บทที่ 2

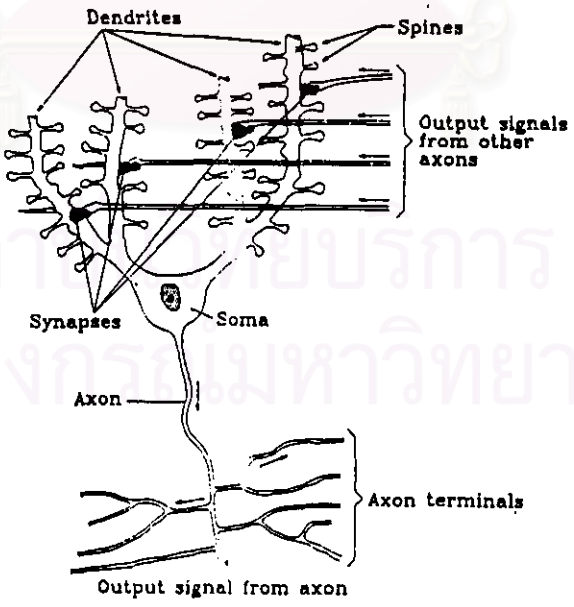
โครงข่ายประสาทเทียมและฮอปฟิลด์เน็ตเวอร์ก

2.1 กล่าวนำ

ในทางประสาทวิทยาทราบว่าสมองของมนุษย์ประกอบด้วยเซลล์ประสาท (neuron cell) จำนวน 10^{10} ถึง 10^{11} เซลล์ [3] ซึ่งโครงสร้างของเซลล์ประสาทประกอบด้วย

1. Soma เป็นตัวเซลล์ประสาท
2. Axon เป็นเส้นใยประสาทที่ต่อออกจากเซลล์ประสาทโดยทำหน้าที่เป็นด้านออกของเซลล์ประสาทเพื่อส่งต่อไปยังเซลล์ประสาทอื่น
3. Dendrites ทำหน้าที่เป็นด้านเข้าของเซลล์ประสาทเพื่อรับสัญญาณจากเซลล์ประสาทตัวอื่นโดยผ่าน axon ซึ่งต่อกับ synapes เพื่อส่งไปยัง soma
4. Synapses เป็นตัวต่อ dendrite กับ axon จากเซลล์ประสาทอื่น

โครงสร้างของเซลล์ประสาทนี้สามารถแสดงได้ดังรูปที่ 2.1



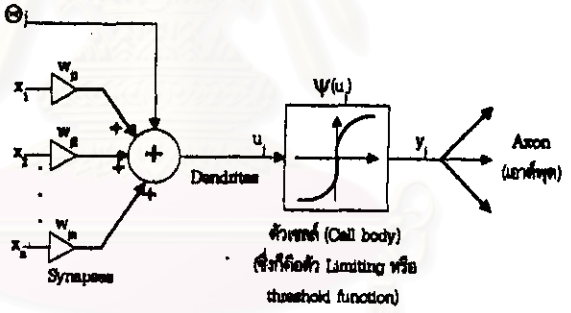
รูปที่ 2.1 โครงสร้างของเซลล์ประสาททางชีวภาพ

หอสมุดกลาง สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

การทำงานของโครงข่ายประสาทจะทำงานกันแบบขนานไปพร้อมๆ กันในแต่ละเซลล์ประสาทจึงทำให้โครงข่ายประสาทสามารถทำการประมวลผลได้อย่างรวดเร็ว ด้วยเหตุนี้จึงได้มีการพยายามจำลองโครงข่ายประสาทขึ้นเพื่อใช้ในการแก้ปัญหาต่างๆ

2.2 โครงข่ายประสาทเทียม

โครงข่ายประสาทเทียม (Artificial neural networks) หรือเรียกสั้นๆ ว่า "นิวรอลเน็ตเวอร์ก" คือโครงข่ายที่สร้างขึ้นเพื่อเรียนแบบการทำงานของโครงข่ายประสาทจริงๆ กล่าวโดยทั่วไปแล้วโครงข่ายประสาทเทียมคือระบบประมวลสัญญาณที่ประกอบด้วยตัวประมวลผลอย่างง่ายๆ จำนวนมากซึ่งเรียกหน่วยประมวลผลเหล่านี้ว่า " นิวรอน (neuron) " มาต่อเข้าด้วยกันเป็นโครงข่าย ซึ่งการทำงานของโครงข่ายนี้จะกระทำแบบขนานไปพร้อมๆ กันในแต่ละนิวรอนเพื่อแก้ปัญหาที่ต้องการ โดยแบบจำลองพื้นฐานของนิวรอนแสดงดังรูปที่ 2.2



รูปที่ 2.2 แบบจำลองพื้นฐานของเซลล์ประสาทเทียม (artificial neuron cell)

จากรูปที่ 2.2 จะเห็นว่าน้ำหนัก (weight, w_{ji}) ทำหน้าที่เสมือนกับ synapses เพื่อต่อด้านเข้า x_i เข้าสู่ตัวรวมซึ่งทำหน้าที่เหมือนกับ dendrites เพื่อส่งสัญญาณเข้าสู่ตัว activation function ($\Psi(u_j)$) ซึ่งเปรียบเสมือนกับ soma ส่วน Θ_j เป็นค่าระดับอ้างอิงที่ป้อนจากภายนอก ผลของการประมวลผล (y_j) จะถูกส่งออกที่ด้านออกของตัวขยายแบบไม่เป็นเชิงเส้นซึ่งเปรียบเสมือนกับ axon สามารถเขียนสมการได้เป็น

$$y_j = \Psi\left(\sum_{i=1}^{n} w_{ji}x_i + \Theta_j\right) \qquad (2.1)$$

ค่าระดับอ้างอิงจากภายนอก Θ_j สามารถเขียนในรูปของน้ำหนักได้โดย $w_{j0} = \Theta_j$ และ $x_0 = 1$ จึงทำให้สมการที่ (2.1) เขียนใหม่ได้เป็น

$$y_j = \Psi\left(\sum_{i=0}^{n} w_{ji} x_i\right) \tag{2.2}$$

เนื่องจากมีหลายวิธีในการต่อนิวรอนเข้าด้วยกันเป็นโครงข่ายขนาดใหญ่ ซึ่งรูปแบบของการต่อนี้เรียกว่าสถาปัตยกรรมซึ่งสถาปัตยกรรมของนิวรอลเน็ตเวอร์กสามารถแบ่งออกได้เป็น

1. Feedforward (multilayer) networks
2. Feedback (recurrent) networks
3. Cellular networks

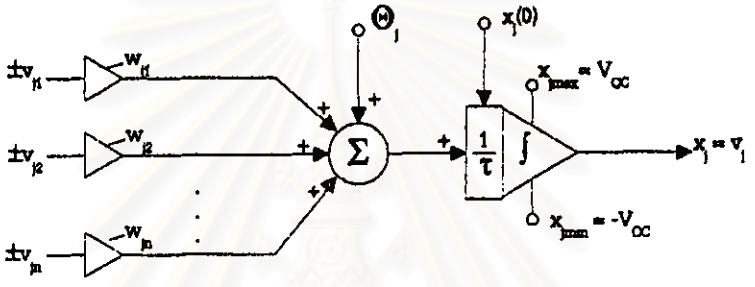
ได้มีการเสนอฮอปฟิลด์เน็ตเวอร์กในการแก้ปัญหาการหาค่าเหมาะที่สุด [4] ซึ่งฮอปฟิลด์เน็ตเวอร์กจัดเป็นนิวรอลเน็ตเวอร์กชนิดหนึ่ง โดยมีสถาปัตยกรรมเป็นแบบรีเคอร์เรนท์เน็ตเวอร์ก จากคุณสมบัติของนิวรอลเน็ตเวอร์กที่การประมวลผลจะกระทำแบบขนานไปพร้อมๆ กันในนิวรอนแต่ละตัว จึงทำให้สามารถประมวลผลได้อย่างรวดเร็ว ดังนั้นในการใช้ฮอปฟิลด์เน็ตเวอร์กเพื่อแก้ปัญหาการหาค่าเหมาะที่สุดจึงสามารถทำได้อย่างรวดเร็ว

วิทยานิพนธ์ฉบับนี้จึงใช้ฮอปฟิลด์เน็ตเวอร์กในการจัดสรรบัฟเฟอร์ด้านเข้าของแพ็กเกตสวิตช์ เนื่องจากในการจัดสรรบัฟเฟอร์ด้านเข้าของแพ็กเกตสวิตช์อาจต้องการความเร็วในการจัดสรรเพื่อให้ทันต่อการเปลี่ยนแปลงของค่าความเข้มทราฟฟิกที่เข้ามายังด้านเข้าต่างๆ ในที่นี้จะกล่าวถึงหลักการพื้นฐานของฮอปฟิลด์เน็ตเวอร์ก และฮอปฟิลด์เน็ตเวอร์กที่ใช้ในการแก้ปัญหาโปรแกรมมิงแบบไม่เป็นเชิงเส้นซึ่งเป็นปัญหาในที่นี้

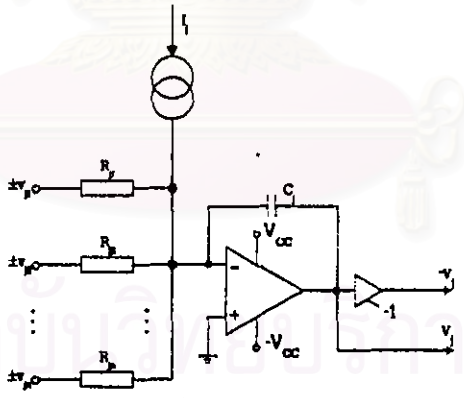
2.3 หลักการพื้นฐานของฮอปฟิลด์เน็ตเวอร์ก

2.3.1 แบบจำลองของนิวรอนที่ใช้ในฮอปฟิลด์เน็ตเวอร์ก

ฮอปฟิลด์เน็ตเวอร์กประกอบด้วยหลายๆ นิวรอนนำมาเชื่อมต่อกันเป็นโครงข่ายโดยมีการป้อนกลับค่าจากด้านออกของนิวรอนแต่ละตัวเพื่อไปยังด้านเข้าของนิวรอนแต่ละตัวรวมทั้งตัวเอง ซึ่งแบบจำลองของตัวนิวรอนในฮอปฟิลด์เน็ตเวอร์คแสดงดังรูปที่ 2.3



(ก)



(ข)

รูปที่ 2.3 (ก) บล็อกไดอะแกรม (ข) วงจรไฟฟ้าพื้นฐานของนิวรอนในฮอปฟิลด์เน็ตเวอร์ก

จากรูปที่ 2.3 จะเห็นว่าประกอบด้วยตัวอินติเกรตซึ่งทำหน้าที่เสมือน soma, ตัวรวมสัญญาณที่เข้ามาซึ่งเสมือน dendrites และค่าน้ำหนัก $w_{ji} = \frac{1}{R_{ji}}$ ซึ่งเปรียบเสมือน synapses

สังเกตว่าที่ด้านออกของตัวอินติเกรตจะให้ค่าที่ด้านออก 2 ค่าคือ $+v_j$ และ $-v_j$ ทั้งนี้เนื่องจากต้องการทำให้ตัวต้านทานซึ่งเปรียบเสมือน synaptic weight เป็นได้ทั้งค่าบวกหรือลบ นั่นคือถ้า synaptic weight เป็นบวกซึ่งได้จากการต่อ R_{ji} กับ $+v_j$ และ synaptic weight เป็นลบได้จากการต่อ R_{ji} กับ $-v_j$ จากรูปที่ 2.3 (ก) สามารถเขียนสมการเชิงอนุพันธ์เป็น

$$\tau_j \frac{dv_j}{dt} = \sum_{i=1}^{n} v_i w_{ji} + \Theta_j \tag{2.3}$$

หรือจากรูปที่ 2.3 (ข) จะได้เป็น

$$\begin{aligned} C_j \frac{dv_j}{dt} &= \sum_{i=1}^{n} \frac{v_i}{R_{ji}} + I_j \\ &= \sum_{i=1}^{n} v_i G_{ji} + I_j \end{aligned} \tag{2.4}$$

เมื่อ $\tau_j = C_j$ เป็นค่าคงที่เวลาของนิวรอน, $\Theta_j = I_j$ เป็นค่าคงที่หรือกระแสที่ป้อนจากภายนอกเพื่อเป็นระดับอ้างอิงของตัวนิวรอน และ G_{ji} เป็นค่าความนำไฟฟ้า (conductance) ซึ่งเป็นส่วนกลับของ R_{ji}

ค่าคงที่เวลาของนิวรอน τ_j นี้จะมีผลต่อการอัตราการเรียนรู้ของนิวรอลเน็ตเวอร์ก μ เนื่องจากเป็นส่วนกลับต่อกันนั่นคือถ้าค่าคงที่เวลาของนิวรอนมีค่ามากก็จะทำให้อัตราการเรียนรู้เร็วแต่ถ้าค่าคงที่เวลาของ นิวรอนมีค่าน้อยก็จะทำให้อัตราการเรียนรู้ของเน็ตเวอร์กช้า

ค่า synaptic weight ก็คือค่าของ $w_{ji} = G_{ji} = \frac{1}{R_{ji}}$ ที่ต่อจากด้านออกของตัวขยายที่ j ไปยังด้านเข้าของตัวขยายที่ i ซึ่งถ้าเป็น excitatory synapse ($w_{ji} > 0$) ก็จะต่อ R_{ji} เข้ากับ $+v_j$ และ inhibitory synapse ($w_{ji} < 0$) ก็จะต่อ R_{ji} เข้ากับ $-v_j$

2.3.2 ฮอปฟิลด์เน็ตเวอร์ก

เมื่อนำนิวรอนที่ได้กล่าวมาแล้วในบทที่ 2.3.1 มาต่อเข้าด้วยกันเป็นโครงข่ายแสดงดังรูปที่ 2.4 จะได้โครงข่ายที่เรียกว่า "ฮอปฟิลด์เน็ตเวอร์ก (Hopfield Network)" ซึ่งประกอบด้วยนิวรอนจำนวน n ตัวที่

ต่อเข้าด้วยกัน โดยด้านออกที่เป็นอินเวอร์สของตัวขยายแต่ละตัวจะทำหน้าที่เพื่อให้ได้น้ำหนักเป็นลบ (inhibitory) และผลรวมของน้ำหนักจะได้จากการรวมของกระแสที่ไหลผ่านตัวต้านทานที่ด้านเข้าของตัวขยายแต่ละตัวดังนั้นจากกฎกระแสของเคอร์ชอฟฟ์ (Kirchhoff's Current Law (KCL)) จะได้

$$\vec{C}\frac{d\vec{x}}{dt} = \vec{W}\vec{x} + \vec{\Theta} \tag{2.5}$$

เมื่อ

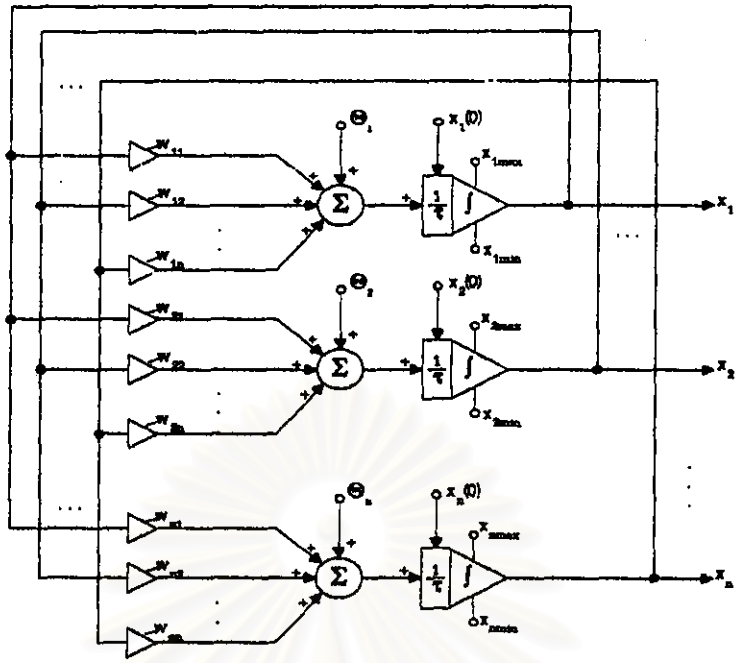
$\vec{C} = diag(C_1, C_2, \ldots, C_n)$ ซึ่งเป็นค่าคงที่เวลา

$\vec{x} = [x_1, x_2, \ldots, x_n]^T$ เวกเตอร์ด้านออกของตัวอินติเกรต

$$\vec{W} = \begin{bmatrix} w_{11} & w_{12} & \cdots & w_{1n} \\ w_{21} & w_{22} & \cdots & w_{2n} \\ . & . & & . \\ . & . & & . \\ w_{n1} & w_{n2} & \cdots & w_{nn} \end{bmatrix}$$ เป็นเมตริกซ์ของการต่อ หรือ เมตริกซ์ของ synaptic weights

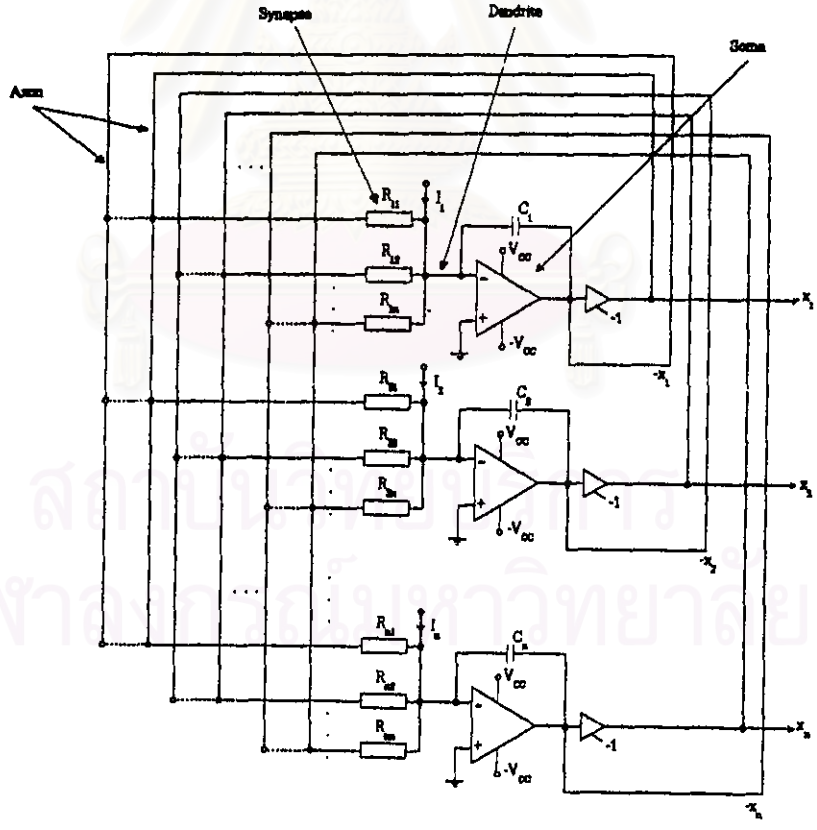
$\vec{\Theta} = [\Theta_1, \Theta_2, \ldots, \Theta_n]^T$ เวกเตอร์ของค่าคงที่ซึ่งป้อนจากภายนอก

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย



(ก)

รูปที่ 2.4 (ก) บล็อกไดอะแกรมพื้นฐานของฮอปฟิลด์เน็ตเวอร์ก



(ข)

รูปที่ 2.4 (ข) วงจรไฟฟ้าพื้นฐานของฮอปฟิลด์เน็ตเวอร์ก

สมการพลังงานของเน็ตเวอร์ก

$$E(\vec{x}) = -\frac{1}{2}\vec{x}^T W \vec{x} - \vec{x}^T \vec{\Theta} \tag{2.6}$$

และ

$$\frac{d\vec{x}}{dt} = -\nabla_x E(\vec{x}) = W\vec{x} + \vec{\Theta} \tag{2.7}$$

เมื่อ

$$\vec{x} = [x_1, x_2, \ldots, x_n]^T$$

$$\nabla_{\vec{x}} E(\vec{x}) = \left[\frac{\partial E}{\partial x_1}, \frac{\partial E}{\partial x_2}, \ldots, \frac{\partial E}{\partial x_n}\right]^T$$

2.4 การเขียนปัญหาการหาค่าที่เหมาะที่สุดแบบไม่มีข้อบังคับในรูปของฮอปฟิลด์เน็ตเวอร์ก

ถ้า $E(\vec{x})$ เป็นฟังก์ชันใดๆ ดังนั้นในการหาเวกเตอร์ $\vec{x}$ ซึ่งทำให้ค่าของฟังก์ชันวัตถุประสงค์ $E(\vec{x})$ มีค่าต่ำสุดโดยไม่มีข้อบังคับโดยใช้วิธีเกรเดียนต์สเซนต์ แสดงได้เป็น

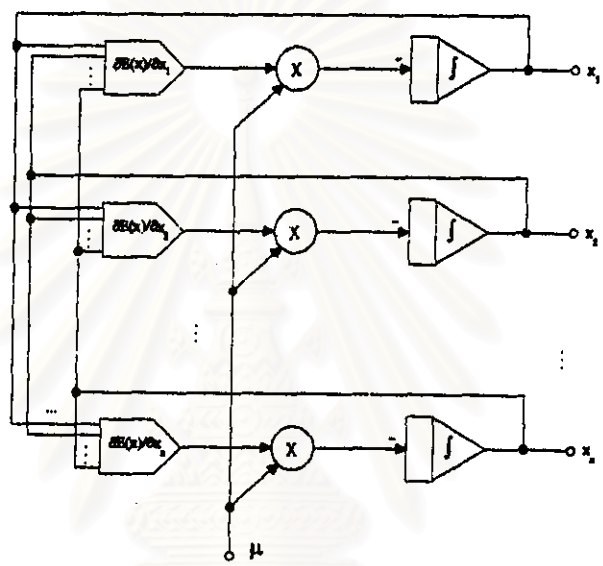
$$\vec{x}^{(k+1)} = \vec{x}^{(k)} - \vec{\mu}\nabla E(\vec{x}^{(k)}) \tag{2.8}$$

$$\frac{d\vec{x}}{dt} = -\vec{\mu}\nabla_x E(\vec{x}) \tag{2.9}$$

$\vec{x}^{(k)}$ เป็นค่าของเวกเตอร์ $\vec{x}$ ในรอบของการทำซ้ำที่ k (k = 0, 1, 2,...) และ $\vec{x}^{(0)}$ คือค่าเวกเตอร์ $\vec{x}$ เริ่มต้นที่ประมาณใกล้จุดที่เวกเตอร์ $\vec{x}$ ทำให้ฟังก์ชันวัตถุประสงค์ต่ำสุด ($\vec{x}^*$) ซึ่งเป็น local minimum ของฟังก์ชัน $E(\vec{x})$ ณ จุดนี้จะทำให้ $\nabla E(\vec{x}) = 0$ และ $\frac{d\vec{x}}{dt} = 0$

ทิศทางของการเคลื่อนตำแหน่งจากจุด $\vec{x}^{(k)}$ ไปยังจุด $\vec{x}^{(k+1)}$ จะไปตามทิศทางของ $-\nabla E(\vec{x})$ โดยความยาวของการเคลื่อนที่ในแต่ละขั้นจะขึ้นอยู่กับค่า $\vec{\mu}$ นั่นคือถ้า $\vec{\mu}$ มีค่ามากความยาวในการเคลื่อน

ที่จากจุด $\bar{x}^{(k)}$ ไปยังจุด $\bar{x}^{(k+1)}$ ก็จะมากจึงทำให้การเคลื่อนเข้าสู่จุด $\bar{x}^*$ เร็ว และถ้า $\bar{\mu}$ มีค่าน้อยความยาวในการเคลื่อนที่จากจุด $\bar{x}^{(k)}$ ไปยังจุด $\bar{x}^{(k+1)}$ ก็จะน้อยจึงทำให้การเคลื่อนเข้าสู่จุด $\bar{x}^*$ ช้า ในการเลือกค่า $\bar{\mu}$ นี้จะต้องเลือกให้พอเหมาะเนื่องจากถ้า ความยาวของการเคลื่อน $\bar{\mu}$ ยาวเกินไปอาจเคลื่อนที่ข้ามจุด $\bar{x}^*$ ไปจึงทำให้ไม่สามารถเข้าสู่จุด $\bar{x}^*$ นี้ได้เนื่องจากจะเกิดการข้ามไปมาบริเวณจุด $\bar{x}^*$ นี้นั่นเอง แต่ถ้า $\bar{\mu}$ มีค่าน้อยเกินไปการเคลื่อนเข้าสู่จุด $\bar{x}^*$ ก็จะช้า จากสมการที่ (2.8)-(2.9) สามารถเขียนเป็นบล็อกไดอะแกรมได้ดังรูปที่ 2.5



รูปที่ 2.5 บล็อกไดอะแกรมของฮอปฟิลด์เน็ตเวอร์กที่ใช้ในการหาค่าต่ำสุดของฟังก์ชันวัตถุประสงค์แบบไม่มีข้อบังคับ

2.5 การแก้ปัญหาโปรแกรมมิงแบบควอดราติกซึ่งมีข้อบังคับโดยใช้ฮอปฟิลด์เน็ตเวอร์ก

ในบทนี้จะเป็นตัวอย่างของการนำวิธีที่กล่าวในบทที่ 2.4 มาใช้ในการแก้ปัญหาโปรแกรมมิงแบบควอดราติกซึ่งเป็นกรณีหนึ่งของปัญหาโปรแกรมมิงแบบไม่เป็นเชิงเส้นโดยมีข้อบังคับเป็นแบบเชิงเส้น โดยที่ปัญหาโปรแกรมมิงแบบควอดราติกเขียนในรูปมาตรฐานได้เป็น

หาค่าต่ำสุด $$f(\bar{x}) = \bar{c}^T\bar{x} + \frac{1}{2}\bar{x}^T G\bar{x} \qquad (2.10)$$

ข้อบังคับ $\bar{A}\bar{x} \geq \bar{b}, \quad \bar{x} \geq 0$

เมื่อ $\bar{G}$ เป็นเมตริกซ์สมมาตรขนาด n x n ซึ่งปัญหาข้างต้นสามารถเขียนในรูปของสเกลาร์ดังนี้

หาค่าต่ำสุด $$f(\bar{x}) = \sum_{j=1}^{n} c_j x_j + \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} g_{ij} x_i x_j \qquad (2.11)$$

ข้อบังคับ $$\sum_{j=1}^{n} a_j x_j \geq b_j \ , \quad x_j \geq 0$$

ถ้า $\bar{G} = \bar{0}$ จะทำให้ปัญหาโปรแกรมมิงแบบควอดราติกกลายเป็นปัญหาโปรแกรมมิงแบบเชิงเส้น

ในการแก้ปัญหาเพื่อหาค่าที่เหมาะที่สุดของสมการในที่นี้จะใช้วิธีพีนอลตี (penalty method) [3] ซึ่งเป็น การเปลี่ยนปัญหาโปรแกรมมิงแบบมีข้อบังคับไปเป็นปัญหาโปรแกรมมิงแบบไม่มีข้อบังคับ โดยการรวมพีนอลตีฟังก์ชัน (penalty function) เข้ากับฟังก์ชันวัตถุประสงค์ $f(\bar{x})$ ซึ่งเทอมของพีนอลตีที่เพิ่มเข้าไปนี้จะมีค่าสูงเมื่อมีการละเมิด ข้อบังคับของปัญหาโปรแกรมมิงที่ต้องการหาค่าต่ำสุด ดังนั้นสามารถเขียนสมการพลังงานได้เป็น

$$E(\bar{x},\kappa) = f(\bar{x}) - \kappa\sum_{i=1}^{m}\left[r_i(\bar{x})\right]_-$$

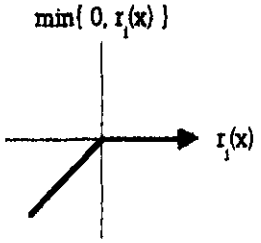
$$= \sum_{j=1}^{n} c_j x_j + \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} g_{ij} x_i x_j - \kappa\sum\left[r_i(\bar{x})\right]_- \qquad (2.12)$$

ซึ่ง $\bar{x} \geq \bar{0}$ และ κ เป็นพีนอลตีพารามิเตอร์ (penalty parameter) และ

$$r_i(\bar{x}) = \sum_{j=1}^{n} a_{ij} x_j - b_i \qquad (2.13)$$

$$\left[r_i\right]_- = \min\{0, r_i\}$$

โดยแสดงความสัมพันธ์ระหว่าง $r_i(\bar{x})$ กับ min{ 0, $r_i(x)$ } ดังรูปที่ 2.6



รูปที่ 2.6 ความสัมพันธ์ระหว่าง $r_i(x)$ กับ $\min\{ 0, r_i(x) \}$

โดยที่

$$\min\{0, r_i\} = 0 \quad \text{ถ้า} \quad r_i > 0$$
$$= r_i \quad \text{ถ้า} \quad r_i \leq 0$$

ซึ่ง $\min\{0, r_i\}$ คือพีนอลตีฟังก์ชัน

ใช้วิธีเกรเดียนเพื่อหาค่าต่ำสุดของสมการพลังงาน ซึ่งเขียนเป็นสมการเชิงอนุพันธ์ได้เป็น

$$\frac{dx_j}{dt} = -\mu \nabla E = -\mu \left[c_j + \sum_{i=1}^{n} g_{ji} x_i + \kappa \sum_{i=1}^{m} S_i a_{ij} \right] \qquad (j = 1,2,\ldots,n) \quad (2.14)$$

เมื่อ $\mu > 0,\ \kappa > 0$ ซึ่ง μ คืออัตราการเรียนรู้ของเน็ตเวอร์ก (ความยาวในการเคลื่อนที่จากจุด $\vec{x}^{(k)}$ ไปยังจุด $\vec{x}^{(k+1)}$)

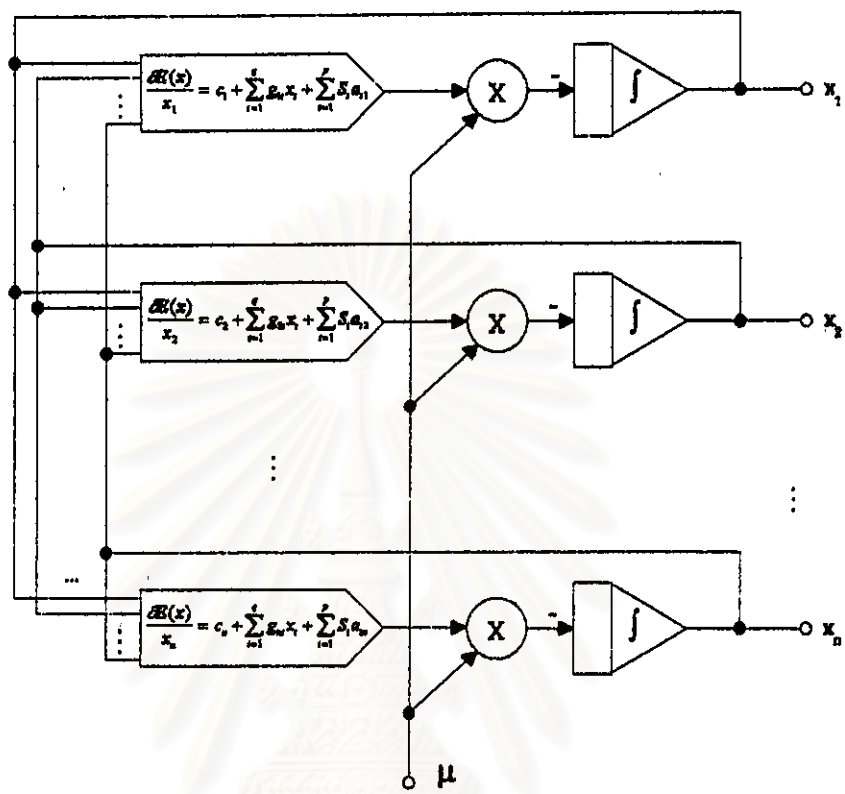
$$S_i = 1, \quad \text{ถ้า } r_i(\vec{x}) < 0$$
$$= 0, \quad \text{ถ้า } r_i(\vec{x}) \geq 0$$

กำหนดให้ $\frac{dx_j^{(k)}}{dt}$ เป็นอัตราการเปลี่ยนแปลงของเอาต์พุตของเน็ตเวอร์กในรอบที่ k ซึ่งจะนำค่าที่ได้นี้ป้อนกลับไปยังอินพุตเพื่อทำการปรับค่าเอาต์พุตของเน็ตเวอร์คในรอบที่ k+1 ดังนั้นจึงเขียนได้ว่า

$$x_j^{(k+1)} = x_j^{(k)} - \mu^{(k)} \left[c_j + \sum_{i=1}^{n} g_{ji} x_j^{(k)} + \kappa \sum_{i=1}^{m} S_j^{(k)} a_{ij} \right] \qquad (2.15)$$

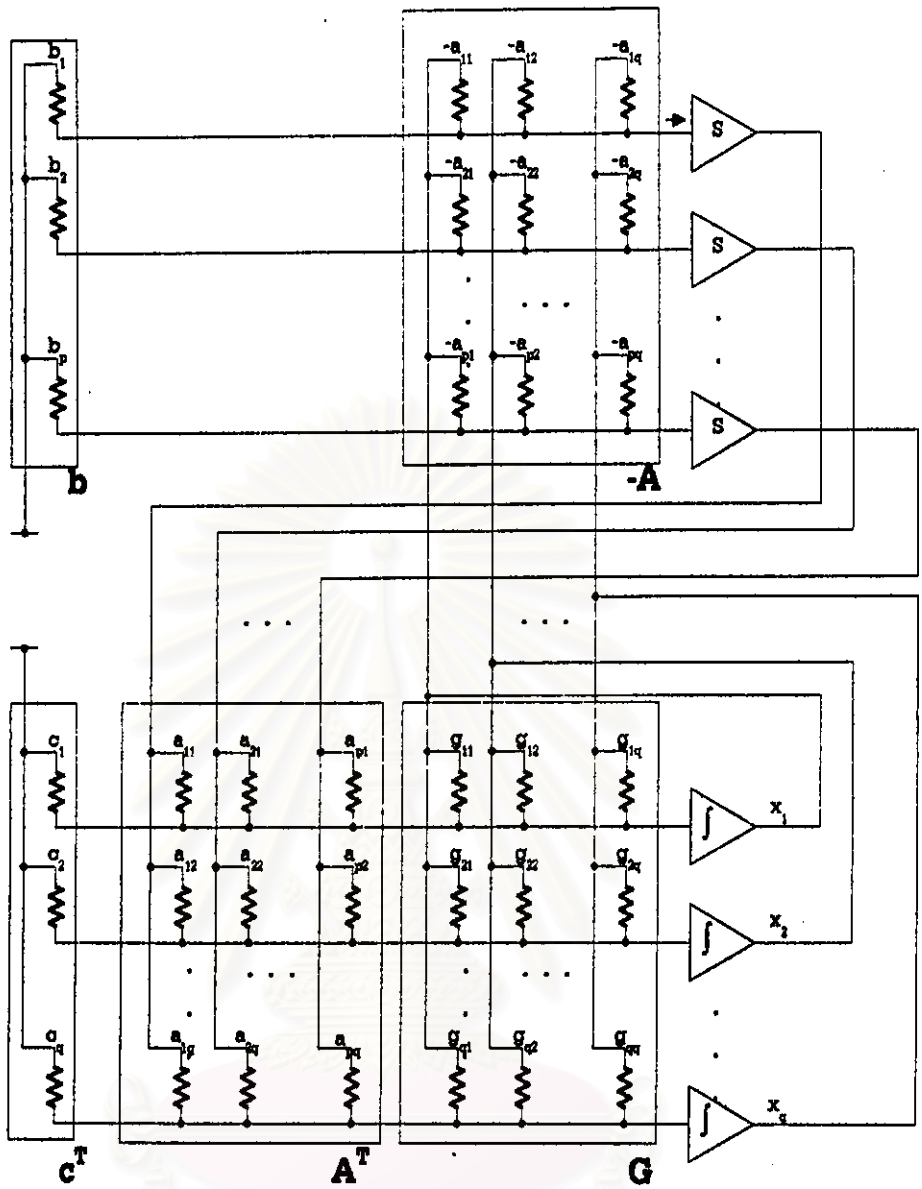
สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

สามารถเขียนบล็อกไดอะแกรมได้ดังรูปที่ 2.7(ก) และวงจรสมมูลย์ดังรูปที่ 2.7(ข)



รูปที่ 2.7 (ก) บล็อกไดอะแกรมของฮอปฟิลด์เน็ตเวอร์กที่ใช้ในการแก้ปัญหาโปรแกรมมิงแบบควอดราติก

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ 2.7 (ข) วงจรสมมูลย์ของฮอปฟิลด์เน็ตเวอร์กที่ใช้ในการแก้ปัญหาโปรแกรมมิงแบบควอดราติก

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย