สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

การประยุกต์ใช้นิวรอลเน็ตเวิร์กในการพยากรณ์หุ้น

AN APPLICATION OF NEURAL NETWORKS IN FORECASTING STOCK

โดย
นายอติเทพ เอื้อถาวรสุข
นายอดิศักดิ์ ก่อเกษมพร

เลขหมู่.....
เลขทะเบียน 46281
วัน, เดือน, ปี 21 มี.ค. 2546

.b.....
.i.....

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาอิเล็กทรอนิกส์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2544

การประยุกต์ใช้นิวรอลเน็ตเวิร์กในการพยากรณ์หุ้น

AN APPLICATION OF NEURAL NETWORKS IN FORECASTING STOCK

โดย

นายอติเทพ เอื้อถาวรสุข เลขประจำตัว 41014502

นายอดิศักดิ์ ก่อเกษมพร เลขประจำตัว 41014503

อาจารย์ที่ปรึกษา

ดร.ยุทธนา คิดใจเดียว

ปริญญานิพนธ์สำหรับปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาอิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2544

ปริญญานิพนธ์ปีการศึกษา 2544

ภาควิชา อิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง การประยุกต์ใช้นิวรอลเน็ตเวิร์กในการพยากรณ์หุ้น

ผู้จัดทำ

1. นายอติเทพ เอื้อถาวรสุข เลขประจำตัว 41014502
2. นายอดิศักดิ์ ก่อเกษมพร เลขประจำตัว 41014503

.....อาจารย์ที่ปรึกษา

(ดร.ยุทธนา คิดใจเดียว)

การประยุกต์ใช้นิวรอลเน็ตเวิร์กในการพยากรณ์หุ้น

An Application of Neural Network in Forecasting Stock

1. นายอดิเทพ เอื้อถาวรสุข เลขประจำตัว 41014502

2. นายอดิศักดิ์ ก่อเกษมพร เลขประจำตัว 41014503

โครงงานได้รับการตรวจสอบแล้ว พร้อมที่จะทำการสอบได้

.....

(ดร.ยุทธนา คิดใจเดียว)

อาจารย์ที่ปรึกษา

การประยุกต์ใช้นิวรอลเน็ตเวิร์กในการพยากรณ์หุ้น

นายอติเทพ เอื้อถาวรสุข
นายอดิศักดิ์ ก่อเกษมพร
ดร.ยุทธนา คิดใจเดียว อาจารย์ที่ปรึกษา
ปีการศึกษา 2544

บทคัดย่อ

ข้อมูลจากการพยากรณ์หุ้นเป็นปัจจัยที่สำคัญในการเลือกซื้อหุ้นของนักเล่นหุ้น ดังนั้นความถูกต้องแม่นยำของการพยากรณ์หุ้นจึงมีความสำคัญต่อนักเล่นหุ้นเป็นอย่างมาก

โครงงานนี้เป็นคู่มือประกอบการสร้างโปรแกรมการพยากรณ์ข้อมูลหุ้น โดยใช้โครงข่ายประสาทเทียม โครงข่ายประสาทเทียมนี้เป็นแนวทางหนึ่งที่ใช้แก้ปัญหาคอมพิวเตอร์ ซึ่งโครงข่ายประสาทเทียมนี้ช่วยในการแก้ปัญหาข้อมูลที่มีความสลับซับซ้อนจากกระบวนการพัฒนาในแบบคลาสสิกเนื่องจากไม่สามารถสรุปเป็นขั้นตอนที่ชัดเจนได้ โดยรายงานนี้ภายในจะประกอบด้วยทฤษฎีโครงข่ายประสาทเทียม ซึ่งจะจำแนกชนิดตามลักษณะการเรียนรู้ และการจำแนกตามลักษณะการใช้งาน และการพยากรณ์ข้อมูลกราฟ ซึ่งจำแนกได้เป็นทฤษฎีและขั้นตอนในการแปลงข้อมูลกราฟไปเป็นข้อมูลไบนารีเพื่อนำไปใช้ในกระบวนการจำและวิเคราะห์ในโครงข่ายประสาทเทียม และสามารถพยากรณ์ผลในจุดต่อไปได้เรื่อยๆ อย่างต่อเนื่อง

An Application of Neural Network in Forecasting Stock

Atithep Euathavornsuk

Adisak Korkasamporn

Dr.Yuttana Kidjaidrew Advisor

2001

ABSTRACT

Information derived from forecasting stock is essential for stock market buying choice. Therefore, the accuracy in forecasting the stock is very important to investor very much.

This project is the handbook of production and designing of time series data stock prediction which is processed by Neural Network. This Neural Network is one choice for programmer to solve the complicated problem which takes place from Classic algorithm that can not conclude in the clear way. This report consists of Neural Network Theory which can be described by means of learning process by ways of duty. In addition, data prediction can be sorted out by theory and steps of graph conversion to binary file, then this binary file were processed and predicted by neural network.

กิตติกรรมประกาศ

ปริญญานิพนธ์นี้สำเร็จลุล่วงได้เป็นอย่างดี ด้วยคำแนะนำและคำปรึกษาจาก ดร.ยุทธนา คิดใจเดียว เป็นอย่างมากซึ่งเป็นอาจารย์ที่ปรึกษา ท่านกรุณาให้ความรู้ คำปรึกษา แนวคิดและข้อเสนอแนะอันเป็นประโยชน์อย่างมากต่อปริญญานิพนธ์ และอาจารย์ทุกท่านที่ได้ให้แนวคิดในการแก้ปัญหา และช่วยให้คำปรึกษาในการทำโครงงานนี้ ผู้จัดทำรู้สึกซาบซึ้งในความอนุเคราะห์จากท่านและกราบขอบพระคุณเป็นอย่างสูง

สุดท้ายนี้ขอขอบพระคุณภาควิชาอิเล็กทรอนิกส์ คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ที่ได้ให้การสนับสนุนในการทำโครงงานครั้งนี้

คุณค่าและประโยชน์ที่ได้รับในการทำงานครั้งนี้ ผู้จัดทำขอมอบแด่ผู้มีพระคุณทุกๆ ท่าน หากปริญญานิพนธ์ฉบับนี้มีข้อผิดพลาดประการใดขออภัยมา ณ โอกาสนี้ด้วย

อติเทพ เอื้อถาวรสุข

(นายอติเทพ เอื้อถาวรสุข)

อดิศักดิ์ ก่อเกษมพร

(นายอดิศักดิ์ ก่อเกษมพร)

สารบัญ

	หน้า
บทคัดย่อ	**I**
Abstract	**II**
กิตติกรรมประกาศ	**III**
สารบัญ	**IV**
สารบัญภาพ	
สารบัญตาราง	**IX**
บทที่ 1 บทนำ	**1**
1.1 วัตถุประสงค์ของโครงงาน	1
1.2 ขอบเขตของโครงงาน	1
1.3 ประโยชน์ที่ได้รับ	2
1.4 โครงร่างของปริญญานิพนธ์	2
บทที่ 2 การวิเคราะห์อนุกรมเวลา (Time Series Analysis)	**4**
2.1 ส่วนประกอบของอนุกรมเวลา	4
2.1.1 แนวโน้ม	4
2.1.2 การแปรผันตามฤดูกาล	5
2.1.3 การแปรผันตามวัฏจักร	5
2.1.4 การแปรผันเนื่องจากเหตุการณ์ผิดปกติ	5
2.2 รูปแบบของข้อมูลอนุกรมเวลา	5
2.2.1 รูปแบบการบวก	6
2.2.2 รูปแบบการคูณ	6
2.3 การวิเคราะห์อนุกรมเวลา	6
2.4 วิธีการแยกส่วนประกอบ	7
2.4.1 การคำนวณหาค่าเฉลี่ยเคลื่อนที่	7
2.4.2 การคำนวณหาแนวโน้ม (T) การแปรผันตามวัฏจักร (C)	8
2.4.3 การคำนวณหาการแปรผันตามฤดูกาล (S) และการแปรผันเนื่องจากเหตุการณ์ผิดปกติ (I)	8

2.5 การวิเคราะห์ความถดถอยอย่างง่าย 8
2.6 ข้อจำกัดของวิธีแยกส่วนประกอบ 10
2.7 การคำนวณหาค่าคลาดเคลื่อนในการพยากรณ์ 10
2.7 สรุป 11

บทที่ 3 นิวรอลเน็ตเวิร์กและวรรณกรรมที่เกี่ยวข้อง 12
3.1 ทฤษฎีของนิวรอลเน็ตเวิร์ก 12
3.2 ประเภทของนิวรอลเน็ตเวิร์ก 13
3.2.1 การจัดประเภทโครงสร้าง 14
3.2.1.1 ฟีดฟอร์ดเวิร์ดเน็ตเวิร์ก 14
3.2.1.2 รีเคอร์เรนท์เน็ตเวิร์ก 15
3.2.2 การจัดประเภทอัลกอริทึมการเรียนรู้ 15
3.2.2.1 ซุปเปอร์ไวส์เลิร์นนิ่ง 16
3.2.2.2 อันซุปเปอร์ไวส์เลิร์นนิ่ง 16
3.2.2.3 ริอินฟอร์ซเม้นท์เลิร์นนิ่ง 17
3.3 แอกติเวชั่นฟังก์ชั่น 17
3.4 การประยุกต์ใช้นิวรอนเน็ตเวิร์ก 18
3.5 Weightless Artificial Neural Network 19

บทที่ 4 การแปลงกราฟไปเป็นไบนารีไฟล์ 21
4.1 การประมวลผลภาพเชิงตัวเลข 21
4.1.1 การแทนภาพด้วยข้อมูลแบบดิจิตอล 21
4.1.2 ลักษณะการจัดเก็บข้อมูลภาพแบบดิจิตอล 22
4.2 การสร้างไบนารี 22

บทที่ 5 หลักการพยากรณ์ข้อมูล 26
5.1 การพยากรณ์โดยข้อมูลผ่านการNormalizeก่อนแล้วหาแนวโน้ม 26
โดยเอาท์พุทเป็นแบบไม่ระบุตำแหน่งแน่นอน
5.1.1 กระบวนการฝึกสอน Neuron 26
5.1.1.1 การฝึกสอนแบบทั่วไป 27

5.1.1.2 การฝึกสอนแบบ Optimize 29
5.1.2 การพยากรณ์ข้อมูลของ Neuron 29
5.2 การพยากรณ์โดยข้อมูลถูกนำมาหาแนวโน้มก่อนผ่านการNormalize 31
โดยเอาท์พุทเป็นแบบไม่ระบุตำแหน่งแน่นอน
5.2.1 การฝึกสอนแบบ Optimize 31
5.2.2 การพยากรณ์ข้อมูลของ Neuron 31
5.3 การพยากรณ์โดยข้อมูลถูกนำมาหาแนวโน้มก่อน แล้วไม่ผ่านการNormalize 32
โดยเอาท์พุทเป็นแบบไม่ระบุตำแหน่งแน่นอน
5.4 การพยากรณ์โดยข้อมูลถูกนำมาหาแนวโน้มก่อนผ่านการNormalize 32
โดยเอาท์พุทของนิวรอนจะให้เอาท์พุทออกมาโดยไม่ผ่านการโหวต
5.4.1 การพยากรณ์ข้อมูลของ Neuron 33
5.5 การพยากรณ์โดยข้อมูลถูกนำมาหาแนวโน้มก่อนผ่านการNormalize โดยเอาท์พุท33
เป็นแบบระบุตำแหน่งแน่นอน และนิวรอนแต่ละตัวจะมาจากนิวรอน 5 ตัวขนานกัน
5.5.1 การพยากรณ์ข้อมูลของ Neuron 33
5.6 การพยากรณ์โดยข้อมูลถูกนำมาหาแนวโน้มก่อนผ่านการNormalize
โดยเอาท์พุทเป็นแบบระบุตำแหน่งแน่นอน แต่มีการ Feedback 34
5.6.1 การพยากรณ์ข้อมูลของ Neuron 34

บทที่ 6 การทดลองและผลการทดลอง 36

6.1 การพยากรณ์โดยข้อมูลผ่านการNormalizeก่อนแล้วหาแนวโน้ม 36
โดยเอาท์พุท เป็นแบบไม่ระบุตำแหน่งแน่นอน
6.2 การพยากรณ์โดยข้อมูลถูกนำมาหาแนวโน้มก่อนผ่านการNormalize 50
โดยเอาท์พุท เป็นแบบไม่ระบุตำแหน่งแน่นอน
6.3 การพยากรณ์โดยข้อมูลถูกนำมาหาแนวโน้มก่อนแต่ไม่ผ่านการ Normalize 52
โดยเอาท์พุทเป็นแบบไม่ระบุตำแหน่งแน่นอน
6.4 การพยากรณ์โดยข้อมูลถูกนำมาหาแนวโน้มก่อนผ่านการNormalize 54
โดยเอาท์พุทของนิวรอนจะให้เอาท์พุทออกมาโดยไม่ผ่านการโหวต
6.5 การพยากรณ์โดยข้อมูลถูกนำมาหาแนวโน้มก่อนผ่านการNormalize โดยเอาท์พุท 56
เป็นแบบระบุตำแหน่งแน่นอน และนิวรอนแต่ละตัวจะมาจากนิวรอน 5 ตัวขนานกัน
6.6 การพยากรณ์โดยข้อมูลถูกนำมาหาแนวโน้มก่อนผ่านการNormalize 58

โดยเอาท์พุท เป็นแบบระบุตำแหน่งแน่นอน แต่มีการ Feedback

บทที่ 7 สรุปและวิเคราะห์ผลการทดลอง 60

7.1 วิเคราะห์และสรุปผลการทดลอง 60

7.1.1 ผลของการพยากรณ์โดยข้อมูลผ่านการNormalizeก่อน แล้วหาแนวโน้ม โดยเอาท์พุท เป็นแบบไม่ระบุตำแหน่งแน่นอน 60

7.1.1.1 ผลของการเปลี่ยนตำแหน่งการจับของจุดของนิวรอนใน windows 60

7.1.1.2 ผลของการเปลี่ยนจำนวนของนิวรอน 60

7.1.1.3 ผลของการเปลี่ยนจำนวนตำแหน่งที่ใช้จับในแต่ละนิวรอน 61

7.1.1.4 ผลของการเปลี่ยนกรอบอ้างอิง 61

7.1.1.5 ผลของการ Nomalize ข้อมูล 61

7.1.2 ผลของการพยากรณ์โดยข้อมูลถูกนำมาหาแนวโน้มก่อนผ่านการ Normalize โดยเอาท์พุทเป็นแบบไม่ระบุตำแหน่งแน่นอน 61

7.1.2.1 ผลของการเปลี่ยนค่าแนวโน้ม 61

7.1.3 ผลของการพยากรณ์โดยข้อมูลถูกนำมาหาแนวโน้มก่อนแต่ไม่ผ่านการNormalize โดยเอาท์พุทเป็นแบบไม่ระบุตำแหน่งแน่นอน 62

7.1.3.1 ผลของการเปลี่ยนค่าแนวโน้ม 62

7.1.4 ผลของการพยากรณ์โดยข้อมูลถูกนำมาหาแนวโน้มก่อนผ่านการ Normalizeโดยเอาท์พุทของนิวรอนจะให้เอาท์พุทออกมาโดยไม่ผ่านการโหวต 62

7.1.4.1 ผลของการเปลี่ยนค่าแนวโน้ม 62

7.1.5 ผลของการพยากรณ์โดยข้อมูลถูกนำมาหาแนวโน้มก่อนผ่านการ Normalize โดยเอาท์พุทเป็นแบบระบุตำแหน่งแน่นอน และนิวรอนแต่ละตัวจะมาจากนิวรอน 5 ตัวขนานกัน 63

7.1.5.1 ผลของการเปลี่ยนค่าแนวโน้ม 63

7.1.6 ผลของการพยากรณ์โดยข้อมูลถูกนำมาหาแนวโน้มก่อนผ่านการ Normalize โดยเอาท์พุทเป็นแบบระบุตำแหน่งแน่นอน แต่มีการ Feedback 63

7.1.6.1 ผลของการเปลี่ยนค่าแนวโน้ม 63

7.2 ประโยชน์ที่ได้รับและการนำไปประยุกต์ใช้ 64
7.3 อุปสรรคในการทำงานและแนวทางแก้ไข 64

ภาคผนวก

หนังสืออ้างอิง

สารบัญภาพ

หน้า

รูปที่ 2.1 แสดงความสัมพันธ์ของ X และ Y ในรูปเส้นตรง 9

รูปที่ 3.1 อะดาไลน์ (Adaline) 14

รูปที่ 3.2 มัลติเลเยอร์เพอร์เซพตรอน (Multilayer Perceptron) 14

รูปที่ 3.3 ฮอปฟิวด์เน็ตเวิร์ก (Hopfield Network) 15

รูปที่ 3.4 จอร์แดนเน็ตเวิร์ก (Jordan Network) 15

รูปที่ 3.5 แผนภาพการทำงานของซุปเปอร์ไวส์เลิร์นนิ่ง 16

รูปที่ 3.6 แผนภาพการทำงานของอันซุปเปอร์ไวส์เลิร์นนิ่ง 16

รูปที่ 3.7 แผนภาพการทำงานของรีอินฟอร์ซเม้นเลิร์นนิ่ง 17

รูปที่ 4.1 แสดงการแปลงรูปเกรย์สเกลไปเป็นภาพไบนารี 23

รูปที่ 4.2 การแปลงกราฟไบนารีโดยการแปลงจุดดำเป็น '1' และจุดขาวเป็น '0' 24

รูปที่ 4.3 การแปลงกราฟไบนารีโดยการทำให้พื้นที่ใต้กราฟป็น '1' 25

รูปที่ 5.1 แสดงข้อมูลผ่านการ Normalize 27

รูปที่ 5.2 ภาพการเลื่อนกรอบของข้อมูล 28

รูปที่ 5.3 ภาพการฝึกสอน Neuron 28

รูปที่ 5.4 ตัวอย่าง Block diagram ส่วนของการเรียนรู้ 28

รูปที่ 5.5 การซ้อนทับกันของกรอบไบนารีในข้อมูลนำมาฝึกสอน 29

รูปที่ 5.6 ตัวอย่าง Block diagram ส่วนของการพยากรณ์ 30

รูปที่ 6.1 กราฟแสดงราคาหุ้นที่พยากรณ์ได้เปรียบเทียบกับราคาจริงเมื่อใช้การจับจุดของ windows แบบฮาร์โมนิก 37

รูปที่ 6.2 กราฟแสดงแนวโน้มของราคาหุ้นที่พยากรณ์ได้เปรียบเทียบกับแนวโน้มของราคาหุ้นจริงเมื่อใช้การจับจุดของ windows แบบฮาร์โมนิก 37

รูปที่ 6.3 กราฟแสดงราคาหุ้นที่พยากรณ์ได้เปรียบเทียบกับราคาจริงเมื่อใช้การจับจุดของ windows แบบrectangular 38

รูปที่ 6.4 กราฟแสดงแนวโน้มของราคาหุ้นที่พยากรณ์ได้เปรียบเทียบกับแนวโน้มของราคาหุ้นจริงเมื่อใช้การจับจุดของ windows แบบrectangular 38

รูปที่ 6.5 กราฟแสดงราคาหุ้นที่พยากรณ์ได้เปรียบเทียบกับราคาจริงเมื่อใช้การจับจุดของ windows แบบoptimize 39

รูปที่ 6.6 กราฟแสดงแนวโน้มของราคาหุ้นที่พยากรณ์ได้เปรียบเทียบกับแนวโน้ม ของราคาหุ้นจริงเมื่อใช้การจับจุดของ windows แบบoptimize 39
รูปที่ 6.7 กราฟแสดงราคาหุ้นที่พยากรณ์ได้เปรียบเทียบกับราคาจริงเมื่อใช้จำนวนนิวรอน50ตัว 40
รูปที่ 6.8 กราฟแสดงแนวโน้มของราคาหุ้นที่พยากรณ์ได้เปรียบเทียบกับแนวโน้ม ของราคาหุ้นจริงเมื่อใช้จำนวนนิวรอน 50 ตัว 41
รูปที่ 6.9 กราฟแสดงราคาหุ้นที่พยากรณ์ได้เปรียบเทียบกับราคาจริงเมื่อใช้จำนวนนิวรอน135ตัว41
รูปที่ 6.10 กราฟแสดงแนวโน้มของราคาหุ้นที่พยากรณ์ได้เปรียบเทียบกับแนวโน้ม ของราคาหุ้นจริงเมื่อใช้จำนวนนิวรอน 135 ตัว 42
รูปที่ 6.11 กราฟแสดงราคาหุ้นที่พยากรณ์ได้เปรียบเทียบกับราคาจริงเมื่อใช้จำนวน ตำแหน่งที่นิวรอนแต่ละตัวจับ 15 จุด 43
รูปที่ 6.12 กราฟแสดงแนวโน้มของราคาหุ้นที่พยากรณ์ได้เปรียบเทียบกับแนวโน้ม ของราคาหุ้นจริงเมื่อใช้จำนวนตำแหน่งที่นิวรอนแต่ละตัวจับเป็น 15 ตัว 43
รูปที่ 6.13 กราฟแสดงราคาหุ้นที่พยากรณ์ได้เปรียบเทียบกับราคาจริงเมื่อใช้จำนวน ตำแหน่งที่นิวรอนแต่ละตัวจับ 25 จุด 44
รูปที่ 6.14 กราฟแสดงแนวโน้มของราคาหุ้นที่พยากรณ์ได้เปรียบเทียบกับแนวโน้ม ของราคาหุ้นจริงเมื่อใช้จำนวนตำแหน่งที่นิวรอนแต่ละตัวจับเป็น 25 ตัว 44
รูปที่ 6.15 กราฟแสดงราคาหุ้นที่พยากรณ์ได้เปรียบเทียบกับราคาจริงเมื่อใช้ขนาด ของกรอบอ้างอิง 15 วัน 45
รูปที่ 6.16 กราฟแสดงแนวโน้มของราคาหุ้นที่พยากรณ์ได้เปรียบเทียบกับแนวโน้ม ของราคาหุ้นจริงเมื่อใช้ขนาดของกรอบอ้างอิง 15 วัน 46
รูปที่ 6.17 กราฟแสดงราคาหุ้นที่พยากรณ์ได้เปรียบเทียบกับราคาจริงเมื่อใช้ขนาด ของกรอบอ้างอิง 20 วัน 46
รูปที่ 6.18 กราฟแสดงแนวโน้มของราคาหุ้นที่พยากรณ์ได้เปรียบเทียบกับแนวโน้ม ของราคาหุ้นจริงเมื่อใช้ขนาดของกรอบอ้างอิง 20 วัน 47
รูปที่ 6.19 กราฟแสดงราคาหุ้นที่พยากรณ์ได้เปรียบเทียบกับราคาจริงเมื่อใช้การ Normalize ข้อมูล 10 วัน 48
รูปที่ 6.20 กราฟแสดงแนวโน้มของราคาหุ้นที่พยากรณ์ได้เปรียบเทียบกับแนวโน้ม ของราคาหุ้นจริงเมื่อใช้การNormalize ข้อมูล 10 วัน 48
รูปที่ 6.21 กราฟแสดงราคาหุ้นที่พยากรณ์ได้เปรียบเทียบกับราคาจริงเมื่อไม่ใช้การ Normalize ข้อมูล 49

รูปที่ 6.22 กราฟแสดงแนวโน้มของราคาหุ้นที่พยากรณ์ได้เปรียบเทียบกับแนวโน้ม ของราคาหุ้นจริงเมื่อไม่ใช้การNormalize ข้อมูล 49

รูปที่ 6.23 กราฟแสดงแนวโน้มของราคาหุ้นที่พยากรณ์ได้เมื่อ delta = 1 เปรียบเทียบกับ แนวโน้มของราคาหุ้นจริงที่ไม่มีการ Normalize ข้อมูล 50

รูปที่ 6.24 กราฟแสดงแนวโน้มของราคาหุ้นที่พยากรณ์ได้เมื่อ delta = 3 เปรียบเทียบกับ แนวโน้มของราคาหุ้นจริงที่ไม่มีการ Normalize ข้อมูล 51

รูปที่ 6.25 กราฟแสดงแนวโน้มของราคาหุ้นที่พยากรณ์ได้เมื่อ delta = 5 เปรียบเทียบกับ แนวโน้มของราคาหุ้นจริงที่ไม่มีการ Normalize ข้อมูล 51

รูปที่ 6.26 กราฟแสดงแนวโน้มของราคาหุ้นที่พยากรณ์ได้เมื่อdelta = 1และไม่มีการNormalize แนวโน้ม เปรียบเทียบกับแนวโน้มของราคาหุ้นจริงที่ไม่มีการ Normalize ข้อมูล 52

รูปที่ 6.27 กราฟแสดงแนวโน้มของราคาหุ้นที่พยากรณ์ได้เมื่อdelta = 3และไม่มีการNormalize แนวโน้ม เปรียบเทียบกับแนวโน้มของราคาหุ้นจริงที่ไม่มีการ Normalize ข้อมูล 53

รูปที่ 6.28 กราฟแสดงแนวโน้มของราคาหุ้นที่พยากรณ์ได้เมื่อdelta = 5และไม่มีการNormalize แนวโน้ม เปรียบเทียบกับแนวโน้มของราคาหุ้นจริงที่ไม่มีการ Normalize ข้อมูล 53

รูปที่ 6.29 กราฟแสดงแนวโน้มของราคาหุ้นที่พยากรณ์ได้เมื่อ delta = 1 โดยเอาท์พุทไม่มีการ โหวตเปรียบเทียบกับแนวโน้มของราคาหุ้นจริงที่ไม่มีการ Normalize ข้อมูล 54

รูปที่ 6.30 กราฟแสดงแนวโน้มของราคาหุ้นที่พยากรณ์ได้เมื่อ delta = 3 โดยเอาท์พุทไม่มีการ โหวตเปรียบเทียบกับแนวโน้มของราคาหุ้นจริงที่ไม่มีการ Normalize ข้อมูล 55

รูปที่ 6.31 กราฟแสดงแนวโน้มของราคาหุ้นที่พยากรณ์ได้เมื่อ delta = 5 โดยเอาท์พุทไม่มีการ โหวตเปรียบเทียบกับแนวโน้มของราคาหุ้นจริงที่ไม่มีการ Normalize ข้อมูล 55

รูปที่ 6.32 กราฟแสดงแนวโน้มของราคาหุ้นที่พยากรณ์ได้เมื่อ delta = 1 โดยนิวรอนแต่ละตัว จะมานิวรอน 5 ตัวขนานกันเปรียบเทียบกับแนวโน้มของราคาหุ้นจริงที่ไม่มีการ Normalize ข้อมูล 56

รูปที่ 6.33 กราฟแสดงแนวโน้มของราคาหุ้นที่พยากรณ์ได้เมื่อ delta = 3 โดยนิวรอนแต่ละตัว จะมานิวรอน 5 ตัวขนานกันเปรียบเทียบกับแนวโน้มของราคาหุ้นจริงที่ไม่มีการ Normalize ข้อมูล 57

รูปที่ 6.34 กราฟแสดงแนวโน้มของราคาหุ้นที่พยากรณ์ได้เมื่อ delta = 5 โดยนิวรอนแต่ละตัว จะมานิวรอน 5 ตัวขนานกันเปรียบเทียบกับแนวโน้มของราคาหุ้นจริงที่ไม่มีการ Normalize ข้อมูล 57

รูปที่ 6.35 กราฟแสดงแนวโน้มของราคาหุ้นที่พยากรณ์ได้เมื่อ delta = 1 และมีการFeedback เปรียบเทียบกับแนวโน้มของราคาหุ้นจริงที่ไม่มีการ Normalize ข้อมูล 58

รูปที่ 6.36 กราฟแสดงแนวโน้มของราคาหุ้นที่พยากรณ์ได้เมื่อ delta = 3 และมีการFeedback เปรียบเทียบกับแนวโน้มของราคาหุ้นจริงที่ไม่มีการ Normalize ข้อมูล 59

รูปที่ 6.37 กราฟแสดงแนวโน้มของราคาหุ้นที่พยากรณ์ได้เมื่อ delta = 5 และมีการFeedback เปรียบเทียบกับแนวโน้มของราคาหุ้นจริงที่ไม่มีการ Normalize ข้อมูล 59

สารบัญตาราง

หน้า

ตารางที่ 3.1 แอกติเวชั่นฟังก์ชั่น (Activation function) 18

บทที่ 1
บทนำ

ในปัจจุบันข้อมูลข่าวสารถือว่าเป็นสิ่งสำคัญในทุกๆ ด้าน ทั้งในด้านเศรษฐกิจ การเมือง และด้านอื่นๆ การได้รับข้อมูลข่าวสารนั้นก็สามารถรับรู้ได้จากสื่อต่างๆ อาทิเช่น สิ่งพิมพ์ โทรทัศน์ วิทยุ เป็นต้น ในด้านของนักลงทุนนั้นซึ่งต้องการข้อมูลข่าวสารที่เกี่ยวกับการลงทุน ไม่ว่าจะเป็น ประวัติของบริษัท กิจการที่ดำเนินอยู่ ผลกำไรที่บริษัทจะได้รับ ต่างๆ อีกมากมาย ซึ่งปัจจัยเหล่านี้มีผลต่อนักลงทุนเป็นอย่างมากในการที่จะเข้าไปซื้อหุ้นของบริษัทนั้น เพื่อจุดประสงค์ในการหากำไรจากเงินปันผล หรือ การค้าหุ้น

ดังนั้นการที่นักลงทุนสามารถคาดการเกี่ยวกับราคาหุ้นในอนาคตได้จึงเป็นสิ่งจำเป็นและสำคัญเป็นอย่างมากต่อการลงทุน เพื่อที่จะให้ได้ผลกำไรที่มากที่สุด เพราะฉะนั้นการพยากรณ์ราคาหุ้นจึงเป็นสิ่งจำเป็นต่อนักลงทุนเป็นอย่างมาก ซึ่งในโครงงานนี้จึงได้จัดทำการพยากรณ์ราคาหุ้นขึ้นมาโดยใช้ นิวรอลเน็ตเวิร์ก ที่มีวิธีการคำนวณของคอมพิวเตอร์ได้ใกล้เคียงกับการตัดสินใจของสมองมนุษย์มากที่สุด

1.1 วัตถุประสงค์ของโครงงาน

1. เพื่อศึกษาโครงสร้างและการทำงานของนิวรอลเน็ตเวิร์ก
2. เพื่อประยุกต์นิวรอลเน็ตเวิร์ก ในการพยากรณ์หุ้น
3. ศึกษาและออกแบบอัลกอริทึม ของนิวรอลเน็ตเวิร์กแบบต่างๆ ให้เหมาะสมกับการพยากรณ์หุ้น เพื่อให้มีประสิทธิภาพมากที่สุด
4. เพื่อเป็นพื้นฐานในการนำนิวรอลเน็ตเวิร์กไปประยุกต์ใช้ในด้านต่างๆ ที่มีลักษณะใกล้เคียง
5. เพื่อเป็นประโยชน์แก่นักลงทุนในการใช้เป็นเครื่องมือในการเลือกซื้อหุ้น

1.2 ขอบเขตของโครงงาน

1. ได้นำนิวรอลเน็ตเวิร์กแบบ Weightless Artificial Neural Network มาประยุกต์ในการพยากรณ์ราคาหุ้น เพื่อที่จะพยากรณ์ให้ได้ใกล้เคียงกับความเป็นจริงมากที่สุด

2. ศึกษาพารามิเตอร์ต่างๆ ของหุ้นและวิธีการต่างๆ เพื่อหาพารามิเตอร์ที่เหมาะสมและวิธีการที่เหมาะสม
3. ทำการปรับสอนนิวรอลเน็ตเวิร์กให้ได้เรียนรู้จากข้อมูลในอดีต เพื่อที่จะสามารถพยากรณ์ค่าในอนาคตได้
4. นำโปรแกรมที่เขียนได้มาทำการทดสอบโดยในการทดสอบจะใช้หุ้นของ บริษัท ชิน คอร์ปอเรชั่นส์ จำกัด (มหาชน) (SHIN)
5. ทำการเปรียบเทียบข้อมูลที่ได้จากการพยากรณ์กับข้อมูลจริง แล้วหาข้อสรุปเกี่ยวกับโปรแกรม

1.3 ประโยชน์ที่ได้รับ

จากการที่ได้ศึกษาในโครงงานนี้ทำให้ได้รับความรู้เกี่ยวกับ นิวรอลเน็ตเวิร์ก ข้อมูลหุ้น และโปรแกรมภาษาจาวา เป็นอย่างดีพอสมควร ซึ่งทำให้สามารถนำนิวรอลเน็ตเวิร์กมาประยุกต์ให้ประโยชน์ในด้านต่างๆ ได้ และสามารถนำเอาข้อมูลที่ได้จากการวิเคราะห์ไปใช้งานในด้านที่ต้องการ ที่ช่วยให้ความสะดวกในชีวิตประจำวัน สามารถนำโปรแกรมที่ได้จากโครงงานไปเป็นตัวช่วยในการซื้อขายหุ้นได้ เพื่อให้ได้ผลประโยชน์มากที่สุด

1.4 โครงร่างของปริญญานิพนธ์

บทที่ 1 บทนำ : เนื้อหาในบทนี้กล่าวถึงวัตถุประสงค์ของโครงงาน และขอบเขตของโครงงาน ประโยชน์ที่ได้รับ

บทที่ 2 การวิเคราะห์อนุกรมเวลา : เนื้อหาในบทนี้กล่าวถึงส่วนประกอบของอนุกรมเวลา รูปแบบของข้อมูลอนุกรมเวลา การวิเคราะห์อนุกรมเวลา วิธีการแยกส่วนประกอบ การวิเคราะห์ความถดถอยอย่างง่าย (Simple Regression) ข้อจำกัดของวิธีแยกส่วนประกอบ การคำนวณหาค่าคลาดเคลื่อนในการพยากรณ์ และสรุป

บทที่ 3 นิวรอลเน็ตเวิร์กและวรรณกรรมที่เกี่ยวข้อง : เนื้อหาในบทนี้กล่าวถึง ทฤษฎีของนิวรอลเน็ตเวิร์ก ประเภทของนิวรอลเน็ตเวิร์ก แอกติเวชั่นฟังก์ชั่น การประยุกต์ใช้นิวรอลเน็ตเวิร์ก และ Weightless Artificial Neural Network

บทที่ 4 การแปลงกราฟไปเป็นไบนารีไฟล์ : เนื้อหาในบทนี้กล่าวถึง การประมวลผลภาพเชิงตัวเลข การสร้างภาพไบนารี

บทที่ 5 หลักการพยากรณ์ข้อมูล : เนื้อหาในบทนี้กล่าวถึง รูปแบบของการพยากรณ์ในแบบต่างๆ ซึ่งแต่ละแบบก็จะประกอบไปด้วย กระบวนการฝึกสอนนิวรอน การพยากรณ์ข้อมูลของนิวรอน

บทที่ 6 การทดลองและผลการทดลอง : เนื้อหาในบทนี้กล่าวถึง ผลการทดลองของรูปแบบต่างๆ และ การเปรียบผลที่ได้จากการพยากรณ์กับค่าจริง เมื่อเราได้ทำการเปลี่ยน ตัวแปรต่างๆ

บทที่ 7 บทสรุป : เนื้อหาในบทที่กล่าวถึง การวิเคราะห์และสรุปผลการทดลอง ประโยชน์ที่ได้รับและการนำไปประยุกต์ใช้งานจากโครงงาน อุปสรรคในการทำงานและแนวทางในการแก้ไข

บทที่ 2
การวิเคราะห์อนุกรมเวลา (Time Series Analysis)

อนุกรมเวลา (Time Series) คือ กลุ่มของข้อมูลที่เก็บรวบรวมมาตามลำดับเวลาที่ได้เกิดขึ้นในช่างเวลาใดเวลาหนึ่ง โดยปกติระยะห่างของการเก็บมักจะเท่ากัน ข้อมูลอนุกรมเวลาเป็นค่าที่แสดงการเปลี่ยนแปลงของตัวแปรตัวใดตัวหนึ่งตามเวลา โดยที่หน่วยของเวลาอาจเป็นปี เดือน สัปดาห์ วัน หรือ ชั่วโมง เช่น ปริมาณการใช้ไฟฟ้าของประเทศไทยในแต่ละปี ราคาหุ้นของบริษัทแห่งหนึ่งในแต่ละวัน หรือ อุณหภูมิของกรุงเทพฯ ในแต่ละชั่วโมง เป็นต้น ประโยชน์ของอนุกรมเวลา คือ ความสามารถในการพยากรณ์ข้อมูลในอนาคต ซึ่งได้จากการวิเคราะห์ข้อมูลที่รวบรวมไว้ ความสามารถนี้เป็นพื้นฐานการวางแผนการดำเนินงานในอนาคตอย่างมีประสิทธิภาพ การศึกษาลักษณะ คุณสมบัติ หรือการเคลื่อนไหวของอนุกรมเวลานั้น จะต้องทำการวิเคราะห์ว่าอนุกรมเวลาชุดหนึ่งนั้น มีส่วนประกอบชนิดใดปนอยู่บ้าง และส่วนประกอบเหล่านั้นรวมกันอยู่ในลักษณะเช่นไร เนื้อหาในบทนี้กล่าวถึง ส่วนประกอบของอนุกรมเวลา รูปแบบของข้อมูลอนุกรมเวลา การวิเคราะห์อนุกรมเวลา วิธีการแยกส่วนประกอบ ข้อจำกัดของวิธีการแยกส่วนประกอบ และการคำนวณหาค่าคลาดเคลื่อนในการพยากรณ์

2.1 ส่วนประกอบของอนุกรมเวลา (Components of Time Series)

ข้อมูลที่ได้เก็บรวบรวมมาตามลำดับเวลาที่เกิดขึ้น มักจะมีความแปรผันอันเกิดจากปัจจัยหรือส่วนประกอบต่างๆ ส่วนประกอบของอนุกรมเวลาสามารถแยกออกได้เป็น 4 ค่าคือ

2.1.1 แนวโน้ม (Long-Term Trends : T)

แนวโน้ม คือการเคลื่อนไหวหรือเปลี่ยนแปลงซึ่งอาจเป็นการเจริญเติบโตหรือการถดถอยของข้อมูลในระยะยาว การเคลื่อนไหวนี้อาจเป็นไปอย่างรวดเร็วหรือช้าๆ ก็ได้ แต่ค่อนข้างจะมีแบบแผนในแต่ละช่วงเวลา ค่าแนวโน้มนี้ปกติแสดงถึงทิศทางที่อนุกรมเวลาชุดนั้นๆ พุ่งไปสู่ค่าแนวโน้มนี้อาจมีลักษณะเป็นเส้นตรง เส้นโค้ง หรือลักษณะอื่นๆ ซึ่งแนวโน้มนี้เป็นส่วนประกอบที่พบในเกือบทุกอนุกรมเวลา

2.1.2 การแปรผันตามฤดูกาล (Seasonal Variations :S)

การแปรผันตามฤดูกาล คือการเคลื่อนไหวหรือเปลี่ยนแปลงของข้อมูลที่ขึ้นกับฤดูกาลอิทธิพลของฤดูกาลมีผลทำให้ข้อมูลผิดไปจากเวลาปกติ โดยจะมีการเคลื่อนไหวขึ้นลงซ้ำๆ กันภายในช่วงเวลาเดียวกัน เช่น ความต้องการพลังงานไฟฟ้ามักจะสูงขึ้นมากในฤดูร้อน และลดน้อยลงในฤดูหนาว หรือ อุณหภูมิในช่วงฤดูร้อนจะสูงกว่าในฤดูหนาว

2.1.3 การแปรผันตามวัฏจักร (Cyclical Variation : C)

การแปรผันตามวัฏจักร คือการเคลื่อนไหวหรือเปลี่ยนแปลงของข้อมูลที่มีลักษณะขึ้นๆ ลงๆ คล้ายลูกคลื่น การเปลี่ยนแปลงชนิดนี้มีความคล้ายคลึงกับการแปรผันตามฤดูกาล แต่มีช่วงเวลาไม่ค่อยคงที่ บางช่วงอาจมีระยะเวลายาวนาน บางช่วงอาจมีระยะเวลาสั้น การเคลื่อนไหวตามวัฏจักรนี้เป็นส่วนที่วิเคราะห์ค่อนข้างยาก เนื่องจากแต่ละวัฏจักรมักมีขนาด และช่วงเวลาของวงจรแตกต่างกันไป เช่น วัฏจักรทางธุรกิจ วัฏจักรทางเศรษฐกิจ วัฏจักรหุ้น

2.1.4 การแปรผันเนื่องจากเหตุการณ์ผิดปกติ (Irregular Variation :I)

การแปรผันเนื่องจากเหตุการณ์ผิดปกติ คือการเปลี่ยนแปลงที่มีลักษณะขึ้นลงไม่แน่นอน เกิดจากสาเหตุที่ไม่อาจคาดการณ์ไว้ล่วงหน้าได้ ซึ่งอาจทำให้เกิดผลกระทบกับข้อมูลเพียงเล็กน้อยปานกลาง หรือรุนแรง เช่น ภาวะน้ำท่วม การเกิดสงคราม ดอกเบี้ยของธนาคาร ดอกเบี้ยธนาคารต่างประเทศ การเลือกตั้ง การเมือง หรือการนัดหยุดงาน

2.2 รูปแบบของข้อมูลอนุกรมเวลา (Models for Time Series Data)

การขึ้นลงของข้อมูลอนุกรมเวลา (Y) ชุดหนึ่งๆ นั้นมีสาเหตุมาจากส่วนประกอบของอนุกรมทั้ง 4 ประการดังที่กล่าวมาแล้ว ดังนั้นในการวิเคราะห์อนุกรมเวลา จะต้องทราบว่าส่วนประกอบต่างๆ นั้น รวมกันอยู่ในลักษณะเช่นไร

รูปแบบที่นิยมใช้สำหรับการวิเคราะห์อนุกรมเวลาที่ถือว่าเป็นรูปแบบที่ดี คือ รูปแบบที่สร้างขึ้นจากการนำเอาการเปลี่ยนแปลงหรือการเคลื่อนไหวต่างๆ นั้นมาสัมพันธ์กัน แบ่งเป็น 2 ลักษณะคือ

2.2.1 รูปแบบการบวก (Additive Model) : ในรูปแบบนี้ถือว่า ข้อมูลในแต่ละอนุกรมเวลาประกอบด้วยผลบวกของส่วนประกอบทั้ง 4 ประการเหล่านั้น นั่นคือ

$$Y = T + S + C + I$$

2.2.2 รูปแบบการคูณ (Multiplicative Model) : รูปแบบนี้ถือว่าข้อมูลในแต่ละอนุกรมเวลาประกอบด้วยผลคูณของส่วนประกอบ 4 ประการเหล่านั้น นั่นคือ

$$Y = T \times S \times C \times I$$

การเลือกใช้รูปแบบการบวกหรือรูปแบบการคูณในการวิเคราะห์นั้น ขึ้นอยู่กับลักษณะของอนุกรมเวลา โดยทั่วไปรูปแบบการคูณเป็นรูปแบบที่ใช้อธิบายการเคลื่อนไหวของอนุกรมเวลาได้ใกล้เคียงความจริงมากกว่ารูปแบบการบวก

ข้อมูลแต่ละชุดไม่จำเป็นต้องประกอบไปด้วยส่วนประกอบทั้ง 4 ส่วน ข้อมูลบางชุดอาจมีส่วนประกอบ 2 ส่วน คือแนวโน้มและการแปรผันตามฤดูกาล ในขณะที่บางชุดอาจมีส่วนประกอบ 3 ส่วน คือการแปรผันตามวัฏจักร การแปรผันตามฤดูกาล และการแปรผันเนื่องจากเหตุการณ์ผิดปกติ หรือบางชุดอาจมีเพียงแนวโน้มอย่างเดียว และในกรณีที่เป็นข้อมูลรายปีจะไม่มีส่วนประกอบของการแปรผันตามฤดูกาล

2.3 การวิเคราะห์อนุกรมเวลา (Time Series Analysis)

การวิเคราะห์อนุกรมเวลา คือ การศึกษาหารูปแบบการเปลี่ยนแปลงของตัวแปร ที่เปลี่ยนไปตามเวลาจากอดีตถึงปัจจุบัน แล้วนำรูปแบบนั้นมาวิเคราะห์เพื่อพยากรณ์ค่าของตัวแปรนั้นในอนาคต ดังนั้นในการวิเคราะห์ขั้นแรกจึงจำเป็นต้องศึกษาลักษณะหรือคุณสมบัติของอนุกรมเวลาเสียก่อน เพื่อใช้เป็นหลักในการวางแผนหรือกำหนดรูปแบบของความสัมพันธ์ระหว่างปัจจัยต่างๆ ที่เกี่ยวข้องกับอนุกรมเวลา ขั้นต่อไปคือแยกส่วนประกอบที่มีอยู่ในอนุกรมเวลาเพื่อดูว่าอนุกรมเวลานั้นๆ มีส่วนประกอบแต่ละส่วนอยู่มากน้อยเพียงใด แล้วนำส่วนประกอบเหล่านั้นมาพยากรณ์อนุกรมเวลาในอนาคต

2.4 วิธีการแยกส่วนประกอบ (Decomposition Method)

วิธีการแยกส่วนประกอบเป็นวิธีการพยากรณ์ข้อมูลอนุกรมเวลา โดยการแยกเอาส่วนประกอบต่างๆ ในอดีตออกมา เพื่อจะทราบว่าในแต่ละอนุกรมเวลามีส่วนประกอบชนิดใดปนอยู่บ้าง และมีอยู่มากน้อยเพียงใด แล้วนำส่วนประกอบแต่ละส่วนไปพยากรณ์ข้อมูลอนุกรมเวลาในอนาคต โดยถือเสมือนว่าปัจจัยหรือสภาพแวดล้อมที่เกี่ยวข้องกับข้อมูลอนุกรมเวลาในอนาคตเหมือนกับในอดีต ขั้นตอนการแยกส่วนประกอบมีดังนี้

2.4.1 การคำนวณหาค่าเฉลี่ยเคลื่อนที่ (Moving Average : MA)

การคำนวณหาค่าเฉลี่ยเคลื่อนที่เป็นวิธีการที่แยกแนวโน้มและการแปรผันตามวัฏจักรออกจากการแปรผันตามฤดูกาลและการแปรผันเนื่องจากเหตุการณ์ผิดปกติ ดังนั้นค่าเฉลี่ยเคลื่อนที่จึงประกอบไปด้วยแนวโน้มและการแปรผันตามวัฏจักร หรือ MA = T × C นั่นเอง และเมื่อนำค่า MA ที่ได้ไปหารกับข้อมูลอนุกรมเวลา (Y) จะได้การแปรผันตามฤดูกาลและการแปรผันเนื่องจากเหตุการณ์ผิดปกติ หรือ Y / MA = S × I

การหาค่าเฉลี่ยเคลื่อนที่ทำได้โดยจัดข้อมูลเป็นกลุ่มๆ แล้วหาค่าเฉลี่ยของข้อมูลแต่ละกลุ่ม เมื่อหาค่าเฉลี่ยของข้อมูลกลุ่มแรกได้แล้ว ให้ตัดข้อมูลตัวแรกออกไปแล้วเอาข้อมูลตัวใหม่ซึ่งอยู่ถัดจากข้อมูลตัวสุดท้ายมาแทนที่แล้วเฉลี่ย จะได้ค่าเฉลี่ยของกลุ่มที่สอง ทำเช่นนี้ไปเรื่อยๆ จนครบถึงข้อมูลตัวสุดท้ายโดยที่จำนวนข้อมูลในแต่ละกลุ่มต้องเท่ากันอาจเขียนเป็นสมการได้ดังนี้

กำหนดให้

k คือ จำนวนระยะที่ใช้ในการเฉลี่ยแต่ละคราว

$\overline{Y_t}$ คือ ค่าเฉลี่ยเคลื่อนที่ ณ.ตำแหน่งที่ t

$$\overline{Y_t} = \overline{Y_{t+1}} + [\textit{ข้อมูลตัวถัดจากชุดที่(t-1)} - \textit{ข้อมูลตัวแรกของชุดที่(t-1)}]/k$$

เนื่องจากค่าเฉลี่ยที่ได้จะเป็นข้อมูลตรงกึ่งกลางของระยะเฉลี่ย ดังนั้นค่าเฉลี่ยเคลื่อนที่จะมีค่าช่วงต้นช่วงปลายบางช่วงขาดหายไป ซึ่งถ้าข้อมูลที่เก็บรวบรวมมามีจำนวนมาก (อย่างน้อย 2L โดยที่ L = จำนวนฤดูกาลใน 1 ปี) ค่าเฉลี่ยที่หายไปบางส่วนคงไม่ก่อให้เกิดปัญหาอะไรแต่ถ้าข้อมูลที่เก็บรวบรวมมามีจำนวนน้อย (น้อยกว่า 2L โดยที่ L = จำนวนฤดูกาลใน 1 ปี) อาจทำให้เกิดการผิดพลาดในการวิเคราะห์ได้

2.4.2 การคำนวณหาแนวโน้ม(T) และการแปรผันตามวัฏจักร (C)

ขั้นตอนแรกในการคำนวณ คือ แยก T ออกจากค่าเฉลี่ยเคลื่อนที่ วิธีการคำนวณหา T มีหลายวิธี เช่น วิธีประมาณด้วยสายตา วิธีเฉลี่ยทีละครึ่ง วิธีกำลังสองน้อยที่สุด เป็นต้น ซึ่งจะต้องพิจารณาถึงลักษณะของความสัมพันธ์ของ T กับเวลาที่เปลี่ยนแปลง

เมื่อคำนวณหา T ได้แล้ว สามารถคำนวณหา C ได้โดยนำ T ที่ได้นั้นไปหารกับค่าเฉลี่ยเคลื่อนที่ ตามสมการต่อไปนี้

$$MA / T = C$$

2.4.3 การคำนวณหาการแปรผันตามฤดูกาล (S) และการแปรผันเนื่องจากเหตุการณ์ผิดปกติ (I)

ขั้นตอนแรก คือ แยก S ออกจาก I โดยนำค่า $S \times I$ ของแต่ละฤดูกาลมาทำการหาค่าเฉลี่ย โดยตัดค่า $S \times I$ ที่มีค่าสูงสุดและต่ำสุดออก เนื่องจากเป็นค่าที่เกิดจากการแปรผันเนื่องจากเหตุการณ์ผิดปกติ ดังนั้นค่าเฉลี่ยที่ได้คือ ค่า S ฤดูกาลนั้นๆ ขั้นตอนต่อไป คือ การคำนวณ I โดยนำ S ที่ได้ไปหาร $S \times I$ ตามสมการต่อไปนี้

$$(S \times I) / S = I$$

2.5 การวิเคราะห์ความถดถอยอย่างง่าย (Simple Regression)

เป็นการศึกษาถึงความสัมพันธ์ระหว่างตัวแปร 2 ตัว หรือลักษณะที่สนใจศึกษา 2 ลักษณะ โดยที่ต้องทราบค่าของตัวแปรตัวหนึ่งหรือต้องกำหนดค่าของตัวแปรตัวหนึ่งไว้ล่วงหน้า เช่น ถ้าศึกษาถึงความสัมพันธ์ระหว่างรายจ่ายกับรายได้ ยอดขายกับค่าโฆษณา เป็นต้น ซึ่งจะต้องทราบหรือกำหนดรายได้ และค่าโฆษณาไว้ล่วงหน้า โดยจะเรียก รายได้ และค่าโฆษณา ซึ่งเป็นตัวแปรที่ต้องกำหนดค่าไว้ล่วงหน้าว่า ตัวแปรอิสระ (Independent Variable) และมักจะใช้สัญลักษณ์ X ส่วนยอดขายกับรายจ่ายจะเรียกว่า ตัวแปรตาม (Dependent Variable) และใช้สัญลักษณ์ Y ซึ่งหมายถึงยอดขายเป็นตัวแปรที่ขึ้นอยู่กับค่าโฆษณา และรายจ่ายเป็นตัวแปรที่ขึ้นกับรายได้

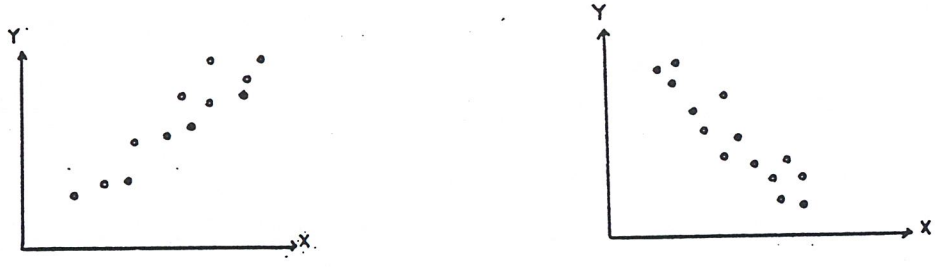
<u>วัตถุประสงค์ของการวิเคราะห์ความถดถอย</u>

การวิเคราะห์ความสัมพันธ์ของตัวแปรต่างๆ มีวัตถุประสงค์ดังนี้

1. เพื่อศึกษาความสัมพันธ์ระหว่างตัวแปรว่ามีความสัมพันธ์กันมากน้อยเพียงใด ถ้า X และ Y มีความสัมพันธ์กันมาก แสดงว่า ถ้า X มีค่าเปลี่ยนแปลงไปจะมีผลกระทบต่อค่าของ Y เป็นอย่างมาก

2. ใช้ความสัมพันธ์ที่วิเคราะห์ได้มาประมาณค่าหรือพยากรณ์ค่า Y ในอนาคต เมื่อกำหนดค่า X

สำหรับการหารูปแบบความสัมพันธ์ระหว่างตัวแปร Y และ X นั้นในขั้นแรกจะนำเอาข้อมูลของตัวแปรทั้งสองมาเขียนกราฟแสดงความสัมพันธ์ ซึ่งจะเรียกกราฟนี้ว่า แผนภาพการกระจาย (Scatter Diagram) ผู้วิเคราะห์จะต้องพิจารณาจากแผนภาพการกระจายว่าความสัมพันธ์ของตัวแปรทั้งสองจะอยู่ในรูปแบบใด เช่น เส้นตรง พาราโบลา เส้นโค้ง อื่นๆ เป็นต้น โดยที่จะต้องสามารถเขียนความสัมพันธ์ให้อยู่ในรูปแบบทางคณิตศาสตร์ได้ ตังอย่างแผนภาพการกระจายของตัวแปร X และ Y แบบเส้นตรงได้แสดงไว้ดังรูป 2.1



ก. ความสัมพันธ์อยู่ในรูปเส้นตรงและเป็นบวก ข. ความสัมพันธ์อยู่ในรูปเส้นตรงและเป็นลบ

รูปที่ 2.1 แสดงความสัมพันธ์ของ X และ Y ในรูปเส้นตรง

ในการวิเคราะห์การถดถอยอย่างง่าย ผู้วิจัยจะได้สมการถดถอยหรือสมการพยากรณ์เชิงเส้นตรงดังนี้

$$Y_I = A + BX_I$$

เมื่อ Y_I คือ คะแนนพยากรณ์ของตัวแปรตาม

A คือ ค่าคงที่

B คือ สัมประสิทธิ์การถดถอย

X_I คือ คะแนนของตัวแปรอิสระ

$$B = \frac{\sum(Y_I - \overline{Y})(X_I - \overline{X})}{\sum(X_I - \overline{X})^2}$$

$$A = \overline{Y} - B\overline{X}$$

A จะมีค่าคงที่เสมอไม่ว่า X จะเปลี่ยนค่าเช่นไรก็ตาม B เป็นค่าที่ชี้ถึงว่า เมื่อตัวแปรอิสระเปลี่ยนแปลงไป 1 หน่วย จะทำให้ตัวแปรตามเปลี่ยนแปลงไป B หน่วย

วิธีที่ง่ายในการสร้างเส้นตรงที่เหมาะสมที่สุดสำหรับการพยากรณ์ ทำได้โดยใช้ค่า X ที่สูงสุดและต่ำสุด กล่าวคือคำนวณหาว่าค่าทั้งสองนั้นพยากรณ์ Y ได้เท่าไร เขียนจุดกราฟตรงตำแหน่งที่ Y พยากรณ์ ตัดกับ X แล้วลากเส้นเชื่อมระหว่างจุดทั้งสองก็จะได้เส้นตรงที่เหมาะสมที่สุดสำหรับพยากรณ์ดังต้องการ ซึ่งเป็นเส้นถดถอยของ Y บน X

2.6 ข้อจำกัดของวิธีแยกส่วนประกอบ

การพยากรณ์ข้อมูลอนุกรมเวลาโดยใช้วิธีแยกส่วนประกอบมีข้อจำกัดดังนี้

1. การแยกข้อมูลอนุกรมเวลาออกเป็น 4 ส่วนนั้น ต้องมีจำนวนข้อมูลมากพอสมควร (อย่างน้อย 5L โดยที่ L = จำนวนฤดูกาลใน 1 ปี) จึงจะสามารถแยกส่วนประกอบเหล่านี้ได้

2. การพยากรณ์โดยนำส่วนประกอบแต่ละส่วนมาคูณกัน ตั้งอยู่บนสมมติฐานว่า ลักษณะการเปลี่ยนแปลงของข้อมูลในอนาคตจะเหมือนกับข้อมูลในอดีต นั้นคือแนวโน้มยังคงอยู่ในลักษณะเดิม การแปรผันตามฤดูกาล และการแปรผันตามวัฏจักร จะมีลักษณะเหมือนในอดีต

2.7 การคำนวณหาค่าคลาดเคลื่อนในการพยากรณ์

วัตถุประสงค์หลักในการพยากรณ์ คือ การได้มาซึ่งข้อมูลที่ถูกต้องและแม่นยำ ในทางสถิติสามารถคำนวณหาความคลาดเคลื่อนของค่าที่ได้จากการพยากรณ์ได้หลายวิธี ตัวอย่างของค่าความคลาดเคลื่อนของการพยากรณ์ข้อมูลอนุกรมเวลาที่นิยมใช้มีดังนี้

กำหนด *N* = จำนวนตัวแปร

Taeget = ค่าจริง

Forecast = ค่าพยากรณ์

1. ค่าความคลาดเคลื่อนเฉลี่ยสัมบูรณ์ (Mean Absolute Error : MAE) : สถิตินี้คำนวณได้จากสมการต่อไปนี้

$$MAE = \frac{1}{N}\sum_{i=1}^{N} \left| Taeget_{I} - Forecast_{i} \right|$$

2. ค่าความคลาดเคลื่อนกำลังสองเฉลี่ย (Mean Square Error : MSE) : สถิตินี้คำนวณได้จากสมการต่อไปนี้

$$MSE = \frac{1}{N}\sum_{i=1}^{N} \left(Taeget_{I} - Forecast_{i} \right)^{2}$$

3. ค่าเบี่ยงเบนมาตรฐานของค่าคลาดเคลื่อนกำลังสองเฉลี่ย (Root Mean Square Error : RMSE) : สถิตินี้คำนวณได้จากสมการต่อไปนี้

$$RMSE = \sqrt{\left[\frac{1}{N}\sum_{i=1}^{N} \left| Taeget_{I} - Forecast_{i} \right| \right]}$$

4. ร้อยละของค่าคลาดเคลื่อนเฉลี่ยสัมบูรณ์(Mean Absolute Percentage Error : MAPE) : สถิตนี้คำนวณได้จากสมการต่อไปนี้

$$MAPE = \frac{1}{N}\sum_{i=1}^{N} \left| (Taeget_{I} - Forecast_{I}) / Taeget_{I} \right| \times 100$$

2.8 สรุป

การวิเคราะห์อนุกรมเวลาเป็นเทคโนโลยีอย่างหนึ่งในวิชาสถิติ มีวัตถุประสงค์เพื่อศึกษาลักษณะการเปลี่ยนแปลงของตัวแปรในอดีต แล้วนำมาพยากรณ์ค่าตัวแปรในอนาคต เพื่อประโยชน์ในด้านการวางแผนการดำเนินงานอย่างมีประสิทธิภาพ การวิเคราะห์อนุกรมเวลามีขั้นตอนที่สำคัญ คือ การวิเคราะห์ลักษณะ และการแยกส่วนประกอบที่สำคัญ 4 ค่า ได้แก่ แนวโน้ม การแปรผันตามฤดูกาล การแปรผันตามวัฏจักร และการแปรผันตามเนื่องจากเหตุการณ์ผิดปกติประสิทธิภาพของการพยากรณ์นั้นขึ้นกับความคลาดเคลื่อนระหว่างค่าจริงและค่าพยากรณ์ ซึ่งมีรูปแบบการคำนวณแตกต่างกันไป

บทที่ 3
นิวรอลเน็ตเวิร์กและวรรณกรรมที่เกี่ยวข้อง

3.1 ทฤษฎีของนิวรอลเน็ตเวิร์ก

นิวรอลเน็ตเวิร์ก เป็นความพยายามอีกอย่างของมนุษย์ ในการที่จะลอกเลียนแบบธรรมชาติ โดยทั่วไปแล้ว นิวรอลเน็ตเวิร์กจะประกอบขึ้นจากหน่วยย่อยที่มีหน้าที่การทำงานคล้ายคลึงกับการทำงานของมนุษย์ จากนั้นหน่วยย่อยเหล่านี้จะถูกจัดเชื่อมต่อกันเป็นโครง่ายในลักษณะที่สัมพันธ์กัน กายวิภาคของสมองหรือบางครั้งอาจจะไม่สัมพันธ์กันก็ได้ และโดยการเชื่อมต่อกันในโครงข่ายนี้ ทำให้นิวรอลเน็ตเวิร์คสามารถแสดงพฤติกรรมมากมายที่มีการตอบสนองคล้ายคลึงกับระบบประสาทในสิ่งมีชีวิต ยกตัวอย่างเช่น ความสามารถในการที่จะเรียนรู้จากประสบการณ์ การจัดกลุ่มของข่าวสารจากตัวอย่างที่ให้ หรือจับลักษณะเด่นของอินพุทที่ประกอบจากข้อมูลที่ไม่สมบูรณ์เป็นต้น

การศึกษาและสิ่งประดิษฐ์ทางด้านนิวรอลเน็ตเวิร์ก ได้ถูกนำเสนอออกมาทั้งในรูปแบบของฮาร์ดแวร์และซอฟแวร์ โดยผลิตภัณฑ์ที่เป็นฮาร์ดแวร์มักเรียกว่า Neurocomputer ซึ่งจะทำการจำลองการทำงานของเซลล์ประสาทลงบนวงจรทางไฟฟ้า หรือทางออปติก จากนั้นอาจถูกนำมาใช้ร่วมกับคอมพิวเตอร์ธรรมดาได้ในลักษณะของ Co-processor ซึ่งสามารถเรียกใช้เหมือนกับเป็นโปรแกรมย่อยอันหนึ่งในระบบได้ หรือในอีกแนวทางหนึ่งจะอยู่ในรูปของซอพแวร์โดยสร้างโปรแกรมเพื่อจำลองเซลล์ประสาทและจุดเชื่อมต่อของเซลล์ประสาทเหล่านั้นลงบนคอมพิวเตอร์เพื่อศึกษาพฤติกรรมของมัน

ถ้าพิจารณาความสามารถพิเศษของนิวรอลเน็ตเวิร์ก ดังเช่นความสามารถในการเรียนรู้จากประสบการณ์หรืออีกอย่าง จะเห็นว่าความสามารถเหล่านั้นสามารถที่ได้จากวิชาการทางด้านปัญญาประดิษฐ์ AI (Artificial Intelligence) ซึ่งจะสามารถเรียนรู้และจดจำกระบวนการคล้ายคลึงกับกระบวนการทางสมองของมนุษย์ โดยเฉพาะทางด้านระบบผู้เชี่ยวชาญ (Expert System) ซึ่งประสบความสำเร็จอย่างมากในหลายที่ แต่ก็ประสบปัญหาความยุ่งยากในการสร้างความสามารถสำหรับการเรียนรู้จากประสบการณ์ไม่น้อย จนหลายคนมองว่านิวรอลเน็ตเวิร์ก อาจเข้ามาแทนที่ AI แต่ที่จริงแล้ว ควรนำเทคนิคทั้งสองแบบมาใช้ร่วมกันได้เป็นระบบโดยที่แต่ละตัวทำงานในส่วนที่ตนเองถนัดมากกว่า

ความแตกต่างระหว่างคอมพิวเตอร์ธรรมดากับนิวรอลเน็ตเวิร์กอีกอย่างหนึ่งคือในคอมพิวเตอร์ธรรมดา เมื่อต้องการให้การทำงานอย่างใดสามารถทำได้โดยการสร้างโปรแกรม แต่ในทางตรงข้ามนิวรอลเน็ตเวิร์ก เมื่อต้องการให้การทำงานอย่างใดสามารถทำได้โดยการสอน (Training) โดยการฝึกสอนนี้จะทำให้นิวรอลเน็ตเวิร์ก นำบทเรียนที่ได้เข้าไปเก็บในโครงสร้างของนิวรอนและจากนั้นจะถูกเรียกกลับขึ้นมาใช้เมื่ออยู่ในระหว่างประมวลผลข้อมูล

คุณสมบัติและความสามารถที่เป็นประโยชน์

1. มีความยืดหยุ่นสูงจนสามารถจำลองขบวนการของปัญหาใดๆ ได้
2. มีความสามารถในการจำชุดของคู่ Input-Output ที่มีความซับซ้อนมากจนไม่สามารถจำลองแบบในเชิงความน่าจะเป็นได้
3. มีความสามารถในการปรับตัวเข้ากับสิ่งแวดล้อมได้
4. มีความสามารถในการตอบสนองข้อมูลที่ไม่เคยเห็น
5. ความรู้การกระจายอยู่ทั่วทั้งโครงสร้างของโครงสร้างของโครงข่าย ทำให้ดำเนินการกับสารสนเทศเชิงอรรถาธิบาย (Contextual Information) ได้อย่างเป็นธรรมชาติ

สรุปนิวรอลเน็ตเวิร์ก (Neural Network) คือ ระบบคอมพิวเตอร์ที่เลียนแบบโครงสร้างและการทำงานของสมองมนุษย์ในรูปแบบทางคณิตศาสตร์ คุณลักษณะที่สำคัญของสมองที่ถูกนำมาประยุกต์ใช้เป็นพื้นฐานของนิวรอลเน็ตเวิร์ก คือ สถาปัตยกรรมแบบกระจาย (Distribute) และขนาน (Parallel) ของเซลล์ต่างๆ และความสามารถในการเรียนรู้และในการปรับตัวตามข้อมูลที่ได้รับจากสภาพแวดล้อมภายนอก เนื้อหาในบทนี้กล่าวถึง ประเภทของนิวรอลเน็ตเวิร์ก แอกติเวชั่นฟังก์ชั่น การประยุกต์ใช้นิวรอลเน็ตเวิร์ก นิวรอลเน็ตเวิร์กที่นำมาใช้

3.2 ประเภทของนิวรอลเน็ตเวิร์ก (Neural Network Categories)

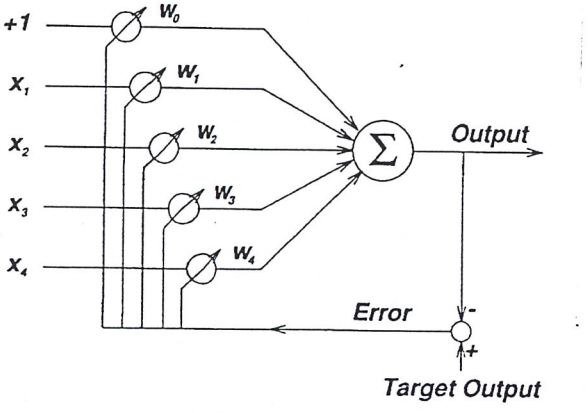
นิวรอลเน็ตเวิร์ก ประกอบด้วยหน่วยประมวลผลที่เรียกว่า นิวรอน (Neural) เชื่อมต่อกันเป็นเน็ตเวิร์ก ซึ่งลักษณะการเชื่อมต่อกันเป็นการกำหนดโครงสร้างของนิวรอลเน็ตเวิร์ก การเชื่อมต่อกันระหว่างนิวรอนจะมีค่าน้ำหนัก (Weight) มาเกี่ยวข้องหรือไม่มีก็ได้ โดยที่มีค่าน้ำหนักนี้สามารถปรับเปลี่ยนได้ตามความเหมาะสม โดยอัลกอริทึมการเรียนรู้ (Learning Algorithm) การจัดประเภทของนิวรอลเน็ตเวิร์กสามารถทำได้โดยพิจารณาจากประเภทโครงสร้าง และประเภทอัลกอริทึมการเรียนรู้ ซึ่งมีรายละเอียดดังนี้

3.2.1 การจัดประเภทโครงสร้าง (Structure Categorisation)

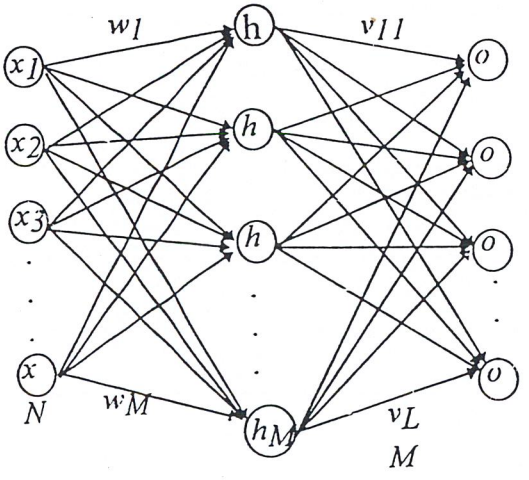
ในการจัดประเภทโครงสร้างของนิวรอลเน็ตเวิร์ก พิจารณาจากทิศทางการเคลื่อนที่ของสัญญาณข้อมูล สามารถแบ่งได้เป็น 2 ชนิด ดังนี้

3.2.1.1 ฟีดฟอร์เวิร์ดเน็ตเวิร์ก (Feedforward Networks)

ในฟีดฟอร์เวิร์ดเน็ตเวิร์ก นิวรอนจะถูกจัดกลุ่มลงในเลเยอร์ (Layer) สัญญาณข้อมูลจะเคลื่อนที่ไปในทิศทางเดียว คือ จากอินพุทเลเยอร์ (Input Layer) ผ่านฮิดเดนเลเยอร์ (Hidden Layer) ไปสู่เอาท์พุทเลเยอร์ (Output Layer) การเชื่อมต่อกันเป็นลักษณะจากเลเยอร์หนึ่งไปยังเลเยอร์ถัดไปโดยไม่มีการเชื่อมต่อกันภายในเลเยอร์เดียวกัน ตัวอย่างเน็ตเวิร์กชนิดนี้แสดงในรูปที่ 3.1 (อะดาไลน์ (Adaline)) และรูปที่ 3.2 (มัลติเลเยอร์เพอร์เซพตรอน (Multilayer Perceptron))



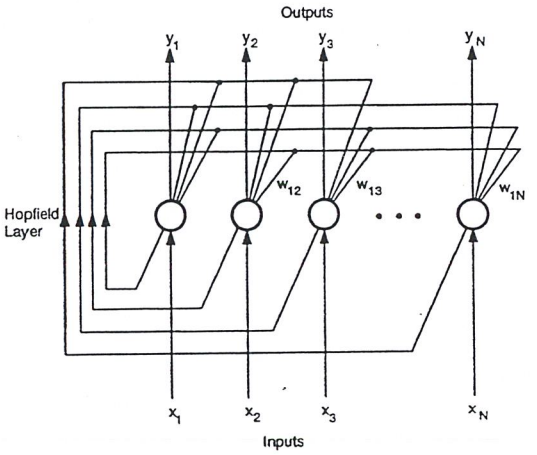
รูปที่ 3.1 อะดาไลน์ (Adaline)



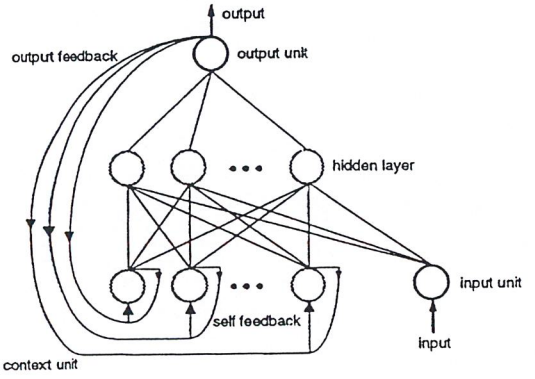
รูปที่ 3.2 มัลติเลเยอร์เพอร์เซพตรอน (Multilayer Perceptron)

3.2.1.2 รีเคอร์เรนท์เน็ตเวิร์ก (Recurrent Network)

ในรีเคอร์เรนท์เน็ตเวิร์ก สัญญาณข้อมูลมีการเคลื่อนที่ไปใน 2 ทิศทาง คือ เคลื่อนที่ไปข้างหน้า และย้อนกลับ โดยที่เอาท์พุทของบางนิวรอนจะมีการเคลื่อนที่ย้อนกลับไปยังตัวมันเอง หรือเคลื่อนที่ย้อนกลับไปยังเลเยอร์ที่มาก่อนหน้า ตัวอย่างของเน็ตเวิร์กชนิดนี้คือ ฮอปฟิวด์เน็ตเวิร์ก (Hopfield Network) ซึ่งแสดงได้ดังรูปที่ 3.3 และ จอร์แดนเน็ตเวิร์ก (Jordan Network) ซึ่งแสดงได้ดังรูปที่ 3.4



รูปที่ 3.3 ฮอปฟิวด์เน็ตเวิร์ก (Hopfield Network)



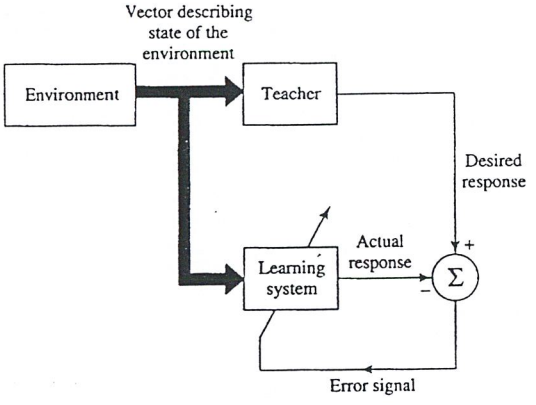
รูปที่ 3.4 จอร์แดนเน็ตเวิร์ก (Jordan Network)

3.2.2 การจัดประเภทอัลกอริทึมการเรียนรู้ (Learning Algorithm Categorisation)

อัลกอริทึมการเรียนรู้ คือ อัลกอริทึมที่ใช้ในการปรับสอนเน็ตเวิร์ก ให้สามารถเรียนรู้ที่จะทำการปรับค่าน้ำหนัก โดยมีวัตถุประสงค์เพื่อให้ได้เอาท์พุทใกล้เคียงหรือเท่ากับเอาท์พุทที่ต้องการ การจัดประเภทอัลกอริทึมการเรียนรู้สามารถแบ่งได้เป็น 3 ชนิดดังนี้

3.2.2.1 ซุปเปอร์ไวส์เลิร์นนิ่ง (supervised Learning)

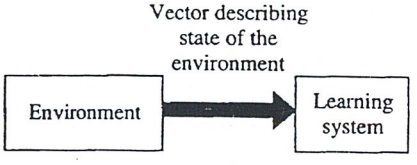
อัลกอริทึมชนิดนี้ต้องมีครูที่ช่วยกำหนดเอาท์พุทเป้าหมาย (Output Target) ให้กับเน็ตเวิร์กดังนั้นชุดข้อมูลในการปรับสอนเน็ตเวิร์ก จะประกอบไปด้วยชุดข้อมูลอินพุท และชุดข้อมูลเอาท์พุทเป้าหมาย เอาท์พุทที่ได้จากเน็ตเวิร์กจะถูกนำมาเปรียบเทียบกับเอาท์พุทเป้าหมาย เพื่อที่จะทำการปรับค่าน้ำหนัก ให้ได้เอาท์พุทที่ใกล้เคียงกับเอาท์พุทเป้าหมาย แผนภาพการทำงานของซุปเปอร์ไวส์เลิร์นนิ่ง แสดงได้ดังรูปที่ 3.5 ตัวอย่างของอัลกอริทึมชนิดนี้คือ แบ็คพรอพาเกชั่นอัลกอริทึม (Black propagation algorithm)



รูปที่ 3.5 แผนภาพการทำงานของซุปเปอร์ไวส์เลิร์นนิ่ง

3.2.2.2 อันซุปเปอร์ไวส์เลิร์นนิ่ง (Unsupervised Learning)

อัลกอริทึมชนิดนี้ไม่ต้องการเอาท์พุทเป้าหมายในการปรับสอนเน็ตเวิร์ก ชุดข้อมูลอินพุทเท่านั้นที่ถูกป้อนเข้าสู่เน็ตเวิร์ก ซึ่งจะมีการปรับค่าน้ำหนักโดยอัตโนมัติเพื่อที่จะทำการจัดกลุ่มอินพุทที่มีลักษณะคล้ายคลึงกันลงในกลุ่มเดียวกัน แผนภาพการทำงานของอันซุปเปอร์ไวส์เลิร์นนิ่งแสดงได้ดังรูปที่ 3.6 ตัวอย่างของเน็ตเวิร์กชนิดนี้คือ ART (Adaptive Resonance Theory) และ โคโฮเนน (Kohonen)

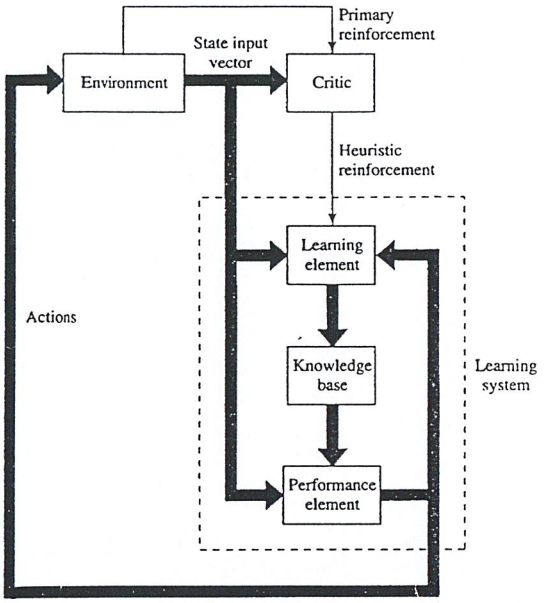


รูปที่ 3.6 แผนภาพการทำงานของอันซุปเปอร์ไวส์เลิร์นนิ่ง

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

3.2.2.3 รีอินฟอร์ซเม้นท์เลิร์นนิ่ง (Reinforcement Learning)

รีอินฟอร์ซเม้นท์เลิร์นนิ่ง คือ ซุปเปอร์ไวส์เลิร์นนิ่งชนิดพิเศษ ซึ่งแทนที่จะกำหนดเอาท์พุทเป้าหมายให้กับเน็ตเวิร์ก แต่อัลกอริทึมชนิดนี้จะใช้ Critic เป็นตัวประเมินค่าให้กับเอาท์พุทที่ได้แทน ตัวอย่างของอัลกอริทึมชนิดนี้คือ Genetic Algorithm แผนภาพการทำงานของรีอินฟอร์ซเม้นท์เลิร์นนิ่ง แสดงได้ดังรูปที่ 3.7



รูปที่ 3.7 แผนภาพการทำงานของรีอินฟอร์ซเม้นท์เลิร์นนิ่ง

3.3 แอกติเวชั่นฟังก์ชั่น (Activation Function)

แอกติเวชั่นฟังก์ชั่นเป็นฟังก์ชั่นที่ใช้ในการคำนวณเอาท์พุทของนิวรอน แอกติเวชั่นฟังก์ชั่นมีหลายชนิด ตัวอย่างของสมการที่นิยมใช้สำหรับแอกติเวชั่นฟังก์ชั่นแสดงได้ดังตารางที่ 3.1

46281

ตารางที่ 3.1 แอกติเวชั่นฟังก์ชั่นที่นิยมใช้

ชนิดของแอกติเวชั้นฟังก์ชั่น	ฟังก์ชั่น
Linear	$f(z) = z$
Sigmoid	$f(z) = 1/(1+exp(-z))$
Hyperbolic Tangent	$f(z) = (1-exp(-2z))/(1+exp(-2z))$
Radial Basis Function	$f(z) = exp(-z^2/\beta^2))$
Threshold	$f(z) = +1, \quad if\ z>z_1$ $= -1, \quad otherwise$

3.4 การประยุกต์ใช้นิวรอลเน็ตเวิร์ก (Neural Network Application)

ในปัจจุบันได้มีการนำนิวรอลเน็ตเวิร์กมาประยุกต์ใช้กับงานหลายด้าน ตัวอย่างกลุ่มงานที่ได้รับความนิยมมีดังนี้

Recognition : การนำนิวรอลเน็ตเวิร์ก มาประยุกต์ใช้กับงานการจดจำได้รับความนิยมมากที่สุดในปัจจุบัน เนื่องจากนิวรอลเน็ตเวิร์กมีความสามารถที่จะจดจำรูปแบบที่มีความซับซ้อนมากได้ดี ตัวอย่างของเน็ตเวิร์กที่เหมาะสมกับงานด้านนี้คือ มัลติเลเยอร์เพอร์เซพตรอน

Clustering : หลักการของงานด้านนี้คือ อินพุทของนิวรอลเน็ตเวิร์กที่มีคุณสมบัติหรือลักษณะที่ใกล้เคียงกันจะถูกจัดให้อยู่ในกลุ่มเดียวกัน เน็ตเวิร์กที่เหมาะสมกับงานด้านนี้ก็คือ อันซุปเปอร์ไวส์นิวรอลเน็ตเวิร์ก ตัวอย่างเช่น ART network และ Kohonen's self-organising feature map

Classification : ในขั้นตอนการปรับสอน จำนวนของคลาส (Class) และคุณสมบัติของแต่ละคลาสจะถูกป้อนเข้าสู่เน็ตเวิร์ก อินพุทของเน็ตเวิร์กจะถูกจัดเข้าคลาสที่เหมาะสม ตัวอย่างของเน็ตเวิร์กที่เหมาะสมกับงานด้านนี้คือ มัลติเลเยอร์เพอร์เซพตรอน และ Linear Vector Quantization (LVQ) เน็ตเวิร์ก

Forecasting : อัลกอริทึมการเรียนรู้ที่ใช้กับงานด้านนี้คือ ซุปเปอร์ไวส์อัลกอริทึม เน็ตเวิร์กจะถูกปรับสอนด้วยชุดข้อมูลในอดีต ซึ่งประกอบด้วยอินพุทและเอาท์พุทเป้าหมาย เน็ตเวิร์กที่เหมาะสมและเป็นที่นิยมใช้กับงานด้านนี้คือ มัลติเลเยอร์เพอร์เซพตรอนที่มีการเรียนรู้แบบแบ็คพรอพาเกชั่น

3.5 Weightless Artificial Neural Network

Weightless Artificial Neural Network จะมีลักษณะโครงสร้างเหมือน Artificial Neural Network ต่างกันตรงที่ ไม่มีค่าน้ำหนักหรือ Weight ซึ่งทำให้กระบวนการในการตัดสินใจนั้นง่ายกว่าและเร็วกว่าแบบ Artificial Neural Network มาก จากข้อได้เปรียบที่กล่าวมาเราจึงนำ Weightless Artificial Neural Network มาประยุกต์ใช้ในการพยากรณ์หุ้น ซึ่งในการใช้งานนี้ การทำงานของโครงข่ายจะเป็นลักษณะของ Single Layer Feed Forward Neural คือมี 1 ชั้นและมีทั้งการป้อนกลับและไม่มีการป้อนกลับ Weightless Artificial Neural Network ประกอบไปด้วยกลุ่มของ discriminators ที่เป็นตัวแทนของค่าความแตกต่างกันของ pattern ในแต่ละชั้นซึ่ง discriminators ทุกตัวในชั้นแรกเราจะ set ค่าให้เป็น 0 ในระหว่างการ Training แต่ละ pattern จะถูกเก็บไว้ใน RAMs ของ discriminator ที่ตรงกันเท่านั้น หลังจากการ Training ในแต่ละ class เสร็จเรียบร้อยแล้ว การรู้จำสมบูรณ์แบบก็จะเกิดขึ้นจากการที่เราทำการป้อน unknown pattern ให้กับ discriminators ทุกตัวแล้วทำการหาคำตอบที่แน่นอน ใน Weightless Artificial Neural Network ถ้าเกิดความกำกวมขึ้นระหว่าง การเก็บ pattern และ set 0 state ตอนเริ่มต้นก่อนที่จะเกิดการ Training ซึ่งจะแก้ปัญหาเหล่านี้ได้โดยการใช้ Probabilistic Logic Node (PLN) แต่ละ Node ยังคงมี binary output แต่เป็นการทำงานที่ระดับ 3 อีกข้อหนึ่ง คือ u-state เป็นการแทนค่าเริ่มต้นหรือเป็น state ที่ไม่มีข้อมูลเก็บไว้ ซึ่งจะก่อให้เกิดผลเสียไปถึง output คือ จะมีโอกาสเกิด 1 และ 0 ได้อย่างละ 50% ใน recalling mode รายละเอียดในลำดับขั้นถัดไป จะเป็นการกล่าวถึงเกี่ยวกับการทำงานของ PLNs

การพัฒนาของ PLNs ที่เป็นที่รู้จักในนามของ Goal Seeking Neural (GNS) ซึ่งจะต่างจาก PLNs คือ จะมี u-state เป็นแบบการแพร่ผ่าน Network แทนที่จะเป็นการบังคับ node เพื่อที่จะได้เอาท์พุทออกมาระหว่าง 0 และ 1 ซึ่ง Multi-addressing ใน GSNs มีจุดประสงค์เพื่อลดจำนวนของ u-state location และเป็นการหลีกเลี่ยงความลังเลระหว่าง retrieval phase เนื่องจาก u-state ที่เข้ามาเป็นจำนวนมาก Version อื่นๆ ของ RAM-based neurons ที่ประกอบด้วยค่าจริงที่ต่อเนื่องที่มีชื่อว่า The multi-value probabilistic RAMs (pRAMs) ซึ่งแทนที่จะมีค่าเพียงค่า binary เก็บไว้ใน look-up table output จะคำนวณได้จากค่าถ่วงน้ำหนักผลรวมของเวกเตอร์ทางด้านอินพุทต่อเนื่อง Version ต่อมาเราเรียกว่า Cascaded Single Weightless Neural Network (CSWNN) โมเดลนี้จะประกอบด้วยกลุ่มของ RAM Neurons ซึ่งแต่ละกลุ่มจะมี feedback ต่ออยู่ภายในตัวของมันเองและเพื่อต่อโดยตรงกับ Network กลุ่มต่อไป (Cascade network) เป็นกลไกเฉพาะเพื่อที่จะทำเกี่ยวกับปัญหาการเรียนรู้ด้านภาษา

เนื่องจากใช้แค่เพียง Location ซึ่งหามาจากการที่เรา trained pattern เป็นการพัฒนาระหว่าง การ training PLNs อาจเกิดความเสียหายจากการ delay เมื่อ training set มีค่าน้อยเกินไป (เช่นมีประมาณ 2-3 ชุด) ความเสียหายอื่นๆ ที่เป็น generalization เกิดขึ้นในระดับชั้น Network เทคนิคของการ Generalization ที่ไม่สนใจอินพุทที่ node level เรียกว่า spreading process ซึ่งเป็นขบวนการที่ต่อจากช่วง learning phase กระบวนการทำงานโดย การกำหนดค่า u-state location จากค่าใกล้เคียงที่สุดใน Hamming distance

ถ้ามีการขัดแย้งระหว่างค่าที่ใกล้เคียงที่สุด (เลือกค่าใดค่าหนึ่งไม่ได้) Location ของ u-state จะไม่มีการเปลี่ยนแปลงค่า generalization ของ network ขึ้นอยู่กับระยะรัศมีของ Spreading regions ที่แต่ละ node ถ้าค่ารัศมีเท่ากับจำนวนทั้งหมดของ binary output หมายความว่า network มีการกระจายเต็มซึ่ง network ที่มีคุณสมบัติดังนี้เราจะเรียกว่า Generalizing RAMs (GRAMs) ขนาดของ Generalization ใน GRAMs จะถูกกำหนดโดยจำนวนของ patterns ในการ training เพราะว่า spreading region จะมีขนาดเล็กลงเมื่อจำนวน pattern มากขึ้น

บทที่4
การแปลงกราฟไปเป็นไบนารีไฟล์

4.1 การประมวลผลภาพเชิงตัวเลข (Digital Image Processing)

การประมวลผลภาพเชิงตัวเลข หมายถึง การนำภาพที่พบทั่วไปมาประมวลด้วยเครื่องคอมพิวเตอร์ โดยนำภาพที่นำมาประมวลผลด้วยเครื่องคอมพิวเตอร์นี้จะถูกแทนที่ด้วยตัวเลขให้อยู่ในรูปของเมตริกซ์ แต่ภาพที่ได้โดยส่วนมากแล้วจะเป็นภาพที่ได้จากตัวรับสัญญาณ ซึ่งอยู่ในรูปของฟังก์ชั่น f(x,y) ที่ต่อเนื่องในระนาบสองมิติ (คือแกน x และแกน y) โดยจะเป็นสัดส่วนกับความสว่างหรือความเข้มของภาพที่ตำแหน่ง (x,y) ซึ่งเรียกว่าระดับสีเทา (Gray Level)

4.1.1 การแทนภาพด้วยข้อมูลแบบดิจิตอล

ภาพข้อมูลแบบดิจิตอล (Digital Image) เป็นภาพที่ถูกแปลงมาจากอนาลอก อยู่ในรูปของตัวเลข โดยภาพอนาลอกถูกแบ่งเป็นพื้นที่สี่เหลี่ยมเล็กๆ ที่เรียกว่า พิกเซล (Pixel) ในแต่ละพิกเซลจะถูกระบุตำแหน่งโดย (x,y) และค่าระดับสีเทาของพิกเซล โดยเราสามารถแปลงภาพเป็นข้อมูลแบบดิจิตอลได้โดยมีขั้นตอนและวิธีการดังนี้

เมื่อเรานำสัญญาณอนาลอกที่ต้องการประมวลมาให้เป็นสัญญาณดิจิตอล จากนั้นทำการควอนไตซ์ (Quantizing) เพื่อที่จะประมวลสัญญาณด้วยระบบคอมพิวเตอร์ ฟังก์ชันที่ได้เรียกว่าการควอนไตเซชั่นระดับสีเทา (Gray Level Quantizing) ก็จะได้ข้อมูลที่เป็นดิจิตอล

สมมุติว่าสัญญาณภาพระนาบ X และ Y เป็นช่วงเท่าๆ กัน เราสามารถจัด f(x,y) ให้อยู่ในรูปของเมตริกซ์ ขนาด N×N ได้ดังสมการที่ 4.1

$$f(x,y) = \begin{matrix} f(0,0) & f(0,1) & f(0,2)\ldots\ldots f(0,N-1) \\ f(0,0) & f(0,1) & f(0,2)\ldots\ldots f(0,N-1) \\ . & . & . \quad . \\ . & . & . \quad . \\ f(N-1,0) & f(N-1,1) & f(N-1,2)\ldots\ldots f(N-1,N-1) \end{matrix} \qquad \ldots\ldots\ldots(4.1)$$

ซึ่งทางขวาของสมการ จะเรียกได้ว่า ภาพดิจิตอล และทุกๆ สมาชิกของเมตริกซ์ จะเรียกว่า พิกเซล จากขบวนการสร้างภาพดิจิตอลข้างต้นจะเห็นได้ว่า เราสามารถทราบขนาดของความละเอียดของภาพ N×N พิกเซล และจำนวนระดับของเกรย์สเกล ในทางปฏิบัติการทำความควอนไตเซชั่นในระบบภาพดิจิตอล จะมีค่าดังสมการ 4.2

$$B = N \times N \times M \quad \text{บิต} \qquad \ldots\ldots\ldots\ldots (4.2)$$

เมื่อ B คือ ขนาดของข้อมูลภาพที่เป็นดิจิตอล

G คือ จำนวนของเกรย์สเกลที่ต้องการใช้ในการเก็บข้อมูลภาพ

M คือ จำนวนบิตที่ใช้ในการแทนข้อมูลภาพ 1 พิกเซล

โดย M สามารถหาได้จาก G มีค่าเท่ากับ 2

4.1.2 ลักษณะการจัดเก็บข้อมูลภาพแบบดิจิตอล

โดยทั่วไปแล้วข้อมูลภาพจะมีค่าความเข้มตั้งแต่ 2 ระดับขึ้นไปแต่ที่ใช้กันมากจะมีระดับความเข้มของจุดภาพเท่ากับ 256 ระดับ ซึ่งจะทำให้ค่าของจุดภาพอยู่ในช่วง (0-255) โดยใช้เนื้อที่ในการเก็บข้อมูลภาพขนาด 1 ไบต์ หรือ 8 บิต สำหรับข้อมูล 1 จุดภาพ ($2^8 = 256$) ในกรณีที่ต้องการภาพที่มีความละเอียดสูงๆ อาจจะต้องการจำนวนบิตสำหรับการเก็บข้อมูลมากกว่า 8 บิต คืออาจจะเป็น 16 หรือ 24 บิต โดยค่าความเข้มของจุดภาพจะเท่ากับ 2^{16} และ 2^{24} โดยจะแยกให้เห็นชัดเจนดังนี้

1. ภาพ 2 ระดับ คือ มีเพียงแค่จุดขาวกับดำเท่านั้น โดยแต่ละจุดภาพเป็นข้อมูลขนาด 1 บิต
2. ภาพ 16 ระดับ คือ ในแต่ละจุดภาพจะมีขนาดของข้อมูล 4 บิต ซึ่งทำให้สามารถแสดงภาพได้ 16 ระดับสี หรือ 16 เกรย์สเกล ขึ้นอยู่กับภาพนั้นเป็นภาพสีหรือขาวดำ
3. ภาพ 256 ระดับ คือ ในแต่ละจุดภาพจะมีขนาดของข้อมูล 8 บิต ซึ่งทำให้สามารถแสดงภาพได้ 256 ระดับสี หรือ 256 เกรย์สเกล ขึ้นอยู่กับภาพนั้นเป็นภาพสีหรือขาวดำ

4.2 การสร้างไบนารี

ในการพยากรณ์ข้อมูลด้วยนิวรอลเน็ตเวิร์กนี้เราต้องอาศัยข้อมูลทางดิจิตอลเพื่อให้ง่ายต่อการวิเคราะห์และเข้าใจของโปรแกรมดังนั้นก่อนที่เราจะทำการวิเคราะห์ข้อมูลใดๆ นั้นเราจึงต้องทำการแปลงข้อมูลจากภาพซึ่งก็คือ กราฟไปเป็นข้อมูลทางดิจิตอลซึ่งก็คือไบนารีไฟล์นั่นเอง นั่นหมาย

ถึงการแปลงข้อมูลที่มีระดับความเข้มหลายระดับ (Multi Level Image) ให้เป็นภาพที่มีระดับความเข้มเพียง 2 ระดับ นั่นคือ 1 จุดภาพที่ได้ 2 ค่าเท่านั้นคือ 0 กับ 1 โดยจุดภาพที่แทนด้วย 1 จะหมายถึงจุดภาพที่มีระดับสีดำ ส่วนจุดภาพที่แทนด้วย 0 จะหมายถึงจุดภาพที่มีสีขาว

ในการสร้างภาพไบนารี สามารถทำได้โดยใช้เทคนิคการทำเทรชโฮล (Threshold Technique) โดยพิจารณาว่าจุดภาพใดควรจะเป็นจุดขาวหรือจุดดำ จะกระทำโดยการเปรียบเทียบระหว่างจุดภาพเริ่มต้นกับค่าคงที่ค่าหนึ่งที่เรียกว่า "ค่าเทรชโฮล" (Threshold Value) เทคนิคนี้ใช้กันมากในกรณีที่ข้อมูลภาพมีลักษณะแตกต่างกันระหว่างวัตถุ (Object) และพื้นหลัง (Background) โดยค่าของจุดภาพใดๆ ที่มีค่าน้อยกว่าค่าเทรชโฮล จะถูกกำหนดให้เป็น 0 (จุดขาว) และถ้าค่าของจุดภาพใดๆ ที่มีค่ามากกว่า หรือเท่ากับค่าเทรชโฮลจะถูกเปลี่ยนให้เป็น 1 (จุดดำ) ซึ่งการทำงานสามารถแสดงได้ดังสมการที่ 4.3

โดย

$$b(x,y) = \begin{cases} 1 & ; \ g(x,y) > Thr \\ 0 & ; \ g(x,y) \leq Thr \end{cases} \qquad \text{.....(4.3)}$$

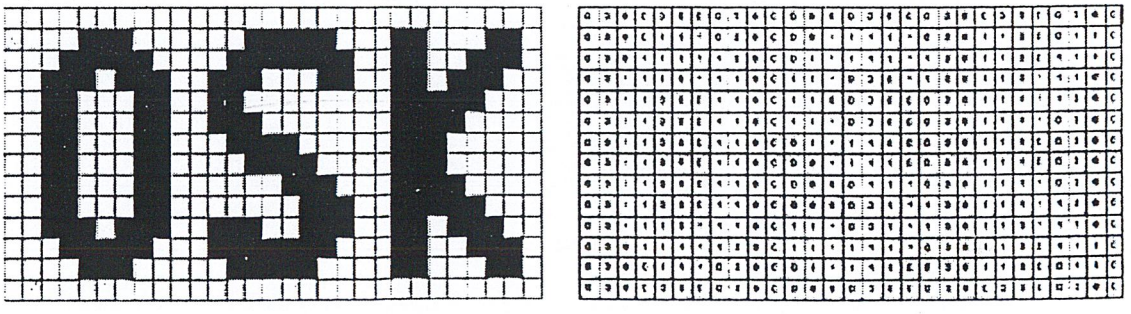
b(x,y) ข้อมูลภาพผลลัพธ์เป็นภาพไบนารี

g(x,y) ข้อมูลภาพอินพุทที่มีระดับความเข้ม 0 ถึง L ระดับ

Thr ค่าเทรชโฮล เป็นค่าคงที่ที่อยู่ระหว่าง 0 ถึง L ระดับ

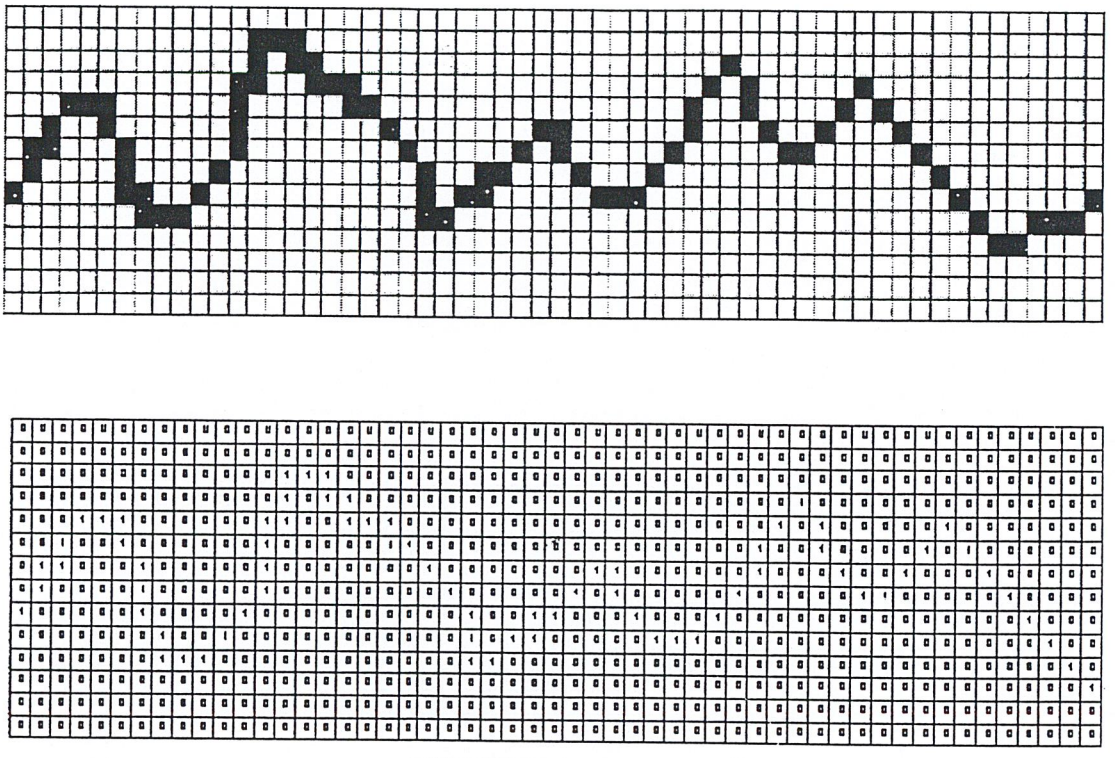
: โดยที่ L คือระดับความเข้มของจุดภาพสูงสุด

ดังนั้นเมื่อเรากำหนดค่าเทรชโฮลให้มีค่าเป็น 0 เมื่อความเข้มของภาพอยู่ที่ระดับ 0 หรือเป็นสีขาว ดังนั้นเมื่อข้อมูลภาพอินพุทมีสีขาวเราจะได้ข้อมูลภาพผลลัพธ์เป็นภาพไบนารีที่มี b(x,y) = 0 และเมื่อข้อมูลภาพอินพุทที่มีความเข้มของภาพที่ระดับอื่นๆ หรือสีอื่นที่ไม่ใช่สีขาว เราจะได้ข้อมูลภาพผลลัพธ์ที่เป็นภาพไบนารีที่มี b(x,y) = 1



รูปที่ 4.1 แสดงการแปลงรูปเกรย์สเกลไปเป็นภาพไบนารี

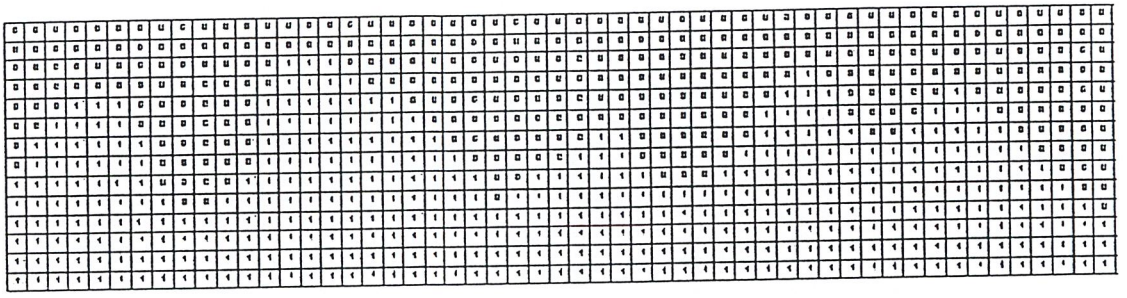
เริ่มแรกให้โปรแกรมในส่วนของการแปลงไฟล์นี้สแกนหาจุดตามแนวแกน y โดยสแกนจากบนลงล่าง (vertical scan) โดยเริ่มจากขอบบนสุดของกรอบ และถ้าพบจุด pixel ที่มีค่าระดับความเข้มน้อยกว่าหรือเท่ากับค่าเทรชโฮล ก็จะทำการเซตข้อมูลในไบนารีไฟล์ที่ตำแน่ง x,y นั้นให้มีค่าเท่ากับ '0' ในทางตรงข้ามถ้าพบจุด pixel ที่มีค่าระดับความเข้มมากกว่าค่าเทรชโฮล ก็จะทำการเซตข้อมูลในไบนารีไฟล์ที่ตำแหน่ง x,y นั้นให้มีค่าเท่ากับ '1'



รูปที่ 4.2 การแปลงกราฟไปเป็นภาพไบนารีโดยการแปลงจุดดำเป็น '1' และจุดขาวเป็น '0'

แต่ในทางปฏิบัติในการแปลงรูปแบบนี้จะมีเปอร์เซ็นต์ความผิดพลาดสูงเมื่อนำไปใช้ในการประมวลผลด้วยนิวรอลเน็ตเวิร์ก สาเหตุเนื่องจากเมื่อนำนิวรอน (กล่าวโดยละเอียดในบทถัดไป) มาจับในตำแหน่งต่างๆ ซึ่งถ้าเราใช้การแปลงในรูปแบบนี้จะมีโอกาสที่นิวรอนจะจับไม่พบตำแหน่งที่มีค่าเป็น '1' สูงดังรูปที่ 4.2 ทำให้มีโอกาสเป็นไปได้ว่าโปรแกรมจะทำงานผิดพลาดเนื่องจากเข้าในว่าไม่มีข้อมูล ณ ตำแหน่งนั้น

ดังนั้นเราจึงต้องประยุกต์โดยการทำให้พื้นที่ใต้กราฟของข้อมูลในไบนารีไฟล์ b(x,y) มีค่าเป็น ‘1’ ทั้งหมดดังรูปที่ 4.3



รูปที่ 4.3 การแปลงกราฟไปเป็นภาพไบนารีโดยการทำให้พื้นที่ใต้กราฟเป็น ‘1’

จะเห็นได้ว่าถ้าเราทำให้พื้นที่ใต้กราฟของข้อมูลในไบนารีไฟล์ b(x,y) มีค่าเป็น ‘1’ ทั้งหมดแล้วจะสามารถขจัดข้อผิดพลาดนี้ได้เพราะนิวรอนสามารถจับข้อมูลในไบนารีไฟล์ b(x,y) มีค่าเป็น ‘1’ ได้แน่นอนแม้ว่าจะไม่ใช่จุดแรกจากด้านบนที่สแกนลงมาแต่ก็ใกล้เคียงมากเนื่องจากเรากำหนดให้นิวรอนแต่ละตัวสามารถจับข้อมูลได้อย่างละเอียด

บทที่ 5
หลักการพยากรณ์ข้อมูล

ในโปรแกรมนี้ผู้เขียนใช้นิวรอลเน็ตเวิร์กชนิดไม่ถ่วงน้ำหนัก (Weightless Artificial Neural Network) ซึ่งไม่มีการป้อนกลับและไม่มีการถ่วงน้ำหนัก ทำให้โปรแกรมนี้มีลักษณะคล้ายการจดจำข้อมูลในอดีตแล้วนำมาวิเคราะห์เพื่อหาจุดต่อไปในอนาคต ซึ่งมีหลักการและตัวอย่างการคำนวณอยู่ในบทนี้ โดยในบทนี้จะทำให้เราสามารถทำความเข้าใจกระบวนการของการวิเคราะห์ของนิวรอนในการจดจำและการลงคะแนนเสียง (voting) และท้ายสุดด้วยการพยากรณ์ต่อไป การวิเคราะห์นี้สามารถทำได้ใกล้เคียงของจริงก็ต่อเมื่อต้องมีการสอน (Training) โดยการป้อนข้อมูลซึ่งเป็นตัวอย่างที่นิวรอนใช้ในการเรียนรู้เป็นจำนวนมากๆ ยิ่งมากเท่าใด ความแม่นยำก็ยิ่งมีมากเท่านั้น ซึ่งในการพยากรณ์โดยนิวรอลเน็ตเวิร์กนั้นเราได้ทำการวิเคราะห์อยู่หลายรูปเพื่อหารูปแบบที่เหมาะสมในการพยากรณ์ ซึ่งแต่ละรูปแบบก็จะมีอยู่ 2 ส่วนใหญ่ๆ คือ การฝึกสอน (Training) และการพยากรณ์ (Prediction) ซึ่งรูปแบบของการพยากรณ์โดยนิวรอลเน็ตเวิร์กมีรูปแบบดังนี้

5.1 การพยากรณ์โดยข้อมูลผ่านการNormalizeก่อนแล้วหาแนวโน้ม โดยเอาท์พุทเป็นแบบไม่ระบุตำแหน่งแน่นอน

ในรูปแบบนี้เราได้แบ่งข้อมูลเป็น 2 ส่วน คือ ส่วนของการฝึกสอน และส่วนของการพยากรณ์ ในส่วนของการฝึกสอนเราจะแบ่งออกได้เป็นอีก 2 แบบ ซึ่งแต่ละส่วนมีรายละเอียดดังนี้

5.1.1 กระบวนการฝึกสอน Neuron

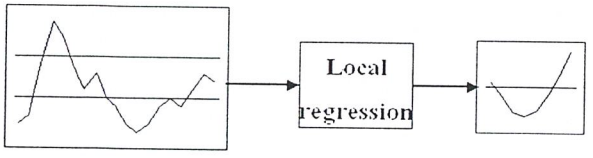
เพื่อให้การพยากรณ์เป็นไปอย่างแม่นยำ จึงต้องมีการฝึกสอน (Training) โดยการใช้ข้อมูลที่เป็นตัวอย่างในอดีตที่มีความสัมพันธ์กับข้อมูลในปัจจุบันที่ต้องการจะทำนายดังนั้นการที่จะสามารถทำนายได้อย่างแม่นยำนั้นก็ต้องอาศัยการเรียนรู้จำนวนมากๆ นั้นก็คือข้อมูลที่เราป้อนให้กับโปรแกรมนั้นเอง ซึ่งหากมีปริมาณมากเท่าใดก็จะยิ่งทำให้นิวรอนฉลาดยิ่งขึ้น ซึ่งก็ทำให้สามารถทำนายได้แม่นยำยิ่งขึ้นด้วย

ในการฝึกสอนนั้นผู้เขียนได้ใช้ข้อมูลตัวอย่างเป็นค่าของราคาซื้อขายหุ้นเมื่อตลาดปิดทำการ และปริมาณของหุ้นที่ทำการซื้อขายในแต่ละวัน จากตลาดหลักทรัพย์แห่งประเทศไทย

5.1.1.1 การฝึกสอนแบบทั่วไป

หลักการฝึกสอนนิวรอน เราจะทำการฝึกสอนทั้ง 2 ส่วน คือ ส่วนของข้อมูลราคาปิด และส่วนของข้อมูลปริมาณการซื้อขาย

ขั้นที่ 1 ทำการ Normalize ข้อมูล ซึ่งข้อมูลที่จะนำมาฝึกสอนนั้นเราจะมีการNormalizeก่อนเพื่อกำจัด noise จะทำให้ข้อมูลเรียบขึ้น โดยใช้วิธีการ Local regression ดังรูปที่ 5.1

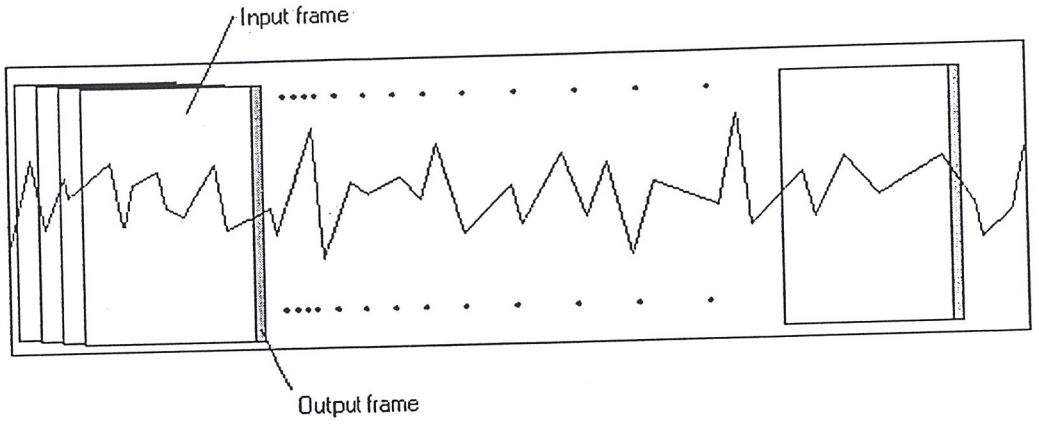


รูปที่ 5.1 แสดงข้อมูลผ่านการ Normalize

ขั้นที่ 2 หาค่าแนวโน้มของข้อมูลในการฝึกสอนนั้นเราจะใช้ค่าแนวโน้มมาเป็นตัวฝึกสอนนิวรอน ซึ่งค่าแนวโน้มในที่นี้ก็คือ ข้อมูลที่อยู่ในรูปของผลต่างของค่าข้อมูลในวันถัดไปลบด้วยค่าข้อมูลในวันปัจจุบัน

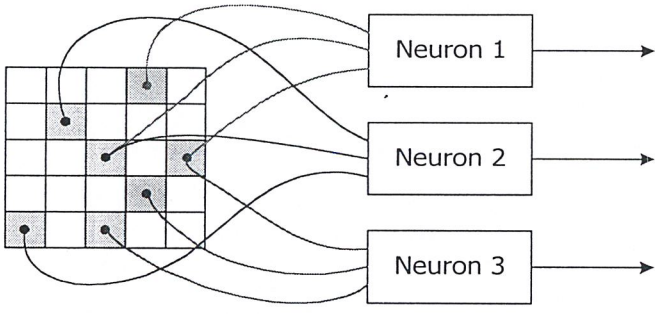
ขั้นที่ 3 กำหนดกรอบของภาพไบนารี (ไบนารีไฟล์ที่แปลงมาจากกราฟของแนวโน้มนั้นเอง) เมื่อเราได้ขนาดของกรอบ ซึ่งกรอบนี้ก็คือขอบเขตของนิวรอนทั้งหมดที่สามารถจับข้อมูลได้ในขณะนั้น ณ กรอบนั้น ซึ่งค่าเอาท์พุทของกรอบไบนารีนั้นจะอยู่ในรูปของแนวโน้มในข้อมูลถัดไป

ขั้นที่ 4 กำหนดจำนวนนิวรอน จำนวนข้อมูล และกำหนดตำแหน่งของข้อมูลที่นิวรอนแต่ละตัวต้องจับของกรอบไบนารี และทำการเลื่อนกรอบไบนารีไปจากจุดเดิมในแนวแกน X เป็นจำนวน 1 วัน ซึ่งกรอบนี้จะมีส่วนที่จะไปทับกับกรอบเดิมในลักษณะเหลื่อมกันดังรูปที่ 5.2 เป็นผลให้ในกรอบอ้างอิงใหม่นี้มีรูปแบบของข้อมูลในภาพไบนารีเปลี่ยนไป เพื่อให้นิวรอนแต่ละตัวทำการจับข้อมูลในกรอบไบนารีใหม่ ซึ่งจำนวนข้อมูลและตำแหน่งของข้อมูลที่นิวรอนจับนั้นจะเหมือนกันทุกกรอบไบนารี ทำจนหมดข้อมูล

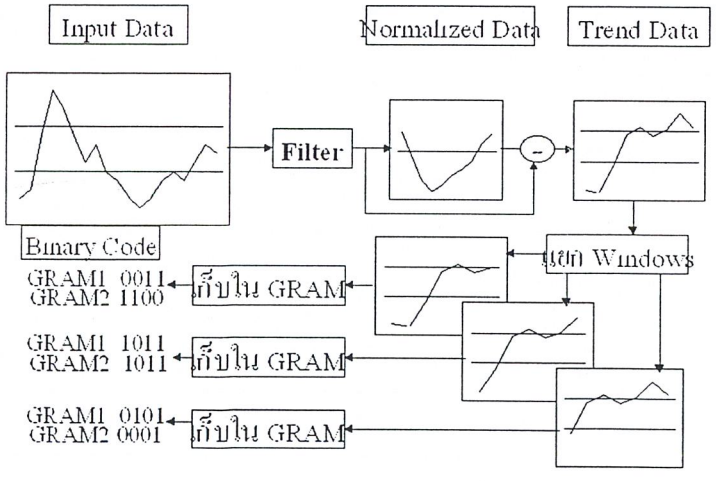


รูปที่ 5.2 ภาพการเลื่อนกรอบของข้อมูล

ขั้นที่ 5 ทำการเก็บค่าข้อมูลของแต่ละของนิวรอนแต่ละกรอบไบนารีไว้ ดังรูปที่ 5.3



รูปที่ 5.3 การฝึกสอน Neuron



รูปที่ 5.4 ตัวอย่าง Block diagram ส่วนของการเรียนรู้

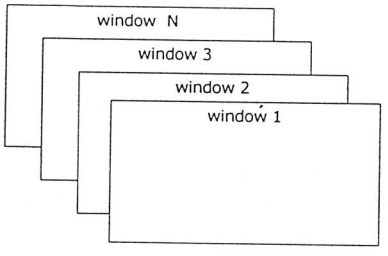
5.1.1.2 การฝึกสอนแบบ Optimize

ขั้นที่ 1 ทำการ Normalize ข้อมูล ซึ่งข้อมูลที่จะนำมาฝึกสอนนั้นเราจะมีการNormalizeก่อน เพื่อกำจัด noise จะทำให้ข้อมูลเรียบขึ้น โดยใช้วิธีการ Local regression

ขั้นที่ 2 หาค่าแนวโน้มของข้อมูล ในการฝึกสอนนั้นเราจะใช้ค่าแนวโน้มมาเป็นตัวฝึกสอนนิวรอน ซึ่งค่าแนวโน้มในที่นี้ก็คือ ข้อมูลที่อยู่ในรูปของผลต่างของค่าข้อมูลในวันถัดไปลบด้วยค่าข้อมูลในวันปัจจุบัน

ขั้นที่ 3 กำหนดกรอบของภาพไบนารี ให้มีค่าเท่ากันทุกๆ กรอบของข้อมูลที่ใช้ในการฝึกสอน

ขั้นที่ 4 นำกรอบไบนารีที่ได้จากข้อมูลฝึกสอนทั้งหมดที่แบ่งไว้ มาซ้อนทับกัน ดังรูปที่ 5.5



รูปที่ 5.5 การซ้อนทับกันของกรอบไบนารีในข้อมูลนำมาฝึกสอน

ขั้นที่ 5 หาจุดที่มีการเปลี่ยนแปลงของข้อมูลมากที่สุด โดยขั้นตอนการหาคือหลังจากที่นำกรอบไบนารีมาซ้อนทับกันแล้ว นับค่าที่เป็น 1 ในแต่ละพิกเซล แต่จะใช้ตำแหน่งพิกเซลที่มีค่า 1 เป็นจำนวนกึ่งกลางของพิกเซลทั้งหมด เป็นตำแหน่งที่ให้นิวรอนจับ เมื่อตำแหน่งนั้นมีจำนวนไม่เพียงพอ ก็จะใช้ตำแหน่งที่เลื่อนจากค่ากลางไปยังค่ามากสุด สลับกับค่าน้อยสุด ไปเรื่อยๆจนกว่าจะได้ตำแหน่งที่นิวรอนต้องการจับจนครบ

5.1.2 การพยากรณ์ข้อมูลของ Neuron

เราจะใช้ข้อมูลในส่วนของราคาปิดของหุ้นและข้อมูลในส่วนของปริมาณการซื้อขาย มาเป็นส่วนประกอบในการพยากรณ์ค่าแนวโน้มของหุ้น เพื่อให้ได้ราคาซื้อขายของหุ้นในวันถัดไป

ขั้นที่ 1 รับค่าของข้อมูลที่จะใช้ในการพยากรณ์ แล้วทำการ Normalize เช่นเดียวกับข้อมูลในส่วนของการฝึกสอน

ขั้นที่ 2 หาค่าแนวโน้มของข้อมูลอินพุท

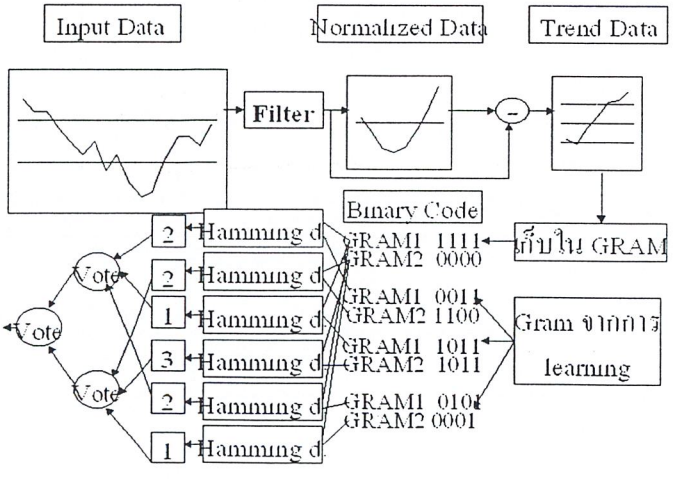
ขั้นที่ 3 กำหนดกรอบไบนารีให้มีขนาดเท่ากับข้อมูลในส่วนของการฝึกสอน

ขั้นที่ 4 กำหนดจำนวนนิวรอน จำนวนข้อมูล และกำหนดตำแหน่งของข้อมูลที่นิวรอนแต่ละตัวต้องจับของกรอบไบนารี เป็นอย่างเดียวกับข้อมูลในส่วนของการฝึกสอน

ขั้นที่ 5 เป็นการเปรียบเทียบข้อมูล Binary ของอินพุท ทำทั้งในส่วนของราคา และปริมาณการซื้อขาย ซึ่งเราจะทำการเปรียบเทียบแต่ละนิวรอนกับกรอบไบนารีทุกๆ กรอบของข้อมูลที่นำมาฝึกสอน แล้วทำการเก็บข้อมูลไว้ว่าในนิวรอนตัวที่ N นั้นมีข้อมูลเหมือนกรอบไบนารีไหนบ้าง หรือถ้าไม่มีที่เหมือนก็หาที่มีลักษณะคล้ายมากที่สุด ทำเช่นนี้ทุกๆ นิวรอน

ขั้นที่ 6 การโหวตหรือลงคะแนนของนิวรอน เมื่อเราได้ทำการเปรียบเทียบแต่ละนิวรอนในส่วนของข้อมูล input กับข้อมูลฝึกสอน จากนิวรอนแต่ละตัวก็จะมาลงคะแนนเสียงว่าจะให้กรอบใดเป็นกรอบที่เราต้องการ เราจะนำนิวรอนทุกตัวของข้อมูล input มาโหวตดูว่าในนิวรอนทุกตัวมีกรอบไบนารีไหนเหมือนกัน โดยทำการนับคะแนนให้กับกรอบไบนารีที่ซ้ำ ซึ่งในส่วนนี้เราจะทำทั้งในส่วนของราคาและปริมาณการซื้อขาย โดยคะแนนที่ได้ของแต่ละกรอบไบนารีนั้นในส่วนของราคาเราจะให้มีค่าน้ำหนักมากกว่าในส่วนของปริมาณเป็น 4 : 1 ซึ่งเราจะนำคะแนนที่นับได้ของแต่ละกรอบไบนารีในส่วนของราคามาคูณ 4 และส่วนของปริมาณมาคูณ 1 เพราะเราให้ความสำคัญในด้านราคามากกว่าปริมาณ แล้วนำคะแนนที่ได้ทั้งสองส่วนมารวมกัน แล้วหากรอบไบนารีที่มีคะแนนมากที่สุด

ขั้นที่ 7 หาค่าเอาท์พุท เมื่อเราได้กรอบไบนารีที่มีคะแนนมากที่สุด แล้วเราก็นำค่าแนวโน้มถัดไปจากกรอบไบนารีที่เป็นเอาท์พุทของกรอบไบนารีนั้น มาเป็นค่าแนวโน้มของข้อมูลที่ได้จากการป้อนinput ก็คือค่าพยากรณ์ที่ได้ และในส่วนของราคานั้นเราก็นำค่าแนวโน้มที่ได้ไปบวกเข้ากับค่าราคาของข้อมูลตัวสุดท้ายของข้อมูล input



รูปที่ 5.6 ตัวอย่าง Block diagram ส่วนของการพยากรณ์

5.2 การพยากรณ์โดยข้อมูลถูกนำมาหาแนวโน้มก่อนผ่านการNormalize โดยเอาท์พุทเป็นแบบไม่ระบุตำแหน่งแน่นอน

ในรูปแบบนี้เราได้แบ่งข้อมูลเป็น 2 ส่วน คือ ส่วนของการฝึกสอน และส่วนของการพยากรณ์ ในส่วนของการฝึกสอนนั้นเราได้ใช้การฝึกสอนแบบ Optimize เท่านั้น เพราะการฝึกสอนแบบ Optimize นั่นเป็นรูปแบบที่ดีที่สุด แต่จะมีขั้นตอนการทำที่ต่างไปจากเดิม ซึ่งแต่ละขั้นตอนมีรายละเอียดดังนี้

5.2.1 การฝึกสอนแบบ Optimize

ขั้นที่ 1 หาค่าแนวโน้มของข้อมูล ในการฝึกสอนนั้นเราจะใช้ค่าแนวโน้มมาเป็นตัวฝึกสอนนิวรอนก่อน

ขั้นที่ 2 ทำการ Normalize ข้อมูล ซึ่งข้อมูลที่จะนำมาฝึกสอนนั้นเราจะมีการNormalizeก่อนเพื่อกำจัด noise จะทำให้ข้อมูลเรียบขึ้น โดยใช้วิธีการ Local regression

ขั้นที่ 3 กำหนดกรอบของภาพไบนารี ให้มีค่าเท่ากันทุกๆ กรอบของข้อมูลที่ใช้ในการฝึกสอน

ขั้นที่ 4 นำกรอบไบนารีที่ได้จากข้อมูลฝึกสอนทั้งหมดที่แบ่งไว้ มาซ้อนทับกัน

ขั้นที่ 5 หาจุดที่มีการเปลี่ยนแปลงของข้อมูลมากที่สุด โดยขั้นตอนการหา คือหลังจากที่นำกรอบไบนารีมาซ้อนทับกันแล้ว นับค่าที่เป็น 1 ในแต่ละพิกเซล แต่จะใช้ตำแหน่งพิกเซลที่มีค่า 1 เป็นจำนวนกึ่งกลางของพิกเซลทั้งหมด เป็นตำแหน่งที่ให้นิวรอนจับ เมื่อตำแหน่งนั้นมีจำนวนไม่เพียงพอ ก็จะใช้ตำแหน่งที่เลื่อนจากค่ากลางไปยังค่ามากสุด สลับกับค่าน้อยสุด ไปเรื่อยๆจนกว่าจะได้ตำแหน่งที่นิวรอนต้องการจับจนครบ

5.2.2 การพยากรณ์ข้อมูลของ Neuron

เราจะใช้ข้อมูลในส่วนของราคาปิดของหุ้นและข้อมูลในส่วนของปริมาณการซื้อขาย มาเป็นส่วนประกอบในการพยากรณ์ค่าแนวโน้มของหุ้น เพื่อให้ได้ราคาซื้อขายของหุ้นในวันถัดไป

ขั้นที่ 1 รับค่าของข้อมูลที่จะใช้ในการพยากรณ์ แล้วหาค่าแนวโน้มของข้อมูลอินพุท เช่นเดียวกับข้อมูลในส่วนของการฝึกสอน

ขั้นที่ 2 ทำการ Normalize ข้อมูลที่ผ่านการหาค่าแนวโน้ม

ขั้นที่ 3 กำหนดกรอบไบนารีให้มีขนาดเท่ากับข้อมูลในส่วนของการฝึกสอน

ขั้นที่ 4 กำหนดจำนวนนิวรอน จำนวนข้อมูล และกำหนดตำแหน่งของข้อมูลที่นิวรอนแต่ละตัวต้องจับของกรอบไบนารี เป็นอย่างเดียวกับข้อมูลในส่วนของการฝึกสอน

ขั้นที่ 5 เป็นการเปรียบเทียบนิวรอนตัวที่ N ในกรอบไบนารี M ของส่วนพยากรณ์ ไปทำการเปรียบเทียบกับนิวรอน N ทุกกรอบไบนารีในส่วนของการฝึกสอน ว่านิวรอนตัวนี้คล้ายกับนิวรอนตัวไหนของการฝึกสอนในกรอบมากที่สุด ให้นำค่าไบนารีของนิวรอนนั้นมาเป็นเอาท์พุท โดยคะแนนที่ได้ของแต่ละกรอบไบนารีนั้นในส่วนของราคาเราจะให้มีค่าน้ำหนักมากกว่าในส่วนของปริมาณเป็น 4 : 1 ซึ่งเราจะนำคะแนนที่นับได้ของแต่ละกรอบไบนารีในส่วนของราคามาคูณ 4 และส่วนของปริมาณมาคูณ 1 เพราะเราให้ความสำคัญในด้านราคามากกว่าปริมาณ แล้วนำคะแนนที่ได้ทั้งสองส่วนมารวมกัน แล้วหากรอบไบนารีที่มีคะแนนมากที่สุด

5.3 การพยากรณ์โดยข้อมูลถูกนำมาหาแนวโน้มก่อน แล้วไม่ผ่านการNormalize โดยเอาท์พุทเป็นแบบไม่ระบุตำแหน่งแน่นอน

ในรูปแบบนี้เราได้แบ่งข้อมูลเป็น 2 ส่วน คือ ส่วนของการฝึกสอน และส่วนของการพยากรณ์ ซึ่งทั้งสองส่วนนี้จะมีขั้นตอนเหมือนกับในหัวข้อที่ 5.2 แต่ทั้งสองส่วนจะไม่มีการผ่านการ Normalize ของข้อมูลแนวโน้ม

5.4 การพยากรณ์โดยข้อมูลถูกนำมาหาแนวโน้มก่อนผ่านการNormalize โดยเอาท์พุทของนิวรอนจะให้เอาท์พุทออกมาโดยไม่ผ่านการโหวต

ในรูปแบบนี้เราได้แบ่งข้อมูลเป็น 2 ส่วนเช่นเดิม คือ ส่วนของการฝึกสอน และส่วนของการพยากรณ์ ในส่วนของการฝึกสอนนั้นเราได้ใช้การฝึกสอนแบบ Optimize เท่านั้นซึ่งจะเหมือนกับหัวข้อที่ 5.2.1 แต่จะกำหนดตำแหน่งที่จับเป็นเอาท์พุทให้กับนิวรอนแต่ละตัวให้ไม่ซ้ำกัน โดยนิวรอนแต่ละตัวจะมีตำแหน่งเอาท์พุทที่แน่นอน โดยนิวรอนตัวที่ N จะเก็บค่าในตำแหน่งที่ N ซึ่งเป็นไบนารีของวันถัดไปของกรอบไบนารีนั้น ซึ่งส่วนของการพยากรณ์จะมีรายละเอียดดังนี้

5.4.1 การพยากรณ์ข้อมูลของ Neuron

เราจะใช้ข้อมูลในส่วนของราคาปิดของหุ้นและข้อมูลในส่วนของปริมาณการซื้อขาย มาเป็นส่วนประกอบในการพยากรณ์ค่าแนวโน้มของหุ้น เพื่อให้ได้ราคาซื้อขายของหุ้นในวันถัดไป

ขั้นที่ 1 รับค่าของข้อมูลที่จะใช้ในการพยากรณ์ แล้วหาค่าแนวโน้มของข้อมูลอินพุท เช่นเดียวกับข้อมูลในส่วนของการฝึกสอน

ขั้นที่ 2 ทำการ Normalize ข้อมูลที่ผ่านการหาค่าแนวโน้ม

ขั้นที่ 3 กำหนดกรอบไบนารีให้มีขนาดเท่ากับข้อมูลในส่วนของการฝึกสอน

ขั้นที่ 4 กำหนดจำนวนนิวรอน จำนวนข้อมูล และกำหนดตำแหน่งของข้อมูลที่นิวรอนแต่ละตัวต้องจับของกรอบไบนารี เป็นอย่างเดียวกับข้อมูลในส่วนของการฝึกสอน

ขั้นที่ 5 เป็นการเปรียบเทียบนิวรอนตัวที่ N ในกรอบไบนารี M ของส่วนพยากรณ์ ไปทำการเปรียบเทียบกับนิวรอน N ทุกกรอบไบนารีในส่วนของการฝึกสอน ว่านิวรอนตัวนี้คล้ายกับนิวรอนตัวไหนของการฝึกสอนในกรอบมากที่สุด ให้นำค่าไบนารีของนิวรอนนั้นมาเป็นเอาท์พุท โดยให้อัตราส่วนคะแนนระหว่างราคากับปริมาณเป็น 4:1 เหมือนเดิม แล้วนำคะแนนที่ได้ทั้งสองส่วนมารวมกัน แล้วหากรอบไบนารีที่มีคะแนนมากที่สุด จากนั้นนำคำตอบที่ได้จากนิวรอนแต่ละตัว ซึ่งเป็นคำตอบของเอาท์พุทที่อยู่ในตำแหน่งต่างกัน มาตอบเลยโดยไม่ต้องผ่านการโหวตว่าเหมือนกรอบไบนารีใด ซึ่งจะได้คำตอบออกมาเป็นแถวไบนารีบิต

ขั้นที่ 6 นำแถวไบนารีบิตที่ได้มาแปลงเป็นค่าตัวเลข

5.5 การพยากรณ์โดยข้อมูลถูกนำมาหาแนวโน้มก่อนผ่านการNormalize โดยเอาท์พุทเป็นแบบระบุตำแหน่งแน่นอน และนิวรอนแต่ละตัวจะมาจากนิวรอน 5 ตัวขนานกัน

ในรูปแบบนี้เราได้แบ่งข้อมูลเป็น 2 ส่วนเช่นเดิม คือ ส่วนของการฝึกสอน และส่วนของการพยากรณ์ ในส่วนของการฝึกสอนนั้นเราได้ใช้การฝึกสอนแบบ Optimize เท่านั้นซึ่งจะเหมือนกับหัวข้อที่ 5.4 แต่จะต้องมีการทำซ้ำ 5 ครั้ง เพื่อนำไปใช้ในส่วนของการพยากรณ์ ซึ่งส่วนของการพยากรณ์จะมีรายละเอียดดังนี้

5.5.1 การพยากรณ์ข้อมูลของ Neuron

เราจะใช้ข้อมูลในส่วนของราคาปิดของหุ้นและข้อมูลในส่วนของปริมาณการซื้อขาย มาเป็นตัวประกอบในการพยากรณ์ค่าแนวโน้มของหุ้น เพื่อให้ได้ราคาซื้อขายของหุ้นในวันถัดไป

ขั้นที่ 1 รับค่าของข้อมูลที่จะใช้ในการพยากรณ์ แล้วหาค่าแนวโน้มของข้อมูลอินพุท เช่นเดียวกับข้อมูลในส่วนของการฝึกสอน

ขั้นที่ 2 ทำการ Normalize ข้อมูลที่ผ่านการหาค่าแนวโน้ม

ขั้นที่ 3 กำหนดกรอบไบนารีให้มีขนาดเท่ากับข้อมูลในส่วนของการฝึกสอน

ขั้นที่ 4 กำหนดจำนวนนิวรอน จำนวนข้อมูล และกำหนดตำแหน่งของข้อมูลที่นิวรอนแต่ละตัวต้องจับของกรอบไบนารี เป็นอย่างเดียวกับข้อมูลในส่วนของการฝึกสอน

ขั้นที่ 5 เป็นการเปรียบเทียบนิวรอนตัวที่ N ในกรอบไบนารี M ของส่วนพยากรณ์ ไปทำการเปรียบเทียบกับนิวรอน N ทุกกรอบไบนารีในส่วนของการฝึกสอน ว่านิวรอนตัวนี้คล้ายกับนิวรอนตัวไหนของการฝึกสอนในกรอบมากที่สุด ให้นำค่าไบนารีของนิวรอนนั้นมาเป็นเอาท์พุท โดยให้อัตราส่วนคะแนนระหว่างราคากับปริมาณเป็น 4:1 เหมือนเดิม แล้วนำคะแนนที่ได้ทั้งสองส่วนมารวมกัน แล้วหากรอบไบนารีที่มีคะแนนมากที่สุด จากนั้นนำคำตอบที่ได้จากนิวรอนแต่ละตัวเก็บไว้ก่อน

ขั้นที่ 6 ทำซ้ำในขั้นตอนที่ 1 ถึง ขั้นตอนที่ 5 อีก 5 ครั้ง แล้วนำค่าคำตอบที่นิวรอนแต่ละตัวได้จากการพยากรณ์ 5 ครั้ง มาทำการโหวตอีกครั้งหนึ่ง นั่นคือผลคำตอบจากนิวรอน 5 คำตอบ จะเหลือเพียงคำตอบเดียวที่จะเป็นคำตอบของนิวรอนตัวที่ N

5.6 การพยากรณ์โดยข้อมูลถูกนำมาหาแนวโน้มก่อนผ่านการNormalize โดยเอาท์พุทเป็นแบบระบุตำแหน่งแน่นอน แต่มีการ Feedback

ในรูปแบบนี้เราได้แบ่งข้อมูลเป็น 2 ส่วนเช่นเดิม คือ ส่วนของการฝึกสอน และส่วนของการพยากรณ์ ในส่วนของการฝึกสอนนั้นเราได้ใช้การฝึกสอนแบบ Optimize เท่านั้นซึ่งจะเหมือนกับหัวข้อที่ 5.4 แต่ในส่วนนี้จะมีการเพิ่มตำแหน่งที่นิวรอนแต่ละตัวจับอีก 4 จุด ซึ่ง 4 จุดนี้จะสุ่มมาจากไบนารีข้อมูลของวันรองสุดท้ายของกรอบไบนารีนั้น ซึ่งส่วนของการพยากรณ์จะมีรายละเอียดดังนี้

5.6.1 การพยากรณ์ข้อมูลของ Neuron

เราจะใช้ข้อมูลในส่วนของราคาปิดของหุ้นและข้อมูลในส่วนของปริมาณการซื้อขาย มาเป็นตัวประกอบในการพยากรณ์ค่าแนวโน้มของหุ้น เพื่อให้ได้ราคาซื้อขายของหุ้นในวันถัดไป

ขั้นที่ 1 รับค่าของข้อมูลที่จะใช้ในการพยากรณ์ แล้วหาค่าแนวโน้มของข้อมูลอินพุท เช่นเดียวกับข้อมูลในส่วนของการฝึกสอน

ขั้นที่ 2 ทำการ Normalize ข้อมูลที่ผ่านการหาค่าแนวโน้ม

ขั้นที่ 3 กำหนดกรอบไบนารีให้มีขนาดเท่ากับข้อมูลในส่วนของการฝึกสอน

ขั้นที่ 4 กำหนดจำนวนนิวรอน จำนวนข้อมูล และกำหนดตำแหน่งของข้อมูลที่นิวรอนแต่ละตัวต้องจับของกรอบไบนารี เป็นอย่างเดียวกับข้อมูลในส่วนของการฝึกสอน แต่ในส่วนของการพยากรณ์นี้จะมีการเพิ่มตำแหน่งที่นิวรอนแต่ละตัวจับอีก 4 จุด ซึ่ง 4 จุดนี้จะสุ่มมาจากไบนารีข้อมูลของวันรองสุดท้ายของกรอบไบนารีนั้นเช่นเดียวกับส่วนของการฝึกสอน

ขั้นที่ 5 เป็นการเปรียบเทียบนิวรอนตัวที่ N ในกรอบไบนารี M ของส่วนพยากรณ์ ไปทำการเปรียบเทียบกับนิวรอน N ทุกกรอบไบนารีในส่วนของการฝึกสอน ว่านิวรอนตัวนี้คล้ายกับนิวรอนตัวไหนของการฝึกสอนในกรอบมากที่สุด ให้นำค่าไบนารีของนิวรอนนั้นมาเป็นเอาท์พุท โดยคะแนนที่ได้ของแต่ละกรอบไบนารีนั้นในส่วนของราคาเราจะให้มีค่าน้ำหนักมากกว่าในส่วนของปริมาณเป็น 4 : 1 ซึ่งเราจะนำคะแนนที่นับได้ของแต่ละกรอบไบนารีในส่วนของราคามาคูณ 4 และส่วนของปริมาณมาคูณ 1 เพราะเราให้ความสำคัญในด้านราคามากกว่าปริมาณ แล้วนำคะแนนที่ได้ทั้งสองส่วนมารวมกัน แล้วหากรอบไบนารีที่มีคะแนนมากที่สุด

บทที่ 6
การทดลองและผลการทดลอง

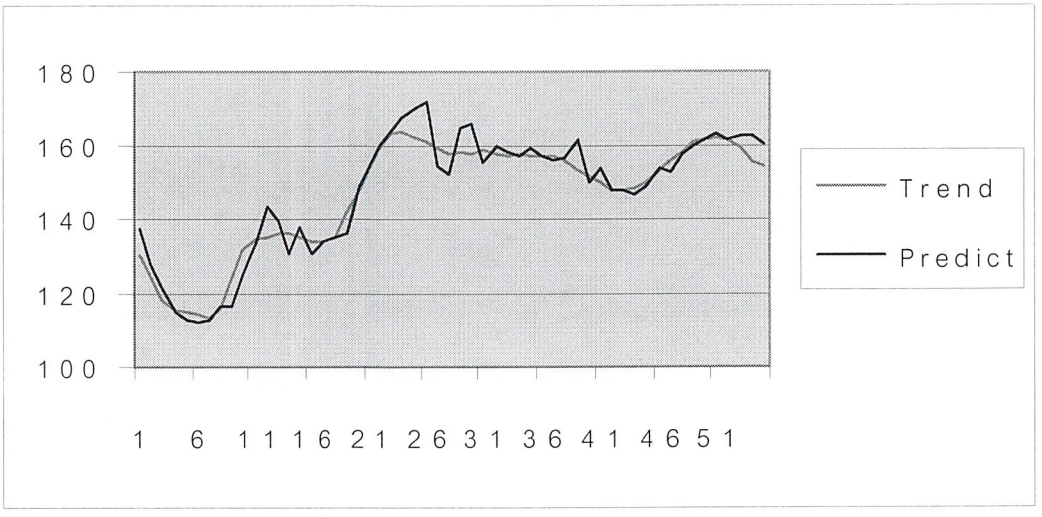
ในการทดลองเพื่อหาค่าแนวโน้มของหุ้นในวันถัดไป ซึ่งจะทำให้ได้ราคาของหุ้นนั้น ในการทดลองนี้ผู้เขียนได้ใช้ข้อมูลจากบริษัท ชิน คอร์ปอเรชั่นส์ จำกัด (มหาชน) (SHIN) เป็นระยะเวลา 8 ปี ตั้งแต่ปี ค.ศ. 1993-2000 แล้วนำค่าที่ได้ไปเปรียบเทียบกับค่าจริงว่ามีความใกล้เคียงกันมากน้อยเพียงใด โดยเราจะทำการวัดออกมาอยู่ในรูปของเปอร์เซ็นต์แนวโน้มที่ถูกต้อง โดยเทียบค่าแนวโน้มที่พยากรณ์ได้กับค่าแนวโน้มของข้อมูลจริง ว่ามีการลดหรือการเพิ่มเหมือนหรือต่างกัน ถ้าเหมือนกันก็แสดงว่าค่าพยากรณ์ที่ได้นั่นถูกต้อง ถ้าต่างกันก็แสดงว่าไม่ถูกต้อง ซึ่งมีการทดลองแบ่งเป็น 6 ตอนตามบทที่ 5 ดังนี้

6.1 การพยากรณ์โดยข้อมูลผ่านการNormalizeก่อนแล้วหาแนวโน้ม โดยเอาท์พุท เป็นแบบไม่ระบุตำแหน่งแน่นอน

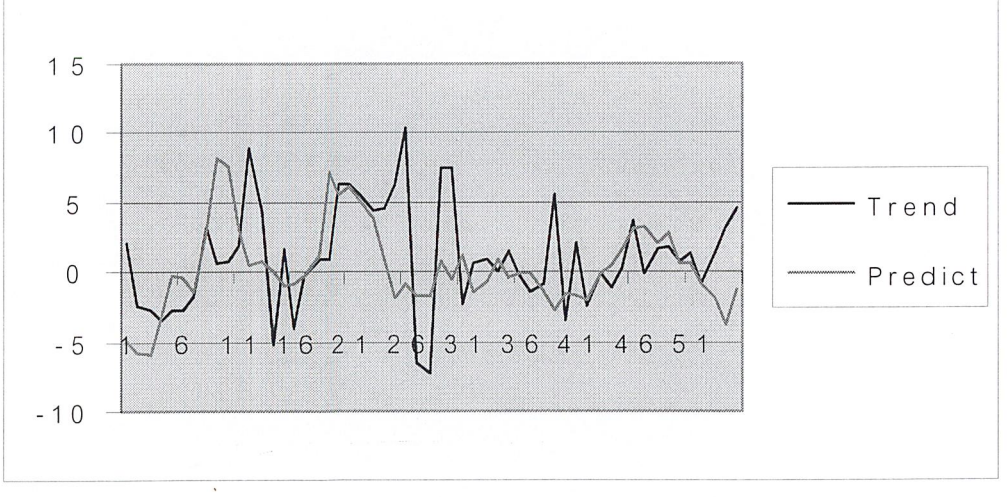
การทดลองในหัวข้อนี้จะใช้ส่วนของโปรแกรมในหัวข้อที่ 5.1 ซึ่งจะได้ทำการเปลี่ยนตัวแปรต่างๆ เพื่อดูผลของการเปลี่ยนแปลง และหาค่าตัวแปรที่เหมาะสม ซึ่งในการทดลองส่วนนี้จะเป็นการเปรียบเทียบค่าที่ได้จากการพยากรณ์กับค่าจริงที่ผ่านการ Normalize แล้ว

<u>การทดลองที่ 6.1.1</u> ผลของการเปลี่ยนตำแหน่งการจับของจุดของนิวรอนใน windows

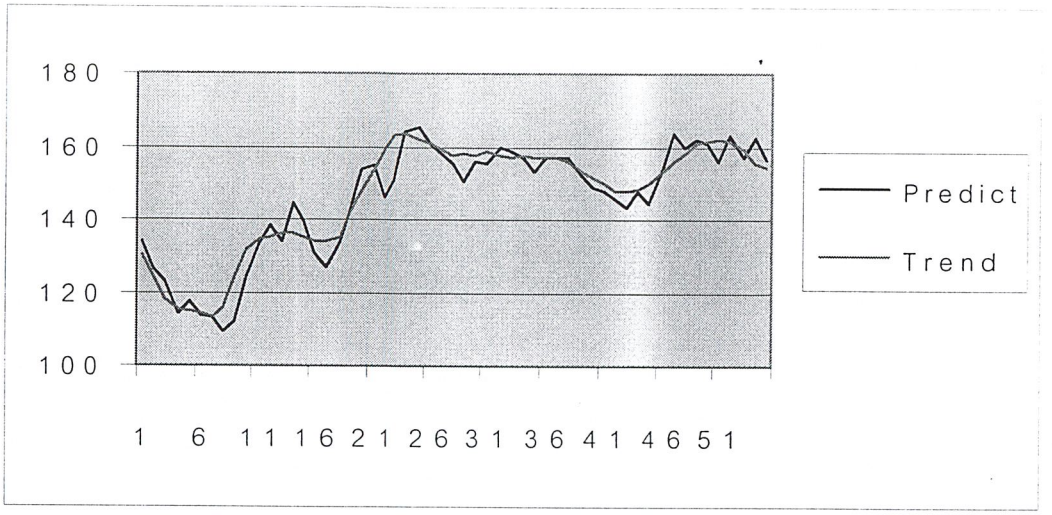
การทดลองตอนนี้จะทำการเปลี่ยนตำแหน่งการจับของจุดใน windowsที่ใช้ในการวิเคราะห์กราฟแนวโน้มราคาหุ้น และกราฟราคาหุ้น เพื่อทดลองว่าจะมีผลต่อค่าคำตอบ (output) ที่ได้จากการวิเคราะห์มากน้อยเพียงใด โดยการทดลองนี้จะใช้นิวรอนทั้งหมด 100 ตัว ตัวละ 20 จุด และใช้ขนาดกรอบอ้างอิง (Frame) 10 วัน ซึ่งจะได้ผลการทดลองดังกราฟที่ 6.1 ถึง 6.6



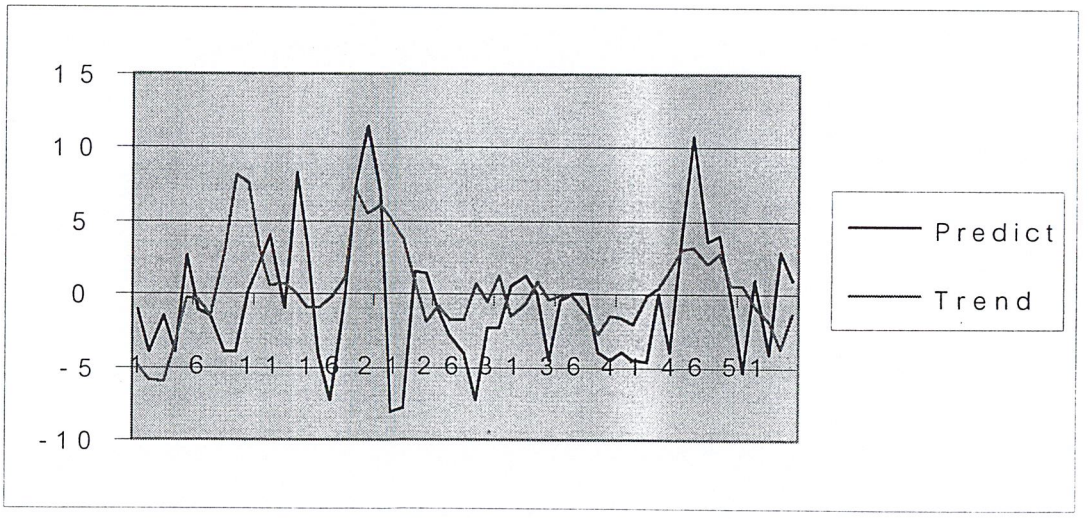
รูปที่ 6.1 กราฟแสดงราคาหุ้นที่พยากรณ์ได้เปรียบเทียบกับราคาจริงเมื่อใช้
การจับจุดของ windows แบบฮาร์โมนิก



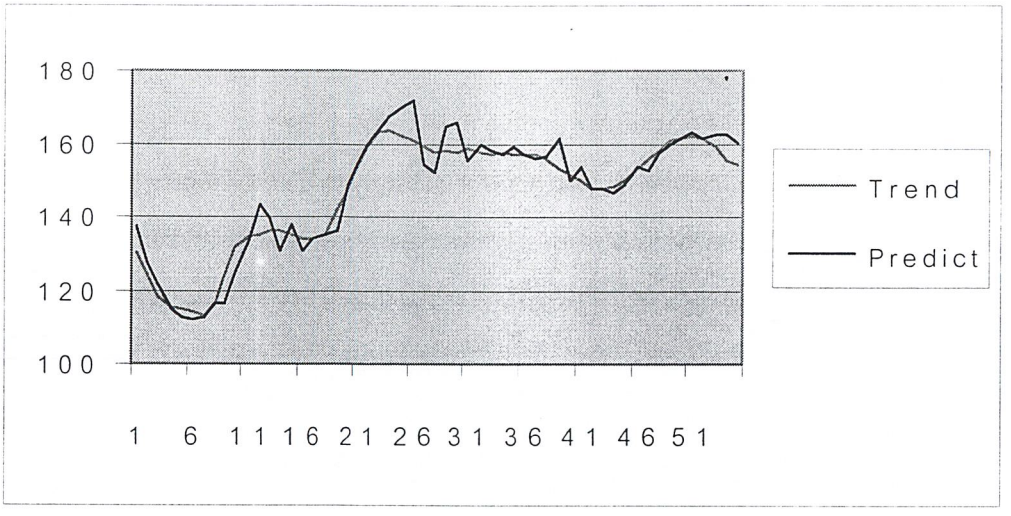
รูปที่ 6.2 กราฟแสดงแนวโน้มของราคาหุ้นที่พยากรณ์ได้เปรียบเทียบกับแนวโน้มของราคาหุ้นจริง
เมื่อใช้การจับจุดของ windows แบบฮาร์โมนิก
ค่าเปอร์เซ็นต์แนวโน้มที่ถูกต้อง 69.07%



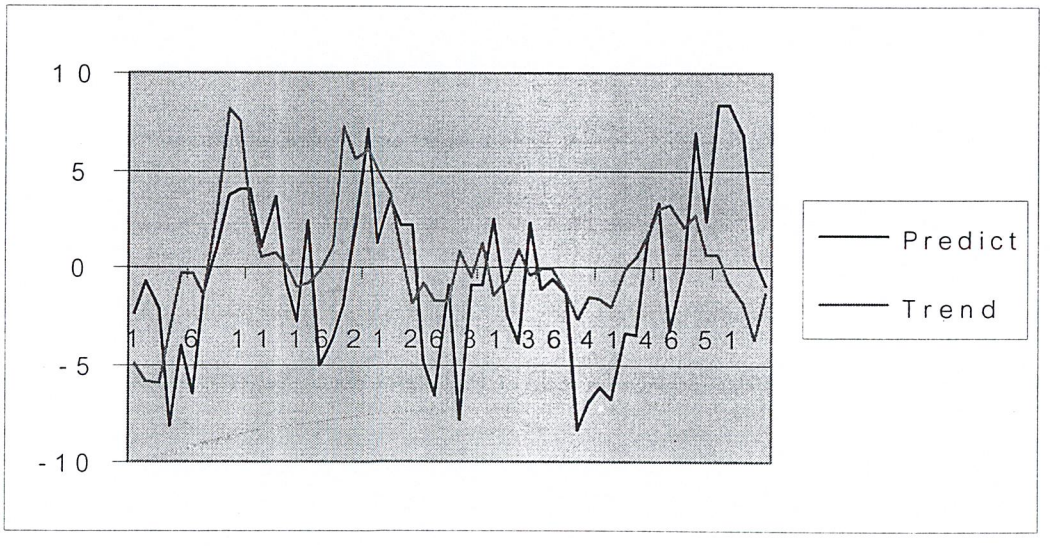
รูปที่ 6.3 กราฟแสดงราคาหุ้นที่พยากรณ์ได้เปรียบเทียบกับราคาจริงเมื่อใช้
การจับจุดของ windows แบบ rectangular



รูปที่ 6.4 กราฟแสดงแนวโน้มของราคาหุ้นที่พยากรณ์ได้เปรียบเทียบกับแนวโน้มของราคาหุ้นจริง
เมื่อใช้การจับจุดของ windows แบบ rectangular
ค่าเปอร์เซ็นต์แนวโน้มที่ถูกต้อง 61.82%



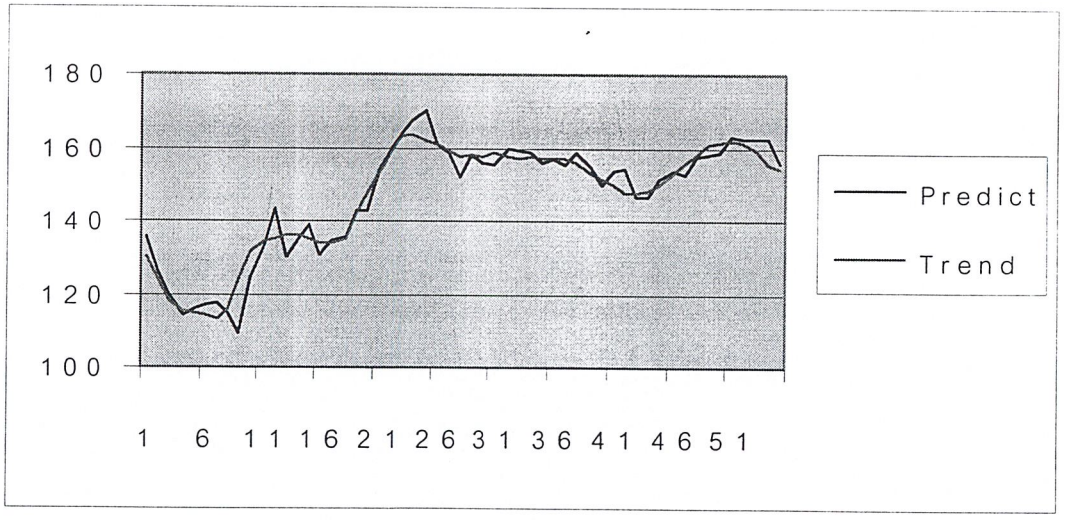
รูปที่ 6.5 กราฟแสดงราคาหุ้นที่พยากรณ์ได้เปรียบเทียบกับราคาจริงเมื่อใช้
การจับจุดของ windows แบบ optimize



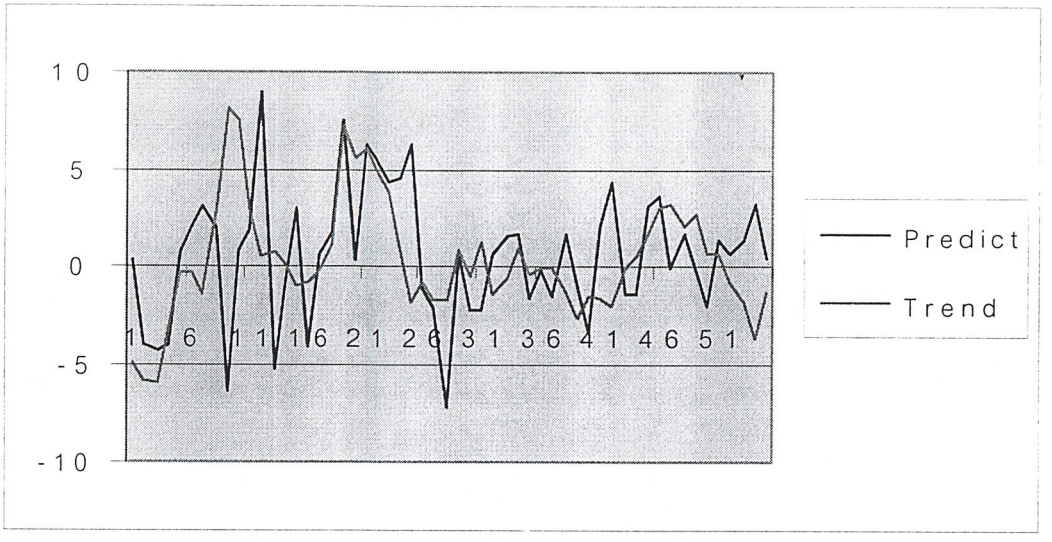
รูปที่ 6.6 กราฟแสดงแนวโน้มของราคาหุ้นที่พยากรณ์ได้เปรียบเทียบกับแนวโน้มของราคาหุ้นจริง
เมื่อใช้การจับจุดของ windows แบบ optimize
ค่าเปอร์เซ็นต์แนวโน้มที่ถูกต้อง 81.05%

การทดลองที่ 6.1.2 ผลของจำนวนนิวรอนที่ใช้ในการวิเคราะห์

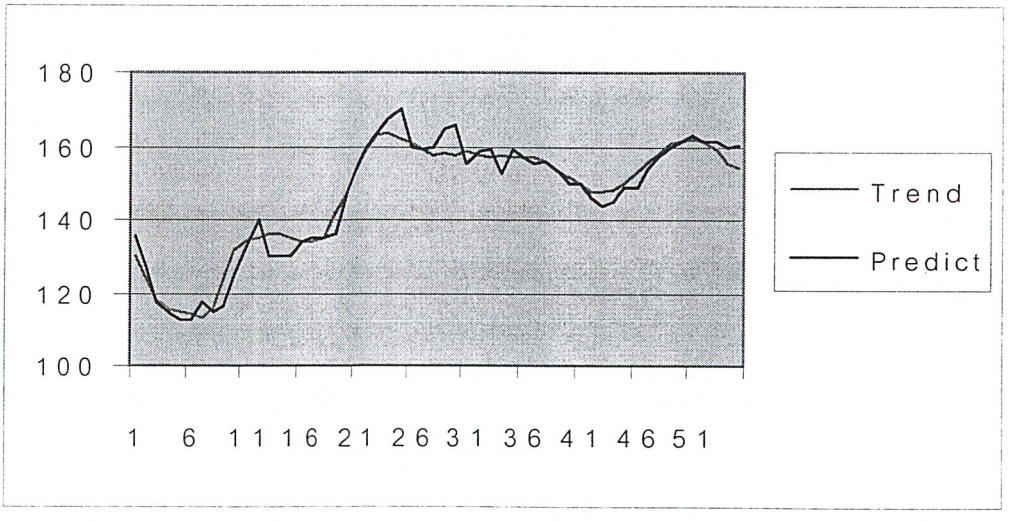
การทดลองตอนนี้จะทำการเปลี่ยนจำนวนนิวรอนที่ใช้ในการวิเคราะห์กราฟแนวโน้มราคาหุ้น และกราฟราคาหุ้น เพื่อทดลองว่าจะมีผลต่อค่าคำตอบ (output) มากน้อยเพียงใด โดยการทดลองนี้จะให้นิวรอนแต่ละตัวทำการจับข้อมูลจำนวน 20 จุด และใช้ขนาดกรอบอ้างอิง (Frame) 10 วัน และตำแหน่งการจับของจุดของนิวรอนใน windows เป็นแบบ optimize ซึ่งจะได้ผลการทดลองดังกราฟที่ 6.7 ถึง 6.10 ในส่วนของการใช้จำนวนนิวรอน 100 ตัว จะได้ค่าเปอร์เซ็นต์แนวโน้มที่ถูกต้องเป็น 81.05% ซึ่งได้กราฟเช่นเดียวกับรูปที่ 6.5 และ 6.6



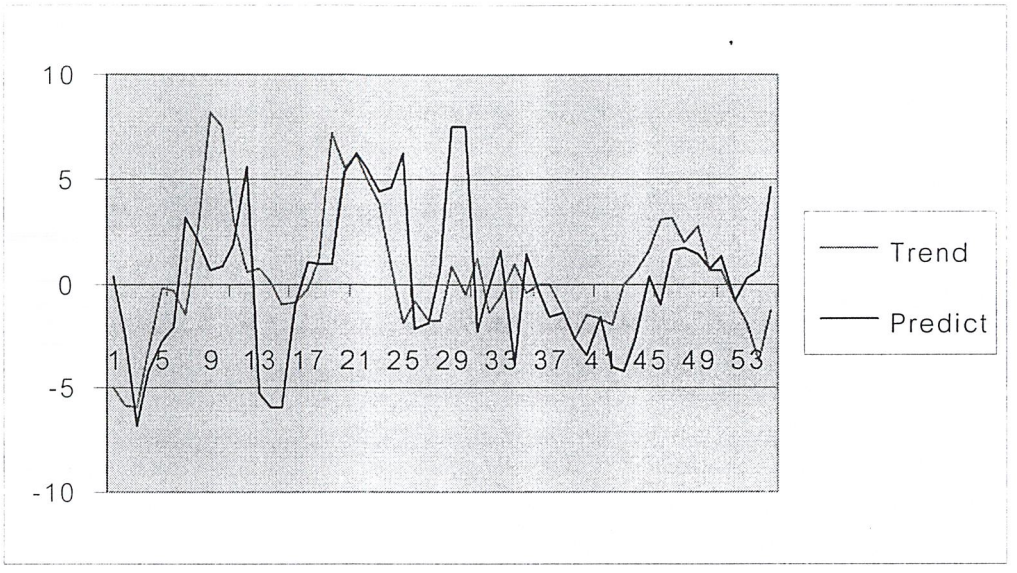
รูปที่ 6.7 กราฟแสดงราคาหุ้นที่พยากรณ์ได้เปรียบเทียบกับราคาจริงเมื่อใช้จำนวนนิวรอน 50ตัว



รูปที่ 6.8 กราฟแสดงแนวโน้มของราคาหุ้นที่พยากรณ์ได้เปรียบเทียบกับแนวโน้มของราคาหุ้นจริงเมื่อใช้จำนวนนิวรอน 50 ตัว
ค่าเปอร์เซ็นต์แนวโน้มที่ถูกต้อง 80.00%



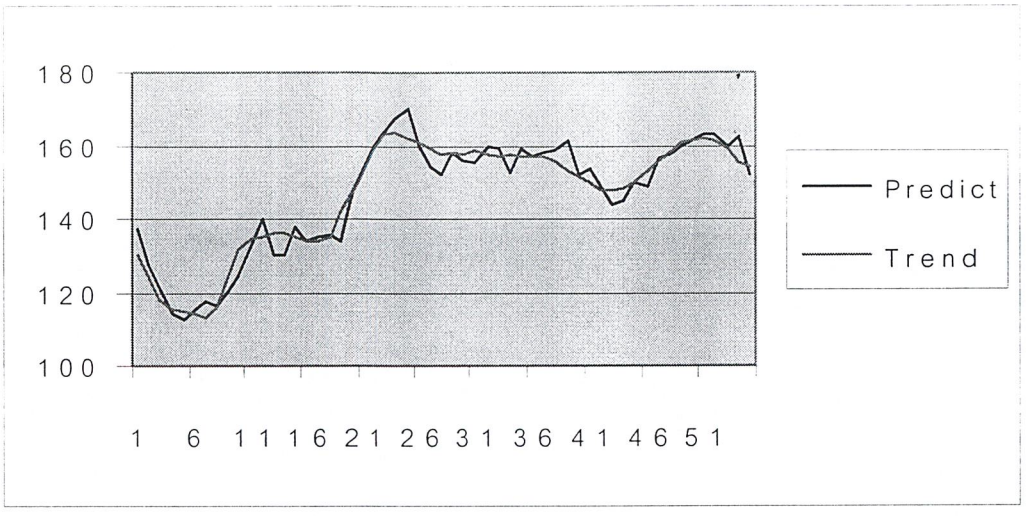
รูปที่ 6.9 กราฟแสดงราคาหุ้นที่พยากรณ์ได้เปรียบเทียบกับราคาจริงเมื่อใช้จำนวนนิวรอน 135 ตัว



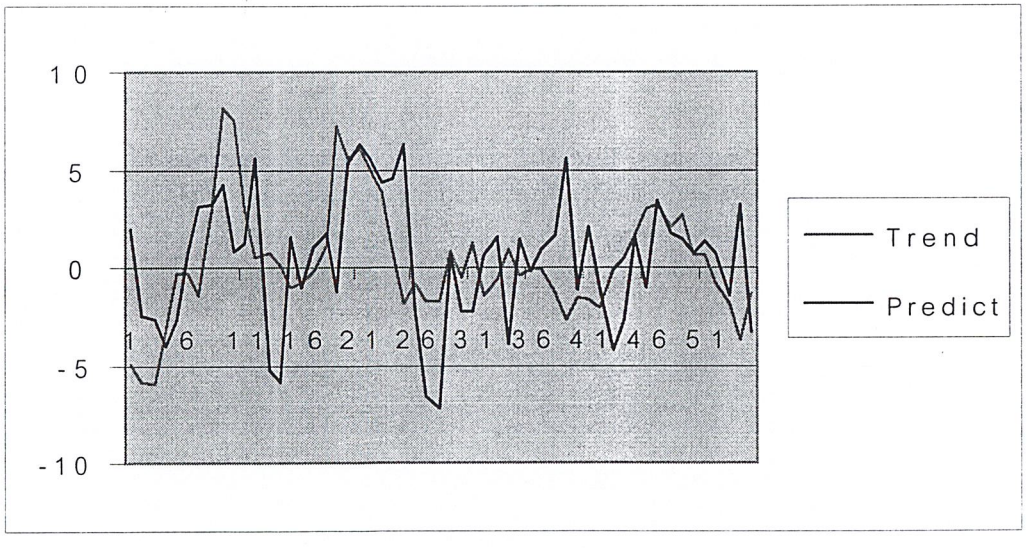
รูปที่ 6.10 กราฟแสดงแนวโน้มของราคาหุ้นที่พยากรณ์ได้เปรียบเทียบกับแนวโน้มของราคาหุ้นจริงเมื่อใช้จำนวนนิวรอน 135 ตัว
ค่าเปอร์เซ็นต์แนวโน้มที่ถูกต้อง 82.11%

การทดลองที่ 6.1.3 ผลของจำนวนตำแหน่งที่นิวรอนแต่ละตัวจับ

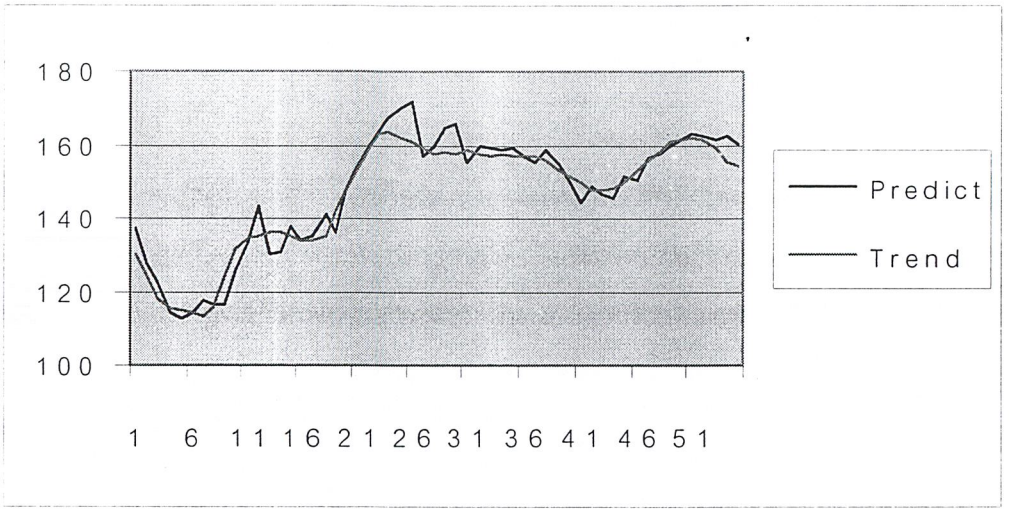
การทดลองตอนนี้จะทำการเปลี่ยนจำนวนตำแหน่งที่นิวรอนแต่ละตัวจับที่ใช้ในการวิเคราะห์กราฟแนวโน้มราคาหุ้น และกราฟราคาหุ้น เพื่อทดลองว่าจะมีผลต่อค่าคำตอบ (output) ที่ได้จากการวิเคราะห์มากน้อยเพียงใด โดยการทดลองนี้จะใช้นิวรอนทั้งหมด 100ตัว และใช้ขนาดกรอบอ้างอิง (Frame) 10 วัน และตำแหน่งการจับของจุดของนิวรอนใน windows เป็นแบบ optimize ซึ่งจะได้ผลการทดลองดังกราฟที่ 6.11 ถึง 6.14 ซึ่งในส่วนของการใช้จำนวนตำแหน่งของนิวรอนแต่ละตัวจับ 20 จุด จะได้ค่าเปอร์เซ็นต์แนวโน้มที่ถูกต้องเป็น 81.05% ซึ่งได้กราฟเช่นเดียวกับรูปที่ 6.5 และ 6.6



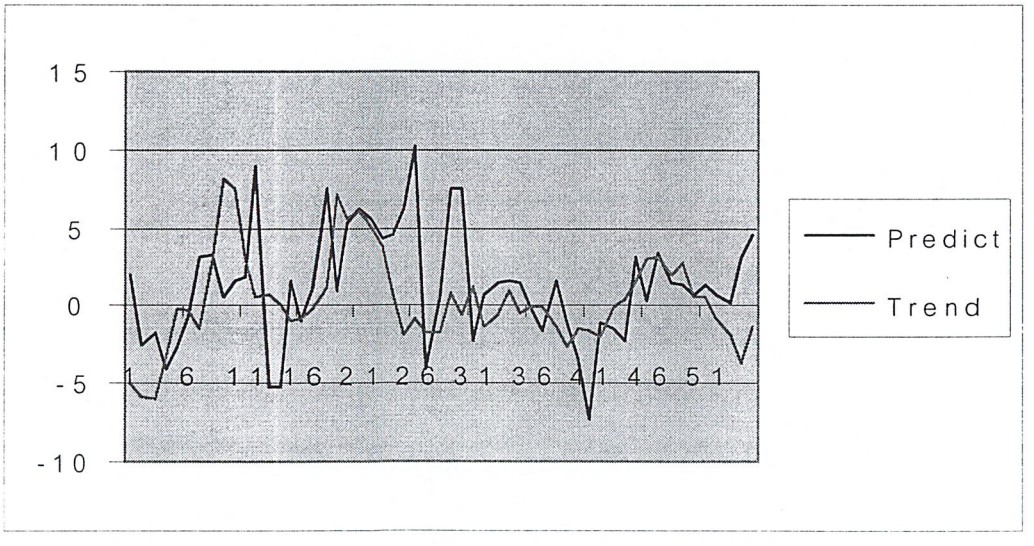
รูปที่ 6.11 กราฟแสดงราคาหุ้นที่พยากรณ์ได้เปรียบเทียบกับราคาจริงเมื่อใช้จำนวนตำแหน่งที่นิวรอนแต่ละตัวจับ 15 จุด



รูปที่ 6.12 กราฟแสดงแนวโน้มของราคาหุ้นที่พยากรณ์ได้เปรียบเทียบกับแนวโน้มของราคาหุ้นจริงเมื่อใช้จำนวนตำแหน่งที่นิวรอนแต่ละตัวจับเป็น 15 จุด
ค่าเปอร์เซ็นต์แนวโน้มที่ถูกต้อง 73.68%



รูปที่ 6.13 กราฟแสดงราคาหุ้นที่พยากรณ์ได้เปรียบเทียบกับราคาจริงเมื่อใช้จำนวนตำแหน่งที่นิวรอนแต่ละตัวจับ 25 จุด

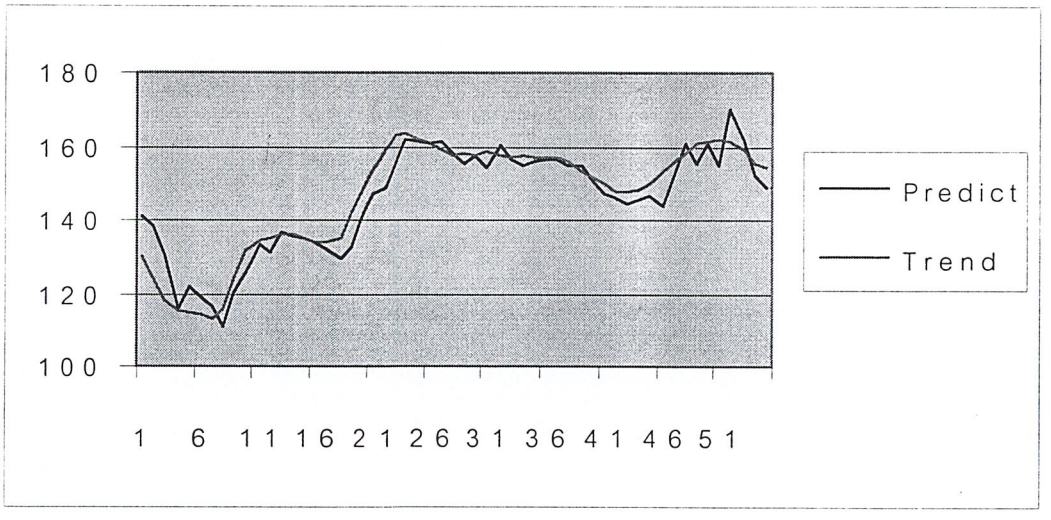


รูปที่ 6.14 กราฟแสดงแนวโน้มของราคาหุ้นที่พยากรณ์ได้เปรียบเทียบกับแนวโน้มของราคาหุ้นจริง เมื่อใช้จำนวนตำแหน่งที่นิวรอนแต่ละตัวจับเป็น 25 จุด

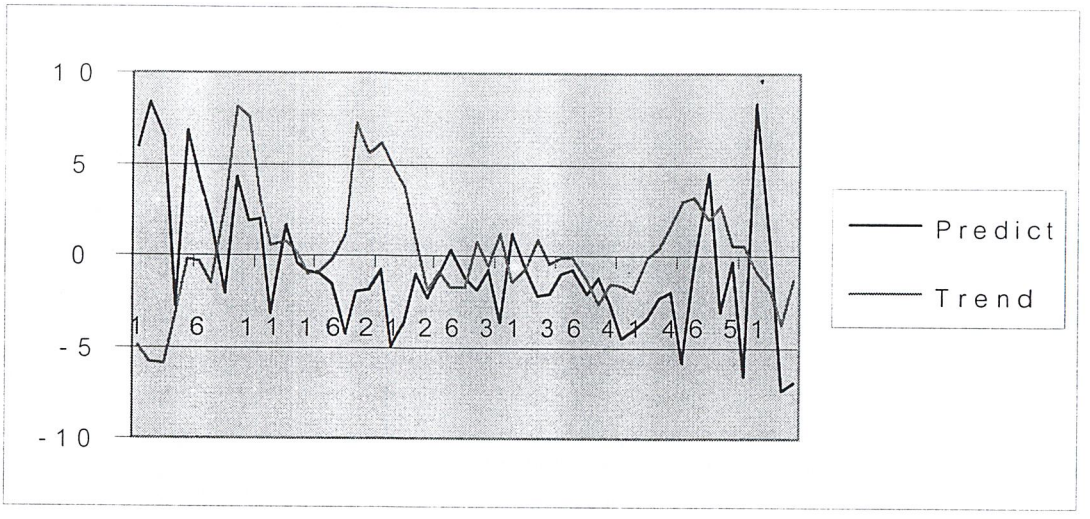
ค่าเปอร์เซ็นต์แนวโน้มที่ถูกต้อง 75.79%

การทดลองที่ 6.1.4 ผลของขนาดของกรอบอ้างอิง

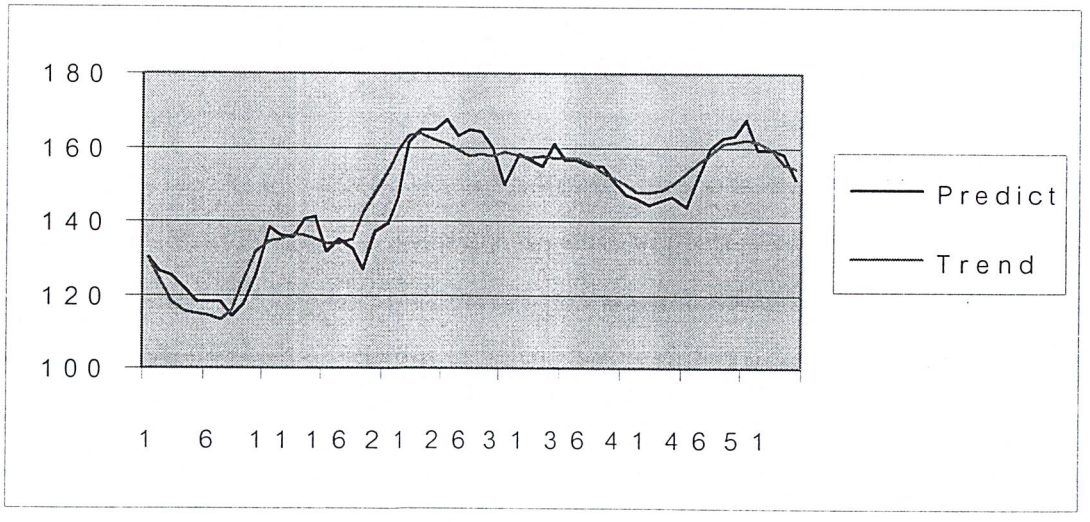
การทดลองตอนนี้จะทำการเปลี่ยนขนาดของกรอบอ้างอิงที่ใช้ในการวิเคราะห์กราฟแนวโน้มราคาหุ้น และกราฟราคาหุ้น เพื่อทดลองว่าจะมีผลต่อค่าคำตอบ (output) ที่ได้จากการวิเคราะห์มากน้อยเพียงใด โดยการทดลองนี้จะใช้นิวรอนทั้งหมด 100ตัว และใช้จำนวนตำแหน่งที่นิวรอนแต่ละตัวจับเป็น 20 จุด และตำแหน่งการจับของจุดของนิวรอนใน windows เป็นแบบ optimize ซึ่งจะได้ผลการทดลองดังกราฟที่ 6.15 ถึง 6.18 ซึ่งในส่วนของขนาดกรอบอ้าง 10 วัน นั้นจะได้ค่าเปอร์เซ็นต์แนวโน้มที่ถูกต้องเป็น 81.05% ซึ่งได้กราฟเช่นเดียวกับรูปที่ 6.5 และ 6.6



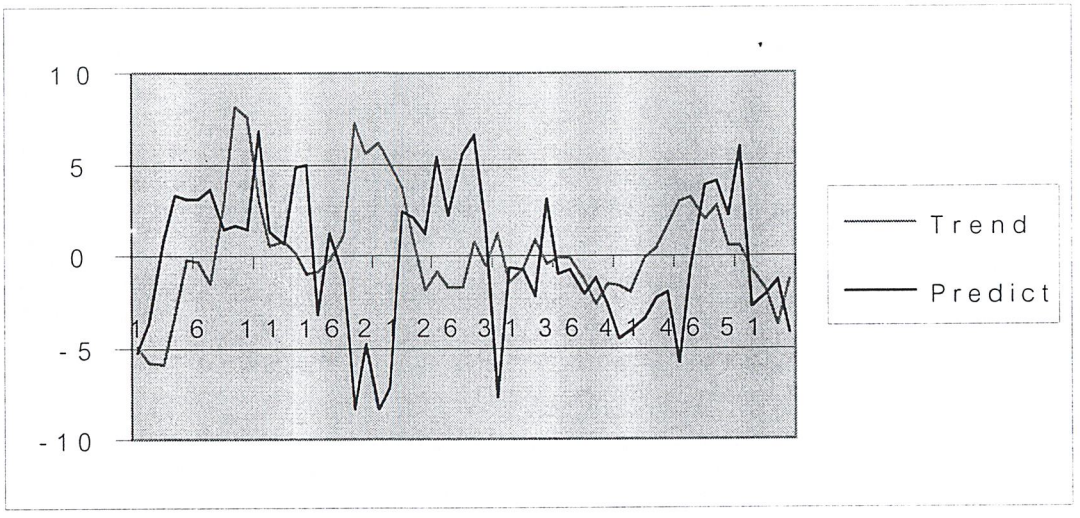
รูปที่ 6.15 กราฟแสดงราคาหุ้นที่พยากรณ์ได้เปรียบเทียบกับราคาจริงเมื่อใช้ขนาดของกรอบอ้างอิง 15 วัน



รูปที่ 6.16 กราฟแสดงแนวโน้มของราคาหุ้นที่พยากรณ์ได้เปรียบเทียบกับแนวโน้มของราคาหุ้นจริง เมื่อใช้ขนาดของกรอบอ้างอิง 15 วัน
ค่าเปอร์เซ็นต์แนวโน้มที่ถูกต้อง 71.58%



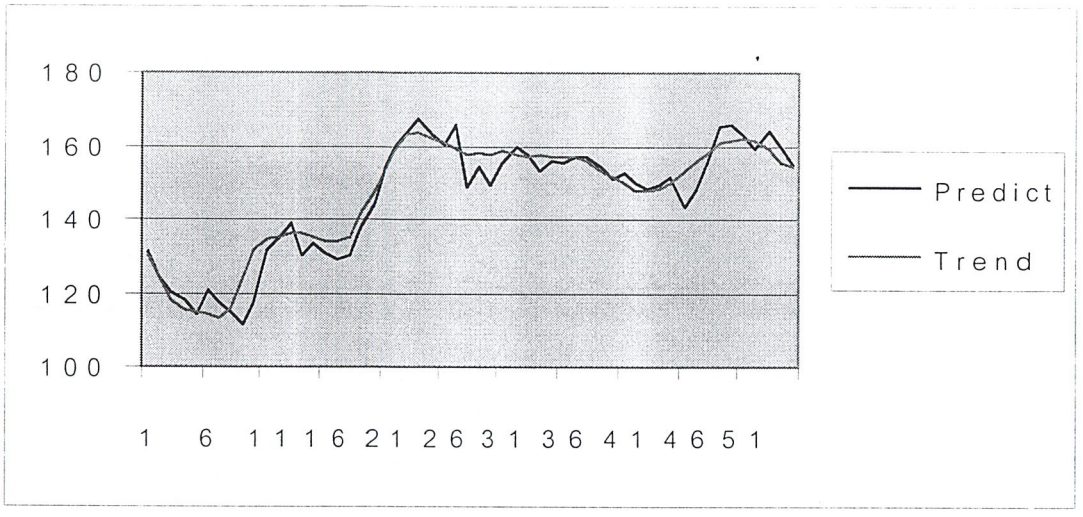
รูปที่ 6.17 กราฟแสดงราคาหุ้นที่พยากรณ์ได้เปรียบเทียบกับราคาจริงเมื่อใช้ขนาดของกรอบอ้างอิง 20 วัน



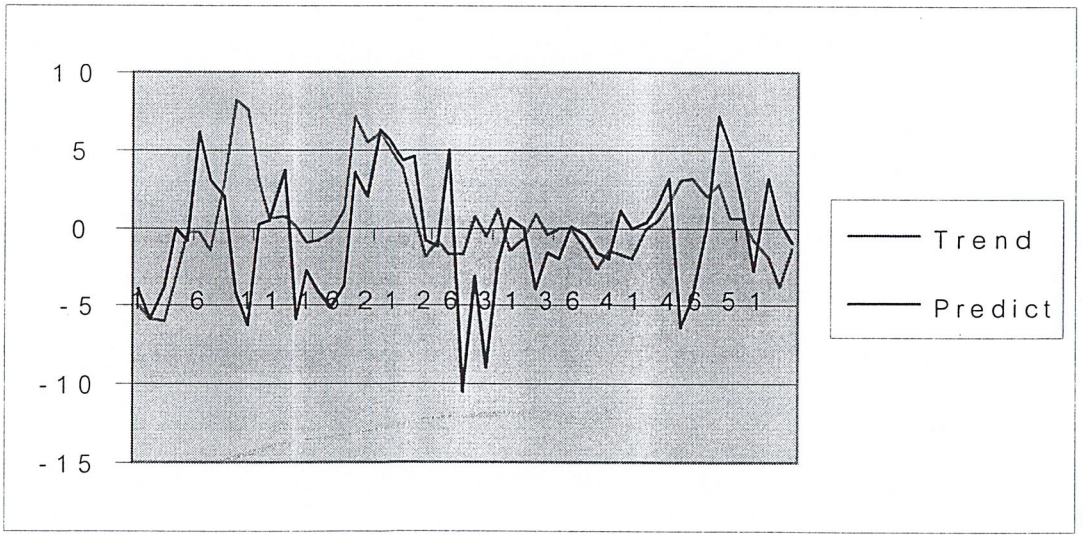
รูปที่ 6.18 กราฟแสดงแนวโน้มของราคาหุ้นที่พยากรณ์ได้เปรียบเทียบกับแนวโน้มของราคาหุ้นจริง
เมื่อใช้ขนาดของกรอบอ้างอิง 20 วัน
ค่าเปอร์เซ็นต์แนวโน้มที่ถูกต้อง 61.05%

การทดลองที่ 6.1.5 ผลของการ Normalize

การทดลองตอนนี้จะทำการเปลี่ยนขนาดของการ Normalize กราฟแนวโน้ม ที่ใช้ในการวิเคราะห์กราฟแนวโน้มราคาหุ้น และกราฟราคาหุ้น เพื่อทดลองว่าจะมีผลต่อค่าคำตอบ (output) ที่ได้จากการวิเคราะห์มากน้อยเพียงใด โดยการทดลองนี้จะใช้นิวรอนทั้งหมด 100ตัว แต่ละตัวจับ 20 จุด และใช้ขนาดกรอบอ้างอิง (Frame) 10 วัน และตำแหน่งการจับของจุดของนิวรอนใน windows เป็นแบบ optimize ซึ่งจะได้ผลการทดลองดังกราฟที่ 6.19 ถึง 6.22

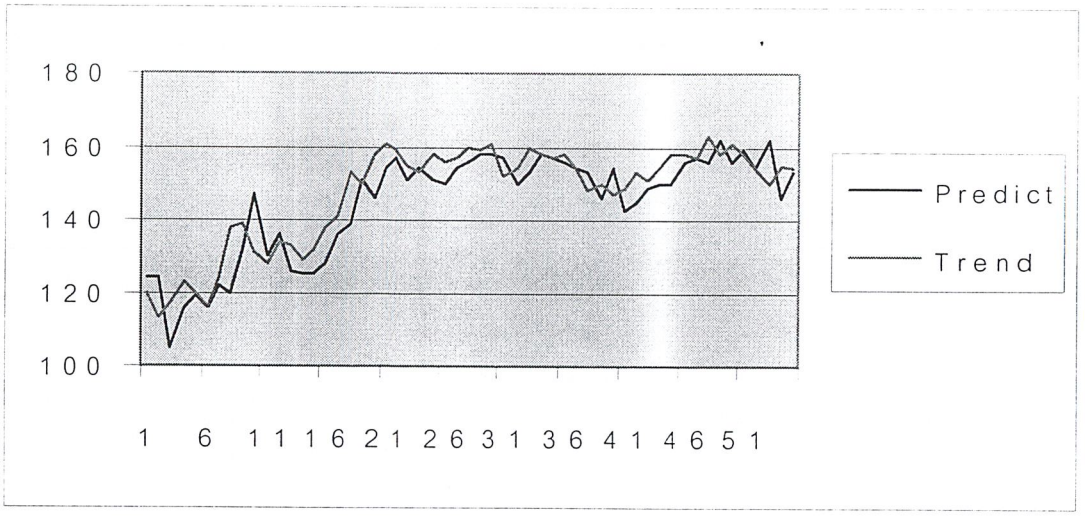


รูปที่ 6.19 กราฟแสดงราคาหุ้นที่พยากรณ์ได้เปรียบเทียบกับราคาจริงเมื่อใช้การ Normalize ข้อมูล 10 วัน

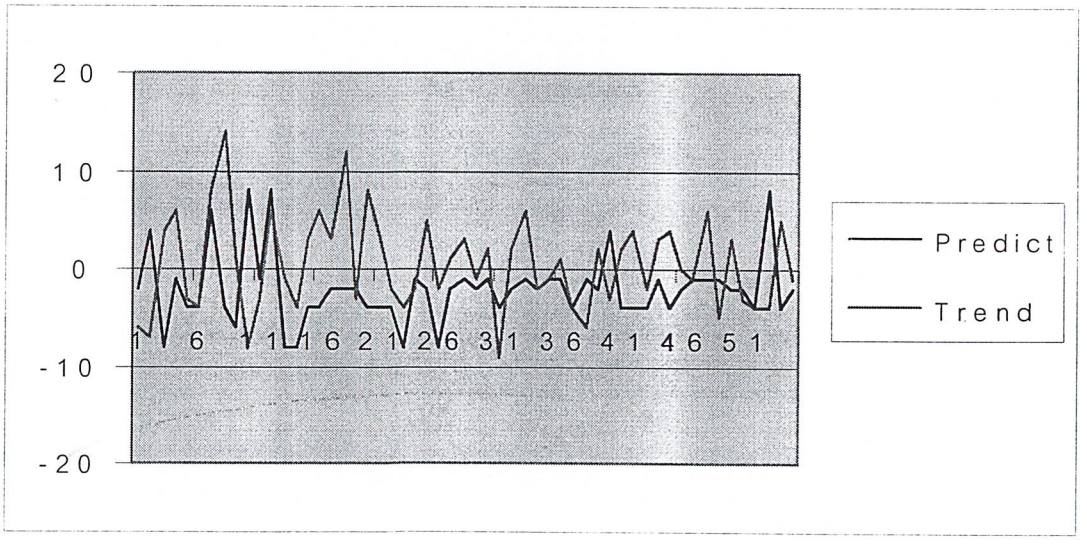


รูปที่ 6.20 กราฟแสดงแนวโน้มของราคาหุ้นที่พยากรณ์ได้เปรียบเทียบกับแนวโน้มของราคาหุ้นจริง เมื่อใช้การ Normalize ข้อมูล 10 วัน

ค่าเปอร์เซ็นต์แนวโน้มที่ถูกต้อง 81.05%



รูปที่ 6.21 กราฟแสดงราคาหุ้นที่พยากรณ์ได้เปรียบเทียบกับราคาจริงเมื่อไม่ใช้
การ Normalize ข้อมูล



รูปที่ 6.22 กราฟแสดงแนวโน้มของราคาหุ้นที่พยากรณ์ได้เปรียบเทียบกับแนวโน้มของราคาหุ้นจริง
เมื่อไม่ใช้การ Normalize ข้อมูล
ค่าเปอร์เซ็นต์แนวโน้มที่ถูกต้อง 45.45%

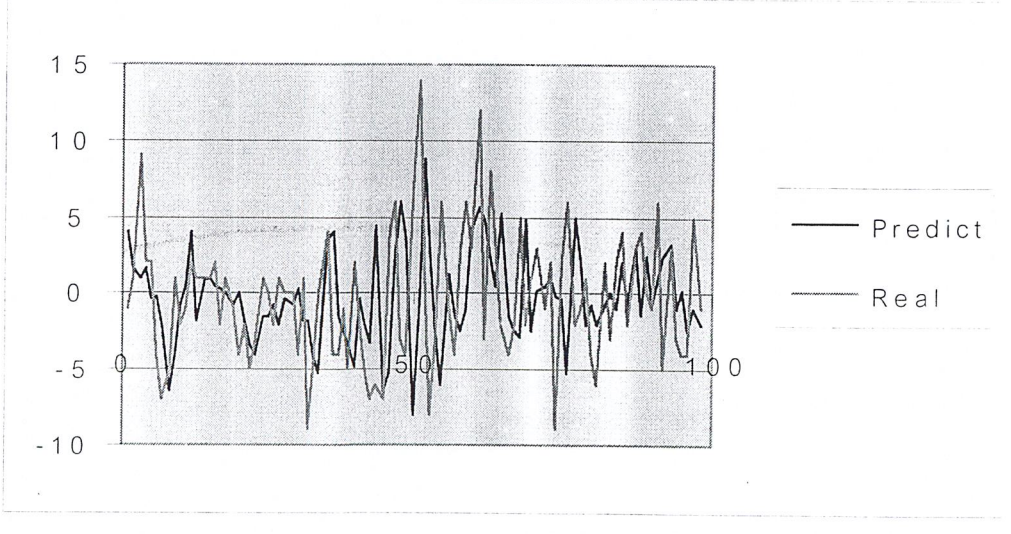
6.2 การพยากรณ์โดยข้อมูลถูกนำมาหาแนวโน้มก่อนผ่านการNormalize โดยเอาท์พุทเป็นแบบไม่ระบุตำแหน่งแน่นอน

<u>การทดลองที่ 6.2.1</u> ผลของการ เปลี่ยนค่าแนวโน้ม

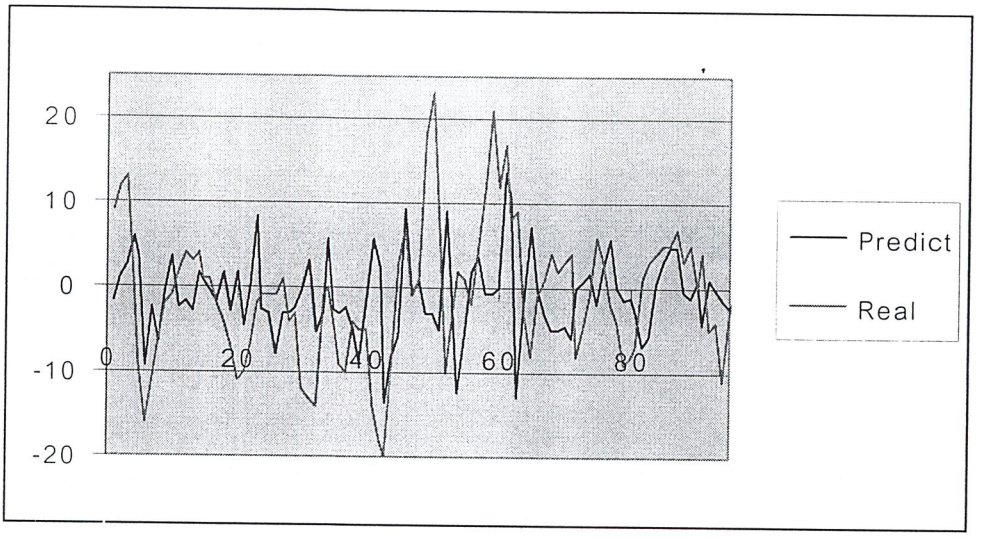
การทดลองในหัวข้อนี้จะใช้ส่วนของโปรแกรมในหัวข้อที่ 5.2 ซึ่งในการทดลองนี้จะมีการหาค่าแนวโน้มก่อน ซึ่งค่าแนวโน้มนี้เราจะมีการเปลี่ยนทั้งหมด 3 รูปแบบ ก็คือ

1. เมื่อค่าแนวโน้มได้จากผลต่างของข้อมูลวันถัดไปกับวันปัจจุบัน (delta = 1)
2. เมื่อค่าแนวโน้มได้จากผลต่างของข้อมูล 2 วันถัดไปกับวันปัจจุบัน (delta = 3)
3. เมื่อค่าแนวโน้มได้จากผลต่างของข้อมูล 4 วันถัดไปกับวันปัจจุบัน (delta = 5)

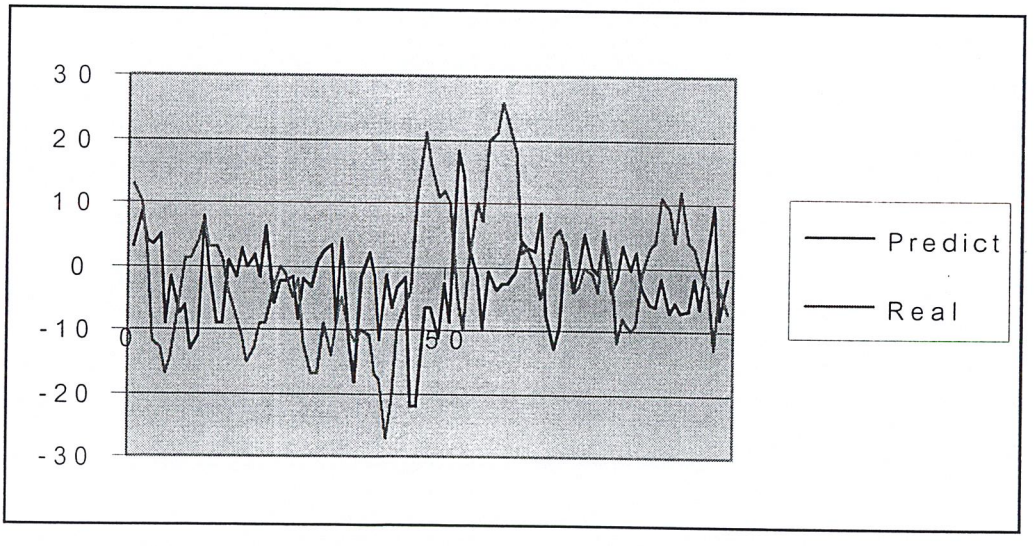
นั่นคือเราจะทำการศึกษาผลของ delta นั่นเอง แล้วจึงทำการ Normalize ในการทดลองนี้จะใช้ขนาดกรอบไบนารีเป็น 10 วัน และจะใช้จำนวนนิวรอนทั้งหมด 100 ตัว ให้นิวรอนแต่ละตัวจับ 20 จุด และตำแหน่งการจับของจุดของนิวรอนใน windows เป็นแบบ optimize ซึ่งในการทดลองส่วนนี้จะเป็นการเปรียบเทียบค่าที่ได้จากการพยากรณ์กับค่าจริงที่ไม่ผ่านการ Normalize ซึ่งจะได้ผลการทดลองดังกราฟที่ 6.23 ถึง 6.25



รูปที่ 6.23 กราฟแสดงแนวโน้มของราคาหุ้นที่พยากรณ์ได้เมื่อ delta = 1 เปรียบเทียบกับแนวโน้มของราคาหุ้นจริงที่ไม่มีการ Normalize ข้อมูล
ค่าเปอร์เซ็นต์แนวโน้มที่ถูกต้อง <u>52.04%</u>



รูปที่ 6.24 กราฟแสดงแนวโน้มของราคาหุ้นที่พยากรณ์ได้เมื่อ delta = 3 เปรียบเทียบกับแนวโน้มของราคาหุ้นจริงที่ไม่มีการ Normalize ข้อมูล
ค่าเปอร์เซ็นต์แนวโน้มที่ถูกต้อง 51.02%



รูปที่ 6.25 กราฟแสดงแนวโน้มของราคาหุ้นที่พยากรณ์ได้เมื่อ delta = 5 เปรียบเทียบกับแนวโน้มของราคาหุ้นจริงที่ไม่มีการ Normalize ข้อมูล
ค่าเปอร์เซ็นต์แนวโน้มที่ถูกต้อง 48.94%

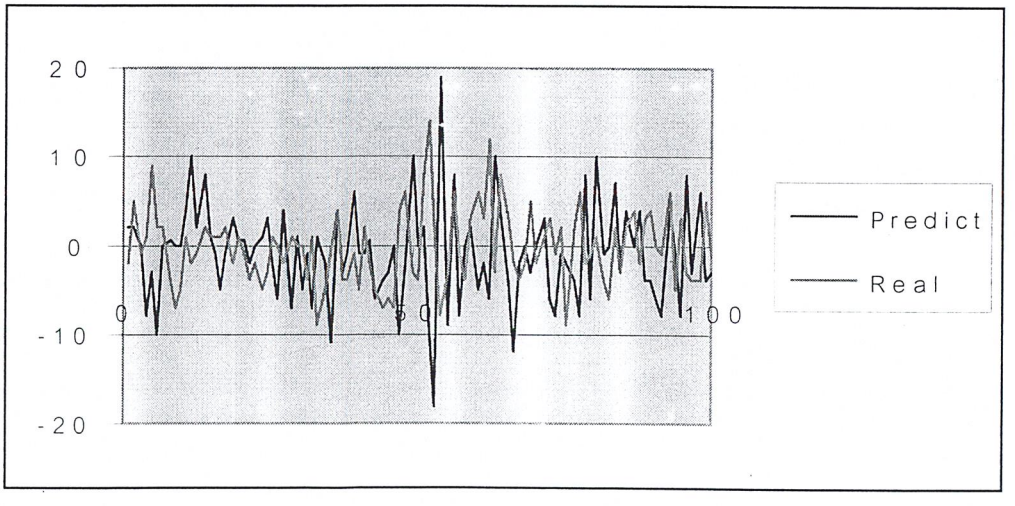
6.3 การพยากรณ์โดยข้อมูลถูกนำมาหาแนวโน้มก่อนแต่ไม่ผ่านการ Normalize โดยเอาท์พุทเป็นแบบไม่ระบุตำแหน่งแน่นอน

<u>การทดลองที่ 6.3.1</u> ผลของการ เปลี่ยนค่าแนวโน้ม

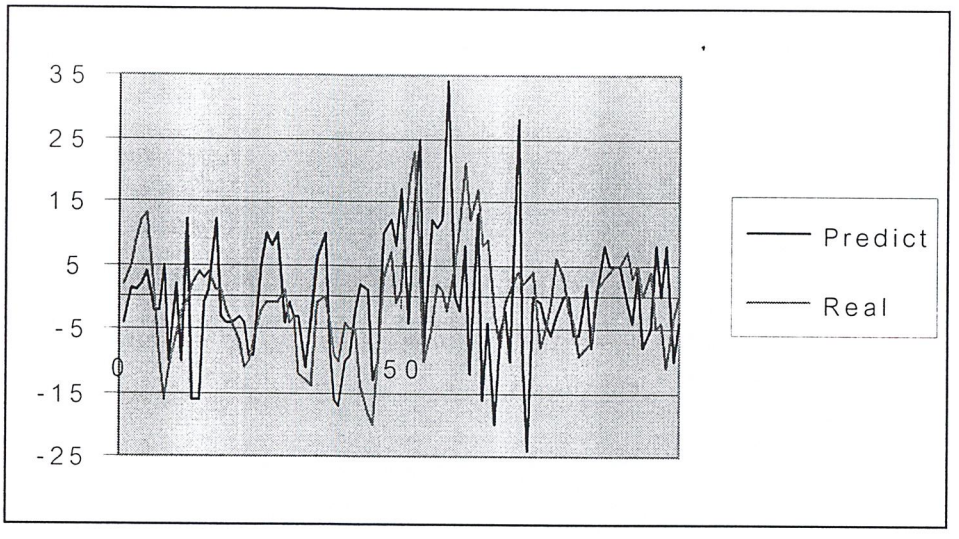
การทดลองในหัวข้อนี้จะใช้ส่วนของโปรแกรมในหัวข้อที่ 5.3 ซึ่งในการทดลองนี้จะมีการหาค่าแนวโน้มก่อน ซึ่งค่าแนวโน้มนี้เราจะมีการเปลี่ยนทั้งหมด 3 รูปแบบ ก็คือ

1. delta = 1
2. delta = 3
3. delta = 5

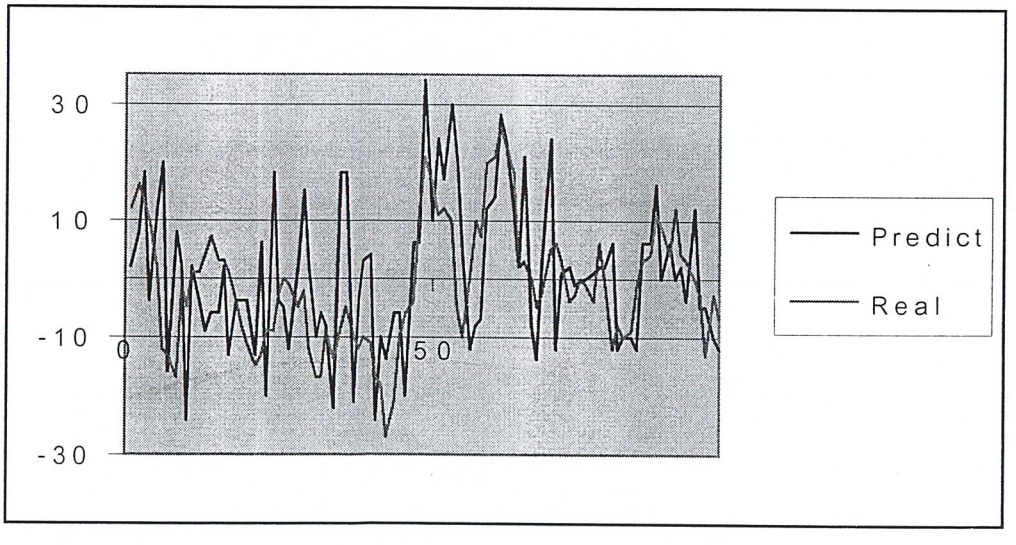
นั่นคือเราจะทำการศึกษาผลของ delta นั่นเอง แต่ไม่มีการ Normalize ข้อมูลที่ผ่านการหาแนวโน้ม ในการทดลองนี้จะใช้ขนาดกรอบไบนารีเป็น 10 วัน และจะใช้จำนวนนิวรอนทั้งหมด 100 ตัว ให้นิวรอนแต่ละตัวจับ 20 จุด และตำแหน่งการจับของจุดของนิวรอนใน windows เป็นแบบ optimize ซึ่งในการทดลองส่วนนี้จะเป็นการเปรียบเทียบค่าที่ได้จากการพยากรณ์กับค่าจริงที่ไม่ผ่านการ Normalize ซึ่งจะได้ผลการทดลองดังกราฟที่ 6.26 ถึง 6.28



รูปที่ 6.26 กราฟแสดงแนวโน้มของราคาหุ้นที่พยากรณ์ได้เมื่อ delta = 1 และไม่มีการ Normalize แนวโน้ม เปรียบเทียบกับแนวโน้มของราคาหุ้นจริงที่ไม่มีการ Normalize ข้อมูล
ค่าเปอร์เซ็นต์แนวโน้มที่ถูกต้อง <u>44.00%</u>



รูปที่ 6.27 กราฟแสดงแนวโน้มของราคาหุ้นที่พยากรณ์ได้เมื่อ delta = 3 และไม่มีการ Normalize แนวโน้ม เปรียบเทียบกับแนวโน้มของราคาหุ้นจริงที่ไม่มีการ Normalize ข้อมูล ค่าเปอร์เซ็นต์แนวโน้มที่ถูกต้อง 60.20%

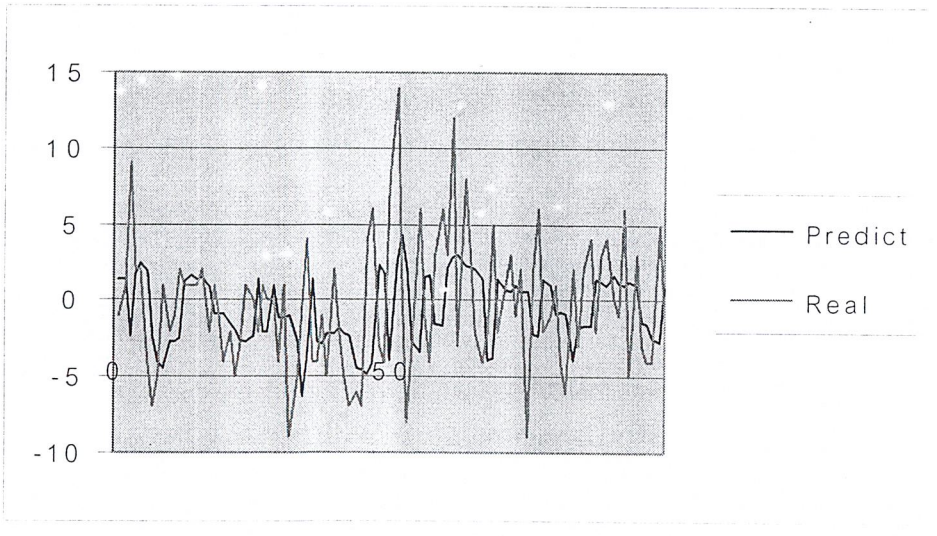


รูปที่ 6.28 กราฟแสดงแนวโน้มของราคาหุ้นที่พยากรณ์ได้เมื่อ delta = 5 และไม่มีการ Normalize แนวโน้ม เปรียบเทียบกับแนวโน้มของราคาหุ้นจริงที่ไม่มีการ Normalize ข้อมูล ค่าเปอร์เซ็นต์แนวโน้มที่ถูกต้อง 64.58%

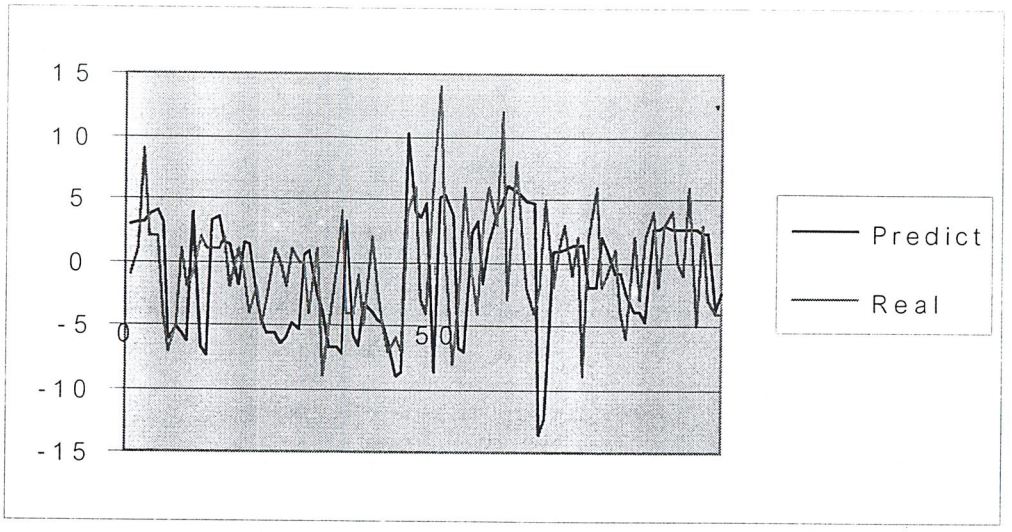
6.4 การพยากรณ์โดยข้อมูลถูกนำมาหาแนวโน้มก่อนผ่านการNormalize โดยเอาท์พุทของนิวรอนจะให้เอาท์พุทออกมาโดยไม่ผ่านการโหวต

<u>การทดลองที่ 6.4.1</u> ผลของการ เปลี่ยนค่าแนวโน้ม

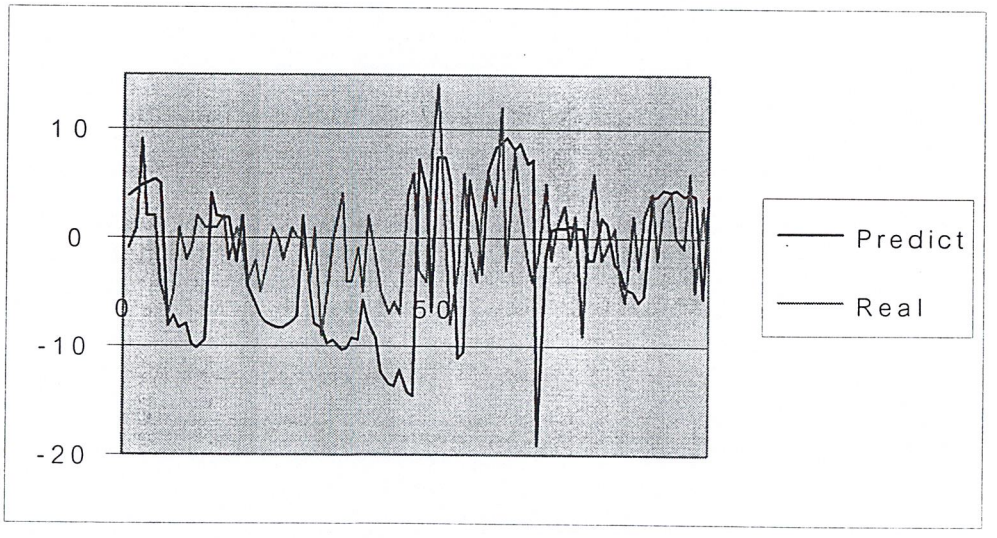
การทดลองในหัวข้อนี้จะใช้ส่วนของโปรแกรมในหัวข้อที่ 5.4 ซึ่งในการทดลองนี้จะมีการหาค่าแนวโน้มก่อน ซึ่งค่าแนวโน้มนี้เราจะมีการเปลี่ยนทั้งหมด 3 รูปแบบ ตามหัวข้อที่ 6.3 ในการทดลองนี้จะใช้ขนาดกรอบไบนารีเป็น 10 วันและจะกำหนดตำแหน่งของเอาท์พุทของนิวรอนแต่ละตัวในวันถัดไป ซึ่งอยู่ในรูปของไบนารี (0,1) และจะใช้จำนวนนิวรอนทั้งหมด 300 ตัว และให้นิวรอนแต่ละตัวจับ 7 จุด ซึ่งในการทดลองส่วนนี้จะเป็นการเปรียบเทียบค่าที่ได้จากการพยากรณ์กับค่าจริงที่ไม่ผ่านการ Normalize ซึ่งจะได้ผลการทดลองดังกราฟที่ 6.29 ถึง 6.31



รูปที่ 6.29 กราฟแสดงแนวโน้มของราคาหุ้นที่พยากรณ์ได้เมื่อ delta = 1 โดยเอาท์พุทไม่มีการโหวต เปรียบเทียบกับแนวโน้มของราคาหุ้นจริงที่ไม่มีการ Normalize ข้อมูล
ค่าเปอร์เซ็นต์แนวโน้มที่ถูกต้อง <u>51.02%</u>



รูปที่ 6.30 กราฟแสดงแนวโน้มของราคาหุ้นที่พยากรณ์ได้เมื่อ delta = 3 โดยเอาท์พุทไม่มีการโหวต เปรียบเทียบกับแนวโน้มของราคาหุ้นจริงที่ไม่มีการ Normalize ข้อมูล
ค่าเปอร์เซ็นต์แนวโน้มที่ถูกต้อง 52.08%

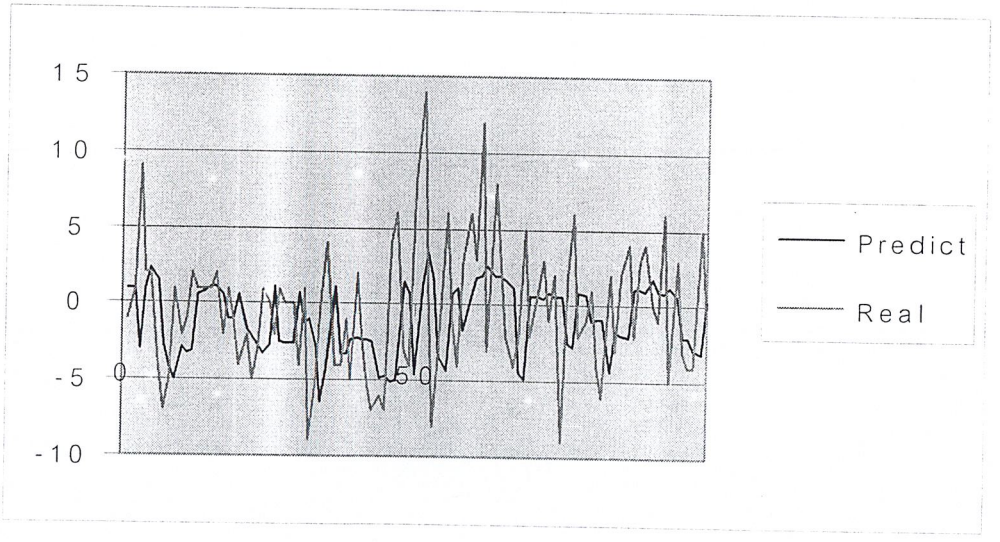


รูปที่ 6.31 กราฟแสดงแนวโน้มของราคาหุ้นที่พยากรณ์ได้เมื่อ delta = 5 โดยเอาท์พุทไม่มีการโหวต เปรียบเทียบกับแนวโน้มของราคาหุ้นจริงที่ไม่มีการ Normalize ข้อมูล
ค่าเปอร์เซ็นต์แนวโน้มที่ถูกต้อง 52.13%

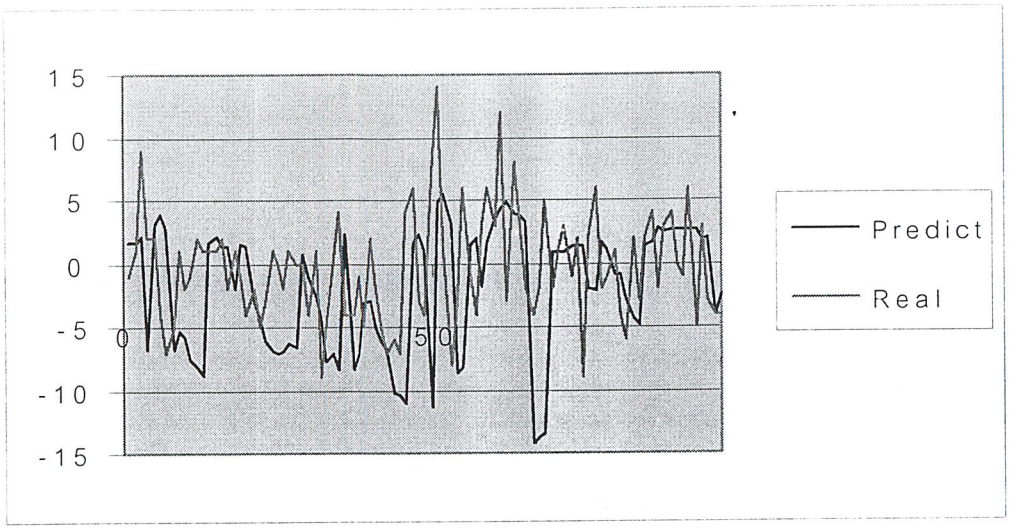
6.5 การพยากรณ์โดยข้อมูลถูกนำมาหาแนวโน้มก่อนผ่านการNormalize โดยเอาท์พุทเป็นแบบระบุตำแหน่งแน่นอน และนิวรอนแต่ละตัวจะมาจากนิวรอน 5 ตัวขนานกัน

<u>การทดลองที่ 6.5.1</u> ผลของการ เปลี่ยนค่าแนวโน้ม

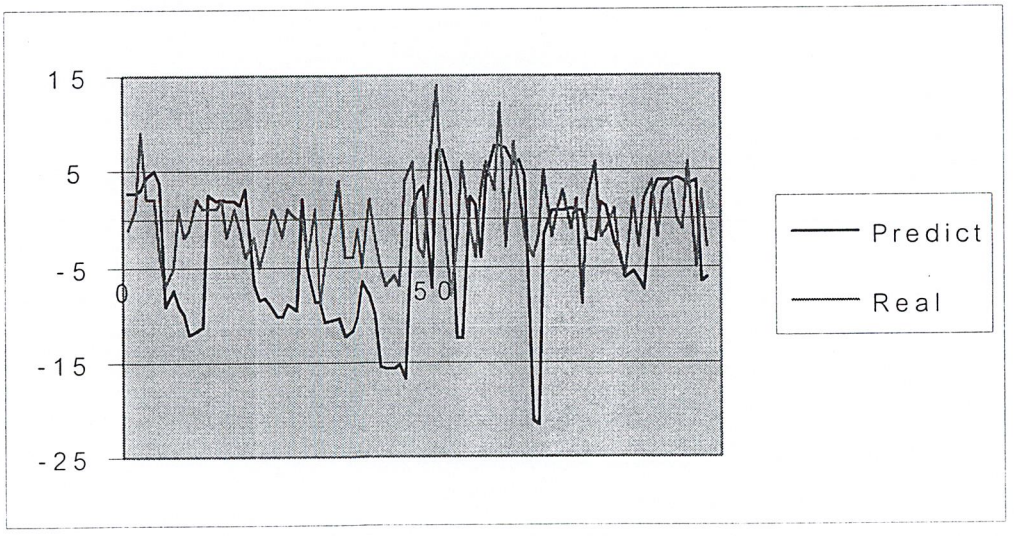
การทดลองในหัวข้อนี้จะใช้ส่วนของโปรแกรมในหัวข้อที่ 5.5 ซึ่งในการทดลองนี้จะมีการหาค่าแนวโน้มก่อน ซึ่งค่าแนวโน้มนี้เราจะมีการเปลี่ยนทั้งหมด 3 รูปแบบ และตัวแปรต่างๆ ตามหัวข้อที่ 6.4 ซึ่งจะได้ผลการทดลองดังกราฟที่ 6.32 ถึง 6.34



รูปที่ 6.32 กราฟแสดงแนวโน้มของราคาหุ้นที่พยากรณ์ได้เมื่อ delta = 1 โดยนิวรอนแต่ละตัวจะมานิวรอน 5 ตัวขนานกันเปรียบเทียบกับแนวโน้มของราคาหุ้นจริงที่ไม่มีการ Normalize ข้อมูล ค่าเปอร์เซ็นต์แนวโน้มที่ถูกต้อง <u>48.98%</u>



รูปที่ 6.33 กราฟแสดงแนวโน้มของราคาหุ้นที่พยากรณ์ได้เมื่อ delta = 3 โดยนิวรอนแต่ละตัวจะมานิวรอน 5 ตัวขนานกันเปรียบเทียบกับแนวโน้มของราคาหุ้นจริงที่ไม่มีการ Normalize ข้อมูล ค่าเปอร์เซ็นต์แนวโน้มที่ถูกต้อง 53.13%

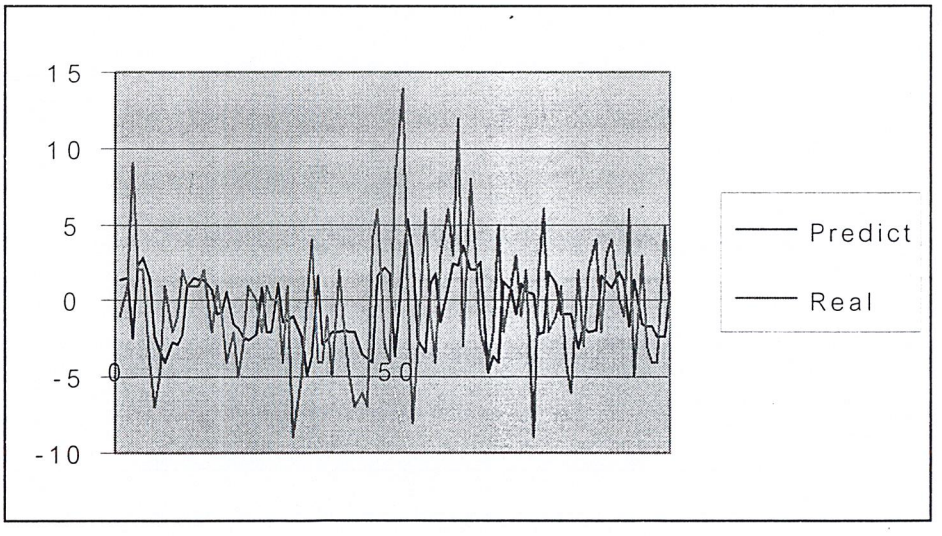


รูปที่ 6.34 กราฟแสดงแนวโน้มของราคาหุ้นที่พยากรณ์ได้เมื่อ delta = 5 โดยนิวรอนแต่ละตัวจะมานิวรอน 5 ตัวขนานกันเปรียบเทียบกับแนวโน้มของราคาหุ้นจริงที่ไม่มีการ Normalize ข้อมูล ค่าเปอร์เซ็นต์แนวโน้มที่ถูกต้อง 53.19%

6.6 การพยากรณ์โดยข้อมูลถูกนำมาหาแนวโน้มก่อนผ่านการNormalize โดยเอาท์พุทเป็นแบบระบุตำแหน่งแน่นอน แต่มีการ Feedback

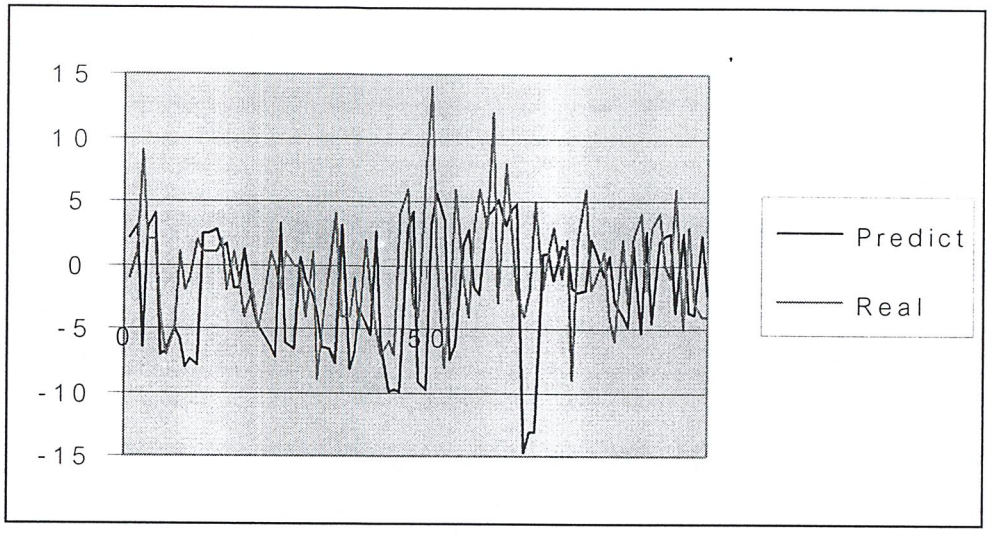
<u>การทดลองที่ 6.6.1</u> ผลของการ เปลี่ยนค่าแนวโน้ม

การทดลองในหัวข้อนี้จะใช้ส่วนของโปรแกรมในหัวข้อที่ 5.6 ซึ่งในการทดลองนี้จะมีการหาค่าแนวโน้มก่อน ซึ่งค่าแนวโน้มนี้เราจะมีการเปลี่ยนทั้งหมด 3 รูปแบบ และตัวแปรต่างๆ ตามหัวข้อที่ 6.3 ซึ่งจะได้ผลการทดลองดังกราฟที่ 6.35 ถึง 6.37

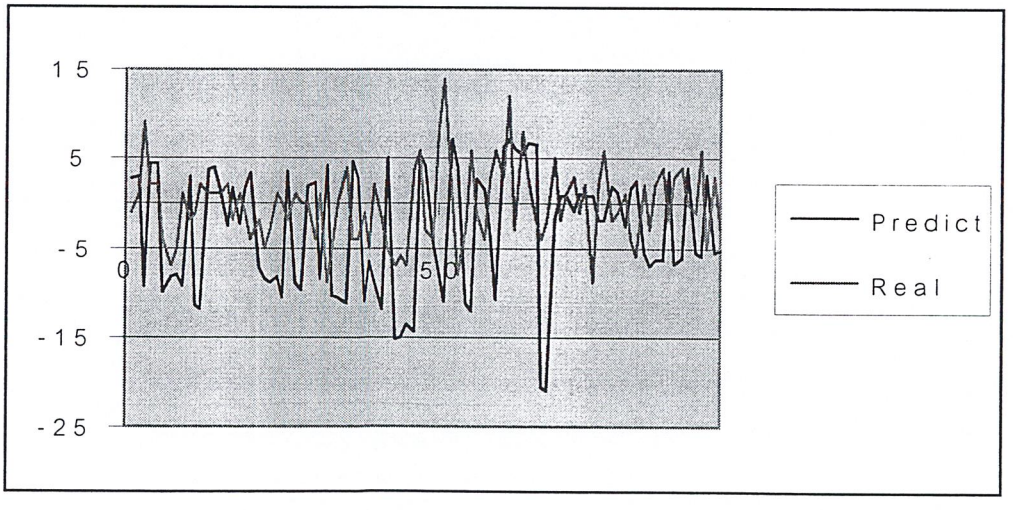


รูปที่ 6.35 กราฟแสดงแนวโน้มของราคาหุ้นที่พยากรณ์ได้เมื่อ delta = 1 และมีการFeedback เปรียบเทียบกับแนวโน้มของราคาหุ้นจริงที่ไม่มีการ Normalize ข้อมูล

ค่าเปอร์เซ็นต์แนวโน้มที่ถูกต้อง <u>50.00%</u>



รูปที่ 6.36 กราฟแสดงแนวโน้มของราคาหุ้นที่พยากรณ์ได้เมื่อ delta = 3 และมีการFeedback เปรียบเทียบกับแนวโน้มของราคาหุ้นจริงที่ไม่มีการ Normalize ข้อมูล
ค่าเปอร์เซ็นต์แนวโน้มที่ถูกต้อง 50.00%



รูปที่ 6.37 กราฟแสดงแนวโน้มของราคาหุ้นที่พยากรณ์ได้เมื่อ delta = 5 และมีการFeedback เปรียบเทียบกับแนวโน้มของราคาหุ้นจริงที่ไม่มีการ Normalize ข้อมูล
ค่าเปอร์เซ็นต์แนวโน้มที่ถูกต้อง 39.36%

บทที่ 7
สรุปและวิเคราะห์ผลการทดลอง

จากผลการทดลองโดยการปรับเปลี่ยนกระบวนการ ทั้งหมด 6 ตอน ที่มีการทำงานต่างๆ กัน และในการพยากรณ์ค่าเอาท์พุท ที่ได้จะมีความแม่นยำมากน้อยต่างกัน ซึ่งผลของกระบวนการดังกล่าวนี้มีสามารถโดยสรุปได้ดังนี้

7.1 วิเคราะห์และสรุปผลการทดลอง

7.1.1 ผลของการพยากรณ์โดยข้อมูลผ่านการNormalizeก่อนแล้วหาแนวโน้ม โดยเอาท์พุทเป็นแบบไม่ระบุตำแหน่งแน่นอน

ซึ่งผลการทดลองในตอนนี้ได้จากการปรับเปลี่ยนตัวแปรหลายๆ อย่างในการพยากรณ์ซึ่งค่าเอาท์พุทที่ได้จะวิเคราะห์ผลได้ดังนี้

7.1.1.1 ผลของการเปลี่ยนตำแหน่งการจับของจุดของนิวรอนใน windows

จากการทดลองที่ 6.1.1 รูปกราฟที่ 6.1 – 6.6 จากการเปลี่ยนตำแหน่งการจับเป็นแบบ ฮาร์โมนิก แบบ rectangular และแบบ optimize จะได้เปอร์เซ็นต์แนวโน้มที่ถูกต้องเป็น 69.07% , 61.82% และ 81.05% ตามลำดับ พบว่าการจับตำแหน่งแบบ optimize จะให้ค่าดีที่สุด เพราะว่าตำแหน่งที่ทำการจับนั้นมีข้อมูลได้มีการเปลี่ยนแปลงมากที่สุด คือระหว่าง 1 กับ 0 เพราะฉะนั้นจึงเลือกใช้วิธีของ optimize

7.1.1.2 ผลของการเปลี่ยนจำนวนของนิวรอน

จากผลการทดลองที่ 6.1.2 จะเห็นได้ว่าในค่าของราคาหุ้นที่ได้จากการพยากรณ์นั้นมีค่าใกล้เคียงกับค่าของราคาหุ้นจริง แต่ค่าแนวโน้มของราคาหุ้นที่ได้เมื่อใช้จำนวนนิวรอน 50 ตัว จะได้เปอร์เซ็นต์แนวโน้มที่ถูกต้อง 80.00% ซึ่งมีค่าน้อยกว่าเมื่อใช้จำนวนนิวรอน 100 ตัวและ 135 ตัว ที่มีค่า 81.05% และ 81.11% ตามลำดับ จะเห็นได้ว่าเมื่อเพิ่มจำนวนนิวรอนขึ้นถึงระดับหนึ่งแล้วจะไม่มีผลต่อค่าเปอร์เซ็นต์ความถูกต้อง แต่กลับจะทำให้โปรแกรมทำช้าลง เพราะฉะนั้นจึงเลือกในนิวรอนจำนวน 100 ตัว

7.1.1.3 ผลของการเปลี่ยนจำนวนตำแหน่งที่ใช้จับในแต่ละนิวรอน

จากผลการทดลองที่ 6.1.3 จากการเปลี่ยนจำนวนจุดโดยเริ่มจาก 15 จุด ได้เปอร์เซ็นต์แนวโน้มที่ถูกต้อง 73.68 % และเมื่อเปลี่ยนจำนวนจุดเป็น 20 จุด ได้เปอร์เซ็นต์แนวโน้มที่ถูกต้อง 81.05 % และเมื่อเปลี่ยนจำนวนจุดเป็น 25 จุด ได้เปอร์เซ็นต์แนวโน้มที่ถูกต้อง 75.79 % พบว่าเมื่อเพิ่มจำนวนตำแหน่งที่ใช้จับในแต่ละนิวรอนขึ้นความสามารถในการเดาจะเพิ่มขึ้น แต่เมื่อเพิ่มถึงจุดหนึ่งจะมีค่าลดลงเนื่องจากความละเอียดในการใช้เดาข้อมูลมีค่าเพิ่มขึ้น นิวรอนจึงไม่สามารถเดาค่าที่ถูกต้องได้ เพราะฉะนั้นให้นิวรอนแต่ละตัวจับ 20 จุดจะดีที่สุด

7.1.1.4 ผลของการเปลี่ยนกรอบอ้างอิง

จากผลการทดลองที่ 6.1.4 จากการเปลี่ยนจำนวนกรอบอ้างอิงจาก 10 วัน เป็น 15 วัน และ 20 วัน จะได้เปอร์เซ็นต์แนวโน้มที่ถูกต้องเป็น 81.05% , 71.58% และ 61.05% ตามลำดับ จะเห็นได้ว่าเมื่อขนาดของกรอบอ้างอิงมีค่าน้อยและว่าจำนวนจุดที่ใช้จับมีปริมาณที่พอเหมาะจึงทำให้นิวรอนพยากรณ์ข้อมูลได้ดี

7.1.1.5 ผลของการ Normalize ข้อมูล

จากผลการทดลองที่ 6.1.5 ดังรูปกราฟที่ 6.19 – 6.22 จากการเปลี่ยนจำนวนวันที่ใช้ Normalize ข้อมูลก่อนการทำเป็นแนวโน้ม ซึ่งมี Normalize ข้อมูล 10 วัน และไม่ผ่านการ Normalize เลย ได้ค่าเปอร์เซ็นต์แนวโน้มที่ถูกต้องเป็น 81.05% และ 45.45% ตามลำดับ พบว่าการ Normalize ข้อมูลแบบ 10 วัน สามารถนำมาหาความถูกต้องได้ดีกว่า ข้อมูลดิบ เนื่องจากข้อมูลดิบจะมี noise มารบกวน

7.1.2 ผลของการพยากรณ์โดยข้อมูลถูกนำมาหาแนวโน้มก่อนผ่านการ Normalize โดยเอาท์พุทเป็นแบบไม่ระบุตำแหน่งแน่นอน

ซึ่งผลการทดลองในตอนนี้ที่ได้จากการเปลี่ยนค่าแนวโน้มเป็นค่าต่างๆ ซึ่งจากการพยากรณ์เอาท์พุทที่ได้ เมื่อเปรียบเทียบกับค่าจริงของข้อมูล ที่ไม่ผ่านการ Normalize จะวิเคราะห์ผลได้ดังนี้

7.1.2.1 ผลของการเปลี่ยนค่าแนวโน้ม

จากผลการทดลองที่ 6.2.1 ดังรูปกราฟที่ 6.23-6.25 จากการเปลี่ยนค่าแนวโน้มของข้อมูล เมื่อ delta = 1, 3, 5 ได้ค่าเปอร์เซ็นต์แนวโน้มที่ถูกต้องเป็น 52.04% , 51.02% และ 48.94% ตามลำดับ ซึ่งจะเห็นไม่ว่าเมื่อนำค่าพยากรณ์ที่ได้มาเปรียบเทียบกับข้อมูลดิบ ค่าเปอร์เซ็นต์ที่ได้มี

ค่าที่ต่ำมากกว่าการเปรียบเทียบกับข้อมูลดิบที่ผ่านการ Normalize ซึ่งแสดงให้เห็นว่าถ้าเราจะนำข้อมูลมาพยากรณ์ควรที่จะมีการจัดการกับข้อมูลดิบเสียก่อนเพื่อกำจัด noise แต่อย่างไรเราต้องเลือกวิธีการที่เหมาะสมในการกำจัด noise เพื่อที่จะไม่ทำให้ค่าข้อมูลดิบผิดเพี้ยนไปจากเดิม ในผลการทดลองที่ได้ เมื่อเรานำข้อมูลมาหาค่าแนวโน้มก่อนการทำ Normalize จะเห็นได้ว่าเมื่อเราใช้ delta = 1 จะให้ผลดีที่สุดเพราะ มีการเปลี่ยนแปลงของค่าแนวโน้มทีละน้อยๆ

7.1.3 ผลของการพยากรณ์โดยข้อมูลถูกนำมาหาแนวโน้มก่อนแต่ไม่ผ่านการ Normalize โดยเอาท์พุทเป็นแบบไม่ระบุตำแหน่งแน่นอน

ซึ่งผลการทดลองในตอนนี้ที่ได้จากการเปลี่ยนค่าแนวโน้มเป็นค่าต่างๆ ซึ่งจากการพยากรณ์เอาท์พุทที่ได้ เมื่อเปรียบเทียบกับค่าจริงของข้อมูล ที่ไม่ผ่านการ Normalize จะวิเคราะห์ผลได้ดังนี้

7.1.3.1 ผลของการเปลี่ยนค่าแนวโน้ม

จากผลการทดลองที่ 6.3.1 ดังรูปกราฟที่ 6.26-6.28 จากการเปลี่ยนค่าแนวโน้มของข้อมูล เมื่อ delta = 1, 3, 5 ได้ค่าเปอร์เซ็นต์แนวโน้มที่ถูกต้องเป็น 44.00% , 60.20% และ 64.58% ตามลำดับ ซึ่งจะเห็นไม่ว่าเมื่อนำค่าพยากรณ์ที่ได้มาเปรียบเทียบกับข้อมูลดิบ ค่าเปอร์เซ็นต์ที่ได้มีค่าที่ต่ำมากกว่าการเปรียบเทียบกับข้อมูลดิบที่ผ่านการ Normalize แต่ในส่วนนี้ผลที่ได้ก็มีเปอร์เซ็นต์สูงกว่าการทดลองที่ 6.2.1 เพราะ ข้อมูล Normalize จะทำให้ข้อมูลผิดเพี้ยนไปมากขึ้นเมื่อนำมาเทียบกับข้อมูลจริง

7.1.4 ผลของการพยากรณ์โดยข้อมูลถูกนำมาหาแนวโน้มก่อนผ่านการ Normalize โดยเอาท์พุทของนิวรอนจะให้เอาท์พุทออกมาโดยไม่ผ่านการโหวต

ซึ่งผลการทดลองในตอนนี้ที่ได้จากการเปลี่ยนค่าแนวโน้มเป็นค่าต่างๆซึ่งจากการพยากรณ์เอาท์พุทที่ได้ เมื่อเปรียบเทียบกับค่าจริงของข้อมูล ที่ไม่ผ่านการ Normalize จะวิเคราะห์ผลได้ดังนี้

7.1.4.1 ผลของการเปลี่ยนค่าแนวโน้ม

จากผลการทดลองที่ 6.4.1 ดังรูปกราฟที่ 6.29-6.31 จากการเปลี่ยนค่าแนวโน้มของข้อมูล เมื่อ delta = 1, 3, 5 ได้ค่าเปอร์เซ็นต์แนวโน้มที่ถูกต้องเป็น 51.02% , 52.08% และ 52.13% ตามลำดับ ซึ่งจะเห็นไม่ว่าเมื่อนำค่าพยากรณ์ที่ได้มาเปรียบเทียบกับข้อมูลดิบ ค่าเปอร์เซ็นต์ที่ได้มีค่าที่ต่ำมาก และเมื่อเราทำการเปลี่ยน delta ค่าเปอร์เซ็นต์ความถูกต้องที่ได้นั้นไม่ค่อยเปลี่ยนแปลง แสดงให้เห็นว่า ในการทดลองแบบนี้ ค่า delta ไม่มีผลต่อการพยากรณ์

7.1.5 ผลของการพยากรณ์โดยข้อมูลถูกนำมาหาแนวโน้มก่อนผ่านการ Normalize โดยเอาท์พุทเป็นแบบระบุตำแหน่งแน่นอน และนิวรอนแต่ละตัวจะมาจากนิวรอน 5 ตัวขนานกัน

ซึ่งผลการทดลองในตอนนี้ที่ได้จากการเปลี่ยนค่าแนวโน้มเป็นค่าต่างๆซึ่งจากการพยากรณ์เอาท์พุทที่ได้ เมื่อเปรียบเทียบกับค่าจริงของข้อมูล ที่ไม่ผ่านการ Normalize จะวิเคราะห์ผลได้ดังนี้

7.1.5.1 ผลของการเปลี่ยนค่าแนวโน้ม

จากผลการทดลองที่ 6.5.1 ดังรูปกราฟที่ 6.32-6.34 จากการเปลี่ยนค่าแนวโน้มของข้อมูล เมื่อ delta = 1, 3, 5 ได้ค่าเปอร์เซ็นต์แนวโน้มที่ถูกต้องเป็น 48.98% , 53.13% และ 53.19% ตามลำดับ ซึ่งจะเห็นไม่ว่าเมื่อนำค่าพยากรณ์ที่ได้มาเปรียบเทียบกับข้อมูลดิบ ค่าเปอร์เซ็นต์ที่ได้มีค่าที่ต่ำมาก และจะเห็นได้ว่าเมื่อเรานำนิวรอน 5 ตัวมาขนานกันเพื่อที่จะได้นิวรอนออกมา 1 ตัวก็ไม่เห็นว่าจะทำให้การพยากรณ์ดีขึ้น เพราะว่าความละเอียดในการใช้เดาข้อมูลมีค่าเพิ่มขึ้น นิวรอนจึงไม่สามารถเดาค่าที่ถูกต้องได้เพิ่มขึ้น แต่การนำนิวรอน 5 ตัวมาขนานกันจะทำให้เห็นผลของค่าแนวโน้มชัดเจนขึ้นในระดับไบนารีบิต นั่นคือจะลดผลของ noise ในการทดลองที่ 6.4.1 ได้ดีขึ้นแต่เปอร์เซ็นต์ความถูกต้องยังคงเท่าเดิม และเมื่อเราทำการเปลี่ยน delta ค่าเปอร์เซ็นต์ความถูกต้องที่ได้นั้นไม่ค่อยเปลี่ยนแปลง

7.1.6 ผลของการพยากรณ์โดยข้อมูลถูกนำมาหาแนวโน้มก่อนผ่านการ Normalize โดยเอาท์พุท เป็นแบบระบุตำแหน่งแน่นอน แต่มีการ Feedback

ซึ่งผลการทดลองในตอนนี้ที่ได้จากการเปลี่ยนค่าแนวโน้มเป็นค่าต่างๆซึ่งจากการพยากรณ์เอาท์พุทที่ได้ เมื่อเปรียบเทียบกับค่าจริงของข้อมูล ที่ไม่ผ่านการ Normalize จะวิเคราะห์ผลได้ดังนี้

7.1.6.1 ผลของการเปลี่ยนค่าแนวโน้ม

จากผลการทดลองที่ 6.6.1 ดังรูปกราฟที่ 6.35-6.37 จากการเปลี่ยนค่าแนวโน้มของข้อมูล เมื่อ delta = 1, 3, 5 ได้ค่าเปอร์เซ็นต์แนวโน้มที่ถูกต้องเป็น 50.00% , 50.00% และ 39.36% ตามลำดับ ซึ่งจะเห็นไม่ว่าเมื่อนำค่าพยากรณ์ที่ได้มาเปรียบเทียบกับข้อมูลดิบ ค่าเปอร์เซ็นต์ที่ได้มีค่าที่ต่ำมาก และเมื่อมีการ Feedback ก็ไม่เห็นว่าจะช่วยให้การพยากรณ์นั้นมีความถูกต้องมากขึ้น เพราะตำแหน่งที่นิวรอนจับเพิ่มอีก 4 จุดนั้นไม่ได้มีการเลือกจุดโดยผ่านกระบวนการ optimize จึงอาจจะมีค่า error ตอนที่มีการ Feedback กลับไปซึ่งทำให้ผลการโหวตไม่ถูกต้องมากขึ้น

7.2 ประโยชน์ที่ได้รับและการนำไปประยุกต์ใช้

จากกการที่ผู้เขียนได้สร้างและทดลองโปรแกรมพยากรณ์ข้อมูลหุ้นโดยใช้หลักกการของนิวรอลเน็ตเวิร์ก สิ่งที่ได้เรียนรู้และได้รับประโยชน์มากมายจากโครงการนี้พอสรุปได้ดังนี้

1. ได้ศึกษาค้นคว้าหาข้อมูลจากหนังสือต่างๆ ที่มีส่วนเกี่ยวข้องกับโครงงานนี้ ไม่ว่าจะเป็นโปรแกรมที่ใช้เขียน ทฤษฎีนิวรอลเน็ตเวิร์กและรูปแบบการใช้งานที่เหมาะสมของแต่ละแบบ การวิเคราะห์สัญญาณ ข้อมูลเกี่ยวกับเรื่องหุ้น และคำปรึกษาจากอาจารย์ที่ปรึกษา (ดร.ยุทธนา คิดใจเดียว)
2. ข้อดีและข้อจำกัดในการนำนิวรอลเน็ตเวิร์กมาประยุกต์ใช้ในการพยากรณ์หุ้น
3. ข้อมูลหุ้นที่มีลักษณะแปรปรวนสูง ทำให้ยากต่อการพยากรณ์โดยนิวรอลเน็ตเวิร์ก
4. ระบบการพยากรณ์ที่สามารถเรียนรู้จากข้อมูลในอดีตยิ่งมากเท่าไร ยิ่งจะทำให้ระบบมีความสามารถในการพยากรณ์มากขึ้นเท่านั้น
5. ควรมีเลือกใช้จำนวนค่าของตัวแปรให้เหมาะสมที่สุด โดยทราบได้จากการทดลอง

การนำไปประยุกต์ใช้งานนั้น เราสามารถนำความรู้ที่ได้จากการทำโครงงานนี้ไปเป็นพื้นฐานในการสร้างโปรแกรมการพยากรณ์ข้อมูลต่างๆ และนำทฤษฎีนิวรอลเน็ตเวิร์กมาประยุกต์ใช้ในชีวิตประจำวันให้เหมาะสม เพื่อที่จะช่วยในการอำนวยความสะดวกต่อมนุษย์

7.3 อุปสรรคในการทำงานและแนวทางการแก้ไข

อุปสรรคที่พบในการสร้าง คือ เริ่มแรกผู้เขียนไม่สันทัดในการใช้โปรแกรมจาวา (Java) จึงทำให้เสียเวลาในการเรียนรู้นานพอสมควร และความรู้เกี่ยวกับการซื้อขายหุ้นนั้นก็มีน้อยจึงต้องใช้เวลาในการศึกษาพอสมควร การหาข้อมูลเกี่ยวกับราคาหุ้นค่อนข้างลำบากเพราะข้อมูลส่วนใหญ่นั้นจะต้องซื้อจากตลาดหลักทรัพย์แห่งประเทศไทย และหลังจากที่ได้ใช้โปรแกรมจาวาเขียน เพื่อมาพยากรณ์นั้นในส่วนของโปรแกรมยังทำงานได้ช้ามากจึงทำให้ต้องใช้เวลานานในการคอมไพล์แต่ละครั้ง และในส่วนของการ Normalize ข้อมูลนั้นข้อมูลที่ผ่านการ Normalize จะมีค่าที่แตกต่างจากข้อมูลจริงพอสมควร จึงทำให้การพยากรณ์ที่ได้นั้นมีค่าไม่ถูกต้อง

จากปัญหาด้านการทำงานของโปรแกรมที่ทำงานได้ช้า ดังที่กล่าวมาแล้วนั้น ทำให้ผู้เขียนคิดว่าการใช้โปรแกรมอื่นที่มีความเร็วในการคอมไพล์ดีกว่า จะช่วยลดปัญหาในด้านนี้ แต่โปรแกรมจาวานั้นมีวิธีการใช้ที่เข้าใจได้ง่ายไม่ยุ่งยากมากนัก และมีความสามารถหลากหลายกว่าโปรแกรมอื่น ซึ่งตรงจุดนี้ถือเป็นข้อดีของโปรแกรม และจะต้องปรับปรุงในเรื่องของการพยากรณ์

เพื่อให้มีค่าแนวโน้มที่พยากรณ์ได้มีค่าแม่นยำมากกว่านี้ ซึ่งก็จะต้องทำการศึกษาและทดลองหานิวรอลเน็ตเวิร์กแบบอื่นๆ ที่เหมาะสมกับการพยากรณ์ข้อมูลแบบนี้ และหาวิธีการ Normalize ข้อมูลที่ดีกว่านี้ เพื่อให้คำตอบถูกต้องมากที่สุด ซึ่งจะต้องศึกษาในระดับที่สูงขึ้น

ภาคผนวก

Myframe สร้าง frame สำหรับสร้างพื้นที่แสดงผลเป็นกราฟทั้งอินพุท และเอาท์พุท

```
package feedback;
import java.awt.*;
public class MyFrame extends Frame {
        Frame f;
        MyMov Mov;
        MyMovCanvas MovCan;
        MyMovCanvasall MovCanall;
MyFrame(Frame f,MyMov Mov){
        this.f=f;
        this.Mov=Mov;
        setSize(720,520);
        f.add("Center",Mov);
        setLocation(150,150);
        f.setVisible(true);
        }
MyFrame(Frame f,MyMovCanvas MovCan){
        this.f=f;
        this.MovCan=MovCan;
        setSize(720,520);
        f.add("Center",MovCan);
        setLocation(150,150);
        f.setVisible(true);
        }
MyFrame(Frame f,MyMovCanvasall MovCanall){
        this.f=f;
        this.MovCanall=MovCanall;
        setSize(720,520);
        f.add("Center",MovCanall);
        setLocation(150,150);
        f.setVisible(true);
        }
}
```

MyMov ใช้วาดกราฟ Form1 แสดงผลเป็นกราฟข้อมูลดิบ และกราฟ Form2 0แสดงผลเป็นกราฟข้อมูลที่ผ่าน Normalize

```
//draw graph//
import java.awt.*;
public class MyMov extends Canvas {
        int j,r,ig,it,bg;
        float mov3[] = new float[j];
        float smax,smin;
public MyMov(int bg,int j,int r,float mov3[],float smax,float smin) {
        this.mov3=mov3;
        this.bg=bg;
        this.r=r;
        this.j=j;
        this.smax=smax;
        this.smin=smin;
}
public Dimension getPreferredSize() {
        return new Dimension(800,400);
}
        public void paint(Graphics g) {
                Dimension d= getSize();
                String a = new String();
                double open,open2,close,subclose,Diffy,minusy,cont,Diffx,x,y;
                int m2,s,i1,i,i2;
                i1=0;
                close = smax - smin;
                subclose = 400/close;
                cont = close*subclose;
                i1=0;
                if (bg == 0)
                g.setColor(Color.blue);
           else
                        g.setColor(Color.red);
```

```
    for (i=0;i<j-1-bg;i++){
//scale 10//
                    open  = (mov3[i++]-smin)*subclose;
                    open2 = (mov3[i--]-smin)*subclose;
                    g.drawLine(i1+46,50+(int)(cont-open),i1+47,50+(int)(cont-open2));
                    i1++;
            }
            g.setColor(Color.black);
    g.drawLine(45,50+(int)cont,i1+47,50+(int)cont);
    g.drawLine(i1+47,50+(int)cont,i1+47,40);
    g.drawLine(i1+47,40,45,40);
    g.drawLine(45,40,45,50+(int)cont);
    g.drawString(""+((bg*9)/10+1),45,50+(int)cont+15);          //Draw First Date Number//
    i=0;i2=(j-bg)/5;
    for(s=1;s<6;s++) {
//Draw Date Number && Line//
              i=i+i2;
              if (s<5)
                    g.drawLine(i+45,50+(int)cont,i+45,40);
            g.drawString(""+i,i+35,50+(int)cont+15);
            }
    for(i=0;i<10;i++)
//Draw -- Line//
            g.drawLine(45,50+(int)(i*400)/10,i1+47,50+(int)(i*400)/10);
            i2 = (int)(close*subclose)/10;
    minusy = 0;
            for(i=0;i<=10;i++){
                    //Draw Stock Value//
                    Diffy = smax-minusy;
                    minusy = (close/10)*(i+1);
                    s = (int)(Diffy*100);
                    Diffy = (double)s/100;
                    g.drawString(""+Diffy,5,50+(i*i2));
                    }
```

```
Diffy = smax;
        //Show Max//
s = (int)(Diffy*100);
        Diffy = (double)s/100;
        g.drawString(a+" Max  "+Diffy,150,35);
        Diffy = smin;
                //Show Min//
        s = (int)(Diffy*100);
        Diffy = (double)s/100;
        g.drawString("  Min  "+Diffy,300,35);
        Diffy = close;
                //Show Close//
        s = (int)(Diffy*100);
        Diffy = (double)s/100;
        g.drawString("  Close  "+Diffy,450,35);
        }
}
```

MyMovCanvas ใช้วาดกราฟ Form3 แสดงผลเป็นกราฟเปรียบเทียบระหว่าง กราฟข้อมูลดิบ และข้อมูล Normalize

```
package feedback;
//draw graph//
import java.awt.*;
public class MyMovCanvas extends Canvas {
        int j,r,ig,it;
        float da[] = new float[j];
        float mov3[] = new float[j];
        float rsmax,rsmin,smax,smin;
MyMovCanvas(int j,int r,float da[],float mov3[],float smax,float smin,float rsmax,float rsmin){
        this.da=da;
        this.r=r;
        this.j=j;
        this.mov3=mov3;
        this.smax=smax;
        this.smin=smin;
        this.rsmax=rsmax;
        this.rsmin=rsmin;
}
public Dimension getPreferredSize() {
        return new Dimension(800,400);
}
        public void paint(Graphics g) {
                Dimension d= getSize();
                double open,open2,close,subclose,Diffy,minusy,cont,Diffx;
                int s,i1,i2,i;
                double zcmax,zcmin;
                double mov1[] = new double[j];
                double mov2[] = new double[j];
           if (smax>rsmax) zcmax=smax;
           else zcmax=rsmax;
           if (smin<rsmin) zcmin=smin;
```

```
    else zcmin=rsmin;
            close = zcmax - zcmin;
            subclose = 400/close;
            cont = close*subclose;
            i1=0;
            g.setColor(Color.red);
            s=0;
    for (i=0;i<j-11;i++){
//scale 10//
                    open  = (mov3[i++]-zcmin)*subclose;
                    open2 = (mov3[i--]-zcmin)*subclose;
                    g.drawLine(i1+46+2,50+(int)(cont-open),i1+47+2,50+(int)(cont-open2));
                    i1++;
                    s++;
            }
    i1=0;
    g.setColor(Color.blue);
    s=0;
    for (i=0;i<j-1;i++){
   //scale 10//
                    open  = (da[i++]-zcmin)*subclose;
                    open2 = (da[i--]-zcmin)*subclose;
                    g.drawLine(i1+46,50+(int)(cont-open),i1+47,50+(int)(cont-open2));
                    i1++;
                    s++;
    }
    g.setColor(Color.black);
    g.drawLine(45,50+(int)cont,i1+47,50+(int)cont);
    g.drawLine(i1+47,50+(int)cont,i1+47,40);
    g.drawLine(i1+47,40,45,40);
    g.drawLine(45,40,45,50+(int)cont);
    i2=j/5;

    g.drawString(""+(ig+1),45,50+(int)cont+15);                    //Draw First Date Number//
```

```
i=0;
for(s=1;s<6;s++) {
//Draw Date Number && Line//
        i=i+i2;
        if (s<5)
                g.drawLine(i+45,50+(int)cont,i+45,40);
        g.drawString(""+(ig+i-1),i+35,50+(int)cont+15);
        }

for(i=0;i<10;i++)
        //Draw -- Line//
        g.drawLine(45,50+(int)(i*400)/10,i1+47,50+(int)(i*400)/10);

        i2 = (int)(close*subclose)/10;
minusy = 0;

        for(i=0;i<=10;i++){
                //Draw Stock Value//
                Diffy = zcmax-minusy;
                minusy = (close/10)*(i+1);
                s = (int)(Diffy*100);
                Diffy = (double)s/100;
                g.drawString(""+Diffy,5,50+(i*i2));
                }

Diffy = zcmax;
        //Show Max//
s = (int)(Diffy*100);
        Diffy = (double)s/100;
        g.drawString("  Max   "+Diffy,150,35);

        Diffy = zcmin;
                //Show Min//
        s = (int)(Diffy*100);
```

```
                Diffy = (double)s/100;
                g.drawString("   Min   "+Diffy,300,35);

                Diffy = close;
                        //Show Close//
                s = (int)(Diffy*100);
                Diffy = (double)s/100;
                g.drawString("   Close   "+Diffy,450,35);
        }
}
```

MyMovCanvasall ใช้แสดงผล Form4 แสดงผลเอาท์พุทจริงกับเอาท์พุทที่มา
จากนิวรอนเนตเวิร์ก

```
package feedback;
import java.awt.*;
public class MyMovCanvasall extends Canvas{
        int j,r,ig,it;
        float da[] = new float[j];
        float mov3[] = new float[j];
        float rsmax,rsmin,smax,smin;
MyMovCanvasall(int j,int r,float da[],float mov3[],float smax,float smin,float rsmax,float rsmin){
        this.da=da;
        this.r=r;
        this.j=j;
        this.mov3=mov3;
        this.smax=smax;
        this.smin=smin;
        this.rsmax=rsmax;
        this.rsmin=rsmin;
}
public Dimension getPreferredSize() {
        return new Dimension(800,400);
}
public void paint(Graphics g) {
                Dimension d= getSize();
                double open,open2,close,subclose,Diffy,minusy,cont,Diffx;
                int s,i1,i2,i;
                double zcmax,zcmin;
                double mov1[] = new double[j];
                double mov2[] = new double[j];
                i1=0;
                if (smax>rsmax) zcmax=smax;
                else zcmax=rsmax;
                if (smin<rsmin) zcmin=smin;
```

```
else zcmin=rsmin;
            close = zcmax - zcmin;
            subclose = 400/close;
            cont = close*subclose;
            i1=0;
            g.setColor(Color.red);
            s=0;
for (i=0;i<j-11;i++){
//scale 10//
                        open  = (mov3[i++]-zcmin)*subclose;
                        open2 = (mov3[i--]-zcmin)*subclose;
                        g.drawLine(i1+46,50+(int)(cont-open),i1+52,50+(int)(cont-open2));

                        i1=i1+6;
                        s++;
            }
i1=0;
g.setColor(Color.blue);
s=0;
for (i=0;i<j-21;i++){
  //scale 10//
                        open  = (da[i++]-zcmin)*subclose;
                        open2 = (da[i--]-zcmin)*subclose;
                        g.drawLine(i1+112,50+(int)(cont-open),i1+118,50+(int)(cont-open2));

                        i1=i1+6;
                        s++;
            }
g.setColor(Color.black);
g.drawLine(45,50+(int)cont,i1+112,50+(int)cont);
g.drawLine(i1+112,50+(int)cont,i1+112,40);
g.drawLine(i1+112,40,45,40);
g.drawLine(45,40,45,50+(int)cont);
i2=((j-10)/5)*6;
```

```
                g.drawString(""+(ig+1),40,50+(int)cont+15);                    //Draw First Date
Number//
                i=0;
                for(s=1;s<6;s++) {
        //Draw Date Number && Line//
                        i=i+i2;
                        if (s<5)
                                g.drawLine(i+45,50+(int)cont,i+45,40);
                        g.drawString(""+(ig+(i/6)-1),i+37,50+(int)cont+15);
                        }

                for(i=0;i<10;i++)
                //Draw -- Line//
                        g.drawLine(45,50+(int)(i*400)/10,i1+112,50+(int)(i*400)/10);

                        i2 = (int)(close*subclose)/10;
                minusy = 0;

                        for(i=0;i<=10;i++){
                                //Draw Stock Value//
                                Diffy = zcmax-minusy;
                                minusy = (close/10)*(i+1);
                                s = (int)(Diffy*100);
                                Diffy = (double)s/100;
                                g.drawString(""+Diffy,5,50+(i*i2));
                                }

                   Diffy = zcmax;
                        //Show Max//
                s = (int)(Diffy*100);
                        Diffy = (double)s/100;
                        g.drawString("  Max   "+Diffy,150,35);
```

```
Diffy = zcmin;
        //Show Min//
s = (int)(Diffy*100);
Diffy = (double)s/100;
g.drawString("  Min  "+Diffy,300,35);

Diffy = close;
        //Show Close//
s = (int)(Diffy*100);
Diffy = (double)s/100;
g.drawString("  Close  "+Diffy,450,35);
}
}
```

Neuron ใช้ประมวลผลทางด้านนิวรอนเนตเวิร์ก

```
package feedback;
import java.awt.*;
import java.util.Random;
public class Neuron extends RealGetStockGraphFrame{
        int j,jj,r,ig,it;
        float mov3[] = new float[j];float mov6[] = new float[j];float mov3f[] = new float[jj];
        float mov6f[]  = new float[jj];float daf[] = new float[jj];float dbf[] = new float[jj];
        float smax,vmax;
float neuronMax(float outGf[],int jj) {
        this.outGf=outGf;
        this.jj = jj;
        float x;float y;
        x=outGf[0];
        sgmaxf=x;
        for(i=0;i<jj-21;i++){
                x = outGf[i++];
                y = outGf[i--];
                if(x<y && sgmaxf<y)
                        sgmaxf=y;
          }
        return sgmaxf;
}
float neuronMin(float outGf[],int jj) {
        this.outGf=outGf;
        this.jj = jj;
        float x;float y;
        x=outGf[0];
        sgminf=x;
        for(i=0;i<jj-20-1;i++){
                x = outGf[i++];
                y = outGf[i--];
                if(x>y && sgminf>y)
```

```
                    sgminf=y;
                }
            return sgminf;
}
float[] Neurons(float da[],float db[],int j,int r,float mov3[],float mov6[],float mov6f[],float smax,
            float vmax,int jj,float mov3f[],float daf[],float dbf[]){
            this.mov3=mov3;
            this.r=r;this.j=j;this.jj=jj;
            this.mov3f=mov3f;this.mov6=mov6;this.mov6f=mov6f;
            this.smax=smax;this.vmax=vmax;this.daf=daf;this.dbf=dbf;
            this.da=da;this.db=db;
            int i1,i2,s,mem1;int neuros = 300;int points = 11;int width = 10;
            int norm = 0;int sweight = 4;int vweight = 1;
            int allpoints = (points-4)*neuros;
            double gg;
            int GRAM[][][] = new int[points][neuros][j-(norm+width-1)];
            int GRAMF[][] = new int[points][neuros];
            float outGf[] = new float[jj-norm];
            int VRAM[][][] = new int[points][neuros][j-(norm+width-1)];
            int VRAMF[][] = new int[points][neuros];
            int VOM[][] = new int[neuros][j-(norm+width-1)];
            int SOM[][] = new int[neuros][j-(norm+width-1)];
            int i,l,p,m,n,g,x,y,z,aaa,bbb,plus,sx,vx,rx;
            int gramrow[][] = new int[points][neuros];int vramrow[][] = new int[points][neuros];
            int vramcol[][] = new int[points][neuros];int gramcol[][] = new int[points][neuros];
            int stoer[] = new int[neuros];int vote[][] = new int[neuros][j-(norm+width)];
            float gramout[] = new float[j-(norm+width-1)];
            float vramout[] = new float[j-(norm+width-1)];
            float error,verror;float resram,vesram;
            float mov4[] = new float[j-1];float mov7[] = new float[j-1];
            float mov5[] = new float[j-1];float mov8[] = new float[j-1];
            float o1[] = new float[j-1-4];float o2[] = new float[j-1-4];
            float o1f[] = new float[jj-1-4];float o2f[] = new float[jj-1-4];
            float mov4f[] = new float[jj-1];float mov7f[] = new float[jj-1];
```

```
float mov5f[] = new float[jj-1];float mov8f[] = new float[jj-1];
float zzzz,yyyy,smmax,vmmax;
int kbox[][] = new int[width-1][301];int jbox[][] = new int[width-1][301];
int vbox[][] = new int[width-1][301];int trackrow[] = new int[allpoints];
int trackcol[] = new int[allpoints];int trackvrow[] = new int[allpoints];
int trackvcol[] = new int[allpoints];int wins[][] = new int[neuros][j-(norm+width)];
int stoda[] = new int[j-(norm+width)];int ch1,ch0,vh1,vh0,ln0,ln1;
for (g=0;g<j;g++){
        mov3[g]=da[g];mov6[g]=db[g];
}
for (g=0;g<jj;g++){
        mov3f[g]=daf[g];mov6f[g]=dbf[g];
}
for (g=0;g<j-1;g++){
        mov4[g] = mov3[g+1] - mov3[g];mov7[g] = mov6[g+1] - mov6[g];
}
float mov11[] = new float[j-1];float mov22[] = new float[j-1];
float mov33[] = new float[j-1];float xbar,ybar,breg,creg,areg,cal2y,caly,calcy,m3,m1;
for (i=5;i<j-1;i++) {
        xbar = 0;ybar = 0;breg = 0;creg = 0;areg = 0;
        cal2y = 0;caly = 0;calcy = 0;m3 = 0;m1 = 0;
        for (s=i-5;s<i;s++) {                                          //find mean//
                m1  = mov4[s];m3  = m1+m3;
                mov11[s] = m1;mov22[s] = s+1;
                xbar = xbar+mov22[s];
                }
        ybar = m3/5;xbar = xbar/5;
        for(s=i-5;s<i;s++) {
                breg = breg+((mov11[s]-ybar)*(mov22[s]-xbar));
                creg = creg+((mov22[s]-xbar)*(mov22[s]-xbar));
                }
        breg = breg/creg;
        areg = ybar-(breg*xbar);
```

```
            cal2y= areg+(breg*mov22[i-3]);
            caly = cal2y;
   mov33[i-5] = caly;
            }
for (g=0;g<j-1-4;g++){
            o1[g]=mov33[g];
}
for (i=5;i<j-1;i++) {
            xbar = 0;ybar = 0;breg = 0;creg = 0;areg = 0;
            cal2y = 0;caly = 0;calcy = 0;m3 = 0;m1 = 0;
            for (s=i-5;s<i;s++) {                                   //find mean//
                        m1   = mov7[s];m3   = m1+m3;
                        mov11[s] = m1;mov22[s] = s+1;
                        xbar = xbar+mov22[s];
                        }
            ybar = m3/5;xbar = xbar/5;
            for(s=i-5;s<i;s++) {
                        breg = breg+((mov11[s]-ybar)*(mov22[s]-xbar));
                        creg = creg+((mov22[s]-xbar)*(mov22[s]-xbar));
                        }
            breg = breg/creg;areg = ybar-(breg*xbar);
            cal2y= areg+(breg*mov22[i-3]);caly = cal2y;
   mov33[i-5] = caly;
            }
for (g=0;g<j-1-4;g++)
            o2[g]=mov33[g];
for (g=0;g<jj-1;g++){
            mov4f[g] = mov3f[g+1] - mov3f[g];
            mov7f[g] = mov6f[g+1] - mov6f[g];
}
for (i=5;i<jj-1;i++) {
            xbar = 0;ybar = 0;breg = 0;creg = 0;areg = 0;
            cal2y = 0;caly = 0;calcy = 0;m3 = 0;m1 = 0;
            for (s=i-5;s<i;s++) {                                   //find mean//
```

```
                m1  = mov4f[s];m3  = m1+m3;
                mov11[s] = m1;mov22[s] = s+1;
                xbar = xbar+mov22[s];
                }
        ybar = m3/5;xbar = xbar/5;
        for(s=i-5;s<i;s++) {
                breg = breg+((mov11[s]-ybar)*(mov22[s]-xbar));
                creg = creg+((mov22[s]-xbar)*(mov22[s]-xbar));
                }
        breg = breg/creg;areg = ybar-(breg*xbar);
        cal2y= areg+(breg*mov22[i-3]);caly = cal2y;
  mov33[i-5] = caly;
        }
for (g=0;g<jj-1-4;g++)
        o1f[g]=mov33[g];
for (i=5;i<jj-1;i++) {
        xbar = 0;ybar = 0;breg = 0;creg = 0;areg = 0;
        cal2y = 0;caly = 0;calcy = 0;m3 = 0;m1 = 0;
        for (s=i-5;s<i;s++) {                                   //find mean//
                m1  = mov7f[s];m3  = m1+m3;
                mov11[s] = m1;mov22[s] = s+1;
                xbar = xbar+mov22[s];
                }
        ybar = m3/5;xbar = xbar/5;
        for(s=i-5;s<i;s++) {
                breg = breg+((mov11[s]-ybar)*(mov22[s]-xbar));
                creg = creg+((mov22[s]-xbar)*(mov22[s]-xbar));
                }
        breg = breg/creg;areg = ybar-(breg*xbar);
        cal2y= areg+(breg*mov22[i-3]);caly = cal2y;
  mov33[i-5] = caly;
        }
for (g=0;g<jj-1-4;g++)
        o2f[g]=mov33[g];
```

```
for (p=0;p<jj-((width-2)+1-2+4);p++){
resram = 0;vesram = 0;
        for (g=0;g<width-3;g++){
mov5f[g] = o1f[p+g];mov8f[g] = o2f[p+g];
mov5f[width-3] = ((mov3f[p+width-1]-mov3f[p+width-2])+(mov3f[p+width]-mov3f[p+width-1])+
(mov3f[p+width+1]-mov3f[p+width]))/3;
mov5f[width-2] = ((mov3f[p+width+1]-mov3f[p+width])+(mov3f[p+width]-mov3f[p+width-1]))/2;
mov8f[width-3] = ((mov6f[p+width-1]-mov6f[p+width-2])+(mov6f[p+width]-mov6f[p+width-1])+
(mov6f[p+width+1]-mov6f[p+width]))/3;
mov8f[width-2] = ((mov6f[p+width+1]-mov6f[p+width])+(mov6f[p+width]-mov6f[p+width-1]))/2;}
        for (g=0;g<j-1-4;g++){
                mov5[g] = o1[g];mov8[g] = o2[g];
        }
        for (i=0;i<(width-1);i++)
                for(s=0;s<301;s++) {
                        kbox[i][s] = 0;vbox[i][s] = 0;jbox[i][s] = 0;
                }
        float sbato,vbato,ssbato,vvbato;
        vmmax = yyyy = 0;smmax = zzzz = 0;
        for (g=0;g<width-2;g++) {
        if(Math.abs(mov5f[g]) <= Math.abs(mov5f[g+1]) && zzzz <= Math.abs(mov5f[g+1])){
                        zzzz = Math.abs(mov5f[g+1]);
                        smmax = Math.abs(mov5f[g+1]);
                }
        if(Math.abs(mov8f[g]) <= Math.abs(mov8f[g+1]) && yyyy <= Math.abs(mov8f[g+1])){
                        yyyy = Math.abs(mov8f[g+1]);
                        vmmax = Math.abs(mov8f[g+1]);
                }
        }
        sbato = (float)(150/smmax);vbato = (float)(150/vmmax);
        ssbato = (float)(sbato/1.1);vvbato = (float)(vbato/1.1);
        int pl;
        for (pl=0;pl<j-1-4-(width-2);pl++)
                for (i=0;i<(width-1);i++){
```

```
                if (mov5[pl+i]<0){
                        if (mov5[pl+i]*sbato < -150) mov5[pl+i] = (float)(-smmax);
                        for (s=149;s>(int)(149+(mov5[pl+i]*sbato));s--){
                           kbox[i][s] = 1+kbox[i][s];
                        }}
                if (mov5[pl+i]>=0){
                        if (mov5[pl+i]*sbato > 150) mov5[pl+i] = (float)smmax;
                        for (s=150;s<(mov5[pl+i]*sbato)+150;s++){
                                kbox[i][s] = 1+kbox[i][s];
                        }}}
l=0;
for (pl=0;pl<j-1-4-(width-2);pl++)
        for (i=0;i<(width-1);i++){
                if (mov8[pl+i]<0){
                        if (mov8[pl+i]*vbato < -150) mov8[pl+i] = (float)(-vmmax);
                        for (s=149;s>(int)(149+(mov8[pl+i]*vbato));s--)
                                vbox[i][s] = 1+vbox[i][s];
                }
                if (mov8[pl+i]>=0){
                        if (mov8[pl+i]*vbato > 150) mov8[pl+i] = (float)vmmax;
                        for (s=150;s<(mov8[pl+i]*vbato)+150;s++)
                                vbox[i][s] = 1+vbox[i][s];
                }}
l=0;
for (i=0;i<(width-1);i++)
        for (s=0;s<301;s++)
                kbox[i][s] = (kbox[i][s]*100)/(j-(width-2)-1-4);
mem1 = 0;int conbox = 50;int div = 1;int nump = 0;
while (mem1 < allpoints){
        for (s=0;s<301;s++)
                for (i=0;i<(width-1);i++)
                        if(kbox[i][s] == conbox && mem1 < allpoints){
                                trackrow[mem1] = s;
                                trackcol[mem1] = i;
```

```
                                        jbox[i][s]=1;
                                        mem1++;
                                }
        nump++;
        div=div*(-1);
        if (div<0)
                conbox = conbox-nump;
        else
                conbox = conbox+nump;
}
mem1 =0;
for (i=0;i<(width-1);i++)
        for (s=0;s<301;s++)
                vbox[i][s] = (vbox[i][s]*100)/(j-(width-2)-1-4);
mem1 = 0;conbox = 50;div = 1;nump = 0;
while (mem1 < allpoints){
        for (i=0;i<(width-1);i++)
                for (s=0;s<301;s++)
                        if(vbox[i][s] == conbox && mem1 < allpoints){
                                trackvrow[mem1] = s;
                                trackvcol[mem1] = i;
                                mem1++;
                                //bpi[i][s] = 1;
                        }
        nump++;
        div=div*(-1);
        if(div<0)
                conbox = conbox-nump;
        else
                conbox = conbox+nump;
}

mem1 = (int)(Math.round(Math.random()*allpoints-1));
for (g=0;g<neuros;g++){
```

```
        for (i=0;i<(points-4);i++){
                gramrow[i][g] = trackrow[mem1];
                gramcol[i][g] = trackcol[mem1];
                vramrow[i][g] = trackvrow[mem1];
                vramcol[i][g] = trackvcol[mem1];
                mem1 = (int)(Math.round(Math.random()*(allpoints-1)));
            }
}
if(p<0)
        for (g=0;g<neuros;g++){
            for (i=points-4;i<points;i++){
                gramcol[i][g] = 7;
                vramcol[i][g] = 7;
                mem1 = (int)(Math.round(Math.random()*299));
                gramrow[i][g] = mem1;
                mem1 = (int)(Math.round(Math.random()*299));
                vramrow[i][g] = mem1;
            }}
        else
        for (g=0;g<neuros;g++){
            for (i=points-4;i<points;i++){
                gramcol[i][g] = 8;
                vramcol[i][g] = 8;
                mem1 = (int)(Math.round(Math.random()*299));
                gramrow[i][g] = mem1;
                mem1 = (int)(Math.round(Math.random()*299));
                vramrow[i][g] = mem1;
            }}
n=0;
pl=0;
while (pl<j-(width-2)-1-4-1) {
        for (g=0;g<neuros;g++){
            for (i=0;i<points;i++){
                if (mov5[pl+gramcol[i][g]]<0){
```

```
                if (gramrow[i][g] < mov5[pl+gramcol[i][g]]*sbato+150)
                                GRAM[i][g][pl] = 0;
                        else
                                GRAM[i][g][pl] = 1;
                        if      (gramrow[i][g] >= 150)
                                GRAM[i][g][pl] = 0;
                }
                if (mov5[pl+gramcol[i][g]]>=0){
                if (gramrow[i][g] > mov5[pl+gramcol[i][g]]*sbato+150)
                                GRAM[i][g][pl] = 0;
                        else
                                GRAM[i][g][pl] = 1;
                        if      (gramrow[i][g] < 150)
                                GRAM[i][g][pl] = 0;
                }
                if (mov8[pl+vramcol[i][g]]<0){
                if (vramrow[i][g] < mov8[pl+vramcol[i][g]]*vbato+150)
                                VRAM[i][g][pl] = 0;
                        else
                                VRAM[i][g][pl] = 1;
                        if      (vramrow[i][g] >= 150)
                                VRAM[i][g][pl] = 0;
                }
                if (mov8[pl+vramcol[i][g]]>=0){
                if (vramrow[i][g] > mov8[pl+vramcol[i][g]]*vbato+150)
                                VRAM[i][g][pl] = 0;
                        else
                                VRAM[i][g][pl] = 1;
                        if      (vramrow[i][g] < 150)
                                VRAM[i][g][pl] = 0;
                }}
        if (o1[pl+9]<0){
                if (g < o1[pl+9]*ssbato+150)
                        SOM[g][pl] = 0;
```

```
                    else
                        SOM[g][pl] = 1;
                    if      (g >= 150)
                        SOM[g][pl] = 0;
                }
                if(o1[pl+9]>=0){
                    if(g > o1[pl+9]*ssbato+150)
                        SOM[g][pl] = 0;
                    else
                        SOM[g][pl] = 1;
                    if      (g < 150)
                        SOM[g][pl] = 0;
                }
                if(o2[pl+9]<0){
                    if(g < o2[pl+9]*vvbato+150)
                        VOM[g][pl] = 0;
                    else
                        VOM[g][pl] = 1;
                    if      (g >= 150)
                        VOM[g][pl] = 0;
                }
                if(o2[pl+9]>=0){
                    if(g > o2[pl+9]*vvbato+150)
                        VOM[g][pl] = 0;
                    else
                        VOM[g][pl] = 1;
                    if      (g < 150)
                        VOM[g][pl] = 0;
                }}
            pl++;
}
pl=0;
for (g=0;g<neuros;g++)
        for (i=0;i<points;i++){
```

```
                if(mov5f[gramcol[i][g]]<0){
                        if (gramrow[i][g] < mov5f[gramcol[i][g]]*sbato+150)
                                GRAMF[i][g] = 0;
                        else
                                GRAMF[i][g] = 1;
                        if      (gramrow[i][g] >= 150)
                                GRAMF[i][g] = 0;
                }
                if(mov5f[gramcol[i][g]]>=0){
                        if (gramrow[i][g] > mov5f[gramcol[i][g]]*sbato+150)
                                GRAMF[i][g] = 0;
                        else
                                GRAMF[i][g] = 1;
                        if      (gramrow[i][g] < 150)
                                GRAMF[i][g] = 0;
                }
                if(mov8f[vramcol[i][g]]<0){
                        if (vramrow[i][g] < mov8f[vramcol[i][g]]*vbato+150)
                                VRAMF[i][g] = 0;
                        else
                                VRAMF[i][g] = 1;
                        if      (vramrow[i][g] >= 150)
                                VRAMF[i][g] = 0;
                }
                if(mov8f[vramcol[i][g]]>=0){
                        if (vramrow[i][g] > mov8f[vramcol[i][g]]*vbato+150)
                                VRAMF[i][g] = 0;
                        else
                                VRAMF[i][g] = 1;
                        if      (vramrow[i][g] < 150)
                                VRAMF[i][g] = 0;
                }}
for (m=0;m<j-(width-2)-1-4;m++)
        for (g=0;g<neuros;g++){
```

```
                vote[g][m] = 0;
                wins[g][m] = 0;
                stoda[m] = 0;
        }
for (m=0;m<j-(width-2)-1-4;m++)
        for (g=0;g<neuros;g++)
                for (i=0;i<points;i++){
                if(GRAMF[i][g] == GRAM[i][g][m]) vote[g][m] = vote[g][m]+sweight;
                if(VRAMF[i][g] == VRAM[i][g][m]) vote[g][m] = vote[g][m]+vweight;
                }
for(g=0;g<neuros;g++){
        mem1 = 0;n=0;z=0;i=0;s=0;plus=0;
                        do{
                                for (m=0;m<j-(width-1)-1-4;m++)
                                if (vote[g][m] == (sweight+vweight)*points-plus){
                                                wins[g][m] = 1;
                                                s = 1;
                                        }
                                plus++;
                                if (plus == (sweight+vweight)*points) s=1;
                        }
                        while(s==0);
}
ch1 = ch0 = vh1 = vh0 =0;error = 0;verror = 0;
for (g=0;g<neuros;g++){
        ch1 = ch0 = vh1 = vh0 =0;
        for (m=0;m<j-(width-1)-1-4;m++)
                if(wins[g][m] == 1){
                        if(SOM[g][m]==1) ch1 = ch1+1;
                        if(SOM[g][m]==0) ch0 = ch0+1;
                        if(VOM[g][m]==1) vh1 = vh1+1;
                        if(VOM[g][m]==0) vh0 = vh0+1;
                }
        if(ch1>ch0) stoer[g]=1;
```

```
                else stoer[g]=0;
                if(vh1>vh0) verror=1;
                else verror=0;
                System.out.print(stoer[g]);
        }
        ln0=ln1=ch0=ch1=0;
        for(g=0;g<150;g++)
                if(stoer[g] == 1)
                        {ch0=ch0+g+1;
                         ln0++;
                        }
        for(g=150;g<300;g++)
                if(stoer[g] == 1)
                        {ch1=ch1+(g-149);
                         ln1++;
                        }
        if(ln0>ln1) resram = (-1)*((ch0/ln0)/ssbato);
        else
                resram = (ch1/ln1)/ssbato;
        System.out.println();
        System.out.println(resram);
        for(g=0;g<neuros;g++)
                        stoer[g] = 0;
        vesram = 0;
        outGf[p] = resram+mov3f[p+(width+1)];
        System.out.println("P ="+p+"ANS ="+outGf[p]+"ANSTRUE ="+(mov3f[p+(width+2)]-
mov3f[p+width+1])+"SMAX ="+smmax);
    }
        for (p=0;p<jj-((width-2)+1-2+4);p++)
                System.out.println("ANSNORM ="+(outGf[p]-mov3f[p+(width+1)])+
                        "ANSTRUE ="+(mov3f[p+(width+2)]-mov3f[p+width+1]));
        return outGf;
}}
```

RealGetStockGraphFrame เป็นส่วนหลักในการแสดงผลติดต่อกับ user และเชื่อมโยงส่วนต่างๆ ทั้งอินพุท และเอาท์พุท

```
package feedback;
import java.io.*;
import java.awt.*;
public class RealGetStockGraphFrame extends Frame {
        DataInputStream input;boolean moreRecords=true;
        int i,j,l,m,sel,jj;int beg=0;int r;int k=2000;
        float ab,ac;float rsmax,rsmin,rvmax,rvmin,smax,smin,vmax,vmin;
        float rsmaxf,rsminf,rvmaxf,rvminf,smaxf,sminf,vmaxf,vminf;
        String days[] = new String[k];
        Integer am[] = new Integer[k];
        float da[] = new float[k];float db[] = new float[k];
        float mov3[] = new float[k];float mov6[] = new float[k];
        float mov3f[] = new float[k];float mov6f[] = new float[k];
        float daf[] = new float[k];float dbf[] = new float[k];
        float outGf[] = new float[k];float sgmaxf,sgminf;
        String daysf[] = new String[k];
        Panel pan1,pan2,pan3;
        Frame f1;
        MyMovCanvasall MovCanall;
        MyFrame Fam;
        MyMov Mov;
        MyMovCanvas MovCan;
        Neuron Neu;
        Button quit    = new Button("Quit");   Button form1  = new Button("Form1");
        Button form2 = new Button("Form2");Button form3    = new Button("Form3");
        Button form4   = new Button("Form4");Button next  = new Button("Next.");
        Button back  = new Button("Back.");
public boolean action(Event event,Object o)
                {if (event.target instanceof Button)
                        {if (event.arg.equals("Quit"))
                                {System.exit(0);return true;}}
```

```
{if (event.arg.equals("Form1"))
        {r = 0;sel = 0;
        if(m!=0) f1.dispose();
        m++;
        f1 = new Frame();
        Mov = new MyMov(0,j,r,da,rsmax,rsmin);
        f1.setSize(800,520);
        Mov.setSize(800,520);
        Fam = new MyFrame(f1,Mov);
        }}
{if (event.arg.equals("Form2"))
        {r = 0;sel = 1;
        if(m!=0) f1.dispose();
        m++;
        f1 = new Frame();
        Mov = new MyMov(10,j,r,mov3,smax,smin);
        f1.setSize(800,520);
        Mov.setSize(800,520);
        Fam = new MyFrame(f1,Mov);
        }}
{if (event.arg.equals("Form3"))
        {r = 0;sel = 2;
        if(m!=0) f1.dispose();
        m++;
        f1 = new Frame();
        MovCan = new MyMovCanvas(j,r,da,mov3,smax,smin,rsmax,rsmin);
        f1.setSize(800,520);
        MovCan.setSize(800,520);
        Fam = new MyFrame(f1,MovCan);
        }}
{if (event.arg.equals("Form4"))
        {r = 0;sel = 3;
        if(m!=0) f1.dispose();
        m++;
```

```
                    f1 = new Frame();
                    if(beg == 0) {
                Neu = new Neuron();
                outGf = Neu.Neurons(da,db,j,r,mov3,mov6,mov6f,smaxf,vmaxf,jj,mov3f,daf,dbf);
                sgmaxf = Neu.neuronMax(outGf,jj);
                sgminf = Neu.neuronMin(outGf,jj);
                    }
                    beg++;
                    MovCanall = new
MyMovCanvasall(jj,r,outGf,mov3f,smaxf,sminf,sgmaxf,sgminf);
                    f1.setSize(800,520);
                    MovCanall.setSize(800,520);
                    Fam = new MyFrame(f1,MovCanall);
                   }}
                {if (event.arg.equals("Next.")) 
                    {r++;
                    if(sel==0) {
                        if(m!=0) f1.dispose();
                        f1 = new Frame();
                        Mov = new MyMov(0,j,r,db,rvmax,rvmin);
                        f1.setSize(800,520);
                        Mov.setSize(800,520);
                        Fam = new MyFrame(f1,Mov);
                       }
                    if(sel==1) {
                        if(m!=0) f1.dispose();
                        f1 = new Frame();
                        Mov = new MyMov(10,j,r,mov6,vmax,vmin);
                        f1.setSize(800,520);
                        Mov.setSize(800,520);
                        Fam = new MyFrame(f1,Mov);
                       }
                    if(sel==2) {
                    if(m!=0) f1.dispose();
```

```
                f1 = new Frame();
                MovCan = new MyMovCanvas(j,r,db,mov6,vmax,vmin,rvmax,rvmin);
                f1.setSize(800,520);
                MovCan.setSize(800,520);
                Fam = new MyFrame(f1,MovCan);
                        }
                if(sel==3) {
                        if(m!=0) f1.dispose();
                        f1 = new Frame();
                        f1.setSize(720,520);
                        MovCanall.setSize(720,520);
                        Fam = new MyFrame(f1,MovCanall);
                        }}}
{if (event.arg.equals("Back."))
                {r--;
                if(sel==0) {
                        if(m!=0) f1.dispose();
                        f1 = new Frame();
                        Mov = new MyMov(0,j,r,da,rsmax,rsmin);
                        f1.setSize(800,520);
                        Mov.setSize(800,520);
                        Fam = new MyFrame(f1,Mov);
                        }
                if(sel==1) {
                        if(m!=0) f1.dispose();
                        f1 = new Frame();
                        Mov = new MyMov(10,j,r,mov3,smax,smin);
                        f1.setSize(800,520);
                        Mov.setSize(800,520);
                        Fam = new MyFrame(f1,Mov);
                        }
                if(sel==2) {
                if(m!=0) f1.dispose();
                f1 = new Frame();
```

```
                                        MovCan = new MyMovCanvas(j,r,da,mov3,smax,smin,rsmax,rsmin);
                                        f1.setSize(800,520);
                                        MovCan.setSize(800,520);
                                        Fam = new MyFrame(f1,MovCan);
                                        }
                                        if(sel==3) {
                                                if(m!=0) f1.dispose();
                                                f1 = new Frame();
                                                f1.setSize(720,520);
                                                MovCanall.setSize(720,520);
                                                Fam = new MyFrame(f1,MovCanall);
                                               }}}
                        return true;
                        }
        public void cleanup() {try {input.close();}
                        catch(IOException e){System.exit(1);}}
        public static void main (String args[]){
                         RealGetStockGraphFrame a =new RealGetStockGraphFrame();
                         a.readRecord();a.setup();a.cleanup();
                         }
public void readRecord()
                        {i=0;
                         String pol,jol,jolto;
                         int cc4,iii; int ig,it,s;
                         float xbar,ybar,breg,creg,areg,cal2y,caly,calcy,weight,m3,m1,wreg,de,x,y;
                        float mov1[] = new float[k];  float mov2[] = new float[k];
                         try{input=new DataInputStream(new FileInputStream("c:/1SHIN93.txt"));}
                         catch(IOException e) {System.err.println("Error\n"+e.toString());
                         System.exit(1);}
                         de = 0;int norm = 5;
                         try {
                                 while (input.readByte() != '\n');
                                 while (moreRecords =true && i<=k-1) {
                                         byte b[] = new byte[9];
```

```
                    input.readFully(b);
                    pol=new String(b);
                    days[i] = pol;
                    while (input.readByte() != '$');
                    byte c[] = new byte[7];
                    input.readFully(c);
                    jol = new String(c);
                    float ab = (new Float(jol)).floatValue();
            da[i]=ab;
                    while (input.readByte() != '$');
                    byte d[] = new byte[7];
                    input.readFully(d);
                    jol = new String(d);
                    float ac = (new Float(jol)).floatValue();
            db[i]=ac;
            while (input.readByte() != '\n');
                    i++;
                    j=i;//use j for paint method// //j=k//
                    }}
catch (NumberFormatException en) {
          System.out.println(days[i-1]);System.out.println(days[i]);
          System.out.println(i);System.exit(1);}
catch (EOFException eof){
          moreRecords =false;
x=da[0];
rsmax=x;rsmin=x;
for(i=0;i<j-1;i++){
         x=da[i++];y=da[i--];
         if(x<y && rsmax<y)
                  rsmax=y;
           if(x>y && rsmin>y)
                  rsmin=y;
         }
x=db[0];rvmax=x;rvmin=x;
```

```
for(i=0;i<j-1;i++){
        x=db[i++];y=db[i--];
        if(x<y && rvmax<y)
                rvmax=y;
          if(x>y && rvmin>y)
                rvmin=y;
        }
for (i=norm;i<j;i++) {
        xbar = 0;ybar = 0;breg = 0;creg = 0;areg = 0;
        cal2y = 0;caly = 0;calcy = 0;weight = 0;
        m3 = 0;m1 = 0;wreg = 0;
        for (s=i-norm;s<i;s++) {
                        m1   = da[s];m3   = m1+m3;
                        mov1[s] = m1;mov2[s] = s+1;
                        xbar = xbar+mov2[s];
                        }
        ybar = m3/norm;xbar = xbar/norm;
        for(s=i-norm;s<i;s++) {
                breg = breg+((mov1[s]-ybar)*(mov2[s]-xbar));
                creg = creg+((mov2[s]-xbar)*(mov2[s]-xbar));
                }
        breg = breg/creg; areg = ybar-(breg*xbar);
        cal2y= areg+(breg*mov2[i-3]);
caly = cal2y;
mov3[i-norm] = caly;
}
x=mov3[0];
        smax=x; smin=x;
for(i=0;i<j-(norm+1);i++){
        x = mov3[i++];y = mov3[i--];
        if(x<y && smax<y)
                smax=y;
                if(x>y && smin>y)
```

```
                    smin=y;
            }
for (i=norm;i<j;i++) {
            xbar = 0;ybar = 0;breg = 0;creg = 0;areg = 0;
            cal2y = 0;caly = 0;calcy = 0;weight = 0;m3 = 0;m1 = 0;wreg = 0;
            for (s=i-norm;s<i;s++) {
                                m1   = db[s];m3   = m1+m3;
                                mov1[s] = m1;mov2[s] = s+1;
                                xbar = xbar+mov2[s];
                                }
            ybar = m3/norm;xbar = xbar/norm;
            for(s=i-norm;s<i;s++) {
                        breg = breg+((mov1[s]-ybar)*(mov2[s]-xbar));
                        creg = creg+((mov2[s]-xbar)*(mov2[s]-xbar));
                        }
            breg = breg/creg;    areg = ybar-(breg*xbar);
            cal2y= areg+(breg*mov2[i-3]);caly = cal2y;
mov6[i-norm] = caly;
}
            x=mov6[0];vmax=x; vmin=x;
            for(i=0;i<j-(norm+1);i++){
                        x = mov6[i++]; y = mov6[i--];
            if(x<y && vmax<y)
                        vmax=y;
                        if(x>y && vmin>y)
                        vmin=y;
            }}
catch (IOException e){System.err.println
            ("Error\n"+e.toString());
             System.exit(1);}
try{input=new DataInputStream(new FileInputStream("c:/SHINFORECAST1.txt"));}
catch(IOException e) {System.err.println("Error\n"+e.toString());
System.exit(1);}
```

```
try {
        i=0;
        while (input.readByte() != '\n');
        while (moreRecords =true && i<=k-1) {
                byte b[] = new byte[9];
                input.readFully(b);
                pol=new String(b);
                daysf[i] = pol;
                while (input.readByte() != '$');
                byte c[] = new byte[7];
                input.readFully(c);
                jol = new String(c);
                float ab = (new Float(jol)).floatValue();
          daf[i]=ab;
                while (input.readByte() != '$');
                byte d[] = new byte[7];
                input.readFully(d);
                jol = new String(d);
                float ac = (new Float(jol)).floatValue();
          dbf[i]=ac;
          while (input.readByte() != '\n');
                i++;
                jj=i;//use j for paint method// //j=k//
                }}
catch (NumberFormatException en) {
        System.out.println(daysf[i-1]);System.out.println(daysf[i]);
        System.out.println(i);System.exit(1);}
catch (EOFException eof){
        moreRecords =false;
x=daf[0]; rsmaxf=x; rsminf=x;
for(i=0;i<jj-1;i++){
        x=daf[i++];  y=daf[i--];
        if(x<y && rsmaxf<y)
                rsmaxf=y;
```

```
            if(x>y && rsminf>y)
                    rsminf=y;
            }
 x=dbf[0]; rvmaxf=x; rvminf=x;
 for(i=0;i<jj-1;i++){
            x=dbf[i++];y=dbf[i--];
            if(x<y && rvmaxf<y)
                    rvmaxf=y;
              if(x>y && rvminf>y)
                    rvminf=y;
            }
 System.out.println("jj ="+jj);
 for (i=norm;i<(jj+1);i++) {
            xbar = 0; ybar = 0; breg = 0; creg = 0; areg = 0; cal2y = 0;
            caly = 0;  calcy = 0;weight = 0;m3 = 0;m1 = 0;wreg = 0;
            for (s=i-norm;s<i;s++) {
                            m1   = daf[s];m3   = m1+m3;
                            mov1[s] = m1;mov2[s] = s+1;
                            xbar = xbar+mov2[s];
                            }
            ybar = m3/norm;    xbar = xbar/norm;
            for(s=i-norm;s<i;s++) {
                    breg = breg+((mov1[s]-ybar)*(mov2[s]-xbar));
                    creg = creg+((mov2[s]-xbar)*(mov2[s]-xbar));
                    }
            breg = breg/creg;    areg = ybar-(breg*xbar);
            cal2y= areg+(breg*mov2[i-3]); caly = cal2y;
mov3f[i-norm] = caly;
}
x=mov3f[0]; smaxf=x; sminf=x;
for(i=0;i<jj-(norm+1);i++){
            x = mov3f[i++];y = mov3f[i--];
            if(x<y && smaxf<y)
                    smaxf=y;
```

```
                    if(x>y && sminf>y)
                    sminf=y;
                }
        for (i=norm;i<jj;i++) {
                xbar = 0; ybar = 0; breg = 0; creg = 0; areg = 0; cal2y = 0;
                caly = 0;  calcy = 0;weight = 0;m3 = 0;m1 = 0;wreg = 0;
                for (s=i-norm;s<i;s++) {
                                m1   = dbf[s];m3   = m1+m3;
                                mov1[s] = m1;mov2[s] = s+1;
                                xbar = xbar+mov2[s];}
                        ybar = m3/10;xbar = xbar/10;
                for(s=i-norm;s<i;s++) {
                        breg = breg+((mov1[s]-ybar)*(mov2[s]-xbar));
                        creg = creg+((mov2[s]-xbar)*(mov2[s]-xbar));
                        }
                breg = breg/creg;areg = ybar-(breg*xbar);
                        cal2y= areg+(breg*mov2[i-3]);caly = cal2y;
                mov6f[i-norm] = caly;
                }
                x=mov6f[0];vmaxf=x;vminf=x;
                for(i=0;i<jj-(norm+1);i++){
                        x = mov6f[i++];y = mov6f[i--];
                if(x<y && vmaxf<y)
                        vmaxf=y;
                        if(x>y && vminf>y)
                        vminf=y;
                }}
        catch (IOException e){System.err.println
                ("Error\n"+e.toString());
                 System.exit(1);}
        }
public void setup() {
        pan1 = new Panel();
        pan2 = new Panel();
```

```
        pan3 = new Panel();
        pan1.add(form1);pan1.add(next);pan1.add(back);
        pan2.add(form2);pan2.add(form3);pan2.add(form4);
        pan3.add(quit);
        add("North",pan1);
        add("Center",pan2);
        add("South",pan3);
        setSize(170,120);setVisible(true);
    }
}
```

หนังสืออ้างอิง

1. กัลยา วานิชย์บัญชา ดร., "การวิเคราะห์สถิติ : สถิติเพื่อการตัดสินใจ" , คณะพาณิชยศาสตร์และการบัญชี จุฬาลงกรณ์มหาวิทยาลัย, พิมพ์ครั้งที่ 2, 2539
2. วิชัย สุรเชิดเกียรติและคณะ, "ทฤษฎีและตัวอย่างโจทย์สถิติ" , แมคกรอฮิลอินเตอร์เนชั่นแนลเอ็นเตอร์ไพร์ส, 2542, หน้า 310-342
3. Martin T. Hagan,Howard B. Demuth,Mark Beale, "Neural Network Design", Boston :PWS, 1996
4. N.K.Bose,P.Liang, "Neural Network fundamentals with graphs,algorithms and Application" , New York, NY ; McGrawHill, 1996
5. Patrick K. Simpson, "Neural Network theory,technology,and Application" , New York:IEEE Technical Activities Board, 1996
6. Robert L. Harvey, "Neural Network principles", Englewood Cliffs, NJ:Prentice-Hall International, 1994
7. Tarun Khanna, "Foundations of Neural Networks", Addison-Wesley Publishing Company, 1990