

การเรียนรู้แบบวิโดว์-ฮอฟฟ์ Widrow-Hoff Learning

ในปีค.ศ. 1960 Bernard Widrow และนักเรียนบัณฑิตศึกษาของเขาชื่อว่า Marcian Hoff [Widrow and M. E. Hoff, 1960] จากมหาวิทยาลัยสแตนฟอร์ด ประเทศสหรัฐอเมริกาได้นำเสนอเครือข่าย ADALINE (ADaptive LInear NEuron) พร้อมกับกฎการเรียนรู้ที่เรียกว่าอัลกอริทึมกำลังสองเฉลี่ยน้อยที่สุด หรือ LMS (least mean square) [Widrow, 1987] เครือข่าย ADALINE มีลักษณะคล้ายคลึงกับเพอร์เซ็ปตรอน เพียงแต่แตกต่างกันที่ฟังก์ชันถ่ายโอนของ ADALINE เป็นฟังก์ชันเส้นตรงแทนที่จะเป็นฟังก์ชันฮาร์ดลิมิตเหมือนในกรณีของเพอร์เซ็ปตรอน ทั้ง ADALINE และเพอร์เซ็ปตรอนจึงมีจุดอ่อนคือสามารถแก้ปัญหาที่แบ่งแยกได้แบบเชิงเส้นเท่านั้น (linearly separable) อย่างไรก็ตามอัลกอริทึม LMS ถือว่ามีประสิทธิภาพเหนือกว่ากฎการเรียนรู้แบบเพอร์เซ็ปตรอนมาก ถึงแม้ว่าการเรียนรู้แบบเพอร์เซ็ปตรอนจะรับประกันว่าเข้าสู่การแบ่งแยกอินพุตตามเป้าหมายที่ได้เสนอ แต่ว่าเครือข่ายที่ได้อาจจะค่อนข้างไวต่อสัญญาณรบกวน เนื่องจากเป้าหมายที่ใช้ในการเรียนรู้มีตำแหน่งอยู่ใกล้กับเส้นแบ่งพื้นที่ ซึ่งต่างกับอัลกอริทึม LMS ที่คำนวณบนหลักของค่าความผิดพลาดกำลังสองเฉลี่ยที่น้อยที่สุด (minimized mean square error) ดังนั้นอัลกอริทึม LMS จะให้ผลของเส้นแบ่งพื้นที่ที่วางตัวไกลจากตำแหน่งของเป้าหมายมากที่สุดอัลกอริทึม LMS ยังมีการประยุกต์ใช้งานจริงได้มากกว่ากฎการเรียนรู้ของเพอร์เซ็ปตรอน โดยเฉพาะในสาขาวิชาที่เกี่ยวข้องกับการประมวลผลสัญญาณดิจิทัล

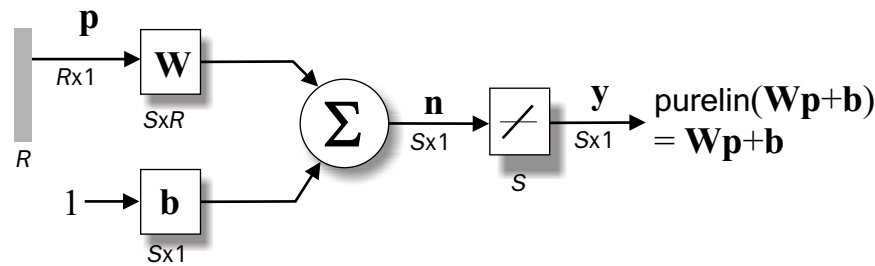
11.1 เครือข่าย ADALINE

เครือข่าย ADALINE แสดงในรูปที่ 11.1 สังเกตว่ามีลักษณะคล้ายๆ กันกับเครือข่ายเพอร์เซ็ปตรอน เพียงแต่มีฟังก์ชันถ่ายโอนเป็นฟังก์ชันเส้นตรง ดังนั้นจะได้เอาต์พุตของเครือข่ายคือ

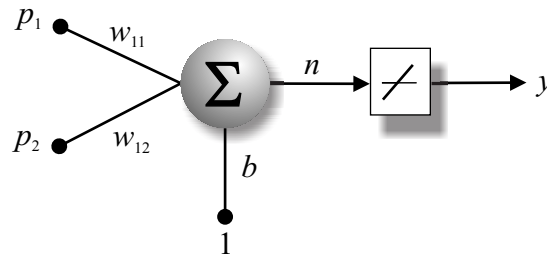
$$y = \text{purelin}(\mathbf{W}\mathbf{p} + \mathbf{b}) = \mathbf{W}\mathbf{p} + \mathbf{b} \quad (11.1)$$

โดยที่ y เป็นเอาต์พุตของเครือข่าย \mathbf{W} คือเมตริกซ์น้ำหนักประสาท \mathbf{p} คือเวกเตอร์อินพุตและ \mathbf{b} คือเวกเตอร์ไบอัส เช่นเดียวกันกับเอาต์พุตของนิวรอนตัวที่ i ในเครือข่าย ADALINE สามารถเขียนในรูปต่อไปนี้ได้

$$y_i = \text{purelin}(n_i) = \text{purelin}(i\mathbf{w}\mathbf{p} + b_i) = i\mathbf{w}\mathbf{p} + b_i \quad (11.2)$$



รูปที่ 11.1: เครือข่าย ADALINE



รูปที่ 11.2: ADALINE เดี่ยวแบบ 2 อินพุต

โดยที่ $i\mathbf{w}$ เป็นเมตริกซ์แถวที่ i ของ \mathbf{W} นั่นคือ

$$i\mathbf{w} = \begin{bmatrix} w_{i1} & w_{i2} & \cdots & w_{iR} \end{bmatrix} \quad (11.3)$$

11.2 ADALINE เดี่ยว

Single ADALINE

พิจารณา ADALINE เดี่ยวที่มี 2 อินพุต (รูปที่ 11.2) จะได้เอาต์พุตของเครือข่ายคือ

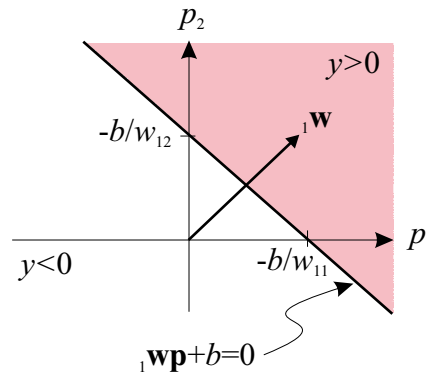
$$\begin{aligned} y &= \text{purelin}(n) \\ &= \text{purelin}({}_1\mathbf{w}\mathbf{p} + b) \\ &= {}_1\mathbf{w}\mathbf{p} + b \\ &= w_{11}p_1 + w_{12}p_2 + b \end{aligned} \quad (11.4)$$

เช่นเดียวกับเพอร์เซพตรอน ADALINE มีเส้นแบ่งพื้นที่โดยแบ่งอินพุตออกเป็น 2 พื้นที่ดังแสดงในรูปที่ 11.3 เราสามารถพิจารณาหาเส้นแบ่งพื้นที่นี้ได้โดยกำหนดให้ $n = {}_1\mathbf{w}\mathbf{p} + b = 0$ ความแตกต่างของผลลัพธ์ที่ได้คือเส้นแบ่งพื้นที่แบ่งเป็นพื้นที่ $y > 0$ และ $y < 0$ (จากฟังก์ชันถ่ายโอน purelin) เช่นเดียวกับเพอร์เซพตรอน ADALINE สามารถใช้กับปัญหาที่สามารถแบ่งแยกได้แบบเชิงเส้น (linearly separable) เท่านั้น

11.3 กฎการเรียนรู้ของวิโดว์-ฮอฟฟ์

Widrow-Hoff Learning Rule

กฎการเรียนรู้ของวิโดว์-ฮอฟฟ์หรือเป็นที่รู้จักในชื่อกฎของเดลตา (delta rule) ใช้อัลกอริทึม LMS (Least Mean Square) ในการฝึกสอนเครือข่าย ADALINE การฝึกสอนเป็นแบบมีผู้ฝึกสอน โดยการกำหนดคู่อินพุตและเป้าหมาย



รูปที่ 11.3: เส้นแบ่งพื้นที่ของ ADALINE เดี่ยวแบบ 2 อินพุต

ที่ต้องการให้กับเครือข่ายดังนี้

$$\{\mathbf{p}_1, t_1\}, \{\mathbf{p}_2, t_2\}, \dots, \{\mathbf{p}_q, t_q\}, \dots, \{\mathbf{p}_Q, t_Q\} \quad (11.5)$$

โดยที่ \mathbf{p}_q เป็นเวกเตอร์อินพุตตัวที่ q ของเครือข่ายและ t_q เป็นเวกเตอร์เป้าหมายของอินพุตที่ q นั้นๆ เมื่ออินพุตถูกป้อนเข้าสู่เครือข่ายเอาต์พุตของเครือข่ายจะถูกนำไปเปรียบเทียบกับเป้าหมายของอินพุตนั้น คู่เวกเตอร์อินพุต-เป้าหมายนี้จะถูกใช้ในการฝึกสอนเครือข่าย ADALINE ดังรายละเอียดในหัวข้อต่อไป

11.3.1 พื้นผิวค่าความผิดพลาดกำลังสองเฉลี่ย Mean-Square-Error Surface

อัลกอริทึม LMS จะทำการปรับค่าน้ำหนักประสาทและไบอัสของ ADALINE ไปในทิศทางที่ทำให้ค่าความผิดพลาดเฉลี่ยกำลังสองมีค่าน้อยที่สุด โดยค่าความผิดพลาดนั้นก็คือค่าความแตกต่างระหว่างเอาต์พุตของเครือข่าย กับเป้าหมายที่ต้องการ เพื่อความสะดวกในการคำนวณ พารามิเตอร์ต่างๆ จะถูกยุบรวมให้เป็นเวกเตอร์เดียวดังนี้

$$\mathbf{x} = \begin{bmatrix} \mathbf{p}^T \\ b \end{bmatrix} \quad (11.6)$$

และทำการปรับเวกเตอร์ของอินพุตดังนี้ (พิจารณาค่า 1 ที่เพิ่มขึ้นมาที่เพิ่มขึ้นในเวกเตอร์ \mathbf{x} เพื่อให้สอดคล้องกับค่าไบอัส b)

$$\mathbf{z} = \begin{bmatrix} \mathbf{p} \\ 1 \end{bmatrix} \quad (11.7)$$

จาก $y = \mathbf{w}^T \mathbf{p} + b$ เราจะได้เอาต์พุตของเครือข่ายดังต่อไปนี้

$$y = \mathbf{x}^T \mathbf{z} \quad (11.8)$$

จากรูปเอาต์พุต y ข้างต้น ทำให้สามารถเขียนรูปของค่าความผิดพลาดเฉลี่ยกำลังสองได้ดังนี้

$$\begin{aligned} F(\mathbf{x}) &= E[e^2] \\ &= E[(t - y)^2] \\ &= E[(t - \mathbf{x}^T \mathbf{z})^2] \end{aligned} \quad (11.9)$$

โดยที่ $E[\cdot]$ คือฟังก์ชันค่าเฉลี่ย (expectation) และมีผลกับคู่เวกเตอร์อินพุตและเวกเตอร์เป้าหมายเท่านั้น กล่าวคือ ค่าน้ำหนักประสาทและไบอัสจะมีผลเปลี่ยนแปลงจากการฝึกสอนเครือข่าย เมื่อทำการกระจายสมการข้างต้น จะได้

$$\begin{aligned} F(\mathbf{x}) &= E[t^2 - 2t\mathbf{x}^T \mathbf{z} + \mathbf{x}^T \mathbf{z} \mathbf{z}^T \mathbf{x}] \\ &= E[t^2] - 2\mathbf{x}^T E[t\mathbf{z}] + \mathbf{x}^T E[\mathbf{z} \mathbf{z}^T] \mathbf{x} \end{aligned} \quad (11.10)$$

จัดรูปใหม่ได้ดังนี้

$$F(\mathbf{x}) = c - 2\mathbf{x}^T \mathbf{h} + \mathbf{x}^T \mathbf{R} \mathbf{x} \quad (11.11)$$

โดยที่ $c = E[t^2]$ $\mathbf{h} = E[t\mathbf{z}]$ และ $\mathbf{R} = E[\mathbf{z}\mathbf{z}^T]$ เวกเตอร์ \mathbf{h} ที่ได้คือสหสัมพันธ์ (cross-correlation) ระหว่างอินพุตและเป้าหมายของอินพุตนั้น ในขณะที่ \mathbf{R} เป็นอัตสัมพันธ์ของอินพุต (auto-correlation) พิจารณาสมการ (11.11) ซึ่งสามารถจัดให้อยู่ในรูปของสมการควอดราติก (quadratic equation) ได้ดังนี้

$$F(\mathbf{x}) = c + \mathbf{d}^T \mathbf{x} + \frac{1}{2} \mathbf{x}^T \mathbf{A} \mathbf{x} \quad (11.12)$$

โดยที่ $\mathbf{d} = -2\mathbf{h}$ และ $\mathbf{A} = 2\mathbf{R}$

พิจารณาสมการควอดราติกซึ่งคุณลักษณะของสมการจะขึ้นอยู่กับเมตริกซ์เฮสเซียน (Hessian) \mathbf{A} ซึ่งมีค่าเป็นสองเท่าของเมตริกซ์ \mathbf{R} โดยที่เมตริกซ์ \mathbf{R} เป็นเมตริกซ์อัตสัมพันธ์ซึ่งเป็นเมตริกซ์ที่เป็นแบบบวกแน่นอน (positive definite) หรือเป็นแบบบวกกึ่งแน่นอน (positive semidefinite) จะได้ว่าค่าลักษณะเฉพาะ (eigenvalue) ของเมตริกซ์ \mathbf{A} จะมีค่ามากกว่าหรือเท่ากับศูนย์เท่านั้น ซึ่งสามารถสรุปได้ดังนี้

- ถ้าค่าลักษณะเฉพาะเป็นบวกเท่านั้น $F(\mathbf{x})$ จะให้คำตอบที่เป็นค่าน้อยที่สุดแบบวงกว้าง (global minimum)
- ถ้าค่าลักษณะเฉพาะมีค่าศูนย์อยู่ด้วย $F(\mathbf{x})$ จะให้คำตอบที่เป็นค่าน้อยที่สุดแบบอ่อน (weak minimum) หรือไม่มีค่าน้อยที่สุดเลย โดยจะขึ้นอยู่กับเวกเตอร์ $\mathbf{d} = -2\mathbf{h}$

เราสามารถหาจุดยอดของฟังก์ชัน $F(\mathbf{x})$ ได้โดยคำนวณจากค่าเกรเดียนต์

$$\begin{aligned} \nabla F(\mathbf{x}) &= \nabla (c + \mathbf{d}^T \mathbf{x} + \frac{1}{2} \mathbf{x}^T \mathbf{A} \mathbf{x}) \\ &= \mathbf{d} + \mathbf{A} \mathbf{x} \\ &= -2\mathbf{h} + 2\mathbf{R} \mathbf{x} \end{aligned} \quad (11.13)$$

กำหนดให้ค่าเกรเดียนต์เป็นศูนย์

$$-2\mathbf{h} + 2\mathbf{R} \mathbf{x} = 0 \quad (11.14)$$

จะได้ว่าถ้าเมตริกซ์ \mathbf{R} เป็นแบบบวกแน่นอน จุดยอดที่ได้จะเป็นจุดที่มีค่าน้อยที่สุดแบบวงแคบเฉพาะถิ่น ณ ตำแหน่ง \mathbf{x}^* ดังนี้

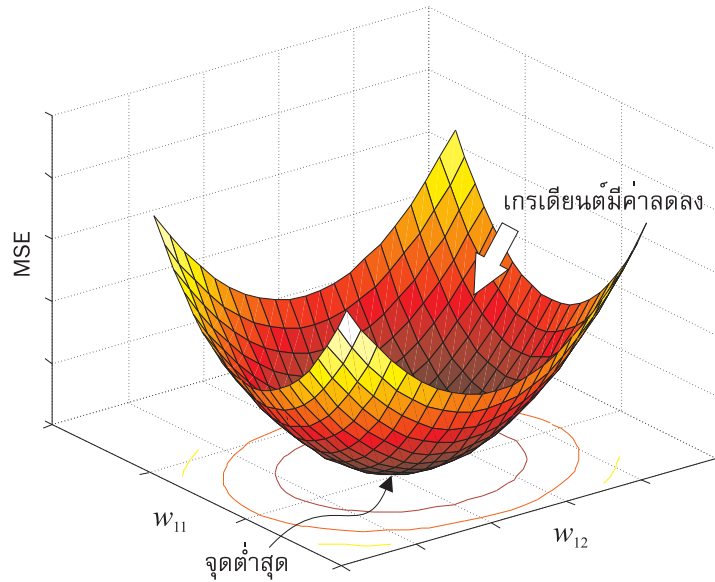
$$\mathbf{x}^* = \mathbf{R}^{-1} \mathbf{h} \quad (11.15)$$

รูปที่ 11.4 แสดงพื้นผิวค่าความผิดพลาด (error surface) โดยทั่วไปของค่าความผิดพลาดเฉลี่ยกำลังสองจากเครือข่าย ADALINE เดียวแบบ 2 อินพุต (ไม่คิดผลของไบอัส) ในรูปแสดงจุดต่ำสุดของพื้นผิว ที่ซึ่งเครือข่ายมีค่าความผิดพลาดเฉลี่ยกำลังสองน้อยที่สุด นั่นคือเครือข่ายมีค่าน้ำหนักประสาท (w_{11} และ w_{12}) ที่ดีที่สุด ทิศทางของเกรเดียนต์ที่มุ่งไปสู่จุดต่ำสุดนี้จะมีทิศทางที่ลดลง (gradient descent) ซึ่งเป็นที่มาของอัลกอริทึมในการฝึกสอนเครือข่าย เพื่อให้ได้ค่าความผิดพลาดเฉลี่ยกำลังสองน้อยที่สุด ดังรายละเอียดในตอนต่อไป

11.3.2 อัลกอริทึม LMS

จากการวิเคราะห์หาจุดยอดจากฟังก์ชันของค่าความผิดพลาดกำลังสองเฉลี่ย ขั้นตอนต่อไปคือการออกแบบอัลกอริทึม ในการคำนวณที่นำไปสู่จุดที่ฟังก์ชันค่าความผิดพลาดกำลังสองเฉลี่ยมีค่าน้อยที่สุด เราสามารถใช้สมการ (11.15) โดยทำการคำนวณเมตริกซ์ผกผัน \mathbf{R}^{-1} ซึ่งในทางปฏิบัติจะเป็นวิธีที่ไม่สะดวก แต่จากเกรเดียนต์ที่มีอยู่ เราสามารถใช้อัลกอริทึมลงแบบชันสุด (steepest descent) ได้ อย่างไรก็ตามก็ยังไม่สะดวกในการที่จะคำนวณหาเมตริกซ์ \mathbf{R} และ \mathbf{h} ในสมการ (11.13) เราจะใช้อัลกอริทึมประมาณค่าลงแบบชันสุด (approximation steepest descent algorithm) ทำการประมาณค่าของเกรเดียนต์จากค่าความผิดพลาดกำลังสองเฉลี่ย $F(\mathbf{x})$ พิจารณาให้ $\hat{F}(\mathbf{x})$ เป็นค่าประมาณของฟังก์ชัน $F(\mathbf{x})$ ดังกล่าว จะได้ว่า ณ รอบที่ k

$$\hat{F}(\mathbf{x})_k = (t(k) - y(k))^2 = e^2(k) \quad (11.16)$$



รูปที่ 11.4: พื้นผิวค่าความผิดพลาดเฉลี่ยกำลังสองของเครือข่าย ADALINE เดี่ยวแบบ 2 อินพุต

แต่ละรอบของการคำนวณ จะได้ค่าของเกรเดียนต์ในรูปดังต่อไปนี้

$$\nabla \hat{F}(\mathbf{x})_k = \nabla e^2(k) \quad (11.17)$$

พิจารณาองค์ประกอบของ $\nabla e^2(k)$ จะได้ว่าองค์ประกอบ R ตัวแรกก็คืออนุพันธ์เทียบกับค่าน้ำหนักประสาท w ของเครือข่ายและองค์ประกอบตัวที่ $R+1$ จะเป็นอนุพันธ์เทียบกับไบอัส b ดังนั้นเราจะได้ว่า สำหรับองค์ประกอบย่อยที่ $j = 1, 2, \dots, R$ ของค่าเกรเดียนต์ $\nabla e^2(k)$ จะได้

$$[\nabla e^2(k)]_j = \frac{\partial e^2(k)}{\partial w_{1j}} \quad (11.18)$$

$$= 2e(k) \frac{\partial e(k)}{\partial w_{1j}} \quad (11.19)$$

และสำหรับองค์ประกอบย่อยที่ $j = R+1$ จะได้

$$[\nabla e^2(k)]_{R+1} = \frac{\partial e^2(k)}{\partial b} \quad (11.20)$$

$$= 2e(k) \frac{\partial e(k)}{\partial b} \quad (11.21)$$

พิจารณาเกรเดียนต์ในสมการที่ (11.19) จะได้

$$\frac{\partial e(k)}{\partial w_{1j}} = \frac{\partial [t(k) - y(k)]}{\partial w_{1j}} \quad (11.22)$$

$$= \frac{\partial}{\partial w_{1j}} [t(k) - (\mathbf{w}_1 \mathbf{p}(k) + b)] \quad (11.23)$$

$$= \frac{\partial}{\partial w_{1j}} \left[t(k) - \left(\sum_{i=1}^R w_{1j} p_i(k) + b \right) \right] \quad (11.24)$$

โดยที่ $p_i(k)$ คือองค์ประกอบที่ i ของเวกเตอร์อินพุต ณ รอบที่ k ดังนั้นจะได้ว่า

$$\frac{\partial e(k)}{\partial w_{1j}} = -p_j(k) \quad (11.25)$$

เช่นเดียวกันสำหรับส่วนของไบอัสจะได้ว่า

$$\frac{\partial e(k)}{\partial b} = -1 \quad (11.26)$$

เมื่อพิจารณา $p_j(k)$ และ 1 แล้ว จะเห็นว่าทั้งสองเทอมเป็นองค์ประกอบของเวกเตอร์อินพุต \mathbf{z} ดังนั้นค่าเกรเดียนต์ที่คำนวณได้ ณ รอบที่ k คือ

$$\nabla \hat{F}(\mathbf{x})_k = \nabla e^2(k) = -2e(k)\mathbf{z}(k) \quad (11.27)$$

จะเห็นว่าจากค่าความผิดพลาดกำลังสองเฉลี่ย เราสามารถคำนวณค่าเกรเดียนต์ ณ รอบที่ k ด้วยการใช้เพียงผลคูณค่าความผิดพลาด $e(k)$ กับเวกเตอร์อินพุต $\mathbf{z}(k)$ ค่าเกรเดียนต์ที่ได้สามารถนำไปใช้ในอัลกอริทึมลงแบบขั้นสุด โดยมีค่าคงที่การเรียนรู้ α ดังนี้

$$\mathbf{x}(k+1) = \mathbf{x}(k) - \alpha \nabla F(\mathbf{x}(k)) \quad (11.28)$$

แทนค่า $\nabla F(\mathbf{x}(k))$ ด้วย $\nabla \hat{F}(\mathbf{x})_k$ จากสมการที่ (11.27) จะได้

$$\mathbf{x}(k+1) = \mathbf{x}(k) + 2\alpha e(k)\mathbf{z}(k) \quad (11.29)$$

หรือเขียนในรูปน้ำหนักประสาทและไบอัสได้เป็น

$${}_1\mathbf{w}(k+1) = {}_1\mathbf{w}(k) + 2\alpha e(k)\mathbf{p}(k) \quad (11.30)$$

และ

$$b(k+1) = b(k) + 2\alpha e(k) \quad (11.31)$$

ทั้งสมการ (11.30) และ (11.31) รวมแล้วคืออัลกอริทึมกำลังสองเฉลี่ยน้อยที่สุด (least mean square หรือ LMS) หรือเป็นที่รู้จักในนามของกฎเดลตา (delta rule) หรือกฎการเรียนรู้แบบวิโดว์-ฮอฟฟ์ (Widrow-Hoff learning rule) ในกรณีเครือข่ายมีเอาต์พุตมากกว่าหนึ่ง (นั่นคือมีหลายนิวรอน) การปรับค่าเมตริกซ์น้ำหนักประสาทของแถวที่ i คือ

$${}_i\mathbf{w}(k+1) = {}_i\mathbf{w}(k) + 2\alpha e_i(k)\mathbf{p}(k) \quad (11.32)$$

โดยที่ $e_i(k)$ เป็นค่าความผิดพลาดตัวที่ i ณ รอบที่ k และเช่นเดียวกันสำหรับไบอัส จะได้ว่า

$$b_i(k+1) = b_i(k) + 2\alpha e_i(k) \quad (11.33)$$

หรือในรูปของเมตริกซ์จะได้

$$\mathbf{W}(k+1) = \mathbf{W}(k) + 2\alpha e(k)\mathbf{p}^T(k) \quad (11.34)$$

และ

$$\mathbf{b}(k+1) = \mathbf{b}(k) + 2\alpha e(k) \quad (11.35)$$

โดยค่าคงที่การเรียนรู้ $\alpha < 2/\lambda_{max}$ ที่ซึ่ง λ_{max} เป็นค่าลักษณะเฉพาะที่มากที่สุดของเมตริกซ์เฮสเซียน \mathbf{A}

11.3.3 การปรับแต่งอัลกอริทึม LMS LMS Algorithm Modifications

อัลกอริทึม LMS ถูกนำมาใช้ในกฎการเรียนรู้ของเครือข่าย ADALINE เรียกว่าเป็นกฎการปรับแก้ค่าความผิดพลาดเชิงเส้น (เป็นเชิงเส้นตามโครงสร้างของเครือข่าย ADALINE) ซึ่งสามารถแบ่งกฎพื้นฐานออกเป็น 2 ชนิดได้แก่ μ -LMS และ α -LMS กฎทั้งสองจะทำการปรับค่าน้ำหนักประสาทของเครือข่ายอย่างเป็นสัดส่วนกับผลต่างระหว่างเอาต์พุตของเครือข่ายและเป้าหมายที่ต้องการ โดยรายละเอียดแล้วกฎการเรียนรู้แบบ μ -LMS ก็คือกฎการเรียนรู้แบบ LMS ธรรมดา นั่นเอง ในขณะที่กฎการเรียนรู้แบบ α -LMS สามารถพิจารณาได้ว่าเป็นกฎ μ -LMS ที่มีการทำให้เป็นบรรทัดฐานด้วยตนเอง (self-normalization) ตามกฎแบบ α -LMS แล้ว ค่าน้ำหนักประสาทของเครือข่ายจะถูกปรับตามความสัมพันธ์ต่อไปนี้

$$\mathbf{W}^{new} = \mathbf{W}^{old} + \frac{\alpha \mathbf{e} \mathbf{p}^T}{\|\mathbf{p}\|_2^2} \quad (11.36)$$

ข้อแตกต่างระหว่าง α -LMS กับ LMS แบบดั้งเดิมคือเทอมการทำให้เป็นบรรทัดฐาน $\|\mathbf{p}\|_2^2$ นั้นเอง อย่างไรก็ตามการใส่เทอมนี้ของกฎทั้งสองให้ความหมายที่แตกต่างกันอย่างสิ้นเชิง กล่าวคืออัลกอริทึม α -LMS ใช้หลักการรบกวนน้อยที่สุด (minimal-disturbance principle) กล่าวคือเมื่อทำการปรับเครือข่ายให้ตอบสนองต่อรูปแบบอินพุตใหม่ๆ ผลตอบสนองของรูปแบบอินพุตที่ฝึกสอนให้กับเครือข่ายไปแล้วควรจะได้รับผลกระทบโดยเฉลี่ยแล้วน้อยที่สุด ดังนั้น μ -LMS หรือ LMS แบบดั้งเดิมจะเป็นอัลกอริทึมที่ทำการหาจุดต่ำสุดของฟังก์ชัน MSE ในขณะที่ α -LMS จะทำการปรับค่าน้ำหนักประสาท เพื่อให้ค่าความผิดพลาดในขณะนั้นๆ ลดลง พิจารณาการปรับเปลี่ยนค่าความผิดพลาดของกฎ α -LMS

$$\Delta \mathbf{e} = \mathbf{e}^{new} - \mathbf{e}^{old} \quad (11.37)$$

$$= [\mathbf{t} - \mathbf{W}^{new} \mathbf{p}] - \mathbf{e}^{old} \quad (11.38)$$

$$= \left\{ \mathbf{t} - \left[\mathbf{W}^{old} + \frac{\alpha \mathbf{e} \mathbf{p}^T}{\|\mathbf{p}\|_2^2} \right] \mathbf{p} \right\} - \mathbf{e}^{old} \quad (11.39)$$

$$= \left[\mathbf{e}^{old} - \frac{\alpha \mathbf{e} \mathbf{p}^T \mathbf{p}}{\|\mathbf{p}\|_2^2} \right] - \mathbf{e}^{old} \quad (11.40)$$

$$= -\alpha \mathbf{e} \quad (11.41)$$

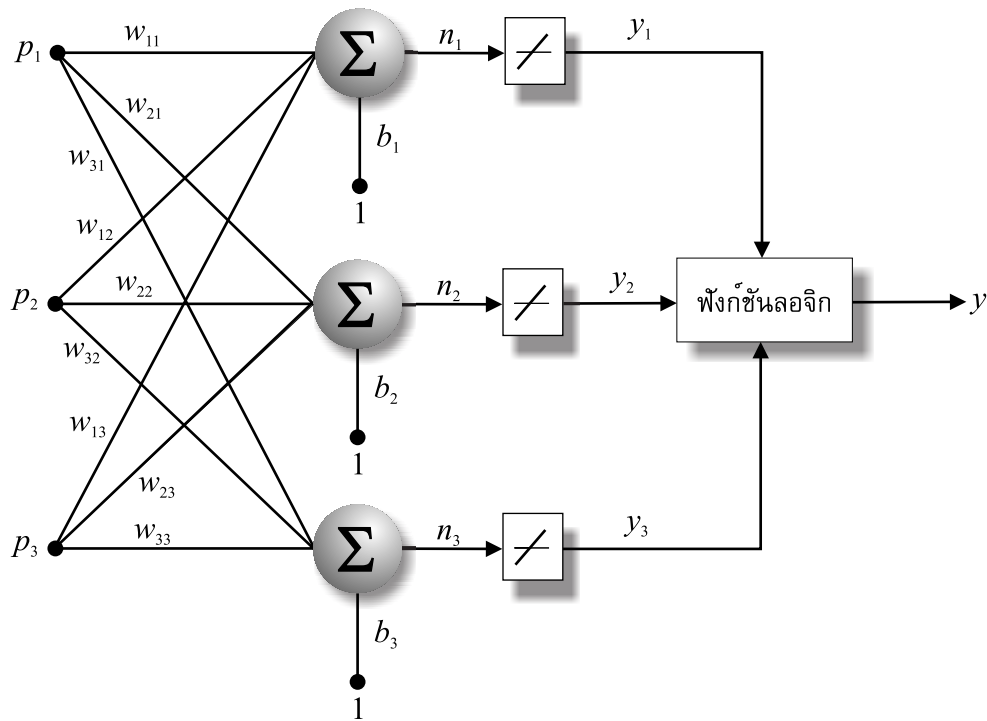
จะได้ว่า $\alpha = \frac{-\Delta \mathbf{e}}{\mathbf{e}^{old}}$ ซึ่งแสดงอัตราการเปลี่ยนแปลงของค่าความผิดพลาดในแต่ละรอบของการฝึกสอน ค่าของ α จะควบคุมเสถียรภาพและความเร็วในการลู่เข้าของเครือข่าย โดยทั่วไปแล้ว α จะมีค่าดังนี้

$$0.1 < \alpha < 1.0 \quad (11.42)$$

อัลกอริทึม α -LMS เป็นการทำให้เป็นบรรทัดฐานด้วยตนเอง นั่นคือค่าของ α ไม่ขึ้นกับขนาดของอินพุตที่ป้อนให้กับเครือข่าย ถ้าอินพุตเป็นค่าสองขั้ว $[-1, 1]$ เทอมบรรทัดฐาน $\|\mathbf{p}\|_2^2$ จะมีค่าเท่ากับจำนวนของน้ำหนักประสาท และจะไม่มีการเปลี่ยนแปลงตามค่าของอินพุต อย่างไรก็ตาม ถ้าอินพุตเป็นค่าฐานสอง (binary) $[0, 1]$ การปรับเปลี่ยนจะไม่เกิดขึ้นกับค่าน้ำหนักประสาทที่ได้รับค่าอินพุตที่เป็นศูนย์ เมื่อเทียบกับอินพุตแบบสองขั้วแล้ว ค่าน้ำหนักประสาททั้งหมดของเครือข่ายจะถูกปรับในแต่ละรอบ มีผลให้การลู่เข้าของเครือข่ายมีความเร็วสูงขึ้น

11.4 MADALINE

เครือข่าย ADALINE เดียวมีข้อจำกัดในการใช้งานได้กับปัญหาที่สามารถแบ่งแยกได้แบบเชิงเส้นเท่านั้น ปัญหาดังกล่าวสามารถแก้ไขได้โดยการทำการแปลงแบบไม่เป็นเชิงเส้น (non-linear transformation) กับอินพุตก่อนที่จะทำการป้อนให้กับเครือข่าย อีกวิธีหนึ่งคือเลือกใช้เครือข่าย MADALINE (Multiple ADALINE) ซึ่งสามารถใช้กับ



รูปที่ 11.5: ตัวอย่างเครือข่าย MADALINE I ที่ประกอบไปด้วย ADALINE 3 ชุด

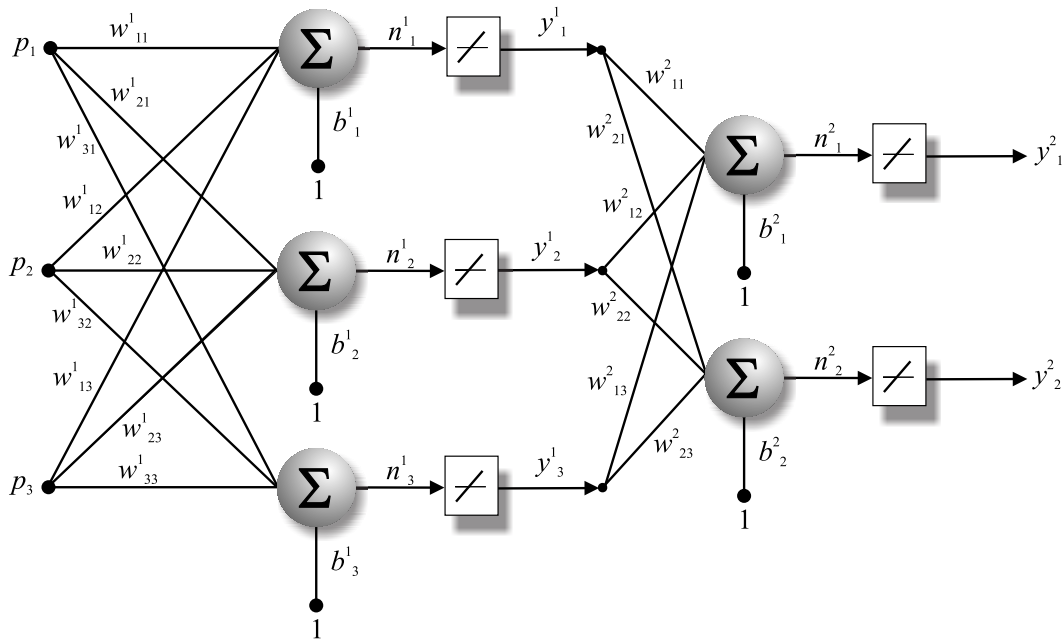
ปัญหาที่ไม่สามารถแบ่งแยกแบบเชิงเส้นได้ โครงสร้างพื้นฐานของเครือข่าย MADALINE ประกอบไปด้วย ADALINE หลายๆ ตัวเชื่อมต่อกันเป็นเครือข่ายไปข้างหน้าเพียงชุดเดียว โดยทั่วไปแล้ว เครือข่าย MADALINE พื้นฐานแบ่งได้ออกเป็น 2 แบบคือ MADALINE I และ MADALINE II

เครือข่าย MADALINE I ถูกนำเสนอเป็นครั้งแรกโดย Widrow [Widrow, 1962] และ Hoff [M. E. Hoff, 1962] ตัวอย่างของเครือข่าย MADALINE I ที่ประกอบไปด้วย ADALINE 3 ชุดมีแสดงในรูปที่ 11.5 ฟังก์ชันลอจิก เช่น OR หรือ AND จะทำการประมวลผลเอาต์พุตจากแต่ละ ADALINE และให้ค่าเอาต์พุตของเครือข่าย MADALINE I เครือข่าย MADALINE II มีโครงสร้างเป็นเครือข่ายหลายชั้นและสามารถมีได้หลายเอาต์พุต [Widrow and Winter, 1988, Widrow et al., 1987, Winter, 1989] รูปที่ 11.6 แสดงตัวอย่างของเครือข่าย MADALINE II ที่ประกอบไปด้วยเครือข่าย 2 ชั้น

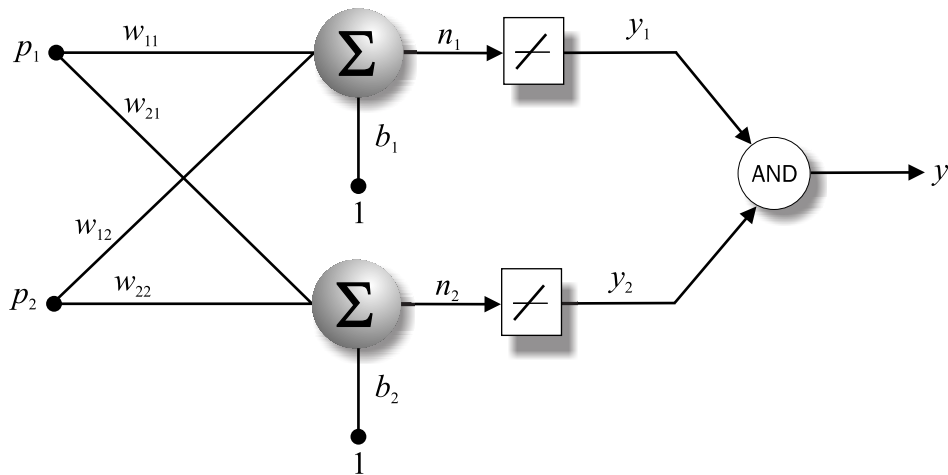
พิจารณาปัญหา XOR (exclusive-OR) ซึ่งสามารถแทนด้วยเวกเตอร์อินพุตและเวกเตอร์เป้าหมายดังนี้

$$\begin{aligned} \left\{ \mathbf{p}_1 = \begin{bmatrix} -1 \\ -1 \end{bmatrix}, t_1 = [-1] \right\} \\ \left\{ \mathbf{p}_2 = \begin{bmatrix} -1 \\ +1 \end{bmatrix}, t_2 = [+1] \right\} \\ \left\{ \mathbf{p}_3 = \begin{bmatrix} +1 \\ +1 \end{bmatrix}, t_3 = [+1] \right\} \\ \left\{ \mathbf{p}_4 = \begin{bmatrix} +1 \\ -1 \end{bmatrix}, t_4 = [-1] \right\} \end{aligned} \quad (11.43)$$

รูปที่ 11.7 แสดงเครือข่าย MADALINE แบบ 2 อินพุตและ 2 ADALINE เอาต์พุตจาก ADALINE แต่ละตัว



รูปที่ 11.6: ตัวอย่างเครือข่าย MADALINE II ที่ประกอบไปด้วยเครือข่าย 2 ชั้น (ชั้นละ 3 นิวรอนและ 2 นิวรอนตามลำดับ)



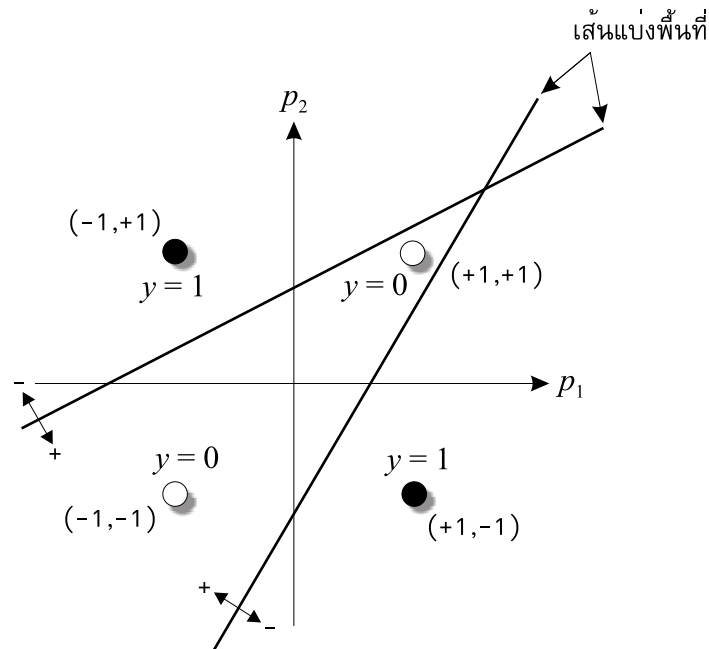
รูปที่ 11.7: เครือข่าย MADALINE I พร้อมฟังก์ชันลอจิก AND

จะป้อนให้กับฟังก์ชันลอจิก AND ที่ซึ่งให้ค่าเอาต์พุต y เป็นค่าเส้นแบ่งพื้นที่ -1 หรือ $+1$ พิจารณาเอาต์พุตของ ADALINE ทั้งสองมีค่าเป็น $n_1 = w_{11}p_1 + p_2w_{12} + b_1$ และ $n_2 = w_{21}p_1 + p_2w_{22} + b_2$ ตามลำดับ เมื่อพิจารณาเส้นแบ่งพื้นที่ที่กำหนดให้ $n_1 = 0$ และ $n_2 = 0$ จะได้เส้นแบ่งพื้นที่ 2 เส้นที่แบ่งพื้นที่ของอินพุต (p_1, p_2) (แต่ละเส้นแบ่งพื้นที่ออกเป็นสองส่วน) สมการของเส้นแบ่งพื้นที่ทั้งสองได้แก่

$$p_1w_{11} + p_2w_{12} + b_1 = 0 \quad (11.44)$$

$$p_1w_{21} + p_2w_{22} + b_2 = 0 \quad (11.45)$$

รูปที่ 11.8 แสดงตัวอย่างเส้นแบ่งพื้นที่ทั้งสอง ที่มีค่าน้ำหนักประสาทและค่าไบอัสที่เหมาะสม ทำให้สามารถแบ่งแยกรูปแบบอินพุตของปัญหา XOR ได้อย่างถูกต้อง



รูปที่ 11.8: เครือข่าย MADALINE I และเส้นแบ่งพื้นที่สำหรับปัญหา XOR

กฎพื้นฐานสำหรับฝึกสอนเครือข่าย MADALINE มี 2 แบบ [Widrow and Lehr, 1990] ดังรายละเอียดต่อไปนี้

- **กฎ MADALINE I (MADALINE RULE I หรือ MRI)** เป็นการปรับค่าน้ำหนักประสาทของเครือข่าย MADALINE I (เช่นเครือข่ายในรูปที่ 11.5) อัลกอริทึมจะทำการปรับค่าน้ำหนักประสาทเพื่อแก้ไขให้เอาต์พุต y ของเครือข่ายเป็นไปตามเป้าหมาย t ที่ต้องการ ค่าน้ำหนักประสาทเริ่มต้นจะได้รับการสุ่มและมีการกำหนดขนาดให้มีขนาดเล็ก กลยุทธ์ในการฝึกสอนนั้นมีอยู่หลายวิธี เช่นวิธีการแก้ไขสมบูรณ์ (absolute correction หรือการฝึกสอนแบบเร็ว) หรือวิธี α - LMS (หรือการฝึกสอนแบบช้า) แนวคิดหลักสำหรับการฝึกสอนเหล่านี้คือการปรับค่าน้ำหนักประสาทของนิวรอน j ที่ซึ่งค่าเอาต์พุตแบบเชิงเส้น n_j มีค่าใกล้เคียงกับค่าศูนย์มากที่สุด (นั่นคือ นิวรอนให้ผลตอบสนองที่ใกล้เคียงกับผลตอบสนองเป้าหมายมากที่สุด) เพราะฉะนั้นนิวรอนเหล่านี้ต้องการปรับน้ำหนักประสาทที่มีขนาดเล็กที่สุด เพื่อให้ได้การเปลี่ยนแปลงของผลตอบสนองเอาต์พุตที่ต้องการ โดยทั่วไปแล้ว MRI ใช้หลักการรบกวนที่น้อยที่สุด (minimal-disturbance principle) กล่าวคือในการแก้ไขเส้นแบ่งพื้นที่ ถ้าไม่มีความจำเป็น จะไม่มีการปรับนิวรอนใดๆ
- **กฎ MADALINE II (MADALINE RULE II หรือ MR II)** เป็นกฎที่ได้รับการพัฒนาต่อจาก MRI สำหรับฝึกสอนเครือข่าย MADALINE II ยกตัวอย่างเครือข่าย MADALINE II แบบ 2 ชั้นในรูปที่ 11.6 ซึ่งมีหลักการฝึกสอนคือทำการปรับค่าน้ำหนักประสาทของเครือข่ายทั้ง 2 ชั้น ค่าน้ำหนักประสาทเริ่มต้นจะถูกสุ่มค่าขึ้นและกำหนดให้มีขนาดเล็ก รูปแบบอินพุตจะถูกป้อนให้กับเครือข่ายแบบสุ่ม โดยมีจุดประสงค์หลักคือการทำให้ค่าความผิดพลาดเฉลี่ยแบบแฮมมิง (Hamming) ของเซตของรูปแบบอินพุตมีค่าน้อยที่สุด

กฎทั้งสองข้างต้นมีจุดอ่อนอยู่คือการลู่เข้าสู่ค่าที่เหมาะสมที่สุดเฉพาะถิ่น (local minima) สิ่งที่ขาดไปของกฎทั้งสองคือความสามารถในการแพร่กลับของค่าความผิดพลาด ให้เข้าไปในโครงสร้างของเครือข่าย เพื่อทำการปรับค่าน้ำหนักประสาทอย่างมีประสิทธิภาพได้ วิธีดังกล่าวเป็นกฎการเรียนรู้แบบแพร่กลับ ที่ซึ่งจะได้กล่าวถึงรายละเอียดต่อไป

ตัวอย่างต่อไปนี้เป็นการศึกษาการใช้งานเครือข่าย ADALINE พร้อมการฝึกสอนด้วยกฎการเรียนรู้แบบวิโดว์-ฮอฟฟ์

■ ตัวอย่างที่ 11.1 การฝึกสอนด้วยกฎการเรียนรู้แบบวิโดว์-ฮอฟฟ์

เราจะทำการทดลองการฝึกสอนด้วยกฎการเรียนรู้แบบวิโดว์-ฮอฟฟ์ โดยใช้เครือข่าย ADALINE ในการตัดแยกอินพุตเวกเตอร์ขนาด 2×1 ทั้งหมด 4 กลุ่มดังนี้ (เวกเตอร์อินพุตกลุ่มเดียวกันจะมีเวกเตอร์เป้าหมายเดียวกัน)

- กลุ่มที่ 1 $\left\{ \mathbf{p}_1 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \mathbf{p}_2 = \begin{bmatrix} 2 \\ 2 \end{bmatrix} \right\}$
- กลุ่มที่ 2 $\left\{ \mathbf{p}_3 = \begin{bmatrix} 1 \\ -1 \end{bmatrix}, \mathbf{p}_4 = \begin{bmatrix} 1 \\ -2 \end{bmatrix} \right\}$
- กลุ่มที่ 3 $\left\{ \mathbf{p}_5 = \begin{bmatrix} -1 \\ 1 \end{bmatrix}, \mathbf{p}_6 = \begin{bmatrix} -2 \\ 1 \end{bmatrix} \right\}$
- กลุ่มที่ 4 $\left\{ \mathbf{p}_7 = \begin{bmatrix} -1 \\ -1 \end{bmatrix}, \mathbf{p}_8 = \begin{bmatrix} -2 \\ -2 \end{bmatrix} \right\}$

เวกเตอร์เป้าหมายคือ

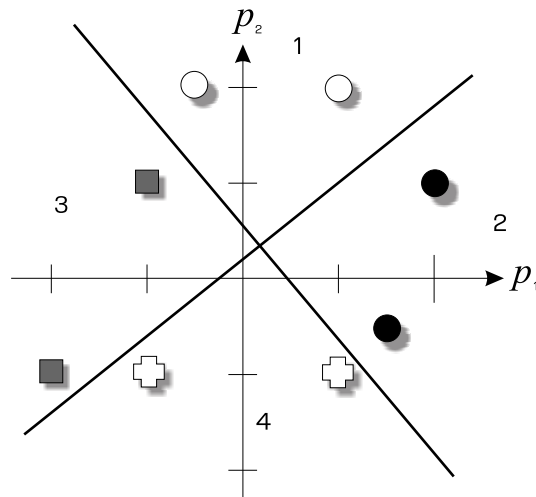
- กลุ่มที่ 1 $\left\{ \mathbf{t}_1 = \begin{bmatrix} -1 \\ -1 \end{bmatrix}, \mathbf{t}_2 = \begin{bmatrix} -1 \\ -1 \end{bmatrix} \right\}$
- กลุ่มที่ 2 $\left\{ \mathbf{t}_3 = \begin{bmatrix} -1 \\ 1 \end{bmatrix}, \mathbf{t}_4 = \begin{bmatrix} -1 \\ 1 \end{bmatrix} \right\}$
- กลุ่มที่ 3 $\left\{ \mathbf{t}_5 = \begin{bmatrix} 1 \\ -1 \end{bmatrix}, \mathbf{t}_6 = \begin{bmatrix} 1 \\ -1 \end{bmatrix} \right\}$
- กลุ่มที่ 4 $\left\{ \mathbf{t}_7 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \mathbf{t}_8 = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \right\}$

รูปที่ 11.9 แสดงอินพุตและเป้าหมายของระบบ เครือข่ายที่ใช้สำหรับระบบนี้แสดงในรูปที่ 11.10 ซึ่งเป็นเครือข่าย ADALINE ขนาด 2 อินพุต 2 นิวรอนและ 2 เอาต์พุต กำหนดค่าเริ่มต้นของ $\mathbf{W} = \mathbf{0}$ และ $\mathbf{b} = \mathbf{0}$ โดยการใช้ค่าคงที่การเรียนรู้ $\alpha = 0.01$ การฝึกสอนจะดำเนินไปเรื่อยๆ จนกว่าความเปลี่ยนแปลงของแต่ละองค์ประกอบในเมตริกซ์น้ำหนักประสาทของแต่ละรอบการคำนวณ มีความแตกต่างกันน้อยกว่าค่าที่ตั้งไว้ ในที่นี้กำหนดให้เป็น 0.005 ดังนั้นการฝึกสอนจะทำการวนซ้ำอินพุตและเป้าหมายไปเรื่อยๆ จนกว่าจะถึงเงื่อนไขที่กำหนด ตัวอย่างค่าเชิงตัวเลขและรายละเอียดของขั้นตอนการฝึกสอนมีดังนี้

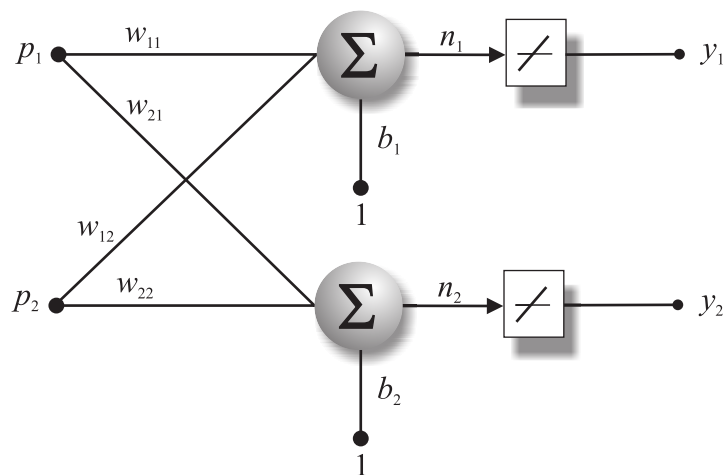
• รอบที่ 1

– เอาต์พุตของเครือข่าย

$$\begin{aligned}
 \mathbf{y}(0) &= \text{purelin}(\mathbf{W}(0)\mathbf{p}(0) + \mathbf{b}(0)) \\
 &= \text{purelin}\left(\begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix}\right) \\
 &= \begin{bmatrix} 0 \\ 0 \end{bmatrix}
 \end{aligned}$$



รูปที่ 11.9: เวกเตอร์อินพุตและเวกเตอร์เป้าหมายในระนาบ 2 มิติ



รูปที่ 11.10: เครือข่าย ADALINE ขนาด 2 นิวรอนสำหรับคัดแยกเวกเตอร์อินพุตขนาด 2×1 ด้วยเอาต์พุตขนาด 2×1

– ค่าความผิดพลาดระหว่างเอาต์พุตและเป้าหมาย

$$\begin{aligned} e(0) &= \mathbf{t}(0) - \mathbf{y}(0) \\ &= \begin{bmatrix} -1 \\ -1 \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \end{bmatrix} \\ &= \begin{bmatrix} -1 \\ -1 \end{bmatrix} \end{aligned}$$

– คำนวณน้ำหนักประสาท

$$\begin{aligned} \mathbf{W}(1) &= \mathbf{W}(0) + 2\alpha e(0) \mathbf{p}^T(0) \\ &= \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} + 2(0.01) \begin{bmatrix} -1 \\ -1 \end{bmatrix} \begin{bmatrix} 1 & 1 \end{bmatrix} \end{aligned}$$

$$= \begin{bmatrix} -0.02 & -0.02 \\ -0.02 & -0.02 \end{bmatrix}$$

– ค่าไบอัส

$$\begin{aligned} \mathbf{b}(1) &= \mathbf{b}(0) + 2\alpha\mathbf{e}(0) \\ &= \begin{bmatrix} 0 \\ 0 \end{bmatrix} + 2(0.01) \begin{bmatrix} -1 \\ -1 \end{bmatrix} \\ &= \begin{bmatrix} -0.02 \\ -0.02 \end{bmatrix} \end{aligned}$$

• รอบที่ 2

– เอาต์พุตของเครือข่าย

$$\begin{aligned} \mathbf{y}(1) &= \text{purelin}(\mathbf{W}(1)\mathbf{p}(1) + \mathbf{b}(1)) \\ &= \text{purelin} \left(\begin{bmatrix} -0.02 & -0.02 \\ -0.02 & -0.02 \end{bmatrix} \begin{bmatrix} 2 \\ 2 \end{bmatrix} + \begin{bmatrix} -0.02 \\ -0.02 \end{bmatrix} \right) \\ &= \begin{bmatrix} -0.1 \\ -0.1 \end{bmatrix} \end{aligned}$$

– ค่าความผิดพลาดระหว่างเอาต์พุตและเป้าหมาย

$$\begin{aligned} \mathbf{e}(1) &= \mathbf{t}(1) - \mathbf{y}(1) \\ &= \begin{bmatrix} -1 \\ -1 \end{bmatrix} - \begin{bmatrix} -0.1 \\ -0.1 \end{bmatrix} \\ &= \begin{bmatrix} -0.9 \\ -0.9 \end{bmatrix} \end{aligned}$$

– ค่าน้ำหนักประสาท

$$\begin{aligned} \mathbf{W}(2) &= \mathbf{W}(1) + 2\alpha\mathbf{e}(1)\mathbf{p}^T(1) \\ &= \begin{bmatrix} -0.02 & -0.02 \\ -0.02 & -0.02 \end{bmatrix} + 2(0.01) \begin{bmatrix} -0.9 \\ -0.9 \end{bmatrix} \begin{bmatrix} 2 & 2 \end{bmatrix} \\ &= \begin{bmatrix} -0.056 & -0.056 \\ -0.056 & -0.056 \end{bmatrix} \end{aligned}$$

– ค่าไบอัส

$$\begin{aligned} \mathbf{b}(2) &= \mathbf{b}(1) + 2\alpha\mathbf{e}(1) \\ &= \begin{bmatrix} -0.02 \\ -0.02 \end{bmatrix} + 2(0.01) \begin{bmatrix} -0.9 \\ -0.9 \end{bmatrix} \\ &= \begin{bmatrix} -0.038 \\ -0.038 \end{bmatrix} \end{aligned}$$

- รอบที่ 26 จะได้ค่าน้ำหนักประสาทและไบอัสดังนี้

$$\mathbf{W}(26) = \begin{bmatrix} -0.4367 & -0.0108 \\ -0.0114 & -0.4396 \end{bmatrix}$$

$$\mathbf{b}(26) = \begin{bmatrix} -0.0348 \\ -0.0245 \end{bmatrix}$$

โดยที่ค่าแต่ละองค์ประกอบของ $\mathbf{W}(26)$ มีการเปลี่ยนแปลงไปจาก $\mathbf{W}(26)$ น้อยกว่า 0.005 การฝึกสอนจึงสิ้นสุดที่ขั้นที่ 26

พิจารณาผลการทดสอบค่าเมตริกซ์น้ำหนักประสาทและไบอัสที่ได้ดังนี้

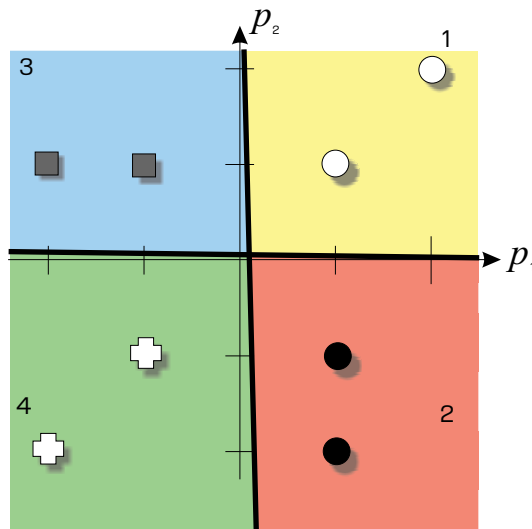
- กลุ่มที่ 1 $\left\{ \mathbf{p}_1 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \mathbf{y}_1 = \begin{bmatrix} -0.4823 \\ -0.4744 \end{bmatrix}, \mathbf{p}_2 = \begin{bmatrix} 2 \\ 2 \end{bmatrix}, \mathbf{y}_2 = \begin{bmatrix} -0.9298 \\ -0.9265 \end{bmatrix} \right\}$
- กลุ่มที่ 2 $\left\{ \mathbf{p}_3 = \begin{bmatrix} 1 \\ -1 \end{bmatrix}, \mathbf{y}_3 = \begin{bmatrix} -0.4607 \\ 0.4036 \end{bmatrix}, \mathbf{p}_4 = \begin{bmatrix} 1 \\ -2 \end{bmatrix}, \mathbf{y}_4 = \begin{bmatrix} -0.4499 \\ 0.8432 \end{bmatrix} \right\}$
- กลุ่มที่ 3 $\left\{ \mathbf{p}_5 = \begin{bmatrix} -1 \\ 1 \end{bmatrix}, \mathbf{y}_5 = \begin{bmatrix} 0.3912 \\ -0.4526 \end{bmatrix}, \mathbf{p}_6 = \begin{bmatrix} -2 \\ 1 \end{bmatrix}, \mathbf{y}_6 = \begin{bmatrix} 0.8279 \\ -0.4412 \end{bmatrix} \right\}$
- กลุ่มที่ 4 $\left\{ \mathbf{p}_7 = \begin{bmatrix} -1 \\ -1 \end{bmatrix}, \mathbf{y}_7 = \begin{bmatrix} 0.4128 \\ 0.4265 \end{bmatrix}, \mathbf{p}_8 = \begin{bmatrix} -2 \\ -2 \end{bmatrix}, \mathbf{y}_8 = \begin{bmatrix} 0.8603 \\ 0.8775 \end{bmatrix} \right\}$

รูปที่ 11.11 แสดงผลของเส้นแบ่งพื้นที่ที่ทำการฝึกสอนได้ข้างต้น ADALINE ได้ทำการแบ่งพื้นที่ของอินพุตออกเป็น 4 ส่วน โดยมีเส้นแบ่งพื้นที่ที่อยู่ห่างจากกลุ่มของข้อมูลทั้ง 4 มากที่สุด ซึ่งเป็นผลจากอัลกอริทึม LM-S ในทางปฏิบัติ ค่าพารามิเตอร์ของเครือข่ายมีผลต่อเอาต์พุตที่ฝึกสอนได้แตกต่างกันไป เช่นค่าคงที่การเรียนรู้เป็นตัวกำหนดเครือข่ายให้สามารถเรียนรู้ในอัตราความเร็วที่แตกต่างกัน ถ้าค่าคงที่การเรียนรู้มีค่ามาก ค่าเมตริกซ์น้ำหนักประสาทจะมีความหยาบและให้ค่าความผิดพลาดสูง แต่เครือข่ายสามารถเรียนรู้ได้เร็วภายในไม่กี่รอบ ในขณะที่ถ้าค่าคงที่การเรียนรู้มีค่าน้อย ค่าเมตริกซ์น้ำหนักประสาทจะมีความละเอียดเพิ่มขึ้น ค่าความผิดพลาดจะน้อยลง แต่เครือข่ายเรียนรู้ด้วยเวลาที่นานกว่า การลู่เข้าของการฝึกสอนแสดงในรูปที่ 11.12 (ในรูปของค่าความผิดพลาดของแต่ละองค์ประกอบใน \mathbf{y} โดย $e_1 = t_1 - y_1$ และ $e_2 = t_2 - y_2$ ตามลำดับ) ■

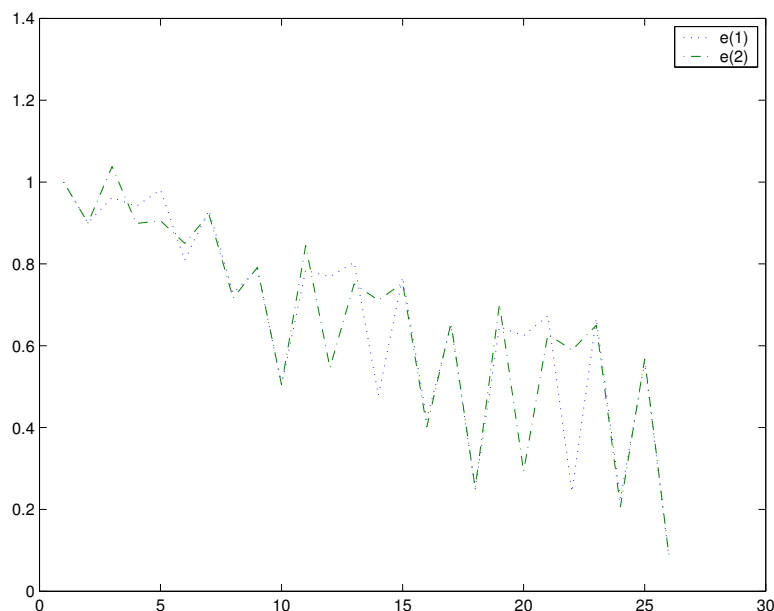
11.5 การประยุกต์ใช้งานการเรียนรู้แบบวิโดว์-ฮอฟฟ์: การตัดออกสัญญาณรบกวนเชิงปรับตัว

ในหัวข้อนี้จะได้กล่าวถึงการประยุกต์ใช้งานเครือข่าย ADALINE ในงานการตัดออกสัญญาณรบกวนเชิงปรับตัว (adaptive noise cancellation) [Hagan et al., 1995] การทำงานของระบบดังกล่าวได้นำเอาเครือข่าย ADALINE มาประยุกต์ใช้เป็นตัวกรองเชิงปรับตัว (adaptive filter) ที่สามารถปรับพารามิเตอร์ภายในระบบให้สอดคล้องกับสิ่งที่กำลังตรวจจับอยู่ (ในที่นี้ก็คือสัญญาณรบกวน) เนื้อหาในหัวข้อนี้ใช้การจำลองสถานการณ์ของระบบด้วยโปรแกรม MATLAB พร้อมกล่องเครื่องมือ Neural Network Toolbox ซึ่งผู้อ่านสามารถศึกษารายละเอียดเพิ่มเติมได้จากคู่มือ Neural Network Toolbox ในโปรแกรม MATLAB

ในปัจจุบันระบบตัดออกสัญญาณรบกวนได้ถูกนำมาประยุกต์ใช้งานจริงในอุปกรณ์อิเล็กทรอนิกส์ต่างๆ ไปที่เกี่ยวข้อง ไม่ว่าจะเป็นหูฟังสำหรับฟังเพลง การกำจัดสัญญาณรบกวนในห้องโดยสารของยานพาหนะ ฯลฯ รูปที่ 11.13 แสดงแผนผังระบบการตัดออกสัญญาณรบกวนทั่วไป

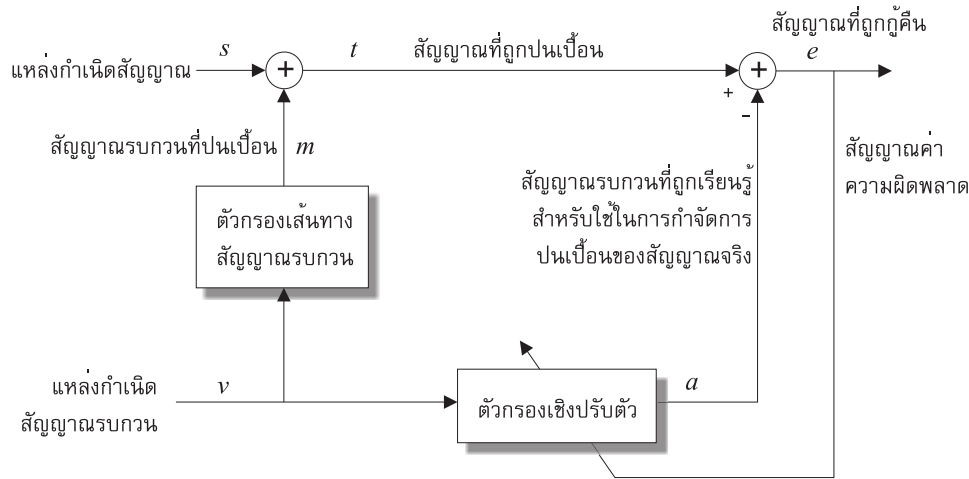


รูปที่ 11.11: ผลการฝึกสอน ADALINE ในการคัดแยกอินพุต 4 กลุ่ม



รูปที่ 11.12: การลู่เข้าของค่าความผิดพลาดด้วยค่าคงที่การเรียนรู้เท่ากับ 0.01 เส้นประของ $e(1)$ แทนค่าความผิดพลาดของนิวรอนตัวที่ 1 คือ $e_1 = t_1 - y_1$ และเส้นประของ $e(2)$ แทนค่าความผิดพลาดของนิวรอนตัวที่ 2 คือ $e_2 = t_2 - y_2$

สัญญาณรบกวนถูกส่งออกมาจากแหล่งกำเนิดสัญญาณรบกวน พร้อมกับถูกป้อนให้กับตัวกรองเชิงปรับตัว พารามิเตอร์ในตัวกรองเชิงปรับตัวจะถูกปรับเพื่อให้ระบบมีสัญญาณค่าความผิดพลาด (error signal) หรือ e มีค่าน้อยที่สุด ในระบบที่ใช้งานจริง สัญญาณ s อาจจะเป็นสัญญาณ EEG จากคนไข้ที่ได้มาจากเครื่องวัด EEG โดยมีสัญญาณรบกวน v จากแหล่งจ่ายไฟ (สัญญาณความถี่ 50 Hz) หรือในกรณีของห้องโดยสารของยานพาหนะไม่ว่าจะเป็นเครื่องบินหรือรถยนต์โดยสาร สัญญาณ s อาจจะเป็นสัญญาณเสียงใดๆ ภายในห้องโดยสารที่ผู้โดยสารต้องการฟัง โดยมีสัญญาณรบกวน v เป็นเสียงจากเครื่องยนต์ ตัวกรองเชิงปรับตัวจะทำการเลียนแบบสัญญาณ t ซึ่งเป็นสัญญาณ s ที่ถูกปนเปื้อนด้วยสัญญาณรบกวน m (ที่ผ่านตัวกรองเส้นทางสัญญาณรบกวน นั่นคือไม่



รูปที่ 11.13: ระบบการตัดออกสัญญาณรบกวนด้วยตัวกรองเชิงปรับตัว

ได้เป็นสัญญาณรบกวนจากแหล่งกำเนิดโดยตรง) ข้อมูลที่ตัวกรองเชิงปรับตัวสามารถใช้งานได้มีเพียงสัญญาณ v จากแหล่งกำเนิดสัญญาณ จะได้ว่าตัวกรองเชิงปรับตัวจะทำการเลียนแบบส่วนของสัญญาณ t ที่ซึ่งสัมพันธ์กับสัญญาณรบกวน v ผ่านสัญญาณ m นั่นเอง จริงๆ แล้วตัวกรองเชิงปรับตัวจะเลียนแบบพฤติกรรมของตัวกรองเส้นทางสัญญาณรบกวน เพื่อที่จะทำให้สัญญาณเอาต์พุต a จากตัวกรองเชิงปรับตัว ใกล้เคียงกับสัญญาณ m ให้มากที่สุด ดังนั้นสัญญาณค่าความผิดพลาด e จะมีค่าใกล้เคียงกับสัญญาณ s ที่ต้องการมากที่สุดนั่นเอง (เพราะสัญญาณรบกวนถูกตัวกรองเชิงปรับตัวเรียนรู้และลบออกไปจากระบบ)

พิจารณาตัวอย่างสัญญาณรบกวน v ที่เกิดจากระบบจ่ายไฟ สัญญาณรบกวนดังกล่าวจะอยู่ในรูปคลื่นไซน์ขนาดความถี่เท่ากับ 50 Hz เครื่องข่าย ADALINE พร้อมตัวหน่วง (tapped delay line) ดังแสดงในรูปที่ 11.14 จะถูกใช้เป็นตัวกรองเชิงปรับตัว จำนวนของตัวหน่วงเป็นการกำหนดจำนวนของค่าย้อนหลังของอินพุต ในที่นี้เราต้องการใช้เพียงค่าอินพุต ณ เวลาปัจจุบันและค่าอินพุต ณ เวลาก่อนหน้าหนึ่งค่า ดังนั้นจำนวนของตัวหน่วงจะเท่ากับหนึ่ง โครงสร้างตัวกรองเชิงปรับตัวด้วย ADALINE แบบ 2 อินพุตนี้สามารถเรียนรู้ (หรือเลียนแบบ) สัญญาณที่ถูกลดทอนและเลื่อนเฟสได้เป็นอย่างดี จึงเพียงพอต่อการประยุกต์ใช้งานในตัวอย่างนี้ จำนวนของตัวหน่วงที่เพิ่มขึ้นจะทำให้เครือข่ายหรือตัวกรองเชิงปรับตัว มีข้อมูลของอินพุตย้อนหลังเพิ่มมากขึ้น การเรียนรู้พฤติกรรมของสัญญาณอินพุตอาจจะดีขึ้น แต่ก็มีผลทำให้เครือข่ายมีความซับซ้อนมากยิ่งขึ้นด้วย ในที่นี้เอาต์พุตของเครือข่ายหรือตัวกรองขนาด 2 อินพุต (1 ตัวหน่วง) ที่ไม่พิจารณาไบอัสมีค่าเท่ากับ (ดูรูปที่ 11.15 ประกอบ)

$$a(k) = w_{11}v(k) + w_{12}v(k-1) \quad (11.46)$$

พิจารณาเมตริกซ์ออตสัมพันธ์ $\mathbf{R} = E[\mathbf{z}\mathbf{z}^T]$ และเวกเตอร์สหสัมพันธ์ $\mathbf{h} = E[t\mathbf{z}]$ ในกรณีของตัวกรองเชิงปรับตัวนี้ อินพุตประกอบไปด้วยค่าจากสัญญาณรบกวน ณ เวลาปัจจุบัน $v(k)$ และ ณ เวลาย้อนหลังหนึ่งหน่วยเวลา $v(k-1)$ นั่นคือ

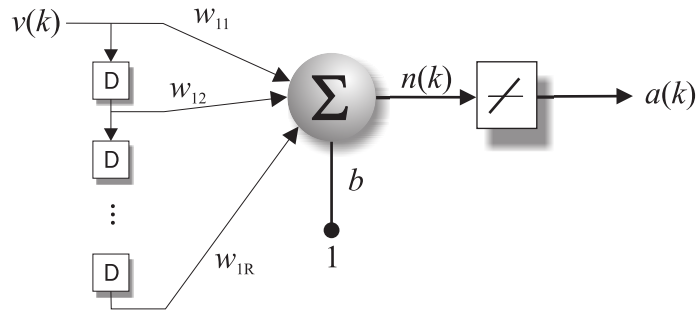
$$\mathbf{z}(k) = \begin{bmatrix} v(k) \\ v(k-1) \end{bmatrix} \quad (11.47)$$

ในที่นี้ค่าเป้าหมาย t คือผลรวมของสัญญาณ s ณ เวลาปัจจุบันและสัญญาณรบกวน m จะได้ว่า

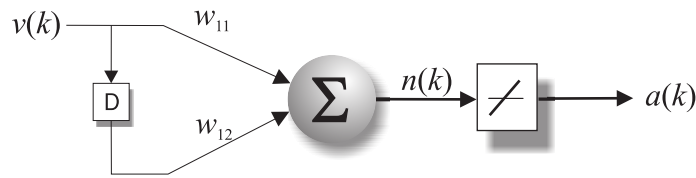
$$t(k) = s(k) + m(k) \quad (11.48)$$

ดังนั้นจะได้เมตริกซ์ \mathbf{R} คือ

$$\mathbf{R} = \begin{bmatrix} E[v^2(k)] & E[v(k)v(k-1)] \\ E[v(k-1)v(k)] & E[v^2(k-1)] \end{bmatrix} \quad (11.49)$$



รูปที่ 11.14: เครือข่าย ADALINE สำหรับตัวกรองเชิงปรับตัว



รูปที่ 11.15: เครือข่าย ADALINE สำหรับตัวกรองเชิงปรับตัวขนาด 2 อินพุตแบบไม่มีไบอัส

และเวกเตอร์ \mathbf{h} คือ

$$\mathbf{h} = \begin{bmatrix} E[(s(k) + m(k))v(k)] \\ E[(s(k) + m(k))v(k-1)] \end{bmatrix} \quad (11.50)$$

พิจารณาสัญญาณ s เป็นสัญญาณสุ่มไวต์ (white random signal) ที่มีการกระจายแบบสม่ำเสมอในช่วง $[-0.2, 0.2]$ แหล่งกำเนิดสัญญาณรบกวนเป็นสัญญาณรูปคลื่นไซน์ขนาด 50 Hz ดังความสัมพันธ์ต่อไปนี้

$$v(k) = 1.2 \sin\left(\frac{2\pi k}{3}\right) \quad (11.51)$$

สัญญาณรบกวน m ที่ผ่านตัวกรองเส้นทางและไปปนเปื้อนกับสัญญาณ s ได้จากการลดทอนสัญญาณจากแหล่งกำเนิด v ด้วยขนาด 10 เท่าพร้อมทั้งเลื่อนเฟสไป $\pi/2$ นั่นคือ

$$m(k) = 0.12 \sin\left(\frac{2\pi k}{3} + \frac{\pi}{2}\right) \quad (11.52)$$

ทำการคำนวณหาค่าต่างๆ ของเมตริกซ์ \mathbf{R} ดังนี้

$$E[v^2(k)] = (1.2)^2 \frac{1}{3} \sum_{k=1}^3 \left(\sin\left(\frac{2\pi k}{3}\right) \right)^2 = 0.72 \quad (11.53)$$

$$E[v^2(k-1)] = E[v^2(k)] = 0.72 \quad (11.54)$$

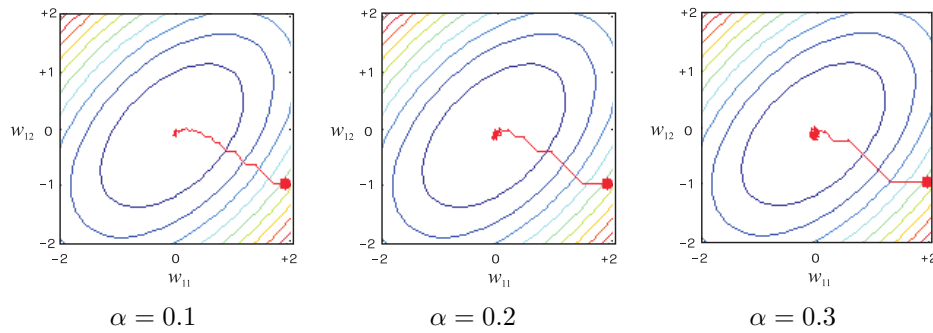
$$E[v(k)v(k-1)] = \frac{1}{3} \sum_{k=1}^3 \left(1.2 \sin \frac{2\pi k}{3} \right) \left(1.2 \sin \frac{2\pi(k-1)}{3} \right) = -0.36 \quad (11.55)$$

ดังนั้นจะได้

$$\mathbf{R} = \begin{bmatrix} 0.72 & -0.36 \\ -0.36 & 0.72 \end{bmatrix} \quad (11.56)$$

ในทำนองเดียวกัน เราสามารถคำนวณหาเทอมต่างๆ ในเวกเตอร์ \mathbf{h} ได้โดยเริ่มต้นที่เทอม

$$E[(s(k) + m(k))v(k)] = E[s(k)v(k)] + E[m(k)v(k)] \quad (11.57)$$



รูปที่ 11.16: การลู่เข้าสู่คำตอบของอัลกอริทึม LMS ที่อัตราการเรียนรู้ค่าต่างๆ

เทอมแรกทางขวามือมีค่าเท่ากับศูนย์ เนื่องจาก $s(k)$ และ $v(k)$ เป็นอิสระต่อกันและมีค่าเฉลี่ยเป็นศูนย์ (zero mean) ในขณะที่เทอมที่สองมีค่าเท่ากับศูนย์เหมือนกัน โดยพิจารณาจากความสัมพันธ์เชิงตัวเลขต่อไปนี้

$$E[m(k)v(k)] = \frac{1}{3} \sum_{k=1}^3 \left(0.12 \sin \left(\frac{2\pi k}{3} + \frac{\pi}{2} \right) \right) \left(0.12 \sin \frac{2\pi k}{3} \right) = 0 \quad (11.58)$$

ดังนั้นเทอมแรกใน \mathbf{h} มีค่าเท่ากับศูนย์
พิจารณาเทอมที่สองใน \mathbf{h} ดังนี้

$$E[(s(k) + m(k))v(k-1)] = E[s(k)v(k-1)] + E[m(k)v(k-1)] \quad (11.59)$$

จะเห็นว่าเทอมแรกทางขวามือมีค่าเป็นศูนย์ เนื่องจาก $s(k)$ และ $v(k-1)$ เป็นอิสระต่อกันและมีค่าเฉลี่ยเป็นศูนย์
พิจารณาเทอมที่สองพร้อมค่าเชิงตัวเลขต่อไปนี้

$$\begin{aligned} E[m(k)v(k-1)] &= \frac{1}{3} \sum_{k=1}^3 \left(0.12 \sin \left(\frac{2\pi k}{3} + \frac{\pi}{2} \right) \right) \left(0.12 \sin \frac{2\pi(k-1)}{3} \right) \\ &= -0.0624 \end{aligned} \quad (11.60)$$

ดังนั้นจะได้ว่า

$$\mathbf{h} = \begin{bmatrix} 0 \\ -0.0624 \end{bmatrix} \quad (11.61)$$

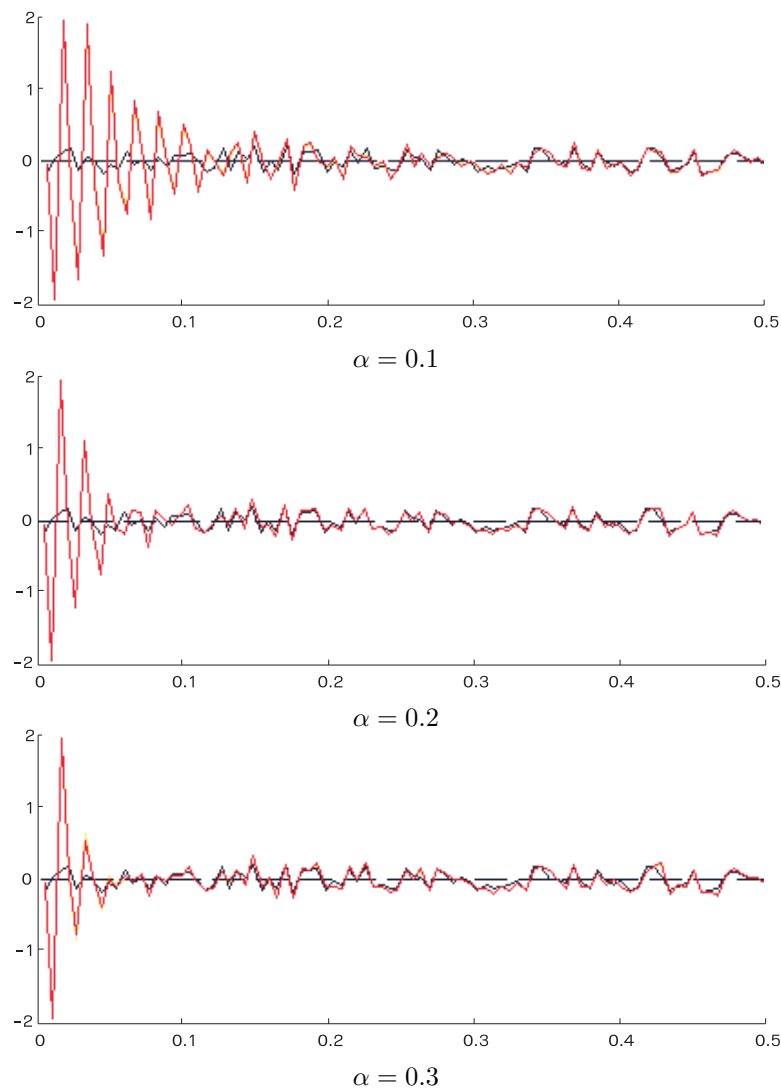
ดังนั้นค่าความผิดพลาดกำลังสองเฉลี่ยที่น้อยที่สุดสำหรับน้ำหนักประสาของเครือข่ายจากสมการที่ 11.15 คือ

$$\mathbf{x}^* = \mathbf{R}^{-1} \mathbf{h} \quad (11.62)$$

$$= \begin{bmatrix} 0.72 & -0.36 \\ -0.36 & 0.72 \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ -0.0624 \end{bmatrix} \quad (11.63)$$

$$= \begin{bmatrix} -0.0578 \\ -0.1156 \end{bmatrix} \quad (11.64)$$

รูปที่ 11.16 แสดงการจำลองสถานการณ์ของเครือข่ายด้วยการใช้อัลกอริทึม LMS ในการคำนวณหาน้ำหนักประสาด้วยอัตราการเรียนรู้ α ค่าต่างๆ โดยมีค่าน้ำหนักประสาเริ่มต้น $w_{11} = 2$ และ $w_{12} = -1$ ตามลำดับ สังเกตว่าค่าอัตราการเรียนรู้ที่น้อยจะทำให้การลู่เข้าเป็นไปอย่างเรียบ (smooth) กว่า แต่ก็ใช้เวลานานกว่าค่าอัตราการเรียนรู้ที่มากกว่า อย่างไรก็ตาม อัตราการเรียนรู้ที่มากเกินไปจะทำให้การเรียนรู้ไม่ลู่เข้าได้



รูปที่ 11.17: สัญญาณต้นฉบับและสัญญาณที่ระบบกู้คืนได้สำหรับอัตราการเรียนรู้ค่าต่างๆ เส้นกราฟสีเข้มแสดงสัญญาณต้นฉบับ ในขณะที่เส้นกราฟสีอ่อนแสดงสัญญาณที่ระบบกู้คืนได้ สังเกตว่าระบบต้องใช้เวลาช่วงหนึ่งในตอนเริ่มต้นก่อนที่จะกู้คืนสัญญาณได้ ดังนั้นในช่วงเริ่มต้นนี้สัญญาณที่กู้คืนได้จะมีค่าความผิดพลาดสูงอยู่

พิจารณาผลการตัดออกสัญญาณรบกวนในตัวอย่างนี้ รูปที่ 11.17 แสดงสัญญาณต้นฉบับ s และสัญญาณที่ระบบสามารถกู้คืนจากการกำจัดสัญญาณรบกวน m ออกไปจากสัญญาณต้นฉบับ จะเห็นได้ชัดเจนว่าด้วยค่าอัตราการเรียนรู้ที่แตกต่างกัน ระบบจะใช้เวลาในการเริ่มกู้คืนสัญญาณที่ถูกต้องได้แตกต่างกันด้วย ในกรณี $\alpha = 0.1$ ระบบใช้เวลาประมาณมากกว่า 0.1 วินาทีก่อนที่จะสามารถเริ่มกู้คืนสัญญาณต้นฉบับได้ ส่วนในกรณี $\alpha = 0.2$ และ 0.3 ระบบใช้เวลาประมาณ 0.1 และ 0.05 วินาทีตามลำดับก่อนที่จะสามารถเริ่มกู้คืนสัญญาณต้นฉบับได้ ตัวอย่างในหัวข้อนี้แสดงให้เห็นถึงความสามารถในการเรียนรู้ของเครือข่าย ADALINE ในการประยุกต์ใช้เป็นตัวกรองเชิงปรับตัวได้อย่างมีประสิทธิภาพ

ระบบกำจัดสัญญาณรบกวนเชิงปรับตัวนี้ ถือเป็นการประยุกต์ใช้งานของเครือข่ายวิโดว์-ฮอฟฟ์ที่ได้รับความสนใจอย่างต่อเนื่อง แม้ว่าตัวเครือข่ายและอัลกอริทึมการเรียนรู้จะได้ถูกนำเสนอมาเป็นเวลานานแล้วก็ตามที่ประสิทธิภาพของเครือข่ายวิโดว์-ฮอฟฟ์และอัลกอริทึมการฝึกสอนเครือข่าย ยังคงเป็นที่ยอมรับว่านำไปใช้งานได้

จริง

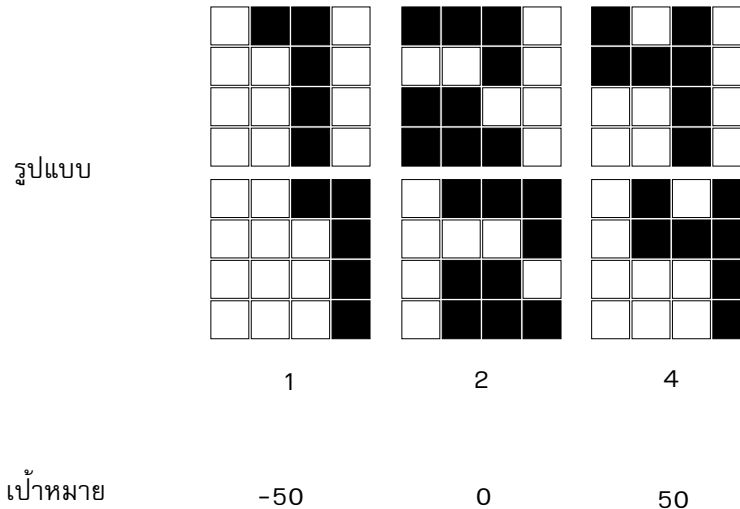
11.6 สรุป

ในบทนี้ได้นำเสนอรายละเอียดของเครือข่าย ADALINE (และ MADALINE) พร้อมทั้งกฎการเรียนรู้แบบ LMS เครือข่าย ADALINE เป็นเครือข่ายที่มีโครงสร้างคล้ายคลึงกับเครือข่ายเพอร์เซพตรอน แตกต่างกันเพียงฟังก์ชันถ่ายโอนที่เป็นแบบเชิงเส้นเท่านั้น เครือข่าย ADALINE เองยังมีข้อจำกัดเดียวกันกับเครือข่ายเพอร์เซพตรอน คือไม่สามารถใช้แก้ปัญหาที่แบ่งแยกได้แบบไม่เป็นเชิงเส้น อย่างไรก็ตาม กฎการเรียนรู้แบบ LMS นั้นมีประสิทธิภาพเหนือกว่ากฎการเรียนรู้ของเพอร์เซพตรอน เนื่องจากตัวอัลกอริทึมมีการหาค่าน้อยที่สุดของค่าความผิดพลาดกำลังสองเฉลี่ยของเครือข่าย ดังนั้นเส้นแบ่งพื้นที่ที่ได้จากกฎการเรียนรู้แบบ LMS จะทนทานต่ออินพุตที่ไม่แน่นอน แม้ว่าเครือข่าย ADALINE และอัลกอริทึม LMS ได้ถูกนำเสนอไว้กว่าหลายสิบปี การนำไปประยุกต์ใช้งานจริงยังคงมีอยู่จนกระทั่งในปัจจุบัน โดยเฉพาะงานทางด้านตัวกรองเชิงปรับตัว นอกไปจากนั้นแล้ว อัลกอริทึม LMS ยังเป็นจุดเริ่มต้นที่นำไปสู่การเรียนรู้ที่ได้รับความนิยมและทรงประสิทธิภาพมากที่สุด นั่นก็คือการเรียนรู้แบบแพร่กลับ ดังจะได้กล่าวถึงรายละเอียดในบทถัดไป



โจทย์คำถาม

- 11.1. จงออกแบบระบบ ADALINE สำหรับจดจำรูปแบบ (pattern recognition) โดยรูปแบบที่จะทำการจดจำมีทั้งหมด 6 ภาพ ดังแสดงในรูปที่ 11.18 รูปแบบทั้ง 6 ภาพใช้แทนตัวเลข 1 2 และ 4 โดยแต่ละตัวเลขมีภาพรูปแบบอยู่ 2 ภาพ ซึ่งแตกต่างกันที่การเลื่อนในแกนแนวนอนเท่านั้น เป้าหมายที่ใช้แทนรูปแบบตัวเลขคือ -50 0 และ 50 สำหรับรูปแบบเลข 1 2 และ 4 ตามลำดับ (หมายความว่าถ้าระบบสามารถจดจำรูปแบบเลข 1 ได้ เอาต์พุตของระบบควรจะเป็นลบและมีค่าใกล้เคียง -50 ฯลฯ) จุดสีดำแต่ละจุดในภาพรูปแบบจะแทนด้วย

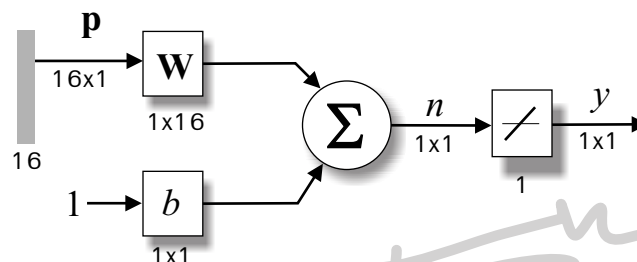


รูปที่ 11.18: รูปแบบตัวเลข 3 แบบที่ต้องการจดจำพร้อมทั้งค่าเป้าหมายที่ต้องการ

ค่า +1 ในขณะที่จุดสีขาวแทนด้วยค่า -1 ก่อนที่จะทำการฝึกสอนให้กับเครือข่าย ADALINE ได้จะต้องทำการแปลงภาพรูปแบบให้อยู่ในรูปของเวกเตอร์ขนาด 16 หลัก โดยนำเอาข้อมูลแต่ละแถวของภาพรูปแบบมาวางเรียงกันเป็นแถวยาว ยกตัวอย่างเช่นภาพรูปแบบตัวเลข 1 จะให้ค่าเวกเตอร์แถวดังนี้

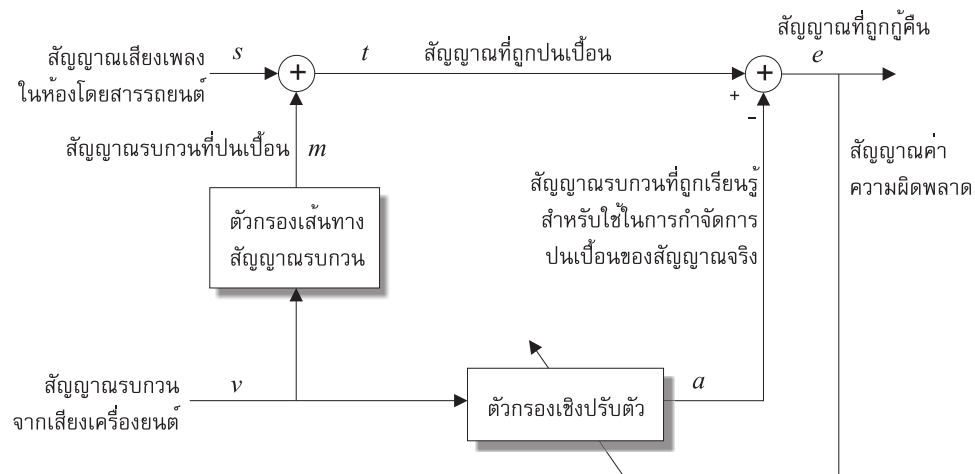
$$p_1 = [-1 \ 1 \ 1 \ -1 \ -1 \ -1 \ 1 \ -1 \ -1 \ -1 \ 1 \ -1 \ -1 \ -1 \ 1 \ -1]$$

- 11.2. เครือข่าย ADALINE ที่ใช้มีแสดงดังรูปที่ 11.19 กำหนดให้ใช้อัลกอริทึม LMS และค่าความผิดพลาดแบบผลรวมกำลังสอง (sum-squared error หรือ sse ใน MATLAB)



รูปที่ 11.19: เครือข่าย ADALINE สำหรับแยกแยะรูปแบบ

- จงออกแบบและฝึกสอนเครือข่ายให้สามารถจดจำรูปแบบตัวเลขทั้งสามได้ โดยทดลองสุ่มเลือกค่าคงที่การเรียนรู้ที่เหมาะสม
- แสดงและวิเคราะห์ผลการทดลอง รวมถึงแสดงการลู่เข้าของการฝึกสอนในรูปของค่าความผิดพลาดด้วย



รูปที่ 11.20: ระบบตัดออกสัญญาณรบกวนสำหรับห้องโดยสายรณนต

- ทดลองสร้างภาพรูปแบบตัวเลข 1 2 และ 4 ให้แตกต่างไปจากอินพุตที่ใช้ฝึกสอน แล้วนำเอาไปทดสอบกับเครือข่ายที่ได้รับการฝึกสอน สังเกตผลการทดลองว่าเครือข่ายสามารถจดจำรูปแบบอื่นๆ ที่แตกต่างจากการเรียนรู้หรือไม่ วิเคราะห์และสรุปผลการทดลอง

11.3. พิจารณาระบบตัดออกสัญญาณรบกวนในห้องโดยสายรณนตตามแผนผังระบบในรูปที่ 11.20 ถ้ากำหนดให้สัญญาณต้นฉบับคือเสียงเพลงที่เปิดฟังในห้องโดยสาย (เราต้องการสภาพบรรยากาศที่เงียบเพื่อฟังเพลง) และกำหนดให้สัญญาณรบกวนมาจากเสียงของเครื่องยนต์ จงออกแบบเครือข่าย ADALINE เพื่อใช้เป็นตัวกรองเชิงปรับตัว สำหรับตัดออกสัญญาณรบกวนในระบบดังกล่าว จงอธิบายการออกแบบระบบและจำลองสถานการณ์ พร้อมทั้งอภิปรายผลการทดลองที่ได้

[Handwritten signature]

- Martin T. Hagan, Howard B. Demuth, and Mark Beale. *Neural network design*. PWS Publishing, 1995.
- Jr M. E. Hoff. *Learning Phenomena in Networks of Adaptive Switching Circuits*. Tech. report 1554-1, Stanford, Stanford Electronics Labs., CA, July 1962.
- B. Widrow. Adaline and madaline - 1963, plenary speech. In *Proceedings of First IEEE International Conference on Neural Networks*, volume 1, pages 145 -- 158, San Diego, CA, June 23 1987.
- B. Widrow. Generalization and information storage in networks of adaline 'neurons'. In M. Yovitz, G. Jacobi, and G. Goldstein, editors, *Self-Organizing Systems 1962*, pages 435--461. Washington: Spartan Books, 1962.
- B. Widrow and M.A. Lehr. 30 years of neural networks: Perceptron, madaline and backpropagation. In *Proceedings of the IEEE*, volume 78, pages 1415--1442, 1990.
- B. Widrow and Jr M. E. Hoff. Adaptive switching circuits. In *IRE WESCON Convention Record*, pages 96--104, New York, 1960. IRE.
- B. Widrow and R. G. Winter. Neural nets for adaptive filtering and adaptive pattern recognition. In *IEEE Computer*, volume 31, pages 25--39, 1988.
- B. Widrow, R. G. Winter, and R. Baxter. Learning phenomena in layered neural networks. In *Proceedings of the 1st IEEE International Conference on Neural Networks*, volume 2, pages 411--429, San Diego, CA, June 1987.
- R. G. Winter. *Madaline Rule II: A New Method for Training Networks of Adalines*. PhD thesis, Stanford University, Stanford, CA, January 1989.



A handwritten signature in grey ink, consisting of several overlapping horizontal and vertical strokes. To the right of the signature is a circled number '2'.