเนตเวิร์กจะหาค่าผลลัพธ์เอง จากความสัมพันธ์ระหว่างข้อมูลนำเข้าและผลลัพธ์

การเรียนรู้ประเภทนี้ได้แก่ Hopfield Network Neocognitron และ Self Organizing Feature Map (SOFM) เป็นต้น

3. การเรียนรู้เชิงบังคับ (Reinforcement Learning)

การเรียนรู้เชิงบังคับเป็นการเรียนรู้ทั้งแบบมีครูและไม่มีครู การเรียนรู้แบบไม่มีครูคือในระหว่างการสอนมีเพียงชุดข้อมูลนำเข้า การเรียนรู้แบบมีครูคือเมื่อได้ผลลัพธ์จะบอกว่าถูกหรือผิด แต่ไม่ได้บอกว่าผลลัพธ์ที่ถูกคืออะไร

การเรียนรู้แบบย้อนกลับ (Backpropagation)

Werbos ได้เสนอแนวความคิดของการเรียนรู้แบบย้อนกลับในปีค.ศ. 1974 จากนั้น Parker ได้นำเสนออีกครั้งในปีค.ศ. 1982 และได้ทำให้เป็นที่รู้จักมากยิ่งขึ้นโดย Rumelhart และคณะในปีค.ศ. 1986 ในหนังสือชื่อ Parallel Distributed Processing ซึ่งกล่าวถึงศักยภาพของนิวรอลเนตเวิร์กและการเรียนรู้แบบย้อนกลับ

งานประยุกต์เริ่มแรกของการเรียนรู้แบบย้อนกลับเป็นโปรแกรม NETTALK ซึ่งสร้างโดย Sejnowski และ Rosenberg ในมหาวิทยาลัยจอห์นฮอปกิน

การเรียนรู้แบบย้อนกลับเป็นรูปแบบหนึ่งของนิวรอลเนตเวิร์กที่ใช้กันอย่างกว้างขวาง และนำมาประยุกต์กับการศึกษาในงานประเภทต่างๆ เช่น ทางการทหาร การช่วยในการวินิจฉัยของแพทย์ การรู้จำคำพูด การรู้จำตัวอักษร

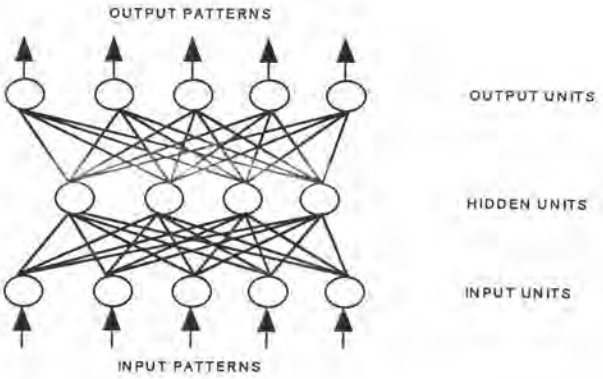
การเรียนรู้แบบย้อนกลับสามารถแก้ปัญหาที่ต้องการรูปแบบ โดยการป้อนรูปแบบเข้าไป นิวรอลเนตเวิร์กก็จะให้รูปแบบผลลัพธ์ที่เกี่ยวข้องกันออกมา (Dayhoff, 1990)

การเรียนรู้แบบย้อนกลับเป็นวิธีการหนึ่งของนิวรอลเนตเวิร์กที่ง่ายต่อความเข้าใจ เนื่องจากกระบวนการเรียนรู้และปรับปรุงแก้ไขนั้นเป็นไปด้วยตัวเอง ถ้านิวรอลเนตเวิร์กให้คำตอบที่ผิด ค่าน้ำหนักก็จะถูกปรับจนกว่าค่าความผิดพลาดจะน้อยลง หรืออยู่ในเกณฑ์ที่ยอมรับได้ นั่นคือค่าที่ได้ในครั้งต่อๆไปจะมีความถูกต้องมากยิ่งขึ้น

การเรียนรู้แบบย้อนกลับมีโครงสร้างเป็นชั้นๆ แต่ละชั้นเชื่อมโยงกันอย่างทั่วถึงกับชั้นที่อยู่ด้านบนและชั้นที่อยู่ด้านล่าง เมื่อเนตเวิร์กได้รับข้อมูลนำเข้าจะทำการคำนวณค่าน้ำหนักของหน่วยประมวลผลนำเข้าไปยังชั้นแอบแฝง และจากชั้นแอบแฝงไปยังชั้นแสดงผลลัพธ์ เมื่อเกิดผลต่างระหว่างผลลัพธ์จริงกับผลลัพธ์เป้าหมาย เนตเวิร์กจะปรับค่าความผิดพลาดจากหน่วยแสดงผลลัพธ์และแพร่ย้อนกลับไปยังชั้นแอบแฝง จากชั้นแอบแฝงไปยังชั้นข้อมูลนำเข้า

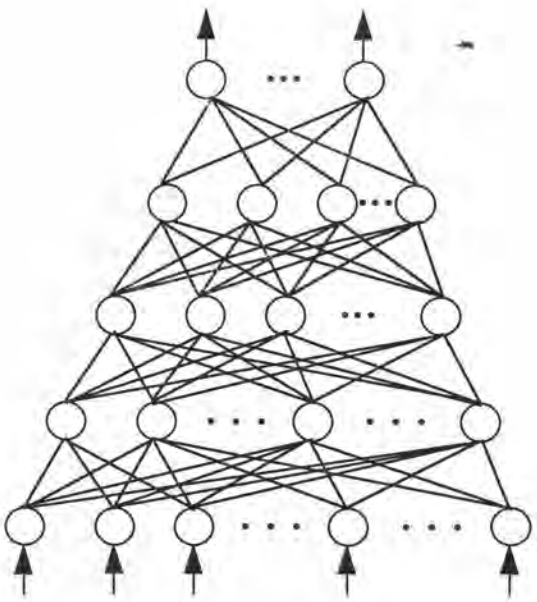
โครงสร้างของการเรียนรู้แบบย้อนกลับ

การเรียนรู้แบบย้อนกลับมีโครงสร้างเป็นชั้นๆ โดยมีโครงสร้างอย่างง่ายๆ 3 ชั้นคือ ชั้นข้อมูลนำเข้า ชั้นแอบแฝงและชั้นแสดงผลลัพธ์ แต่ละชั้นจะติดต่อกันอย่างสมบูรณ์ดังรูปที่ 2.4



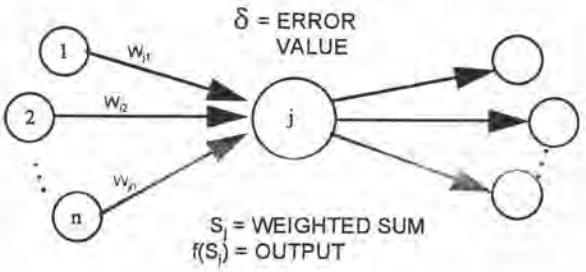
รูปที่ 2.4 รูปแบบของการเรียนรู้แบบย้อนกลับชนิด 3 ชั้น

จากรูปที่ 2.4 แสดงรูปแบบของการเรียนรู้แบบย้อนกลับชนิด 3 ชั้น ชั้นล่างคือชั้นข้อมูลนำเข้าซึ่งนำข้อมูลจากภายนอก ชั้นถัดมาคือชั้นแอบแฝงซึ่งจะติดต่อกับทั้งชั้นข้อมูลนำเข้าและชั้นแสดงผลลัพธ์ที่อยู่ด้านบน



รูปที่ 2.5 รูปแบบของการเรียนรู้แบบย้อนกลับชนิด 5 ชั้น

จากรูปที่ 2.5 เป็นตัวอย่างของการเรียนรู้แบบย้อนกลับชนิด 5 ชั้นที่เชื่อมโยงติดต่อกันอย่างสมบูรณ์ โดยมีชั้นแอบแฝง 3 ชั้น



รูปที่ 2.6 หน่วยประมวลผล

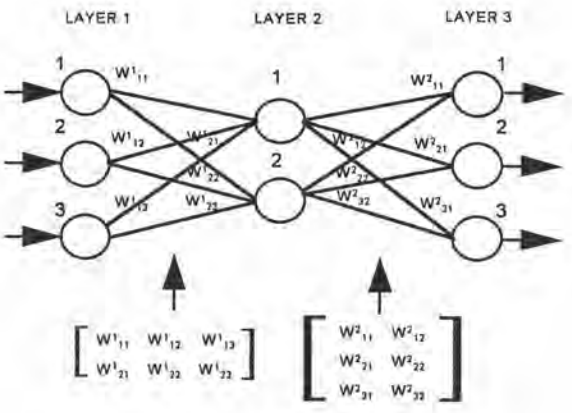
จากรูปที่ 2.6 จะเห็นว่าข้อมูลนำเข้าอยู่ทางด้านซ้าย ทางด้านขวาคือหน่วยที่รับผลลัพธ์จากหน่วยที่ j

หน่วยประมวลผลกลางประกอบด้วย ผลรวมถ่วงน้ำหนักของข้อมูลนำเข้า(S_j) ผลลัพธ์ ($f(S_j)$) และค่าความผิดพลาด(δ_j) ที่ใช้ในการปรับค่าน้ำหนัก

ค่าน้ำหนักที่เกี่ยวเนื่องกับการติดต่อแต่ละชั้นจะถูกปรับแต่งตลอดเวลาระหว่างการเรียนรู้ เพื่อลดค่าความแตกต่างระหว่างผลลัพธ์ที่ได้กับผลลัพธ์ที่ต้องการ

ค่าน้ำหนักจากหน่วยที่ i ไปยังหน่วยที่ j คือ W_{ji} เมื่อจบการเรียนรู้แล้วค่าน้ำหนักนี้จะใช้ในกระบวนการการระลึก

รูปที่ 2.7 แสดงถึงการแทนค่าน้ำหนักด้วยเมตริกซ์ โดยมีค่ากำกับบน (Subscript) บ่งบอกถึงชั้นที่ต่างกัน เช่น W^1_{21} หมายถึงค่าน้ำหนักจากหน่วยประมวลผลที่ 2 ในระดับชั้นที่ 1 ไปยังหน่วยที่ประมวลผลที่ 1 ในระดับชั้นที่ 2



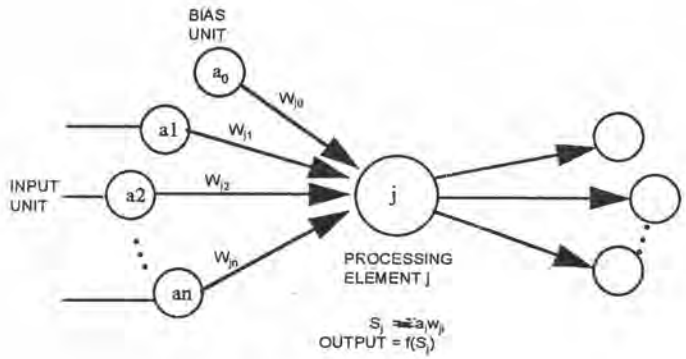
รูปที่ 2.7 การแทนค่าน้ำหนักด้วยเมตริกซ์

<u>ขั้นตอนการเรียนรู้แบบย้อนกลับ</u>

ประกอบด้วย 2 ขั้นตอน

1. การแพร่เดินหน้า (Forward Propagation)

ขั้นตอนนี้เริ่มเมื่อเนตเวิร์กได้รับข้อมูลนำเข้าและค่าของหน่วยประมวลผลในชั้นข้อมูลนำเข้าถูกกำหนด ชั้นอื่นๆจึงจะเริ่มทำการแพร่เดินหน้า



รูปที่ 2.8 ขั้นตอนการแพร่เดินหน้า

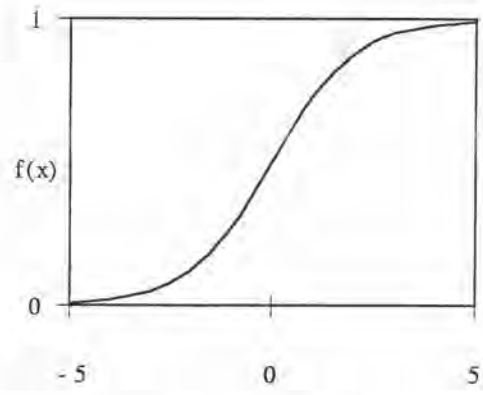
จากรูปเป็นการคำนวณผลรวมของผลลัพธ์ที่เข้ามายังหน่วยที่ j ดังสมการ

$$S_j = \sum a_i w_{ji}$$

โดยที่ a_i = ข้อมูลจากหน่วยที่ i

w_{ji} = ค่าน้ำหนักจากหน่วยที่ i ไปยังหน่วยที่ j

จากนั้นจะทำการแปลงค่าข้อมูลโดยคำนวณหาค่า $f(S_j)$ โดยซิกมอยด์ฟังก์ชัน



รูปที่ 2.9 ซิกมอยด์ฟังก์ชัน

จากรูปที่ 2.9 ปลายทั้งสองข้างของเส้นกราฟมีความชันน้อย แต่เส้นกราฟจะมีความชันมากในช่วงกลาง นั่นคือเมื่อ x น้อยกว่า -3 ค่าของ f(x) จะเข้าใกล้ 0 และเมื่อ x มากกว่า 3 ค่าของ f(x) จะเข้าใกล้ 1 อีกนัยหนึ่งคือ ค่าของ f(x) จะเข้าใกล้ 1 เมื่อ x มีค่ามาก และ f(x) จะเข้าใกล้ 0 เมื่อ x มีค่าน้อย สมการของซิกมอยด์คือ

$$f(x) = 1 / 1 + e^{-x}$$

เมื่อ x เป็นค่าผลรวมของหน่วยที่ j ดังนั้น

$$f(S_j) = 1 / 1 + e^{-Sj}$$

เมื่อได้รับค่า f(Sj) แล้วค่า f(Sj) จะกลายเป็นผลลัพธ์ของหน่วยที่ j ซึ่งก็คือค่า a_j ดังรูปที่ 2.8 โดยจะส่งค่าออกไปยังหน่วยอื่นๆด้วยค่า a_j ที่เท่ากัน

สำหรับชั้นข้อมูลนำเข้าจะเป็นกรณีพิเศษ โดยที่หน่วยประมวลผลในชั้นนี้จะไม่ทำการคำนวณผลรวมจากข้อมูลนำเข้าและจะไม่ทำการแปลงค่าข้อมูล เนื่องจากแต่ละหน่วยประมวลผลจะใช้ค่าของข้อมูลนำเข้าเป็นค่าของตัวเอง

จากรูปที่ 2.8 หน่วยเอนเอียง (Bias Unit) เป็นหน่วยที่มีค่าเป็น 1 และเชื่อมโยงกับหน่วยอื่นทุกๆหน่วย และค่าน้ำหนักของหน่วยนี้จะถูกปรับในระหว่างการเรียนรู้เช่นกัน หน่วยนี้จะให้ค่าคงที่กับหน่วยอื่นซึ่งบางครั้งทำให้การเรียนรู้ (Convergence Time) ของ

พิพิธภัณฑ์ สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

นิวรอลเนตเวิร์กเร็วขึ้น หน่วยเอนเอียงมีผลต่อระดับการกระตุ้น (Threshold) ของหน่วยอื่น นั่นคือ เมื่อค่าน้ำหนักคงที่และ

$$C = w_{j0}$$

$$S_j = \sum a_i w_{ji}$$

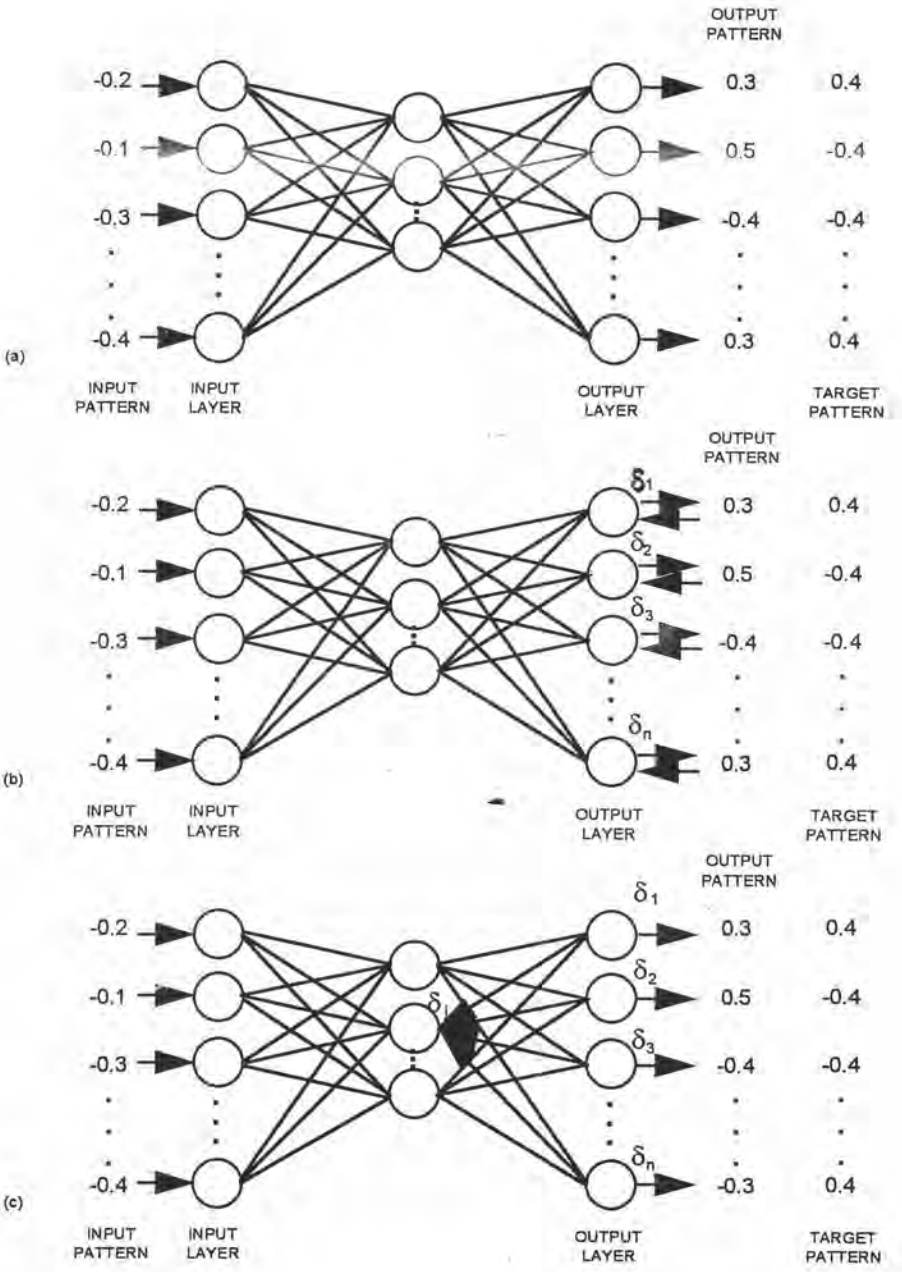
ดังนั้นผลรวมของหน่วยที่ j = S + C ซึ่งจะทำให้กราฟของสมการซิกมอยด์ขยับไปทางซ้าย C หน่วยดัง ด้วยวิธีนี้จะทำให้ระดับการกระตุ้นของหน่วยเป้าหมายเปลี่ยนไป

เหตุที่ต้องใช้ซิกมอยด์เนื่องจากต้องการให้ระดับการกระตุ้นเป็นแบบอ่อน (Soft Threshold) มากกว่าการกระตุ้นแบบแรง (Hard Threshold) นั่นคือซิกมอยด์ฟังก์ชันให้ค่าที่ต่อเนื่องกัน

2. การแพร่ย้อนกลับ (Backward Propagation)

จากรูป 2.10 แสดงขั้นตอนการแพร่ย้อนกลับ ขั้นนี้ค่าความผิดพลาด δ จะถูกคำนวณสำหรับทุกหน่วยและคำนวณค่าน้ำหนักที่จะเปลี่ยนทุกการเชื่อมโยง การคำนวณนี้เริ่มที่ชั้นแสดงผลลัพธ์และแพร่ย้อนกลับไปยังชั้นข้อมูลนำเข้า

ขั้นปรับปรุงความผิดพลาดเกิดหลังจากที่ดำเนินการแพร่เดินหน้าแล้ว หน่วยประมวลผลแต่ละหน่วยในชั้นแสดงผลลัพธ์จะให้ค่าผลลัพธ์ ที่จะเปรียบเทียบกับผลลัพธ์เป้าหมายในชุดการสอน(รูปที่ 2.10a) ความแตกต่างที่เกิดคือค่าความผิดผลาดสำหรับแต่ละหน่วยในชั้นแสดงผลลัพธ์ดังรูปที่ 2.10b ค่าน้ำหนักของทุกการเชื่อมโยงไปยังชั้นแสดงผลลัพธ์จะถูกปรับ จากนั้นค่าความผิดผลาดของหน่วยในชั้นแอบแฝงที่ต่ำกว่าชั้นแสดงผลลัพธ์(รูปที่ 2.10c) จะถูกคำนวณ แล้วค่าน้ำหนักของทุกการเชื่อมโยงไปยังชั้นแอบแฝงจะถูกปรับ กระบวนการนี้จะดำเนินไปจนกว่าชั้นสุดท้ายจะถูกปรับค่าน้ำหนัก



รูปที่ 2.10 ขั้นตอนการแพร่ย้อนกลับ

ค่า δ คือค่าความผิดผลาดท้องถิ่น (Local Error) สามารถหาได้จากการคำนวณง่ายๆ สำหรับชั้นแสดงผลลัพธ์ และซับซ้อนขึ้นสำหรับชั้นแอบแฝง ถ้าหน่วย j เป็นหน่วยที่อยู่ในชั้นแสดงผลลัพธ์แล้วการคำนวณค่าความผิดพลาดจะได้จากสมการ

$$\delta_j = (t_j - a_j)f'(S_j)$$

เมื่อ t_j = ค่าเป้าหมายของหน่วยที่ j

a_j = ค่าผลลัพธ์ของหน่วยที่ j

$f'(x)$ = อนุพันธ์ของซิกมอยด์ฟังชัน

S_j = ผลรวมของข้อมูลนำเข้ามาหน่วยที่ j

ถ้าหน่วย j เป็นหน่วยที่อยู่ในชั้นแอบแฝงแล้วค่าความผิดพลาดจะได้จาก

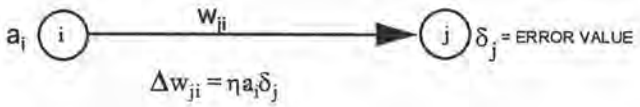
$$\delta_j = [\sum \delta_k w_{kj}]f'(S_j)$$

การปรับค่าน้ำหนักจะปรับโดยใช้ค่าของ δ ทุกหน่วยที่รับผลลัพธ์จากหน่วยที่ j ค่าน้ำหนักแต่ละค่าจะถูกปรับโดยนำค่า δ ของหน่วยที่รับจากข้อมูลนำเข้าจากการเชื่อมโยงนั้น ค่าน้ำหนักจะถูกปรับโดยสมการ

$$w^{new}_{ji} = w^{old}_{ji} + \Delta w_{ji}$$

$$\Delta w_{ji} = \eta \delta_j a_i$$

รูปที่ 2.11 แสดงถึงการปรับค่าน้ำหนัก w_{ji} ซึ่งค่านี้ขึ้นกับตัวแปร 3 ตัวคือ η δ_j และ a_i



รูปที่ 2.11 การปรับค่าน้ำหนัก

ตัวแปร η คือค่าสัมประสิทธิ์การเรียนรู้ซึ่งจะถูกกำหนดโดยผู้ใช้ ถ้าค่า η มากจะทำให้เนตเวิร์กไม่คงที่และการเรียนรู้ไม่เป็นที่น่าพอใจ ถ้าเล็กเกินไปจะทำให้การเรียนรู้ช้า

บางครั้งค่า η สามารถเปลี่ยนได้เพื่อทำให้การเรียนรู้มีประสิทธิภาพขึ้น เช่นเริ่มต้นใช้ค่า η มากๆ และลดลงในระหว่างการสอน

นอกจากค่า η แล้วยังมีค่าโมเมนตัม α ซึ่งคิดค้นโดย Rumelhart Hinton และ William ในปีค.ศ.1986 เพื่อช่วยให้การเรียนรู้เร็วขึ้น ดังนั้นค่าน้ำหนักจะถูกปรับโดยสมการ

$$\Delta w_{ji}(k+1) = \eta \delta_j a_i + \alpha[\Delta w_{ji}(k)]$$

เมื่อ k คือดัชนีเวลา (Time Index) หรือจำนวนรอบของการปรับปรุงค่าน้ำหนัก

<u>การสอนการเรียนรู้แบบย้อนกลับ</u>

การสอนเนตเวิร์กชนิดการเรียนรู้แบบย้อนกลับ เป็นการสอนแบบมีครู โดยมีการกำหนดรูปแบบข้อมูลนำเข้าควบคู่กับรูปแบบเป้าหมาย มีการกำหนดชุดการสอนหลายๆรูปแบบเพื่อให้นิวรอลเนตเวิร์กสามารถเรียนรู้

ในกระบวนการสอนเพื่อให้นิวรอลเนตเวิร์กสามารถเรียนรู้ได้ดีนั้นจะต้องใช้การวนซ้ำสอนหลายๆรอบ การสอนนิวรอลเนตเวิร์กจะสำเร็จหรือไม่ดูจากค่า RMS ดังสมการ

$$RMS = \sqrt{\sum_p \sum_j (t_{jp} - a_{jp})^2 / n_p n_o}$$

โดย n_p = จำนวนของรูปแบบในชุดการสอน

n_o = จำนวนหน่วยในชั้นแสดงผลลัพธ์

t_{jp} = ค่าเป้าหมายของหน่วยที่ j หลังจากเสนอรูปแบบ p

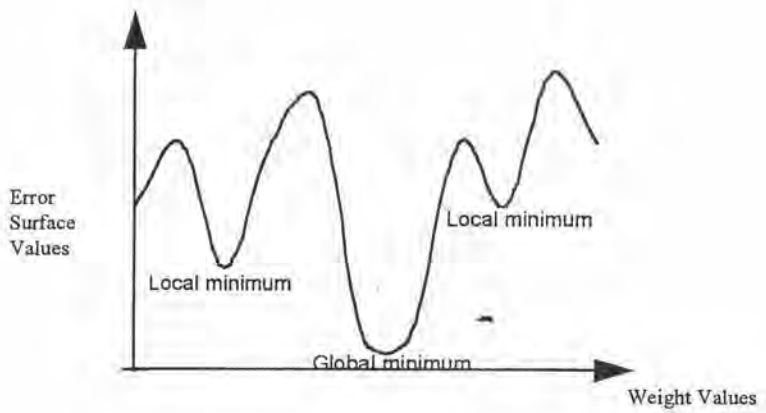
a_{jp} = ค่าผลลัพธ์ที่ได้จากหน่วยที่ j หลังจากเสนอรูปแบบ p

ซึ่งสมการนี้ใช้ในการพิจารณาหาจุดเหมาะสมสำหรับรูปแบบหรือสมการการพยากรณ์ (Optimal Forecasting Models) ด้วยการประมาณค่าพารามิเตอร์ของสมการที่ทำให้ค่า RMS มีค่าน้อยที่สุดด้วยเช่นกัน (สุพล ดุรงค์วัฒนา, 2537)

เมื่อค่า RMS เข้าใกล้ศูนย์ หรือลดลงต่ำกว่า 0.1 หมายความว่าเนตเวิร์กได้เกิดการเรียนรู้แล้ว

ถ้าเนตเวิร์กตกอยู่ภายใต้จุดต่ำสุดท้องถิ่น (Local Minimum) นั่นคือเนตเวิร์กได้หยุดการเรียนรู้แล้ว หรือไม่สามารถระลึกชุดการทดสอบได้เลย แม้ว่าจะระลึกชุดการสอนได้

ถ้าเนตเวิร์กตกอยู่ต่ำสุดโดยรวม (Global Minimum) นั่นคือ ค่า RMS ของเนตเวิร์กได้ถึงจุดต่ำสุดแล้ว และสามารถระลึกได้ทั้งชุดการสอนและชุดการทดสอบได้ในระดับที่พอใจ



รูปที่ 2.12 จุดต่ำสุดท้องถิ่นและจุดต่ำสุดโดยรวม

<u>ข้อดีและข้อจำกัดของการเรียนรู้แบบย้อนกลับ</u> (Dayhoff, 1990)

1. ข้อดี

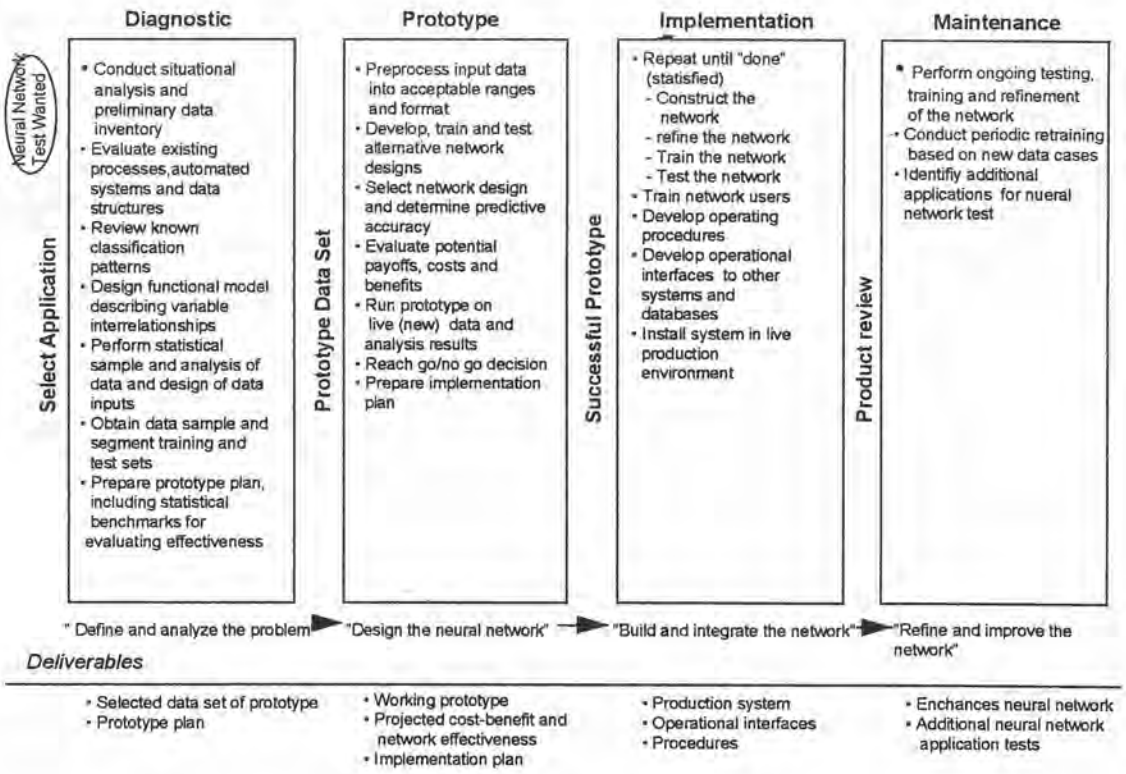
คือมีความสามารถในการจดจำรูปแบบของปัญหา (Pattern Mapping) ซึ่งการเรียนรู้แบบย้อนกลับสามารถที่จะเรียนรู้ความสัมพันธ์ของรูปแบบได้มากมาย โดยการเรียนรู้แบบย้อนกลับต้องการตัวอย่างรูปแบบที่จะเรียนรู้ ไม่ต้องการความรู้ทางด้านคณิตศาสตร์เพื่อแมพ (map) รูปแบบของข้อมูลนำเข้าและผลลัพธ์ ความยืดหยุ่นของการเรียนรู้อยู่ที่ความหลากหลายในการออกแบบทางเลือกต่างๆ เช่น จำนวนชั้น เส้นเชื่อมโยง จำนวนของนิวรอล ค่าสัมประสิทธิ์การเรียนรู้ ค่าโมเมนตัม ที่เรากำหนดขึ้นและการแทนรูปแบบของข้อมูล ความยืดหยุ่นนี้ทำให้การเรียนรู้แบบย้อนกลับสามารถแก้ปัญหางานประยุกต์ได้อย่างมากมาย

2. ข้อจำกัด

คือการใช้เวลามากในการสอนนิวรอลเนตเวิร์กให้เกิดการเรียนรู้ (Convergence Time) สำหรับการเรียนรู้การแก้ปัญหา สำหรับระบบงานจริงอาจมีชุดตัวอย่างถึง 1,000 ตัวอย่างและต้องใช้เวลาในการคำนวณเป็นวันหรือมากกว่านั้นเพื่อให้การสอนสมบูรณ์

จุดอ่อนอีกข้อหนึ่งของการเรียนรู้แบบย้อนกลับคือความล้มเหลวในการสอน นั่นคือเนตเวิร์กตกอยู่ที่จุดต่ำสุดท้องถิ่น โดยที่การเพิ่มชุดการสอนไม่สามารถปรับเนตเวิร์กได้ หรือก็คือเนตเวิร์กหยุดการเรียนรู้แล้ว แต่ข้อจำกัดนี้สามารถแก้ไขได้โดย ปรับค่า η และ α เพิ่มจำนวนนิวรอลในชั้นแอบแฝง หรือเพิ่มจำนวนชั้นแอบแฝง เปลี่ยนฟังก์ชันการแปลงค่า เปลี่ยนชุดการสอน หรือเปลี่ยนโครงสร้างของเนตเวิร์ก

ขั้นตอนการทำให้ระบบงานนิวรอลเนตเวิร์กเกิดผล (Implementation) (Apicella และ Villars , 1993)



รูปที่ 2.13 ขั้นตอนการ implement ระบบงานนิวรอลเนตเวิร์ก

1. ขั้นวินิจฉัย (Diagnostic Phase)

ปัญหาจะถูกวิเคราะห์เพื่อตัดสินใจว่านิวรอลเนตเวิร์กจะเหมาะสมกับปัญหานั้นหรือไม่ ระบบงานบางอย่างก็เหมาะสมกับแนวทางของนิวรอลเนตเวิร์ก ซึ่งส่วนใหญ่จะเป็นการแยกแยะรูปแบบ (Pattern Recognition) หรือ การแมพ (map) ทางสถิติ แต่ระบบงานบางอย่าง เช่น งานเงินเดือน หรืองานบัญชีก็ไม่เหมาะสม เพราะต้องการความถูกต้องเที่ยงตรง

เมื่อปัญหาถูกกำหนดจะมีการศึกษาเรื่องของข้อมูล และการแบ่งหมวดหมู่ของข้อมูล (เช่น การแบ่งกลุ่ม) สาเหตุสำคัญที่จะทำให้ระบบงานนิวรอลเนตเวิร์กจะสำเร็จคือความเข้าใจถึงปัญหาเป็นอย่างดีและรู้ว่าข้อมูลใดที่เหมาะสม เพื่อช่วยให้ความเข้าใจดีขึ้น โมเดลการทำงานจะถูกสร้าง เพื่ออธิบายถึงความสัมพันธ์ระหว่างตัวแปรและระบุถึงตัวแปรอิสระ สิ่งที่จะได้รับในช่วงนี้คือ แผนงานในการสร้างต้นแบบของระบบ และตัวอย่างของข้อมูลที่จะใช้ในการสอนและทดสอบเนตเวิร์ก

2. ขั้นต้นแบบ (Prototype Phase)

ขั้นนี้จะมีการสร้างต้นแบบ แล้วนำไปสอนและทดสอบ ระหว่างการสร้างมีสิ่งที่ควรคำนึงถึงทางด้านเทคนิค ดังนี้

- วิธีการนำข้อมูลนำเข้าที่เหมาะสมกับเนตเวิร์ก
- การเลือกชุดข้อมูลการสอน และชุดข้อมูลทดสอบ
- ขนาดของนิวรอลเนตเวิร์ก จำนวนชั้น และโทโพโลยี (Topology) ของเนตเวิร์ก
- เวลาในการสอนและทดสอบเนตเวิร์ก

เมื่อต้นแบบถูกพัฒนา, สอน, ทดสอบ, ดำเนินงานกับข้อมูลจริง และวิเคราะห์ผลลัพธ์ การคาดการณ์ความแม่นยำของเนตเวิร์ก ค่าใช้จ่ายและผลดีของการใช้จะถูกประเมิน เพื่อตัดสินใจว่าจะดำเนินงานต่อไปหรือไม่ สิ่งที่จะได้รับในขั้นนี้คือ ต้นแบบของงาน, การวิเคราะห์ค่าใช้จ่าย/ ผลดี และแผนการนำระบบไปใช้งาน

3. ขั้นนำไปใช้งาน (Implementation Phase)

จะเป็นขั้นการสร้าง ปรับปรุงระบบ และทดสอบต่อจากขั้นต้นแบบ ในขั้นนี้จะทำงานซ้ำหลายรอบ และเวลาที่ใช้จะไม่แน่นอนขึ้นกับระบบที่จะพัฒนา โปรแกรม

นิวรอลเนตเวิร์กจะใช้เวลาในการติดตั้ง ทั้งการสอนเนตเวิร์กด้วยข้อมูลที่ซับซ้อนก็จะใช้เวลา หลังจากนั้นก็จะใช้เวลาอีกส่วนหนึ่งในการตรวจสอบผล

งานอื่นในขั้นนี้รวมถึงการสอนผู้ใช้, การจัดทำคู่มือการใช้งาน และคู่มือระบบ และการนำไปติดตั้งในสถานที่ใช้งาน สิ่งที่จะได้รับคือ นิวรอลเนตเวิร์กที่สมบูรณ์ ติดตั้งเชื่อมกับระบบงานอื่นๆ และคู่มือการใช้งานของผู้ใช้

4. ขั้นบำรุงรักษา (Maintennance Phase)

การทดสอบ, สอน และ แก้ไขปรับปรุงเนตเวิร์กจะดำเนินต่อ ตามคาบของเวลา บนพื้นฐานของกรณีใหม่ที่เกิดขึ้นของข้อมูล เพื่อให้แน่ใจว่าเนตเวิร์กยังคงทำงานได้อย่างมีประสิทธิภาพ

<u>นิวรอลเนตเวิร์กกับการพยากรณ์อนุกรมเวลา</u>

Ginzberg และ Horn (1991) ได้ทำการทดลองชี้ให้เห็นว่าสามารถสอนนิวรอลเนตเวิร์กให้เรียนรู้อนุกรมเวลา และสามารถใช้นิวรอลเนตเวิร์กพยากรณ์ค่าถัดไปของอนุกรม โดยออกแบบนิวรอลเนตเวิร์กให้มีชั้นนำเข้า 4 นิวรอล ชั้นแอบแฝง 6 นิวรอล และชั้นแสดงผลลัพธ์ 1 นิวรอล ด้วยการใช้ข้อมูลที่สร้างจากสมการ

$$y_n = 4y_{n-1}(1 - y_{n-1})$$

ทำการทดลองเปรียบเทียบการใช้ฟังก์ชันแปลงค่าระหว่างฟังก์ชันซิกมอยด์กับฟังก์ชันพีชคณิต ฟังก์ชันพีชคณิตนี้ประกอบด้วยนิวรอลที่ใช้สมการเส้นตรง 2 นิวรอล นิวรอลที่ใช้สมการควอดราติก (Quadratic) 2 นิวรอล และนิวรอลที่ใช้สมการกำลังสามของผลรวมของค่าน้ำหนักคูณกับข้อมูลนำเข้าที่นิวรอลนั้น 2 นิวรอล จากการทดลองปรากฏว่าผลที่ได้จากการใช้ฟังก์ชันพีชคณิตให้ผลดีมากกว่าการใช้ฟังก์ชันซิกมอยด์ในการพยากรณ์ระยะยาว นั่นคือแสดงว่านิวรอลเนตเวิร์กสามารถค้นพบรูปแบบข้อมูล จากเกณฑ์ที่ว่าค่าความคลาดเคลื่อนจากชุดการสอนและชุดการทดสอบเข้าใกล้ศูนย์

Chan และ Prager (1991) ได้ทำการออกแบบนิวรอลเนตเวิร์กให้เรียนรู้อนุกรมเวลาด้วยการใช้ข้อมูลที่สร้างจากสมการ

$$\text{Tent}: \quad x_{n+1} = 2x_n \ , \ 0 < x_n < 0.5$$

$$x_{n+1} = 2(1-x_n) \ , \ 0.5 < x_n < 1$$

$$\text{Henon}: \quad x_{n+1} = 1 - ax_n^2 + y_n$$

$$y_{n+1} = bx_n$$

เมื่อ a = 1.4 และ b = 0.3 โดยทำการทดลองหาจำนวนนิวรอลของชั้นข้อมูลนำเข้าและชั้นแอบแฝง ด้วยจำนวนนิวรอลของชั้นแสดงผลลัพธ์ 1 นิวรอล และใช้ฟังก์ชันซิกมอยด์เป็นฟังก์ชันแปลงค่า ด้วยชุดการสอน 50 ชุด ผลการทดลองปรากฏว่านิวรอลเนตเวิร์กสามารถค้นพบรูปแบบของสมการทั้ง 2 สมการ

Matsuba (1991) ทำระบบงานการพยากรณ์ราคาหุ้นในระยะยาวโดยใช้ 2 เนตเวิร์ก โดยเนตเวิร์กแรกจะต่อเชื่อมกันอย่างสมบูรณ์ (Fully Connected Network) และ เนตเวิร์กที่สองเป็นแบบฟีดแบกหลายชั้น (Feedback Multilayer) เชื่อมต่อกัน ซึ่งทั้ง 2 เนตเวิร์กใช้หลักการของการเรียนรู้แบบย้อนกลับ โดยข้อมูลนำเข้าของเนตเวิร์กที่สองจะมาจากผลลัพธ์ของเนตเวิร์กแรก โดยใช้จำนวนนิวรอลของชั้นรับข้อมูล, ชั้นแอบแฝง และชั้นแสดงผลลัพธ์ เท่ากับ 20 , 40 และ 10 ตามลำดับ

Refenes (1991) พัฒนาระบบพยากรณ์อัตราการแลกเปลี่ยนเงินตราในระยะยาว โดยใช้การเรียนรู้แบบย้อนกลับ โดยในระหว่างการสอนถ้าเนตเวิร์กมีความผิดพลาดมากจะเพิ่มนิวรอลใหม่เข้าไปในชั้นแอบแฝง แล้วจะทำการสอนเฉพาะนิวรอลนั้นให้แก้ไขข้อผิดพลาดของผลลัพธ์ ข้อดีของวิธีนี้คือ เนตเวิร์กเรียนรู้ได้เร็วขึ้นเพราะสอนเฉพาะนิวรอลใหม่เท่านั้น และแต่ละนิวรอลจะแก้ปัญหาเฉพาะอย่าง วิธีนี้ไม่ต้องเดาขนาดของเนตเวิร์กเพราะจะสร้างเองอัตโนมัติ

Srirengar และ Looi (1991) ได้ทดลองลดเวลาในการสอนเนตเวิร์ก โดยใช้การเรียนรู้แบบย้อนกลับแบบ 2 ชั้นแอบแฝง ซึ่งชั้นแรกจะเชื่อมต่อบางส่วนกับชั้นข้อมูลนำเข้า และชั้นที่สองจะเชื่อมต่อกันอย่างทั่วถึง ในการพยากรณ์เวลาที่จะเกิดจุดดับในดวงอาทิตย์, ยอดขายของห้างสรรพสินค้า และดัชนีทางการเงิน

Miao และ Li (1991) ได้พัฒนานิวรอลเนตเวิร์กเพื่อแก้ปัญหาในการพยากรณ์แบบไม่เป็นเชิงเส้น (Nonlinear) ประกอบด้วย 2 เนตเวิร์ก โดยเนตเวิร์กแรกจะเรียนรู้พฤติกรรมของระบบ (Learning Network) และพยากรณ์โดยเนตเวิร์กที่สอง (Prediction Network) โดยคิดค่าพยากรณ์จากค่าน้ำหนักจากเนตเวิร์กแรก ทั้งสองได้พบว่าวิธีนี้สามารถใช้ได้ กับระบบที่มีตัวแปรที่ไม่ทราบ (Unknown Nonlinear) มากๆ

Weigend Rumelhart และ Huberman (1991) ได้ทดลองกำจัดค่าน้ำหนักที่ไม่จำเป็นของเนตเวิร์กในโมเดลการเรียนรู้แบบย้อนกลับเพื่อการพยากรณ์ผลตอบแทนจากอัตราการแลกเปลี่ยนเงิน โดยใช้ 45 หน่วยรับข้อมูลผลตอบแทนสำหรับเงินเยอรมัน 5 หน่วยรับข้อมูลสำหรับผลตอบแทนของเงินทุกสกุลเฉพาะข้อมูลวันจันทร์ และ 11 หน่วยรับข้อมูลสำหรับแนวโน้ม สำหรับชั้นแอบแฝงมี 5 นิวรอลและใช้ซิกมอยด์ฟังก์ชันในการแปลงค่าและชั้นผลลัพธ์ใช้ 2 นิวรอล โดยนิวรอลแรกจะเป็นค่าพยากรณ์ข้อมูลวันรุ่งขึ้นและปรับข้อผิดผลาดแบบควอดราติก (Quadratic) ส่วนนิวรอลหลังเน้นที่การเปลี่ยนแปลง (บวก,ลบ) และใช้การปรับข้อผิดผลาดโดยซิกมอยด์ฟังก์ชัน