บทที่ 15

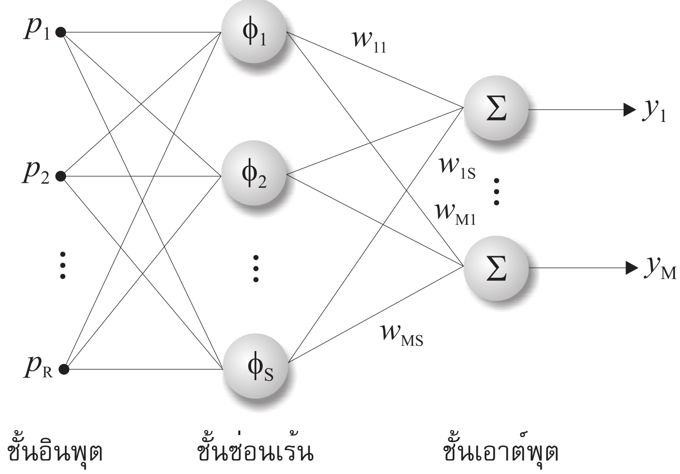
การเรียนรู้ของเครือข่ายฟังก์ชันฐานรัศมี
Learning of Radial Basis Function Network

เครือข่ายฟังก์ชันฐานรัศมี (radial basis function network หรือ RBF network) เป็นเครือข่ายไปข้างหน้าประเภทหนึ่ง ที่ได้รับการยอมรับว่ามีประสิทธิภาพสูงเครือข่ายหนึ่ง เครือข่าย RBF แตกต่างไปจากเครือข่ายเพอร์เซ็พตรอนแบบหลายชั้น (multi-layer perceptron) ตรงที่เครือข่าย RBF นั้นมีชั้นซ่อนเร้นเพียงชั้นเดียว David Broomhead และ David Lowe [Broomhead and Lowe, 1988] ถือเป็นผู้บุกเบิกนำเอา RBF มาประยุกต์ใช้ เครือข่ายฟังก์ชันฐานรัศมีสามารถพิจารณาเป็นฟังก์ชันการส่ง (mapping function) ของความสัมพันธ์ระหว่างคู่รูปแบบอินพุตและเอาต์พุตได้ โดยการเรียนรู้ของเครือข่ายเป็นการปรับค่าน้ำหนักประสาทให้ได้ฟังก์ชันการส่งที่เหมาะที่สุด ดังนั้นเราสามารถกล่าวได้ว่าเครือข่าย RBF คือกระบวนการปรับเส้นโค้ง (curve fitting) ระหว่างข้อมูลอินพุตกับเอาต์พุตนั่นเอง

15.1 สถาปัตยกรรมของเครือข่าย RBF

รูปที่ 15.1 แสดงเครือข่าย RBF ทั่วๆ ไป ซึ่งประกอบไปด้วยชั้นของนิวรอน 3 ชั้นดังนี้

- **ชั้นอินพุต** - แต่ละอินพุตจะแทนคุณลักษณะของเวกเตอร์อินพุต เหมือนกับในเครือข่ายเพอร์เซ็พตรอนแบบหลายชั้นทั่วๆ ไป ในที่นี้เวกเตอร์อินพุตมีขนาดเท่ากับ R
- **ชั้นซ่อนเร้น** - แต่ละนิวรอนในชั้นซ่อนเร้นจะมีฟังก์ชันถ่ายโอนซึ่งมีลักษณะพิเศษ ที่ซึ่งให้ผลตอบสนองของฟังก์ชันที่ขึ้นอยู่กับระยะห่างระหว่างอินพุตกับจุดศูนย์กลางของฟังก์ชัน กล่าวคือถ้าเวกเตอร์อินพุตอยู่ใกล้จุดศูนย์กลางมาก เอาต์พุตที่ได้จะมาก ถ้าเวกเตอร์อินพุตอยู่ห่างออกจากจุดศูนย์กลาง เอาต์พุตที่ได้จะลดลงตามลำดับ ในที่นี้จำนวนนิวรอนในชั้นซ่อนเร้นมีขนาดเท่ากับ S
- **ชั้นเอาต์พุต** - มีหน้าที่รวมเอาต์พุตที่ได้จากแต่ละนิวรอนในชั้นซ่อนเร้น เครือข่ายให้เอาต์พุตในรูปของเวกเตอร์ขนาดเท่ากับ M



รูปที่ 15.1: เครือข่าย RBF ที่มีขนาดของอินพุตเท่ากับ R จำนวนนิวรอนในชั้นซ่อนเร้นเท่ากับ S และขนาดของเอาต์พุตเท่ากับ M

ดังนั้นเราสามารถพิจารณาเครือข่าย RBF เป็นการฟังก์ชันการส่งระหว่างปริภูมิของอินพุต $\mathbf{p} \in \mathbb{R}^{R\times 1}$ ไปยังปริภูมิของเอาต์พุต $\mathbf{y} \in \mathbb{R}^{M\times 1}$ ได้ จากเครือข่าย RBF ในรูปข้างต้น จะได้ว่าเอาต์พุตตัวที่ i ของเครือข่ายมีค่าเท่ากับ

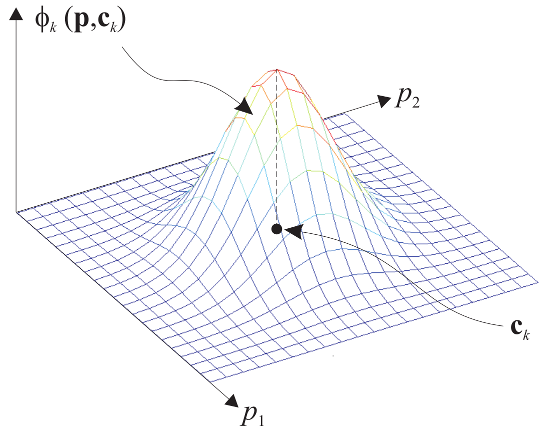
$$y_i = \sum_{k=1}^{S} w_{ik}\phi_k(\mathbf{p}, \mathbf{c}_k) \tag{15.1}$$

$$= \sum_{k=1}^{S} w_{ik}\phi_k(\|\mathbf{p} - \mathbf{c}_k\|_2) \tag{15.2}$$

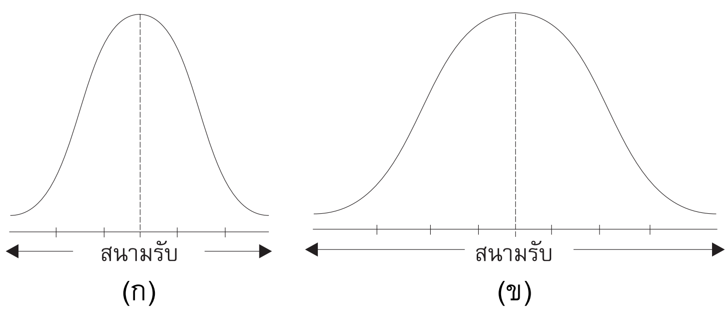
โดยที่ $\phi_k(\cdot)$ = ฟังก์ชันส่งค่าจาก $\mathbb{R}^+$ ไปยัง $\mathbb{R}$ ของนิวรอนตัวที่ k ในชั้นซ่อนเร้น
$\|\cdot\|_2$ = ฟังก์ชันระยะทางแบบยุคลิด
$\mathbf{w}_{ik}$ = ค่าน้ำหนักประสาทของนิวรอนตัวที่ k ในชั้นซ่อนเร้น
S = จำนวนนิวรอนทั้งหมดในชั้นซ่อนเร้น
$\mathbf{c}_k \in \mathbb{R}^{R\times 1}$ = เวกเตอร์จุดศูนย์กลางในปริภูมิของเวกเตอร์อินพุตสำหรับนิวรอนตัวที่ k ในชั้นซ่อนเร้น

สำหรับแต่ละนิวรอนในชั้นซ่อนเร้น ค่าระยะทางยุคลิดระหว่างเวกเตอร์จุดศูนย์กลาง $\mathbf{c}_k$ กับเวกเตอร์อินพุต $\mathbf{p}$ จะถูกคำนวณ เอาต์พุตของนิวรอนในชั้นซ่อนเร้นนี้จะได้จากฟังก์ชัน $\phi_k(\cdot)$ ซึ่งเป็นฟังก์ชันแบบไม่เป็นเชิงเส้น สุดท้ายแล้วเอาต์พุตของเครือข่ายจะได้จากผลรวมของค่าน้ำหนักประสาท กับเอาต์พุตของนิวรอนจากชั้นซ่อนเร้น ตัวอย่างของฟังก์ชัน $\phi_k(\cdot)$ ที่ใช้ในเครือข่าย RBF เช่น

- ฟังก์ชันเชิงเส้น $\phi(p) = p$
- ฟังก์ชันประมาณกำลังสาม $\phi(p) = p^3$
- ฟังก์ชัน thin-plate-spline $\phi(p) = p^2 \ln p$
- ฟังก์ชันเกาส์เซียน $\phi(p) = e^{-p^2/\sigma^2}$
- ฟังก์ชันรากกำลังสอง $\phi(p) = \sqrt{p^2 + \sigma^2}$
- ฟังก์ชันรากกำลังสองผกผัน $\phi(p) = \frac{1}{\sqrt{p^2+\sigma^2}}$



รูปที่ 15.2: ฟังก์ชันฐานรัศมีแบบเกาส์เซียน (แสดงตัวอย่างใน 2 มิติสำหรับปริภูมิของอินพุตเวกเตอร์ $\mathbf{p} = [p_1 \ \ p_2]^T$)



รูปที่ 15.3: ผลของพารามิเตอร์การกระจาย σ กับความกว้างของ RBF

การปรากฏของฟังก์ชัน ϕ ดังกล่าวเป็นที่มาของชื่อ Radial Basis Function หรือ RBF นั่นเอง ฟังก์ชันที่นิยมใช้ใน RBF มากที่สุดก็คือฟังก์ชันเกาส์เซียน รูปร่างของฟังก์ชันเกาส์เซียนมีแสดงในรูปที่ 15.2 โดยที่พารามิเตอร์ σ เป็นตัวควบคุมความกว้างของ RBF หรือเรียกว่าพารามิเตอร์การกระจาย (spread parameter)

เวกเตอร์ศูนย์กลาง $\mathbf{c}_k$ ของนิวรอนซ่อนเร้นตัวที่ k จะรับอินพุตจากเวกเตอร์ $\mathbf{p}$ ที่มีมิติเท่ากับ R พารามิเตอร์ σ_k ทำหน้าที่ควบคุมความกว้างของแต่ละ RBF โดยปกติแล้ว ถ้าเวกเตอร์อินพุต $\mathbf{p}$ มีระยะห่างจากจุดศูนย์กลาง $\mathbf{c}_k$ มากขึ้น กล่าวคือ $\|\mathbf{p} - \mathbf{c}_k\|2$ มีค่ามากขึ้น ค่าที่ได้จากฟังก์ชัน ϕ_k จะลดลง (ดูรูปที่ 15.3) พื้นที่ของฟังก์ชัน ϕ เรียกว่าสนามรับ (receptive field) ของนิวรอนนั้นๆ [Wasserman, 1993] เอาต์พุต y_j ของเครือข่ายได้จากผลรวมของเอาต์พุตของฟังก์ชัน ϕ จากทุกนิวรอนในชั้นซ่อนเร้น โดยปกติแล้วเวกเตอร์จุดศูนย์กลาง $\mathbf{c}_k$ จะถูกเลือกจากปริภูมิของเวกเตอร์อินพุต ที่ซึ่งจะต้องมีจำนวนเวกเตอร์จุดศูนย์กลาง (หรือจำนวนนิวรอนในชั้นซ่อนเร้น) เพียงพอและครอบคลุมปริภูมิของอินพุตได้ รายละเอียดพารามิเตอร์ต่างๆ ของเครือข่าย RBF จะได้กล่าวถึงในหัวข้อต่อไป

15.2 การฝึกสอนเครือข่าย RBF

การฝึกสอนเครือข่าย RBF นั้นได้มีผู้นำเสนอไว้หลายรูปแบบ ในลักษณะเดียวกันกับการฝึกสอนเครือข่ายประสาทเทียมแบบอื่นๆ การฝึกสอนเครือข่าย RBF ก็คือการค้นหาพารามิเตอร์ของเครือข่ายซึ่งประกอบไปด้วย

- ค่าน้ำหนักประสาท w_{ik} สำหรับ $i = 1, \ldots, M$ และ $k = 1, \ldots, S$
- เวกเตอร์จุดศูนย์กลางของนิวรอนตัวที่ k ในชั้นซ่อนเร้น $\mathbf{c}_k$ สำหรับ $k = 1, \ldots, S$

- ค่าพารามิเตอร์การกระจาย σ_k สำหรับ $k = 1, \ldots, S$

ในที่นี้จะได้นำเสนอวิธีการฝึกสอนแบบจุดศูนย์กลางคงที่ (fixed center) และแบบเกรเดียนต์เฟ้นสุ่ม (stochastic gradient) ซึ่งในการฝึกสอนแบบแรกเวกเตอร์จุดศูนย์กลางจะถูกสุ่มมาจากเวกเตอร์อินพุต และจะไม่มีการเปลี่ยนแปลงตำแหน่งของจุดศูนย์กลางในระหว่างการฝึกสอน ในขณะที่การฝึกสอนแบบที่สองจะมีการปรับพารามิเตอร์ของเครือข่ายทั้งหมด รายละเอียดการฝึกสอนทั้งสองแบบมีดังต่อไปนี้ [Ham and Kostanic, 2001]

15.2.1 การฝึกสอนเครือข่าย RBF แบบจุดศูนย์กลางคงที่

พิจารณาความสัมพันธ์เอาต์พุตตัวที่ i ของเครือข่าย RBF ต่อไปนี้

$$y_i = \sum_{k=1}^{S} w_{ik}\phi_k(\mathbf{p}, \mathbf{c}_k) \tag{15.3}$$

$$= \sum_{k=1}^{S} w_{ik}\phi_k(\|\mathbf{p} - \mathbf{c}_k\|_2) \tag{15.4}$$

จะเห็นได้ว่าพารามิเตอร์ที่ควบคุมการส่งค่าระหว่างอินพุตกับเอาต์พุตของเครือข่ายก็คือค่าน้ำหนักประสาท w_{ik} ในชั้นเอาต์พุตและเวกเตอร์จุดศูนย์กลาง $\mathbf{c}_k$ ของ RBF (ในที่นี้คือฟังก์ชันเกาส์เซียน) ดังนั้นการฝึกสอนเครือข่าย RBF ที่ง่ายที่สุดก็คือกำหนดให้เวกเตอร์จุดศูนย์กลางมีค่าคงที่ โดยปกติแล้วในขั้นตอนการฝึกสอนจะทำการสุ่มเลือกเวกเตอร์จุดศูนย์กลางจากเวกเตอร์อินพุต [Broomhead and Lowe, 1988] สิ่งสำคัญอย่างหนึ่งในการฝึกสอนแบบนี้ก็คือ จำนวนเวกเตอร์จุดศูนย์กลางที่สุ่มเลือกมาจะต้องมี*จำนวนเพียงพอ*ที่จะครอบคลุมปริภูมิของอินพุต ที่ซึ่งไม่มีวิธีการที่แน่นอนในการหาว่าจำนวนของเวกเตอร์ดังกล่าวควรจะมีค่าเป็นเท่าไร หลักการอย่างหนึ่งก็คือเลือกเวกเตอร์จุดศูนย์กลางให้มีจำนวนมากพอที่จะครอบคลุมปริภูมิของอินพุต แล้วในขณะฝึกสอนเราสามารถกำจัดเวกเตอร์จุดศูนย์กลาง (นั่นก็คือนิวรอนในชั้นซ่อนเร้น) ออกจากเครือข่าย โดยที่ไม่ทำให้เครือข่ายลดประสิทธิภาพลงแต่อย่างใด รายละเอียดการฝึกสอนเครือข่ายหลังจากที่กำหนดจำนวนเวกเตอร์จุดศูนย์กลาง (นั่นคือจำนวนนิวรอนในชั้นซ่อนเร้น) แล้วมีดังต่อไปนี้

▷ **อัลกอริทึมการฝึกสอนเครือข่าย RBF แบบจุดศูนย์กลางคงที่**

1. กำหนดให้จำนวนคู่เวกเตอร์อินพุต/เอาต์พุตมีทั้งหมด Q คู่ จะได้ว่าเอาต์พุต $\tilde{y}_q$ ของนิวรอนตัวที่ q ของเครือข่ายคือ

$$\tilde{y}_q = \sum_{k=1}^{S} w_{ik}\phi(\mathbf{p}_q, \mathbf{c}_k),\ q = 1, \ldots, Q \tag{15.5}$$

   สำหรับเวกเตอร์อินพุต $\mathbf{p}_q$ ชุดที่ q และเวกเตอร์จุดศูนย์กลาง $\mathbf{c}_k$ ของนิวรอนตัวที่ k ในชั้นซ่อนเร้น

2. เขียนให้อยู่ในรูปเมตริกซ์ได้ดังนี้

$$\begin{bmatrix} \tilde{y}_1 \\ \vdots \\ \tilde{y}_Q \end{bmatrix} = \begin{bmatrix} \phi(\mathbf{p}_1, \mathbf{c}_1) & \cdots & \phi(\mathbf{p}_1, \mathbf{c}_S) \\ \vdots & \ddots & \vdots \\ \phi(\mathbf{p}_Q, \mathbf{c}_1) & \cdots & \phi(\mathbf{p}_Q, \mathbf{c}_S) \end{bmatrix} \begin{bmatrix} w_1 \\ \vdots \\ w_S \end{bmatrix} \tag{15.6}$$

   หรือ

$$\tilde{\mathbf{y}} = \Phi\mathbf{w} \tag{15.7}$$

   โดยที่ $\tilde{\mathbf{y}} \in \mathbb{R}^{Q\times 1}$ = เอาต์พุตของเครือข่าย
   $\mathbf{w} \in \mathbb{R}^{S\times 1}$ = เวกเตอร์น้ำหนักประสาทในชั้นซ่อนเร้น
   $\Phi \in \mathbb{R}^{Q\times S}$ = เมตริกซ์ของ RBF ในชั้นซ่อนเร้น

3. เนื่องจากจุดศูนย์กลางของ RBF ถูกกำหนดให้คงที่ ดังนั้นการฝึกสอนจะทำการคำนวณหาเพียงค่าของน้ำหนักประสาท โดยใช้ค่าวัตถุประสงค์เป็นค่าความผิดพลาดเฉลี่ยกำลังสอง (MSE หรือ Mean-Squared Error) ระหว่างเอาต์พุตของเครือข่าย $\tilde{\mathbf{y}}$ กับข้อมูลเอาต์พุตจริง $\hat{\mathbf{y}}$ ดังนั้นฟังก์ชันวัตถุประสงค์สำหรับฝึกสอนเครือข่ายคือ

$$J(\mathbf{w}) = \frac{1}{2}\sum_{q=1}^{Q}\left[\hat{y}_q - \tilde{y}_q\right]^2 \quad (15.8)$$

$$= \frac{1}{2}\left(\hat{\mathbf{y}} - \tilde{\mathbf{y}}\right)^T\left(\hat{\mathbf{y}} - \tilde{\mathbf{y}}\right) \quad (15.9)$$

โดยที่ $\hat{\mathbf{y}} \in \mathbb{R}^{Q\times 1}$ คือเวกเตอร์ของเอาต์พุตที่ต้องการ (จากคู่อินพุต/เอาต์พุต)

4. แทนสมการที่ 15.7 ลงในสมการที่ 15.9 จะได้

$$J(\mathbf{w}) = \frac{1}{2}\left(\hat{\mathbf{y}} - \Phi\mathbf{w}\right)^T\left(\hat{\mathbf{y}} - \Phi\mathbf{w}\right) \quad (15.10)$$

5. ทำการอนุพันธ์เพื่อหาค่าน้อยที่สุดของ $J(\mathbf{w})$

$$\frac{\partial J(\mathbf{w})}{\partial \mathbf{w}} = 0 \quad (15.11)$$

จะได้

$$-\Phi^T\hat{\mathbf{y}} + \Phi^T\Phi\mathbf{w} = 0 \quad (15.12)$$

แก้สมการข้างต้นด้วยเมตริกซ์ผกผันเทียม จะได้ค่าน้ำหนักประสาทของเครือข่ายจากค่าความผิดพลาดที่น้อยที่สุดคือ

$$\mathbf{w} = (\Phi^T\Phi)^{-1}\Phi^T\hat{\mathbf{y}} \quad (15.13)$$

$$= \Phi^{+}\hat{\mathbf{y}} \quad (15.14)$$

โดยที่ Φ^{+} คือเมตริกซ์ผกผันเทียมของฟังก์ชัน Φ $\triangleleft$

จะเห็นได้ว่า ในกรณีที่กำหนดจุดศูนย์กลาง RBF ให้คงที่ การฝึกสอนเครือข่ายจะให้ผลลัพธ์เป็นผลเฉลยรูปแบบปิด ดังนั้นการฝึกสอนจะสามารถทำได้อย่างรวดเร็ว จึงทำให้มีผู้สนใจนำเอาเครือข่าย RBF ไปใช้งานอย่างมากมาย นอกไปจากนั้นแล้ว ขนาดของเครือข่าย RBF ยังมีผลจากสมการที่ 15.12 ที่ซึ่งจะทำให้สามารถหาคำตอบของสมการได้แบบหนึ่งคำตอบอย่างเป็นเอกลักษณ์ แบบหาได้ขาด (underdetermined) หรือแบบหาได้เกิน (overdetermined) กล่าวคือถ้าจุดศูนย์กลาง (หรือนิวรอนในชั้นซ่อนเร้น) มีจำนวนมากกว่าหรือเท่ากับจำนวนของคู่ตัวอย่างอินพุต/เอาต์พุตสำหรับฝึกสอน ค่าความผิดพลาดที่ได้จากการฝึกสอนเครือข่ายจะมีค่าน้อย โดยเฉพาะถ้าใช้สมการที่ 15.14 แล้ว ค่าความผิดพลาดจะเป็นศูนย์

การตั้งค่าพารามิเตอร์การกระจาย

ในกรณีที่ใช้ฟังก์ชันเกาส์เซียนเป็น RBF พารามิเตอร์ที่สำคัญอย่างหนึ่งก็คือพารามิเตอร์การกระจาย σ ซึ่งโดยปกติแล้วจะกำหนดด้วยความสัมพันธ์ต่อไปนี้

$$\sigma = \frac{d_{max}}{\sqrt{K}} \quad (15.15)$$

โดยที่ d_{max} คือค่าระยะทางยุคลิดที่มากที่สุดระหว่างจุดศูนย์กลางที่กำลังพิจารณาและ K คือจำนวนของจุดศูนย์กลาง ดังนั้นจะได้ RBF ของนิวรอนในชั้นซ่อนเร้นคือ

$$\phi(\mathbf{p}, \mathbf{c}_k) = e^{-\frac{K}{d_{max}^2}\|\mathbf{p}-\mathbf{c}_k\|^2} \quad (15.16)$$

15.2.2 การฝึกสอนเครือข่าย RBF แบบเกรเดียนต์เฟ้นสุ่ม

การฝึกสอนแบบจุดศูนย์กลางคงที่มีขั้นตอนที่ง่าย แต่มีข้อจำกัดที่จำนวนของนิวรอนหรือจุดศูนย์กลางจะต้องมากเพียงพอ เป็นผลให้เครือข่าย RBF อาจมีขนาดใหญ่เกินไปได้ แม้จะใช้กับงานที่ไม่ซับซ้อนก็ตาม การฝึกสอนแบบเกรเดียนต์เฟ้นสุ่ม (stochastic gradient) เป็นการฝึกสอนที่ทำการปรับพารามิเตอร์ทั้งหมดของเครือข่าย ซึ่งได้แก่ น้ำหนักประสาท จุดศูนย์กลางนิวรอนและความกว้างของ RBF ทำให้การปรับเส้นโค้งระหว่างอินพุตกับเอาต์พุตมีความยืดหยุ่นมากยิ่งขึ้น และนำไปสู่ขนาดของเครือข่ายที่เหมาะสมกับปัญหา รายละเอียดขั้นตอนการฝึกสอนแบบเกรเดียนต์เฟ้นสุ่มสามารถสรุปได้ดังนี้ (t คือหน่วยเวลา)

▷ **อัลกอริทึมการฝึกสอนเครือข่าย RBF แบบเกรเดียนต์เฟ้นสุ่ม**

1. พิจารณาฟังก์ชันวัตถุประสงค์ของเครือข่ายต่อไปนี้

$$J(t) = \frac{1}{2}|e(t)|^2 \quad (15.17)$$

$$= \frac{1}{2}|\hat{y}(t) - \tilde{y}(t)|^2 \quad (15.18)$$

$$= \frac{1}{2}\left[\hat{y}(t) - \sum_{k=1}^{S} w_k(t)\phi\left(\mathbf{p}(t), \mathbf{c}_k(t)\right)\right]^2 \quad (15.19)$$

โดยที่ ϕ คือ RBF และ $\hat{y}$ เป็นค่าเอาต์พุตที่ต้องการให้เครือข่ายเรียนรู้ $e(t) = \tilde{y}(t) - \hat{y}(t)$ เป็นค่าความแตกต่างระหว่างเอาต์พุตที่ต้องการให้เครือข่ายเรียนรู้กับเอาต์พุตที่ได้จริงจากเครือข่าย ส่วนเทอม $\tilde{y}(t) = \sum_{k=1}^{S} w_k(t)\phi\left(\mathbf{p}(t), \mathbf{c}_k(t)\right)$ เป็นเอาต์พุตจริงที่ได้จากเครือข่าย ณ เวลารอบการเรียนรู้ t

2. ถ้าเลือกใช้ฟังก์ชันเกาส์เซียนเป็น RBF จะได้

$$J(t) = \frac{1}{2}\left[\hat{y}(t) - \sum_{k=1}^{S} w_k(t) e^{-\frac{\|\mathbf{p}(t)-\mathbf{c}_k(t)\|_2^2}{\sigma_k^2(t)}}\right]^2 \quad (15.20)$$

3. ทำการปรับพารามิเตอร์ต่างๆ ของเครือข่ายดังต่อไปนี้

   - <u>น้ำหนักประสาท</u>

$$\mathbf{w}(t+1) = \mathbf{w}(t) - \mu_w \frac{\partial}{\partial \mathbf{w}} J(t)\bigg|_{\mathbf{w}=\mathbf{w}(t)} \quad (15.21)$$

$$= \mathbf{w}(t) + \mu_w e(t)\Psi(t) \quad (15.22)$$

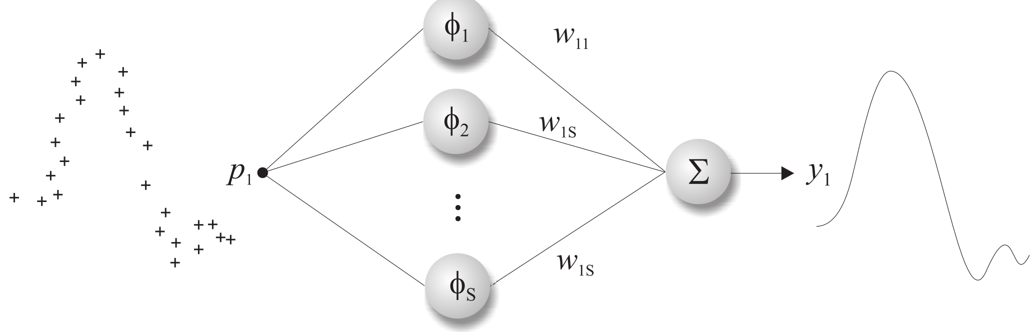
$$(15.23)$$

   - <u>จุดศูนย์กลาง</u>

$$\mathbf{c}_k(t+1) = \mathbf{c}_k(t) - \mu_c \frac{\partial}{\partial \mathbf{c}_k} J(t)\bigg|_{\mathbf{c}_k=\mathbf{c}_k(t)} \quad (15.24)$$

$$= \mathbf{c}_k(t) + \mu_c \frac{e(t)w_k(t)}{\sigma_k^2(t)}\phi\left(\mathbf{p}(t), \mathbf{c}_k(t), \sigma_k\right)\left[\mathbf{p}(t) - \mathbf{c}_k(t)\right] \quad (15.25)$$

$$(15.26)$$



รูปที่ 15.4: โครงสร้างเครือข่าย RBF ทั่วไปสำหรับการประมาณค่าฟังก์ชัน 1 มิติ

- พารามิเตอร์การกระจาย

$$\sigma_k(t+1) = \sigma_k(t) - \mu_\sigma \frac{\partial}{\partial \sigma_k} J(t)\bigg|_{\sigma_k=\sigma_k(t)} \quad (15.27)$$

$$= \sigma_k(t) + \mu_\sigma \frac{e(t)w_k(t)}{\sigma_k^3(t)} \phi(\mathbf{p}(t), \mathbf{c}_k(t), \sigma_k) \|\mathbf{p}(t) - \mathbf{c}_k(t)\|^2 \quad (15.28)$$

โดยที่ $\Psi(t) = [\phi(\mathbf{p}(t), \mathbf{c}_1, \sigma_1) \ldots \phi(\mathbf{p}(t), \mathbf{c}_S, \sigma_S)]^T$ และ $e(t) = \tilde{y}(t) - \hat{y}(t)$ เป็นค่าความผิดพลาดหรือค่าความแตกต่าง ระหว่างเอาต์พุตที่ต้องการกับเอาต์พุตที่ได้จริงของเครือข่าย โดยที่ $\hat{y}(t)$ เป็นค่าเอาต์พุตที่ต้องการ (จากคู่เวกเตอร์อินพุต/เอาต์พุตสำหรับฝึกสอน) และ μ_w μ_c และ μ_σ เป็นค่าคงที่การเรียนรู้สำหรับปรับค่าน้ำหนักประสาท ค่าเวกเตอร์จุดศูนย์กลางและค่าพารามิเตอร์การกระจายตามลำดับ (นั่นคือสำหรับพารามิเตอร์แต่ละตัว ไม่จำเป็นจะต้องมีอัตราการเรียนรู้ที่เท่ากัน)

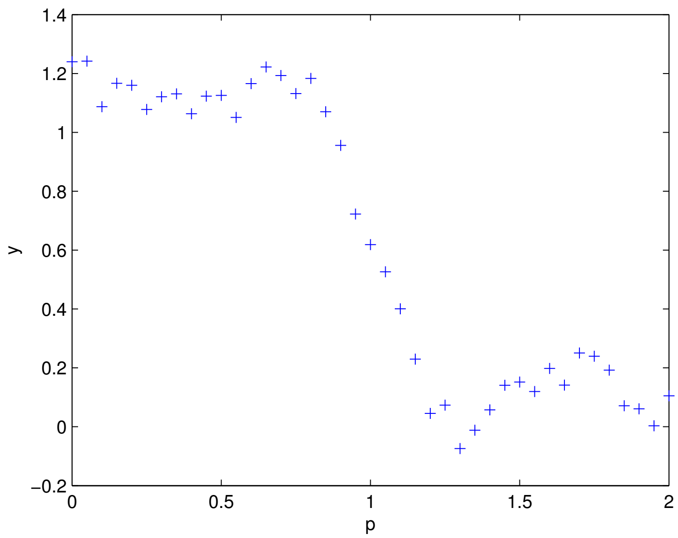
4. วนรอบการฝึกสอนจนกระทั่งเครือข่ายลู่เข้าสู่ค่าความผิดพลาด $e(t)$ ที่ยอมรับได้ ◁

ความสามารถในการปรับค่าจุดศูนย์กลางและพารามิเตอร์การกระจายของนิวรอนในชั้นซ่อนเร้น เป็นการเพิ่มประสิทธิภาพของเครือข่ายขึ้นอย่างชัดเจน ถ้าเปรียบเทียบเครือข่ายขนาดเดียวกันแล้ว เครือข่ายที่ฝึกสอนด้วยวิธีเกรเดียนต์เฟ้นสุ่มจะมีประสิทธิภาพเหนือกว่าที่ได้จากการฝึกสอนแบบจุดศูนย์กลางคงที่ แต่การฝึกสอนแบบเกรเดียนต์เฟ้นสุ่มจะมีความยุ่งยากซับซ้อนมากกว่ามาก นั่นคือเครือข่ายจะต้องใช้เวลาในการประมวลผลมากกว่า การฝึกสอนเครือข่าย RBF แบบอื่นๆ นั้นได้มีผู้นำเสนอไว้หลากหลาย ผู้อ่านที่สนใจสามารถศึกษาเพิ่มเติมจากตำราเครือข่ายประสาทเทียมที่เกี่ยวข้องได้

■ **ตัวอย่างที่ 15.1** การประมาณค่าฟังก์ชันของเครือข่าย RBF

ตัวอย่างนี้จะกล่าวถึงการทำงานและการฝึกสอนเครือข่าย RBF ด้วยการพิจารณาเครือข่าย RBF ให้เป็นฟังก์ชันการประมาณ ซึ่งเป็นคุณสมบัติเด่นอย่างหนึ่งของเครือข่ายประสาทเทียมหลายๆ แบบ โดยเฉพาะเครือข่าย RBF โครงสร้างเครือข่าย RBF สำหรับเป็นฟังก์ชันประมาณ 1 มิติแสดงในรูปที่ 15.4 เครือข่ายมีอินพุตและเอาต์พุตขนาด 1×1 จำลองสถานการณ์การทำงานด้วยโปรแกรม MATLAB พิจารณาข้อมูลคู่อินพุต/เอาต์พุตในรูปที่ 15.5 จำนวน 41 คู่ ซึ่งได้มาจากฟังก์ชันสี่เหลี่ยมที่ถูกประมาณด้วยอนุกรมฟูริเยร์จำนวน 3 เทอม ดังนี้

$$y = 0.5 + \frac{2}{\pi}(\cos\frac{\pi p}{2} - \cos\frac{3\pi p}{2} + \cos\frac{5\pi p}{2}) \quad (15.29)$$



รูปที่ 15.5: ค่าชักตัวอย่างฟังก์ชันสี่เหลี่ยมจากอนุกรมฟูริเยร์จำนวน 3 เทอม

ทำการสร้างเครือข่าย RBF ด้วยคำสั่ง

```
[net,tr] = newrb(P,T,GOAL,SPREAD,MN,DF)
```

โดยที่

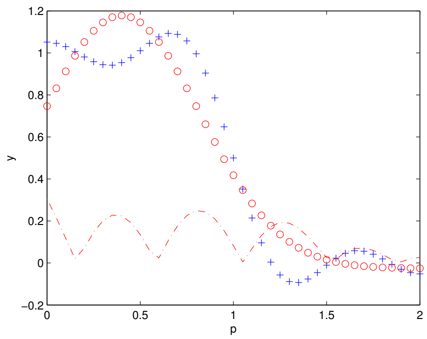
`P` = เมตริกซ์ขนาด $R \times Q$ ของเวกเตอร์อินพุตขนาด R จำนวน Q เวกเตอร์
`T` = เมตริกซ์ขนาด $S \times Q$ ของเวกเตอร์เอาต์พุตขนาด S จำนวน Q เวกเตอร์
`GOAL` = ค่าความผิดพลาดกำลังสองเฉลี่ย ค่าปกติ $= 0.0$
`SPREAD` = ค่าพารามิเตอร์การกระจายของ RBF ค่าปกติ $= 1.0$
`MN` = จำนวนนิวรอนสูงสุด ค่าปกติ $= Q$
`DF` = จำนวนนิวรอนที่จะเพิ่มในเครือข่ายระหว่างการแสดงผลการฝึกสอน ค่าปกติ $= 25$

โดยปกติแล้ว ฟังก์ชัน `newrb` ในกล่องเครื่องมือของ MATLAB จะเริ่มสร้างเครือข่ายจากจำนวนนิวรอนในชั้นซ่อนเริ่มเป็น 1 แล้วทำการเพิ่มจำนวนนิวรอนจนกระทั่งได้ค่าความผิดพลาดของเครือข่ายตามที่กำหนด หรือสูงสุดเท่ากับจำนวนเวกเตอร์อินพุต Q ที่ป้อนให้กับเครือข่าย ดังนั้นเราสามารถเลือกจำนวนนิวรอนสำหรับฝึกสอนได้รวมทั้งค่าพารามิเตอร์การกระจาย ซึ่งทั้งสองพารามิเตอร์มีผลต่อประสิทธิภาพของเครือข่าย RBF โดยตรง

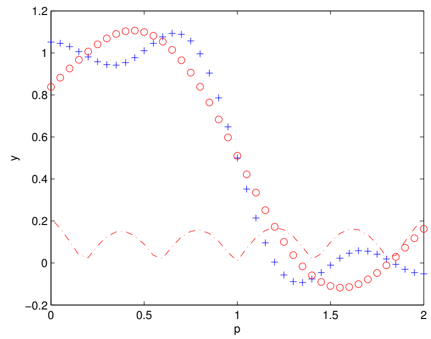
รูปที่ 15.6 แสดงผลการฝึกสอนพร้อมทั้งทดสอบผลการประมาณค่าฟังก์ชัน จากเครือข่ายที่มีพารามิเตอร์ต่างๆ ค่าอินพุตที่ใช้ในการทดสอบยังคงเป็นชุดเดียวกับที่ใช้ในการฝึกสอน เราจะทำการทดสอบด้วยชุดอินพุตนอกเหนือไปจากชุดข้อมูลที่ใช้ฝึกสอน เพื่อทดสอบการทำให้เป็นทั่วไป (generalization) ของเครือข่ายในภายหลัง ในที่นี้เรากำหนดให้เป้าหมายค่าความผิดพลาดของเครือข่าย `GOAL=0.02` และค่าพารามิเตอร์การกระจาย `SPREAD=0.5`

เราจะเห็นได้ว่า ผลการฝึกสอนได้ค่าความผิดพลาดแบบผลรวมกำลังสอง (sum-squared error หรือ SSE) ต่ำสุดเท่ากับ 0.2623 ที่จำนวนนิวรอนเท่ากับ 7 นิวรอน ไม่ว่าจะทำการเพิ่มจำนวนนิวรอนขึ้นจนกระทั่งเท่ากับจำนวนของคู่อินพุต/เอาต์พุต (นั่นคือ 41) ค่า SSE ของเครือข่ายก็ยังคงเดิม ดังนั้นเราจะทำการทดลองปรับค่าพารามิเตอร์การกระจาย σ เพื่อให้เครือข่ายลู่เข้าสู่สภาวะที่ดีขึ้น ผลการทดลองแสดงในรูปที่ 15.7 จะเห็นได้ชัดเจนว่าเมื่อทำการปรับพารามิเตอร์การกระจายที่เหมาะสม เครือข่ายสามารถลู่เข้าสู่ค่า SSE ที่ดีขึ้นได้ (0.1822) ถึงแม้ว่าจะใช้จำนวนนิวรอนเพียง 8 ตัว

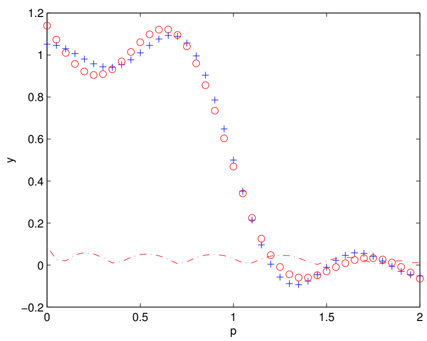
ในกล่องเครื่องมือของ MATLAB มีฟังก์ชัน `newrbe` ที่ใช้ในการสร้างเครือข่าย RBF พร้อมกับฝึกสอนเพื่อให้ได้โครงสร้างของเครือข่ายเป็นไปตามข้อกำหนด (ตัวย่อ e คือ exact) เครือข่ายที่ได้จากฟังก์ชันดังกล่าวจะมีการปรับพารามิเตอร์ทุกๆ ส่วน เพื่อให้ได้ค่าความผิดพลาดของเครือข่ายที่น้อยที่สุด รูปที่ 15.8 แสดงตัวอย่างการใช้



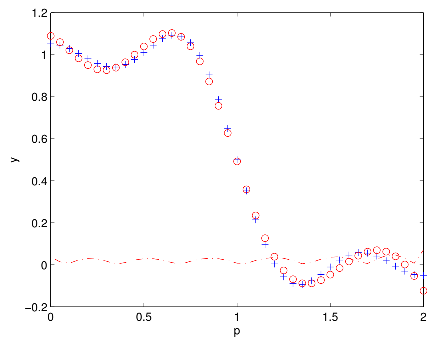
(ก) 1 นิวรอน SSE = 4.9220



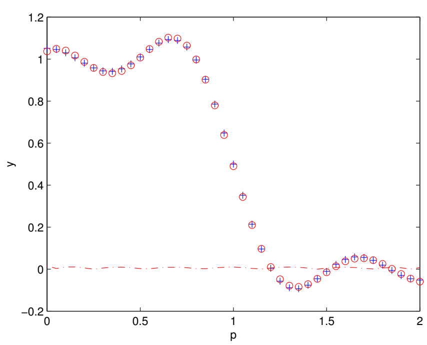
(ข) 3 นิวรอน SSE = 4.4087



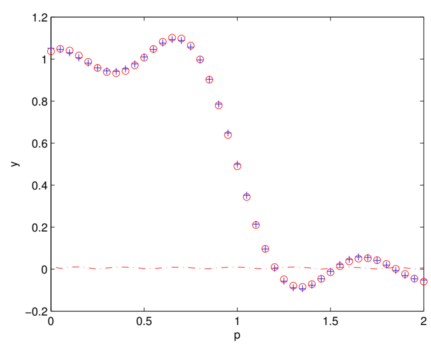
(ค) 5 นิวรอน SSE = 1.2686



(ง) 6 นิวรอน SSE = 0.9697



(จ) 7 นิวรอน SSE = 0.2623

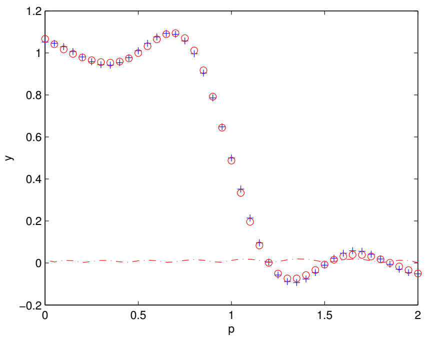


(ฉ) 41 นิวรอน SSE = 0.2623

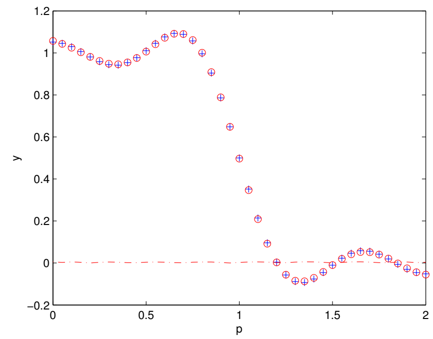
รูปที่ 15.6: ผลการประมาณค่าฟังก์ชันด้วยเครือข่าย RBF ด้วยจำนวนนิวรอนต่างๆ ($\sigma = 0.5$) สัญลักษณ์ '+' แทนข้อมูลอินพุต/เอาต์พุตที่ใช้ฝึกสอน สัญลักษณ์ 'o' แทนเอาต์พุตที่ได้จากเครือข่ายที่ผ่านการฝึกสอนแล้ว และเส้นประด้านล่างของกราฟแสดงผลรวมค่ากำลังสองของความผิดพลาด (SSE) ระหว่างเอาต์พุตจากข้อมูลจริงและเอาต์พุตที่ได้จากเครือข่าย

ฟังก์ชัน `newrbe` ในการสร้างเครือข่าย RBF สำหรับประมาณค่าฟังก์ชัน ผลลัพธ์ที่ได้เป็นเครือข่ายที่มีจำนวนนิวรอนเท่ากับ 41 (เท่ากับจำนวนข้อมูลอินพุต) ค่า SSE ที่ได้มีค่าน้อยมาก แสดงถึงประสิทธิภาพในการประมาณค่าฟังก์ชันที่ดีขึ้นด้วย

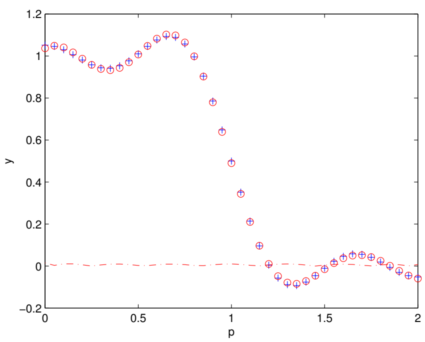
รูปที่ 15.9 แสดงผลการทดสอบเครือข่ายกับข้อมูลอินพุตที่ไม่ได้อยู่ในชุดฝึกสอน ข้อมูลดังกล่าวถูกชักตัวอย่างด้วยความถี่เพิ่มขึ้น 2.5 เท่า ผลลัพธ์ที่ได้แสดงถึงความทำให้เป็นทั่วไปของเครือข่าย RBF ได้เป็นอย่างดี นอกไปจากนั้น รูปที่ 15.10 แสดงผลการทดสอบกับเครือข่ายเมื่อข้อมูลอินพุต/เอาต์พุตสำหรับฝึกสอนถูกรบกวนด้วย



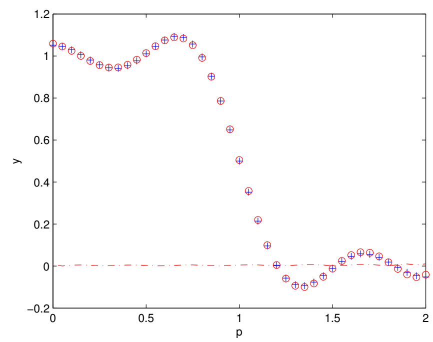
(ก) $\sigma = 0.35$ SSE = 0.4279



(ข) $\sigma = 0.45$ SSE = 0.1362



(ค) $\sigma = 0.50$ SSE = 0.2623



(ง) $\sigma = 0.55$ SSE = 0.1822

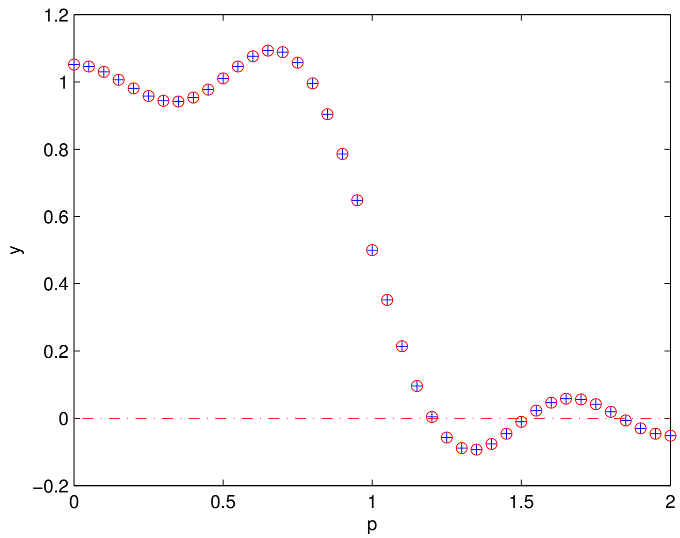
รูปที่ 15.7: ผลการประมาณค่าฟังก์ชันด้วยเครือข่าย RBF ขนาด 8 นิวรอนด้วย σ ค่าต่างๆ สัญลักษณ์ '+' แทนข้อมูลอินพุต/เอาต์พุตที่ใช้ฝึกสอน สัญลักษณ์ 'o' แทนเอาต์พุตที่ได้จากเครือข่ายที่ผ่านการฝึกสอนแล้ว และเส้นประด้านล่างของกราฟแสดงผลรวมค่ากำลังสองของความผิดพลาด (SSE) ระหว่างเอาต์พุตจากข้อมูลจริงและเอาต์พุตที่ได้จากเครือข่าย

สัญญาณสุ่มที่มีการกระจายแบบสม่ำเสมอขนาด 0.2 (ประมาณ 15% ของขนาดสูงสุดของข้อมูลอินพุต/เอาต์พุต) ผลที่ได้ยังคงแสดงให้เห็นความสอดคล้องของการประมาณฟังก์ชันด้วยเครือข่าย RBF ถึงแม้จะมีสัญญาณรบกวนก็ตาม ดังนั้นเครือข่าย RBF มีความทนทานต่อสัญญาณรบกวนได้ในระดับหนึ่ง ซึ่งเป็นคุณสมบัติที่ดีเหมือนกับเครือข่ายประสาทเทียมแบบอื่นๆ ทั่วไป ■

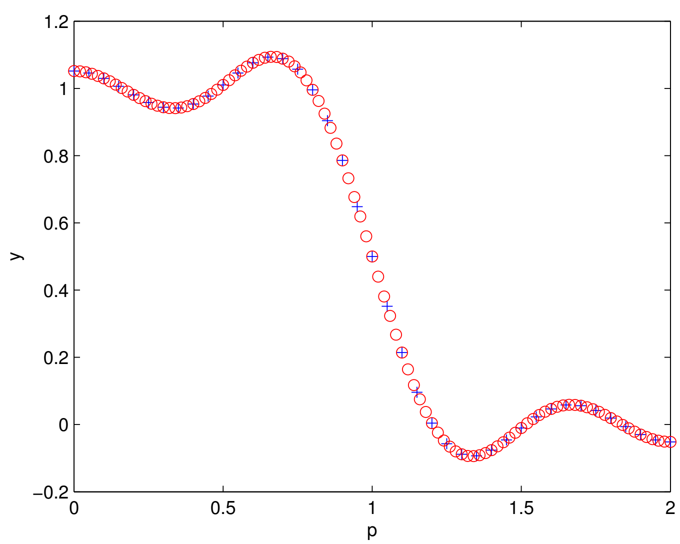
ตัวอย่างข้างต้นแสดงประสิทธิภาพในการใช้เครือข่าย RBF เป็นฟังก์ชันการประมาณค่า เครือข่าย RBF เป็นที่ยอมรับว่าสามารถใช้ประมาณค่าฟังก์ชันใดๆ ได้เป็นอย่างดี จึงได้มีการนำเอาเครือข่าย RBF ไปประยุกต์ใช้อย่างมากมาย

15.3 สรุป

เครือข่ายฟังก์ชันฐานรัศมีหรือ RBF เป็นเครือข่ายที่มีโครงสร้างที่ไม่ซับซ้อน องค์ประกอบภายในเครือข่ายและอัลกอริทึมการเรียนรู้ของเครือข่าย RBF เองแสดงให้เห็นถึงความสามารถในการเป็นฟังก์ชันประมาณแบบเลขจำนวนจริง ได้เป็นอย่างแม่นยำ เมื่อเทียบกันกับเครือข่ายเพอร์เซ็พตรอนแบบหลายชั้นแล้ว การฝึกสอนเครือข่าย RBF จะใช้เวลาเร็วกว่าเครือข่ายแบบหลายชั้นมาก แต่เนื่องจากการที่เป็นเครือข่ายชั้นเดียว (ชั้นซ่อนเร้น) จำนวนนิว

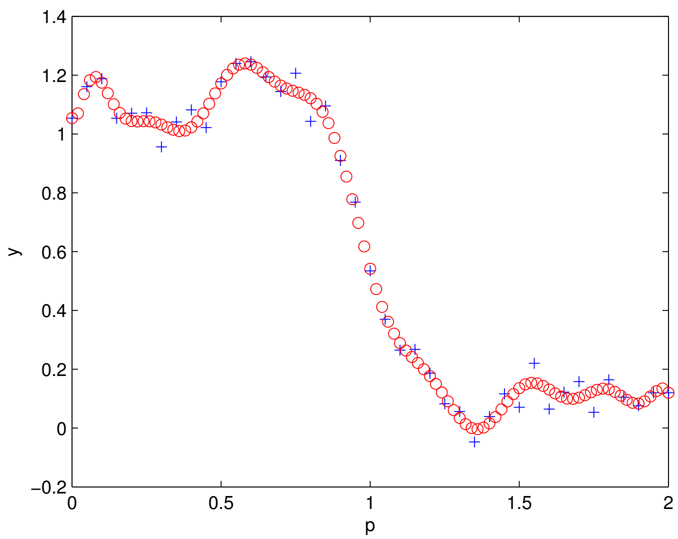


รูปที่ 15.8: ผลการประมาณค่าฟังก์ชันด้วยเครือข่าย RBF ขนาด 41 นิวรอนจากการฝึกสอนด้วยคำสั่ง newrbe ให้ค่า SSE $= 3.6575 \times 10^{-7}$ สัญลักษณ์ '+' แทนข้อมูลอินพุต/เอาต์พุตที่ใช้ฝึกสอน สัญลักษณ์ 'o' แทนเอาต์พุตที่ได้จากเครือข่ายที่ผ่านการฝึกสอนแล้ว และเส้นประด้านล่างของกราฟแสดงผลรวมค่ากำลังสองของความผิดพลาด (SSE) ระหว่างเอาต์พุตจากข้อมูลจริงและเอาต์พุตที่ได้จากเครือข่าย



รูปที่ 15.9: ตัวอย่างผลการประมาณค่าฟังก์ชันด้วยข้อมูลที่ไม่ใช่ข้อมูลในการฝึกสอน สัญลักษณ์ '+' แทนข้อมูลอินพุต/เอาต์พุตที่ใช้ฝึกสอนและสัญลักษณ์ 'o' แทนเอาต์พุตที่ได้จากเครือข่ายที่ผ่านการฝึกสอนแล้ว

รอนที่ต้องการสำหรับปัญหาหนึ่งๆ อาจจะต้องมีจำนวนที่มากกว่า อย่างไรก็ดี เครือข่าย RBF ได้เป็นที่ยอมรับและถูกนำเอามาประยุกต์ใช้อย่างหลากหลาย

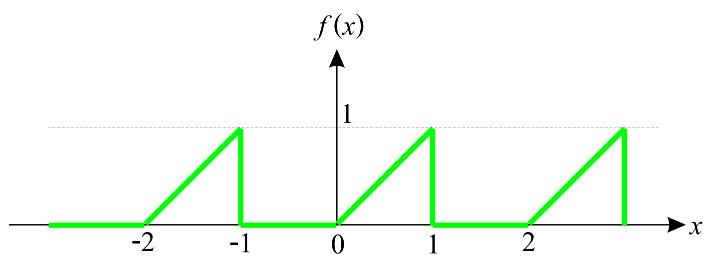


รูปที่ 15.10: ผลการประมาณค่าฟังก์ชันด้วยข้อมูลที่มีสัญญาณรบกวน สัญลักษณ์ '+' แทนข้อมูลอินพุต/เอาต์พุตที่ใช้ฝึกสอนและสัญลักษณ์ 'o' แทนเอาต์พุตที่ได้จากเครือข่ายที่ผ่านการฝึกสอนแล้ว

โจทย์คำถาม

15.1. จงออกแบบพร้อมทั้งอธิบายรายละเอียดเครือข่าย RBF พร้อมทั้งฝึกสอนเครือข่ายให้เรียนรู้การประมาณค่าฟังก์ชันในรูปที่ 15.11

- ใช้ค่าชักตัวอย่างจากอนุกรมฟูริเยร์ (ประมาณ 2-3 เทอมแรก)
- ใช้ค่าชักตัวอย่างจากรูปคลื่นสามเหลี่ยมโดยตรง
- ทดลองปรับพารามิเตอร์ของเครือข่าย วิเคราะห์และสรุปผลการทดลองที่ได้
- ทดลองใส่สัญญาณรบกวนแบบเกาส์เซียนขาว (Gaussian white noise) ที่มีค่าเฉลี่ยเท่ากับศูนย์และค่าเบี่ยงเบนมาตรฐานเท่ากับหนึ่งเข้าไปในระบบ พร้อมทั้งทำการทดลองเครือข่ายทั้งหมดใหม่อีกครั้ง วิเคราะห์และอภิปรายผลที่ได้



รูปที่ 15.11: รูปคลื่นสามเหลี่ยม

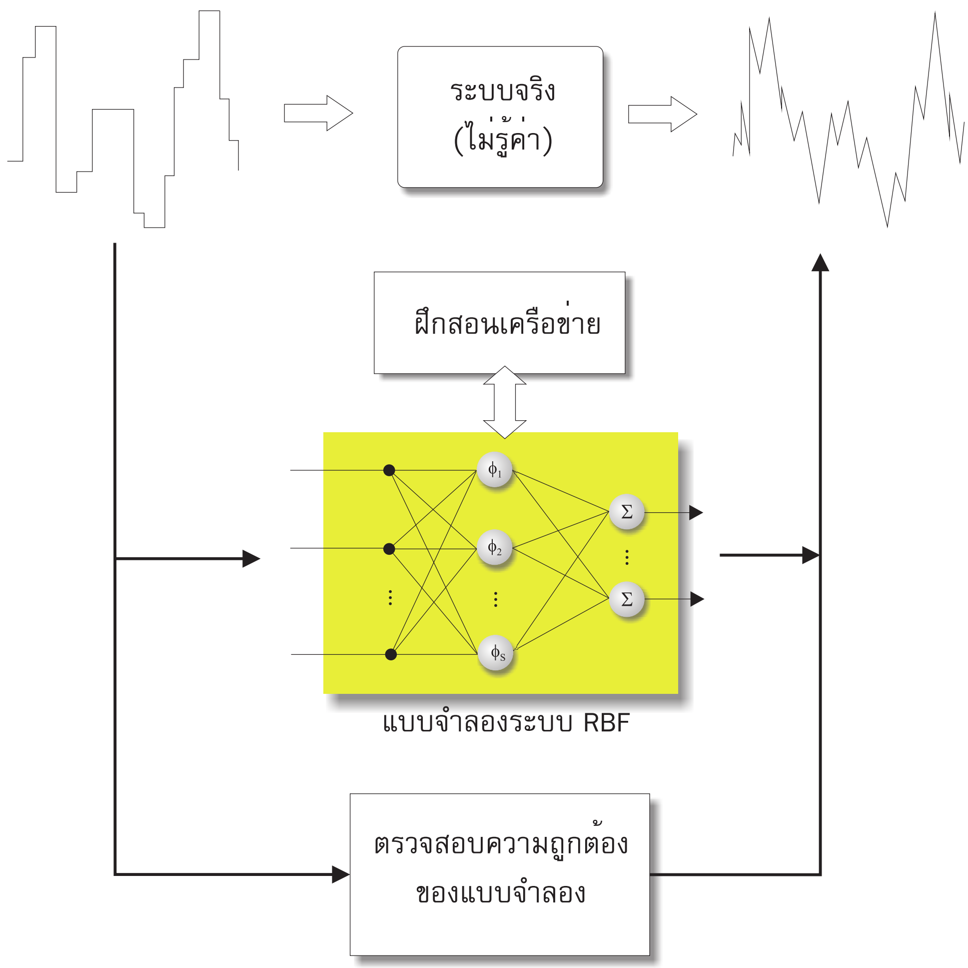
15.2. จงออกแบบการฝึกสอนเครือข่าย RBF ด้วยวิธีการค้นหาคำตอบแบบชาญฉลาดต่อไปนี้ (ใช้ตัวอย่างข้อมูลในโจทย์คำถามข้อที่ 1 ในการทดสอบการฝึกสอนเครือข่าย)

- จีนเนติกอัลกอริทึม (GA)
- การค้นหาแบบตาบูเชิงปรับตัว (ATS)
- การหาค่าเหมาะที่สุดด้วยการเคลื่อนที่ของกลุ่มอนุภาพ (PSO)

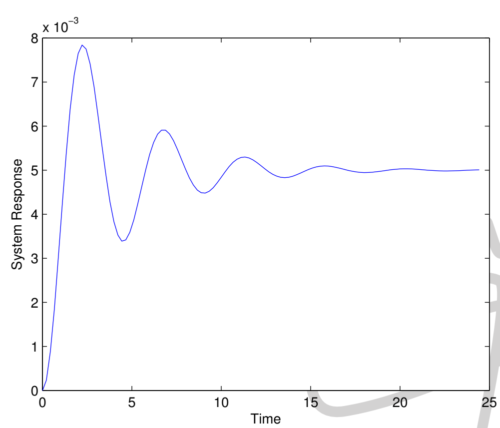
15.3. เครือข่าย RBF และความสามารถในการปรับเส้นโค้งสามารถนำมาประยุกต์ใช้ในการระบุเอกลักษณ์ได้ ดังแผนผังในรูปที่ 15.12 การระบุเอกลักษณ์นั้นมีหลายรูปแบบ พิจารณาการระบุเอกลักษณ์แบบเชิงเส้น (Linear System Identification หรือ LSI) โดยระบบมีฟังก์ชันถ่ายโอนคือ (ผลตอบสนองฟังก์ชันขั้นบันไดหนึ่งหน่วยของระบบแสดงในรูปที่ 15.13)

$$H(s) = \frac{1.11 \times 10^{-16} s + 0.01}{s^2 + 0.5s + 2.0}$$

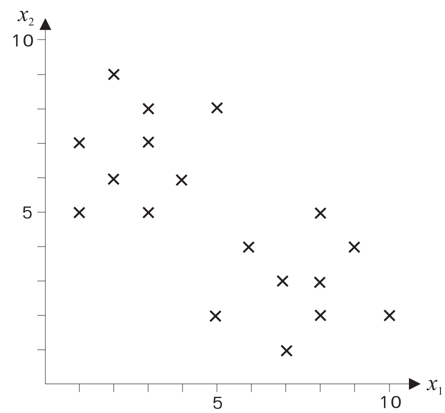
- ให้ทำการจำลองระบบด้วยฟังก์ชันถ่ายโอนที่กำหนดให้ พร้อมทั้งบันทึกค่าอินพุต/เอาต์พุตของระบบเอาไว้
- ให้ออกแบบใช้เครือข่าย RBF สำหรับระบุเอกลักษณ์ระบบข้างต้น โดยใช้ค่าอินพุต/เอาต์พุตที่บันทึกเอาไว้ในการฝึกสอนเครือข่าย วิเคราะห์พารามิเตอร์ต่างๆ ของระบบพร้อมทั้งอภิปรายผลที่ได้
- ทดลองฝึกสอนเครือข่ายด้วยวิธีการเรียนรู้แบบต่างๆ เปรียบเทียบผลที่ได้
- ทดลองป้อนอินพุตแบบอื่นๆ ให้กับระบบ (นอกเหนือไปจากฟังก์ชันขั้นบันไดหนึ่งหน่วย) เพื่อทดสอบการเป็นทั่วไปของเครือข่าย รวมไปทั้งเพิ่มสัญญาณรบกวนแบบต่างๆ ให้กับระบบ วิเคราะห์ผลที่ได้



รูปที่ 15.12: แผนผังการระบุเอกลักษณ์ด้วยแบบจำลองเครือข่าย RBF



รูปที่ 15.13: ผลตอบสนองฟังก์ชันขั้นบันไดหนึ่งหน่วยของระบบ



รูปที่ 15.14: ข้อมูลสำหรับการจับกลุ่ม

15.4. พิจารณาเครือข่าย RBF (Radial Basis Function) และ MLP (Multi-Layer Perceptron)

- จงอธิบายความแตกต่างระหว่างเครือข่ายทั้งสอง
- จงแสดงอธิบายการออกแบบสร้างเครือข่าย RBF จากเครือข่าย MLP
- ทดสอบเครือข่าย RBF ที่สร้างจากเครือข่าย MLP กับโจทย์ปัญหาข้อที่ 1

15.5. เครือข่าย RBF สามารถถูกออกแบบให้ทำหน้าที่จับกลุ่มข้อมูลได้ (clustering) จงออกแบบเครือข่าย RBF พร้อมทั้งเลือกพารามิเตอร์ต่างๆ ให้เหมาะสม ทดสอบการจับกลุ่มข้อมูลในรูปที่ 15.14

บรรณานุกรม

D. S. Broomhead and D. Lowe. Multivariable functional interpolation and adaptive networks. In *Complex Systems*, volume 2, pages 269--303, 1988.

F. M. Ham and I. Kostanic. *Principles of neurocomputing for science & engineering*. McGraw-Hill, 2001.

P.D. Wasserman. *Advanced Methods in Neural Computing*. Van Nostrand Reinhold, New York, 1993.