บทที่ 3

วิธีดำเนินการวิจัย

ในบทนี้จะกล่าวถึงการนำทฤษฎีที่ได้ศึกษามาแล้วนั้น มาประยุกต์ใช้กับการวิจัย และพัฒนาอัลกอริทึมฝึกสอนโครงข่ายประสาทเทียม โดยได้แบ่งเนื้อหาในบทนี้เป็น 2 ส่วน คือ ส่วนที่ 1 เครื่องมือและอุปกรณ์ที่ใช้ในการดำเนินการวิจัย และส่วนที่ 2 ขั้นตอนการดำเนินการวิจัย

เครื่องมือและอุปกรณ์ที่ใช้ในการดำเนินการวิจัย

1. เครื่องคอมพิวเตอร์แล็ปทอป 1 เครื่อง สำหรับเขียนและรัน (RUN) โปรแกรมเพื่อบันทึกผลการทดลอง มีรายละเอียดดังนี้

1.1 หน่วยประมวลผลกลาง Intel Pentium R ความเร็วสูงสุด 1.90 GHz

1.2 หน่วยความจำหลัก 1.99 GB.

1.3 ความจุฮาร์ดดิสก์ 2.93 GHz.

1.4 ใช้ซอฟแวร์ระบบปฏิบัติการไมโครซอฟท์ วินโดวส์เอ็กซ์พี โปรเฟสชันเนล (Windows XP Procession)

2. ภาษาที่ใช้ในการพัฒนาอัลกอริทึมฝึกสอนโครงข่ายประสาทเทียม คือ MATLAB เวอร์ชั่น 2008a พร้อม Neural Network

ขั้นตอนของการดำเนินการวิจัย

1. ศึกษารวบรวมข้อมูล

2. การประยุกต์ใช้วิธีการหาค่าเหมาะสุดโดยอาศัยการรบกวนแบบสุ่มเพื่อฝึกสอนโครงข่ายประสาทเทียม

1. ศึกษารวบรวมข้อมูล

ในการศึกษาเกี่ยวกับอัลกอริทึมฝึกสอนโครงข่ายประสาทเทียมที่ผ่านมา พบว่า ปัจจุบันได้มีอัลกอริทึมฝึกสอนโครงข่ายประสาทเทียมมากมายที่ถูกคิดค้นขึ้น ซึ่งความซับซ้อนและขั้นตอนการทำงานของแต่ละอัลกอริทึมนั้นขึ้นอยู่กับความต้องการของผู้ใช้ ว่ามีความต้องการพัฒนาเพื่อนำอัลกอริทึมนั้นไปประยุกต์ใช้กับงานชนิดใด ซึ่งงานบางลักษณะต้องการความรวดเร็วและประสิทธิภาพในการประมวลผลสูง จากบทที่ 2 ได้ทำการทบทวนเอกสารที่ศึกษาเกี่ยวกับอัลกอริทึมที่ใช้ฝึกสอนโครงข่ายประสาทเทียมที่มีความสามารถในการประมวลผลได้รวดเร็วและมี

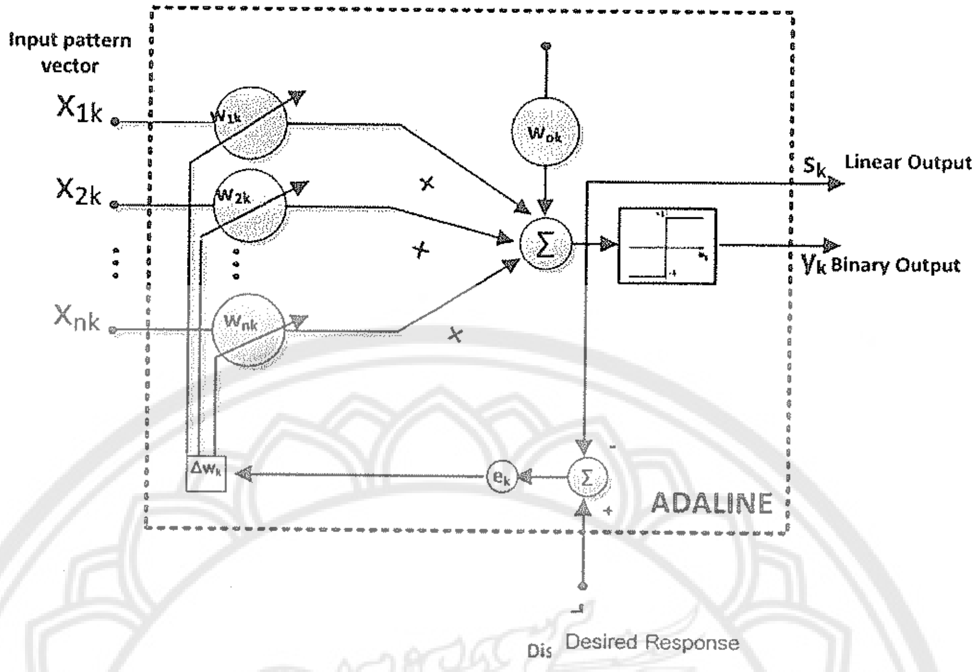
ประสิทธิภาพสูง เช่น Back propagation Algorithm และ Levenverg Marquartd Algorithm ซึ่งอัลกอริทึมเหล่านั้น สามารถประมวลผลได้รวดเร็วและมีประสิทธิภาพสูงเมื่อเปรียบเทียบการทำงานกับอัลกอริทึมอื่นๆ เมื่อทดลองนำอัลกอริทึม Back-propagation Algorithm และ Levenverg Marquartd Algorithm ไปทดสอบการทำงานบนดิจิตอลคอมพิวเตอร์ที่มีความสามารถสูง อัลกอริทึมเหล่านั้นสามารถทำงานได้อย่างรวดเร็วและมีประสิทธิภาพสูง แต่จากงานวิจัยที่ผ่านมาพบว่า หากจำเป็นต้องทดสอบการทำงานของโครงข่ายประสาทเทียมด้วยอัลกอริทึมดังกล่าว โดยใช้คอมพิวเตอร์ขนาดเล็ก เช่น ไมโครคอนโทรลเลอร์แล้ว พบว่า ไม่สามารถจะบรรจุกระบวนการทำงานของอัลกอริทึมเหล่านั้นลงในคอมพิวเตอร์ขนาดเล็กได้ทั้งหมด เนื่องจากข้อจำกัดเรื่องหน่วยความจำที่มีอยู่ในอนาล็อกคอมพิวเตอร์ ซึ่งส่วนมากแล้วมักมีจำนวนจำกัด จากการทบทวนเอกสารงานวิจัยเกี่ยวข้องที่ผ่านมานั้น ทำให้ผู้วิจัยต้องการศึกษาและพัฒนาอัลกอริทึมฝึกสอนโครงข่ายประสาทเทียม ที่มีขั้นตอนในการประมวลผลไม่ซับซ้อนและสามารถใช้ทดสอบการทำงานของโครงข่ายประสาทเทียมด้วยเครื่องมือคอมพิวเตอร์ขนาดเล็กได้

จากการทบทวนเอกสารงานวิจัยที่เกี่ยวข้อง พบว่า โครงข่ายประสาทเทียมจะมีข้อดีเหนือกว่า Genetic Algorithm และ Paticle Swarm Optimization คือ การทำงานของโครงข่ายประสาทเทียมนั้นไม่มีการสุ่มสร้างประชากร (Population) ทำให้การทำงานของโครงข่ายประสาทเทียมใช้หน่วยความจำน้อยกว่าการทำงานของ Genetic Algorithm และ Paticle Swarm Optimization ดังนั้น จากข้อดีดังกล่าว ผู้วิจัยจึงได้เลือกศึกษาการทำงานของโครงข่ายประสาทเทียมเพื่อนำไปประยุกต์ใช้กับงานชนิดอื่นๆ ที่มีข้อจำกัดในเรื่องของหน่วยความจำต่อไป

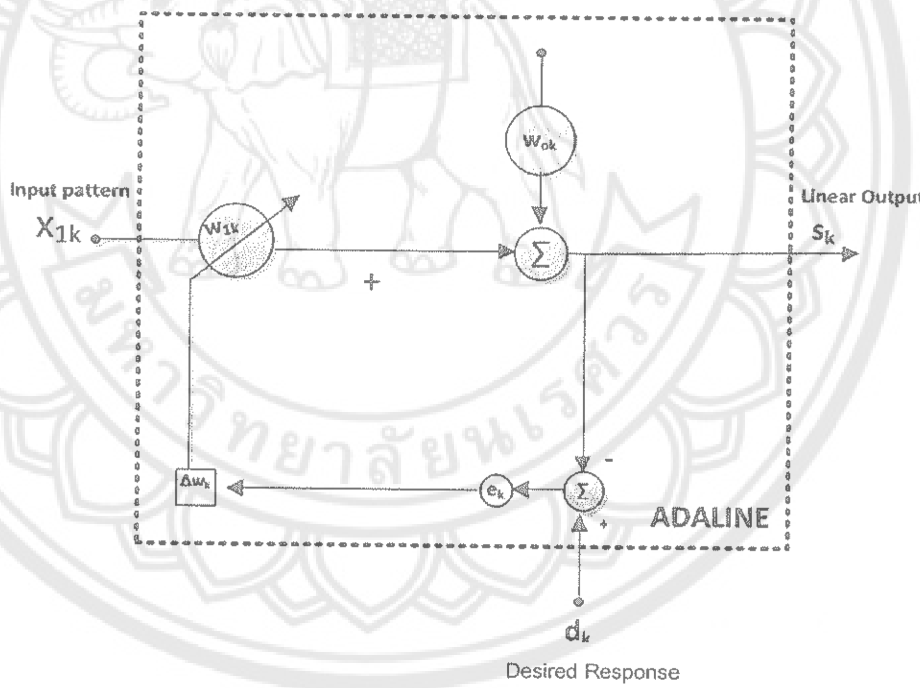
2. ออกแบบการทดลอง

ในขั้นตอนเริ่มต้นการทดลองได้ทดลองสร้างระบบปรับตัวเองอย่างง่าย โดยกำหนดให้ใช้ Adaptive Linear Combiner เพียงตัวเดียว เพื่อปรับค่าหนักของโครงข่าย โดยการทดลองปรับค่าน้ำหนักของโครงข่ายด้วยอัลกอริทึมที่ใช้หลักการการรบกวนค่าน้ำหนักแบบสุ่ม เพื่อปรับน้ำหนักโครงข่าย

ภาพ 7 แสดงโครงสร้างของ Adaline เปรียบเทียบกับระบบปรับตัวเองอย่างง่ายที่มีค่าน้ำหนักเพียงตัวเดียว



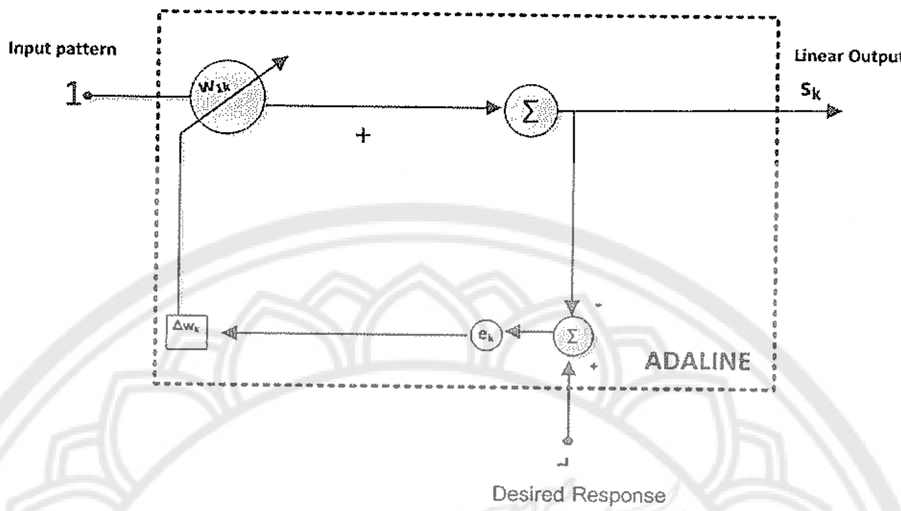
(a)



(b)

ภาพ 7 (a) โครงสร้าง Adaline (b) ระบบปรับตัวเองอย่างง่ายที่มีค่าน้ำหนักเพียงตัวเดียว

เพื่อความง่ายในการคำนวณ กำหนดให้ อินพุทโครงข่ายมีค่าเท่ากับ 1 แสดงดังภาพ 8



ภาพ 8 ระบบปรับตัวเองอย่างง่ายที่มีค่าน้ำหนักเพียงตัวเดียว และมีค่าอินพุทเข้าโครงข่ายเป็น 1

โดยทั่วไปแล้ว การทำงานของโครงข่ายประสาทเทียมด้วยดิจิตอลคอมพิวเตอร์ จะเริ่มต้นจากการมีอินพุทฝึกสอนและเอาท์พุทฝึกสอนโครงข่ายเข้าสู่โครงข่าย จากนั้นนำอินพุทฝึกสอนโครงข่ายคูณด้วยค่าน้ำหนักโครงข่าย (Weight) แต่ละสาขาของของอินพุทฝึกสอนนั้นๆ ได้ผลลัพธ์เป็นเอาท์พุทเชิงเส้น (Linear Output) และถูกนำไปเปรียบเทียบกับค่าระดับหรือเทรสโฮลด์ที่กำหนดไว้ หากเอาท์พุทเชิงเส้นที่ได้นั้นมีค่ามากกว่าค่าระดับก็จะส่งเอาท์พุทออกไปยังหน่วยประมวลผลตัวต่อไปซึ่งมีการเชื่อมต่อกันเป็นโครงข่าย โดยทั่วไปแล้ว อัลกอริทึมปรับค่าน้ำหนักของโครงข่าย จะต้องสามารถปรับค่าน้ำหนักของโครงข่ายเพื่อให้เอาท์พุทโครงข่ายนั้น มีค่าใกล้เคียงกับเอาท์พุทฝึกสอนมากที่สุด

เมื่อกำหนดให้ค่าอินพุทที่เข้าสู่โครงข่ายมีค่าเท่ากับ 1 แล้ว สามารถคำนวณค่าเอาท์พุทโครงข่าย (s_k) ได้ดังสมการ (66)

$$s_k = (1 \cdot w_{1k}) \tag{66}$$

จะได้เป็นเอาท์พุทเชิงเส้น (s_k) ดังสมการ (67)

$$s_k = w_{1k} \tag{67}$$

จากสมการ (67) จะเห็นได้ว่า เมื่อกำหนดให้อินพุทฝึกสอนโครงข่ายมีค่าเท่ากับ 1 ค่า เอาท์พุทเชิงเส้น (s_k) จะมีค่าเท่ากับค่าน้ำหนักโครงข่าย (w_{1k})

โดยทั่วไปแล้ว ในการฝึกสอนโครงข่ายประสาทเทียม เราจะวัดประสิทธิภาพของการฝึกสอนโครงข่ายจากค่าความผิดพลาดกำลังสอง (Squared Error) ที่ได้จากการฝึกสอนโครงข่ายในแต่ละรอบ ซึ่งคำนวณได้ตามสมการ (68)

$$e^2 = (d_k - s_k)^2 \tag{68}$$

เมื่อ

e^2 คือ ค่าความผิดพลาดกำลังสอง

d_k คือ ค่าเอาท์พุทฝึกสอนโครงข่าย

s_k คือ ค่าเอาท์พุทเชิงเส้น (Linerar Output)

ซึ่งการฝึกสอนโครงข่ายประสาทเทียมส่วนใหญ่แล้ว มีจุดประสงค์หลัก คือ การทำให้ค่าความผิดพลาดกำลังสองมีค่าน้อยที่สุด (Minimization) หรือทำให้ค่าความผิดพลาดกำลังสองมีค่าเท่ากับ 0

ดังนั้น สมการ (68) จะกลายเป็น

$$0 = (d_k - s_k)^2 \tag{69}$$

จากการแก้สมการ (69) เราทราบว่า เอาท์พุทเชิงเส้น หรือ s_k มีค่าตามสมการ (67)

$$s_k = w_{1k} \tag{70}$$

ดังนั้น แทนค่า $s_k = w_{1k}$ ในสมการ (70) ลงในสมการ (69) จะได้เป็นดังสมการ (71)

$$0 = (d_k - w_{1k})^2 \tag{71}$$

ทำการแก้สมการ (71) จะได้เป็น

$$w_{1k} = d_k \tag{72}$$

ในกรณีที่เป็นการฝึกสอนโครงข่ายแบบมีผู้ฝึกสอน (Supervisor Training) เราจะต้องทราบค่าเอาท์พุทฝึกสอน (d_k) ก่อนที่จะทำการฝึกสอนโครงข่าย ดังนั้น ค่า d_kจะมีค่าเป็นค่าคงที่ค่าหนึ่ง ตามที่ผู้ฝึกสอนโครงข่ายกำหนดขึ้น ซึ่งจะมีตัวอย่างแสดงการคำนวณอย่างละเอียดในส่วนของตัวอย่างการคำนวณต่อไป

จะเห็นว่าในกรณีที่ค่าอินพุทฝึกสอนที่เข้าสู่โครงข่ายมีค่าเท่ากับ 1 แล้ว ค่าน้ำหนักโครงข่าย(w_{1k}) ที่ทำให้ค่าความผิดพลาดกำลังสองมีค่าเท่ากับ 0 คือ ค่าน้ำหนักนั้นต้องมีค่าเท่ากับค่าอินพุทฝึกสอนโครงข่าย (d_k) ดังแสดงไว้ในสมการ (72)

3. **ตัวอย่างการคำนวณด้วยมือ**

ในการฝึกสอนโครงข่ายประสาทเทียมดังอัลกอริทึมต่างๆ นั้น โดยทั่วไปแต่ละอัลกอริทึมจะมีขั้นตอนในการปรับค่าน้ำหนักแตกต่างกันไป ซึ่งอัลกอริทึมที่ใช้ปรับค่าน้ำหนักส่วนมากนั้น มักจะมีการดัดแปลงมาจากเทคนิคการหาค่าเหมาะสุด (Optimization Technique)

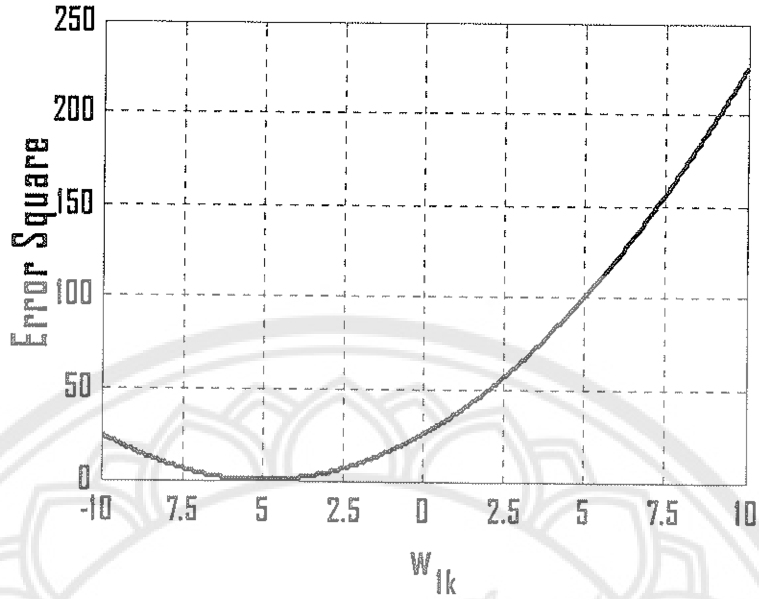
ในส่วนนี้ ได้ศึกษาการปรับค่าหนักเหมาะสุด โดยอาศัยหลักการ การรบกวนแบบสุ่ม (Random Perturbation) ตามเอกสารที่ได้ทบทวนมาแล้ว ดังนี้

1. Bernard Widrow and Samuel D.Stearns (1985)
2. Akaraphunt Vongkunghae (2005)

การทดลองในส่วนนี้ ได้ยึดหลักการการปรับค่าน้ำหนักตามงานวิจัยทั้งสองรายการที่กล่าวมา เพื่อเปรียบเทียบสมรรถนะและประสิทธิภาพในการปรับค่าน้ำหนักจากทั้งสองวิธี และเป็นเหตุผลประกอบในการวิเคราะห์ผลวิจัยต่อไป ในส่วนต่อไป เป็นตัวอย่างการคำนวณอย่างละเอียด ในการฝึกสอนโครงข่ายประสาทเทียม โดยอาศัยการปรับค่าน้ำหนักจาก Bernard Widrow and Samuel D.Stearns (1985) และ Akaraphunt Vongkunghae (2005)

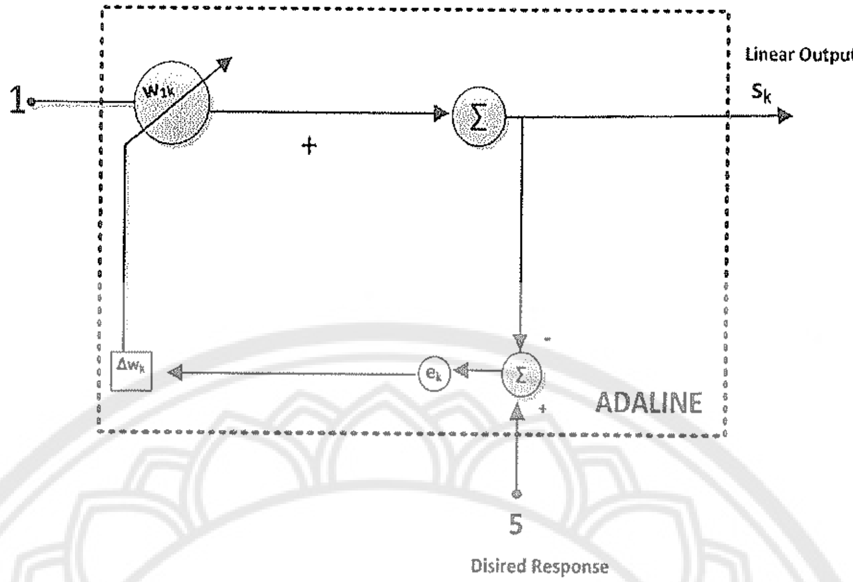
ตัวอย่างการคำนวณ

ในการหาค่าเหมาะสุดของโครงข่าย ระบบต้องการหาค่าที่สุด (Minimization) กราฟเส้นดังกล่าว กำหนดให้ จุดที่ต่ำสุด คือ 5 ดังแสดงในภาพ 9 ซึ่งภาพนี้ได้จากความสัมพันธ์ $\varepsilon^2 = (d_k - w_{1k} \cdot 1)^2$ เมื่อ $d_k = 5$



ภาพ 9 ค่าความผิดพลาดกำลังสองเทียบกับค่าน้ำหนักของโครงข่าย ขั้นตอนการคำนวณ

จากภาพ 9 จะเห็นว่า ค่าของ w_{1k} ทำให้มีที่ทำให้ค่าความผิดพลาดกำลังสองมีค่าน้อยสุดหรือมีค่าเท่ากับศูนย์ คือ 5 เนื่องจากการฝึกสอนโครงข่ายที่ทำการทดลองนี้ เป็นการฝึกสอนโครงข่ายแบบมีผู้ฝึกสอน (Supervisor Training) ดังนั้น เราจึงทราบได้ว่า ค่าเอาท์พุทฝึกสอนโครงข่ายหรือ d_k ของปัญหานี้ คือ 5 และกำหนดให้ระบบมีอินพุทฝึกสอนโครงข่ายเริ่มต้นมีค่าเท่ากับ 1 แสดงดังภาพ 10



ภาพ 10 โครงข่ายที่มีค่าอินพุทโครงข่ายมีค่าเท่ากับ 1 ค่าเอาท์พุทฝึกสอนโครงข่าย $d_k = 5$

เมื่อกำหนดให้ค่าอินพุทโครงข่ายเริ่มต้น มีค่าเท่ากับ 1 แล้ว สามารถคำนวณค่าเอาท์พุทเชิงเส้น (s_k) ของโครงข่ายได้เป็น ดังสมการ (73)

$$s_k = (1 \cdot w_{1k}) \tag{73}$$

จะได้เป็นดังสมการ (74)

$$s_k = w_{1k} \tag{74}$$

กำหนดให้ความผิดพลาดกำลังสองมีค่าน้อยที่สุด (Minimization) มีค่าเท่ากับ 0 สมการ (68) จะกลายเป็นดังสมการ (75)

$$0 = (d_k - s_k)^2 \tag{75}$$

จากการแก้สมการ (75) เราทราบว่าเอาท์พุทเชิงเส้นของโครงข่าย (s_k) คำนวณได้ดังสมการ (70)

$$s_k = w_{1k} \tag{76}$$

ดังนั้น แทนค่า $s_k = w_{1k}$ ตามสมการ (76) ลงในสมการ (75) จะได้เป็น

$$0 = (d_k - w_{1k})^2 \tag{77}$$

ในกรณีเราทราบว่าระบบของเรานั้น มีเอาท์พุทฝึกสอนโครงข่าย หรือ $d_k = 5$ แทนค่าลงในสมการ (77) จะได้ดังสมการ (78)

$$0 = (5 + w_{1k})^2 \tag{78}$$

กระจายสมการกำลังสอง ของสมการ (78) จะได้ดังสมการ (79)

$$w_{1k}{}^2 + 2(5)(w_{1k}) + 25 = 0 \tag{79}$$

จะได้เป็น

$$w_{1k}{}^2 + 10w_{1k} + 25 = 0 \tag{80}$$

กระจายผลต่างกำลังสองของสมการ (80) ได้เป็นดังสมการ (81)

$$(w_{1k} + 5)(w_{1k} + 5) = 0 \tag{81}$$

ดังนั้น จากสมการ (81) จะได้ค่า (w_{1k}) เป็นดังสมการ (82)

$$w_{1k} = -5 \tag{82}$$

4. ตัวอย่างการคำนวณด้วยโปรแกรม MATLAB

จากหัวข้อที่ผ่านมาเป็นการคำนวณด้วยมือ ในส่วนนี้ได้ทำการทดลองปรับค่าน้ำหนักในโครงข่ายด้วยโปรแกรม MATLAB

โดยในการทดลองได้ปรับค่าน้ำหนักโดยใช้หลักการจากทั้งสองวิธีนี้ ได้แก่

1. Bernard Widrow and Samuel D.Stearns (1985) เรียกวิธีดังกล่าวว่า Linear Random Search Algorithm (LRS) จะปรับค่าน้ำหนักโดยรายละเอียดดังนี้

$$w_{k+1} = w_k + \frac{\mu}{\sigma^2}[\varepsilon(w_k) - \varepsilon(w_k + U_k)]U_k$$

2. Akaraphunt Vongkunghae (2005) เรียกวิธีดังกล่าวว่า Random Perturbation จะปรับค่าน้ำหนักโดยโดยรายละเอียดดังนี้

$$\Delta e = e(W_n + v\Delta P) - e(W_n - v\Delta P)$$
$$W_{n+1} = W_n - \mu \cdot \Delta e \cdot v\Delta P$$

ขั้นตอนการปรับค่าน้ำหนักโดยโปรแกรม MATLAB นั้น ได้กำหนดค่าตัวแปรเริ่มต้นที่สำคัญของทั้งสองวิธีให้มีค่าเท่ากัน ดังต่อไปนี้

จำนวนรอบ (Epoch) ที่ใช้ฝึกโครงข่ายเท่ากับ 10,000 รอบ

v คือ ค่าเบี่ยงเบนมาตรฐาน (Standard Deviation) ของการกระจายตัวแปรสุ่ม

$$v = 0.001 \tag{83}$$

ΔP คือ การกระจายตัวแปรสุ่มเกาซ์เซี่ยน (Gaussian Distribution)

$$\Delta P = randn[-\infty, \infty] \tag{84}$$

$v \cdot \Delta P$ คือ การกระจายตัวแปรสุ่มแบบเกาซ์เซี่ยน (Gaussian Distribution) ที่มีค่าเบี่ยงเบนมาตรฐานเท่ากับ v

$$v \cdot \Delta P = v \cdot randn[-\infty, \infty] \tag{85}$$

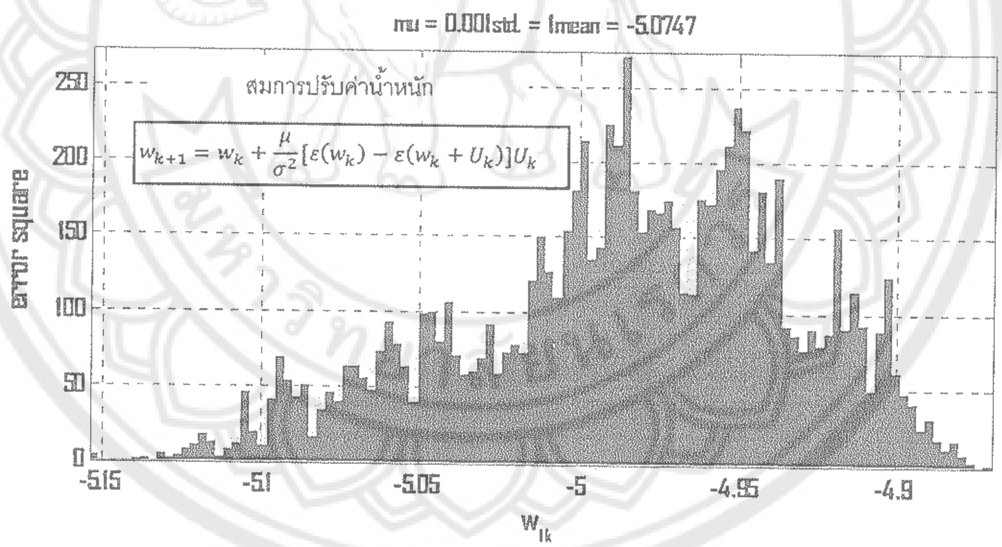
μ คือ ค่าอัตราการเรียนรู้ (Learning rate)

$$\mu = 0.001 \tag{86}$$

จากสมการ (82) เราทราบว่า ค่าน้ำหนักที่ถูกต้อง คือ $w_{1k} = -5$ หลักจากใช้อัลกอริทึมปรับค่าน้ำหนักของโครงข่ายตาม Bernard Widrow and Samuel D.Stearns (1985) และ Akaraphunt Vongkunghae (2005) แล้ว ค่าน้ำหนักที่ได้จากการใช้อัลกอริทึมทั้งสองปรับค่าน้ำหนักของโครงข่าย ได้ทำการพล็อตกราฟเพื่อแสดงความความหนาแน่นความน่าจะเป็น (Probability Density Function : PDF) ของค่าหนัก w_{1k} ซึ่งได้แสดงในภาพ 10 (a) และ (b) จากทั้งสองวิธีตามลำดับ

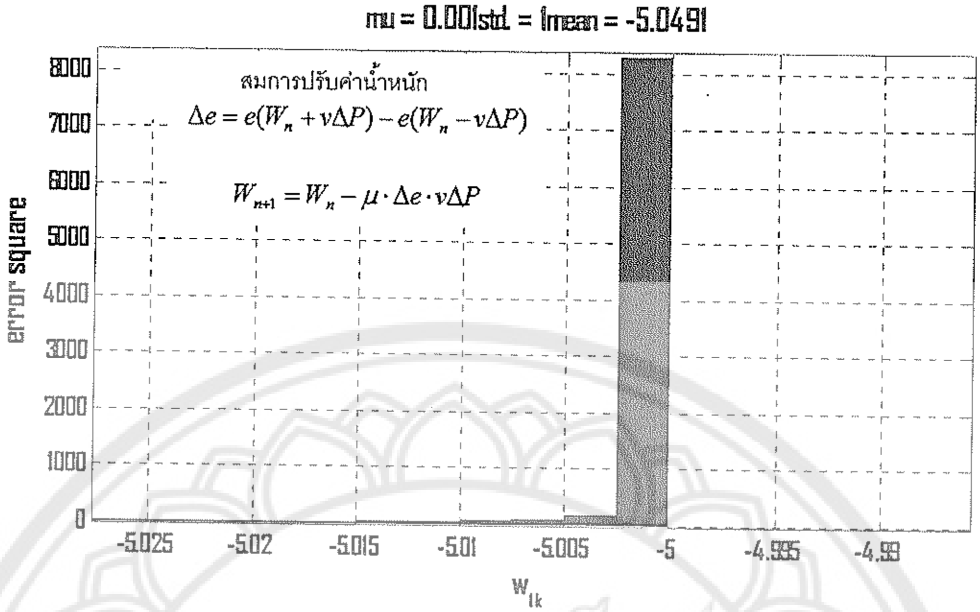
5. วิเคราะห์และเปรียบเทียบผลการทดลอง

จากภาพ 11 จะเห็นได้ว่า การกระจายตัวของค่าน้ำหนักจากทั้งสองวิธีนั้นมีความแตกต่างกัน ดังนั้น เพื่อเปรียบเทียบประสิทธิภาพของของค่าน้ำหนักที่ได้จากกระบวนการปรับค่าน้ำหนักของทั้งสองวิธี โดยได้พิจารณาค่าน้ำหนักที่มีค่าอยู่ระหว่าง -5 ± 0.001 ของการการหาค่าน้ำหนักทั้งหมด 10,000 ค่า ทั้งจากวิธีของ Bernard Widrow and Samuel D.Stearns (1985) และ Akaraphunt Vongkunghae (2005) จากนั้นค่าน้ำหนักที่ได้มาคำนวณค่าความน่าจะเป็น พบว่า



(a)

ภาพ 11 ฟังก์ชั่นความหนาแน่นความน่าจะเป็นของค่าหนัก w_{1k} โดย (a) ปรับค่าน้ำหนักโดยใช้อัลกอริทึม Bernard Widrow and Samuel D.Stearns (1985) (b) ปรับค่าน้ำหนักโดยใช้อัลกอริทึม Akaraphunt Vongkunghae (2005)



(b)

ภาพ 11 (ต่อ)

ค่าน้ำหนักที่มีค่าอยู่ระหว่าง -5 ± 0.001 โดยวิธีของ Bernard Widrow and Samuel D.Stearns (1985) มีค่าคำนวณได้ดังสมการ (87) ตามลำดับ

$$\frac{3906}{10000} X100 = 39.06\,\% \tag{87}$$

ค่าน้ำหนักที่มีค่าอยู่ระหว่าง -5 ± 0.001 โดยวิธีของ Akaraphunt Vongkunghae (2005) มีค่าคำนวณได้ดังสมการ และ (88) ตามลำดับ

$$\frac{8193}{10000} X100 = 81.93\,\% \tag{88}$$

จากการคำนวณค่าความน่าจะเป็นของค่าน้ำหนักที่มีค่าอยู่ระหว่าง -5 ± 0.001 จากทั้งสองอัลกอริทึมแล้วนั้น พบว่า อัลกอรทึมที่นำเสนอใน Akaraphunt Vongkunghae (2005) มีค่าความน่าจะเป็นของน้ำหนักที่ถูกต้อง สูงกว่า อัลกอริทึมของ Bernard Widrow and Samuel D.Stearns (1985)

ดังนั้น ในวิทยานิพนธ์เรื่องนี้ผู้วิจัยจึงได้นำอัลกอริทึมนี้ ใช้เป็นวิธีการหลักในการฝึกสอยโครงข่ายประสาทเทียมแบบป้อนผลการคำนวณไปข้างหน้า (Feedforward Neural Network) เพื่อเปรียบเทียบกับอัลกอริทึมอื่นๆ

6. การประยุกต์ใช้วิธีการหาค่าเหมาะสม (Optimization) โดยอาศัยการรบกวนแบบสุ่ม (Random Perturbation)

ในส่วนนี้เป็นการนำหลักการ การปรับค่าน้ำหนักแบบสุ่ม จาก Akaraphunt Vongkunghae (2005) มาฝึกสอนโครงข่ายประสาทเทียมแบบป้อนผลการคำนวณไปข้างหน้ามีขั้นตอนสรุป ได้ดังนี้

ขั้นที่ 1 ทำการคำนวณค่าความผิดพลาดกำลังสอง (error square) ของโครงข่ายดังสมการ (89) เพื่อเก็บไว้ใช้ตรวจสอบการลู่เข้าสู่คำตอบและใช้ในเงื่อนไขที่จะหยุดฝึกสอนเมื่อค่าความผิดพลาดกำลังสองเฉลี่ยเป็นที่ยอมรับได้

$$\varepsilon(\mathbf{w}_k) = \sum_{\mathbf{p}=1}^{\mathbf{N}} (t_p - y_{p\,w_k})^2 \tag{89}$$

ขั้นที่ 2 ทำการสุ่มค่าเวกเตอร์ $\Delta\mathrm{P}$

ขั้นที่ 3 ทำการรบกวนค่าน้ำหนักของโครงข่ายด้วย $v \cdot \Delta\mathrm{P}$ ทั้งด้านบวกและด้านลบดังแสดง ในสมการ (90) และ (91)

$$\mathbf{w}_+ = \mathbf{w}_k + v \cdot \Delta\mathrm{P} = [w_{+1} \quad w_{+2} \ldots w_{+i}]^{\mathrm{T}} \tag{90}$$

$$\mathbf{w}_- = \mathbf{w}_k - v \cdot \Delta\mathrm{P} = [w_{-1} \quad w_{-2} \ldots w_{-i}]^{\mathrm{T}} \tag{91}$$

ขั้นที่ 4 ทำการคำนวณค่าความผิดพลาดกำลังสอง ของโครงข่ายที่ถูกรบกวนค่าน้ำหนักด้วย $v \cdot \Delta\mathrm{P}$ ทั้งด้านบวกและด้านลบดังแสดง ในสมการ (92) และ (93)

$$\varepsilon(\mathbf{w}_+) = \sum_{\mathbf{p}=1}^{\mathbf{N}} (t_p - y_{p\,w_+})^2 \tag{92}$$

$$\varepsilon(\mathbf{w}_-) = \sum_{\mathbf{p}=1}^{\mathbf{N}} (t_p - y_{p\,w_-})^2 \tag{93}$$

ขั้นที่ 5 ทำการคำนวณค่าผลต่างของค่าความผิดพลาดกำลังสอง $\Delta\varepsilon$, คำนวณได้ดังสมการที่ (94)

$$\Delta\varepsilon = \varepsilon(\mathbf{w}_+) - \varepsilon(\mathbf{w}_-) \tag{94}$$

ขั้นที่ 6 ทำการปรับค่าน้ำหนักโครงข่าย โดยใช้หลักการการค่าสัมพันธ์ระหว่างผลต่างค่าความผิดพลาดกำลังสอง ($\Delta\varepsilon$) กับค่าการรบกวน ($v \cdot \Delta\mathbf{P}$) ดังสมการที่ (95)

$$\mathbf{w}_{\mathbf{k+1}} = \mathbf{w}_{\mathbf{k}} - \mu \cdot \Delta\varepsilon \cdot v\Delta\mathbf{P} \tag{95}$$

ขั้นที่ 7 ย้อนกลับไปทำขั้นที่ 1

โดยที่สัญลักษณ์ในสมการมีความหมายดังที่แสดงไว้ดังต่อไปนี้

$\varepsilon(\mathbf{w}_\mathbf{k})$ คือ ผลรวมความผิดพลาดกำลังสอง เมื่อค่าน้ำหนัก คือ $\mathbf{w}_k$

$\varepsilon(\mathbf{w}_+)$ คือ ผลรวมความผิดพลาดกำลังสอง เมื่อค่าน้ำหนัก คือ $\mathbf{w}_+$

$\varepsilon(\mathbf{w}_-)$ คือ ผลรวมความผิดพลาดกำลังสอง เมื่อค่าน้ำหนัก คือ $\mathbf{w}_-$

i คือ จำนวนสมาชิกในเวกเตอร์ค่าน้ำหนัก

K คือ ครั้งที่ปรับค่าน้ำหนัก 1,2,3,....

$\mathbf{w}_k$ คือ เวกเตอร์ค่าน้ำหนักก่อนปรับค่า $\mathbf{w}_k = [w_{k1} \quad w_{k2} \dots \quad w_{ki}]^{\mathrm{T}}$

$\mathbf{w}_{\mathbf{k+1}}$ คือ เวกเตอร์ค่าน้ำหนักหลังถูกปรับค่า

$\mathbf{w}_+$ คือ เวกเตอร์ค่าน้ำหนักที่ถูกรบกวนด้านบวก

$$\mathbf{w}_+ = \mathbf{w}_k + v \cdot \Delta\mathbf{P} = [w_{+1} \quad w_{+2} \dots w_{+i}]^{\mathrm{T}}$$

$\mathbf{w}_-$ คือ เวกเตอร์ค่าน้ำหนักที่ถูกรบกวนด้านลบ

$$\mathbf{w}_- = \mathbf{w}_k - v \cdot \Delta\mathbf{P} = [w_{-1} \quad w_{-2} \dots w_{-i}]^{\mathrm{T}}$$

t_p คือ ค่าเอาท์พุทที่ต้องการ ของข้อมูลอินพุทฝึกสอน ลำดับที่ p ค่าน้ำหนัก คือ $\mathbf{w}_{+k}$ ฝึกสอนโครงข่าย

$y_{p\,w_+}$ คือ ค่าเอาท์พุทโครงข่ายของข้อมูลอินพุทฝึกสอน ลำดับที่ p เมื่อค่าน้ำหนัก คือ $\mathbf{w}_+$

$y_{p\,w_-}$ คือ ค่าเอาท์พุทโครงข่ายของข้อมูลอินพุทฝึกสอน ลำดับที่ p เมื่อ ค่าน้ำหนัก คือ $\mathbf{w}_-$

$y_{p\,w_k}$ คือ ค่าเอาท์พุทโครงข่ายของข้อมูลอินพุทฝึกสอน ลำดับที่ p เมื่อ ค่าน้ำหนัก คือ $\mathbf{w}_k$

N คือ จำนวนอินพุทฝึกสอนทั้งหมด

v คือ ค่าคงที่ควบคุมลักษณะการกระจายตัวแปรสุ่ม

$\Delta\mathrm{P}$ คือ เวกเตอร์การรบกวน มีลักษณะการกระจายตัวแปรสุ่มแบบเกาส์เซียนปรกติ (Normal Gaussian Distribution) มีค่าเบี่ยงเบนมาตรฐาน เท่ากับ 1.0

$v \cdot \Delta\mathrm{P}$ คือ เวกเตอร์ค่าการรบกวน มีลักษณะการกระจายตัวแปรสุ่มเกาซ์เซี่ยน (Gaussian Distribution) มีค่าเบี่ยงเบนมาตรฐาน เท่ากับ v

$\Delta\varepsilon$ คือ ผลต่างระหว่างค่าความผิดพลาดกำลังสองของค่าน้ำหนักที่ถูกรบกวนด้านบวก $(\mathbf{w} + v \cdot \Delta\mathrm{P})$ และค่าความผิดพลาดกำลังสองของค่าน้ำหนักที่ถูกรบกวนด้านลบ $(\mathbf{w} - v \cdot \Delta\mathrm{P})$

μ คือ ค่าอัตราการเรียนรู้

จากสมการ (95) เป็นสมการปรับค่าน้ำหนัก โดยใช้สหสัมพันธ์ไขว้น้อยสุดระหว่างผลต่างค่าความผิดพลาดกำลังสอง $(\Delta\varepsilon)$ กับค่าการรบกวน $(v \cdot \Delta\mathrm{P})$ จะเห็นว่าในทุกขั้นตอนก่อนการปรับค่าน้ำหนักโครงข่าย จะไม่มีการคำนวณค่าอนุพันธ์ใดๆ ซึ่งทำให้ขั้นตอนการประมวลผลไม่ซับซ้อนเมื่อเปรียบเทียบกับอัลกอริทึมอื่นๆ ที่ปรับค่าน้ำหนักโดยการอนุพันธ์เพื่อคำนวณค่าเกรเดียนท์ จึงทำให้ประหยัดทรัพยากรในการคำนวณ เมื่อนำสมการ (95) มาหาค่าเฉลี่ยหรือค่าคาดหวัง (Expect Value) ดังแสดงในสมการ (96)

$$\mathrm{E}[\mathbf{w_{k+1}}] = \mathrm{E}[\mathbf{w_k} - \mu \cdot \Delta\varepsilon \cdot v\Delta\mathrm{P}\,] \tag{96}$$

จากสมการ (96) ในขณะที่ทำการพิจารณานี้ เรากำลังพิจารณาที่จุด $\mathbf{w_k}$ จุดใดจุดหนึ่ง ดังนั้นเราจะเห็นว่าจุด $\mathbf{w_k}$ นั้นเป็นตัวแปรคงที่อิสระไม่ขึ้นอยู่กับ $\mu \cdot \Delta\varepsilon \cdot v\Delta\mathrm{P}$ และเมื่อพิจารณาพจน์ $\mu \cdot \Delta\varepsilon \cdot v\Delta\mathrm{P}$ จะเห็นว่า ตัวแปร μ นั้น มีค่าคงที่ไม่เปลี่ยนแปลง ดังนั้น สมการที่ (96) จึงถูกเขียนใหม่ ดังสมการ (97)

$$\mathrm{E}[\mathbf{w_{k+1}}] = \mathrm{E}[\mathbf{w_k}] - \mu \cdot \mathrm{E}[\Delta\varepsilon \cdot v\Delta\mathrm{P}\,] \tag{97}$$

เมื่อพิจารณาสมการ (97) เราจะสังเกตว่าเมื่อสหสัมพันธ์ไขว้ระหว่าง $\Delta\varepsilon$ และ $v\Delta P$ นั้นมีค่าเป็นศูนย์สามารถเขียนเป็นสมการ (98)

$$\mathbb{E}[\Delta\varepsilon \cdot v\Delta \mathrm{P}\,] = 0 \tag{98}$$

นำสมการ (98) แทนลงใน (97) จะเป็นผลให้ การปรับค่าน้ำหนักนั้นไม่ขึ้นอยู่กับการรบกวน แสดงในสมการ (99)

$$\mathrm{E}[\mathbf{w}_{\mathbf{k+1}}] = \mathrm{E}[\mathbf{w}_{\mathbf{k}}] \tag{99}$$

ที่จุดนี้เราค่าหวังว่า $\mathbf{w}_{\mathbf{k+1}}$ จะมีค่าเท่ากับ $\mathbf{w}_{\mathbf{k}}$ และเราคาดหวังว่า $\mathbb{E}[\Delta\varepsilon \cdot v\Delta \mathrm{P}\,]$ จะมีค่าเป็นศูนย์ ซึ่งนั่น คือ ผลการรบกวนค่าน้ำหนักทั้งที่เป็นทางด้านบวกและทางด้านลบให้ผลของค่าความผิดพลาดกำลังสองมีค่าเท่ากัน หรือ $\varepsilon(\mathbf{w}_+) = \varepsilon(\mathbf{w}_-)$ เป็นผลทำให้ค่า $\Delta\varepsilon$ มีค่าเท่ากับศูนย์

ถ้าหากว่าเราพิจารณาในกรณีที่ $\Delta\varepsilon \cdot v\Delta \mathrm{P}$ มีค่าเป็นบวก ย่อมแสดงว่า การรบกวนค่าน้ำหนักในด้านบวก เป็นผลให้เกิด $\varepsilon(\mathbf{w}_+)$ มีค่ามากกว่า $\varepsilon(\mathbf{w}_-)$ ที่เกิดจากการรบกวนค่าน้ำหนักในด้านลบ จากสมการ (94) จะเห็นว่า ในกรณีนี้ค่าน้ำหนักจะถูกปรับให้มีค่าลดค่าลงเท่ากับ $\mu \cdot \Delta\varepsilon \cdot v\Delta \mathrm{P}$ เมื่อ $\Delta\varepsilon \cdot v\Delta \mathrm{P}$ เป็นบวกเกิดจากการรบกวนค่าน้ำหนักด้านบวกเป็นผลให้เกิดค่าความผิดพลาดกำลังสองมากกว่าการรบกวนค่าน้ำหนักด้านลบ $\varepsilon(\mathbf{w}_+) > \varepsilon(\mathbf{w}_-)$ จะเห็นว่าการทำเช่นนี้ เป็นวิธีการหาค่าเหมาะสมอีกวิธีการหนึ่ง โดยที่ค่าน้ำหนักนี้ จะถูกปรับเข้าหาค่าๆ หนึ่ง ซึ่งเป็นผลให้ค่าสหสัมพันธ์ไขว้น้อยสุดระหว่าง $\Delta\varepsilon$ และ $v\Delta\,\mathrm{P}$ มีค่าเป็นศูนย์ ด้วยเหตุนี้เองจึงเรียกวิธีการปรับค่าน้ำหนักด้วยวิธีนี้ วิธีการสหสัมพันธ์ไขว้น้อยสุด (LCC: Least Cross Correlation) ขอให้สังเกตว่าการลู่เข้าสู่ค่าเหมาะสม (Optimum Point) ซึ่งเป็นคำตอบของการหาค่าเหมาะสมที่สุดนั้น ไม่ใช่คำตอบเดียวกับการใช้วิธีการที่อยู่บนพื้นฐานของการคำนวณค่าอนุพันธ์ คำตอบของวิธีการที่นำเสนอนี้จะอยู่ ณ จุดที่ $\mathbb{E}[\Delta\varepsilon \cdot v\Delta \mathrm{P}\,] = 0$ ไม่ใช่จุดที่ค่าเกรเดียนท์หรือ $\nabla = \frac{\partial\varepsilon}{\partial w} = 0$ คำตอบที่แตกต่างกันนี้จะแสดงให้เห็นเพิ่มเติมในหัวข้อถัดไป