บทที่ 3
กรอบแนวความคิดและระเบียบวิธีวิจัย

3.1 กรอบแนวคิด

กรอบแนวคิดแบ่งออกเป็น 4 ส่วน ได้แก่ แบบจำลองนิวรอลเน็ตเวิร์ค, การทดสอบความนิ่งของข้อมูล (Unit Root Test), แบบจำลอง ARIMA และแบบจำลอง GARCH–M ซึ่งมีรายละเอียดดังต่อไปนี้

3.1.1 แบบจำลองนิวรอลเน็ตเวิร์ค

ขจรศักดิ์ คันธพนิต และยิ่งศักดิ์ ศรีมารัตน์ (2542) ได้กล่าวไว้ว่า Artificial Neural Networks หรือข่ายสมองเทียม คือระบบที่มนุษย์สร้างขึ้นหรือพัฒนาขึ้นมาเพื่อเลียนแบบการทำงานของสมองมนุษย์ โดยอาจจะเป็นซอฟแวร์หรือฮาร์ดแวร์ หรือระบบผสมผสานระหว่างซอฟแวร์และฮาร์ดแวร์ก็ได้ โดยข่ายสมองต้องมีคุณสมบัติที่สำคัญๆของสมองได้แก่

1) สามารถเรียนรู้และเก็บความรู้ไว้ได้

2) สามารถนำความรู้ที่เก็บไว้นั้นมาใช้ประกอบในการตัดสินใจอย่างชาญฉลาด

โดยการทำงานของสมองมนุษย์นั้นประกอบขึ้นด้วยส่วนที่เล็กที่สุดคือเซลล์สมองหรือนิวรอน (Neuron) จำนวนประมาณ 10^{11} เซลล์ ต่อเชื่อมระหว่างกันด้วยไซแนพส์ (Synapses) โดยการเชื่อมต่อจะเป็นไปหลายๆ แบบและหลายๆ กลุ่มแล้วแต่การทำงานของแต่ละส่วนของสมอง

โดยมีส่วนประกอบที่สำคัญคือตัวเซลล์ (Cell Body or Soma) มีนิวเคลียส (Nucleus) อยู่ตรงกลางทำหน้าที่ประมวลผลสัญญาณกระตุ้นทางไฟฟ้า (Electrical Impulses) ที่ส่งเข้ามาจากเซลล์อื่นผ่านทางเดนไดรท์ (Dendrites) ซึ่งต่อเชื่อมกับเดนไดรท์ของเซลล์อื่นๆ รอบๆ ข้าง หากมีสัญญาณกระตุ้นเข้ามามากเพียงพอ เซลล์นี้ก็จะส่งสัญญาณกระตุ้นทางไฟฟ้าขนาดประมาณ 50 มิลลิโวลท์ในช่วงเวลาประมาณ 2 มิลลิวินาที ต่อไปให้เซลล์อื่นๆผ่านทางเอกซอน (Axon) ไปยังเดนไดรท์เพื่อส่งออกให้แก่เดนไดรท์ของเซลล์อื่น โดยการเชื่อมต่อระหว่างเดนไดรท์ของเซลล์ต่างๆ นั้นจะผ่านจุดปลายคือ Terminals โดยจะมีช่องว่างไซแนพส์ (Synaptic Gap) อยู่เล็กน้อย โดยการส่งสัญญาณไฟฟ้าของเซลล์สมองนั้นเกิดจากการไหลหรือถ่ายเทของสารเคมีชนิดต่างๆ ผ่านเดนไดรท์อีกทีหนึ่ง จากการทำงานของสมองมนุษย์นี้ จึงได้มีการพัฒนาเพื่อเลียนแบบการทำงานของ

ลิขสิทธิ์มหาวิทยาลัยเชียงใหม่
Copyright© by Chiang Mai University
All rights reserved

สมองมนุษย์เพื่อนำไปประยุกต์ใช้ในงานวิจัยในสาขาต่างๆ โดยในทางเศรษฐศาสตร์นั้นก็เพิ่งเริ่มที่จะนำ Artificial Neural Networks มาใช้ในการวิจัยเมื่อไม่นานมานี้ ดังเช่นในงานวิจัยของคมสัน สุริยะ ซึ่งตีพิมพ์อย่างเป็นทางการใน Komsan (2006) ได้ทำการพยากรณ์ราคาน้ำมันดิบด้วยแบบจำลอง Neural Networks การศึกษานี้ใช้แบบจำลองนิวรอลเน็ตเวิร์คแบบแพร่ย้อนกลับ (Back Propagation) โดยมีการประเมินผลด้วย MAPE (Mean Absolute Percentage Error) มีสูตรการคำนวณดังนี้

$$MAPE = \frac{1}{n}\sum_{i=1}^{n} APE_i$$

$$APE_i = \frac{|t_i - y_i|}{t_i} \times 100$$

โดยที่ t_i = ค่าจริง

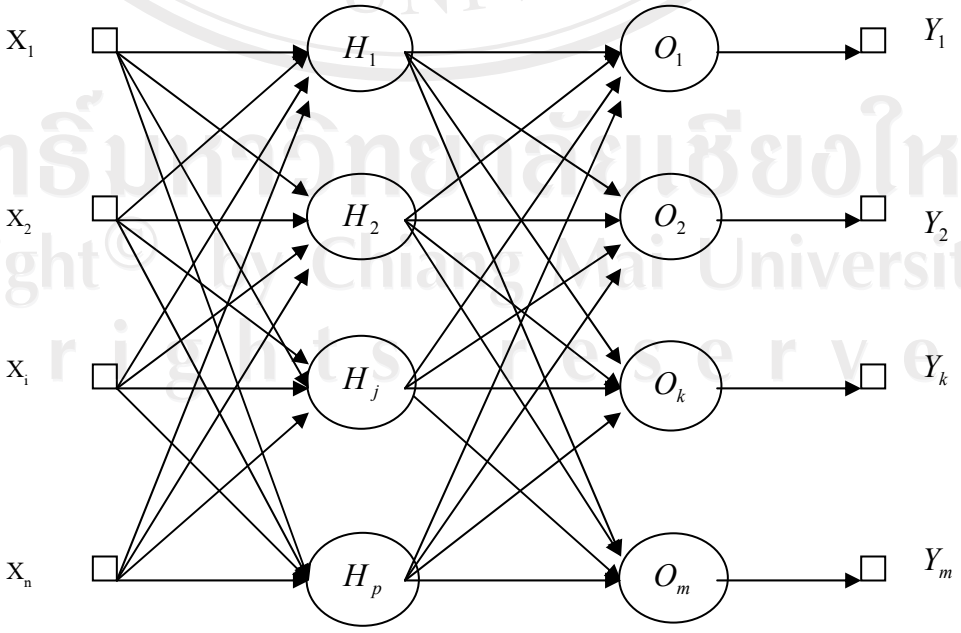
y_i = ค่าพยากรณ์

n = จำนวนข้อมูลของชุดข้อมูล Testing

นิวรอลเน็ตเวิร์คแบบแพร่ย้อนกลับ (Back Propagation) เป็นนิวรอลเน็ตเวิร์คแบบหนึ่งที่อยู่ในกลุ่มของ Multilayer Feed Forward Neural Networks (MLFF) ซึ่งจะรับข้อมูลนำเข้า (Input) แล้วทำการประมวลผลโดยนิวรอลในชั้น Hidden Layer และชั้น Output จากนั้นจะส่งผลลัพธ์ออกมาเป็นคำตอบ การที่สัญญาณนำเข้าถูกถ่ายทอดไปข้างหน้าเรื่อยๆ เรียกว่า Feed Forward และการที่มีจำนวนนิวรอลหลายชั้นเรียกว่า Multi Layer

นิวรอลเน็ตเวิร์คแบบแพร่ย้อนกลับสามารถแสดงได้ดังแผนภาพดังต่อไปนี้

รูปที่ 3.1 นิวรอลเน็ตเวิร์คแบบแพร่ย้อนกลับ



ลิขสิทธิ์มหาวิทยาลัยเชียงใหม่
Copyright© by Chiang Mai University
All rights reserved

องค์ประกอบของนิวรอลเน็ตเวิร์คแบบแพร่ย้อนกลับ ดังต่อไปนี้

1) Input

Input มีจำนวน n ตัว คือ X = { X_1 , X_2 ,..., X_i ...., X_n }

2) Output

Output มีจำนวน m ตัว คือ Y = {Y_1 , Y_2 ,..., Y_k ,....,Y_m }

3) นิวรอลใน Hidden Layer

นิวรอลใน Hidden Layer มี p ตัว คือ H = {H_1 , H_2 ,..., H_j ,....,H_p }

4) นิวรอลใน output Layer

นิวรอลใน Output Layer มี m ตัว คือ O = {O_1 , O_2 ,..., O_k ,....,O_m }

5) ค่าน้ำหนักจาก Input layer สู่ hidden layer

ค่าน้ำหนักจาก input layer สู่ hidden layer สำหรับนิวรอลแต่ละตัวใน hidden layer มีจำนวน n+1 ตัว คือ $W_{ij}^H = \{W_{0j}^H, W_{1j}^H, W_{2j}^H, ..., W_{ij}^H, W_{nj}^H\}$

ดังนั้นเมื่อมีจำนวน input จำนวน n ตัว และมีนิวรอลใน hidden layer จำนวน p ตัว แล้วจะได้ว่าจำนวนค่าน้ำหนักในส่วนนี้มีจำนวนทั้งหมดเท่ากับ (n+1)p ตัว

6) ค่าน้ำหนักจาก hidden layer สู่ output layer

ค่าน้ำหนักจาก Hidden layer สู่ output layer สำหรับนิวรอลแต่ละตัวใน output layer มีจำนวน p+1 ตัว คือ $W_{jk}^0 = \{W_{0k}^0, W_{1k}^0, W_{2k}^0, ..., W_{jk}^o, W_{pk}^0\}$

ดังนั้นเมื่อมีจำนวน Hidden layer จำนวน p ตัว และมีoutput จำนวน m ตัวแล้วจะได้ว่าจำนวนค่าน้ำหนักในส่วนนี้มีจำนวนทั้งหมดเท่ากับ (p+1)m ตัว

การคำนวณแบบแพร่ย้อนกลับ (Back Propagation)

ขั้นตอนของ Back-propagation Algorithm มีดังนี้

1) กำหนดค่าอัตราเร็วในการเรียนรู้ (rate parameter : r)

2) สำหรับแต่ละตัวอย่าง input ให้ทำตามขั้นตอนต่อไปนี้จนกว่าได้ระดับ performance ที่ต้องการ

- คำนวณหาค่า output โดยใช้ค่าน้ำหนักเริ่มต้นซึ่งอาจได้จากการสุ่ม
- คำนวณหาค่า β : แทนประโยชน์ที่จะได้รับสำหรับการเปลี่ยนค่า output ของแต่ละ node

ลิขสิทธิ์มหาวิทยาลัยเชียงใหม่
Copyright© by Chiang Mai University
All rights reserved

จากกฎการปรับค่าน้ำหนักของ Gradient Descent

$$\Delta w = \eta(t-y)\left[\nabla_w f(a)\right] \tag{3.1}$$

โดยกำหนด $y = f(a) = f\left(w^t x\right) = \dfrac{1}{1+e^{-a}}$ เมื่อ $a = w^T x$

ดังนั้น

$$\nabla_w f(a) = \begin{pmatrix} \partial f(a)/\partial w_0 \\ \partial f(a)/\partial w_1 \\ . \\ . \\ . \\ \partial f(a)/\partial w_n \end{pmatrix} = \frac{\partial f(a)}{\partial w} = \frac{\partial f(a)}{\partial a}\frac{\partial a}{\partial w}$$

$$= f'\frac{\partial\left(w^T x\right)}{\partial w} = f'\frac{\sum_{i=0}^{n} w_i x_i}{\partial w_i} = f' x_i$$

$$= f'\begin{bmatrix} x_0 \\ x_1 \\ . \\ . \\ . \\ x_n \end{bmatrix} = f' x$$

จึงสรุปได้ว่า $\Delta w = \eta(t-y) f' x$ (จิตติ ตันเสนีย์, 2549)

การปรับค่าน้ำหนักในชั้น Output

ค่าน้ำหนักแต่ละตัวที่เชื่อมโยงจาก Neuron ใน Hidden Layer มายังชั้น Output คือ W_{jk}^H จึงจะมีการปรับค่าน้ำหนักดังนี้

$$\Delta \mathrm{W}_{jk}^O = \eta(\mathrm{t_k} - \mathrm{y_k})\mathrm{f'}\,\mathrm{x_{jk}^O} \tag{3.2}$$

แต่ Input ของชั้น Output คือ Output ของชั้น Hidden Layer ดังนั้น

$$\Delta \mathrm{W}_{jk}^O = \eta(\mathrm{t_k} - \mathrm{y_k})\mathrm{f'}\,\mathrm{y_{jk}^H} \tag{3.3}$$

เมื่อ f' คือ derivative ของ transformation function จากชั้น Hidden Layer สู่ชั้น Output

ลิขสิทธิ์มหาวิทยาลัยเชียงใหม่
Copyright© by Chiang Mai University
All rights reserved

การปรับค่าน้ำหนักในชั้น Hidden Layer

ค่าน้ำหนักแต่ละตัวที่เชื่อมโยงจาก input มายัง Neuron ใน Hidden Layer คือ W_{ij}^{H} จึงจะมีการปรับน้ำหนักดังนี้

$$\Delta W_{ij}^{H} = \eta(t_j - y_j)g' x_{ij}^{H} \tag{3.4}$$

เมื่อ g' คือ derivative ของ transformation function จากชั้น Input สู่ชั้น Hidden Layer (อรทัย เมืองใจ, 2548: 22)

แต่ปัญหาที่พบก็คือ เราไม่สามารถหาค่า t_j ได้เพราะว่าในชั้น Hidden layer เราไม่ทราบค่าของ y_j ที่นิวรอลแต่ละตัวส่งออกมาว่าผิดหรือถูกเราจึงไม่สามารถหา error ได้ดังนั้นจึงได้มีการสร้างค่า error เทียมออกมาโดยสมมติให้ นิวรอลที่ส่งข้อมูลมาชั้นผลลัพธ์ (out put) มากก็ต้องรับผิดชอบค่า error ที่เกิดขึ้นจาก (t-y) มากตาม ดังนั้นความรับผิดชอบสำหรับค่า Output ที่นิวรอลในชั้น Hidden layer ตัวที่ j จะเป็นผลรวมของค่ารับผิดชอบของ Output ทั้งหมดจึงสามารถแก้ปัญหาการปรับค่าน้ำหนักในชั้น Hidden Layer ได้ดังนี้

$$\Delta W_{ij}^{H} = \eta\left(t_j - y_j\right)g' x_{ij}^{H} \tag{3.5}$$

โดยที่ $t_j - y_j = \sum_{k=1}^{m} W_{jk}^{0}\left(t_k - y_k\right)f'$ ซึ่งเป็นผลรวมของความรับผิดชอบที่มีต่อ Output แต่ละตัว

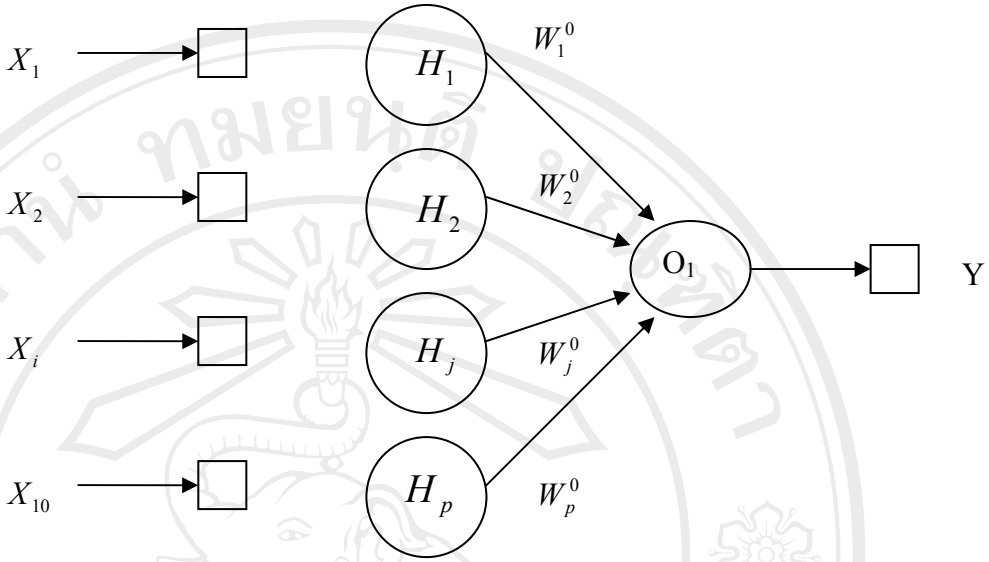
ดังนั้นจึงได้กฎการปรับค่าน้ำหนักจากชั้น Input มายังชั้น Output ดังนี้

$$\Delta W_{ij}^{H} = \eta\left(\sum_{k=1}^{m} W_{jk}^{0}\left(t_k - y_k\right)f'\right)g' x_{ij}^{H} \tag{3.6}$$

ด้วยแนวคิดดังกล่าว จึงเขียนเป็นแผนผังได้ดังนี้

ลิขสิทธิ์มหาวิทยาลัยเชียงใหม่
Copyright© by Chiang Mai University
All rights reserved

รูปที่ 3.2 น้ำหนักจากชั้น Hidden Layer สู่ชั้น Output



จากรูปจะเห็นว่า ความคลาดเคลื่อนของ Output แต่ละตัว สามารถแบ่งความรับผิดชอบไปยัง Neuron ใน Hidden Layer ทุก ๆ ตัวได้โดยผ่านค่าน้ำหนัก ซึ่งหาก Neuron ใน Hidden Layer ตัวใดมีค่าน้ำหนักมากก็ย่อมต้องรับผิดชอบกับความคลาดเคลื่อนที่เกิดขึ้นมาก

ดังนั้น ความคลาดเคลื่อนที่ Neuron ตัวที่ j ใน Hidden Layer จะต้องรับผิดชอบสำหรับ Output ตัวที่ 1 ก็คือ $W_{jl}^0(t_1 - y_1)f'$ อนึ่ง การที่ติดค่า derivative ของ transformation function มาด้วยนั้นก็เพราะการเปลี่ยนแปลงรูปแบบของ transformation function ย่อมมีผลกระทบกับค่าน้ำหนักด้วย

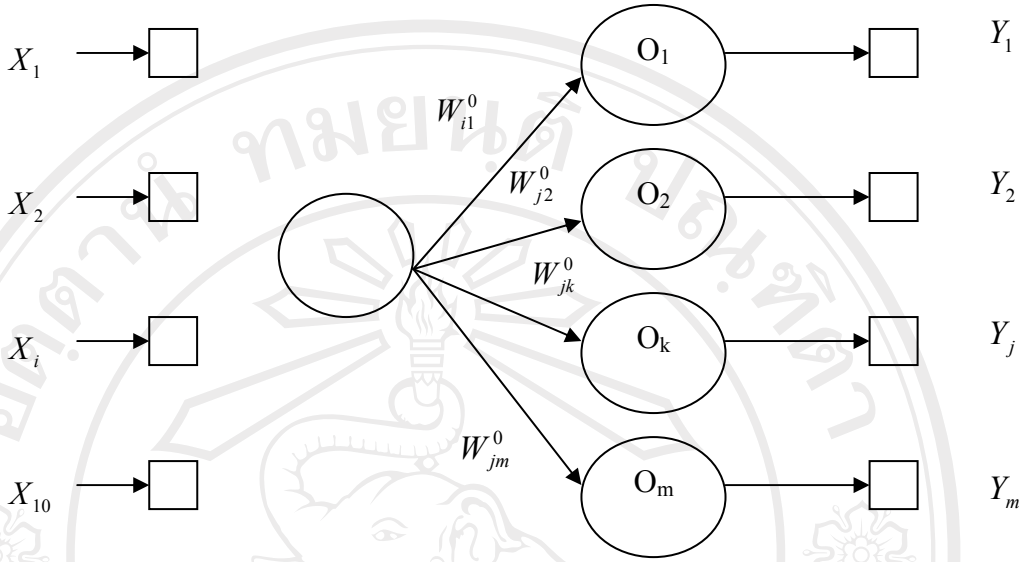
ความรับผิดชอบรวมทั้งหมดที่ Neuron ตัวที่ j ใน Hidden Layer จะต้องรับผิดชอบสำหรับ Output ทุกตัว ก็ย่อมเท่ากับ ผลบวกของความรับผิดชอบที่จะต้องมีต่อ Output แต่ละตัวนั่นเอง ดังนี้

$$\sum_{k=1}^{m} W_{jk}^0(t_k - y_k)f' \tag{3.7}$$

ซึ่งเขียนเป็นแผนภาพได้ดังนี้

ลิขสิทธิ์มหาวิทยาลัยเชียงใหม่
Copyright© by Chiang Mai University
All rights reserved

รูปที่ 3.3 ความรับผิดชอบของ Neuron ตัวที่ j ในชั้น Hidden Layer ที่มีต่อ Output



เมื่อนำเอาความรับผิดชอบที่ Neuron ตัวดังกล่าวต้องรับผิดชอบต่อ Output ทั้งหมดมาเป็น error เทียม จะทำให้สามารถแก้ปัญหาการปรับค่าน้ำหนักในชั้น Hidden Layer ได้ดังนี้

$$\Delta W_{ij}^{H} = \eta\left(t_j - y_j\right)g' x_{ij}^{H} \tag{3.8}$$

แต่ $t_j - y_j = \sum_{k=1}^{m} W_{jk}^{0}\left(t_k - y_k\right)f'$

ดังนั้นจึงจะได้กฎการปรับค่าน้ำหนักจากชั้น Input มายังชั้น Output ดังนี้

$$\Delta W_{ij}^{H} = \eta\left(\sum_{k=1}^{m} W_{jk}^{0}\left(t_k - y_k\right)f'\right)g' x_{ij}^{H} \tag{3.9}$$

การกำหนดจำนวนนิวรอลในชั้นซ่อนเร้น

การค้นหาแบบจำลองที่ดีที่สุด ใช้การเปลี่ยนจำนวนนิวรอลในชั้นซ่อนเร้น ด้วยหลักการหาค่าต่ำสุดแบบ Quadratic interpolation ดังนี้

Quadratic interpolation เป็นการหาแบบจำลองที่ดีที่สุด ในการกำหนดจำนวนนิวรอลในชั้นซ่อนเร้นโดยมีวิธีการหาค่าต่ำสุดด้วยวิธี Quadratic Interpolation ซึ่งจะทำได้เมื่อมีข้อมูลจำนวน 3 ชุดและมั่นใจว่าข้อมูลทั้งสามชุดอยู่บนฟังก์ชั่น Quadratic เดียวกัน การมีข้อมูล 3 ชุดจะเพียงพอต่อการหา Unique Solution สำหรับฟังก์ชั่น Quadratic ดังนี้

ลิขสิทธิ์มหาวิทยาลัยเชียงใหม่
Copyright© by Chiang Mai University
All rights reserved

เมื่อมีข้อมูลดังต่อไปนี้

ชุดที่ 1 : $\{x, y\}=\{\alpha, f(\alpha)\}$

ชุดที่ 2 : $\{x, y\}=\{\beta, f(\beta)\}$

ชุดที่ 3 : $\{x, y\}=\{\gamma, f(\gamma)\}$

ที่จุด α : $f(\alpha) = a\alpha^2 + b\alpha + c$

ที่จุด β : $f(\beta) = a\beta^2 + b\beta + c$

ที่จุด γ : $f(\gamma) = a\gamma^2 + b\gamma + c$

การแก้สมการเพื่อหา Unique Solution

$$\begin{bmatrix} \alpha^2 & \alpha & 1 \\ \beta^2 & \beta & 1 \\ \gamma^2 & \gamma & 1 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} f(\alpha) \\ f(\beta) \\ f(\gamma) \end{bmatrix}$$

แก้สมการ

$$\begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} \alpha^2 & \alpha & 1 \\ \beta^2 & \beta & 1 \\ \gamma^2 & \gamma & 1 \end{bmatrix}^{-1} \begin{bmatrix} f(\alpha) \\ f(\beta) \\ f(\gamma) \end{bmatrix}$$

กำหนดให้ $\psi = \det \begin{bmatrix} \alpha^2 & \alpha & 1 \\ \beta^2 & \beta & 1 \\ \gamma^2 & \gamma & 1 \end{bmatrix}$

แล้วจะได้ว่า $\psi = (\alpha-\beta)(\beta-\gamma)(\gamma-\alpha)$

นอกจากนั้นยังจะพบว่า

$$a = \frac{1}{\psi}\left[(\gamma-\beta)f(\alpha) + (\alpha-\gamma)f(\beta) + (\beta-\gamma)f(\gamma)\right]$$

$$b = \frac{1}{\psi}\left[(\beta^2-\gamma^2)f(\alpha) + (\gamma^2-\alpha^2)f(\beta) + (\alpha^2-\beta^2)f(\gamma)\right]$$

$$c = \frac{1}{\psi}\left[\beta\gamma(\gamma-\beta)f(\alpha) + \gamma\alpha(\alpha-\gamma)f(\beta) + \alpha\beta(\beta-\gamma)f(\gamma)\right]$$

ลิขสิทธิ์มหาวิทยาลัยเชียงใหม่
Copyright© by Chiang Mai University
All rights reserved

แล้วค่า x ที่ทำให้พบกับจุดต่ำสุดของฟังก์ชั่นสามารถหาได้จากสูตรดังนี้

$$x* = -\frac{b}{2a}$$

ดังนั้น $x^* = \frac{1}{2}\left[\frac{(\beta^2-\gamma^2)f(\alpha)+(\gamma^2-\alpha^2)f(\beta)+(\alpha^2-\beta^2)f(\gamma)}{(\beta-\gamma)f(\alpha)+(\gamma-\alpha)f(\beta)+(\alpha-\beta)f(\gamma)}\right]$

3.1.2 การทดสอบความนิ่งของข้อมูล (Unit Root Test)

การวิเคราะห์ข้อมูลทางเศรษฐศาสตร์ ซึ่งข้อมูลที่ใช้เป็นข้อมูลแบบอนุกรมเวลา (Time series data) มีความจำเป็นที่จะต้องมีการทดสอบว่า ข้อมูลที่เราใช้นั้นมีความนิ่ง (Stationary) หรือไม่ เพราะหากเราละเลยที่จะทดสอบข้อมูล แล้วข้อมูลเกิดมีความไม่นิ่ง (nonstationary) จะทำให้สมการถดถอยระหว่างตัวแปรอนุกรมเวลาสองตัวแปร จะได้ค่า R^2 ที่มีค่าสูงมากและค่าสถิติ t จะมีนัยสำคัญ ทั้งที่ตัวแปรทั้งสองนั้นไม่มีความสัมพันธ์กันเลยในทางเศรษฐศาสตร์ (Ender, 1995: 216; Gujarati, 1995: 709)

เหตุที่ทำให้ได้ค่า R^2 สูงเช่นนี้เป็นเพราะอนุกรมเวลามีแนวโน้ม ไม่ใช่เนื่องจากความสัมพันธ์ที่แท้จริงระหว่างตัวแปรอนุกรมเวลาทั้งสองตัวแปร ส่วนกรณีที่ค่าสถิติ t มีนัยสำคัญเป็นเพราะอนุกรมเวลาทั้งสองมีแนวโน้มที่แข็งแรงมาก (Strong Trend) โดยสรุปแล้วการละเลยการทดสอบความนิ่งของข้อมูลอาจนำไปสู่การลงความเห็นที่ผิดพลาดได้ในที่สุด

ข้อมูลที่มีลักษณะนิ่ง (Stationary) หมายถึง ข้อมูลอนุกรมเวลาที่มีค่าเฉลี่ย (Mean) และความแปรปรวน (Variance) เท่ากันตลอดระยะเวลาที่ศึกษา

ส่วนข้อมูลที่มีลักษณะไม่นิ่ง (nonstationary) หมายถึง ข้อมูลอนุกรมเวลาที่มีค่าเฉลี่ย (Mean) และความแปรปรวน (Variance) ไม่เท่ากันตลอดระยะเวลาที่ศึกษา

ทั้งนี้การวิเคราะห์ข้อมูลที่เป็นอนุกรมเวลาส่วนมากจะพบปัญหาความไม่นิ่งของข้อมูล ซึ่งสามารถแก้ไขได้ด้วยการทำให้ข้อมูลมีความนิ่งเสียก่อน โดยอาจใช้วิธีการหาผลต่าง (Difference) ของข้อมูล การแปลงให้อยู่ในรูป Logarithm หรือการทดสอบหาความสัมพันธ์ของตัวแปรในระยะยาว (cointegration) เป็นต้น

ในการการทดสอบ unit root หรือ อันดับความสัมพันธ์ของข้อมูล (order of Integration) เป็นการทดสอบตัวแปรทางเศรษฐกิจต่างๆ ที่ใช้ในสมการเพื่อดูความเป็น station (I(0); integrated of order 0) หรือ non-stationary โดยส่วนมากแล้วจะนิยมการทดสอบโดยวิธี Dicky-Fuller test ซึ่งสามารถแบ่งออกได้เป็น 2 วิธี คือ

ลิขสิทธิ์มหาวิทยาลัยเชียงใหม่
Copyright© by Chiang Mai University
All rights reserved

วิธีที่ 1 Dickey-Fuller Test (DF)

วิธีนี้จะทำการทดสอบตัวแปรที่เคลื่อนไหวไปตามช่วงเวลาที่มีลักษณะเป็น autoregressive model โดยพิจารณาสมการ 3 รูปแบบที่แตกต่างกัน ดังนี้

$\Delta X_t = \theta x_{t-1} + \varepsilon_t$ (random walk process)

$\Delta X_t = \alpha + \theta x_{t-1} + \varepsilon_t$ (random walk with drift)

$\Delta X_t = \alpha + \beta_t \theta x_{t-1} + \varepsilon_t$ (random walk with drift and linear time trend)

โดยที่ ΔX_t คือ ค่าความแตกต่างครั้งที่ 1 ของตัวแปรที่ทำการศึกษา

α, β, θ คือ ค่าคงที่

t คือ ค่าคงที่

ε_t คือ ตัวแปรสุ่มที่มีค่าเฉลี่ยเท่ากับศูนย์ และค่าความแปรปรวนที่คงที่ หรือ $\varepsilon_t \sim$ iid $(0, \delta_\varepsilon^2)$

การทดสอบ จะพิจารณาค่า θ โดยเปรียบเทียบค่าสถิติ t (t-statistic) ที่คำนวณได้กับค่าที่เหมาะสมจากตาราง Dickey-Fuller ซึ่งมีสมมติฐานการทดสอบ ดังนี้

$H_0: \theta = 0$: non-stationary

$H_1: \theta < 0$: stationary

ถ้ายอมรับ $H_0: \theta = 0$ จะได้ว่า ตัวแปรที่สนใจ (X_t) มี unit root หรือ X_t มีลักษณะเป็น nonstationary

แต่ถ้ายอมรับ $H_1: \theta \neq 0$ จะได้ว่าตัวแปรที่สนใจ (X_t) ไม่มีunit root หรือ X_t มีลักษณะเป็น stationary

วิธีที่ 2 Augmented Dickey-Fuller Test (ADF)

เป็นการทดสอบ unit root อีกวิธีหนึ่งที่พัฒนามาจาก DF Test เนื่องจากวิธี DF ไม่สามารถทำการทดสอบตัวแปรในกรณีที่เป็น serial correlation ในค่าความคลาดเคลื่อน error term (ε_t) ที่มีลักษณะความสัมพันธ์กันเองในระดับสูง โดยมีสมการดังนี้

$$\Delta X_t = \theta x_t + \sum_{j=1}^{p} \phi_j \Delta X_{t-j} + \varepsilon_t$$

$$\Delta X_t = \alpha + \theta x_t + \sum_{j=1}^{p} \phi_j \Delta X_{t-j} + \varepsilon_t$$

$$\Delta X_t = \alpha + \beta t \theta x_t + \sum_{j=1}^{p} \phi_j \Delta X_{t-j} + \varepsilon_t$$

ลิขสิทธิ์มหาวิทยาลัยเชียงใหม่
Copyright© by Chiang Mai University
All rights reserved

การใส่ค่า Lagged Term (ρ) ว่ามีจำนวนเท่าใดจึงจะเหมาะสมสำหรับแต่ละข้อมูลอนุกรมนั้นมีหลักในการเลือก lag length ที่เสนอโดย Walter Ender (1995) ว่า ควรจะเริ่ม lag length ที่มีค่าที่มากพอ แล้วพิจารณาความมีนัยสำคัญทางสถิติ ที่ระดับนัยสำคัญต่างๆ ($\sigma = 0.01, 0.05$ และ 0.1) เมื่อพบว่าที่ lag length ที่เลือกมีค่า t – statistic ที่ไม่มีนัยสำคัญทางสถิติ ณ ระดับนัยสำคัญ ร้อยละ 10 แล้วจึงทำการลด lag length ลงที่ละ 1 ช่วง จนกระทั่งสามารถปฏิเสธสมมติฐานว่าง (Null Hypothesis)

3.1.3 แบบจำลอง ARIMA

แบบจำลอง ARIMA เป็นแบบจำลองที่ใช้ในการวิเคราะห์ข้อมูลอนุกรมเวลา (Time series data) ที่ได้รับความนิยมและมีการพัฒนาปรับปรุงเป็น ARCH, GARCHและ EGARCH

การพยากรณ์ด้วยแบบจำลอง ARIMA ด้วยวิธีการ Box and Jenkins เป็นวิธีการพยากรณ์ค่าในอนาคตที่พัฒนาโดยนักสถิติผู้มีชื่อเสียงสองท่านคือ George E.P. Box และ Gwilym M. Jenkins ในปี ค.ศ. 1970 ซึ่งเป็นวิธีที่ให้ค่าพยากรณ์ที่ดี คือ ค่าเฉลี่ยความคลาดเคลื่อนกำลังสอง (Mean Square Error) ค่อนข้างต่ำ

การพยากรณ์อนุกรมเวลาโดยวิธี Box and Jenkins (ทรงศิริ แต้สมบัติ, 2539) ในรูปแบบ ARIMA (p,d,q) ต้องพิจารณาข้อมูลอนุกรม Y ว่ามีคุณสมบัติ stationary หรือไม่เสียก่อน โดยพิจารณาได้จาก

1) ค่าเฉลี่ย $E(Y_t)$ คงที่ สำหรับทุกค่าของ t หรือไม่ ทำได้โดยการแบ่งอนุกรมเวลาออกเป็นส่วนๆแล้วทำการหาค่าเฉลี่ยในแต่ละส่วน ถ้าค่าเฉลี่ยแต่ละส่วนย่อยไม่แตกต่างกันมาก จะสรุปได้ว่า $E(Y_t)$ คงที่

2) ค่าความแปรปรวน [Var(Y_t)] คงที่ สำหรับทุกค่าของ t หรือทำได้โดยการแบ่งอนุกรมเวลาออกเป็นส่วนๆ แล้วทำการหาค่าความแปรปรวนแต่ละส่วน ถ้าค่าความแปรปรวนในแต่ละส่วนไม่มีความแตกต่างกันมากนัก จะสรุปได้ว่า Var(Y_t) คงที่

3) พิจารณาถึงแนวโน้มและปัจจัยของฤดูกาลว่ามีผลต่ออนุกรมเวลาหรือไม่

4) พิจารณา correlogram ของค่า autocorrelation ของตัวอย่าง (R_k) กรณีที่อนุกรมเวลามีความนิ่ง (stationary) ค่า correlogram ของค่า autocorrelation จะมีค่าลดลงค่อนข้างเร็วเมื่อ k เพิ่มมากขึ้น ดังนั้นหากค่า autocorrelation มีค่าลดลงค่อนข้างช้าอธิบายได้ว่าอนุกรมเวลาชุดนี้มีแนวโน้ม (trend) แต่ถ้าค่า autocorrelation มีค่าลดลงค่อนข้างช้าและมีค่าค่อนข้างสูงที่ k = L,2L, 3L จะอธิบายได้ว่าอนุกรมเวลาชุดนี้มีแนวโน้มและอิทธิพลของฤดูกาล และหากการเคลื่อนไหวของ

ลิขสิทธิ์มหาวิทยาลัยเชียงใหม่
Copyright© by Chiang Mai University
All rights reserved

correlogram ของค่า autocorrelation เป็นลักษณะลูกคลื่น โดยลูกคลื่นจะครบรอบในแต่ละช่วงเวลา แสดงว่ามีอิทธิพลของฤดูกาลเข้ามาเกี่ยวข้อง

หลังทำการตรวจสอบ หากพบว่าข้อมูลอนุกรมเวลาไม่มีความนิ่ง (nonstationary) จะต้องแปลงให้นิ่ง (stationary) เสียก่อนจากนั้นจึงเข้าสู่ 4 ขั้นตอนของ Box and Jenkins ดังนี้

1) การกำหนดรูปแบบ (identification) ให้กับอนุกรมเวลาที่มีความนิ่ง (stationary) เป็นการหารูปแบบ ARIMA (p,d,q) ที่คาดว่าเหมาะสมให้กับอนุกรมเวลาโดยที่ autocorrelation: P_k มีค่าอยู่ในช่วง [1,-1] โดยพิจารณาเปรียบเทียบค่า autocorrelation (R_k) ของอนุกรมตัวอย่างกับค่า autocorrelation (P_k) ของอนุกรมเวลาของประชากรที่มีช่วงเวลาย้อนหลังไป k หน่วย

Partial Autocorrelation: R_{kk} คือ การวัดความสัมพันธ์ของแต่ละช่วงเวลาโดยมีช่วงเวลาที่ย้อนหลังไป k หน่วยเวลาโดยพิจารณาเปรียบเทียบค่า Partial Autocorrelation ของอนุกรมเวลาตัวอย่าง (R_{kk}) กับค่า Partial Autocorrelation ของอนุกรมเวลาประชากร (P_{kk}) ที่มีช่วงเวลาย้อนหลังไป k หน่วยเวลา ซึ่งมีสูตรดังนี้

$$R_k = \frac{\sum_{t-a}^{n-k}(Y_t - q)(Y_t + k - q_t)}{\sum_{t-a}^{n}(Y_t - q)^2}$$

การพิจารณาแต่ละรูปแบบ ต้องพิจารณา R_k, R_{kk} กับ P_k, P_{kk} พร้อมกันหลายๆ ค่าจึงมักพิจารณาจากรูปแบบที่เรียกว่า correlogram ที่ได้จากการพล็อต R_k, R_{kk} กับ P_k, P_{kk} ในช่วงเวลาที่ k ด้วยเหตุนี้การเปรียบเทียบรูปแบบจึงเป็นการเปรียบเทียบ correlogram ของค่า autocorrelation(R_k) ของอนุกรมตัวอย่างกับค่า autocorrelation (P_k) ของอนุกรมเวลาของประชากร และ correlogram ของค่า Partial Autocorrelation ของอนุกรมเวลาตัวอย่าง (R_{kk}) กับค่า Partial Autocorrelation ของอนุกรมเวลาประชากร (P_{kk}) สำหรับแต่ละรูปแบบจะมี correlogram ของ P_k และ P_{kk} ต่างกัน อนุกรมเวลาที่จะนำมากำหนดรูปแบบจะต้องเป็นอนุกรมเวลาที่มีความนิ่ง (stationary)เท่านั้น ดังนั้นหากข้อมูลไม่ stationary จะต้องทำให้ stationary ก่อนทุกครั้ง

2) การประมาณค่าพารามิเตอร์ในรูปแบบ Estimator จะทำได้โดยการหาค่าประมาณแบบง่ายหรือค่าประมาณที่ได้จากการวิเคราะห์ตัวเลข สำหรับค่าประมาณแบบง่ายจะทำได้โดยการสร้างสมการที่ได้มาจากความสัมพันธ์ระหว่าง P_k และพารามิเตอร์ โดนสมการที่สร้างขึ้นจะมีจำนวนพารามิเตอร์ที่ต้องการประมาณ ส่วนค่าประมาณที่ได้จากการวิเคราะห์ตัวเลขจะได้จากการแก้สมการที่สร้างขึ้นจากวิธีกำลังสองน้อยที่สุด ในขั้นตอนของการวิเคราะห์ตัวเลขจะต้องมีการ

ลิขสิทธิ์มหาวิทยาลัยเชียงใหม่
Copyright© by Chiang Mai University
All rights reserved

กำหนดค่าประมาณเริ่มต้นซึ่งส่วนใหญ่จะใช้การประมาณแบบง่ายเป็นจุดเริ่มต้น หลังวิเคราะห์เสร็จแล้วจะใช้ค่าประมาณสุดท้ายที่นำไปใช้ประโยชน์ในการสร้างสมการพยากรณ์

3) การตรวจสอบแบบจำลอง (Diagnostic checking) เมื่อกำหนดรูปแบบและประมาณค่าพารามิเตอร์ในแบบจำลอง จะต้องทำการตรวจสอบทุกครั้งว่ารูปแบบที่กำหนดนั้นมีความเหมาะสมจริงหรือไม่ โดยการตรวจสอบสามารถทำได้หลายวิธี ได้แก่ การพิจารณา correlogram ของ R_k หรือของความคลาดเคลื่อน การทดสอบค่าพารามิเตอร์ในแบบจำลองด้วยการทดสอบแบบ t – test และการทดสอบความเหมาะสมของแบบจำลองโดยการทดสอบของ Box และ Pierce หรือ การทดสอบของ Box และ Jung หากตรวจสอบพบว่าแบบจำลองนั้นมีความเหมาะสมแล้ว จะสามารถใช้รูปแบบนั้นพยากรณ์ต่อไป แต่หากแบบจำลองไม่มีความเหมาะสมต้องกลับไปเริ่มขั้นตอนที่ 1 ใหม่

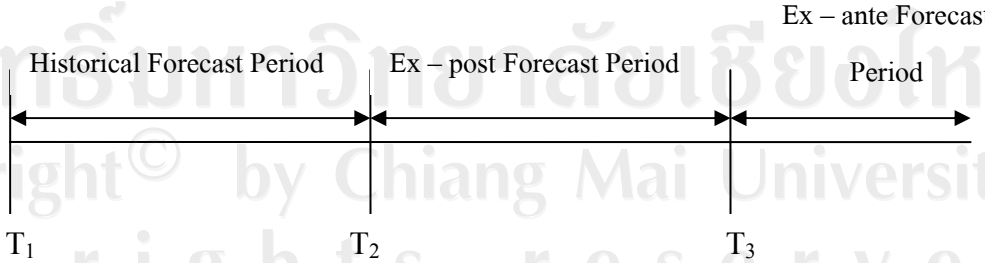
4) การพยากรณ์ (Forecasting) สามารถทำได้ทั้งแบบจุด (Point forecast) และแบบช่วง (Interval forecast) โดยการพยากรณ์จะใช้สมการพยากรณ์ที่สร้างจากรูปแบบการพยากรณ์ที่กำหนดและผ่านการตรวจสอบในขั้นตอนที่แล้วมา

เมื่อได้แบบจำลองที่มีความเหมาะสมหลังจากการตรวจสอบวามถูกต้องแล้วก็สามารถนำแบบจำลองไปใช้ในการพยากรณ์โดยแบ่งการพยากรณ์ออกเป็น 3 ช่วง ดังนี้

1) Historical Forecast – การพยากรณ์ตั้งแต่อดีตจนถึงช่วงเวลาที่เริ่มพิจารณา

2) Ex – post Forecast – การพยากรณ์โดยการตัดข้อมูลออกมาส่วนหนึ่ง แล้วทำการพยากรณ์เปรียบเทียบกับข้อมูลที่ได้จาก การพยากรณ์

3) Ex – ante Forecast – การพยากรณ์ข้อมูลไปข้างหน้า โดยนำแบบจำลองที่หาได้ในช่วงที่ 2 มาทำการพยากรณ์



ลิขสิทธิ์มหาวิทยาลัยเชียงใหม่
Copyright© by Chiang Mai University
All rights reserved

3.1.4 แบบจำลอง GARCH – in – mean (GARCH-M)

$$Y_t = \alpha + \sum_{i=1}^{p} Y_{t-i} + \sum_{j=1}^{q} \epsilon_{t-j} + \gamma \quad (3.10)$$

$$\sigma_t^2 = \alpha + \sum_{i=1}^{q} \alpha_i \varepsilon_{t-i}^2 + \sum_{i=1}^{p} \beta_i \sigma_{t-i}^2 \quad (3.11)$$

จากสมการ (3.10) เรียกว่าสมการค่าเฉลี่ย (Mean Equation) และสมการที่ (3.11) เรียกว่า Generalized Autoregressive Conditional Heteroscedasticity: GARCH (p,q) โดยแบบจำลอง GARCH–M นั้น เป็นแบบจำลองที่ได้นำเอาค่าจากสมการ GARCH(p,q) ที่ได้ทำการยกกำลังเศษหนึ่งส่วนสองแล้วนำไปใส่ในสมการ Mean Equation เพื่อทำการประมาณค่าของตัวแปรที่ถูกศึกษา

3.2วิธีการวิจัย

วิธีการวิจัยแบ่งได้ออกเป็น 3 ส่วนคือ วิธีการศึกษาด้วยแบบจำลองนิวรอลเน็ตเวิร์ค, วิธีการศึกษาด้วยแบบจำลอง GARCH –M และวิธีการศึกษาด้วยแบบจำลอง ARIMA ซึ่งรายละเอียดดังต่อไปนี้

3.2.1 วิธีการศึกษาด้วยแบบจำลองนิวรอลเน็ตเวิร์ค

การศึกษาด้วยแบบจำลองนิวรอลเน็ตเวิร์คประกอบไปด้วย 3 การทดลอง ดังนี้คือ

1) การทดลองเบื้องต้นด้วย Hidden Layer จำนวน 1 ชั้น

2) การทดลองปรับจำนวน Hidden Layer ให้เป็น 2 ชั้น

3) การทดลองปรับเปลี่ยนจำนวนนิวรอลใน Hidden Layer ด้วยวิธี Quadratic Interpolation

รายละเอียดของแต่ละการทดลอง มีดังนี้

1) การทดลองเบื้องต้นด้วย Hidden Layer จำนวน 1 ชั้น

ขั้นที่1 แบ่งข้อมูลที่มีอยู่ออกเป็น 3 ช่วง คือ Training 450 ข้อมูล, Validation 50 ข้อมูล และ Testing 50 ข้อมูล

ขั้นที่ 2 จัดเรียงข้อมูล เพื่อกำหนด input ให้เท่ากับ 10 ตัว

ขั้นที่ 3 กำหนดจำนวนนิวรอลในชั้น Hidden Layer เท่ากับ 10 โดยให้แบบจำลองเรียนรู้จากข้อมูลชุด Training จำนวน 100 รอบ และคำนวณค่า MSE ดังสมการที่ (3.12) ออกมาด้วยการทำนายค่าของชุดข้อมูล Validation

ลิขสิทธิ์มหาวิทยาลัยเชียงใหม่
Copyright© by Chiang Mai University
All rights reserved

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - t_i)^2 \quad (3.12)$$

โดยที่ y_i = ค่าพยากรณ์

t_i = ค่าจริง

n = จำนวนข้อมูลของชุดข้อมูล Validation

ขั้นที่ 4 ให้แบบจำลองเรียนรู้เพิ่มอีก 100 รอบ จากนั้นจึงคำนวณค่า MSE ออกมาอีกครั้ง

ขั้นที่ 5 ทำซ้ำครั้งที่ 4 จนกระทั่งครบ 1000 รอบ

ขั้นที่ 6 เปลี่ยนจำนวนนิวรอลในชั้น Hidden Layer ให้เท่ากับ 20 แล้วทำซ้ำขั้นที่ 4 และขั้นที่ 5

ขั้นที่ 7 ทำซ้ำขั้นที่ 6 โดยเปลี่ยนจำนวนนิวรอลเป็น 30, 40, 50, 60, 70, 80, 90 และ 100

ขั้นที่ 8 เปรียบเทียบค่า MSE ที่ได้ทั้งหมด แล้วเลือกแบบจำลองที่ดีที่สุดด้วยค่า MSE ที่น้อยที่สุด

ขั้นที่ 9 ใช้แบบจำลองที่ดีที่สุดพยากรณ์ค่าของชุดข้อมูล Testing และคำนวณค่า MAPE ดังสมการที่ (3.13) ออกมา

$$MAPE = \frac{1}{n}\sum_{i=1}^{n} APE_i \quad (3.13)$$

$$APE_i = \frac{|t_i - y_i|}{t_i} \times 100$$

โดยที่ t_i = ค่าจริง

y_i = ค่าพยากรณ์

n = จำนวนข้อมูลของชุดข้อมูล Testing

หมายเหตุ : โครงสร้างของแบบจำลองนิวรอลเน็ตเวิร์คที่ใช้

จำนวนชั้น Hidden Layer	: 1
จำนวนนิวรอลในชั้นแสดงผล	: 1
ฟังก์ชั่นแปลงค่าในชั้นซ่อนเร้น	: purelin
ฟังก์ชั่นแปลงค่าในชั้นแสดงผล	: purelin
วิธีการเรียนรู้	: traincgf
โปรแกรมที่ใช้	: Matlab 6.5

ลิขสิทธิ์มหาวิทยาลัยเชียงใหม่
Copyright© by Chiang Mai University
All rights reserved

2) การทดลองปรับจำนวน Hidden Layer ให้เป็น 2 ชั้น

ขั้นที่ 1 จากการทดลองเบื้องต้นด้วย Hidden Layer จำนวน 1 ชั้น นำค่าของจำนวนนิวรอลในชั้น Hidden Layer ที่ดีที่สุดมากำหนดไว้ให้เป็นค่าคงที่ แล้วทำการหาจำนวนนิวรอลที่เหมาะสมในชั้น Hidden Layer ที่สองโดยกำหนดให้มีค่าเท่ากับ 10 20 30 40 50 60 70 80 90 และ 100 และหาจำนวนรอบการเรียนรู้ที่เหมาะสม โดยกำหนดให้มีค่าเท่ากับ 100 200 300 400 500 600 700 800 900 และ 1000 รอบ

ขั้นที่ 2 เลือกจำนวนนิวรอลในชั้น Hidden Layer ที่สองและจำนวนรอบการเรียนรู้ที่เหมาะสม โดยพิจารณาจากค่า MSE ที่ต่ำที่สุด แล้วนำไปพยากรณ์ราคาทองคำ

ขั้นที่ 3 นำผลการพยากรณ์ราคาทองคำที่ได้คำนวณค่า MAPE

หมายเหตุ : โครงสร้างของแบบจำลองนิวรอลเน็ตเวิร์คที่ใช้

จำนวนชั้น Hidden Layer	: 2
จำนวนนิวรอลในชั้นแสดงผล	: 1
ฟังก์ชั่นแปลงค่าในชั้นซ่อนเร้นที่ 1	: purelin
ฟังก์ชั่นแปลงค่าในชั้นซ่อนเร้นที่ 2	: purelin
ฟังก์ชั่นแปลงค่าในชั้นแสดงผล	: purelin
วิธีการเรียนรู้	: traincgf
โปรแกรมที่ใช้	: Matlab 6.5

3) การทดลองปรับเปลี่ยนจำนวนนิวรอลใน Hidden Layer ด้วยวิธี Quadratic Interpolation

ขั้นที่ 1 กำหนดจำนวนข้อมูลนำเข้าให้เท่ากับ 10

ขั้นที่ 2 กำหนดจำนวนนิวรอลในชั้น Hidden Layer ที่หนึ่งเท่ากับ 10

ขั้นที่ 3 ทำการหาจำนวนนิวรอลในชั้น Hidden Layer ที่สอง ด้วยวิธี Quadratic Interpolation โดยกำหนดให้จำนวนนิวรอลในชั้น Hidden Layer ที่สองมีค่าเท่ากับ 10 30 และ 50

ขั้นที่ 4 เลือกจำนวนนิวรอลในชั้น Hidden Layer ที่สองที่ให้ค่า MSE ที่ต่ำที่สุดเพื่อใช้ในการคำนวณ Quadratic Interpolation ในขั้นต่อไป จนกว่าจะได้ค่า MSE ของจำนวนนิวรอลในชั้น Hidden Layer ที่สองที่มีค่า MSE ต่ำที่สุด

ขั้นที่ 5 เมื่อได้จำนวนนิวรอลในชั้น Hidden Layer ที่สองที่เหมาะสมแล้ว ทำการกำหนดให้มีค่าคงที่ แล้วทำการหาจำนวนนิวรอลในชั้น Hidden Layer ที่หนึ่งเช่นเดียวกับการหาจำนวนนิวรอลในชั้น Hidden Layer ที่สอง

ลิขสิทธิ์มหาวิทยาลัยเชียงใหม่
Copyright© by Chiang Mai University
All rights reserved

ขั้นที่ 6 เมื่อได้จำนวนนิวรอลในชั้น Hidden Layer ที่หนึ่งและชั้นที่สองที่เหมาะสมแล้ว ให้ทำการหาจำนวนข้อมูลนำเข้าโดยวิธี Quadratic Interpolation โดยกำหนดให้จำนวนข้อมูลนำเข้ามีค่าระหว่าง 10 ถึง 50

ขั้นที่ 7 เมื่อได้จำนวนนิวรอลในชั้น Hidden Layer ที่1, ชั้นที่สองและจำนวนข้อมูลนำเข้าที่เหมาะสมแล้ว จึงนำไปพยากรณ์ราคาทองคำและคำนวณหาค่า MAPE

หมายเหตุ : โครงสร้างของแบบจำลองนิวรอลเน็ตเวิร์คที่ใช้

จำนวนชั้น Hidden Layer	: 2
จำนวนนิวรอลในชั้นแสดงผล	: 1
ฟังก์ชั่นแปลงค่าในชั้นซ่อนเร้นที่ 1	: purelin
ฟังก์ชั่นแปลงค่าในชั้นซ่อนเร้นที่ 2	: purelin
ฟังก์ชั่นแปลงค่าในชั้นแสดงผล	: purelin
วิธีการเรียนรู้	: traincgf
โปรแกรมที่ใช้	: Matlab 6.5

3.2.2 วิธีการศึกษาด้วยแบบจำลอง ARIMA

ขั้นที่ 1 ทดสอบความนิ่งของข้อมูลด้วยวิธี Unit Root Test

ขั้นที่ 2 กำหนดรูปแบบของแบบจำลอง ARIMA (p,d,q) โดยการพิจารณาจาก Correlogram เพื่อจะสามารถระบุได้ว่าแบบจำลองควรจะมี Autoregressive (q) เท่าใด และ Moving average (q) เท่าใด

ขั้นที่ 3 การประมาณค่าพารามิเตอร์ เพื่อนำค่าพารามิเตอร์ไปทำการพยากรณ์ราคา

ขั้นที่ 4 เปรียบเทียบแบบจำลองเพื่อนำแบบจำลองที่เหมาะสมที่สุดไปใช้ในการพยากรณ์

ขั้นที่ 5 พยากรณ์ราคาทองคำ 50 วัน

3.2.3 วิธีการศึกษาด้วยแบบจำลอง GARCH –M

ขั้นที่ 1 จากรูปแบบของแบบจำลอง ARIMA ในหัวข้อที่ 3.2.2 ที่เหมาะสมต่อการพยากรณ์ นำแบบจำลองที่ได้มาประมาณค่าพารามิเตอร์ โดยวิเคราะห์ตามวิธีแบบจำลอง GARCH-M เพื่อหารูปแบบที่เหมาะสมที่จะใช้ในการพยากรณ์

ขั้นที่ 2 ตรวจสอบความถูกต้องของแบบจำลองโดยพิจารณาจากค่า Q-Statistic

ขั้นที่ 3 พยากรณ์ราคาทองคำ 50 วัน

ลิขสิทธิ์มหาวิทยาลัยเชียงใหม่
Copyright© by Chiang Mai University
All rights reserved