การปรับค่าอัตราการเรียนรู้สำหรับการเรียนรู้แบบออนไลน์ด้วยวิธีเคอร์เนล

ADAPTIVE LEARNING RATE IN ONLINE KERNEL METHOD

นางสาววารี คำแสน รหัส 48280125

นายสุนทร ศิลปวงศ์ รหัส 48380162

นายเทวา ตาเปี้ยสืบ รหัส 48380171

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมไฟฟ้า ภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

ปีการศึกษา 2552

ใบรับรองปริญญานิพนธ์

ชื่อหัวข้อโครงงาน การปรับค่าอัตราการเรียนรู้สำหรับการเรียนรู้แบบออนไลน์ด้วยวิธีเคอร์เนล

ผู้ดำเนินโครงงาน นางสาววารี คำแสน รหัส 48280125
นายสุนทร ศิลปวงศ์ รหัส 48380162
นายเทวา ตาเปี้ยสืบ รหัส 48380171

ที่ปรึกษาโครงงาน ดร. ศุภวรรณ พลพิทักษ์ชัย

สาขาวิชา วิศวกรรมไฟฟ้า

ภาควิชา วิศวกรรมไฟฟ้าและคอมพิวเตอร์

ปีการศึกษา 2552

....

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร อนุมัติให้ปริญญานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมไฟฟ้า

.... ที่ปรึกษาโครงงาน

(ดร. ศุภวรรณ พลพิทักษ์ชัย)

.... กรรมการ

(ดร. นิพัทธ์ จันทรมินทร์)

.... กรรมการ

(ดร. มุฑิตา สงฆ์จันทร์)

ชื่อหัวข้อโครงงาน	การปรับค่าอัตราการเรียนรู้สำหรับการเรียนรู้แบบออนไลน์ ด้วยวิธีเคอร์เนล
ผู้ดำเนินโครงงาน	นางสาววารี คำแสน รหัส 48280125
	นายสุนทร ศิลปวงศ์ รหัส 48380162
	นายเทวา ตาเปี้ยสืบ รหัส 48380171
ที่ปรึกษาโครงงาน	ดร. ศุภวรรณ พลพิทักษ์ชัย
สาขาวิชา	วิศวกรรมไฟฟ้า
ภาควิชา	วิศวกรรมไฟฟ้าและคอมพิวเตอร์
ปีการศึกษา	2552

บทคัดย่อ

ปริญญานิพนธ์ฉบับนี้ได้ศึกษาการปรับค่าอัตราการเรียนรู้สำหรับการเรียนรู้แบบออนไลน์ด้วยวิธีเคอร์เนลซึ่งนำมาประยุกต์ใช้สำหรับหาความสัมพันธ์ในรูปแบบการแก้ไขปัญหาที่สามารถใช้งานที่แตกต่างกัน อาทิ เช่น การประมวลผลสัญญาณ การควบคุม เครื่องกลและการประมาณค่าฟังก์ชัน โดยส่วนใหญ่แล้วเราจะมุ่งเน้นศึกษาการออกแบบระบบหรือการหาฟังก์ชันการประมาณค่าเป็นตัวแทนฟังก์ชันที่เราไม่รู้ค่าจากข้อมูลขาเข้าและขาออกที่เก็บค่าได้โดยวิธีเคอร์เนล เพื่อให้สามารถจำลองฟังก์ชันที่เปลี่ยนแปลงตามเวลาได้ จึงต้องให้อัตราการเรียนรู้มีการปรับค่าตลอดดเวลา

การหาค่าฟังก์ชันด้วยการปรับค่าอัตราการเรียนรู้สำหรับการเรียนรู้แบบออนไลน์ด้วยวิธีเคอร์เนลนี้ เราจะหาจากการหาฟังก์ชันการประมาณค่า ในการที่จะปรับปรุงค่าแอลฟา จะต้องใช้ค่าอัตราการเรียนรู้ซึ่งต้องเป็นค่าที่ถูกปรับตลอดเวลา เพื่อให้ทันต่อฟังก์ชัน ไม่รู้จักค่าที่เปลี่ยนแปลงตลอดเวลา ส่วนประสิทธิภาพของการเรียนรู้จะถูกวัดด้วยค่าความผิดพลาดเฉลี่ยกำลังสอง นอกจากนั้นการทดลองยังแสดงให้เห็นถึงผลของค่าพารามิเตอร์แต่ละตัว เช่น อัตราการเรียนรู้เรกกูลาไรเซชันพารามิเตอร์ ความกว้างของเคอร์เนลฟังก์ชัน ว่ามีผลต่อความผิดพลาดเฉลี่ยกำลังสองอย่างไรบ้าง

Project title	Adaptive Learning Rate in Online Kernel Method
Name	Miss. Waree Khumsan ID. 48280125
	Mr. Suntorn Sinlapawong ID. 48380162
	Mr. Tewa Tapiaseub ID. 48380171
Project advisor	Supawan Phonphitakchai, Ph.D.
Major	Electrical Engineering
Department	Electrical and Computer Engineering
Academic year	2009

....

Abstract

This project studies the methods to adapt learning rate for online learning in kernel method. The learning method can be applied for finding the relation between incoming and outgoing data in form of approximation function. Generally, batch learning is used in learning method but it causes some computational problems because a whole data set is processed in one time. The method of online learning can overcome this problem as the method uses only one data pair in each iteration for updating their parameters thus this method can be applied with huge data set. This project also presents three rules to adjust the learning rate in order to track nonstationary systems. The experiments show that the performance of adaptive learning rates is better than the constant one. The effects of parameters using in learning rules on the performance of learning are also given.

กิตติกรรมประกาศ

โครงงานนี้สำเร็จลุล่วงไปด้วยความกรุณาเป็นอย่างยิ่งจาก ดร. ศุภวรรณ พลพิทักษ์ชัย ซึ่งเป็นอาจารย์ที่ปรึกษาโครงงาน และได้คอยชี้แนะแนวทางตลอดการทำโครงงาน คณะผู้ดำเนินโครงงานขอกราบขอบพระคุณเป็นอย่างสูงและขอระลึกถึงความกรุณาของท่านไว้ตลอดไป

ขอขอบคุณคณาจารย์ทุกท่านที่ประสิทธิ์ประสาทวิชาความรู้ให้กับคณะผู้ดำเนินโครงงาน นอกจากนี้ยังต้องขอขอบคุณภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์ที่ให้ยืมอุปกรณ์และเครื่องมือต่าง ๆ จนทำให้โครงงานนี้สำเร็จลุลวงไปได้

เหนือสิ่งอื่นใด คณะผู้ดำเนินโครงงานขอกราบขอบพระคุณบิดามารดา ผู้มอบความรักเมตตา สติปัญญา รวมทั้งเป็นผู้ให้ทุกสิ่งทุกอย่างตั้งแต่วัยเยาว์จวบจนถึงปัจจุบัน คอยเป็นกำลังใจทำให้ได้รับความสำเร็จอย่างทุกวันนี้ และขอบคุณทุก ๆ คนในครอบครัวของคณะผู้ดำเนินโครงงานที่ไม่ได้กล่าวไว้ ณ ที่นี้ด้วย

นางสาววารี คำแสน

นายสุนทร ศิลปวงศ์

นายเทวา ตาเปี้ยสืบ

สารบัญ

หน้า

สารบัญ (ต่อ)

หน้า

สารบัญรูป

รูปที่ หน้า

บทที่ 1

บทนำ

1.1 ที่มาและความสำคัญของโครงงาน

เครื่องการเรียนรู้ (Machine learning) หรือเครื่องการเรียนรู้หมายถึงการที่กำหนดให้เครื่องคอมพิวเตอร์ สามารถปฏิบัติงานได้ดีขึ้น โดยเรียนรู้จากการกระทำ หรือสิ่งที่ทำไปก่อนหน้านั้น หรืออาจเป็นการเรียนรู้จากการถูกสั่งให้ทำ จากตัวอย่างหรือจากการเปรียบเทียบก็ได้ นอกจากนั้นยังเป็นวิทยาศาสตร์ที่เกี่ยวข้องกับการออกแบบและพัฒนาอัลกอลิทึม (Algorithms) ที่อนุญาตให้ใช้คอมพิวเตอร์ในการเรียนรู้จากข้อมูลเช่นจากเซ็นเซอร์ฐานข้อมูล หลักของการเรียนรู้ในด้านการวิจัยคือเรียนรู้วิธีการที่ซับซ้อนรู้จักรูปแบบ และทำให้ตัดสินใจตามข้อมูล อาจกล่าวได้ว่าการเรียนรู้เป็นการหาความสัมพันธ์ของข้อมูลขาเข้าและขาออกซึ่งจะแสดงในรูปของฟังก์ชันเบื้องหลัง (Underlying function)

ส่วนใหญ่แล้วการเรียนรู้จากข้อมูลจะใช้แบบข้อมูลทั้งหมด (Batch learning) ในการสอน (Train) เพียงครั้งเดียวนั้นคือมีข้อมูลเท่าไรก็จะป้อนเท่านั้น อย่างไรก็ตามวิธีนี้จะมีข้อเสียในกรณีที่ข้อมูลมีจำนวนมาก การสอนแบบนี้จะทำให้คอมพิวเตอร์มีการประมวลผลค่อนข้างที่จะยาก และอาจจะทำให้ระบบหยุดการทำงานได้เลย ดังนั้น เราจึงใช้การเรียนรู้แบบออนไลน์ (Online learning) แทน ซึ่งวิธีนี้จะใช้การข้อมูลเข้าไปทีละตัว โดยจะมีการปรับปรุงแบบจำลองในทุกครั้งของการใส่ข้อมูลซึ่งจะขึ้นอยู่กับค่าความผิดพลาด (Error) ที่เกิดขึ้นว่ามีค่ามากน้อยเพียงใด วิธีสอนแบบนี้จะทำให้คอมพิวเตอร์ของเราทำงานได้อย่างมีประสิทธิภาพ เนื่องจากการป้อนข้อมูลไปทีละตัว นอกจากนั้นยังมีการเรียนรู้ชนิด Supervised learning และ Unsupervised learning

การเรียนรู้แบบมีผู้สอน (Supervised Learning) เป็นเทคนิคหนึ่งของการเรียนรู้ของเครื่อง ซึ่งสร้างฟังก์ชันจากข้อมูลสอน ข้อมูลสอนประกอบด้วยคู่ข้อมูลขาเข้าและข้อมูลขาออก ผลจากการเรียนรู้จะเป็นฟังก์ชัน เรียกวิธีการว่า การถดถอย (Regression) หรือใช้ทำนายประเภทของวัตถุ เรียกว่า การแบ่งประเภท (Classification) เช่น การเรียนรู้เพื่อรู้จำลายมือ

การเรียนรู้แบบไม่มีผู้สอน (Unsupervised Learning) เป็นเทคนิคหนึ่งของการเรียนรู้โดยการสร้างแบบจำลองที่เหมาะสมกับข้อมูล การเรียนรู้แบบนี้แตกต่างจากการเรียนรู้แบบมีผู้สอน คือจะไม่มีการระบุผลที่ต้องการหรือประเภทไว้ก่อน โดยจะพิจารณาวัตถุเป็นเซตของตัวแปร แล้วจึงสร้างแบบจำลองความหนาแน่นร่วมของชุดข้อมูล

ในปี 1980 ได้มีผู้เสนอการเรียนรู้วิธีเคอร์เนล (Kernel method) ซึ่งมีข้อแตกต่างกับวิธีการเรียนรู้แบบเดิมเช่น วิธีระบบโครงข่ายประสาท (Neural Network) หรือโครงข่ายใยประสาทเสมือน (Artificial Neural Network: ANN) คือคอมพิวเตอร์ที่สามารถเลียนแบบการทำงานของสมองมนุษย์

ได้ ด้วยการประมวลผลข้อมูลสารสนเทศ และองค์ความรู้ได้ในคราวละมาก ๆ ข้อดีของวิธีเคอร์เนล คือความสามารถในการหาค่าที่เหมาะสมที่สุด (Optimization) ซึ่งรับรองได้ว่าเป็นค่าต่ำที่สุด (Global minima) ต่างจากวิธีระบบโครงข่ายประสาทซึ่งมักจะมีปัญหาว่าจุดต่ำสุดที่ได้มักจะเป็นค่าที่ต่ำสุดเฉพาะที่ (Local minima) ในปัจจุบันการเรียนรู้วิธีเคอร์เนลถูกนำไปใช้ในงานต่างๆ อาทิ เช่น วิธีเอสวีเอ็ม (SVM) ที่สามารถนำไปใช้วิเคราะห์รูปแบบของความสัมพันธ์ของข้อมูล เช่นกลุ่มการจัดอันดับ องค์ประกอบหลัก ความสัมพันธ์ การจำแนกประเภท เป็นต้น

โครงงานนี้เป็นการศึกษาทฤษฎีของการเรียนรู้แบบออนไลน์ด้วยวิธีเคอร์เนล โดยจะมุ่งเน้นศึกษาการนำไปใช้จำลองระบบจากข้อมูลที่เปลี่ยนแปลงตามเวลา ซึ่งการใช้งานในระบบนี้พารามิเตอร์ที่เรียกว่าอัตราการเรียนรู้ (Learning rate หรือ step size) จะถูกปรับให้มีค่าเปลี่ยนแปลงตลอดเวลา ประสิทธิภาพของการเรียนรู้จะถูกวิเคราะห์โดยโปรแกรมแมทแลป (Mat lab)

1.2 วัตถุประสงค์ของโครงงาน

1) เพื่อศึกษาทฤษฎีของการเรียนรู้แบบออนไลน์ด้วยวิธีเคอร์เนล
2) เพื่อศึกษาการปรับค่าอัตราการเรียนรู้เพื่อให้เหมาะสมกับระบบที่เปลี่ยนแปลงตามเวลา
3) เพื่อศึกษาการใช้งานโปรแกรมแมทแลปและนำความรู้ที่ได้มาสร้างอัลกอลิทึมสำหรับการเรียนรู้

1.3 ขอบเขตของโครงงาน

1) ศึกษาทฤษฎีของการเรียนรู้แบบออนไลน์ด้วยวิธีเคอร์เนล
2) ปรับค่าอัตราการเรียนรู้สำหรับการเรียนรู้ในระบบที่เปลี่ยนแปลงตามเวลา
3) วิเคราะห์ประสิทธิภาพที่ได้จากปรับค่าอัตราการเรียนรู้ในแต่ละวิธี โดยเปรียบเทียบจากค่าความผิดพลาดที่เกิดขึ้น
4) ใช้โปรแกรมแมทแลปช่วยในการคำนวณและแสดงผลเปรียบเทียบ

1.4 ขั้นตอนและแผนการดำเนินงาน

รายละเอียด	ปี2552							ปี2553		
	มิ.ย.	ก.ค.	ส.ค.	ก.ย.	ต.ค.	พ.ย.	ธ.ค.	ม.ค.	ก.พ.	มี.ค.
1.ศึกษาทฤษฎีของการเรียนรู้แบบออนไลน์ด้วยวิธีเคอร์เนล										
2.ปรับค่าอัตราการเรียนรู้สำหรับการเรียนรู้ในระบบที่เปลี่ยนแปลงตามเวลา										
3.วิเคราะห์ประสิทธิภาพที่ได้จากปรับค่าอัตราการเรียนรู้ในแต่ละวิธี										
4.สรุปผลการดำเนินงาน										
5.จัดทำปริญญานิพนธ์ฉบับสมบูรณ์										

1.5 ประโยชน์ที่คาดว่าจะได้รับจากโครงงาน

1) เข้าใจถึงเรื่องทฤษฎีของการเรียนรู้แบบออนไลน์ด้วยวิธีเคอร์เนล
2) เข้าใจถึงทฤษฎีการปรับค่าอัตราการเรียนรู้สำหรับการเรียนรู้ในระบบที่เปลี่ยนแปลงตามเวลา
3) เป็นพื้นฐานในการศึกษาทฤษฎีการปรับค่าอัตราการเรียนรู้สำหรับการเรียนรู้ในระบบที่เปลี่ยนแปลงตามเวลาขั้นสูงต่อไป

1.6 งบประมาณ

1) ค่าถ่ายเอกสารและค่าเข้าเล่มรายงานฉบับสมบูรณ์	เป็นเงิน	1,500	บาท
2) ค่าพิมพ์เอกสาร	เป็นเงิน	500	บาท
3) ค่าวัสดุคอมพิวเตอร์	เป็นเงิน	1,000	บาท
รวมเป็นเงินทั้งสิ้น (สองพันบาทถ้วน)		**3,000**	**บาท**

หมายเหตุ: ถัวเฉลี่ยทุกรายการ

บทที่ 2

การปรับค่าอัตราการเรียนรู้สำหรับการเรียนรู้แบบออนไลน์ด้วยวิธีเคอร์เนล

ในบทนี้จะนำเสนอแนวทางในการแก้ไขปัญหาที่สามารถใช้ในงานแตกต่างกัน อาทิเช่น การประมวลผลสัญญาณ การควบคุม เครื่องกลและการประมาณค่าฟังก์ชัน โดยส่วนใหญ่แล้วเราจะมองถึงปัญหาเหล่านี้ในรูปแบบของการประมาณการระบบที่ไม่รู้จักโดยอาศัยข้อมูลตัวอย่าง ในปัจจุบันนี้วิธีอาร์เคเอชเอส (RKHS – Reproducing kernel Hilbert Spaces) ได้ถูกศึกษาและนำไปใช้กับปัญหาต่าง ๆที่ได้กล่าวมาแล้วดังนั้นในบทนี้จะเป็นการแสดงทฤษฎีที่เกี่ยวข้องกับการแก้ปัญหาด้วย อาร์เคเอชเอสแบบออนไลน์ นั้นคือ เป็นการหาฟังก์ชันการประมาณค่าที่เป็นตัวแทนของระบบที่เราไม่รู้จักจากจำนวนข้อมูลจำกัด ซึ่งสามารถเก็บค่าได้จากระบบนั้นๆ

2.1 การประมาณค่าฟังก์ชันในอาร์เคเอชเอส

สมมติว่าฟังก์ชันไม่รู้ค่า f ซึ่งสามารถสังเกตค่าจำนวนจำกัดได้เป็นส่วนหนึ่งของอาร์เคเอชเอส $\mathcal{F}$ และ f นิยามบนเซต $\mathcal{X}$ สามารถถือได้ว่าเป็นการข้อมูลขาเข้าโดยที่ $x \in \mathcal{X}$ ดังนั้น $f(x)$ จะสามารถแสดงถึงการประมาณค่าของ f ที่ x และเซตข้อมูลขาเข้า x จะถูกมองว่าเป็นซับเซตปริภูมิยูคลิเดียน (Euclidian space) นั้นคือ $\mathcal{X} \subseteq R^n$ โดยที่ x แต่ละตัวเป็นเวกเตอร์มิติ n

เซตจำกัดของการสังเกตหรือข้อมูลขาออกของฟังก์ชัน $\{z_i\}_{i=1}^{N}$ จะสอดคล้องกับการข้อมูลขาเข้า $\{x_i\}_{i=1}^{N}$ โดยเราสามารถกล่าวได้ว่า เซตของข้อมูลขาออกอยู่ในปริภูมิ z ถ้าไม่มีข้อผิดพลาด ค่าการสังเกตจะแสดงได้ดังนี้

$$z_i = L_i f \tag{2.1}$$

เมื่อ $\{L_i\}_{i=1}^{N}$ เป็นเซตของการประเมินผลฟังก์ชันนัลแบบ (Linear evaluation functional) ถ้ากำหนดบน $\mathcal{F}$ ซึ่งสอดคล้องกับ f เราสามารถแสดงค่าทั้งหมดของการสังเกต $\{z_i\}_{i=1}^{N}$ ได้ดังนี้

$$z_i = L_i f = \sum_{i=1}^{N}(L_i f)s_i \tag{2.2}$$

โดยที่ $s_i \subseteq R^n$ เป็นเวกเตอร์มาตรฐาน (Standard basis vector) ที่ลำดับ i โดยในทางปฏิบัติ สามารถเขียนได้เป็น

$$z_i = f(x_i) \tag{2.3}$$

ซึ่งสามารถนำไปใช้ในปัญหาการประมาณค่า

ปัญหาฟังก์ชันการประมาณค่า ซึ่งเป็นตัวแทนฟังก์ชันที่ไม่รู้ค่า สามารถกระทำได้โดยกำหนดคลาสของ $\mathcal{F}$ ของฟังก์ชัน และค่าสังเกต $\{z_i\}_{i=1}^{N}$ ของฟังก์ชันนัล L_i ซึ่งถูกกำหนดบน $\mathcal{F}$ ดังนั้นจะสามารถฟังก์ชัน f บน $\mathcal{F}$ ซึ่งสอดคล้องกับ (2.1) และ (2.3)

ตามหลักแล้วเราสามารถกำหนดอาร์เคเอชเอสให้เป็นปริภูมิฮิลเบิร์ต (Hilbert space) ของฟังก์ชันบน $\mathcal{X}$ ด้วยคุณสมบัติที่ว่าในแต่ละ $x \in \mathcal{X}$ ฟังก์นัล L_i ซึ่งเชื่อมโยง f ด้วย $f(x_i)$ นั่นคือ $L_i \to f(x_i)$ จะเป็นฟังก์ชันนัลแบบเส้นตรงที่ถูกกำหนดขอบเขตและการมีขอบเขตหมายถึงการมีของค่า M โดยที่

$$|L_i f| \;=\; |f(x_i)| \le M||f| \quad \text{สำหรับทุกๆ } f \text{ ใน RKHS}$$

เมื่อ $||\cdot||$ เป็นค่านอร์มในอาร์เคเอชเอส

เมื่อฟังก์ชันนัล L_i ถูกกำหนดขอบเขต ตามทฤษฎีบทของรีซ (Riesz representation theorem) ซึ่งเราสามารถแสดงค่าการสังเกตได้ดังนี้

$$L_i f \;=\; \langle f, k_i \rangle \quad i = 1, \dots, N \tag{2.4}$$

เมื่อ $\langle\cdot,\cdot\rangle$ หมายถึงผลคูณภายใน (Inner product) ใน $\mathcal{F}$ และ $\{k_i\}_{i=1}^{N}$ เป็นชุดของฟังก์ชันที่เรียกว่ารีโพรดิวซิ่งเคอร์เนล (reproducing kernels) ซึ่งแต่เป็นส่วนหนึ่งของ $\mathcal{F}$ และจะไม่ถูกกำหนดซ้ำกันโดยฟังก์ชันนัล L_i

ดังนั้นการหาปัญหาการประมาณจะสามารถกำหนดใหม่ได้อีกแบบนั่นคือ กำหนดปริภูมิฮิลเบิร์ตของเซตของฟังก์ชัน $\{k_i\}_{i=1}^{N} \subset \mathcal{F}$ และค่าการสังเกต $\{z_i\}_{i=1}^{N}$ จากเงื่อนไขดังกล่าวมาแล้วสามารถหาฟังก์ชันการประมาณค่า $f \in \mathcal{F}$ ที่สอดคล้องกับ (2.4) ได้

ในทุกอาร์เคเอชเอสจะมีฟังก์ชันบวกแน่นอน (Positive – definite function) ที่เรียกว่ารีโพรดิวซิ่งเคอร์เนล (k) ที่กำหนดบน $\mathcal{X} \times \mathcal{X}$

(i) $k(\cdot, x') \in \mathcal{F}$; และ

(ii) $\langle f, k(\cdot, x') \rangle_{\mathcal{F}} \;=\; f(x')$

สำหรับทุก f ใน $\mathcal{F}$

จากคุณสมบัติข้างต้นแสดงให้เห็นว่า ฟังก์ชันรีโพรดิวซิ่งเคอร์เนล k_i เป็นฟังก์ชันบน $\mathcal{X} \times \mathcal{X}$ แต่อย่างไรก็ตาม เราสามารถแสดงเคอร์เนลฟังก์ชันรูปแบบของฟังก์ชันบน $\mathcal{X}$ เท่านั้นได้ด้วยการพิจารณาฟังก์ชันเคอร์เนล $k(x, x_i)$ เมื่อ x_i เป็นจุดที่ไม่เปลี่ยนแปลงค่าและ $x_i \in \mathcal{X}$ ดังนั้นการเขียน $k(x, x_i) = k(x_i)$ (หรือ k_i) จะสามารถสรุปได้ว่า $k_i \in \mathcal{F}$ บน $\mathcal{X}$ มีจุด

ศูนย์กลางบน x_i โดยส่วนใหญ่แล้วจุด x_i มักจะถูกกำหนดให้เป็นศูนย์กลางของฟังก์ชันเคอร์เนล และยังเป็นไฮเปอร์พารามิเตอร์ (Hyper parameter) ของฟังก์เคอร์เนลอีกด้วย ซึ่งโดยทั่วไปแล้วศูนย์กลางเหล่านี้จะสอดคล้องกับข้อมูลขาเข้า

ดังนั้นเราจึงสามารถแสดงฟังก์ชันทั่วๆไป f บน อาร์เคเอชเอส $\mathcal{F}$ ด้วยรีโพรดิวซิ่งเคอร์เนล k ได้ดังนี้

$$f(x) = \sum_i \alpha_i k(x, x_i) = \sum_i \alpha_i k_i(x) \tag{2.5}$$

เมื่อ $\alpha_i \in \mathbb{R}, i \in \mathbb{N}$ เพราะฉะนั้นการแก้ปัญหาการประมาณค่าฟังก์ชันจะเปลี่ยนรูปเป็นการประมาณค่าที่เหมาะสมสำหรับตัวแปร α_i ใน (2.5)

เราจะสนใจในการกำหนดนิยามของ RKHS ซึ่งเป็นฟังก์ชันสามารถหาค่าได้ดังนี้

$$f(x) = \sum_{i=1}^{p} \alpha_i k_i(x) \tag{2.6}$$

ที่ $p \in \mathbb{N}$

ฟังก์ชันที่ถือว่าเป็นฟังก์ชันเคอร์เนลมีหลายชนิดที่นิยมใช้ ยกตัวอย่างเช่น ฟังก์ชันเกาเสียนเรเดียลเบสิสที่ใช้ในระบบโครงข่ายประสาทเทียม

$$k(x, x') = \exp\left(-\beta ||x - x'||^2\right) \tag{2.7}$$

โดยที่ $\beta > 0$

ฟังก์ชันโพลิโนเมียลของค่าชุดของค่า x กำลัง d แสดงได้โดย

$$k(x, x') = \sum_{k=0}^{d} \frac{1}{k!\rho_k} (xx')^k \tag{2.8}$$

เมื่อ ρ_k คือเซตของค่าคงที่ที่เรียกว่าค่าน้ำหนักและมีค่าเท่ากับ

$$k(x, x') = (1 + xx')^d \tag{2.9}$$

ฟังก์ชันเคอร์เนลสามารถแสดงได้ด้วยปริภูมิ พาเลย์ – เวียเนอร์ ของฟังก์ชันจำกัดแถบ (Band limited function)

$$k(x, x') = \frac{\sin\pi(x - x')}{\pi(x - x')} \tag{2.10}$$

2.2 การประมาณค่าฟังก์ชันในอาร์เคเอชเอสแบบออนไลน์

ในกรณีนี้ได้ตั้งสมมติฐานว่าในแต่ละรอบของการทำซ้ำ เราจะทราบค่าสังเกตเพียงแค่ค่าเดียว นั่นคือ z_n ดังนั้น

$$L_n f = z_n \tag{2.11}$$

จากนั้นเราจะได้ฟังก์ชันนัล $\hat{g}_{reg}$ ที่เวลา n ที่เป็นฟังก์ชันไม่เป็นค่าลบ โดยที่ $\hat{g}_{reg}: Z \to R$

$$\hat{g}_{reg}(f_n) = \frac{1}{2}||L_{n+1}f_n - z_{n+1}||^2 + \frac{\rho}{2}||f_n||^2 \tag{2.12}$$

ตั้งค่าเริ่มต้น f_0 ซึ่งโดยส่วนใหญ่จะให้มีค่าเท่ากับศูนย์ จากนั้นเราจะหาค่า f_n ที่จะทำให้ (2.12) มีค่าต่ำที่สุดโดยอาศัยวิธีสโตแคสติกเกรเดียนเดสเซนต์ (Stochastic gradient descent,SGD) ที่แสดงโดย

$$f_{n+1} = f_n - \eta_n \nabla \hat{g}_{reg}(f_n) \tag{2.13}$$

เมื่อที่ $\nabla \hat{g}_{reg}(f_n)$ เป็นเกรเดียน ณ เวลาปัจจุบันของ $\hat{g}_{reg}$ เทียบกับ f_n ดังนั้น

$$\begin{aligned}\hat{g}_{reg}(f_n) &= L^*_{n+1}L_{n+1}f_n - L^*_{n+1}z_{n+1} + \rho f_n \\ &= L^*_{n+1}(L_{n+1}f_n - z_{n+1}) + \rho f_n\end{aligned}$$

แทนค่าในสมการที่ (12) จะได้ว่า

$$f_{n+1} = (1-\eta_n\rho)f_n - \eta_n L^*_{n+1}(L_{n+1}f_n - z_{n+1}) \tag{2.14}$$

โดยที่สำหรับค่าคงที่ใดๆ a จะสามารถแสดงได้ว่า $L^*_{n+1}a = k_{n+1}a$ และ $L_{n+1}f_n = f_n(x_{n+1})$ ดังนั้น

$$\nabla \hat{g}_{reg}(f_n) = k_{n+1}[f_n(x_{n+1}) - z_{n+1}] + \rho f_n \tag{2.15}$$

เพราะฉะนั้น (2.13) สามารถเขียนได้เป็น

$$f_{n+1} = (1-\eta_n\rho)f_n - \eta_n k_{n+1}[f_n(x_{n+1}) - z_{n+1}] \tag{2.16}$$

สมมติว่าการประมาณค่าฟังก์ชัน ณ เวลาปัจจุบันมีเคอร์เนลจำนวน p เทอมดังนั้นการฟังก์ชันการประมาณค่าเหล่านี้สามารถเขียนใหม่เป็น

$$f_{n+1}(x) = (1-\eta_n\rho)\sum_{i=1}^{p}\alpha_n^i k_i(x) - \eta_n e_{n+1}k_{n+1}(x) \tag{2.17}$$

เมื่อ α_{n+1}^{i} ถูกคำนวณมาก่อนแล้วดังสมการต่อไปนี้

$$\alpha_{n+1}^{i} = (1-\eta_n\rho)\alpha_n^{i} \quad \text{โดยที่ } i \le p \tag{2.18}$$

และ

$$\alpha_{n+1}^{i} = -\eta_n e_{n+1} \qquad \text{โดยที่ } i = p+1 \tag{2.19}$$

จากสมการที่ผ่านมาเราสามารถสรุปได้ว่า เมื่อค่าน้ำหนัก (α) ค่าใหม่ถูกเพิ่มให้กับแบบจำลอง (f_{n+1}) โดยมีการคูณเข้ากับฟังก์ชันเคอร์เนล ค่าน้ำหนักค่าเก่าจะถูกปรับปรุงโดยการคูณด้วยเทอม $(1-\eta_n\rho)$ ซึ่งหมายถึงการลดลงของค่าน้ำหนักค่านั้นๆ

$$f_{n+1} = \sum_{i=1}^{p+1}(1-\eta_n\rho)^{n+1-i}\,\eta e_i k_i \tag{2.20}$$

2.3 อัตราการเรียนรู้ชนิดปรับตัวได้

หลักการทำงานของวิธีนี้คืออัตราการเรียนรู้จะมีการปรับในทุกๆรอบของการคำนวณ เพื่อที่จะติดตามฟังก์ชันไม่รู้ค่าที่เปลี่ยนแปลงตามเวลาให้ได้ดีที่สุด โดยที่ค่าพารามิเตอร์ ρ จะถูกกำหนดใช้ให้เป็นค่าคงที่ ฟังก์ชันการประมาณค่า f_{n+1} จะแสดงดังนี้

$$f_{n+1} = f_n - \eta_n \frac{\partial \hat{g}_{reg}(f_n)}{\partial f_n} \tag{2.21}$$

การปรับอัตราการเรียนรู้จะมี 3 วิธี อาศัยแนวคิดของการปรับค่าอัตราการเรียนรู้ในงานด้านสโตแคสติกออพติไมเซชัน(stochastic optimisation)

2.3.1 การปรับอัตราการเรียนรู้วิธีที่ 1

สมมติว่าฟังก์ชันไม่รู้ค่าของเรามีการเปลี่ยนแปลงอย่างช้าๆ ดังนั้นเราจึงต้องการให้อัตราการเรียนรู้มีการปรับตัวอย่างช้าๆด้วย โดยกระบวนการปรับค่าอัตราการเรียนรู้ คือ

$$\eta_n = \eta_{n-1} - \gamma \frac{\partial \hat{g}_{reg}(f_{n+1})}{\partial \eta_n} \tag{2.22}$$

เมื่อ $\gamma \in \mathbb{R}^+$ เป็นค่าแฟคเตอร์ของอัตราการเรียนรู้ที่ปรับตัวได้ เมื่อ

$$\frac{\partial \hat{g}_{reg}(f_{n+1})}{\partial \eta_n} = \frac{\partial \hat{g}_{reg}(f_{n+1})}{\partial f_{n+1}} \cdot \frac{\partial f_{n+1}}{\partial \eta_n} \tag{2.23}$$

จะได้

$$\frac{\partial f_{n+1}}{\partial \eta_n} = \frac{\partial\left\{f_n - \eta_n \frac{\partial \hat{g}_{reg}(f_n)}{\partial f_n}\right\}}{\partial \eta_n}$$

$$= -\frac{\partial \hat{g}_{reg}(f_n)}{\partial f_n} \qquad (2.24)$$

แทนค่าลงในสมการ (2.23) จะได้

$$\frac{\partial \hat{g}_{reg}(f_{n+1})}{\partial \eta_n} = -\frac{\partial \hat{g}_{reg}(f_{n+1})}{\partial f_{n+1}} \cdot \frac{\partial \hat{g}_{reg}(f_n)}{\partial f_n} \qquad (2.25)$$

ใช้สมการข้างบนแทนค่าในสมการที่ (2.22) ดังนั้น

$$\eta_n = \eta_{n-1} - \gamma \left\langle \frac{\partial \hat{g}_{reg}(f_{n+1})}{\partial f_{n+1}}, \frac{\partial \hat{g}_{reg}(f_n)}{\partial f_n} \right\rangle \qquad (2.26)$$

สมการข้างต้นเรียกว่ากฎการปรับตัว (update rule) ซึ่งจะมีการใช้ค่าที่เวลา $n+1$ ที่เป็นค่าในอนาคตไม่สามารถหาได้ เราจะใช้สมมติฐานที่กล่าวมาแล้วว่าระบบที่เราสนใจมีการเปลี่ยนแปลงอย่างช้าๆ ดังนั้นจึงสามารถใช้ค่าที่เวลาก่อนหน้านี้ในสมการ (2.28) ได้แสดงดังสมการต่อไปนี้

$$\eta_n = \eta_{n-1} - \gamma \left\langle \frac{\partial \hat{g}_{reg}(f_{n+1})}{\partial f_{n+1}}, \frac{\partial \hat{g}_{reg}(f_n)}{\partial f_n} \right\rangle$$

$$= \eta_{n-1} + \gamma \langle \nabla \hat{g}_{reg}(f_n), \nabla \hat{g}_{reg}(f_{n-1}) \rangle \qquad (2.27)$$

ในการหาค่า η_n จะต้องอาศัยการคำนวณผลคูณภายในของ $\langle \nabla \hat{g}_{reg}(f_n), \nabla \hat{g}_{reg}(f_{n-1}) \rangle$ โดยอาศัยสมการดังต่อไปนี้

$$\nabla \hat{g}_{reg}(f_n) = L^*_{n+1}(f_n(x_{n+1}) - z_{n+1})\rho f_n$$

$$= \eta_{n-1} + \gamma \langle \nabla \hat{g}_{reg}(f_n), \nabla \hat{g}_{reg}(f_{n-1}) \rangle \qquad (2.28)$$

และ

$$\nabla \hat{g}_{reg}(f_{n-1}) = L^*_n(f_{n-1}(x_n) - z_n) + \rho f_{n-1}$$

$$= L^*_{n+1} e_{n+1} + \rho f_n \qquad (2.29)$$

ซึ่งจะได้ผลคูณภายใน

$$\langle \nabla \hat{g}_{reg}(f_n), \nabla \hat{g}_{reg}(f_{n-1}) \rangle = \langle L^*_{n+1} e_{n+1} + \rho f_n, L^*_n e_n + \rho f_{n-1} \rangle$$

$$= \langle L^*_n e_n, L^*_{n+1} e_{n+1} \rangle + \langle L^*_n e_n, \rho f_n \rangle$$

$$+ \langle \rho f_{n-1},, L^*_{n+1} e_{n+1} \rangle + \rho^2 \langle f_{n-1}, f_n \rangle \qquad (2.30)$$

สามารถเขียนได้เป็น

$$\langle \nabla \hat{g}_{reg}(f_n), \nabla \hat{g}_{reg}(f_{n-1}) \rangle = \langle L_{n+1} L_n^* e_n, e_{n+1} \rangle + \rho \langle e_n, L_n f_n \rangle$$
$$+\rho \langle L_{n+1} f_{n-1}, e_{n+1} \rangle + \rho^2 \langle f_{n-1}, f_n \rangle \quad (2.31)$$

แปลงอยู่ในรูปเมตริกซ์ได้ดังนี้

$$\langle \nabla \hat{g}_{reg}(f_n), \nabla \hat{g}_{reg}(f_{n-1}) \rangle = e_n k(x_n, x_{n+1}) + \rho\big(e_n f_n(x_n) + e_{n+1} f_{n-1}(x_{n+1})\big)$$
$$+ \rho^2 \alpha_{n-1}^T K_{n-1,n} \alpha_n \quad (2.32)$$

เมื่อ $e_{n+1} = f_n(x_{n+1}) - z_{n+1}, e_n = f_n(x_n) - z_n, K_{n-1,n} \in \mathbb{R}^{n \times n+1}, K_{ij} = k(x_n, x_j)$ และ α_{n-1}, α_n เป็นเวกเตอร์ของพารามิเตอร์ที่เวลา $n-1$ และ n ตามลำดับ

2.3.2การปรับอัตราการเรียนรู้วิธีที่ 2

การปรับค่าด้วยวิธีการนี้ ค่า η_n จะถูกทำให้มีค่ามากๆ ได้อย่างรวดเร็วโดยอาศัยการปรับค่าด้วยวิธีเรขาคณิต นั่นคือสามารถกระทำได้โดยการเลือกค่าแฟคเตอร์เป็น $\gamma\eta_{n-1}$ ดังนั้นเราจะได้กฎการปรับตัว

$$\eta_n = \eta_{n-1} + \gamma \eta_{n-1} \langle \nabla \hat{g}_{reg}(f_n), \nabla \hat{g}_{reg}(f_{n-1}) \rangle$$
$$= \eta_{n-1} \{1 + \gamma \langle \nabla \hat{g}_{reg}(f_n), \nabla \hat{g}_{reg}(f_{n-1}) \rangle\} \quad (2.33)$$

อย่างไรก็ตามวิธีการปรับค่านี้ค่อนค้างจะขึ้นกับฟังก์ชั่น $\nabla \hat{g}_{reg}(f_n)$ มาก ดังนั้นวิธีต่อไปจะเป็นการกำจัดการขึ้นกับฟังชันที่ได้กล่าวมาแล้ว

2.3.3การปรับอัตราการเรียนรู้วิธีที่ 3

ปกติการปรับจะไม่ขึ้นอยู่กับการอนุพันธ์บางส่วนซึ่ง $\hat{g}_{reg}$ เป็นผลมาจากการใช้อัตราการเรียนรู้แบบสมดุล $\frac{\gamma\eta_{n-1}}{u_n}$ ที่ μ_n ได้มาโดย

$$u_n = \mu u_{n-1} + (1-\mu)\|\nabla \hat{g}_{reg}(f_n)\|^2 \quad (2.34)$$

รวมทั้ง μ_0 เป็นค่าเริ่มต้น รูปแบบของเมตริก $\|\nabla \hat{g}_{reg}(f_n)\|^2$ มีข้อกำหนดโดย $\langle \nabla \hat{g}_{reg}(f_n) = L_{n+1}^* e_{n+1} + \rho f_n \rangle$ ตามที่

$$
\begin{aligned}
\|\nabla \hat{g}_{reg}(f_n)\|^2 &= \langle \nabla \hat{g}_{reg}(f_n), \nabla \hat{g}_{reg}(f_n)\rangle \\
&= \langle L_{n+1}^{*} e_{n+1} + \rho f_n, L_{n+1}^{*} e_{n+1} + \rho f_n \rangle \\
&= \langle L_{n+1}^{*} e_{n+1}, L_{n+1}^{*} e_{n+1}\rangle + 2\langle L_{n+1}^{*} e_{n+1}, \rho f_n\rangle + \langle \rho f_n, \rho f_n\rangle \\
&= \langle L_{n+1} L_{n+1}^{*} e_{n+1}, e_{n+1}\rangle + 2\rho\langle e_{n+1}, L_{n+1} f_n\rangle + \rho^2 \langle f_n, f_n\rangle \\
&= k(x_{n+1}, x_{n+1}) e_{n+1}^2 + 2 e_{n+1} \rho f_n(x_{n+1}) + \rho^2 \alpha_n^T K_{n,n} \alpha_n \qquad (2.35)
\end{aligned}
$$

ปัจจุบันเรากำหนดกฎการปรับค่าใหม่เหมือนกฎการปรับค่าตามปกติโดย

$$
\eta_n = \eta_{n-1} \{1 + \frac{\gamma}{u_n} \langle \nabla \hat{g}_{reg}(f_n), \nabla \hat{g}_{reg}(f_{n-1})\rangle\} \qquad (2.36)
$$

บทที่ 3
การออกแบบฟังก์ชันการประมาณค่าที่เปลี่ยนแปลงตามเวลา
ด้วยวิธีการปรับค่าอัตราการเรียนรู้

โครงงานนี้ได้มุ่งเน้นศึกษาการออกแบบระบบหรือการหาฟังก์ชันการประมาณค่าคงเป็นที่ตัวแทนฟังก์ชันไม่รู้ค่าจากข้อมูลขาเข้าและขาออกที่เก็บค่าได้โดยใช้วิธีเคอร์เนล เพื่อให้สามารถจำลองฟังก์ชันที่เปลี่ยนแปลงตามเวลาได้ จึงต้องให้อัตราการเรียนรู้มีการปรับค่าตลอดเวลาดังทฤษฎีที่เสนอไปแล้วในบทที่ 2

3.1 หลักการออกแบบระบบด้วยวิธีการประมาณค่าฟังก์ชัน

การประมาณค่าฟังก์ชันไม่รู้ค่าที่เปลี่ยนแปลงตามเวลาด้วยวิธีเคอร์เนลจะมีขั้นตอนดังต่อไปนี้

1. กำหนดค่าเริ่มต้นค่า β , η , ρ และฟังก์ชัน $f_0 = 0$
2. กำหนดค่า n เริ่มที่ $n = 0$
3. หาค่าความผิดพลาดจากสมการ $e_{n+1} = f_n(x_{n+1} - z_{n+1})$
4. หาค่า α จากสมการที่ (2.18) และ (2.19) โดยที่อัตราการเรียนรู้ที่ใช้ในสมการดังกล่าวมีการปรับค่า 4 วิธี
5. หาค่า k จาก สมการที่ (2.7)
6. หาค่าฟังก์ชันจากสมการ $f_{n+1}(x) = \sum_{i=1}^{n+1} \alpha_{n+1}^{i} k_i(x)$
7. เพิ่มค่า n ทำซ้ำตามขั้นตอนที่ 3 ถึง 6

3.2 วิธีการปรับค่าอัตราการเรียนรู้

จากขั้นตอนการหาฟังก์ชันการประมาณค่า ในการที่จะปรับปรุงค่า α ในข้อที่ 4 จะต้องใช้ค่าอัตราการเรียนรู้ซึ่งต้องเป็นค่าที่ถูกปรับตลอดเวลาเพื่อให้ทันต่อฟังก์ชันไม่รู้ค่าที่เปลี่ยนแปลงตลอดเวลา ซึ่งจะมี 3 วิธีที่ใช้

3.2.1 การปรับอัตราการเรียนรู้แบบคงที่

การปรับตัววิธีนี้จะใช้อัตราเรียนรู้คงที่ตลอดเวลาแสดงดังสมการ (2.18) และ (2.19)

3.2.2 การปรับอัตราการเรียนรู้วิธีที่ 1

กฎการปรับตัวจะแสดงดังสมการ (2.27) นั่นคือ

$$\begin{aligned}\eta_n &= \eta_{n-1} - \gamma \langle \frac{\partial \hat{g}_{reg}(f_{n+1})}{\partial f_{n+1}}, \frac{\partial \hat{g}_{reg}(f_n)}{\partial f_n} \rangle \\ &= \eta_{n-1} + \gamma \langle \nabla\, \hat{g}_{reg}(f_n), \nabla\, \hat{g}_{reg}(f_{n-1}) \rangle\end{aligned}$$

3.2.3 การปรับอัตราการเรียนรู้วิธีที่ 2

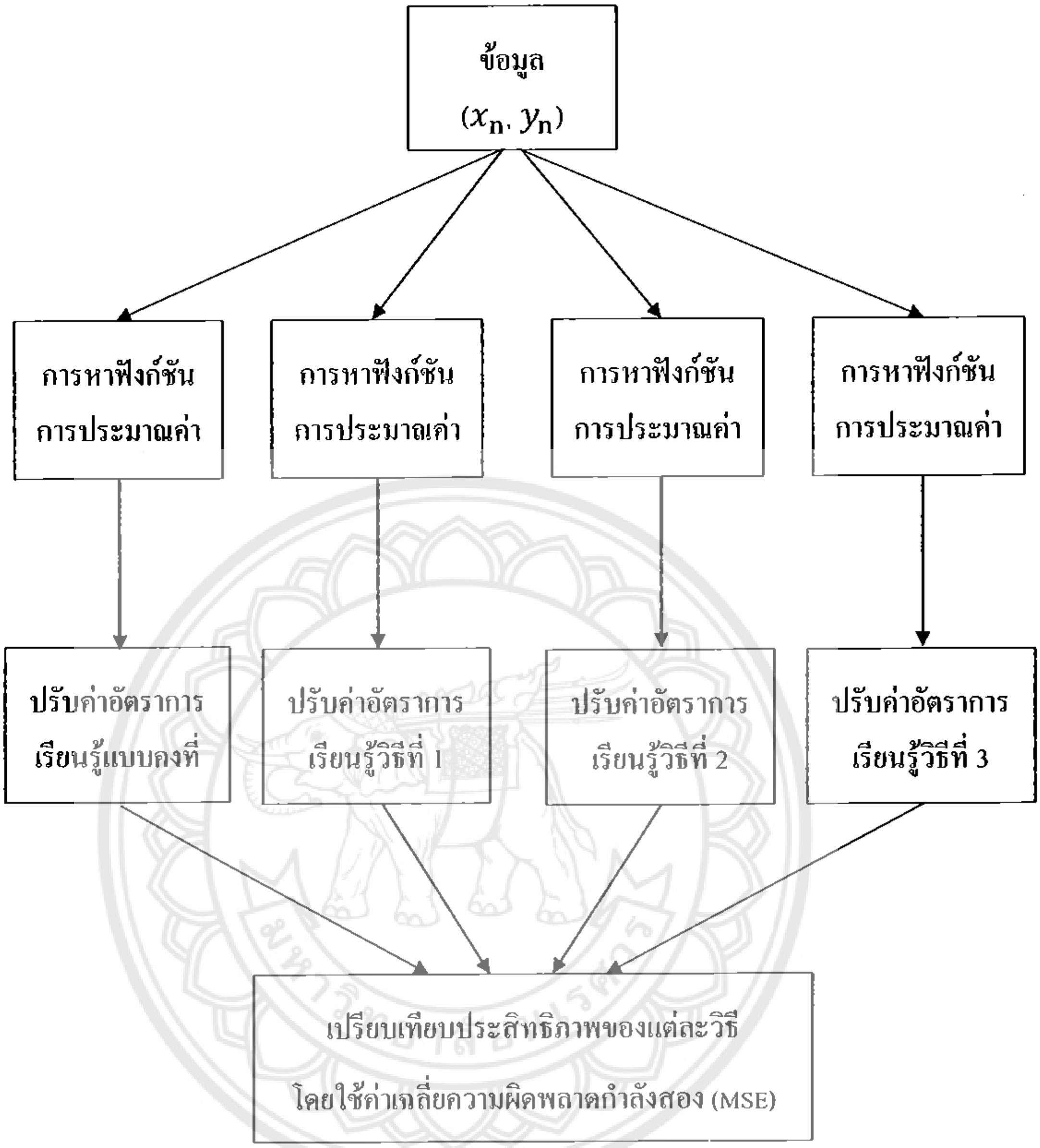
กฎการปรับตัวจะแสดงดังสมการ (2.33) นั่นคือ

$$\begin{aligned}\eta_n &= \eta_{n-1} + \gamma \eta_{n-1} \langle \nabla\, \hat{g}_{reg}(f_n), \nabla\, \hat{g}_{reg}(f_{n-1}) \rangle \\ &= \eta_{n-1}\{1 + \gamma \langle \nabla\, \hat{g}_{reg}(f_n), \nabla\, \hat{g}_{reg}(f_{n-1}) \rangle\}\end{aligned}$$

3.2.4 การปรับอัตราการเรียนรู้วิธีที่ 3

กฎการปรับตัวจะแสดงดังสมการ (2.36) นั่นคือ

$$\begin{aligned}\eta_n &= \eta_{n-1}\{1 + \frac{\gamma}{u_n} \langle \nabla \hat{g}_{reg}(f_n), \nabla \hat{g}_{reg}(f_{n-1}) \rangle\} \qquad \text{และ} \\ u_n &= \mu u_{n-1} + (1-\mu) \|\nabla \hat{g}_{reg}(f_n)\|^2\end{aligned}$$



รูปที่ 3.1 ขั้นตอนแสดงการทดลองการหาค่าความผิดพลาดของแต่ละวิธี

บทที่ 4

ผลการทดลองและการวิเคราะห์ผล

ในบทนี้จะกล่าวถึงผลการทดลองของวิธีการประมาณค่าฟังก์ชันด้วยวิธีเคอร์เนลแบบอัตราการเรียนรู้คงที่และวิธีการปรับค่าอัตราการเรียนรู้ที่เปลี่ยนแปลงตามเวลาที่ได้ศึกษามา ซึ่งจะทำการเปลี่ยนค่าพารามิเตอร์ต่างๆและดูผลที่เกิดขึ้น โดยแต่ละค่าพารามิเตอร์ที่แตกต่างกันจะให้ผลของค่าความผิดพลาด (ค่าความผิดพลาดเฉลี่ยกำลังสองหรือ MSE) ที่แตกต่างกันออกไป แต่สิ่งที่ต้องการคือมีการลดลงของค่าความผิดพลาดและการลดลงต้องค่อยๆลดลงซึ่งเป็นที่ต้องการ

4.1 ผลการทดลองการประมาณค่าฟังก์ชันด้วยวิธีเคอร์เนล

การประมาณค่าฟังก์ชันด้วยวิธีเคอร์เนลนั้นสามารถทำได้โดยใช้สมการที่ (2.20) และจะใช้ฟังก์ชันแกาเสียนเรเดียลเบสิสดังสมการที่ (2.7) เป็นฟังก์ชันเคอร์เนล ในการทดลองการประมาณค่าฟังก์ชัน จะมีขั้นตอนตามที่ได้กล่าวมาแล้วในบทที่ 3 หัวข้อที่ 3.1 ซึ่งจะใช้โปรแกรมแมทแลปในการประมาณค่าฟังก์ชัน และคำนวณหาค่าความผิดพลาดจากข้อมูลที่ใช้ทดสอบได้ดังสมการต่อไปนี้

$$\text{ค่าความผิดพลาด (MSE)} = \frac{e_1^2+e_2^2+e_n^2}{n} \tag{4.1}$$

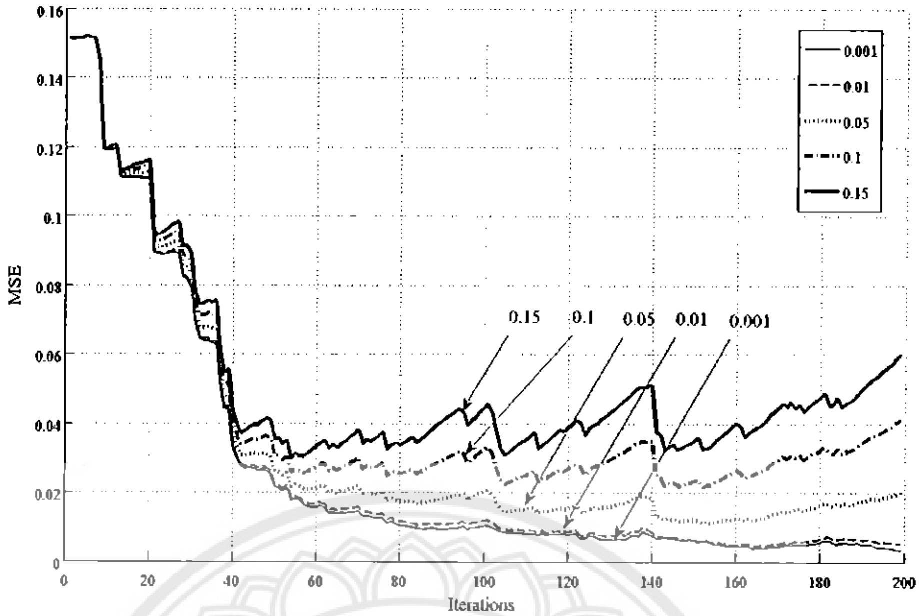
เมื่อ n คือจำนวนของจำนวนข้อมูลของค่าความผิดพลาด

ข้อมูลที่ใช้ทดสอบนี้ถูกสร้างขึ้นโดยใช้โปรแกรมแมทแลป ซึ่งแบ่งข้อมูลออกเป็น 2 ชุด คือที่ใช้ชุดสอนจำนวน 200 ค่า และชุดที่ใช้ทดสอบจำนวน 50 ค่า

ในการทดลองนี้จะมีการปรับค่าพารามิเตอร์ต่างๆให้ได้ค่าพารามิเตอร์ที่ดีที่สุด เพื่อหาค่าความผิดพลาด ซึ่งจะมีหลักการในการเลือกโดยพิจารณาจากค่าความผิดพลาดที่น้อยที่สุดของระบบ โดยจะมีการปรับค่าพารามิเตอร์ดังนี้

4.1.1 ผลของการปรับค่าเรกกูลาไรเซชัน

ในการปรับค่าเรกกูลาไรเซชันนั้นจากการทดสอบค่าอัตราการเรียนรู้เราจะได้ค่าอัตราการเรียนรู้ที่มีค่าความผิดพลาดที่ดีที่สุด คือ กราฟในช่วงแรกจะมีค่าลดลงอย่างฉับพลัน แล้วจากนั้นจะค่อย ๆ ลดลงจนมีลักษณะที่คงที่มีค่าอัตราการเรียนรู้เท่ากับ 0.1 และค่าเบต้าเท่ากับ 100 แล้วทำการปรับค่าเรกกูลาไรเซชัน



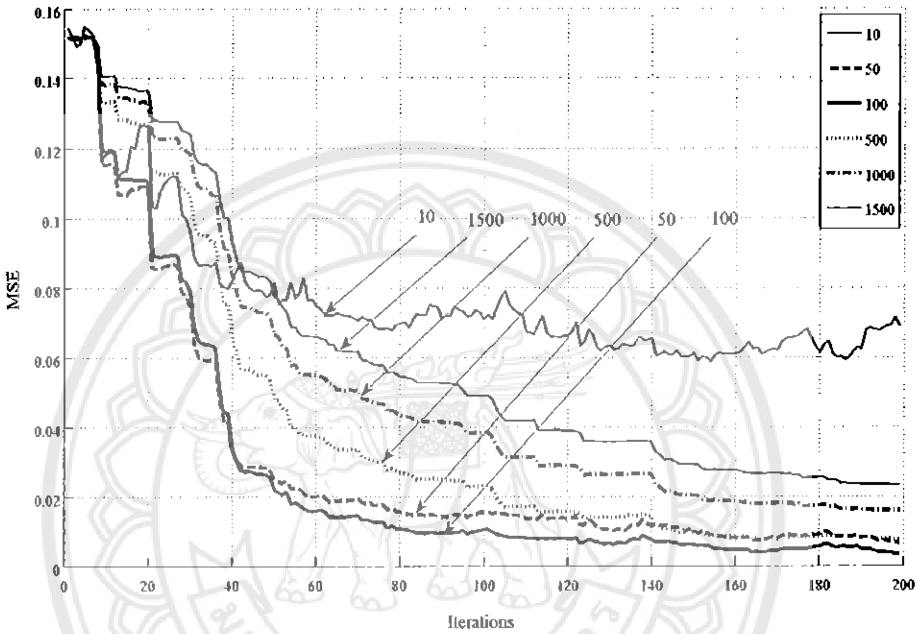
รูปที่ 4.1 การเปรียบเทียบค่าความผิดพลาดเนื่องจากการปรับค่าเรกกูลาไรเซชัน

จากรูปที่ 4.1 ค่าความผิดพลาดที่ได้จากการปรับค่าเรกกูลาไรเซชันต่าง ๆ จะเห็นว่าค่าความผิดพลาดที่ ρ = 0.001 มีลักษณะการลดลงของค่าความผิดพลาดที่ดีที่สุด โดยในช่วงแรกจะลดลงอย่างฉับพลันตั้งแต่รอบที่ 0-50 แล้วหลังจากนั้นจะค่อย ๆ ลดลงจนค่าความผิดพลาดนั้นมีลักษณะคงที่ ส่วนที่ค่า ρ = 0.01, 0.05, 0.1 และ 0.15 ค่าความผิดพลาดมีลักษณะที่ไม่ดี คือในช่วงแรกค่าความผิดพลาดจะลดลงอย่างฉับพลัน แล้วหลังจากนั้นจะค่อย ๆ เพิ่มขึ้นอีก ดังรูปที่ 4.1

4.1.2ผลของการปรับค่าเบต้า

ในการปรับค่าเบต้านั้นเราจะกำหนดให้ค่าเรกกูลาไรเซชันเท่ากับ 0.001 และค่าอัตราการเรียนรู้เท่ากับ 0.1 โดยสองค่านี้จะให้มีค่าคงที่ แล้วทำการปรับค่าเบต้า

ผลที่ได้จากการปรับค่าค่าเบต้า แล้วหาค่าความผิดพลาดจากสมการที่ (4.1) สามารถแสดงได้ดังรูปที่ 4.2



รูปที่ 4.2 กราฟแสดงการเปรียบเทียบค่าความผิดพลาดเนื่องจากการปรับค่าเบต้า

จากรูปที่ 4.2 ค่าความผิดพลาดที่ได้จากการปรับค่าเบต้าต่าง ๆ จะเห็นว่าค่าความผิดพลาดที่ β = 100 มีการลดลงของค่าความผิดพลาดที่ดี โดยในช่วงแรกจะลดลงอย่างฉับพลันตั้งแต่รอบที่ 0–50 แล้วหลังจากนั้นจะค่อย ๆ ลดลงจนค่าความผิด พลาดนั้นมีลักษณะคงที่ ส่วนที่ค่า β = 10, 50, 500, 1000 และ 1500 ค่าความผิดพลาดที่ได้จะมีค่ามากและการลดลงของค่าความผิดพลาดไม่ดี ดังรูปที่ 4.2

4.2 ค่าความผิดพลาดของการประมาณค่าฟังก์ชันด้วยวิธีการปรับค่าอัตราการเรียนรู้

4.2.1 การปรับค่าอัตราการเรียนรู้วิธีที่ 1

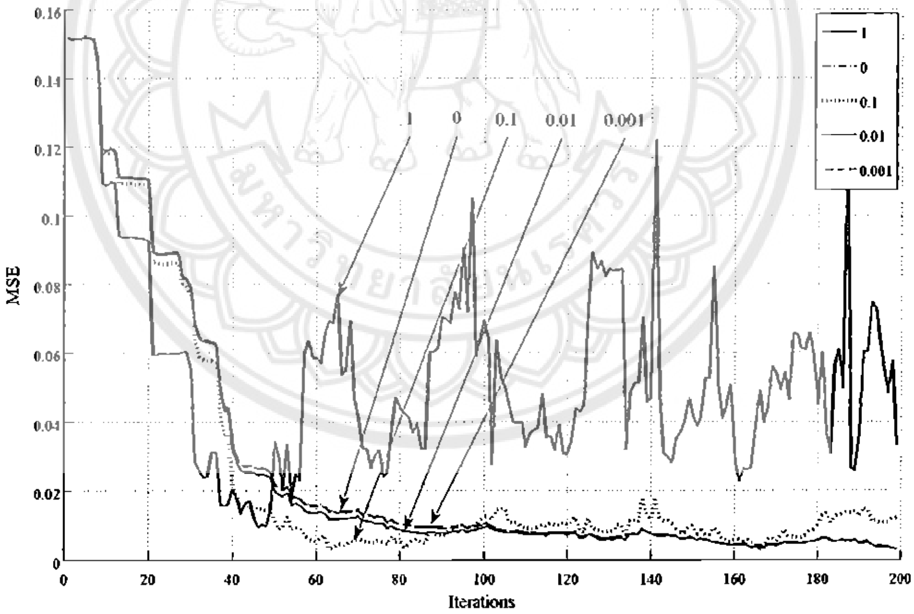
กฎการปรับตัวจะแสดงดังสมการ (2.27) ข้อมูลที่ใช้จะมีชุดสอนและชุดทดสอบซึ่งในที่นี้ชุดสอนจะมี 200 ค่าและชุดทดสอบมี 50 ค่าและจะกำหนดให้ค่าอัตราการเรียนรู้เท่ากับ

$$\eta_n = \eta_{n-1} - \gamma \left\langle \frac{\partial \hat{g}_{reg}(f_{n+1})}{\partial f_{n+1}}, \frac{\partial \hat{g}_{reg}(f_n)}{\partial f_n} \right\rangle$$

$$= \eta_{n-1} + \gamma \langle \nabla \hat{g}_{reg}(f_n), \nabla \hat{g}_{reg}(f_{n-1}) \rangle$$

เมื่อ ค่าแกมมา (γ) เป็นค่าแฟคเตอร์ของอัตราการเรียนรู้ที่ปรับตัวได้ เพื่อให้ค่าอัตราการเรียนรู้มีการเปลี่ยนแปลงตลอดเวลาอย่างช้า ๆ

ในการทดลองนี้เราจะให้ค่าอัตราการเรียนรู้เริ่มต้นเท่ากับ 0.1 ค่าเรกกูลาไรเซชัน 0.001 ค่าเบต้าเท่ากับ 100 และเราจะทำการปรับค่าแกมมา เพื่อให้ค่าเฉลี่ยความผิดพลาดมีค่าน้อยที่สุด สามารถแสดงได้ดังรูปที่ 4.3



รูปที่ 4.3 การเปรียบเทียบค่าความผิดพลาดเนื่องจากการปรับค่าแกมมา

จากรูปที่ 4.3 ค่าความผิดพลาดที่ได้จากการปรับค่าแกมมา จะเห็นว่าค่าความผิดพลาดที่ γ = 0.01 มีการลดลงของค่าความผิดพลาดที่ดี โดยในช่วงแรกจะลดลงอย่างฉับพลันตั้งแต่รอบที่ 0–50 แล้วหลังจากนั้นจะค่อย ๆ ลดลงจนค่าความผิดพลาดนั้นมีลักษณะคงที่ ส่วนที่ค่า γ = 1, 0, 0.1 และ 0.001 ค่าความผิดพลาดที่ได้จะมีค่ามากและการลดลงของค่าความผิดพลาดไม่ดี

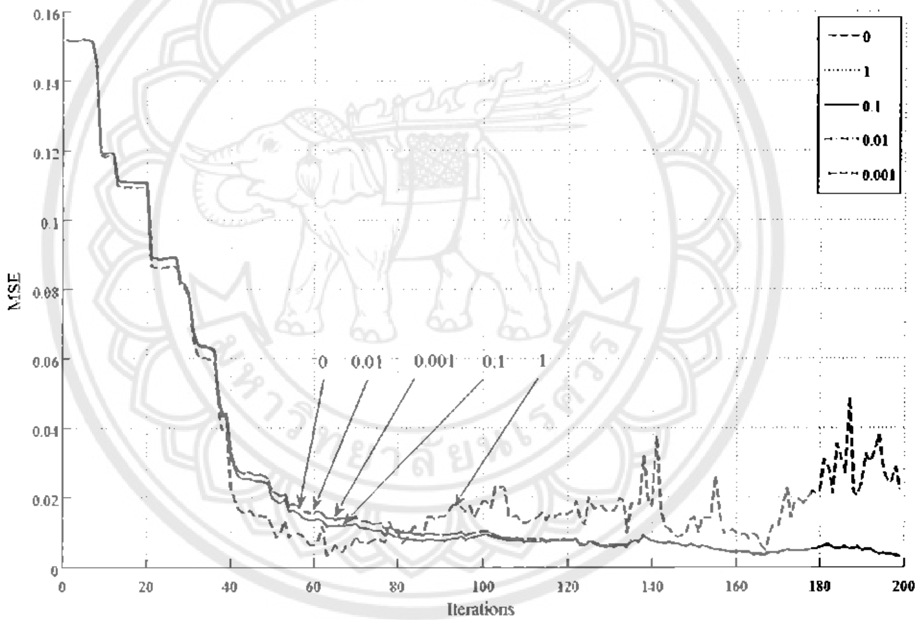
4.2.2 การปรับค่าอัตราการเรียนรู้วิธีที่ 2

กฎการปรับตัวจะแสดงดังสมการ (2.33) และจะกำหนดให้ค่าอัตราการเรียนรู้เท่ากับ

$$\eta_n = \eta_{n-1} + \gamma\eta_{n-1}\langle\nabla\,\hat{g}_{reg}(f_n), \nabla\,\hat{g}_{reg}(f_{n-1})\rangle$$
$$= \eta_{n-1}\{1 + \gamma\langle\nabla\,\hat{g}_{reg}(f_n), \nabla\,\hat{g}_{reg}(f_{n-1})\rangle\}$$

เมื่อค่าแกมมาเป็นค่าแฟคเตอร์ของอัตราการเรียนรู้ที่ปรับตัวได้ เพื่อให้ค่าอัตราการเรียนรู้มีการเปลี่ยนแปลงตลอดเวลาอย่างรวดเร็ว

ผลที่ได้จากการปรับค่าแกมมา แล้วหาค่าความผิดพลาดจากสมการที่ (4.1) โดยเราจะให้ค่าอัตราการเรียนรู้เริ่มต้นเท่ากับ 0.1 ค่าเรกกูลาไรเซชัน 0.001 ค่าเบต้าเท่ากับ 100 จากนั้นจะทำการปรับค่าแกมมา เพื่อให้ค่าเฉลี่ยความผิดพลาดมีค่าน้อยที่สุด สามารถแสดงได้ดังรูปที่ 4.4



รูปที่ 4.4 การเปรียบเทียบค่าความผิดพลาดเนื่องจากการปรับค่าแกมมา

จากรูปที่ 4.4 ค่าความผิดพลาดที่ได้จากการปรับค่าแกมมา จะเห็นว่าค่าความผิดพลาดที่ γ = 0.1 มีการลดลงของค่าความผิดพลาดที่ดี โดยในช่วงแรกจะลดลงอย่างฉับพลันตั้งแต่รอบที่ 0–50 แล้วหลังจากนั้นจะค่อย ๆ ลดลงจนค่าความผิดพลาดนั้นมีลักษณะคงที่ ส่วนที่ค่า γ = 1, 0, 0.01 และ 0.001 ค่าความผิดพลาดที่ได้จะมีค่ามากและการลดลงของค่าความผิดพลาดไม่ดี

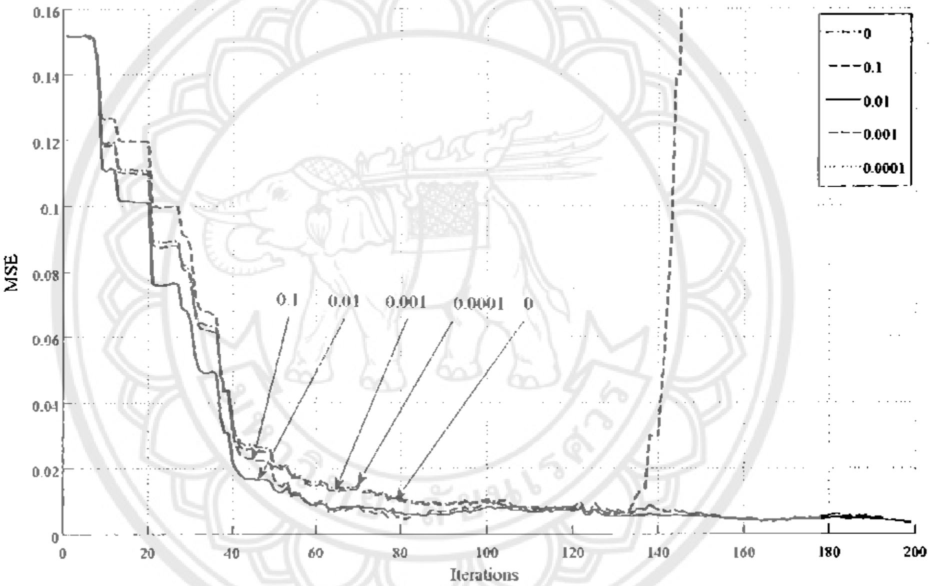
4.2.3การปรับค่าอัตราการเรียนรู้วิธีที่ 3

กฎการปรับตัวจะแสดงดังสมการ (2.36) และจะกำหนดให้ค่าอัตราการเรียนรู้เท่ากับ

$$\eta_{\mathrm{n}} = \eta_{n-1}\{1 + \frac{\gamma}{u_n}\langle \nabla \hat{g}_{reg}(f_n), \nabla \hat{g}_{reg}(f_{n-1})\rangle\}$$

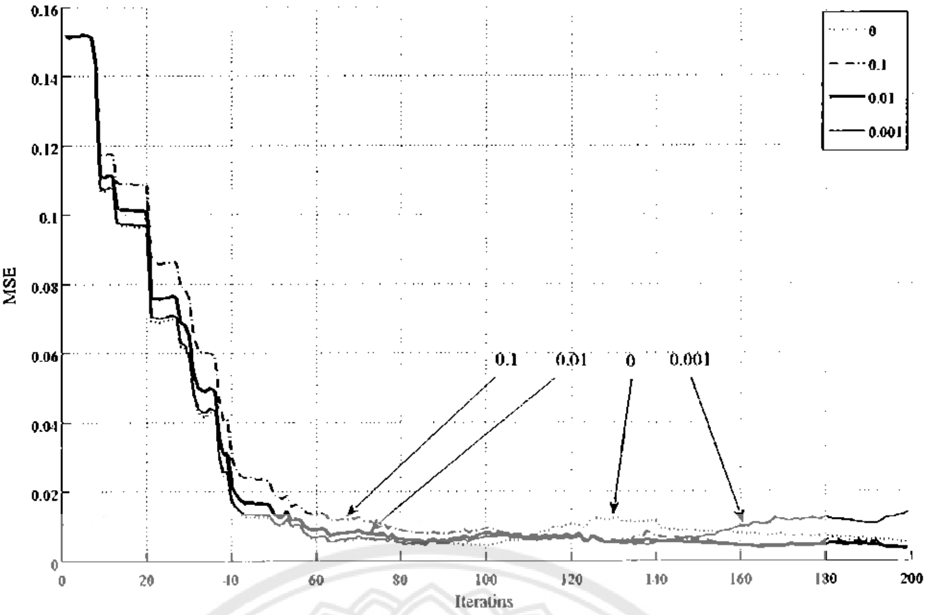
$$u_{\mathrm{n}} = \mu u_{n-1} + (1-\mu)\|\nabla \hat{g}_{reg}(f_n)\|^2$$

ผลที่ได้จากการปรับค่าแกมมา และค่ามิว (μ) แล้วหาค่าความผิดพลาดจากสมการที่ (4.1) โดยเราจะให้ค่าอัตราการเรียนรู้เริ่มต้นเท่ากับ 0.1 ค่าเรกกูลาไรเซชัน 0.001 ค่าเบต้าเท่ากับ 100 และเราจะทำการปรับค่าแกมมา เพื่อให้ค่าเฉลี่ยความผิดพลาดมีค่าน้อยที่สุด สามารถแสดงได้ดังรูปที่ 4.5



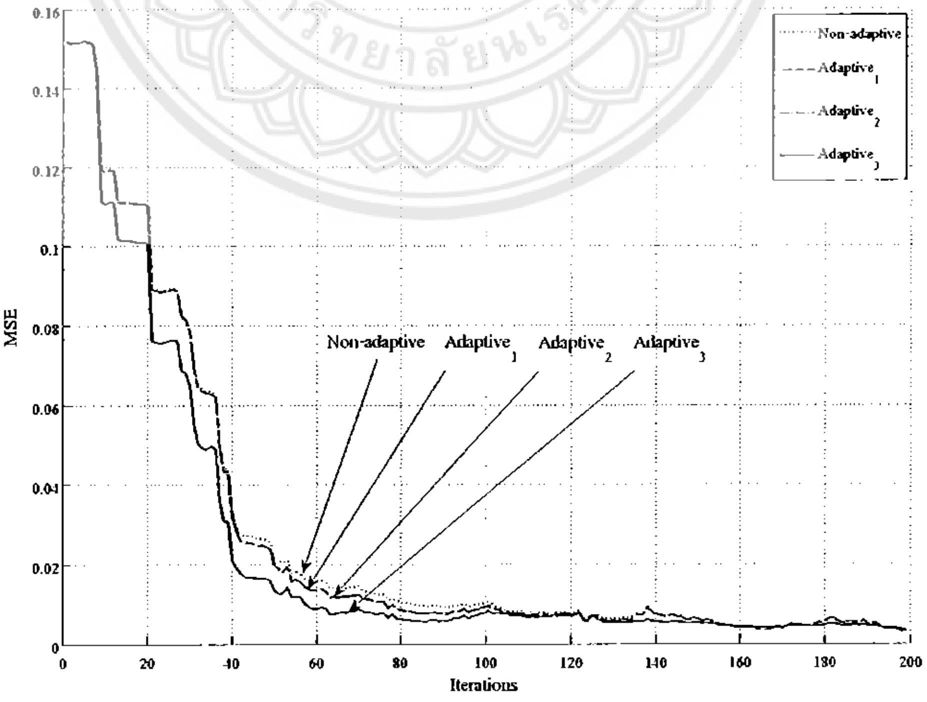
รูปที่ 4.5 การเปรียบเทียบค่าความผิดพลาดเนื่องจากการปรับค่าแกมมา (ค่ามิวคงที่เท่ากับ 0.01)

จากรูปที่ 4.5 ค่าความผิดพลาดที่ได้จากการปรับค่าแกมมา จะเห็นว่าค่าความผิดพลาดที่ μ = 0.01 และγ = 0.01 มีการลดลงของค่าความผิดพลาดที่ดี โดยในช่วงแรกจะลดลงอย่างฉับพลัน ตั้งแต่รอบที่ 0–50 แล้วหลังจากนั้นจะค่อย ๆ ลดลงจนค่าความผิดพลาดนั้นมีลักษณะคงที่ γ = 0, 0.1, 0.001 และ 0.0001 ค่าความผิดพลาดที่ได้จะมีค่ามากและการลดลงของค่าความผิดพลาดไม่ดี



รูปที่ 4.6 การเปรียบเทียบค่าความผิดพลาดเนื่องจากการปรับค่ามิว (ค่าแกมมาคงที่เท่ากับ 0.01)

จากรูปที่ 4.6 ค่าความผิดพลาดที่ได้จากการปรับค่ามิว จะเห็นว่าค่าความผิดพลาดที่ γ = 0.01 และ μ = 0.01 มีการลดลงของค่าความผิดพลาดที่ดี โดยในช่วงแรกจะลดลงอย่างฉับพลันตั้งแต่รอบที่ 0–50 แล้วหลังจากนั้นจะค่อย ๆ ลดลงจนค่าความผิดพลาดนั้นมีลักษณะคงที่ ส่วนที่ค่า μ = 0, 0.1 และ 0.001 ค่าความผิดพลาดที่ได้จะมีค่ามากและการลดลงของค่าความผิดพลาดไม่ดี



รูปที่ 4.7 การเปรียบเทียบค่าความผิดพลาดทั้ง 4 วิธี

จากรูปที่ 4.7 ค่าความผิดพลาดที่ได้จากการปรับค่าของแต่ละวิธี จะเห็นว่าค่าความผิดพลาดทั้ง 3 วิธี จะมีค่าความผิดพลาดที่ดีกว่า วิธีการใช้อัตราการเรียนรู้แบบคงที่ (Non-Adaptive) เนื่องจากทั้ง 3 วิธี ค่าอัตราการเรียนรู้ถูกปรับตลอดเวลาทำให้ค่าความผิดพลาดมีค่าที่ดีกว่า ค่าพารามิเตอร์ของแต่ละวิธีได้เลือกมาจากค่าที่ดีที่สุดจากการทดลองจากกราฟที่ 4.1 – 4.6

วิธีที่ 1 ค่าแกมมาที่ดีที่สุดเท่ากับ 0.01

วิธีที่ 2 ค่าแกมมาที่ดีที่สุดเท่ากับ 0.1

วิธีที่ 3 ค่าแกมมาที่ดีที่สุดเท่ากับ 0.01 และ ค่ามิวที่ดีที่สุดเท่ากับ 0.01 และวิธีนี้ทำให้ค่าความผิดพลาดมีค่าน้อยที่สุด ซึ่งเป็นสิ่งที่ระบบต้องการ

บทที่ 5

สรุปผลและข้อเสนอแนะ

ในบทนี้จะกล่าวถึงผลของการทดลองที่เกิดขึ้นในการหาค่าความผิดพลาดด้วยวิธีการปรับค่าอัตราการเรียนรู้ที่เปลี่ยนแปลงตามเวลา โดยใช้วิธีการเรียนรู้แบบเคอร์เนล เพื่อให้สามารถจำลองฟังก์ชันที่เปลี่ยนแปลงตามเวลาได้ จึงต้องให้อัตราการเรียนรู้มีการปรับค่าตลอดเวลา การทดลองจะแสดงให้เห็นสิ่งที่จะได้ค่าความผิดพลาดที่เพิ่มขึ้นหรือลดลงอย่างไร เพื่อที่จะหาวิธีการที่สามารถลดค่าความผิดพลาดใหเหลือน้อยที่สุด

5.1 สรุปผลการทดลอง

ในโครงงานนี้ได้ศึกษาการประมาณค่าฟังก์ชันด้วยวิธีเคอร์เนลและใช้ประมาณค่าฟังก์ชันเปลี่ยนแปลงตามเวลาด้วยวิธีการปรับค่าอัตราการเรียนรู้ เพื่อหาความผิดพลาดจากข้อมูลที่ใช้ทดลองแล้วเลือกค่าความผิดพลาดที่มีค่าน้อยที่สุดมาใช้

5.1.1 การหาค่าพารามิเตอร์ที่ดีที่สุดของการประมาณค่าฟังก์ชัน

จากการทดลองในบทที่ 4 จะมีการปรับค่าพารามิเตอร์ 3 ตัว คือ ค่าอัตราการเรียนรู้ ค่าเรกกูลาไรเซชัน และค่าเบต้า เพื่อหาค่าความผิดพลาดที่ดีที่สุด ซึ่งผลการทดลองที่ได้จะเห็นว่าที่ค่าอัตราการเรียนรู้เท่ากับ 0.10 ค่าเรกกูลาไรเซชันเท่ากับ 0.001 และค่าเบต้าเท่ากับ 100 เป็นค่าที่ดีที่สุดที่ทำให้ค่าความผิดพลาดน้อยที่สุด

5.1.2 การประมาณค่าที่เปลี่ยนแปลงตามเวลาด้วยวิธีการปรับค่าอัตราการเรียนรู้

ในการที่จะปรับปรุงค่า α จะต้องใช้ค่าอัตราการเรียนรู้ซึ่งต้องเป็นค่าที่ถูกปรับตลอดเวลาเพื่อให้ทันต่อฟังก์ชันไม่รู้ค่าที่เปลี่ยนแปลงตามเวลา

จากการทดลองในบทที่ 4.2 ค่าเฉลี่ยความผิดพลาดของการประมาณค่าฟังก์ชันด้วยวิธีการปรับค่าอัตราการเรียนรู้ทั้ง 3 วิธีนี้ จะมีค่าความผิดพลาดลดลงอย่างต่อเนื่องจนค่าที่ลดลงนั้นมีลักษณะค่อนข้างคงที่และค่าความผิดพลาดที่ได้มีค่าน้อย

การประมาณการประมาณค่าฟังก์ชันด้วยวิธีการปรับค่าอัตราการเรียนรู้ทั้ง 3 วิธีนี้ จะขึ้นอยู่กับค่าแกมมา ส่วนวิธีที่ 3 จะขึ้นอยู่กับค่ามิวและค่าแกมมาซึ่งแสดงให้เห็นแล้วว่า วิธีที่ 3 เป็นวิธีที่ดีที่สุด

วิธีที่ 1 ค่าแกมมาที่ดีที่สุดเท่ากับ 0.01

วิธีที่ 2 ค่าแกมมาที่ดีที่สุดเท่ากับ 0.1

วิธีที่ 3 ค่าแกมมาที่ดีที่สุดเท่ากับ 0.01 และ ค่ามิวที่ดีที่สุดเท่ากับ 0.01 ทั้ง 3 วิธีจะมีค่าความผิดพลาดที่ได้มีค่าน้อยกว่าวิธีการใช้อัตราการเรียนรู้แบบคงที่ เพราะค่าอัตราการเรียนรู้ถูกปรับตลอดเวลา ซึ่งทำให้ค่าความผิดพลาดที่ได้มีค่าน้อยลง และนั่นเป็นสิ่งที่ระบบต้องการ

เอกสารอ้างอิง

[1] J. Platt, "**A resource - allocating network for function interpolation**," *Neural Computation,* vol. 3, pp. 213-225, 1991.

[2] N. Aronszajn, "**Theory of reproducing kernels**," *Transactions of the American Mathematical So city*, vol. 68, pp. 337-404, 1950.

[3] T.J. Dodd, "**Gradient descent approach to approximation in reproducing kernel Hilbert spaces**," Department of Automatic Control and Systems Engineering, University of Sheffield,UK, Tech. Rep. 821, 2002.

[4] สัญฉกร วุฒิสิทธิกุลกิจ และคณะ, "**MATLABการประยุกต์ใช้งานทางวิศวกรรมไฟฟ้า**", พิมพ์ครั้งที่ 3, สำนักพิมพ์แห่งจุฬาลงกรณ์มหาวิทยาลัย, 2551.

ภาคผนวก ก

แสดงข้อมูลที่ได้จากการทดลอง

	η	ρ	β	จำนวนรอบในการทดลอง		
				50	100	199
ปรับค่า η	0.1	0.01	100	0.0858	0.05	0.0288
	0.125	0.01	100	0.0775	0.0438	0.0253
	0.15	0.01	100	0.0708	0.0394	0.0234
	0.175	0.01	100	0.0654	0.0361	0.0216
	0.2	0.01	100	0.0610	0.0337	0.0206
ปรับค่า ρ	0.10	0.1	100	0.09	0.0591	0.0441
	0.10	0.2	100	0.0944	0.0659	0.0617
	0.10	0.01	100	0.0858	0.05	0.0288
	0.10	0.02	100	0.0863	0.0509	0.03
	0.10	0.001	100	0.0854	0.0492	0.0280
	0.10	0.002	100	0.0854	0.0493	0.0280
ปรับค่า β	0.10	0.001	50	0.0836	0.0508	0.0311
	0.10	0.001	100	0.0854	0.0492	0.0280
	0.10	0.001	200	0.0921	0.0537	0.0297
	0.10	0.001	300	0.0964	0.0575	0.0318
	0.10	0.001	400	0.0991	0.0603	0.0336
	0.10	0.001	500	0.1010	0.0625	0.0351

	γ	u	จำนวนรอบในการทดลอง		
			50	100	199
การปรับค่าอัตราการเรียนรู้วิธีที่ 1	0	-	0.0856	0.0498	0.0285
	0.0025	-	0.0855	0.0495	0.0283
	0.005	-	0.0853	0.0491	0.0281
	0.0075	-	0.0852	0.0488	0.0279
	0.01	-	0.085	0.0486	0.0277
	0.025	-	0.0842	0.0472	0.0271
	0.05	-	0.0830	0.0457	0.0268
	0.075	-	0.0819	0.0448	0.027
	0.1	-	0.081	0.0443	0.0274
	0.25	-	0.0771	0.0444	0.0315
	0.5	-	0.0730	0.0482	0.0397
	0.75	-	0.0701	0.0528	0.0473
	1	-	0.0677	0.0577	0.0533

	γ	u	**จำนวนรอบในการทดลอง**		
			50	**100**	**199**
การปรับค่าอัตราการเรียนรู้วิธีที่ 2	0	-	0.0852	0.0490	0.0277
	0.0025	-	0.0852	0.0490	0.0276
	0.005	-	0.0852	0.0489	0.0276
	0.0075	-	0.0852	0.0489	0.0276
	0.01	-	0.0852	0.0489	0.0276
	0.025	-	0.0851	0.0487	0.0274
	0.05	-	0.0849	0.0483	0.0272
	0.075	-	0.0848	0.0480	0.0271
	0.1	-	0.0846	0.0477	0.0269
	0.25	-	0.0838	0.0463	0.0265
	0.5	-	0.0827	0.0450	0.0271
	0.75	-	0.0818	0.0448	0.0281
	1	-	0.0812	0.0457	0.0319

	γ	u	จำนวนรอบในการทดลอง		
			50	100	199
การปรับค่าอัตราการเรียนรู้วิธีที่ 3	0.01	0.01	0.0755	0.0417	0.0236
	0.01	0.02	0.0778	0.0433	0.0245
	0.01	0.03	0.0793	0.0443	0.0250
	0.01	0.04	0.0802	0.0450	0.0254
	0.01	0.05	0.0808	0.0455	0.0257
	0.01	0.1	0.0827	0.0469	0.0265
	0.01	0.2	0.0840	0.0478	0.0270
	0.01	0.3	0.0846	0.0483	0.0272
	0.01	0.4	0.0848	0.0485	0.0273
	0.01	0.5	0.0849	0.0485	0.0274
	0.1	0.01	0.0887	0.0483	9.619
	0.1	0.02	0.0670	0.0372	0.0275
	0.1	0.03	0.0646	0.0361	0.0306
	0.1	0.04	0.0647	0.0360	12.739
	0.1	0.05	0.0655	0.0363	15.386
	0.1	0.1	0.0700	0.0383	0.659
	0.1	0.2	0.0761	0.0414	0.0260
	0.1	0.3	0.0799	0.0437	0.0272
	0.1	0.4	0.0820	0.0450	0.0275
	0.1	0.5	0.0827	0.0453	0.0275

ภาคผนวก ข

โปรแกรมที่ใช้ในการทดลอง

โปรแกรมที่ใช้ในการทดลองในบทที่ 4

1) โปรแกรมที่ใช้ในการทดลองที่ใช้อัตราการเรียนรู้แบบคงที่

```
clear all; clc
rand('seed',1032423);
randn('seed',42434123);

x_train = rand(200,1); %ข้อมูลที่ใช้สอน (x)
y_train = sinc((20*x_train-10)/pi);
y_train = y_train + 0.2*randn(size(x_train ));
x_test  = [0.02:0.02:1]'; %ข้อมูลที่ใช้ทดสอบ (x)
y_idealtest = sinc((20*x_test-10)/pi); %ข้อมูลที่ใช้ทดสอบ (y)

no_data = size(x_train,1);
%กำหนดค่าพารามิเตอร์
beta =100; %ค่าเบต้า
learnrate  =0.1; %ค่าอัตราการเรียนรู้
rho  =0.001; %ค่าเรกกูลาไรเซชั่น

for n=1;
    f=[0];%Initiative %กำหนดค่าเริ่มต้น
    e(n)=-y_train(n);
    alpha=(-learnrate)*e(n);
for i=1:50;
     xtest = x_test(i);
     yidealtest=y_idealtest(i);
     y_predict(i)=alpha*exp(-beta*(abs((x_train(n)-xtest))).^2);
     ff=[y_predict(i)]';
     error(i)=((ff-yidealtest).^2);
end
   MSE=sum(error)/length(y_idealtest);
   f_predict=alpha*exp(-beta*(abs(x_train(n)-x_train(n+1))).^2);
end

for n=2:199
```

```
    e(n)=f_predict-y_train(n);
    alpha=[alpha*(1-learnrate*rho); -learnrate*e(n)];


    for i=1:50
      xtest = x_test(i);
      yidealtest = y_idealtest(i);
      for a=1:n
        ff(a)=alpha(a)*exp(-beta*(abs(x_train(a)- x_test(i))).^2);
      end
      ff= sum(ff);%[F(i,1)]+[F(i,2)];


      error(i)=((ff-y_idealtest(i)).^2);


    end
      MSE(n)=sum(error)/length(y_idealtest);
    for a=1:n %Values provided
      ff(a)=alpha(a)*exp(-beta*(abs(x_train(n+1)-x_train(a))).^2) ;
    end
      f_predict=sum(ff);
end
 hold on
 plot(MSE) %กราฟของค่าความผิดพลาด
mean(MSE) %ค่าเฉลี่ยความผิดพลาดกำลังสอง
```

2) โปรแกรมที่ใช้ในการทดลองที่ใช้อัตราการเรียนรู้แบบ 1

```
clear all; clc
rand('seed',1032423);
randn('seed',42434123);

x_train = rand(200,1); %ข้อมูลที่ใช้สอน (x)
y_train = sinc((20*x_train-10)/pi);
y_train = y_train + 0.2*randn(size(x_train ));
x_test  = [0.02:0.02:1]'; %ข้อมูลที่ใช้ทดสอบ (x)
y_idealtest = sinc((20*x_test-10)/pi); %ข้อมูลที่ใช้ทดสอบ (y)

no_data = size(x_train,1);
%กำหนดค่าพารามิเตอร์
beta =100; %ค่าเบต้า
learnrate  =0.1; %ค่าอัตราการเรียนรู้
rho  =0.001; %ค่าเรกกูลาไรเซชั่น
gramma = 0.01; %ค่าแกมมา

for n=1;
   f=[0];%Initiative %กำหนดค่าเริ่มต้น
   e(n)=-y_train(n);
   alpha=(-learnrate(n))*e(n);
for i=1:50;
     xtest = x_test(i);
     yidealtest=y_idealtest(i);
     y_predict(i)=alpha*exp(-beta*(abs((x_train(n)-xtest))).^2);
     ff=[y_predict(i)]';
     error(i)=((ff-yidealtest).^2);
end
  MSE=sum(error)/length(y_idealtest);
  f_predict=alpha*exp(-beta*(abs(x_train(n)-x_train(n+1))).^2);
  A = (f_predict-y_train(n+1))*e(n)*exp(-beta*(abs(x_train(n)-x_train(n+1))).^2) ;
```

```
    B = rho*((e(n)*alpha*exp(-beta*(abs(x_train(n)).^2))));
    C = (f_predict-y_train(n+1))*alpha*exp(-beta*(abs(x_train(n+1))).^2);
end

for n=2:199
    e(n)=f_predict-y_train(n);
    alpha=[alpha*(1-learnrate*rho); -learnrate*e(n)];
    greg = A + B + C + ((rho^2)*alpha(n-1)*alpha(n)*(exp(-beta*(abs(x_train(n-1)-
x_train(n))).^2)));
    learnrate = learnrate + gramma*greg;

    for i=1:50
        xtest = x_test(i);
        yidealtest = y_idealtest(i);
        for a=1:n
            ff(a)=alpha(a)*exp(-beta*(abs(x_train(a)- x_test(i))).^2);
        end
        ff= sum(ff);%[F(i,1)]+[F(i,2)];

        error(i)=((ff-y_idealtest(i)).^2);

    end
        MSE(n)=sum(error)/length(y_idealtest);
    for a=1:n %Values provided
        ff(a)=alpha(a)*exp(-beta*(abs(x_train(n+1)-x_train(a))).^2) ;
    end
        f_predict=sum(ff);
        A = (f_predict-y_train(n))*e(n)*exp(-beta*(abs(x_train(n+1)-x_train(a))).^2);
        B = rho*((e(n)*alpha(n)*exp(-beta*(abs(x_train(a))).^2)));
        C = (f_predict-y_train(n+1))*exp(-beta*(abs(x_train(a+1))).^2);
        greg = A + B + C + ((rho^2)*alpha(n-1,1)*alpha(n,1)*(exp(-beta*(abs(x_train(n-1)-
x_train(n))).^2)));
```

```
end
 hold on
 plot(MSE) %กราฟของค่าความผิดพลาด
 mean (MSE) %ค่าเฉลี่ยความผิดพลาดกำลังสอง
```

3) โปรแกรมที่ใช้ในการทดลองที่ใช้อัตราการเรียนรู้แบบ 2

```
clear all; clc
rand('seed',1032423);
randn('seed',42434123);

x_train = rand(200,1); %ข้อมูลที่ใช้สอน (x)
y_train = sinc((20*x_train-10)/pi);
y_train = y_train + 0.2*randn(size(x_train ));
x_test  = [0.02:0.02:1]'; %ข้อมูลที่ใช้ทดสอบ (x)
y_idealtest = sinc((20*x_test-10)/pi); %ข้อมูลที่ใช้ทดสอบ (y)

no_data = size(x_train,1);
%กำหนดค่าพารามิเตอร์
beta =100;%ค่าเบต้า
learnrate  =0.1; %ค่าอัตราการเรียนรู้
rho  =0.001; %ค่าเรกกูลาไรเซชั่น
gramma = 0.1; %ค่าแกมมา

for n=1;
    f=[0];%Initiative %กำหนดค่าเริ่มต้น
    e(n)=-y_train(n);
    alpha=(-learnrate(n))*e(n);
for i=1:50;
     xtest = x_test(i);
     yidealtest=y_idealtest(i);
     y_predict(i)=alpha*exp(-beta*(abs((x_train(n)-xtest))).^2);
     ff=[y_predict(i)]';
     error(i)=((ff-yidealtest).^2);
end
  MSE=sum(error)/length(y_idealtest);
  f_predict=alpha*exp(-beta*(abs(x_train(n)-x_train(n+1))).^2);
  A = (f_predict-y_train(n+1))*e(n)*exp(-beta*(abs(x_train(n)-x_train(n+1))).^2) ;
```

```
  B = rho*((e(n)*alpha*exp(-beta*(abs(x_train(n)).^2))));
  C = (f_predict-y_train(n+1))*alpha*exp(-beta*(abs(x_train(n+1))).^2);
end

for n=2:199
    e(n)=f_predict-y_train(n);
    alpha=[alpha*(1-learnrate*rho); -learnrate*e(n)];
    greg = A + B + C + ((rho^2)*alpha(n-1)*alpha(n)*(exp(-beta*(abs(x_train(n-1)-
x_train(n))).^2)));
    learnrate = learnrate*(1 + (gramma*greg));

    for i=1:50
      xtest = x_test(i);
      yidealtest = y_idealtest(i);
      for a=1:n
        ff(a)=alpha(a)*exp(-beta*(abs(x_train(a)- x_test(i))).^2);
      end
      ff= sum(ff);%[F(i,1)]+[F(i,2)];

      error(i)=((ff-y_idealtest(i)).^2);

    end
      MSE(n)=sum(error)/length(y_idealtest);
    for a=1:n %Values provided
      ff(a)=alpha(a)*exp(-beta*(abs(x_train(n+1)-x_train(a))).^2) ;
    end
      f_predict=sum(ff);
      A = (f_predict-y_train(n))*e(n)*exp(-beta*(abs(x_train(n+1)-x_train(a))).^2);
      B = rho*((e(n)*alpha(n)*exp(-beta*(abs(x_train(a))).^2)));
      C = (f_predict-y_train(n+1))*exp(-beta*(abs(x_train(a+1))).^2);
      greg = A + B + C + ((rho^2)*alpha(n-1,1)*alpha(n,1)*(exp(-beta*(abs(x_train(n-1)-
x_train(n))).^2)));
```

```
end
 hold on
 plot(MSE) %กราฟของค่าความผิดพลาด
 mean(MSE) %ค่าเฉลี่ยความผิดพลาดกำลังสอง
```

4) โปรแกรมที่ใช้ในการทดลองที่ใช้อัตราการเรียนรู้แบบ 3

```
clear all; clc
rand('seed',1032423);
randn('seed',42434123);

x_train = rand(200,1); %ข้อมูลที่ใช้สอน (x)
y_train = sinc((20*x_train-10)/pi);
y_train = y_train + 0.2*randn(size(x_train ));
x_test  = [0.02:0.02:1]'; %ข้อมูลที่ใช้ทดสอบ (x)
y_idealtest = sinc((20*x_test-10)/pi); %ข้อมูลที่ใช้ทดสอบ (y)

no_data = size(x_train,1);
%กำหนดค่าพารามิเตอร์
beta =100; %ค่าเบต้า
learnrate  =0.1; %ค่าอัตราการเรียนรู้
rho  =0.01; %ค่าเรกกูลาไรเซชั่น
gramma = 0.01; %ค่าแกมมา
mu = 0.01;  %ค่ามิว


for n=1;
  u=[0];%Initiative %กำหนดค่าเริ่มต้น
  f=[0];%Initiative %กำหนดค่าเริ่มต้น
  e(n)=-y_train(n);
  Un(n) = 1;
  alpha=(-learnrate(n))*e(n);
  for i=1:50;
  xtest = x_test(i);
  yidealtest=y_idealtest(i);
  y_predict(i)=alpha*exp(-beta*(abs((x_train(n)-xtest))).^2);
  ff=[y_predict(i)]';
  error(i)=((ff-yidealtest).^2);
```

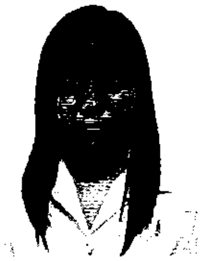
```
    end
    MSE=sum(error)/length(y_idealtest);
    f_predict=alpha*exp(-beta*(abs(x_train(n)-x_train(n+1))).^2);
    A = (f_predict-y_train(n+1))*e(n)*exp(-beta*(abs(x_train(n)-x_train(n+1))).^2) ;
    B = rho*((e(n)*alpha*exp(-beta*(abs(x_train(n)).^2))));
    C = (f_predict-y_train(n+1))*alpha*exp(-beta*(abs(x_train(n+1))).^2);
    AA = (exp(-beta*(abs(x_train(n+1)-x_train(n+1))).^2))*((f_predict-y_train(n+1))^2);
    BB = 2*(f_predict-y_train(n+1))*rho*(alpha*exp(-beta*(abs((x_train(n+1)))).^2));
end

for n=2:199
    e(n)=f_predict-y_train(n);
    alpha=[alpha*(1-learnrate*rho); -learnrate*e(n)];
    greg_2 = AA + BB + ((rho^2)*alpha(n)*alpha(n)*(exp(-beta*(abs(x_train(n)-x_train(n))).^2)));
    Un = (mu*Un)+((1-mu)*greg_2);
    greg = A + B + C + ((rho^2)*alpha(n-1)*alpha(n)*(exp(-beta*(abs(x_train(n-1)-
x_train(n))).^2)));
    learnrate = learnrate*(1 + (gramma/Un)*greg)  ;
    for i=1:50
        xtest = x_test(i);
        yidealtest = y_idealtest(i);
        for a=1:n
            ff(a)=alpha(a)*exp(-beta*(abs(x_train(a)- x_test(i))).^2);
        end
        ff= sum(ff);%[F(i,1)]+[F(i,2)];
        error(i)=((ff-y_idealtest(i)).^2);
    end
    MSE(n)=sum(error)/length(y_idealtest);
    for a=1:n %Values provided
        ff(a)=alpha(a)*exp(-beta*(abs(x_train(n+1)-x_train(a))).^2) ;
    end
    f_predict=sum(ff);
```

```
    A = (f_predict-y_train(n))*e(n)*exp(-beta*(abs(x_train(n+1)-x_train(a))).^2);
    B = rho*((e(n)*alpha(n)*exp(-beta*(abs(x_train(a))).^2)));
    C = (f_predict-y_train(n+1))*exp(-beta*(abs(x_train(a+1))).^2);
    AA = exp(-beta*(abs(x_train(n+1)-x_train(a+1))).^2)*((f_predict-y_train(n))^2);
    BB = 2*(f_predict-y_train(n))*rho*exp(-beta*(abs(x_train(a+1))).^2);
    greg = A + B + C + ((rho^2)*alpha(n-1,1)*alpha(n,1)*(exp(-beta*(abs(x_train(n-1)-
x_train(n))).^2)));

end
 hold on
 plot(MSE) %กราฟของค่าความผิดพลาด
 mean (MSE) %ค่าเฉลี่ยความผิดพลาดกำลังสอง
```

ประวัติผู้ดำเนินโครงงาน



ชื่อ นางสาววารี คำแสน

ภูมิลำเนา 44 หมู่ 4 ต. ศาลา อ. เกาะคา จ. ลำปาง 52130

ประวัติการศึกษา

– จบระดับมัธยมศึกษาจากโรงเรียนมัธยมวิทยา

– ปัจจุบันกำลังศึกษาระดับปริญญาตรีชั้นปีที่ 5 สาขาวิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์ ม.นเรศวร

Email: pooy_princess@hotmail.com

ชื่อ นายสุนทร ศิลปวงศ์

ภูมิลำเนา 471 หมู่ 13 ต.ต้นธงชัย อ. เมือง จ. ลำปาง 52000

ประวัติการศึกษา

– จบระดับมัธยมศึกษาจากโรงเรียนมัธยมวิทยา

– ปัจจุบันกำลังศึกษาระดับปริญญาตรีชั้นปีที่ 5 สาขาวิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์ ม.นเรศวร

Email: S.Sinlapawong@gmail.com

ชื่อ นายเทวา ตาเปี้ยสืบ

ภูมิลำเนา 8/1 หมู่ 7 ต. นาสัก อ. แม่เมาะ จ. ลำปาง 52220

ประวัติการศึกษา

– จบระดับมัธยมศึกษาจากโรงเรียนแม่เมาะวิทยา

– ปัจจุบันกำลังศึกษาระดับปริญญาตรีชั้นปีที่ 5 สาขาวิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์ ม.นเรศวร

Email: T.Tatiaseub@yahoo.com