ห้องสมุดคณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

การเรียนรู้ชนิดหลายแบบจำลองด้วยไฮเปอร์พารามิเตอร์

A MULTIPLE MODEL APPROACH WITH HYPERPARAMETERS

นายวีระพงษ์ ศรีโชติ รหัส 48380158

นายรัฐพล สุคนธวิโรจน์ รหัส 48380360

งสมุดคณะวิศวกรรมศาสตร์
วันที่รับ 11 มี.ค. 2555
เลขทะเบียน 1573339
เลขเรียกหนังสือ ปร.
มหาวิทยาลัยนเรศวร 28461
2552

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมไฟฟ้า ภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

ปีการศึกษา 2552

ใบรับรองปริญญานิพนธ์

ชื่อหัวข้อโครงงาน การเรียนรู้ชนิดหลายแบบจำลองด้วยไฮเปอร์พารามิเตอร์

ผู้ดำเนินโครงงาน นายวีระพงษ์ ศรีโชติ รหัส 48380158

นายรัฐพล สุคนธวิโรจน์ รหัส 48380360

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	การเรียนรู้ชนิดหลายแบบจำลองด้วยไฮเปอร์พารามิเตอร์
ผู้ดำเนินโครงงาน	นายวีระพงษ์ ศรีโชติ รหัส 48380158
	นายรัฐพล สุคนธวิโรจน์ รหัส 48380360
ที่ปรึกษาโครงงาน	ดร. ศุภวรรณ พลพิทักษ์ชัย
สาขาวิชา	วิศวกรรมไฟฟ้า
ภาควิชา	วิศวกรรมไฟฟ้าและคอมพิวเตอร์
ปีการศึกษา	2552

....

บทคัดย่อ

โครงงานนี้ได้ศึกษาการเรียนรู้ชนิดหลายแบบจำลองด้วยไฮเปอร์พารามิเตอร์ ซึ่งนำมาประยุกต์ใช้สำหรับหาความสัมพันธ์ในรูปแบบฟังก์ชันระหว่างข้อมูลขาเข้าและขาออก สิ่งที่ได้จะเรียกว่าฟังก์ชันการประมาณค่าที่ประกอบไปด้วยผลบวกของฟังก์ชันเคอร์เนลที่แต่ละเทอมถูกให้ค่าน้ำหนักด้วยพารามิเตอร์ในรูปเมตริกซ์แอลฟา การคำนวณโดยวิธีออนไลน์จะแก้ข้อเสียของวิธีทั้งหมด เนื่องจากใช้ข้อมูลแค่ทีละตัวในการปรับค่าแอลฟาที่กล่าวมาแล้ว ดังนั้นจึงไม่มีปัญหาในการใช้งานกับข้อมูลจำนวนมากๆ การเรียนรู้ชนิดหลายแบบจำลองเป็นการใช้แบบจำลองมากกว่าหนึ่งในการหาฟังก์ชันการประมาณค่า โดยที่แต่ละแบบจำลองจะใช้ค่าของเรกกูลาไรเซชันพารามิเตอร์และความกว้างเคอร์เนลที่ต่างกัน ค่าเอาท์พุตและความผิดพลาดกำลังสองเฉลี่ยที่แบบจำลองชนิดหลายแบบจำลองทำนายได้จะเป็นการนำเอาค่าทั้งสองของแต่ละแบบจำลองมารวมกันแบบคิดค่าน้ำหนัก ซึ่งค่าน้ำหนักจะมีความสัมพันธ์กับความผิดพลาดของแต่ละแบบจำลอง

Project title	A Multiple Model Approach with Hyperparameters
Name	Mr.Weerapong Srichot ID. 48380158
	Mr.Rattapol Sukontaviloj ID. 48380360
Project advisor	Supawan Phonphitakchai, Ph.D.
Major	Electrical Engineering
Department	Electrical and Computer Engineering
Academic year	2009

Abstract

This project studies a multiple model approach with hyperparameters with kernel method which is applied to investigate a relation between input and output data. The relation is presented in the form of approximation function as a summation of kernel terms corresponding to the weight parameters, called alpha matrix.

This online learning method is introduced to overcome a disadvantage of batch learning that it uses only one data pair to be processed by updating the parameters and then discarded. For this reason, the online method is able to apply with a huge data set without a computation problem. Learning with hyperparameters uses more than one model to estimate the approximation function. Each model is assigned with different values of regularization parameter and kernel width whereas the learning rate is kept at a constant. Output and mean square error of multiple models are the summation values of each model associated with their weights. Amount of weight of each model is related to the mean square error of the corresponding models.

กิตติกรรมประกาศ

ทางคณะผู้จัดทำโครงงาน "การเรียนรู้ชนิดหลายแบบจำลองด้วยไฮเปอร์พารามิเตอร์" ขอขอบคุณ ดร. ศุภวรรณ พลพิทักษ์ชัย ที่ให้ความช่วยเหลือในโครงงานนี้ให้สามารถดำเนินการไปได้ด้วยดี โดยช่วยให้คำแนะนำปรึกษาเกี่ยวกับโครงงานตลอดทั้งให้ความเอื้อเฟื้อสถานที่ในการทำงานและอุปกรณ์เครื่องมือต่างๆ และคณะกรรมการอีก 2 ท่าน คือ ดร.มุฑิตา สงฆ์จันทร์ และ ดร. นิพัทธ์ จันทรมินทร์ ที่ได้ให้คำแนะนำและยังคอยช่วยแก้ไขปัญหาในการปฏิบัติงาน นอกจากนี้ขอขอบคุณภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์ที่ได้ให้เงินสนับสนุนการทำโครงงานนี้

ในโอกาสนี้ทางคณะผู้จัดทำโครงงานจึงขอขอบคุณทุกท่านที่มีส่วนร่วมในการทำโครงงานนี้ ที่ได้ให้คำเสนอแนะ จึงทำให้โครงงานนี้สำเร็จไปด้วยดี

นายวีรพงษ์ ศรีโชติ
นายรัฐพล สุคนธวิโรจน์

สารบัญ

สารบัญ (ต่อ)

สารบัญรูป

บทที่ 1

บทนำ

1.1 ที่มาและความสำคัญของโครงงาน

เครื่องการเรียนรู้ (Machine learning) เป็นกลุ่มงานวิจัยหลายแขนงคือรวมเอาหัวข้อหลายอย่างไว้จึงค่อนข้างยากและยังมีเป้าหมายชัดเจน อาจจะบอกได้ว่าเป็นกลุ่มของเทคนิคในการทำให้คอมพิวเตอร์ทำงานบางอย่างได้ดีขึ้นโดยการใช้ข้อมูลพื้นฐานข้อมูลฝึกสอนหรือกลุ่มของเทคนิคในการทำให้คอมพิวเตอร์ทำงานได้อย่างอัตโนมัติ งานบางอย่างที่กล่าวถึงนี้ก็มี การจำแนกประเภท (Classification) การเลือกค่าแบบจำลอง (Model selection) การคำนวณการกระจายของข้อมูล (Density estimation) และ การทำนายค่า (Prediction หรือ Regression) ทฤษฎีด้านสถิติมักจะตั้งสมมติฐานว่ามีข้อมูลจำนวนมากไม่จำกัดขณะที่งานด้านทฤษฎีส่วนมากในเครื่องการเรียนรู้ จะสนใจการวิเคราะห์ข้อมูลที่มีจำกัด

การเรียนรู้ความสัมพันธ์ระหว่างข้อมูลขาเข้าและขาออกหมายถึงการหาข้อมูลออกมาในรูปแบบของฟังก์ชันของระบบที่เราไม่รู้จัก เมื่อเราทราบถึงฟังก์ชันดังกล่าวจะสามารถปรับปรุงและทำนายการทำงานของระบบในอนาคตได้

ส่วนใหญ่แล้วการเรียนรู้จากข้อมูลจะใช้แบบข้อมูลทั้งหมด (Batch learning) ในการสอน (train) เพียงครั้งเดียว อย่างไรก็ตามวิธีนี้จะมีข้อเสียในกรณีที่ข้อมูลมีจำนวนมาก การทำแบบนี้จะทำให้คอมพิวเตอร์มีปัญหาในการประมวลผลจากข้อมูลที่เป็นกลุ่มใหญ่ และอาจจะทำให้ระบบหยุดการทำงานได้เลย

ดังนั้น เราจึงใช้การเรียนรู้แบบออนไลน์ (Online learning) ซึ่งวิธีนี้จะใช้การป้อนข้อมูลเข้าไปทีละตัวเพื่อใช้ในการสร้าง โดยจะมีแบบจำลอง (Model) และจะทำให้คอมพิวเตอร์ของเราไม่เกิดปัญหาเนื่องจากการประมวลผลข้อมูลชุดใหญ่และทำงานได้อย่างมีประสิทธิภาพ เนื่องจากจะป้อนข้อมูลไปทีละตัว นอกจากนั้นยังมีการเรียนรู้ชนิด Supervised learning และ Unsupervised learning

Supervised learning คือ ระบบการควบคุมการเรียนรู้เป็นเครื่องมือการเรียนรู้เทคนิคการใช้สำหรับการควบคุมฟังก์ชันข้อมูล การควบคุมฟังก์ชันข้อมูลประกอบด้วยขาอินพุตและขาเอาต์พุต ที่ข้อมูลเอาต์พุตของฟังก์ชันอาจเป็นข้อมูลที่มีค่าต่อเนื่อง หรือเป็นข้อมูลที่สุ่มค่าจากกลุ่มของข้อมูลที่ป้อนเข้ามา งานของระบบการควบคุมการเรียนรู้ คือ การสุ่มค่าที่ถูกต้องที่สุดจากข้อมูลที่ป้อนเข้ามา

Unsupervised learning คือ เครื่องมือการเรียนรู้ของการเรียนรู้แบบ Unsupervised เป็นปัญหาหนึ่ง เพื่อให้ระบบตัดสินใจว่าข้อมูลที่จัดเก็บเป็นข้อมูลที่ถูกแยกมาจากระบบการควบคุมการเรียนรู้ (Supervised learning) และการเรียนรู้แบบ Unsupervised ยังเป็นระบบการเรียนรู้ที่จะคอยรวบรวมข้อมูลที่อยู่อย่างอิสระให้เข้ามารวมอยู่ด้วยกัน

ในปี 1980 ได้มีผู้เสนอการเรียนรู้วิธีเคอร์เนล (Kernel method) ซึ่งมีข้อแตกต่างกับวิธีการเรียนรู้แบบเดิมเช่น วิธีระบบโครงข่ายประสาท (Neural Network) หรือ "โครงข่ายใยประสาทเสมือน (Artificial Neural Network: ANN) หมายถึง คอมพิวเตอร์ที่สามารถเลียนแบบการทำงานของสมองมนุษย์ได้ ด้วยการประมวลผลข้อมูลสารสนเทศ และองค์ความรู้ได้ในคราวละมากๆ อย่างไรก็ตามวิธีนี้จะมีข้อเสียในแง่ของการหาค่าที่เหมาะสมที่สุด (Optimization) พูดถึงค่าที่ต่ำสุดเฉพาะที่(Localminima)

ในปัจุบันนี้การเรียนรู้วิธีเคอร์เนล ถูกนำไปใช้ในงานต่างๆ อาทิเช่น รูปแบบการวิเคราะห์ซึ่งรู้จักกันดีที่สุดคือ เอสวีเอ็ม (Supper vector machine หรือ SVM) การวิเคราะห์รูปแบบของงานคือการศึกษาประเภทของความสัมพันธ์ เช่น การจัดกลุ่ม การจัดอันดับ และ การจำแนกประเภท

โครงงานนี้เป็นการศึกษาทฤษฏีของ การเรียนรู้แบบออนไลน์ด้วยวิธีเคอร์เนล เบื้องต้นโดยจะมุ่งเน้นในการด้านการใช้หลายแบบจำลองสำหรับไฮเปอร์พารามิเตอร์ (Hyper parameter) เพื่อหาค่าความสำคัญของแต่ละแบบจำลอง โดยที่แต่ละแบบจำลองมีค่าไฮเปอร์พารามิเตอร์ที่ต่างกัน การสร้างการเรียนรู้วิธีเคอร์เนลจะใช้โปรแกรมแมทแลบ (MatLab)

1.2 วัตถุประสงค์ของโครงงาน

1. เพื่อศึกษาทฤษฏีการเรียนรู้แบบออนไลน์วิธีเคอร์เนล
2. เพื่อศึกษาเกี่ยวกับไฮเปอร์พารามิเตอร์ของการเรียนรู้
3. เพื่อศึกษาการใช้วิธีหลายแบบจำลองสำหรับไฮเปอร์พารามิเตอร์ที่แตกต่างกัน

1.3 ขอบเขตของโครงงาน

1. ศึกษาเรื่องทฤษฏีของ การเรียนรู้แบบออนไลน์ด้วยวิธีเคอร์เนล
2. กำหนดไฮเปอร์พารามิเตอร์ของการเรียนรู้
3. สร้างแบบจำลองชนิดหลายแบบจำลองสำหรับค่าไฮเปอร์พารามิเตอร์ที่แตกต่างกัน โดยหาค่าความสำคัญของแต่ละแบบจำลอง
4. เปรียบเทียบผลที่ได้และวิเคราะห์ผลที่ได้จากการเรียนรู้ชนิดหลายแบบจำลอง

1.4 ขั้นตอนการดำเนินงานและแผนการดำเนินงานตลอดโครงงานวิจัย

รายละเอียด	ปี2552						ปี2553		
	ก.ค.	ส.ค.	ก.ย.	ต.ค.	พ.ย.	ธ.ค.	ม.ค.	ก.พ.	มี.ค.
1. ศึกษาทฤษฎีการเรียนรู้ด้วยวิธีเคอร์เนล									
2. กำหนดไฮเปอร์พารามิเตอร์ของการเรียนรู้									
3. สร้างแบบจำลองชนิดหลายแบบจำลองสำหรับค่าไฮเปอร์พารามิเตอร์ที่แตกต่างกัน									
4.เปรียบเทียบผลที่ได้และวิเคราะห์ผลที่ได้จากการเรียนรู้ชนิดหลายแบบจำลอง									
5.จัดทำรายงานและสรุปผล									

1.5 ประโยชน์ที่คาดว่าจะได้รับจากโครงงาน

1. เข้าใจถึงวิธีและทฤษฎีของ การเรียนรู้แบบออนไลน์ด้วยวิธีเคอร์เนล
2. เข้าใจถึงการกำหนดไฮเปอร์พารามิเตอร์ของการเรียนรู้
3. สร้างแบบจำลองชนิดหลายแบบจำลองสำหรับค่าไฮเปอร์พารามิเตอร์แตกต่างกันได้
4. นำความรู้ที่ได้ไปประยุกต์ใช้งานตามที่ได้ศึกษามา

1.6 งบประมาณ

1. ค่าถ่ายเอกสารและค่าเข้าเล่มปริญญานิพนธ์	1,000	บาท
2. ค่าหนังสือMatlab	500	บาท
3. ค่าพิมพ์เอกสาร	500	บาท
รวมเป็นเงินทั้งสิ้น(สองพันบาทถ้วน)	**2,000**	**บาท**

หมายเหตุ ถัวเฉลี่ยทุกรายการ

บทที่ 2
การประมาณค่าในฟังก์ชันอาร์เคเอชเอส

ในการศึกษาเรื่องการหาแบบจำลองบนอาร์เคเอชเอส (RKHS - Resproducing kernel Hilbert spaces) ได้ถูกนำมาประยุกต์ใช้เพื่อแก้โจทย์ปัญหาที่เป็นที่รู้จักกันเป็นอย่างดี อาทิ เช่น การประมวลผลสัญญาณ การควบคุมเครื่องกล การเรียนรู้ และการประมาณค่าฟังก์ชัน วิธี(RKHS - Resproducing kernel Hilbert spaces) ถูกใช้เป็นแนวทางในการประมาณค่าฟังก์ชันที่ไม่รู้ค่าโดยการใช้ตัวอย่างข้อมูลที่มีจำนวนจำกัด ในปัจจุบันวิธีการหาแบบจำลองบนอาร์เคเอชเอส ได้ถูกประยุกต์ใช้เป็นอย่างมากในงานด้านเครื่องกลการเรียนรู้ เช่น เอสวีเอ็ม (SVM -Support vector machine), ระบบเกาเสียน (Gaussian processes) และเครือข่ายเรกกูลาไรเซชัน (Regularization network)

ในบทนี้จะแสดงทฤษฎีที่เกี่ยวกับสิ่งที่เรานิยามว่าเป็นการเรียนรู้แบบออนไลน์โดยการที่ดำเนินการกับข้อมูลเดียวที่แต่ละเวลานั้นและละทิ้งข้อมูลที่ผ่านมาทั้งหมด แต่ข้อมูลเกี่ยวกับข้อมูลที่ผ่านมาทั้งหมดจะถูกเก็บไว้ในโมเดล ดังนั้นการเรียนรู้แบบออนไลน์จึงเป็นทั้งการเรียนรู้ที่ค่อยๆเพิ่มขึ้นและเป็นการปฏิบัติการแบบซ้ำๆ

จุดประสงค์ของการเรียนรู้ด้วยอาร์เคเอชเอส คือการนำไปประมาณค่าฟังก์ชัน จากฟังก์ชันไม่รู้ค่าหรือระบบที่เราไม่รู้จักจากข้อมูลที่เราสามารถสังเกตได้จากระบบนั้นๆ ซึ่งจะแสดงให้เห็นว่าการประมาณค่าฟังก์ชันดังกล่าวมีความเป็นไปได้ถ้าหากเราสันนิษฐานว่าฟังก์ชันเป็นของการหาแบบจำลองบนอาร์เคเอชเอส

2.1 การประมาณค่าฟังก์ชันใน RKHS

สมมติว่าฟังก์ชันที่ไม่รู้จัก ซึ่งก็คือ f สามารถที่จะสังเกตค่าที่ได้เป็นจำนวนจำกัด f เป็นส่วนหนึ่งของการหาแบบจำลองบนอาร์เคเอชเอส ส่วน $\mathcal{F}$ และ f นิยามบนเซต x ซึ่งสามารถพิจารณาว่าเป็นชุดข้อมูลขาเข้าโดยที่ $x \in \mathcal{X}$ และ $f(x)$ หมายถึงการประเมินค่าของ f ที่ x ชุดเซตของข้อมูลขาเข้า x จะถูกพิจารณาว่าเป็นซับเซ็ท ของปริภูมิยูคลิเดียน (Euclindian space) แสดงโดย $x \subseteq \mathbb{R}^n$ และดังนั้นแต่ละ x จึงเป็นเวกเตอร์ n มิติ

เซตจำกัดของการสังเกตหรือข้อมูลขาออกของฟังก์ชัน $\{z_i\}_{i=1}^{N}$ ถูกสร้างขึ้นให้สอดคล้องกับ ข้อมูลขาเข้า $\{x_i\}_{i=1}^{N}$ เราสันนิษฐานว่า ปริภูมิของการสังเกตที่เป็นไปได้ทั้งหมดคือ เมทริกซ์สเปซ Z โดยการไม่คิดค่าความผิดพลาดต่างๆ การสังเกตเกิดขึ้นได้ดังนี้

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

ซึ่ง $\{L_i\}_{i=1}^{N}$ เป็นเซตของการประเมินผลฟังก์ชันนัลแบบเชิงเส้น (Linear evaluation functional) ที่ถูกกำหนดบน $\mathcal{F}$ ซึ่งสอดคล้องกับฟังก์ชัน f และเราสามารถนำเสนอชุดการสังเกตที่สมบูรณ์ $\{z_i\}_{i=1}^{N}$ นี้

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

เมื่อ $S_i \in \mathbb{R}^n$ คือ เวคเตอร์มาตรฐาน (Standard basis vector) ที่ i ในทางปฏิบัติเราสนใจในกรณีดังต่อไปนี้

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

ซึ่งนำไปสู่ปัญหาการคำนวณค่าฟังก์ชัน

ปัญหาการประมาณค่าฟังก์ชันซึ่งเป็นตัวแทนของฟังก์ชันไม่รู้ค่าสามารถกระทำได้ โดยการกำหนดคลาส $\mathcal{F}$ ของฟังก์ชันและชุดข้อมูลค่าสังเกต $\{z_i\}_{i=1}^{N}$ ของฟังก์ชันนัล L_i ที่ถูกกำหนดไว้บน $\mathcal{F}$ ในฟังก์ชัน f ที่สอดคล้องกับสมการที่ (2.1) และสมการที่ (2.3) ได้

ตามหลักแล้วเราสามารถกำหนดให้การหาแบบจำลองบนอาร์เคเอชเอส เป็นปริภูมิของฮิลเบิร์ต (Hilbert space) ของฟังก์ชันต่างๆ บน X ได้ โดยการมีคุณสมบัติที่ว่า โดยที่แต่ละ $x \in X$ การประเมินค่าฟังก์ชัน L_i ซึ่งสัมพันธ์กับ f โดยเขียนได้เป็น $L_i \rightarrow f(X_i)$ คือฟังก์ชันนัลแบบเส้นตรงที่ถูกกำหนดขอบเขต (Bounded linear functional) ซึ่งการกำหนดขอบเขตหมายความว่ามีค่าคงที่ M ที่สอดคล้องกับสมการข้างล่างดังนี้

$|L_i f| = |f(x_i)| \le M\|f\|$ สำหรับทุกๆ f ในการหาแบบจำลองบนอาร์เคเอชเอส ซึ่ง $\|.\|$ คือ ค่าปกติในการหาแบบจำลองบนอาร์เคเอชเอส

เนื่องจาก L_i ถูกจำกัด ขอบเขต ดังนั้นจึงสามารถใช้ทฤษฎีบทการนำเสนอของของรีซ (Riesz representation theorem) แสดงการสังเกตได้เป็น

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

ซึ่ง $\langle .,. \rangle$ หมายถึงผลคูณภายใน (Inner product) ใน $\mathcal{F}$ และ $\{k_i\}_{i=1}^{N}$ คือเซ็ทของฟังก์ชัน ที่ถูกเรียกว่ารีโพรดิวซิ่งเคอร์เนล (Reproducing kernels) ซึ่งแต่ละเคอร์เนลเป็นของ $\mathcal{F}$ และถูกกำหนดให้เป็นฟังก์ชัน L_i

ปัญหาการประมาณค่าขณะนี้สามารถกล่าวได้ใหม่ว่ากำหนดปริภูมิฮิลเบิร์ต เซตของฟังก์ชัน $\{k_i\}_{i=1}^{N}$ และการสังเกต $\{z_i\}_{i=1}^{N}$ จะสามารถหาฟังก์ชันการประมาณค่า $f \in \mathcal{F}$ ที่สอดคล้องกับสมการที่ (2.4) ได้

สำหรับทุกๆการหาแบบจำลองบนอาร์เคเอชเอส จะมีฟังก์ชันบวกแน่นอน (Positine-definitefunction) ที่เรียกว่ารีโพรดิวซิ่งเคอร์เนล (k)

ในการหาแบบจำลองบนอาร์เคเอชเอส ในรูปของ ฟังก์ชันนัลเส้นตรงที่จำกัดนั้นเรามีคุณสมบัติต่างๆ ดังต่อไปนี้ซึ่งนิยามบนการหาแบบจำลองบนอาร์เคเอชเอส นั่นคือสำหรับทุกๆ $x, x' \in X$ จะได้ $k(.,x')$ คือฟังก์ชันที่ถูกนิยามบน X ซึ่งมีค่า x ใน X เท่ากับ $k(x,x')$ และมีคุณสมบัติต่อไปนี้

(i) $k(.,x') \in \mathcal{F}$; และ

(ii) $(f, k(.,x'))_{\mathcal{F}} = f(x')$

สำหรับทุกๆ f ใน F

จากคุณสมบัติข้างต้นนี้เราจะเห็นได้ว่าฟังก์ชั่น รีโพรดิวซิ่งเคอร์เนล (Reproducing kernels), K_1^0 คือฟังก์ชันโดยทั่วไปบน $X \times X$ แต่อย่างไรก็ตามเราสามารถแปลงฟังก์ชันเคอร์เนล ให้เป็นฟังก์ชันบน X เท่านั้นก็ได้ โดยการพิจารณาฟังก์ชันเคอร์เนล $k(x,xi)$ ให้ x_i คือจุดที่ไม่เปลี่ยนแปลง $x_i \in X$ ดังนั้นการเขียน $k(x,xi) = k_i(x)$ (หรือ k_i ซึ่งง่ายมากกว่า) สามารถสรุปได้ว่า $k_i \in \mathcal{F}$ บน X ถูกทำให้มีศูนย์กลางบน x_i ซึ่งเป็นจุดที่ไม่เปลี่ยนแปลงโดยทั่วไป x_i

ควรที่จะถูกกำหนดให้เป็นศูนย์กลางของฟังก์ชัน เคอร์เนล และเป็น ไฮเปอร์พารามิเตอร์ (Hyperparameter) ของฟังก์ชันเคอร์เนล ศูนย์กลางเหล่านี้จะสัมพันธ์กับค่าข้อมูลขาเข้า

ดังนั้นเราจึงสามารถนำเสนอฟังก์ชันทั่วไปใดๆ ก็ได้ใน (RKHS-Resproducing kernel Hilbert spaces) ด้วยรีโพรดิวซิ่งเคอร์เนล k ดังนี้

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

ซึ่ง $\propto_i \in \mathbb{R}, i \in \mathbb{N}$ (ในทางปฏิบัติค่าผลบวกมักจะเป็นค่าจำกัดเสมอ) จากนั้นปัญหาการประมาณค่าฟังก์ชันก็จะเปลี่ยนไปเป็นหาค่าของการประมาณค่าที่เหมาะสมสำหรับพารามิเตอร์ α_i ในสมการที่ (2.5)

ฟังก์ชันที่ถือว่าเป็นฟังก์ชั่นเคอร์เนล k มีหลายชนิดที่นิยมใช้ ยกตัวอย่าง เช่น ฟังก์ชันเกาเสียนเรเดียลเบสิสที่ใช้ในระบบข่ายประสาท

$$k(x,x_i)=\exp(-\beta\|x-x'\|^2) \tag{2.6}$$

โดยที่ $\beta > 0$

ฟังก์ชันโพลิโนเมียลของค่าชุดของค่า x กำลัง d แสดงได้โดย

$$k(x,x')=\sum_{k=0}^{d}\frac{1}{k!\rho_k}(xx')^k \tag{2.7}$$

เมื่อ ρ_k คือเซตของค่าคงที่ที่เรียกว่าค่าน้ำหนักและมีค่าเท่ากับ

$$k(x,x')=(1+xx')^d \tag{2.8}$$

ฟังก์ชั่นเคอร์เนลสามารถแสดงได้ด้วยปริภูมิพาเลย์ – เวียเนอร์ของฟังก์ชันจำกัดแถบ (Bandlimited function)

$$k(x,x') = \frac{\sin \pi(x-x')}{\pi(x-x')} \tag{2.9}$$

การประมานค่าฟังก์ชันในอาร์เคเอชเอสแบบออนไลน์

ในกรณีนี้ให้ถือว่าที่แต่ละการทำซ้ำเราสังเกตเพียงส่วนหนึ่งของ z นั่นคือ z_n (การสังเกตที่ n) และ

$$L_n f = z_n \tag{2.10}$$

นิยามฟังก์ชั่น $\hat{g}_{reg}$ ที่ทีค่าไม่เป็นลบ ณ เวลาปัจจุบันโดยที่ $\hat{g}_{reg} : \mathrm{Z} \to \mathbb{R}$

$$\hat{g}_{reg}(f_n) = \frac{1}{2}\|L_{n+1} f_n - z_{n+1}\|^2 + \frac{\rho}{2}\|f_n\|^2 \tag{2.11}$$

โดยให้ค่าการประมาณเริ่มแรกไว้คือ f_0 ส่วนใหญ่จะกำหนดค่าเป็นศูน**ย์จากนั้นจะใช้วิธี สโตแค**สติกเกรเดียนเดสเซนต์ (SGD- Stochastic gradient descent) หาค่า **ฟังก์ชัน f ที่ทำให้ สมการ**ที่ (2.11) มีค่าต่ำสุดคือ

$$f_{n+1} = f_{n+1} - \eta_n \nabla \hat{g}_{reg}(f_n) \tag{2.12}$$

ซึ่ง $\nabla \hat{g}_{reg}(f_n)$ คือค่าเกรเดียน ณ เวลาปัจจุบัน ที่ f_n

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

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

ดังนั้น

$$f_{n+1} = (1-\eta_n\rho) f_n - \eta_n L^{*}_{n+1}(L_{n+1} f_n - z_n) \tag{2.13}$$

แต่สำหรับค่าคงที่ a จะได้ $L^{*}_{n+1}a = k_{n+1}a$ และ $L_{n+1}f_n = f_n(x_{n+1})$ เพราะฉะนั้นค่าเกรเดียนสามารถเขียนได้เป็น

$$\nabla \hat{g}_{reg}(f_n) = k_{n+1}\left[f_n(x_{n+1}) - z_{n+1}\right] + \rho f_n \tag{2.14}$$

แทนค่าในสมการที่ (2.13) จะได้การกระทำซ้ำหนึ่งของขั้นตอนคือ

$$f_{n+1} = (1-\eta_n\rho) f_n - \eta_n k_{n+1}\left[f_n(x_{n+1}) - z_{n+1}\right] \tag{2.15}$$

โดยอาศัยการเรียนรู้ f ในสมการที่ (2.5) และสมบัติว่าด้วยการประมาณค่าปัจจุบันมีเทอมเคอร์เนล ρ เทอมสมการที่ (2.15) สามารถเขียนใหม่ได้ดังนี้

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

$$= \sum_{i=1}^{p+1}\alpha_{n+1}^i k_i(x) \tag{2.16}$$

ซึ่งค่า α^i_{n+1} จะถูกคำนวณมาจาก

$$\alpha_{n+1}^i = (1-\eta_n\rho)\alpha_n^i \text{ สำหรับ } i \le p \tag{2.17}$$

และ

$$\alpha_{n+1}^i = -\eta_n e_{n+1} \text{ สำหรับ } i = p+1 \tag{2.18}$$

จากสมการที่ (2.16) สังเกตว่าขณะที่มีการเพิ่มค่าน้ำหนักเข้าใหม่ (α) ไปที่โมเดลของ f_{n+1} (โดยสัมพันธ์กับเทอมเคอร์เนล) ค่าน้ำหนักเก่าที่มีอยู่ก็ถูกอัพเดทด้วยเช่นกัน โดยถูกคูณด้วยเทอม 1 - $\eta_n\rho$ ซึ่งเป็นการลดค่าลงเหมือนกับความจำที่ลดลง

2.2 วิธีการหาแบบจำลองชนิดหลายแบบจำลอง (Multiple Model Algorithm)

จากหัวข้อที่แล้วแสดงให้เห็นว่าฟังก์ชันการประมาณค่า หรือ โมเดลที่ได้จะอยู่ในรูปผลบวกของฟังก์ชันเคอร์เนล $k(\cdot,\cdot)$ ซึ่งมีการใช้ฟังก์ชั่นเคอร์เนลที่หลากหลาย ซึ่งหนึ่งในนั้นก็คือเกาเสียนเคอร์เนล ที่มีลักษณะซับซ้อนน้อยที่สุด นั่นคือ

$$k(x,x') = \exp(\frac{\|x-x'\|^2}{2\sigma^2}) \tag{2.19}$$

ลักษณะปกติของฟังก์ชันเคอร์เนลคือการมีค่าไฮเปอร์พารามิเตอร์ ในกรณีนี้ σ สิ่งเหล่านี้ควบคุมลักษณะของการประมาณค่า เช่น ค่า σ มาก ๆ จะให้การประมาณค่าที่เรียบ ค่าต่อมาที่เราสนใจคือเรกกูลาไรเซชันพารามิเตอร์ (Regularisation parameter-ρ) ซึ่งถูกใช้ในการควบคุมความซับซ้อนของโมเดลค่า ไฮเปอร์พารามิเตอร์เหล่านี้จะต้องถูกเลือกให้เหมาะสม ดังนั้นเราจึงเสนอวิธีการหาหลายแบบจำลอง (Multiple model approach) แทนโดยวิธีที่แต่ละโมเดลในกลุ่มหนึ่งๆ ใช้ค่าไฮเปอร์พารามิเตอร์ที่แตกต่างกันและโมเดลคำตอบที่ได้ จะถูกรวมเข้าไว้ด้วยกันเพื่อสร้างฟังก์ชันการประมาณค่า

กำหนดให้มีแต่ละ M โมเดล ซึ่งแต่ละโมเดลสอดคล้องกับชุดไฮเปอร์พารามิเตอร์หนึ่งที่แตกต่างกัน จากนั้นจะใช้วิธีหัวข้อที่แล้วประมาณค่าพารามิเตอร์ของแต่ละโมเดล โดยใช้ค่าไฮเปอร์พารามิเตอร์ที่ถูกกำหนดไว้แต่แรกของแต่ละชุด

เราหมายความว่าเราแทนโมเดลที่ j ณ เวลา i โดย $f_{j,i}$ และการประมาณที่ได้ของโมเดลนี้ที่เวลา i+ρ ดังนั้นเราจะเขียนการประมาณค่าที่เวลา $i+\rho$ ของ หลายแบบจำลองโดยมีข้อมูลตั้งแต่ค่าเริ่มต้นจนถึงเวลา i ดังนี้

$$y_{i+p|i} = \sum_{j=0}^{M} w_{j,i}\hat{y}_{i+p}^{j} \tag{2.20}$$

ซึ่ง (w_{j+i}) ค่าน้ำหนักของแต่ละโมเดล (M) ที่ถูกคำนวณได้จากการใช้ค่าเฉลี่ยความผิดพลาดกำลังสองแบบเคลื่อนที่ (MASE- Moving Average Squared error) ของแต่ละโมเดลโดยคำนวณจาก

$$\eta_{j,i} = \sum_{l=0}^{R} (\hat{y}^{j}_{i-l+1|i-l} - y_{i-l+i})^2 \forall j \in [1, M] \tag{2.21}$$

ซึ่ง *R* คือค่าการคาดคะเนแนวนอนที่ถูกเลือกใช้ก่อน

สำหรับค่าเฉลี่ยความผิดพลาดกำลังสองแบบเคลื่อนที่ สำหรับโมเดลที่ j, $\eta_{j,i}$ คือการวัดค่าที่มีมาแล้วทั้งหมดของประสิทธิภาพการทำนายล่วงหน้าหนึ่งข้อมูล สำหรับข้อมูลจำนวน R ข้อมูล ที่ผ่านมา ค่าที่ต่ำกว่าสำหรับ $\eta_{j,i}$ ชี้ให้เห็นว่าโมเดล *i* ทำให้เกิดความถูกต้องมากกว่าสำหรับ *ค่า* R ที่เป็นศูนย์นั้น ค่าเฉลี่ยความผิดพลาดกำลังสองแบบเคลื่อนที่ จะเป็นการวัดค่าความถูกต้องของโมเดล ณ เวลาปัจจุบัน ซึ่งเราจะไม่ใช้ค่านี้เป็นตัวบ่งบอกประสิทธิภาพของโมเดลเนื่องจากไม่มีการเฉลี่ยค่าของประสิทธิภาพโมเดลที่มีต่อข้อมูลหลาย ๆ ค่า ดังนั้นเราควรกำหนดให้ R มากกว่าศูนย์ แต่อย่างไรก็ตามค่าที่จะเลือกก็ยังคงเป็นคำถามที่เปิดอยู่ ในทางเดียวกันจำนวนโมเดล M ก็ยังคงเป็นปัญหาในการเลือกเหมือนกันยังคงเหลือเพียงพารามิเตอร์ซึ่งดังนั้นเราจึงกล่าวได้ว่า R และ M เป็นเมตาพารามิเตอร์ (Metaparameter)

ค่าเฉลี่ยความผิดพลาดกำลังสองแบบเคลื่อนที่ที่เป็นจำนวนตัวเลขที่กลับกัน ได้ให้ไว้โดย

$$\gamma_{j,i} = \frac{1}{\eta_{j,i}} \forall j \in [1, M] \tag{2.22}$$

และท้ายที่สุดการให้ค่าน้ำหนักสำหรับโมเดลที่ *j* ก็ถูกคำนวณโดยใช้

$$w_{j,i} = \frac{\gamma_{j,i}}{\sum_j \gamma_{j,i}} \forall j \in [1, M] \tag{2.23}$$

โดยการนิยามการให้น้ำหนักโมเดลเป็นการรับรองดังที่เราได้คาดไว้ ซึ่งก็คือ $\sum_{j=1}^{M} w_{j,i} = 1$ สำหรับทุก ๆ ค่า i

วิธีการที่กล่าวมาแล้วนี้เป็นการรวมค่าการทำนาย (j) ของแต่ละโมเดลเข้าด้วยกันต่อเนื่อง โมเดลที่แสดงผลที่ดีที่สุดตลอดช่วงค่า R จะให้น้ำหนักที่มากกว่าอย่างสอดคล้องกัน นั่นคือค่าการทำนายมีลักษณะเอียงไปหาโมเดลตัวที่ดีที่สุด

บทที่ 3
การออกแบบระบบด้วยวิธีการหาแบบจำลองหลายแบบจำลอง

โครงงานนี้ได้มุ่งเน้นศึกษาการหาฟังก์ชันการประมาณค่าหรือแบบจำลองที่เป็นตัวแทนของระบบที่ไม่รู้จัก โดยอาศัยข้อมูลที่เก็บค่าได้จากระบบนั้นๆ ด้วยวิธีการหาแบบจำลองหลายแบบจำลอง (Multiple Model Algorithm) ซึ่งมีรายละเอียดดังนี้

3.1 หลักการออกแบบระบบด้วยวิธีการหาแบบจำลองหลายแบบจำลอง

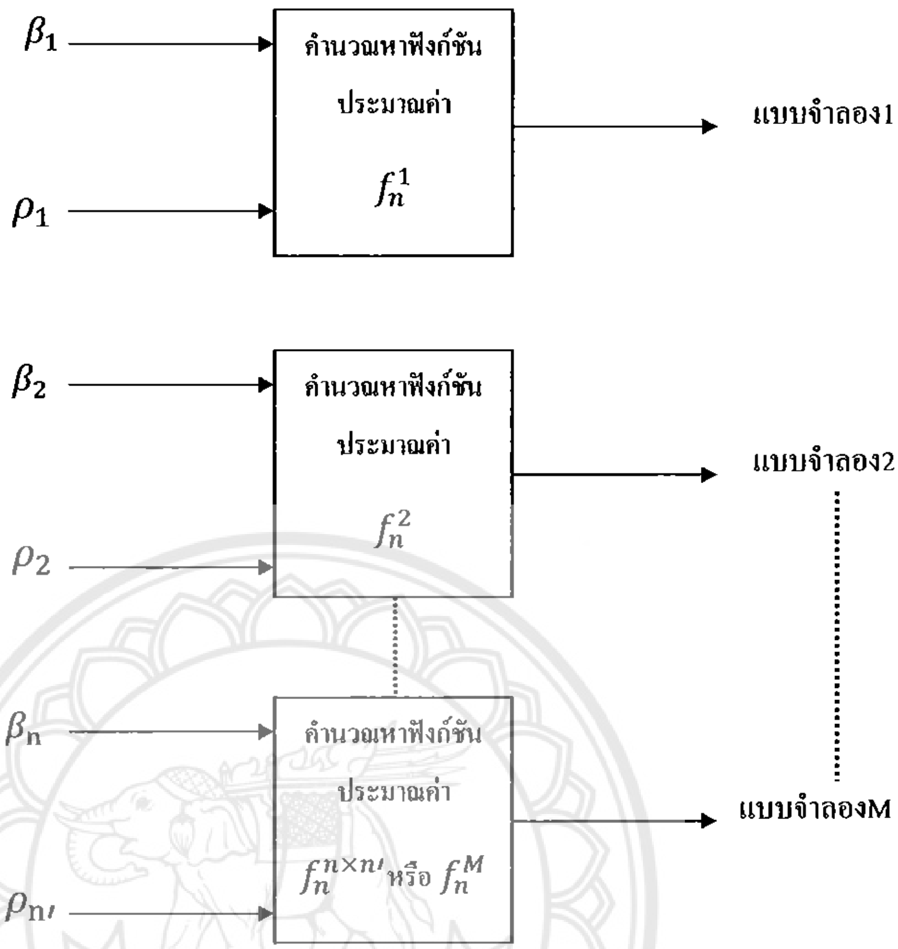
การใช้วิธีการหาแบบจำลองหลายแบบจำลองเป็นการระบุเซตของค่าไฮเปอร์พารามิเตอร์สำหรับการคำนวณหาฟังก์ชันการประมาณค่าโดยที่แต่ละค่าของไฮเปอร์พารามิเตอร์จะหมายถึงหนึ่งแบบจำลองในการทำนายในอนาคตจะใช้ค่าจากแต่ละแบบจำลองมาคิดรวมกัน ซึ่งจะมีหลักการในการให้คำนวณน้ำหนักของแต่ละแบบจำลองที่เหมาะสมโดยพิจารณาจากค่าที่ผิดพลาดน้อยที่สุด

ขั้นตอนในการออกแบบวิธีการหาแบบจำลองหลายแบบจำลอง มีดังนี้

1. เก็บข้อมูลสัญญาณขาเข้าและสัญญาณขาออกของระบบที่ต้องการหาฟังก์ชันการประมาณค่า
2. สร้างฟังก์ชันการประมานค่าของระบบโดยใช้สมการที่ (2.16)
3. สร้างค่าเซตของไฮเปอร์พารามิเตอร์ที่จะใช้ในแต่ละแบบจำลอง
4. ให้ค่าน้ำหนักกับแต่ละแบบจำลอง จากการพิจารณาค่าความผิดพลาดของแต่ละแบบจำลองโดยใช้สมการที่ (2.23)

ดังนั้น เมื่อเสร็จสิ้นกระบวนการออกแบบระบบ จะสามารถทำนายค่าในอนาคตโดยใช้ผลรวมของแต่ละแบบจำลอง

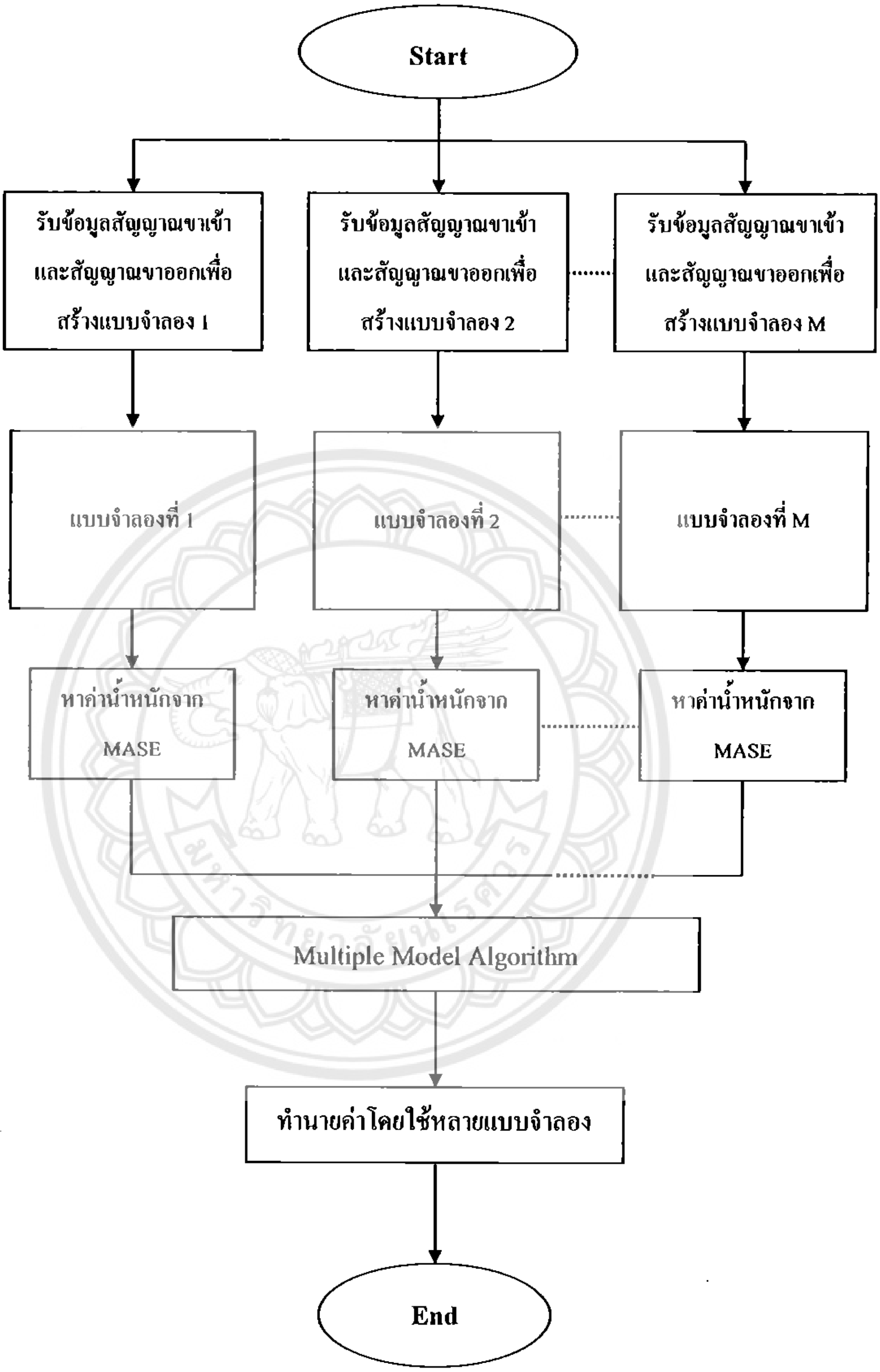
กำหนดให้ไฮเปอร์พารามิเตอร์ที่เรากำลังสนใจคือ β และ ρ โดยที่ β หมายถึงขนาดของโมเดลและกำหนดไว้ n ค่า ρ หมายถึงระยะเวลา กำหนดไว้ n' ค่าดังนั้นจะได้จำนวนแบบจำลองทั้งหมด M แบบจำลองนั้นคือสัญญาณขาออกในเวลาที่ต่อเนื่องที่ใช้ในการออกแบบ $n \times n'$ ชุด คือ $\beta_1, \beta_2, \ldots\ldots, \beta_n$ และ $\rho_1, \rho_2, \ldots\ldots, \rho_{n'}$ จะได้ว่า ขั้นตอนในการออกแบบระบบด้วยวิธีการหาแบบจำลองหลายแบบจำลอง M แบบจำลอง คือ



รูปที่ 3.1 การหาแบบจำลองโดยใช้ฟังก์ชันการประมาณค่า

จากค่าไฮเปอร์พารามิเตอร์ใช้ β จำนวน n ค่าและ ρ จำนวน n' ค่า จะได้ค่าไฮเปอรพารามิเตอร์ของแต่ละแบบจะลองของฟังก์ชันการประมาณค่า คือ (β_1, ρ_1), $(\beta_2, \rho_2), \ldots, (\beta_n, \rho_{n'})$ จะได้แบบจำลองจำนวน M ชุด

เมื่อป้อนสัญญาณขาเข้าและสัญญาณขาออกชุดที่ 1 ก็จะได้แบบจำลองที่ 1 ออกมาและเมื่อป้อนสัญญาณขาเข้าและสัญญาณขาออกชุดที่ 2 ก็จะได้แบบจำลองที่ 2 ดังนั้น ถ้ามีข้อมูลของสัญญาณขาเข้าและสัญญาณขาออกเป็นจำนวน M ชุด จะได้ระบบที่ได้จากการคำนวณหาฟังก์ชันประมาณค่าเป็นจำนวน M ชุด ดังรูปที่ 3.1



รูปที่ 3.2 การทำนายค่าที่ได้จากวิธีการหาแบบจำลองชนิดหลายแบบจำลอง

การใช้หลายแบบจำลองในการทำนายค่า มีวิธีการ โดยละเอียดดังต่อไปนี้

ทำการป้อนข้อมูลจำนวน M ชุดเข้าตัวรับสัญญาณขาเข้าและสัญญาณขาออกตั้งแต่ชุดที่ 1 ถึงชุดที่ M จากนั้นจะผ่านแบบจำลองและสมการหาค่าน้ำหนักจาก (MASE) จากชุดที่ 1 ถึง M เช่นเดียวกัน จากนั้นจะป้อนเข้าสมการหาแบบจำลองหลายแบบจำลอง (Multiple Model Algolithm) ทีละ M ชุดเลย เพราะว่าเราจะใช้ค่าโมเดลทุกชุด และก็จะเข้าตัวทำนายค่าโดยใช้หลายแบบจำลองเพื่อหาโมเดลตัวที่มีความผิดพลาดน้อยที่สุด

เนื่องจากระบบที่เหมาะสมที่สุดสำหรับข้อมูลของสัญญาณขาเข้าและสัญญาณขาออกทั้ง M ชุดควรเป็นระบบที่มีความผิดพลาดน้อยที่สุดเพื่อให้ได้สัญญาณขาออกเป็นไปตามที่กำหนดไว้ ดังนั้นเมื่อได้ค่าผิดพลาดของแต่ละระบบแล้ว ทำการเปรียบเทียบค่าผิดพลาดของแต่ละระบบเพื่อหาระบบที่มีความผิดพลาดน้อยที่สุด เลือกระบบดังกล่าวเป็นระบบที่มีความเหมาะสมที่สุดสำหรับข้อมูลทั้ง M ชุด จากรูปที่ 3.2

บทที่ 4

ผลการทดลองและการวิเคราะห์ผล

ในบทนี้จะกล่าวถึงผลการทดลองของวิธีการประมาณค่าฟังก์ชั่นด้วยวิธีเคอร์เนลและวิธีการหาแบบจำลองชนิดหลายแบบจำลองที่ได้ศึกษามา ซึ่งจะทำการเปลี่ยนค่าพารามิเตอร์ต่างๆ และดูผลที่เกิดขึ้น โดยแต่ละค่าพารามิเตอร์ที่แตกต่างกันจะให้ผลของค่าความผิดพลาด (ค่าความผิดพลาดเฉลี่ยกำลังสองหรือ MSE) ที่แตกต่างกันออกไป แต่สิ่งที่ต้องการคือมีการลดลงของค่าความผิดพลาดและการลดลงต้องค่อยๆลดลงซึ่งเป็นที่ต้องการ

4.1 ผลการทดลองการประมาณค่าฟังก์ชันด้วยวิธีเคอร์เนล

การประมาณค่าฟังก์ชันด้วยวิธีเคอร์เนลนั้นสามารถทำได้โดยใช้สมการที่ (2.16) และจะใช้ฟังก์ชันเกาเสียนเรเดียลเบสิสดังสมการที่ (2.6) เป็นฟังก์ชันเคอร์เนล ในการทดลองการประมาณค่าฟังก์ชัน โดยจะใช้โปรแกรมแมทแลปในการประมาณค่าฟังก์ชัน และคำนวณหาค่าความผิดพลาดจากข้อมูลที่ใช้ทดสอบ ได้ดังสมการต่อไปนี้

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

เมื่อ n คือจำนวนของจำนวนข้อมูลของค่าความผิดพลาด

ข้อมูลที่ใช้ทดสอบนี้ถูกสร้างขึ้นโดยใช้โปรแกรมแมทแลป ซึ่งแบ่งข้อมูลออกเป็น 2 ชุด คือที่ใช้ชุดสอนจำนวน 200 ค่า และชุดที่ใช้ทดสอบจำนวน 50 ค่า

ในการทดลองนี้จะมีการปรับค่าพารามิเตอร์ต่างๆได้แก่ ค่าเรกกูลาไรเซชัน ค่าอัตราการเรียนรู้ และค่าเบต้า เพื่อให้ได้กราฟของความผิดพลาด ซึ่งจะมีหลักการในการเลือกโดยพิจารณาจากค่าความผิดพลาดที่น้อยที่สุดของระบบ โดยจะมีการปรับค่าพารามิเตอร์ โดยใช้วิธีการหาแบบจำลองชนิดหลายแบบจำลอง ซึ่งวิธีนี้จะต่างกับวิธีที่ใช้แบบจำลองเดียวจุดประสงค์คือ ในการสร้างแบบจำลองเราไม่รู้ค่าที่ดีที่สุดของพารามิเตอร์ต่างๆ ดังนั้นเราจึงเลือกค่าที่คิดว่าดีที่สุดของพารามิเตอร์แต่ละตัวมาจำนวนหนึ่ง เพื่อสร้างเป็นแบบจำลองหลายๆแบบจำลอง กล่าวคือ จะมีการให้ค่าเรกกูลาไรเซชันหลายค่าเท่ากับจำนวนแบบจำลอง และให้ค่าเบต้าเท่ากับจำนวนแบบจำลองเช่นเดียวกันแต่จะมีค่าอัตราการเรียนรู้เพียงค่าเดียว ซึ่งต่างจากแบบจำลองเดียวคือจะมีการให้ค่าพารามิเตอร์เพียงอย่างละ 1 ค่าเท่านั้นคือ เรกกูลาไรเซชัน 1 ค่า เบต้า 1 ค่า และค่าอัตราการ

เรียนรู้ 1 ค่า ซึ่งก็จะเป็นค่าที่เหมาะสมที่ได้จากการสอน และในวิธีการหาเอาท์พุตที่ได้จากแบบจำลองชนิดหลายแบบจำลองนั้น จะเป็นการเฉลี่ยค่าเอาท์พุตได้จากแต่ละแบบจำลอง โดยเป็นการเฉลี่ยแบบให้ค่าน้ำหนัก ในการหาแบบจำลองเดียวไม่มีการให้ค่าน้ำหนักของแบบจำลอง ซึ่งในการทดลองนั้น จะมีการกำหนดค่าเรกกูลาไรเซชัน ค่าเบต้า ค่าอัตราการเรียนรู้ ขึ้นมาเพื่อกำหนดรูปแบบของแบบจำลอง หรือกราฟที่ได้จากการรันของโปรแกรมแมทแลป ดังต่อไปนี้

ค่าอัตราการเรียนรู้ (η) คือ ช่วงของการลู่เข้าสู่ค่าความผิดพลาดที่ศูนย์

ค่าเรกกุลาไรเซชั่น (ρ) คือ ความซับซ้อนของแบบจำลอง

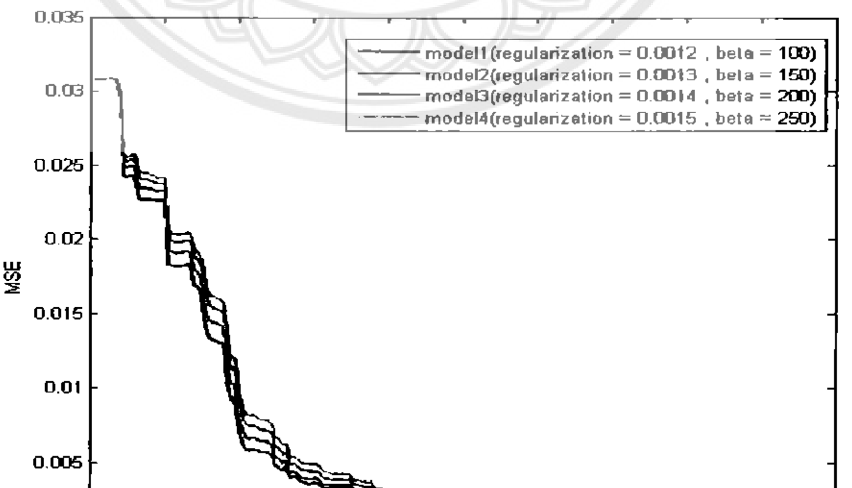
ค่าเบต้า (β) คือ ความกว้างของเคอร์เนลฟังก์ชัน

4.1.1 ผลของการปรับค่าอัตราการเรียนรู้

โดยเราจะปรับค่าอัตราการเรียนรู้ตั้งแต่ 0.1- 0.2 โดยเพิ่มทีละ 0.025 เพื่อหาค่าพารามิเตอร์ที่ดีที่สุด โดยที่ค่าเรกกูลาไรเซชันและค่าเบต้าของแต่ละแบบจำลองเราจะกำหนดมาดังต่อไปนี้

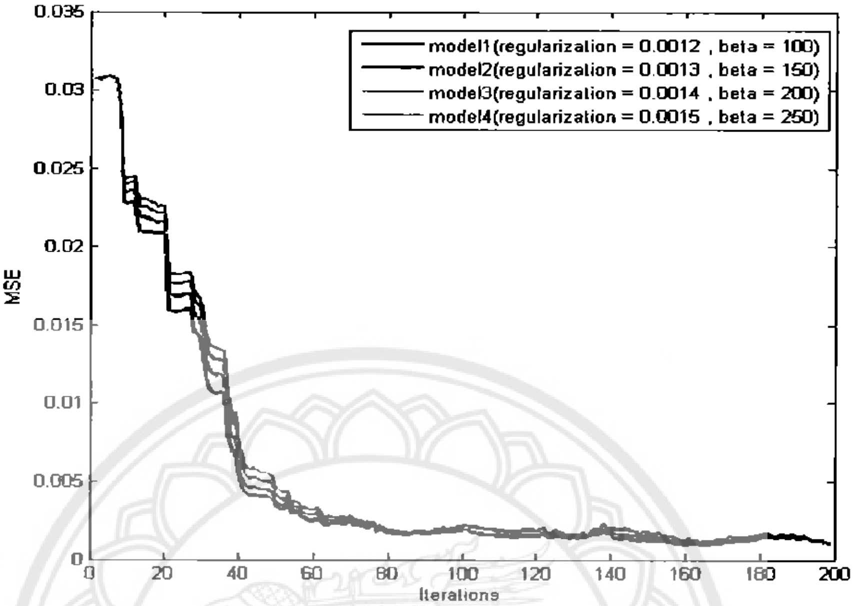
- แบบจำลองที่ 1 ค่าเรกกูลาไรเซชันเท่ากับ 0.0012 ค่าเบต้าเท่า 100
- แบบจำลองที่ 2 ค่าเรกกูลาไรเซชันเท่ากับ 0.0013 ค่าเบต้าเท่า 150
- แบบจำลองที่ 3 ค่าเรกกูลาไรเซชันเท่ากับ 0.0014 ค่าเบต้าเท่า 200
- แบบจำลองที่ 4 ค่าเรกกูลาไรเซชันเท่ากับ 0.0015 ค่าเบต้าเท่า 250

เมื่อปรับค่าอัตราการเรียนรู้เท่ากับ 0.1 จะได้ค่าพารามิเตอร์ของแต่ละแบบจำลอง ดังรูปที่ 4.1



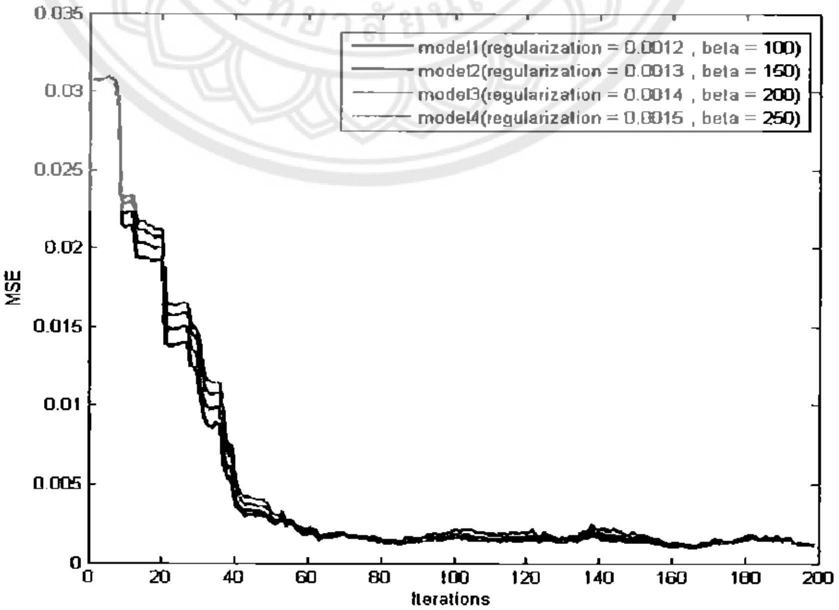
รูปที่ 4.1 กราฟแสดงผลของแต่ละแบบจำลองเมื่อปรับค่าอัตราการเรียนรู้เป็น 0.1

เมื่อปรับค่าอัตราการเรียนรู้เท่ากับ 0.125 จะได้ค่าพารามิเตอร์ของแต่ละแบบจำลอง ดังรูปที่ 4.2



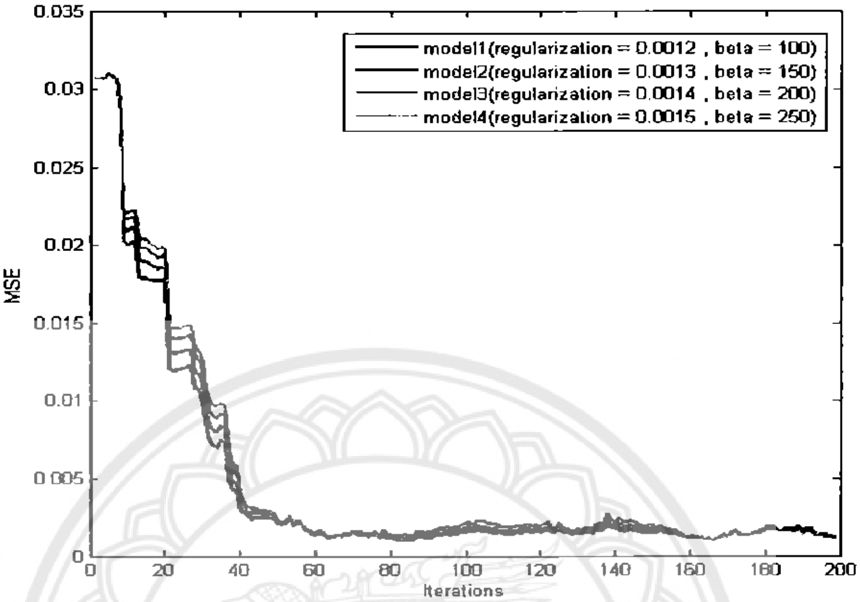
รูปที่ 4.2 กราฟแสดงผลของแต่ละแบบจำลองเมื่อปรับค่าอัตราการเรียนรู้เป็น 0.125

เมื่อปรับค่าอัตราการเรียนรู้เท่ากับ 0.150 จะได้ค่าพารามิเตอร์ของแต่ละแบบจำลอง ดังรูปที่ 4.3



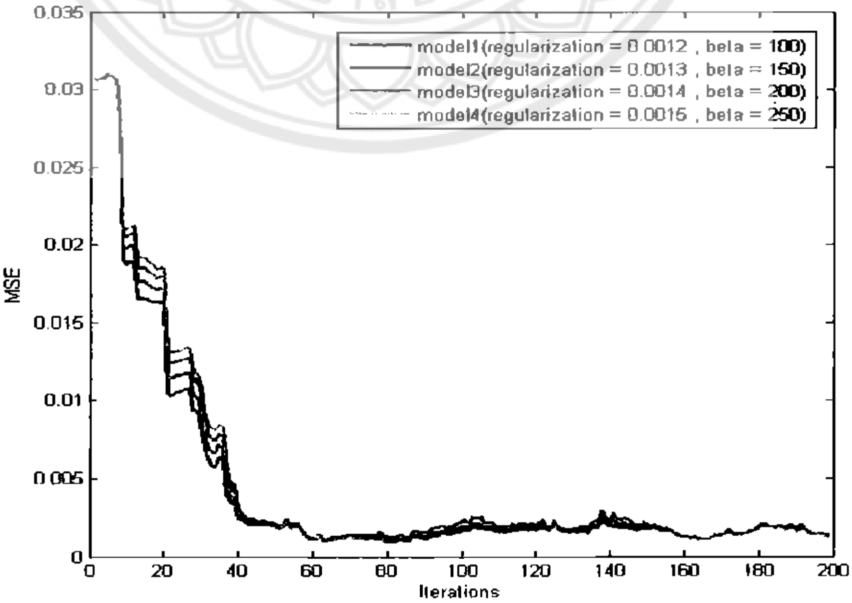
รูปที่ 4.3 กราฟแสดงผลของแต่ละแบบจำลองเมื่อปรับค่าอัตราการเรียนรู้เป็น 0.150

เมื่อปรับค่าอัตราการเรียนรู้เท่ากับ 0.175 จะได้ค่าพารามิเตอร์ของแต่ละแบบจำลอง ดังรูปที่ 4.4



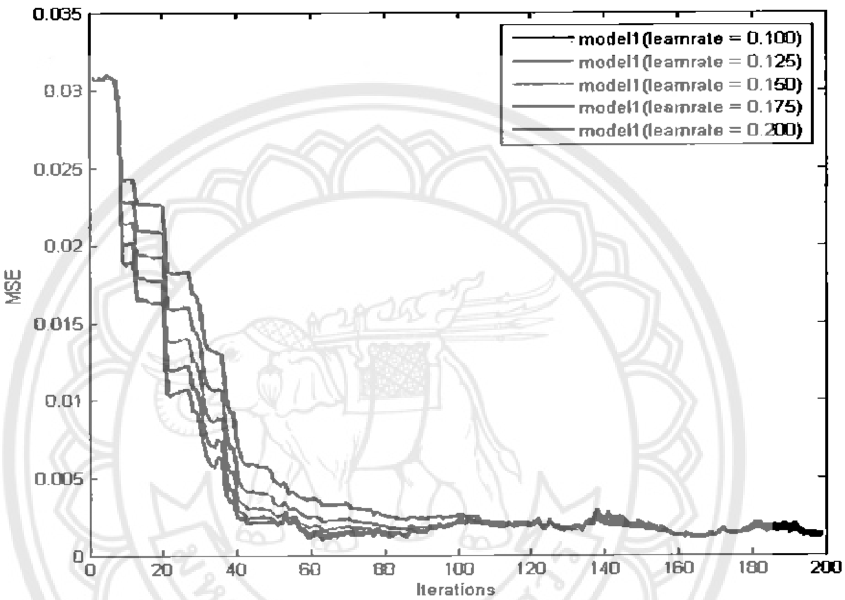
รูปที่ 4.4 กราฟแสดงผลของแต่ละแบบจำลองเมื่อปรับค่าอัตราการเรียนรู้เป็น 0.175

เมื่อปรับค่าอัตราการเรียนรู้เท่ากับ 0.2 จะได้ค่าพารามิเตอร์ของแต่ละแบบจำลอง ดังรูปที่ 4.5



รูปที่ 4.5 กราฟแสดงผลของแต่ละแบบจำลองเมื่อปรับค่าอัตราการเรียนรู้เป็น 0.2

จากกราฟแสดงผลของการปรับค่าอัตราการเรียนรู้ของทั้งหมด 4 แบบจำลองที่ได้มานั้น จะนำมาเลือกเหลือกราฟเพียงแค่เส้นเดียว คือ กราฟของแบบจำลองที่ 1 และให้มีการปรับค่าอัตราการเรียนทั้งหมด 5 ค่า คือ 0.100 , 0.125 , 0.150 , 0.175 และ 0.200 ค่าเรกกูลาไรเซชันเท่ากับ 0.0012 และค่าเบต้าเท่ากับ 100 เพื่อให้ง่ายต่อการสังเกตการเปลี่ยนแปลงของกราฟว่ามีการเปลี่ยนแปลงไปในลักษณะใด ดังจะได้เห็นในรูปที่ 4.6



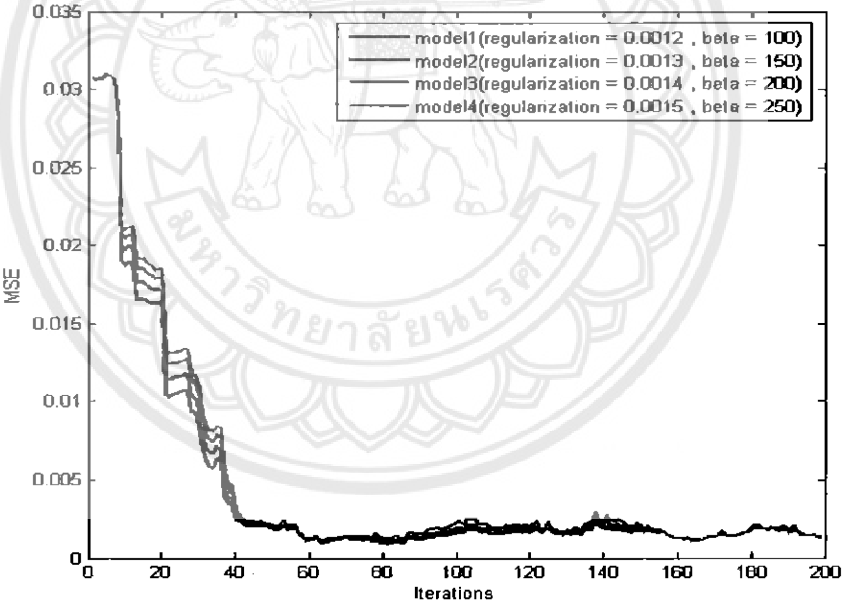
รูปที่ 4.6 กราฟแสดงผลของการปรับค่าอัตราการเรียนรู้ของแบบจำลองที่ 1

จากกราฟแสดงผลของการปรับค่าการเรียนรู้ ที่ค่า 0.1 กราฟของแบบจำลองที่ 1 จะมีความชันน้อยกว่าเมื่อค่าเข้าใกล้ 0.2 ที่มีความชันค่อนข้างมากดังนั้นจึงสังเกตได้ว่าค่าอัตราการเรียนรู้จะเป็นตัวกำหนดความชันของกราฟของแบบจำลองที่ 1 ดังนั้นเราจึงเลือกค่าอัตราการเรียนรู้ ที่ค่า 0.2 เพราะว่าความชันมากจะทำให้มีการเรียนรู้ที่เร็วกว่าค่าที่มีความชันน้อยเพราะจะเรียนรู้ได้ช้าเห็นได้จากกราฟที่แสดงดังรูปที่ 4.6

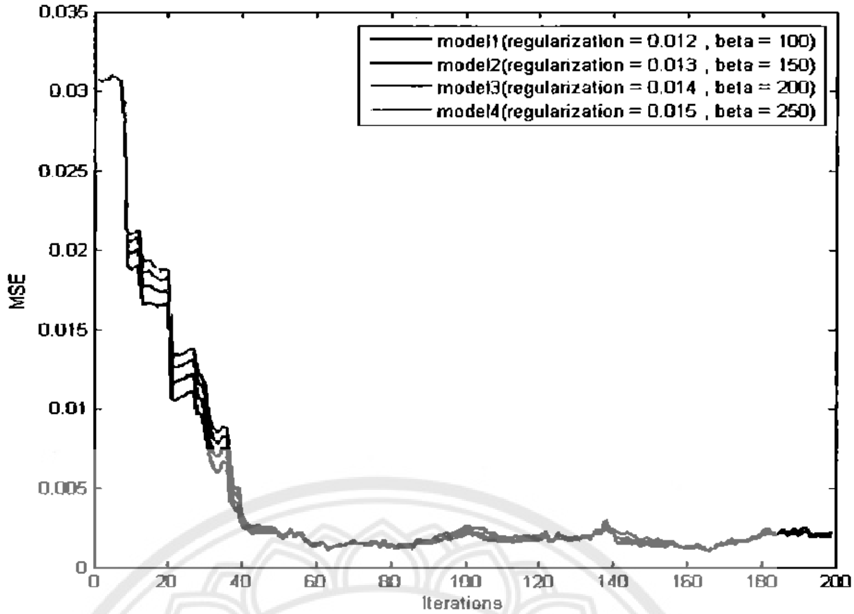
4.1.2 ผลของการปรับค่าเรกกูลาไรเซชัน

ในการปรับค่าเรกกูลาไรเซชันนั้นเราจะกำหนดให้ค่าอัตราการเรียนรู้เท่ากับ 0.2 คงที่และค่าเบต้าของแต่ละแบบจำลองมีค่าเท่าเดิม คือ 100 , 150 , 200 , 250 แล้วทำการปรับค่าเรกกูลาไรเซชันให้เข้าใกล้ 1.00 ซึ่งจะมีการปรับค่าเรกกูลาไรเซชันดังต่อไปนี้

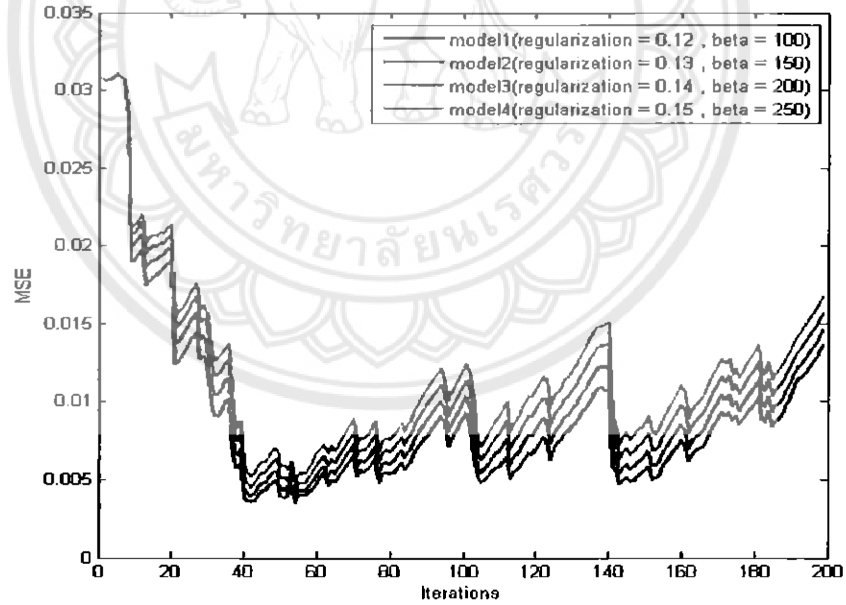
- ปรับค่าเรกกูลาไรเซชันให้เท่ากับ 0.0012 , 0.0013 , 0.0014 0.0015 ตามลำดับของแต่ละแบบจำลอง แล้วหาค่าความผิดพลาด ผลของกราฟแสดงได้ดังรูปที่ 4.7
- ปรับค่าเรกกูลาไรเซชันให้เท่ากับ 0.012 , 0.013 , 0.014 0.015 ตามลำดับของแต่ละแบบจำลอง แล้วหาค่าความผิดพลาด ผลของกราฟแสดงได้ดังรูปที่ 4.8
- ปรับค่าเรกกูลาไรเซชันให้เท่ากับ 0.12 , 0.13 , 0.14 0.15 ตามลำดับของแต่ละแบบจำลอง แล้วหาค่าความผิดพลาด ผลของกราฟแสดงได้ดังรูปที่ 4.9



รูปที่ 4.7 กราฟแสดงผลของการปรับค่าเรกกูลาไรเซชันของแต่ละแบบจำลอง



รูปที่ 4.8 กราฟแสดงผลของการปรับค่าเรกกูลาไรเซชันของแต่ละแบบจำลอง

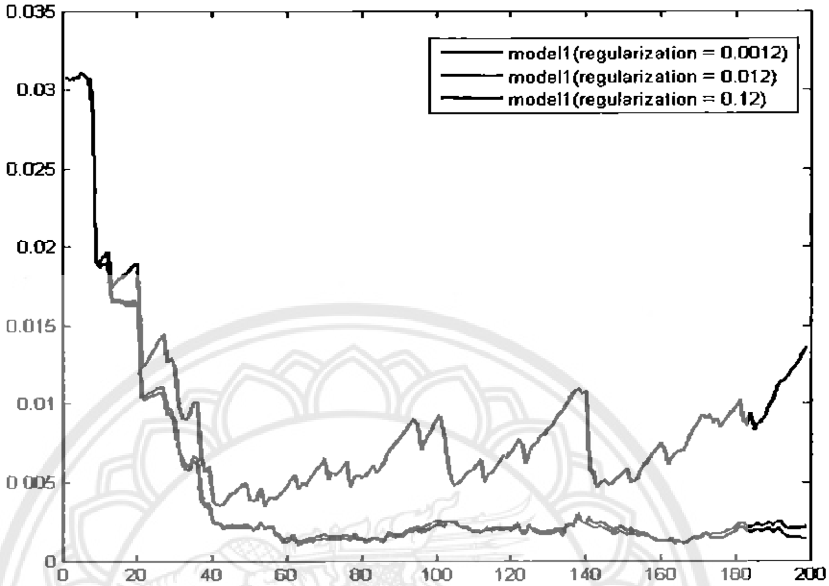


รูปที่ 4.9 กราฟแสดงผลของการปรับค่าเรกกูลาไรเซชันของแต่ละแบบจำลอง

จากกราฟแสดงผลการปรับค่าเรกกูลาไรเซชันที่ได้มาทั้งหมดนั้น จะนำมาเลือกให้เหลือเพียงแค่หนึ่งกราฟตัวอย่าง เพื่อให้ง่ายต่อการเปรียบเทียบกราฟขณะปรับค่าเรกกูลาไรเซชัน ว่าค่าความผิดพลาดจะมีลักษณะเปลี่ยนแปลงเป็นเช่นไร ในที่นี้จะเลือกกราฟการแสดงผลของ

ห้องสมุดคณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

แบบจำลองที่ 1 ที่มีค่าอัตราการเรียนรู้เท่ากับ 0.2 ค่าเบต้าเท่ากับ 100 และทำการปรับค่าเรกกูลาไรเซชันคือ 0.0012 , 0.012 และ 0.12 ตามลำดับ และสังเกตการเปลี่ยนแปลงดังรูปที่ 4.10



15733339
2552

รูปที่ 4.10 กราฟแสดงผลการปรับค่าเรกกูลาไรเซชันของแบบจำลองที่ 1

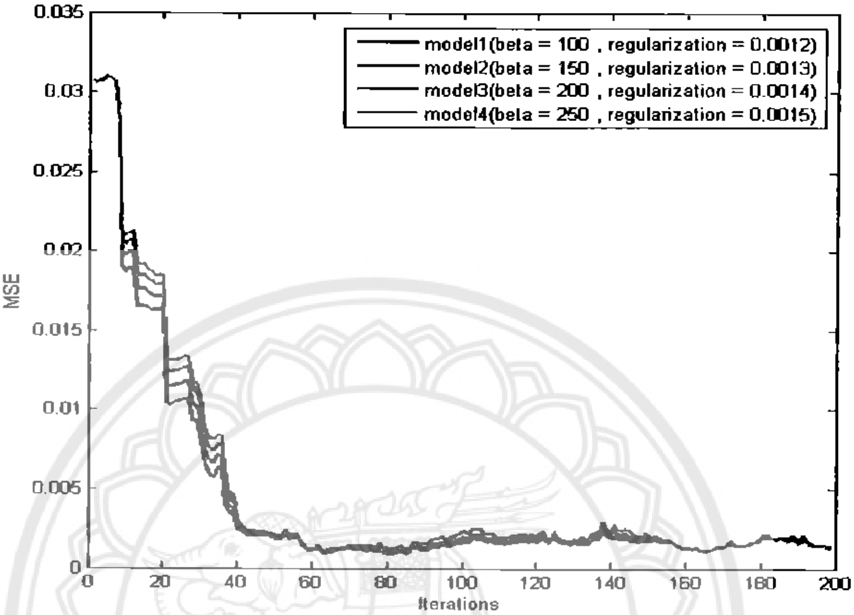
จากกราฟที่แสดงผลการปรับค่าเรกกูลาไรเซชันของแต่ละแบบจำลองนั้น จะเห็นได้ว่าเมื่อปรับค่าเรกกูลาไรเซชันให้เข้าใกล้ 1.00 ความราบเรียบของกราฟที่แสดงก็จะเปลี่ยนไปโดยค่าที่เข้าใกล้ 1.00 มีความราบเรียบน้อยลงกว่าค่าเรกกูลาไรเซชันที่ต่ำกว่า 1.00 ดังนั้นจึงแสดงให้เห็นว่าการปรับค่าเรกกูลาไรเซชันมีผลต่อความราบเรียบของค่าความผิดพลาด

4.1.3 ผลของการปรับค่าเบต้า

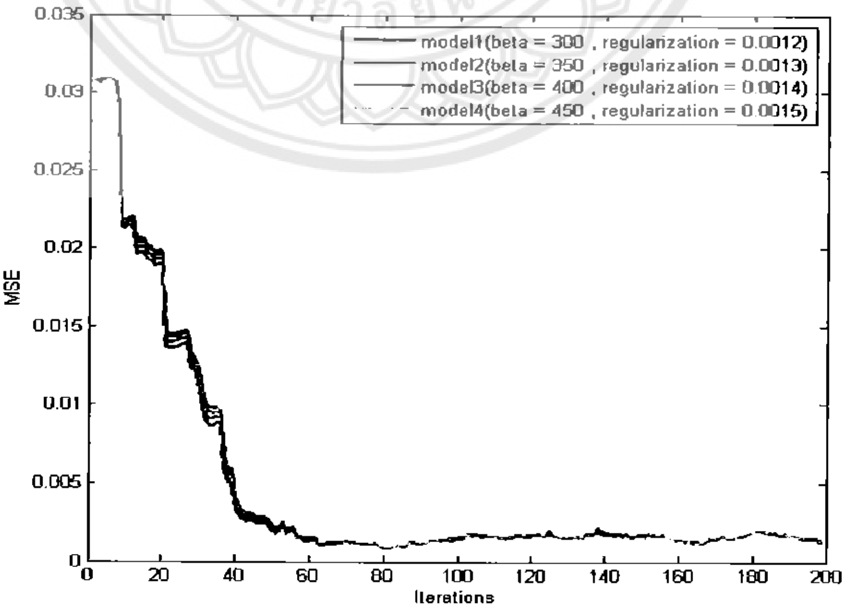
ในการปรับค่าเบต้านั้นเราจะกำหนดให้ค่าเรกกูลาไรเซชันเท่าเดิม คือ 0.0012 , 0.0013 , 0.0014 และ 0.0015 ของแต่ละแบบจำลองตามลำดับ และค่าอัตราการเรียนรู้เท่ากับ 0.2 คงที่ แล้วทำการปรับค่าเบต้า

- ปรับค่าเบต้าให้เท่ากับ 100 , 150 , 200 , 250 ตามลำดับของแต่ละแบบจำลอง แล้วหาค่าความผิดพลาด ผลของกราฟแสดงได้ดังรูปที่ 4.11
- ปรับค่าเบต้าให้เท่ากับ 300 , 350 , 400 , 450 ตามลำดับของแต่ละแบบจำลอง แล้วหาค่าความผิดพลาด ผลของกราฟแสดงได้ดังรูปที่ 4.12

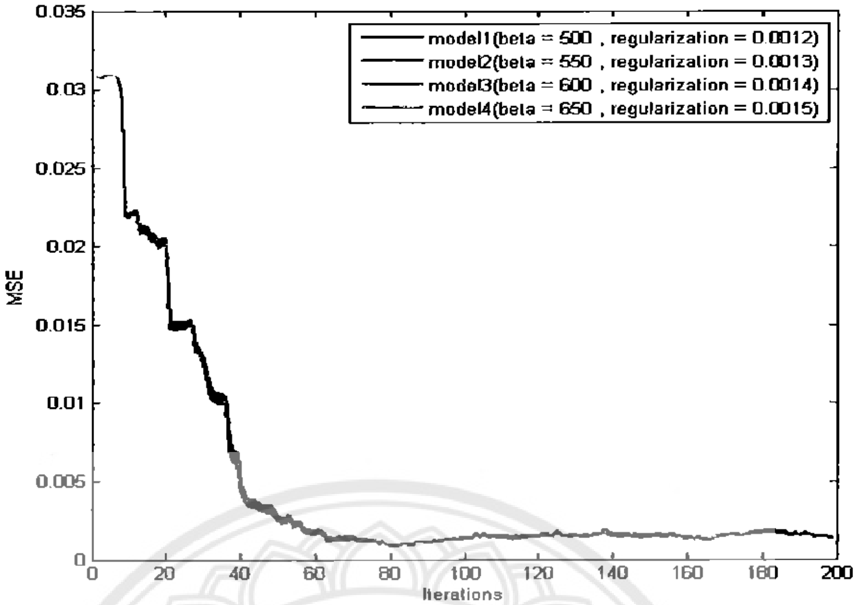
- ปรับค่าเบต้าให้เท่ากับ 500 , 550 , 600 , 650 ตามลำดับของแต่ละแบบจำลอง แล้วหาค่าความผิดพลาด ผลของกราฟแสดงได้ดังรูปที่ 4.13



รูปที่ 4.11 กราฟแสดงการปรับค่าเบต้าของแต่ละแบบจำลอง

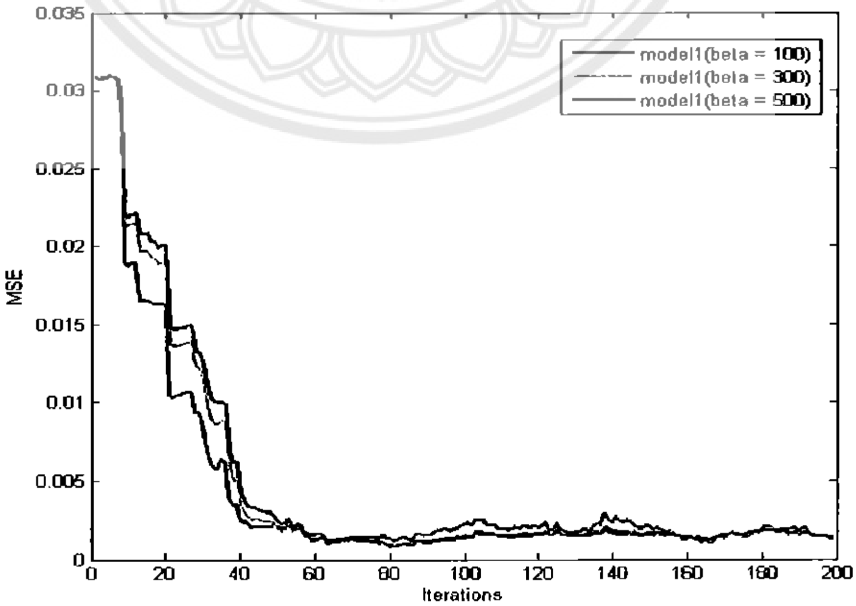


รูปที่ 4.12 กราฟแสดงการปรับค่าเบต้าของแต่ละแบบจำลอง



รูปที่ 4.13 กราฟแสดงการปรับค่าเบต้าของแต่ละแบบจำลอง

จากกราฟแสดงผลการปรับค่าเบต้าที่ได้มาทั้งหมดนั้น จะนำมาเลือกให้เหลือเพียงแค่หนึ่งกราฟตัวอย่าง เพื่อให้ง่ายต่อการเปรียบเทียบกราฟขณะปรับค่าเบต้า ว่ากราฟจะมีลักษณะเปลี่ยนแปลงเป็นเช่นไร ในที่นี้จะเลือกกราฟการแสดงผลของแบบจำลองที่ 1 ที่มีค่าอัตราการเรียนรู้เท่ากับ 0.2 ค่าเรกกูลาไรเซชันเท่ากับ 0.0012 และทำการปรับค่าเบต้าคือ 100 , 300 และ 500 ตามลำดับ และสังเกตการเปลี่ยนแปลงดังรูปที่ 4.14



รูปที่ 4.14 กราฟแสดงผลการปรับค่าเบต้าของแบบจำลองที่ 1

จากกราฟแสดงการปรับค่าเบต้าของแต่ละแบบจำลอง จะสังเกตได้ว่าค่าเบต้าที่ปรับให้มีค่าเพิ่มขึ้นมีความชันของกราฟน้อยลงจึงทำให้ทราบว่าแบบจำลองในแต่ละแบบจำลองจะมีค่าความผิดพลาดเพิ่มขึ้น ดังนั้นค่าเบต้าจะเป็นตัวบ่งชี้ค่าความผิดพลาดที่เพิ่มขึ้นเมื่อมีค่าเพิ่มขึ้นดังกราฟที่แสดงในรูปที่ 4.11 , 4.12 และ 4.13

4.2 ค่าความผิดพลาดของการประมาณค่าฟังก์ชันด้วยวิธีการหาแบบจำลองหลายแบบจำลอง

วิธีการหาแบบจำลองหลายแบบจำลองนี้เราจะกำหนดข้อมูลทั้งหมด 200 ค่า สำหรับใช้ใน 4 แบบจำลอง โดยเราจะกำหนดค่าเรกกูลาไรเซชันและค่าเบต้าของแต่ละแบบจำลองขึ้นมาอย่างละ 4 ค่า ซึ่งค่าของแต่ละแบบจำลองจะมีความแตกต่างกัน ในการประมาณค่าฟังก์ชันแบบจำลองจะมีชุดสอนและชุดทดสอบซึ่งในที่นี้ชุดสอนจะมี 200 ค่าและชุดทดสอบมี 50 ค่าจะกำหนดให้ค่าอัตราการเรียนรู้เท่ากับ 0.2 คงที่แล้วหาค่าความผิดพลาดของแต่ละแบบจำลองแล้วต่อจากนั้นก็นำค่าที่ได้มาหาค่าเฉลี่ยความผิดพลาดของแต่ละแบบจำลองโดยคำนวณได้จากสมการที่ (2.21)

ค่าความผิดพลาดของแต่ละแบบจำลองจะมีค่าไม่เท่ากันขึ้นอยู่กับการกำหนดค่าเรกกูลาไรเซชันกับเบต้าของแต่ละแบบจำลองทั้ง 4 แบบแสดงดังต่อไปนี้

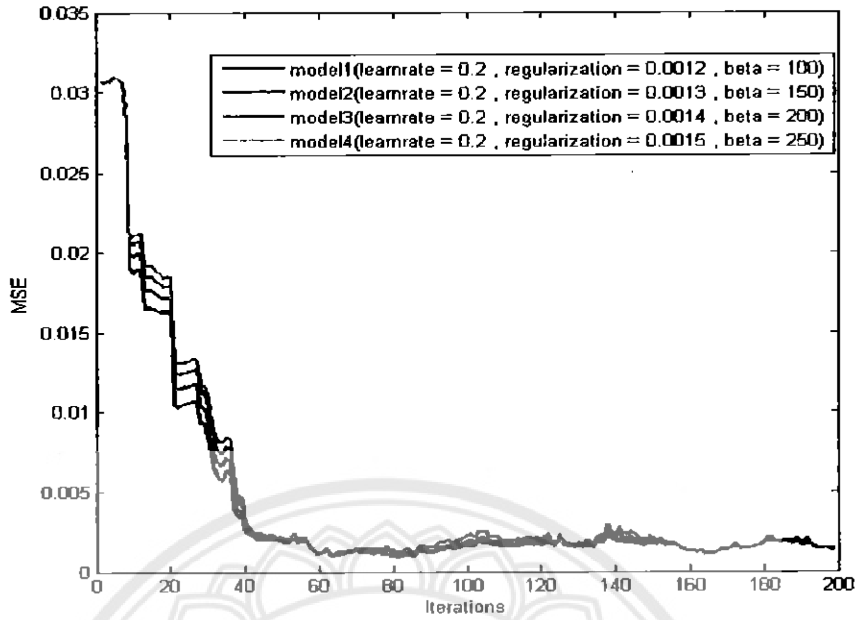
แบบจำลองที่ 1 ค่าเรกกูลาไรเซชันเท่ากับ 0.0012 และค่าเบต้าเท่ากับ 100

แบบจำลองที่ 2 ค่าเรกกูลาไรเซชันเท่ากับ 0.0013 และค่าเบต้าเท่ากับ 150

แบบจำลองที่ 3 ค่าเรกกูลาไรเซชันเท่ากับ 0.0014 และค่าเบต้าเท่ากับ 200

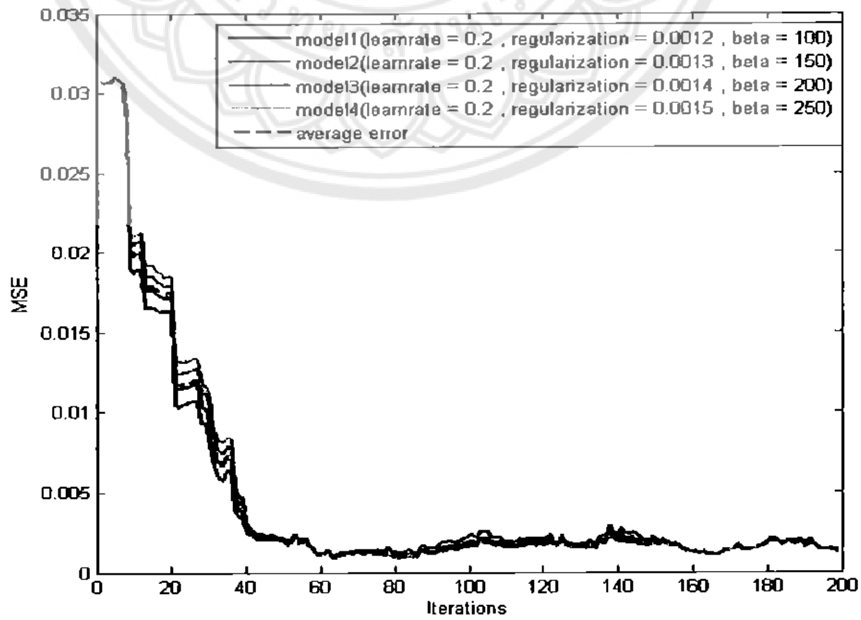
แบบจำลองที่ 4 ค่าเรกกูลาไรเซชันเท่ากับ 0.0015 และค่าเบต้าเท่ากับ 250

ดังนั้นค่าความผิดพลาดของแต่ละแบบจำลองจะมีค่าดังกราฟค่าความผิดพลาดที่แสดงในรูปที่ 4.15



รูปที่ 4.15 กราฟแสดงการเปรียบเทียบค่าความผิดพลาดของแบบจำลอง **ทั้ง 4 แบบจำลอง**

เมื่อเราได้ค่าความผิดพลาดของแต่ละแบบจำลองมา แล้วก็นำค่าที่ได้มาคำนวณหาค่าน้ำหนักของแต่ละแบบจำลองว่าแบบจำลองใดมีความน่าเชื่อถือมากที่สุดจะมีค่าน้ำหนักที่มาก ซึ่งการให้น้ำหนักของแบบจำลองจะหาได้จากสมการที่ (2.23)



รูปที่ 4.16 กราฟแสดงการให้น้ำหนักของแบบจำลองที่เหมาะสมที่สุด

จะเห็นได้ว่าแบบจำลองที่ได้จากฟังก์ชันการประมาณค่า ทั้ง 4 แบบจำลองจะนำไปคำนวณต่อโดยใช้วิธีการหาแบบจำลองชนิดหลายแบบจำลองโดยนำค่าทั้ง 4 ของแบบจำลองมารวมกันและให้น้ำหนัก โดยที่ผลของการคำนวณของวิธีการหาแบบจำลองชนิดหลายแบบจำลองจะให้ค่าความผิดพลาดออกมาตามเส้น average error ซึ่งเป็นการเฉลี่ยค่าความผิดพลาดของแต่ละแบบจำลองแบบให้ค่าน้ำหนัก

บทที่ 5
สรุปผลการทดลอง

ในบทนี้จะกล่าวถึงผลของการทดลองที่เกิดขึ้นในการหาค่าความผิดพลาดด้วยวิธีการประมาณค่าฟังก์ชันด้วยวิธีเคอร์เนลและการประมาณค่าที่ได้ด้วยวิธีการแบบจำลองชนิดหลายแบบจำลองที่ได้ศึกษามา ว่าจะได้ค่าความผิดพลาดที่เพิ่มขึ้นหรือลดลงอย่างไรเพื่อที่จะหาวิธีการที่สามารถลดค่าความผิดพลาดที่ดีที่สุด

5.1 วิธีการประมาณค่าฟังก์ชันด้วยวิธีเคอร์เนล

5.1.1 กรณีของการปรับค่าอัตราการเรียนรู้

การปรับค่าการเรียนรู้โดยเริ่มปรับค่าตั้ง 0.1 จนถึง 0.2 นั้นจะแสดงให้เห็นว่ามีผลต่อลักษณะของกราฟ กล่าวคือ ถ้าปรับให้ค่าอัตราการเรียนรู้ให้เข้าใกล้ 0.1 กราฟจะมีความชันน้อยและค่อยลดลงเข้าหาศูนย์ ซึ่งจะเป็นกราฟที่ไม่เหมาะสมเนื่องจากความชันน้อยแสดงว่ามีการเรียนรู้ที่ช้า แต่ถ้าปรับค่าอัตราการเรียนรู้ให้เข้าใกล้สอง 0.2 กราฟก็จะมีความชันมากและค่อยๆลดลงเข้าหาศูนย์ ซึ่งจะเป็นกราฟที่มีความเหมาะสมเพราะเป็นกราฟที่มีความชันมากแสดงว่าเรียนรู้ได้เร็ว ดังที่แสดงในรูปที่ 4.1-4.6 ดังนั้นในกรณีนี้จึงเลือกค่าอัตราการเรียนรู้ที่ 0.2 ซึ่งเป็นค่าที่มีความเหมาะสมที่สุด

5.1.2 กรณีผลของการปรับค่าเรกกูลาไรเซชัน

การปรับค่าเรกกูลาไรเซชันนั้นเราจะกำหนดให้ค่าอัตราการเรียนรู้เท่ากับ 0.2 และค่าเบต้าทั้ง 4 ของแต่ละแบบจำลองมีค่าเท่าเดิม แล้วทำการปรับค่าเรกกูลาไรเซชันให้เข้าใกล้ 1.00

- ปรับค่าเรกกูลาไรเซชันให้เท่ากับ 0.0012 , 0.0013 , 0.0014 และ 0.0015 ตามลำดับของแต่ละแบบจำลอง แล้วหาค่าความผิดพลาด
- ปรับค่าเรกกูลาไรเซชันให้เท่ากับ 0.012 , 0.013 , 0.014 และ 0.015 ตามลำดับของแต่ละแบบจำลอง แล้วหาค่าความผิดพลาด
- ปรับค่าเรกกูลาไรเซชันให้เท่ากับ 0.12 , 0.13 , 0.14 และ 0.15 ตามลำดับของแต่ละแบบจำลองแล้วหาค่าความผิดพลาด

ผลที่ได้จากการปรับค่าเรกกูลาไรเซชันจะมีความราบเรียบน้อยที่สุดจะเห็นได้ ดังรูปที่ 4.9 และในกรณีนี้เราจึงต้องเลือกค่าเรกกูลาไรเซชันที่ มีค่าให้เข้าใกล้ศูนย์มากที่สุด จึงจะทำให้กราฟมีความราบเรียบที่เหมาะสมที่สุด

5.1.3 กรณีผลของการปรับค่าเบต้า

การปรับค่าเบต้านั้นเราจะกำหนดให้ค่าเรกกูลาไรเซชันเท่าเดิมในแต่ละแบบจำลอง และค่าอัตราการเรียนรู้เท่ากับ 0.2 แล้วจึงทำการปรับค่าเบต้า ดังนี้

- ปรับค่าเบต้าให้เท่ากับ 100 , 150 , 200 และ 250 ตามลำดับของแต่ละแบบจำลอง แล้วหาค่าความผิดพลาด
- ปรับค่าเบต้าให้เท่ากับ 300 , 350 , 400 และ 450 ตามลำดับของแต่ละแบบจำลอง แล้วหาค่าความผิดพลาด
- ปรับค่าเบต้าให้เท่ากับ 500 , 550 , 600 และ 650 ตามลำดับของแต่ละแบบจำลอง แล้วหาค่าความผิดพลาด

ผลที่ได้จากการปรับค่าเบต้า ในการทดลองจะปรับค่าให้เพิ่มขึ้นและเมื่อค่าเพิ่มขึ้นก็จะทำให้กราฟมีความชันน้อยลงเรื่อย ๆ ซึ่งก็จะทำให้ทราบว่าเมื่อปรับค่าเบต้าเพิ่มขึ้นก็จะทำให้ค่าความผิดพลาดเพิ่มขึ้นตามขนาดของค่าเบต้า ซึ่งในขณะเดียวกันถ้าค่าเบต้ามีความชันมากจนเกินไปหรือมีการลดค่าเบต้าจนน้อยเกินไปก็จะทำให้มีค่าความผิดพลาดเพิ่มขึ้นได้อีกด้วย เพราะฉะนั้นเราจึงควรเลือกค่าที่เหมาะสม คือไม่มากเกินไปและก็ไม่น้อยจนเกินไป ดังค่าที่ได้เลือกมาใช้ในการทดสอบ

5.2 วิธีการประมาณค่าฟังก์ชันด้วยการหาแบบจำลองชนิดหลายแบบจำลอง

วิธีการประมาณค่าฟังก์ชันด้วยวิธีการหาแบบจำลองชนิดหลายแบบจำลอง จะเป็นการหาโดยใช้วิธีแยกเป็น 4 แบบจำลอง โดยจะกำหนดค่าเรกกูลาไรเซชัน 4 ค่าและค่าเบต้า 4 ค่าเช่นเดียวกัน โดยที่จะกำหนดค่าอัตราการเรียนรู้เป็นค่าคงที่เพียงค่าเดียว เพื่อนำค่าที่ได้ในแต่ละแบบจำลองมาหาค่าความผิดพลาดของแต่ละแบบจำลอง หลังจากนั้นก็จะนำค่าความผิดพลาดที่ได้ทั้งหมดของแต่ละแบบจำลอง มาหาค่าความผิดพลาดของแบบจำลองทั้ง 4 แบบจำลอง แบบให้ค่าน้ำหนัก ซึ่งจะคำนวณมาจากค่าความผิดพลาดของแต่ละแบบจำลอง เอาท์พุตที่ได้จากวิธีหลายแบบจำลองจะเป็นการนำเอาเอาท์พุตของแต่ละแบบจำลองมาเฉลี่ยตามค่าน้ำหนักและค่าเฉลี่ยความผิดพลาดที่ได้จะใกล้เคียงกับแบบจำลองที่ดีที่สุด เพราะมีค่าน้ำหนักมากที่สุด

ค่าสุดท้ายที่ได้จากการทดสอบหาค่าเฉลี่ยความผิดพลาด โดยคิดจากค่าเฉลี่ยของน้ำหนักจะมีเพียงแบบจำลองเดียวเท่านั้นที่มีค่าใกล้เคียงกับค่าเฉลี่ยความผิดพลาด ซึ่งก็แสดงให้เห็นดังแสดงในรูปที่ 4.16 นั้นก็คือแบบจำลองที่ 2 ที่มีค่าใกล้เคียงที่สุด

5.3 ปัญหาและข้อเสนอแนะเพิ่มเติม

ในการทดลองการประมาณค่าฟังก์ชันด้วยวิธีการหาแบบจำลองชนิดหลายแบบจำลองนั้น เมื่อข้อมูลที่ใช้ในการทดลองมีจำนวนมากแล้วใช้โปรแกรมแมทแลปในการคำนวณหาค่าความผิดพลาดจะทำให้การประมวลผลของคอมพิวเตอร์ล่าช้ามาก แนวทางแก้ไขคือค้นคว้าหาวิธีการคำนวณที่ลดการคำนวณที่ซ้ำซ้อนกันให้มากที่สุด และถ้าต้องการความละเอียดในการหาค่าความผิดพลาดให้มากที่สุดนั้นจะต้องทำการเพิ่มจำนวนแบบจำลองให้มากขึ้น จึงจะได้ความละเอียดที่มากตามไปด้วย

สำหรับโปรแกรมการทดลองทั้งหมดจะรวมอยู่ในแผ่นซีดีที่แนบมาพร้อมกับปริญญานิพนธ์เล่มนี้แล้ว ซึ่งสามารถนำไปใช้ในการทดลองหรือนำไปประยุกต์ใช้งานต่อไป

เอกสารอ้างอิง

[1] T.J. Dodd , B. Mitchinson and R.F. Harrison **"Sequential Kernel Methods: A Multiple Model Approach to Hyperparameters,"** ; Department of Automatic Control and Systems Engineering; The University of Sheffield, No. 830; January 2003

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

[3] T. 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] สัญญฉกร วุฒิสิทธิกุลกิจ และคณะ. **" MATLABการประยุกต์ใช้งานทางวิศวกรรมไฟฟ้า",** พิมพ์ครั้งที่ 3, สำนักพิมพ์แห่งจุฬาลงกรณ์มหาวิทยาลัย, 2551.

ภาคผนวก

โปรแกรมที่ใช้ในการทดลอง

โปรแกรมที่ใช้ในการทดลองในบทที่ 4

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

x_train = rand(1000,1); %ข้อมูลที่ใช้สอน (x)
y_train = sinc((20*x_train-10)/pi);
y_train = y_train + 0.2*randn(size(x_train )); %ข้อมูลที่ใช้สอน (y)


x_test  = [0.1:0.1:5]'; %ข้อมูลที่ใช้ทดสอบ (x)
y_idealtest = sinc((20*x_test-10)/pi); %ข้อมูลที่ใช้ทดสอบ (y)

no_data = size(x_train,1);
learnrate  =0.1; %ปรับค่าอัตราการเรียนรู้เท่ากับ 0.1 - 0.2 ตามลำดับ

%กำหนดค่าพารามิเตอร์
beta1=100; %กำหนดให้ค่าเบต้า
rho1=0.0012; %กำหนดให้ค่าเรกกูลาไรเซชัน
beta2=150; %กำหนดให้ค่าเบต้า
rho2=0.0013; %กำหนดให้ค่าเรกกูลาไรเซชัน
beta3=200; %กำหนดให้ค่าเบต้า
rho3=0.0014; %กำหนดให้ค่าเรกกูลาไรเซชัน
beta4=250; %กำหนดให้ค่าเบต้า
rho4=0.0015; %กำหนดให้ค่าเรกกูลาไรเซชัน

for n=1;
    f=[0]; %กำหนดค่าเริ่มต้น
    e(n)=-y_train(n);
    alpha1=(-learnrate)*e(n);
```

```
    alpha2=(-learnrate)*e(n);
    alpha3=(-learnrate)*e(n);
    alpha4=(-learnrate)*e(n);
for i=1:50;
     xtest = x_test(i);
     yidealtest=y_idealtest(i);
     y_predict1(i)=alpha1*exp(-beta1*(abs((x_train(n)-xtest))).^2);
     error1(i)=(y_predict1(i)-yidealtest)^2; %หาค่าความผิดพลาดของแบบจำลองที่ 1

     y_predict2(i)=alpha2*exp(-beta2*(abs((x_train(n)-xtest))).^2);
     error2(i)=((y_predict2(i)-yidealtest)^2); %หาค่าความผิดพลาดของแบบจำลองที่ 2

     y_predict3(i)=alpha3*exp(-beta3*(abs((x_train(n)-xtest))).^2);
     error3(i)=((y_predict3(i)-yidealtest)^2); %หาค่าความผิดพลาดของแบบจำลองที่ 3

     y_predict4(i)=alpha4*exp(-beta4*(abs((x_train(n)-xtest))).^2);
     error4(i)=((y_predict4(i)-yidealtest)^2); %หาค่าความผิดพลาดของแบบจำลองที่ 4

end
     eta1(n) = sum(error1);
     gamma1(n) = 1/(eta1(n));
     eta2(n) = sum(error2);
     gamma2(n) = 1/(eta2(n));
     eta3(n) = sum(error3);
     gamma3(n) = 1/(eta3(n));
     eta4(n) = sum(error4);
     gamma4(n) = 1/(eta4(n));

     fff = (gamma1(n)+gamma2(n)+gamma3(n)+gamma4(n));
```

```
%ให้น้ำหนักแบบจำลอง
weigth1(n) = gamma1(n)/fff;
weigth2(n) = gamma2(n)/fff;
weigth3(n) = gamma3(n)/fff;
weigth4(n) = gamma4(n)/fff;

y_mm = (weigth1(n)*y_predict1)+(weigth2(n)*y_predict2)+(weigth3(n)*y_predict3)+
    (weigth4(n)*y_predict4);
MSE_mm(n) = sum((y_mm'-y_idealtest).^2)/length(y_idealtest);

f_predict1=alpha1*exp(-beta1*(abs(x_train(n)-x_train(n+1))).^2);
f_predict2=alpha2*exp(-beta2*(abs(x_train(n)-x_train(n+1))).^2);
f_predict3=alpha3*exp(-beta3*(abs(x_train(n)-x_train(n+1))).^2);
f_predict4=alpha4*exp(-beta4*(abs(x_train(n)-x_train(n+1))).^2);

MSE1=sum(error1)/length(y_idealtest); %ค่าเฉลี่ยความผิดพลาดกำลังสองของแบบจำลองที่
1 โดยคิดจากการให้น้ำหนัก
MSE2=sum(error2)/length(y_idealtest); %ค่าเฉลี่ยความผิดพลาดกำลังสองของแบบจำลองที่
2 โดยคิดจากการให้น้ำหนัก
MSE3=sum(error3)/length(y_idealtest); %ค่าเฉลี่ยความผิดพลาดกำลังสองของแบบจำลองที่
3 โดยคิดจากการให้น้ำหนัก
MSE4=sum(error4)/length(y_idealtest); %ค่าเฉลี่ยความผิดพลาดกำลังสองของแบบจำลองที่
4 โดยคิดจากการให้น้ำหนัก

end

for n=2:199
    e(n)=f_predict1-y_train(n);
    alpha1=[alpha1*(1-learnrate*rho1); -learnrate*e(n)];
    e(n)=f_predict2-y_train(n);
    alpha2=[alpha2*(1-learnrate*rho2); -learnrate*e(n)];
```

```
e(n)=f_predict3-y_train(n);
alpha3=[alpha3*(1-learnrate*rho3); -learnrate*e(n)];
e(n)=f_predict4-y_train(n);
alpha4=[alpha4*(1-learnrate*rho4); -learnrate*e(n)];

for i=1:50
    xtest = x_test(i);
    yidealtest = y_idealtest(i);
  for a=1:n
      ypredict1(a)=alpha1(a)*exp(-beta1*(abs(x_train(a)- x_test(i))).^2);
      ypredict2(a)=alpha2(a)*exp(-beta2*(abs(x_train(a)- x_test(i))).^2);
      ypredict3(a)=alpha3(a)*exp(-beta3*(abs(x_train(a)- x_test(i))).^2);
      ypredict4(a)=alpha4(a)*exp(-beta4*(abs(x_train(a)- x_test(i))).^2);
  end
      y_predict1(i)= sum(ypredict1);
      y_predict2(i)= sum(ypredict2);
      y_predict3(i)= sum(ypredict3);
      y_predict4(i)= sum(ypredict4);
      error1(i)=((y_predict1(i)-y_idealtest(i))^2); %หาค่าความผิดพลาดของแบบจำลองที่ 1
      error2(i)=((y_predict2(i)-y_idealtest(i))^2); %หาค่าความผิดพลาดของแบบจำลองที่ 2
      error3(i)=((y_predict3(i)-y_idealtest(i))^2); %หาค่าความผิดพลาดของแบบจำลองที่ 3
      error4(i)=((y_predict4(i)-y_idealtest(i))^2); %หาค่าความผิดพลาดของแบบจำลองที่ 4


end
      eta1(n) = sum(error1);
      gamma1(n) = 1/(eta1(n));
      eta2(n) = sum(error2);
      gamma2(n) = 1/(eta2(n));
      eta3(n) = sum(error3);
      gamma3(n) = 1/(eta3(n));
```

```
eta4(n) = sum(error4);
gamma4(n) = 1/(eta4(n));
fff = (gamma1(n)+gamma2(n)+gamma3(n)+gamma4(n));

% ให้น้ำหนักแบบจำลอง
weigth1(n) = gamma1(n)/fff;
weigth2(n) = gamma2(n)/fff;
weigth3(n) = gamma3(n)/fff;
weigth4(n) = gamma4(n)/fff;
y_mm = (weigth1(n)*y_predict1)+(weigth2(n)*y_predict2)+(weigth3(n)*y_predict3)+
    (weigth4(n)*y_predict4);
MSE_mm(n) = sum((y_mm' -y_idealtest).^2)/length(y_idealtest);
MSE1(n)=sum(error1)/length(y_idealtest); %ค่าเฉลี่ยความผิดพลาดกำลังสองของ
แบบจำลองที่ 1 โดยคิดจากการให้น้ำหนัก
MSE2(n)=sum(error2)/length(y_idealtest); %ค่าเฉลี่ยความผิดพลาดกำลังสองของ
แบบจำลองที่ 2 โดยคิดจากการให้น้ำหนัก
MSE3(n)=sum(error3)/length(y_idealtest); %ค่าเฉลี่ยความผิดพลาดกำลังสองของ
แบบจำลองที่ 3 โดยคิดจากการให้น้ำหนัก
MSE4(n)=sum(error4)/length(y_idealtest); %ค่าเฉลี่ยความผิดพลาดกำลังสองของ
แบบจำลองที่ 4 โดยคิดจากการให้น้ำหนัก
for j=1:n
    f_predict1(j)=alpha1(j)*exp(-beta1*(abs(x_train(n+1)-x_train(j))).^2);
    f_predict2(j)=alpha2(j)*exp(-beta2*(abs(x_train(n+1)-x_train(j))).^2);
    f_predict3(j)=alpha3(j)*exp(-beta3*(abs(x_train(n+1)-x_train(j))).^2);
    f_predict4(j)=alpha4(j)*exp(-beta4*(abs(x_train(n+1)-x_train(j))).^2);
end
f_predict1=sum(f_predict1);
f_predict2=sum(f_predict2);
f_predict3=sum(f_predict3);
f_predict4=sum(f_predict4);
```

```
end
figure
plot(MSE1)    %กราฟค่าเฉลี่ยความผิดพลาดกำลังสองของแบบจำลองที่ 1 โดยคิดจากการให้น้ำหนัก
hold on
plot(MSE2,'r')  %กราฟค่าเฉลี่ยความผิดพลาดกำลังสองของแบบจำลองที่ 2 โดยคิดจากการให้น้ำหนัก
plot(MSE3,'g') %กราฟค่าเฉลี่ยความผิดพลาดกำลังสองของแบบจำลองที่ 3 โดยคิดจากการให้น้ำหนัก
plot(MSE4,'c') %กราฟค่าเฉลี่ยความผิดพลาดกำลังสองของแบบจำลองที่ 4 โดยคิดจากการให้น้ำหนัก
figure
plot(MSE_mm) %กราฟค่าเฉลี่ยความผิดพลาดกำลังสองรวมทั้ง 4 แบบจำลอง
```