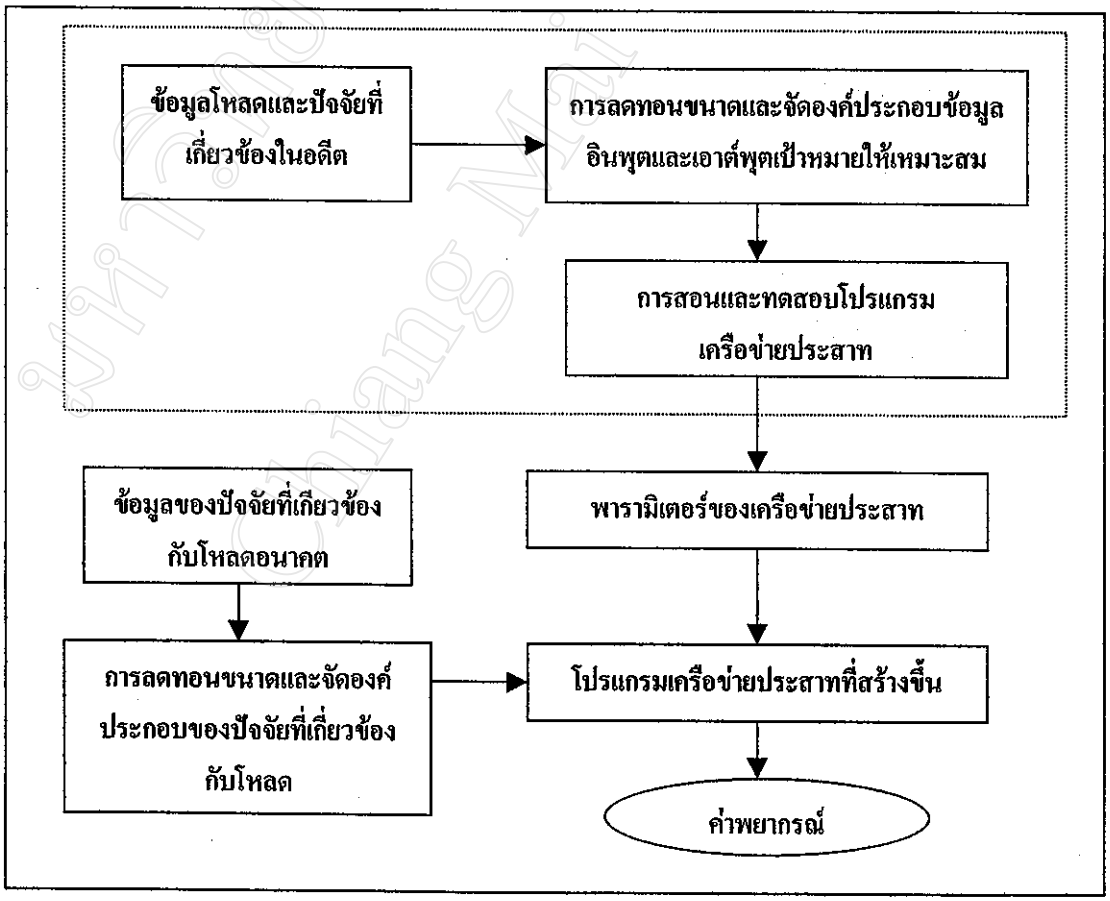
บทที่ 4
โปรแกรมคอมพิวเตอร์ในการพยากรณ์ความต้องการไฟฟ้า

4.1 โปรแกรมคอมพิวเตอร์เครือข่ายประสาท

การสร้างแบบจำลองเครือข่ายประสาทที่จะใช้ในการทดลองพยากรณ์ความต้องการไฟฟ้า จะอาศัยโปรแกรม MatLab ซึ่งเป็นภาษาคอมพิวเตอร์ที่เหมาะสมกับการคำนวณ โดยที่โปรแกรม MatLab มีการสร้างไฟล์ของฟังก์ชันเกี่ยวกับเครือข่ายประสาท (neural network toolbox) ไว้แล้วพอสมควร ดังนั้นจึงเป็นการสะดวกที่จะนำมาใช้ในงานวิจัย การเขียนโปรแกรมจะแบ่งเป็นสองส่วน คือ ส่วนการสอนและการเรียนรู้ (training and learning phase) กับ ส่วนของการทดลองหาคำตอบ (recall phase) แสดงรายละเอียดแต่ละส่วนเป็นแผนผังในรูปที่ 4.1 ซึ่งจะได้อธิบายถึงวิธีการเขียนโปรแกรมแต่ละส่วนต่อไป



รูปที่ 4.1 แผนผังแสดงโครงสร้างของโปรแกรมเครือข่ายประสาทที่ใช้พยากรณ์ความต้องการไฟฟ้า

แบบจำลองที่จะสร้างขึ้นแยกเป็นสามแบบ คือ แบบจำลองสำหรับพยากรณ์ในฤดูร้อน ในฤดูหนาว และในฤดูฝน โดยให้เอาต์พุตเป็นค่าความต้องการไฟฟ้าสูงสุดและต่ำสุด จากนั้นจึงนำมาแปลงเป็นความต้องการไฟฟ้ารายชั่วโมงด้วยค่าแฟคเตอร์รูปแบบ

4.1.1 การเตรียมข้อมูลโหลดและปัจจัยที่เกี่ยวข้องในอดีต

จากที่ได้กล่าวมาแล้วในบทก่อนว่า ข้อมูลที่จะนำมาใช้ในการการพยากรณ์ความต้องการไฟฟ้าในอนาคต ได้แก่ โหลดในอดีตและอุณหภูมิ แต่เนื่องจาก การพยากรณ์ความต้องการไฟฟ้าจะแยกเป็นเป็นฤดูกาล และประเภทของวัน ตามลักษณะของแฟคเตอร์รูปแบบที่ได้หาไว้แล้ว ดังนั้น ข้อมูลที่จะใช้เป็นอินพุตสำหรับเครือข่ายประสาท จึงประกอบด้วย

ก. โหลดสูงสุด (Lp_1) และโหลดต่ำสุด (Lv_1) ของวันก่อนวันที่จะพยากรณ์ ซึ่งเป็นค่าจำนวนจริงทศนิยมหนึ่งตำแหน่ง

ข. อุณหภูมิต่ำสุด(Tmi_1) และอุณหภูมิสูงสุด (Tma_1) ของวันก่อนที่จะพยากรณ์ ซึ่งเป็นค่าจำนวนจริงทศนิยมหนึ่งตำแหน่ง

ค. อุณหภูมิต่ำสุด (Tmi) และสูงสุด (Tma) ของวันที่จะพยากรณ์ เป็นจำนวนจริงทศนิยมหนึ่งตำแหน่ง

ง. ดัชนีแสดงฤดูกาล ด้วยเลขไบนารี (binary) ขนาดสามบิต (bit) คือ ฤดูฝน (1 0 0) ฤดูหนาว (0 1 0) และแทนฤดูร้อน (0 0 1)

จ. ดัชนีแทนประเภทวัน ด้วยเลขไบนารีขนาดสามบิต คือ วันทำงาน (1 0 0) วันหยุดประจำสัปดาห์ (0 1 0) และวันหยุดพิเศษ (0 0 1)

ฉ. ดัชนีแสดงวัน ใช้เลขไบนารีเข้ารหัส ขนาดสามบิต แทนดังนี้

วันจันทร์ (0 0 1) วันอังคาร (0 1 0) วันพุธ (0 1 1) วันพฤหัสบดี (1 0 0)
วันศุกร์ (1 0 1) วันเสาร์ (1 1 0) วันศุกร์ (1 1 1)

ดังนั้น ส่วนอินพุตของเครือข่ายประสาทจึงประกอบด้วย นิวรอนอินพุตจำนวน 15 นิวรอน โดยแต่ละนิวรอนใช้สำหรับสัญญาณอินพุตต่างๆ ข้างต้น สำหรับข้อมูลอินพุตที่จะใช้ในการสอน จะเลือกข้อมูลอุณหภูมิและโหลดในอดีต สองสัปดาห์ (14 วัน) จึงมีแบบแผนหรือแพทเทอร์น (pattern) ที่ใช้ในการสอนแต่ละแบบจำลอง เป็น 14 แพทเทอร์น ดังนี้

(1) ข้อมูลอินพุตสำหรับฤดูหนาว คือ วันที่ 1 ถึงวันที่ 14 กุมภาพันธ์ 2543

(2) ข้อมูลอินพุตสำหรับฤดูร้อน คือ วันที่ 1 ถึง วันที่ 14 เมษายน 2543

(3) ข้อมูลอินพุตสำหรับฤดูฝน คือ วันที่ 18 ถึงวันที่ 30 กันยายน และวันที่ 1 ตุลาคม 2543

ข้อมูลอินพุตทั้ง 14 แพทเทอร์น จะต้องจัดทำอยู่ในรูปเมตริกซ์ (matrix form) ซึ่งแต่ละแพทเทอร์น จะอยู่ในแนวสดมภ์ (column) ของเมตริก ตัวอย่างแพทเทอร์นข้อมูลอินพุตของฤดูฝน ดังรูปที่ 4.2 จะเห็นได้ว่า ขนาดของเมตริกซ์ เป็น 15 x 14 โดยที่จำนวนแถวเป็น 15 คือ จำนวนอินพุตนิวรอน ส่วนจำนวนสดมภ์เป็น 14 คือ จำนวนแพทเทอร์น

```
p=[1  1  1  1  1  1  1  1  1  1  1  1  1  1;
   0  0  0  0  0  0  0  0  0  0  0  0  0  0;
   0  0  0  0  0  0  0  0  0  0  0  0  0  0;
   1  1  1  1  1  0  0  1  1  1  1  1  0  0;
   0  0  0  0  0  1  1  0  0  0  0  0  1  1;
   0  0  0  0  0  0  0  0  0  0  0  0  0  0;
   0  0  0  1  1  1  1  0  0  0  1  1  1  1;
   0  1  1  0  0  1  1  0  1  1  0  0  1  1;
   1  0  1  0  1  0  1  1  0  1  0  1  0  1;
   31.0 31.1 31.8 31.5 30.5 32.1 32.2 32.3 29.1 31.5 31.7 32.0 28.8 28.3;
   20.8 21.4 22.0 22.2 23.8 22.3 22.0 24.0 23.4 22.2 22.9 22.9 22.7 23.3;
   25.6 30.9 30.3 31.0 30.8 32.2 29.0 28.0 29.5 31.2 30.4 29.7 27.7 25.2;
   13.5 13.6 13.9 14.8 14.8 14.2 14.5 15.3 14.6 15.0 15.0 15.8 13.5 13.7;
   31.1 31.8 31.5 30.5 32.1 32.2 32.3 29.1 31.5 31.7 32.7 28.8 28.3 30.0;
   21.4 22.0 22.2 23.8 22.3 22.0 24.0 23.4 22.2 22.9 22.9 22.7 23.3 23.8];
```

รูปที่ 4.2 อินพุตสำหรับสอนเครือข่ายประสาทที่ใช้พยากรณ์ความต้องการไฟฟ้าในฤดูฝน

นอกจากนี้ ในกระบวนการเรียนรู้ของเครือข่ายประสาทแบบแพร่กลับความผิดพลาด จะต้องมีค่าเอาต์พุตเป้าหมาย (target output) เพื่อใช้เป็นค่าอ้างอิงสำหรับเอาต์พุตที่ได้จากการสอนและการเรียนรู้แต่ละครั้งด้วย เนื่องจากเอาต์พุตที่ต้องการซึ่งในที่นี้ จะเป็นค่าโหลดสูงสุด และโหลดต่ำสุดของวันที่จะพยากรณ์ ดังนั้น เอาต์พุตจึงมีสองนิวรอน และมีจำนวนแพทเทอร์นที่ใช้เท่ากันกับแพทเทอร์นของอินพุต คือ 14 แพทเทอร์น ดังรูปที่ 4.3

```
t=[30.9 30.3 31.0 30.8 32.2 29.0 28.0 29.5 31.2 30.4 29.7 27.7 25.2 25.7;
   13.6 13.9 14.8 14.8 14.2 14.5 15.3 14.6 15.0 15.0 15.8 13.5 13.7 14.0];
```

รูปที่ 4.3 เมตริกซ์เอาท์พุตเป้าหมายที่ใช้ในการสอนและการเรียนรู้ของเครือข่ายประสาทที่ใช้ในการพยากรณ์ความต้องการไฟฟ้าในฤดูฝน

4.1.2 การจัดข้อมูลให้เหมาะสมกับการสอนและเรียนรู้

ข้อมูลอินพุตรวมทั้งข้อมูลเอาต์พุตเป้าหมาย จะต้องถูกลดทอนขนาดให้เหมาะสมกับเครือข่าย (preprocessing) คือ อยู่ในย่าน (-1 , 1) โดยใช้สมการที่ (4.1)

และ สมการที่ (4.2) ตามลำดับ

$$Pn = [2 (P - Pmin) / (Pmax - Pmin)] - 1 \quad \text{..... (4.1)}$$

$$Tn = [2 (T - Tmin) / (Tmax - Tmin)] - 1 \quad \text{..... (4.2)}$$

เมื่อ Pn และ Tn คือ ค่าอินพุตและเอาต์พุตเป้าหมายที่ลดทอนขนาดแล้ว

P และ T คือ ค่าอินพุตและค่าเอาต์พุตเป้าหมายที่ต้องการลดทอนขนาด

Pmin และ Tmin คือ ค่าอินพุตและค่าเอาต์พุตเป้าหมายที่น้อยที่สุด

Pmax และ Tmax คือ ค่าอินพุตและค่าเอาต์พุตเป้าหมายที่มากที่สุด

คำสั่งใน MabLab ที่ใช้ในการลดทอนขนาดของอินพุตและเอาต์พุตเป้าหมาย คือ premnmx(p,t) ซึ่งนำมาใช้งานเป็นดังรูปที่ 4.4

```
[pn,minp,maxp,tn,mint,maxt]=premnmx(p,t);
```

รูปที่ 4.4 คำสั่งในการลดทอนขนาดของอินพุต และเอาต์พุตเป้าหมาย

4.1.3 การสร้างเครือข่ายประสาทและองค์ประกอบการสอนและเรียนรู้

MatLab มีคำสั่งที่ใช้ในการสร้างเครือข่ายประสาท และกำหนดองค์ประกอบที่ใช้ในกระบวนการสอน และการเรียนรู้ของเครือข่าย ดังแสดงในรูปที่ 4.5 จากตัวอย่างคำสั่งนี้เป็นการกำหนดเครือข่ายแบบหนึ่งชั้นซ่อน โดยในชั้นซ่อนมี 7 นิวรอน และชั้นเอาต์พุตมี 2 นิวรอน ในขณะที่ฟังก์ชันการกระตุ้นที่นิวรอนในชั้นทั้งสองกำหนดให้เป็น ไบโพลาซิกมอยด์ หรือ tansig ส่วนการสอนเป็นแบบป้อนไปข้างหน้ามีการเรียนรู้โดยแพร่กลับความผิดพลาดแบบ gradient descent ที่ปรับค่าอัตราการเรียนรู้ หรือ traingda การเรียนรู้ไวยิ่งขึ้นโดยอาศัยค่าโมเมนตัม โดยการกำหนดให้การเรียนรู้เป็น learngdm

```
net=newff(minmax(pn),[7,2],{'tansig','tansig'},'traingda','learngdm');
```

รูปที่ 4.5 รูปแบบคำสั่งที่ใช้ในการสร้างเครือข่ายประสาทและองค์ประกอบการสอนและการเรียนรู้

องค์ประกอบที่เกี่ยวข้องกับการสอนและการเรียนรู้ของเครือข่ายให้ประสบความสำเร็จ ยังขึ้นกับพารามิเตอร์อื่นๆ เช่น อัตราการเรียนรู้ ค่าโมเมนตัม ค่าเป้าหมายของการสอน เป็นต้น ดังนั้น จะต้องมีคำสั่งที่กำหนดพารามิเตอร์ที่เกี่ยวข้องด้วย ดังนี้

(1) net.trainParam.show คือ จำนวนครั้ง (epoch) ของการประมวลผลก่อนที่จะแสดงผลหนึ่งครั้ง ซึ่งถ้าไม่กำหนดให้ การประมวลผลจะแสดงผลทุก 50 ครั้ง หรือ default = 50

(2) net.trainParam.lr คือ ค่าอัตราการเรียนรู้ มี default = 0.05

(3) net.trainParam.epochs คือ จำนวนครั้งสูงสุดในการสอน มี default =300

(4) net.trainParam.goal คือ สมรรถนะเป้าหมาย (performance goal) ที่ต้องการจากการสอน มี default = 1e-5

(5) net.trainParam.mc คือ ค่าโมเมนตัม มี default = 0.9

(6) net.trainParam.lr_inc คือ อัตราในการเพิ่มขึ้นของอัตราการเรียนรู้ มี default = 1.05

(7) net.trainParam.lr_dec คือ อัตราในการลดลงของอัตราการเรียนรู้ มี default = 0.7

(8) net.trainParam.max_fail คือ ค่า validation failures สูงสุด มี default = 5

(9) net.trainParam.max_perf_inc คือ ค่าสมรรถนะเพิ่มขึ้นสูงสุด มี default = 1.04

(10) net.trainParam.min_grad คือ ค่า gradient ของสมรรถนะน้อยที่สุด มี default = 1e-10

(11) net.trainParam.time คือ เวลาที่ใช้ในการสอนสูงสุด มี default เป็นค่าอนันต์ หรือ infinity

ตัวอย่างการใช้คำสั่งเหล่านี้ แสดงในรูปที่ 4.6 ซึ่งจากรูป บางคำสั่งอาจจะไม่ต้องเขียน ถ้าหากว่า ตรงกับค่า default ที่ตั้งไว้แล้ว หลังจากที่ได้กำหนดพารามิเตอร์ต่างๆแล้ว จะเป็นการกำหนดค่าน้ำหนักเริ่มต้น และไบอัส (bias) ให้แก่เครือข่าย โดยคำสั่ง net = init(net) ซึ่งคำสั่งนี้ จะเป็นการเรียกคำสั่ง net.initFcn มาทำการหาค่าของน้ำหนักและไบอัสเริ่มต้น ตามที่เรากำหนดพารามิเตอร์ไว้ในคำสั่ง net.initParam

ปกติฟังก์ชันที่จะหาค่าน้ำหนักและไบอัสเริ่มต้น จะตั้งไว้เป็น ' initlay' หมายถึง ให้ทำการตั้งค่าน้ำหนักและไบอัสเริ่มต้นของชั้น i ตามคำสั่ง net.layers{i}.initFcn ซึ่งใน

การเรียนรู้แบบแพร่กลับความผิดพลาดนี้ จะตั้งฟังก์ชันนี้ เป็น 'initnw' หมายถึงให้คำนวณน้ำหนักและไบอัสเริ่มต้นของชั้นที่ i ด้วยวิธีการของงิวเย็น –วิดโรว์ ดังที่กล่าวมาในบทก่อน ส่วนเครือข่ายประสาทแบบอื่นจะตั้งไว้เป็น 'initwb' ซึ่งหมายถึง จะหาค่าน้ำหนักและไบอัสเริ่มต้นตามฟังก์ชันของมัน ซึ่งที่นิยม คือ RANDS (การสุ่มค่า) โดยค่าน้ำหนักและไบอัสที่ได้ จะสุ่มค่าระหว่าง –1 กับ 1

```
net.trainParam.show=500;
net.trainParam.lr=0.05;
net.trainParam.epochs=400000;
net.trainParam.goal=1e-3;
net.trainParam.mc=0.8;
net.trainParam.lr_inc=1.05;
net.trainParam.lr_dec=0.7;
net.trainParam.max_fail=5;
net.trainParam.max_perf_inc=1.04;
net.trainParam.min_grad=1e-10;
net.trainParam.time=inf;
```

รูปที่ 4.6 คำสั่งที่ใช้ในการกำหนดพารามิเตอร์เริ่มต้นของเครือข่ายประสาท

```
net.initFcn='initlay';
for i=1:2
   net.layers{i}.initFcn = 'initwb';
end
net.IW{1,1}=[0.7 0.7 0.7 0.7 0.7 0.7 0.7 0.7 0.7 0.7 0.7 0.7 0.7 0.7 0.7;
             0.7 0.7 0.7 0.7 0.7 0.7 0.7 0.7 0.7 0.7 0.7 0.7 0.7 0.7 0.7;
             0.7 0.7 0.7 0.7 0.7 0.7 0.7 0.7 0.7 0.7 0.7 0.7 0.7 0.7 0.7;
             0.7 0.7 0.7 0.7 0.7 0.7 0.7 0.7 0.7 0.7 0.7 0.7 0.7 0.7 0.7;
             0.7 0.7 0.7 0.7 0.7 0.7 0.7 0.7 0.7 0.7 0.7 0.7 0.7 0.7 0.7;
             0.7 0.7 0.7 0.7 0.7 0.7 0.7 0.7 0.7 0.7 0.7 0.7 0.7 0.7 0.7;
             0.7 0.7 0.7 0.7 0.7 0.7 0.7 0.7 0.7 0.7 0.7 0.7 0.7 0.7 0.7];
net.LW{2,1}=[0.2 -0.2 0.2 -0.2 0.2 -0.2 0.2;
             0.2 -0.2 0.2 -0.2 0.2 -0.2 0.2];
net.b{1,1}=[0.2;-0.2;0.2;-0.2;0.2;-0.2;0.2];
net.b{2,1}=[0.2;-0.2];
net=init(net);
```

รูปที่ 4.7 แสดงคำสั่งในการกำหนดค่าน้ำหนักและไบอัสเริ่มต้นเอง

สำหรับการกำหนดฟังก์ชันเราสามารถจะกำหนดค่าน้ำหนักและไบอัสเริ่มต้นเองนั้น จะใช้คำสั่งเป็น net = initlay (net) กรณีนี้ จะต้องมีคำสั่งกำหนดฟังก์ชันที่จะสร้าง น้ำหนักและไบอัสเริ่มต้นตามมาอีกด้วย การเขียนคำสั่งกำหนดค่าน้ำหนักเริ่มต้นเอง แสดงไว้ในรูปที่ 4.7

4.1.4 การสอนให้แก่เครือข่ายประสาทและการหาคำตอบ

การสอนจะใช้คำสั่ง net = train (net, p, t) ซึ่งคำสั่งนี้ จะเรียกเครือข่ายที่ทำการสร้างไว้ มาทำการสอนตามฟังก์ชันที่กำหนด โดยใช้พารามิเตอร์สำหรับการสอนที่กำหนดให้ หรือ ค่า default (ในกรณีถ้าไม่กำหนดให้) จะกำหนดพารามิเตอร์มากน้อยแค่ไหนขึ้นกับฟังก์ชันการสอนที่เลือก ส่วนการดูผลจากการสอน จะใช้การหาคำตอบเพื่อทดสอบสมรรถนะของเครือข่ายที่สอนเสร็จแล้ว จะใช้คำสั่งดังแสดงในรูปที่ 4.8 โดยคำสั่ง sim (net, pn) เป็นการนำอินพุตที่ผ่านการลดทอนขนาดแล้ว คือ pn มาทำการคำนวณในเครือข่าย net ได้ผลลัพธ์ เป็น an ซึ่งผลลัพธ์นี้ จะต้องทำการแปลงกลับไปเป็นค่าจริง (post-processing) ด้วยคำสั่ง postmnmx ได้คำตอบเป็น a

```
an=sim(net,pn);
a=postmnmx(an,mint,maxt);
```

รูปที่ 4.8 คำสั่งการทดสอบหาคำตอบของเครือข่ายประสาท

4.2 การทดลองพยากรณ์โดยใช้เครือข่ายประสาท

การสร้างแบบจำลองเครือข่ายประสาทด้วยโปรแกรมคอมพิวเตอร์ มีปัญหาเกี่ยวกับการเลือกรูปแบบ และการกำหนดพารามิเตอร์ ซึ่งไม่มีกฎเกณฑ์ตายตัว จึงต้องทำการทดลองหลายๆครั้ง (trial) การทดลองที่ดำเนินการ ประกอบด้วย การหาค่าโมเมนตัม จำนวนนิวรอนในชั้นซ่อน และจำนวนชั้นซ่อน ในที่นี้ จะพิจารณาเปรียบเทียบระหว่าง กรณีหนึ่งชั้นซ่อน และกรณีสองชั้นซ่อน โดยที่ กรณีสองชั้นซ่อนนั้น จะแบ่งเป็น แต่ละชั้นซ่อนมีจำนวนนิวรอนเท่ากัน ไม่เท่ากัน โดยการทดสอบสมรรถนะ จะดูจากเปอร์เซ็นต์ความผิดพลาดของค่าโหลดสูงสุด และต่ำสุดที่พยากรณ์ได้

4.2.1 การทดลองเครือข่ายประสาทแบบหนึ่งชั้นซ่อน

มีจุดประสงค์จะหาค่าของโมเมนตัม และจำนวนนิวรอนในชั้นซ่อน โดยทำการเขียนโปรแกรมเครือข่ายประสาท แล้วทดลองสอน โดยใช้ ค่าพารามิเตอร์ดังนี้

(1) อัตราการเรียนรู้ (lr) เป็น 0.05 การสอนเป็นแบบ traingda ซึ่งจะปรับค่า lr เพิ่มขึ้นด้วยอัตรา (lr_inc) เท่ากับ 1.05 และปรับค่า lr ลดลงด้วยอัตรา (lr_dec) เท่ากับ 0.7

(2) สมรรถนะเป้าหมาย (goal) มีค่า MSE เท่ากับ 0.001

(3) ค่าน้ำหนักและไบอัสเริ่มต้น เป็นค่าคงที่ใดๆ

(4) เวลาที่ใช้ในการสอนไม่จำกัด ซึ่งรวมถึงจำนวนครั้ง (epoch) ในการสอน

การทดลองเปลี่ยนค่าของโมเมนตัม 6 ค่า คือ 0.8 , 0.6, 0.5 , 0.4, 0.2 และ 0.0 ที่แต่ละค่าโมเมนตัม เปลี่ยนจำนวนนิวรอนในชั้นซ่อน ตั้งแต่ 1 – 10 นิวรอน โดยข้อมูลที่ใช้สอนเป็น ค่าความต้องการไฟฟ้าและอุณหภูมิในอดีต 42 วัน คือ ฤดูหนาว ระหว่างวันที่ 1 - 14 กุมภาพันธ์ ฤดูร้อนในระหว่าง วันที่ 1 – 14 เมษายน และ ฤดูฝนในระหว่างวันที่ 18 – 31 ตุลาคม 2543 จากนั้นทดสอบสมรรถนะของเครือข่ายประสาทที่ผ่านการสอนด้วย ข้อมูลของโหลดและอุณหภูมิในระหว่างวันที่ 1 – 17 พฤษภาคม 2543 ผลการทดลองเป็นไปดังตารางที่ 4.1 ซึ่งผลการทดสอบ เปรียบเทียบค่าเฉลี่ยของเปอร์เซ็นต์ความผิดพลาดสัมบูรณ์ (mean absolute percentage error) หรือ MAPE จะได้ค่าโมเมนตัมที่เหมาะสม คือ 0.8 ที่จำนวนนิวรอนในชั้นซ่อนเป็น 7 กล่าวคือ MAPE ของค่าโหลดสูงสุด (Lpeak) เป็น 10.34 % และ MAPE ของโหลดต่ำสุด (Lvalley) เป็น 18.19 % ที่โมเมนตัมเท่ากับ 6 แต่ค่าเฉลี่ยของ MAPE กรณีที่จำนวนนิวรอนในชั้นซ่อนเป็น 7 จะน้อยกว่า และข้อสังเกตอีกประการหนึ่ง คือ ค่าโมเมนตัมที่เปลี่ยนไป มีผลน้อยต่อ สมรรถนะของเครือข่ายในการพยากรณ์

4.2.2 การทดลองเครือข่ายประสาทสองชั้นซ่อน

เป็นการทดลองว่า กรณีใช้เครือข่ายประสาทที่มีสองชั้นซ่อนจะให้ผลการพยากรณ์ดีกว่าแบบหนึ่งชั้นซ่อนหรือไม่ ซึ่งในเบื้องต้น กำหนดให้เครือข่ายที่สร้างขึ้น มีจำนวนนิวรอนในชั้นซ่อนเท่ากัน ค่าพารามิเตอร์ของเครือข่ายเหมือนกับกรณีในข้อ 4.2.1 แต่ให้จำนวนนิวรอนในชั้นซ่อนเป็น 4 , 6 , 8 , 10 , 12 ตามลำดับ ได้ผลการทดสอบตามตารางที่ 4.2 จะเห็นว่า ค่าโมเมนตัมที่เหมาะสม คือ 0.0 ให้ค่า MAPE ของ Lpeak เท่ากับ 15.57 % นอกจากนี้ อีกจุดหนึ่ง คือ ที่ ค่าโมเมนตัมเท่ากับ 0.6 จะได้ MAPE ของ Lvalley เท่ากับ 20.50 % โดยเฉลี่ยแล้ว ทั้งสองค่าโมเมนตัมนี้ ให้สมรรถนะในการพยากรณ์ใกล้เคียงกัน จึงตัดสินใจเลือก ค่าโมเมนตัมทั้งสอง ไปใช้ในการทดสอบอีกกรณีหนึ่ง คือ กรณีที่ จำนวนนิวรอนในชั้นซ่อนไม่เท่ากัน ซึ่งให้ผลดังตารางที่ 4.3 ซึ่งพบว่า ค่า MAPE ของ Lpeak แ ละ Lvalley ที่ดีที่สุด เมื่อ จำนวนนิวรอนในชั้นซ่อนที่หนึ่งเป็น 4 และจำนวนนิวรอนในชั้นซ่อนที่สอง เป็น 2 ที่ค่าโมเมนตัม 0.6 และในทำนองเดียวกัน ค่าโมเมนตัมที่เปลี่ยนแปลง มีผลน้อยต่อสมรรถนะของเครือข่ายประสาทแบบสองชั้นซ่อนนี้

อย่างไรก็ตาม เมื่อเปรียบเทียบสมรรถนะของเครือข่ายประสาท ทั้งกรณีหนึ่งชั้นซ่อน และกรณีสองชั้นซ่อนแล้ว พบว่า กรณีหนึ่งชั้นซ่อน มีความแม่นยำในการพยากรณ์ดีกว่า ดังนั้น จะเลือกใช้แบบจำลองเครือข่ายประสาทแบบหนึ่งชั้นซ่อน ที่มีจำนวนนิวรอนในชั้นซ่อนเป็น 7 และค่าโมเมนตัมที่เหมาะสม เป็น 0.8 และไม่มีความจำเป็นต้องมีการเปลี่ยนแปลงค่าโมเมนตัม เนื่องจากมีผลน้อยต่อสมรรถนะของเครือข่ายประสาท

ตารางที่ 4.1 ผลการทดสอบเครือข่ายหนึ่งชั้นซ่อน หาค่าโมเมนตัมและจำนวนนิวรอนในชั้นซ่อน

กลุ่มที่	โมเมนตัม	แบบจำลองที่	จำนวนหน่วยนิวรอนในชั้นซ่อน	mean absolute percentage error (MAPE)	
				Lpeak	Lvalley
1	0.8	1	1	10.67	19.96
		2	2	19.71	23.54
		3	3	19.07	22.85
		4	4	19.08	22.68
		5	5	10.92	18.19
		6	6	10.95	18.19
		7	7	10.34	18.59
		8	8	10.44	18.69
		9	9	10.62	19.08
		10	10	10.69	19.21
2	0.6	1	1	10.67	19.96
		2	2	19.71	23.53
		3	3	19.07	22.83
		4	4	19.08	22.67
		5	5	10.96	18.35
		6	6	10.98	18.32
		7	7	10.34	18.59
		8	8	10.44	18.70
		9	9	10.62	19.09
		10	10	10.69	19.22
3	0.5	1	1	10.67	19.96
		2	2	19.71	23.53
		3	3	19.06	22.82
		4	4	19.08	22.67
		5	5	10.96	18.39
		6	6	10.99	18.35
		7	7	10.34	18.59
		8	8	10.44	18.70
		9	9	10.62	19.09
		10	10	10.69	19.22

ตารางที่ 4.1 (ต่อ)

กลุ่มที่	โมเมนตัม	แบบจำลองที่	จำนวนหน่วยนิวรอนในชั้นซ่อน	mean absolute percentage error (MAPE)	
				Lpeak	Lvalley
4	0.4	1	1	10.67	19.96
		2	2	19.71	23.53
		3	3	19.06	22.81
		4	4	19.08	22.67
		5	5	10.97	18.42
		6	6	10.99	18.37
		7	7	10.34	18.59
		8	8	10.44	18.70
		9	9	10.62	19.09
		10	10	10.69	19.22
5	0.2	1	1	10.67	19.96
		2	2	19.71	23.53
		3	3	19.06	22.80
		4	4	19.08	22.67
		5	5	10.98	18.46
		6	6	10.99	18.39
		7	7	10.34	18.59
		8	8	10.44	18.70
		9	9	10.62	19.09
		10	10	10.69	19.22
6	0	1	1	10.67	19.96
		2	2	19.71	23.52
		3	3	19.05	22.80
		4	4	19.08	22.67
		5	5	10.99	18.49
		6	6	11.00	18.41
		7	7	10.34	18.59
		8	8	10.44	18.70
		9	9	10.62	19.09
		10	10	10.69	19.22

ตารางที่ 4.2 ผลการทดสอบเครือข่ายสองชั้นซ่อน กรณีจำนวนนิวรอนในชั้นซ่อนเท่ากัน

แบบจำลองที่	จำนวนหน่วยในชั้นซ่อน	โหลดสูงสุด (Lpeak) ที่โมเมนตัม					โหลดต่ำสุด (Lvalley) ที่โมเมนตัม				
		0	0.2	0.4	0.6	0.8	0	0.2	0.4	0.6	0.8
1	4	16.21	16.22	16.22	16.24	16.43	21.15	21.15	21.14	21.13	21.16
2	6	16.73	16.73	16.74	16.74	16.75	22.47	22.47	22.47	22.48	22.49
3	8	15.57	15.60	15.65	15.75	16.05	20.68	20.72	20.79	20.93	21.47
4	10	17.56	17.56	17.57	15.91	16.17	24.47	24.49	24.55	22.80	21.84
5	12	16.09	16.05	15.97	15.80	20.36	20.59	20.58	20.56	20.50	29.38

ตารางที่ 4.3 ผลการทดสอบเครือข่ายสองชั้นซ่อน กรณีจำนวนนิวรอนในชั้นซ่อนไม่เท่ากัน

กลุ่มที่	แบบจำลองที่	จำนวนหน่วยนิวรอนในชั้นซ่อน		MAPE ของ Lpeak		MAPE ของ Lvalley	
		ที่ 1	ที่ 2	โมเมนตัม = 0	โมเมนตัม = 0.6	โมเมนตัม = 0	โมเมนตัม = 0.6
1	1	4	2	13.78	13.76	18.37	18.30
	2		6	17.39	17.37	23.15	23.09
	3		8	18.19	18.18	24.46	24.44
	4		10	18.70	18.68	25.24	25.22
	5		12	19.03	19.02	25.75	25.73
2	1	6	2	14.73	14.73	18.33	18.33
	2		4	14.29	14.30	19.38	19.40
	3		8	16.73	16.73	22.41	22.41
	4		10	16.88	16.86	22.58	22.54
	5		12	17.95	17.90	24.36	24.29
3	1	8	2	15.79	15.79	19.63	19.63
	2		4	15.68	15.69	20.06	20.07
	3		6	15.63	15.64	21.33	21.34
	4		10	15.14	14.96	20.16	22.13
	5		12	18.17	18.24	25.81	25.83

ตารางที่ 4.3 (ต่อ)

กลุ่มที่	แบบจำลองที่	จำนวนหน่วยนิวรอนในชั้นซ่อน		MAPE ของ Lpeak		MAPE ของ Lvalley	
		ที่ 1	ที่ 2	โมเมนตัม = 0	โมเมนตัม = 0.6	โมเมนตัม = 0	โมเมนตัม = 0.6
4	1	10	2	16.73	16.74	20.84	20.86
	2		4	16.57	16.51	21.98	22.06
	3		6	16.57	16.58	21.64	21.62
	4		8	16.35	16.37	21.62	21.57
	5		12	15.54	15.13	20.50	20.20
5	1	12	2	16.75	16.75	21.07	21.67
	2		4	16.70	16.70	21.35	21.35
	3		6	16.48	16.48	21.58	21.58
	4		8	17.14	16.91	22.76	22.45
	5		10	18.88	18.96	27.41	27.43

4.2.3 การทดลองเครือข่ายประสาทแบบแยกฤดูกาล

จากรูปแบบ (topology) ของเครือข่ายประสาทที่ได้ จะทำการทดลองเพิ่มเติมเพื่อปรับปรุงสมรรถนะในการพยากรณ์ของเครือข่ายให้ดียิ่งขึ้น ได้แก่ การแยกแบบจำลองเครือข่ายประสาทออกเป็นสามแบบ คือ แบบจำลองสำหรับพยากรณ์ในฤดูหนาว ฤดูร้อน และฤดูฝน โดยที่ ข้อมูลที่ใช้ในการสอนและเรียนรู้ของเครือข่ายประสาทแต่ละแบบ จะใช้ ดังนี้

(1) แบบจำลองพยากรณ์ในฤดูหนาว ข้อมูลที่ใช้สอน ได้แก่ โหลดและอุณหภูมิในวันที่ 1 – 14 กุมภาพันธ์ 2543 รวมอินพุต 14 แพทเทอร์น ข้อมูลที่ใช้ทดสอบ คือ วันที่ 18 – 25 กุมภาพันธ์ และวันที่ 1 – 5 มีนาคม 2543

(2) แบบจำลองพยากรณ์ในฤดูร้อน ข้อมูลที่ใช้สอน ได้แก่ โหลดและอุณหภูมิในวันที่ 1 – 14 เมษายน 2543 รวมอินพุต 14 แพทเทอร์น ข้อมูลที่ใช้ทดสอบคือ วันที่ 1 – 17 พฤษภาคม 2543

(3) แบบจำลองพยากรณ์ในฤดูฝน ข้อมูลที่ใช้สอน ได้แก่ โหลดและอุณหภูมิในวันที่ 18 – 31 ตุลาคม 2543 รวมอินพุต 14 แพทเทอร์น ข้อมูลที่ใช้ทดสอบคือ วันที่ 14 – 30 กันยายน 2543

ทั้งนี้ โดยมีสมมติฐานว่า การเลือกข้อมูลจำนวน 14 วัน ในแต่ละฤดูมาทำการสร้างแบบจำลอง จะสามารถทำการพยากรณ์ความต้องการไฟฟ้าได้ ไม่ว่าจะเป็นวันใดๆ ในฤดูกาล

เดียวกัน การทดลองหาแบบจำลองและเปรียบเทียบค่าสมรรถนะของเครือข่ายประสาทเทียมของแต่ละฤดูกาล จะกระทำแยกจากกัน โดยที่มีการเปรียบเทียบระหว่าง MSE เป้าหมายเป็น 0.001 กับ 0.0001 และการเปลี่ยนค่าอินพุตที่เป็นดัชนีระบุฤดูกาล ประเภทวัน และวันในสัปดาห์ จากเดิมที่เป็นตัวเลขไบนารี 0 กับ 1 ให้เป็นเลขไบโพลา คือ –1 กับ 1 โดยใช้ค่าน้ำหนักและไบอัสเริ่มต้นเป็นแบบสุ่มตามวิธีของงิวเยน – วิดโรว์ มีผลการทดสอบสำหรับฤดูหนาว ฤดูร้อน และฤดูฝน เป็นไปตามตารางที่ 4.4 ตารางที่ 4.5 และตารางที่ 4.6 ตามลำดับ

จากตารางที่ 4.4 ซึ่งเป็นผลการพยากรณ์ในฤดูหนาว จะเห็นได้ว่า การใช้ค่าดัชนีอินพุตเป็นไบนารี จะใช้จำนวนครั้งในการสอนเฉลี่ยต่ำกว่า ขณะที่ ค่า MAPE ของค่า peak และ MAPE ของ valley โดยเฉลี่ยมีค่าต่ำที่สุด ส่วนการใช้ค่า MSE เป้าหมายเป็น 0.0001 ทำให้ใช้จำนวนครั้งในการสอนสูงมาก โดยที่ไม่ได้เพิ่มสมรรถนะของการพยากรณ์แต่อย่างใด

จากตารางที่ 4.5 ซึ่งเป็นผลการพยากรณ์ในฤดูร้อน พบว่ามีผลในลักษณะเดียวกันกับกรณีฤดูหนาว ส่วนผลการทดลองในตารางที่ 4.6 แสดงให้เห็นว่า การใช้ดัชนีอินพุต เป็นไบนารี และการใช้ค่า MSE เป้าหมายเป็น 0.001 ยังคงให้ค่าสมรรถนะการพยากรณ์ดี เช่นกัน

ดังนั้น โดยสรุปแล้ว การใช้เครือข่ายประสาท ที่แยกแบบจำลองให้พยากรณ์ความต้องการไฟฟ้าแต่ละฤดูกาล ออกจากกัน ใช้ค่าดัชนีอินพุต แบบไบนารี โดยที่มี ค่า MSE เป้าหมายของการสอน เป็น 0.001 มีจำนวนชั้นซ่อนเพียงหนึ่งชั้น และมีจำนวนนิวรอนในชั้นซ่อน เป็น 7 จะให้ผลการพยากรณ์ดีกว่าแบบอื่นๆ

อย่างไรก็ตาม การที่ ค่าน้ำหนักและไบอัสเริ่มต้น ยังคงเป็นแบบสุ่มค่า จึงเป็นไปได้ว่าสมรรถนะของการพยากรณ์ ยังคงเปลี่ยนแปลงไปตามการสอนให้แก่เครือข่ายแต่ละครั้ง ดังนั้นการหาแบบจำลองที่ดีที่สุด สำหรับการพยากรณ์ความต้องการไฟฟ้าในแต่ละฤดู จึงต้องทำการทดลองซ้ำๆ เพื่อคัดเลือกแบบจำลองที่มีสมรรถนะดีที่สุดมาใช้ ในที่นี้ ทำการทดลองแบบสุ่มจำนวน 20 ครั้ง ได้ผลดังตารางที่ 4.7 ซึ่งแบบจำลองที่ดีที่สุด ที่เลือกสำหรับฤดูหนาวคือ WR1-19 ซึ่งมีค่า MAPE ของ peak และของ valley เป็น 8.37 % และ 13.6 % ตามลำดับ สำหรับฤดูร้อนคือ SR1-8 ซึ่งมีค่า MAPE ของ peak และของ valley เป็น 13.42 % และ 3.46 % ตามลำดับ ในขณะที่แบบจำลองฤดูฝน จะเลือก FR1-11 ซึ่งมี MAPE ของ peak และ valley เป็น 5.52 % และ 9.39 % ตามลำดับ ข้อสังเกตอีกประการหนึ่ง คือ การใช้ค่าน้ำหนักและไบอัสเริ่มต้นแบบสุ่มค่า ในการสอนนั้น ถึงแม้ว่าเครือข่ายจะสามารถเรียนรู้ โดยลู่เข้าหาเอาต์พุตเป้าหมายได้รวดเร็ว และโดยมีจำนวนครั้งที่สอนส่วนมากใกล้เคียงกัน แต่ความสามารถในการนำไปใช้พยากรณ์จะต่างกันอย่างเห็นได้ชัด โดยดูจากค่า MAPE ของโหลดสูงสุด และต่ำสุด ตามลำดับ

ตารางที่ 4.4 ผลการทดลองแบบจำลองเครือข่ายประสาทสำหรับพยากรณ์ในฤดูหนาวเมื่อเปลี่ยนดัชนีอินพุตและค่า MSE เป้าหมาย

เป้าหมายของค่าความคลาดเคลื่อนกำลังสองเฉลี่ย (mean square error : MSE) เท่ากับ 0.001									
อินพุตแบบไบนารี					อินพุตแบบไบโพลา				
model	MSEเริ่มต้น	epoch	MAPE-peak	MAPE-valley	model	MSEเริ่มต้น	epoch	MAPE-peak	MAPE-valley
WR1-1	0.887	606	14.66	18.63	WR2-1	1.279	2231	19.21	23.38
WR1-2	0.692	1359	16.80	23.94	WR2-2	0.640	2985	16.82	22.38
WR1-3	1.197	539	17.82	23.76	WR2-3	1.665	1086	12.03	19.67
WR1-4	1.013	1333	12.63	22.07	WR2-4	1.703	655	23.10	24.37
WR1-5	1.494	389	12.73	21.17	WR2-5	1.073	1573	14.23	25.83
WR1-6	0.692	523	15.76	23.33	WR2-6	0.768	1147	13.27	19.85
WR1-7	1.274	661	16.55	24.49	WR2-7	1.590	1035	13.03	25.35
WR1-8	1.560	1044	14.15	20.32	WR2-8	1.005	2197	20.80	23.49
WR1-9	1.207	511	20.32	23.24	WR2-9	0.642	1908	15.44	22.49
WR1-10	0.906	313	13.67	22.17	WR2-10	1.552	1045	17.27	23.14
เฉลี่ย		727.8	15.51	22.31	เฉลี่ย		1568.2	16.52	23.00
เป้าหมายของค่าความคลาดเคลื่อนกำลังสองเฉลี่ย (mean square error : MSE) เท่ากับ 0.0001									
อินพุตแบบไบนารี					อินพุตแบบไบโพลา				
model	MSEเริ่มต้น	epoch	MAPE-peak	MAPE-valley	model	MSEเริ่มต้น	epoch	MAPE-peak	MAPE-valley
WR1-11	0.711	2008	13.46	18.97	WR2-11	1.113	4490	21.65	26.13
WR1-12	0.634	4027	16.79	21.09	WR2-12	1.399	3914	13.56	22.25
WR1-13	1.299	2960	16.08	25.39	WR2-13	1.380	4476	12.85	19.45
WR1-14	1.362	2342	13.89	21.15	WR2-14	0.567	6157	19.45	18.94
WR1-15	0.730	2647	21.55	23.44	WR2-15	1.301	8976	18.41	23.39
WR1-16	0.806	2132	16.47	25.83	WR2-16	1.421	10007	14.91	24.36
WR1-17	1.672	4707	22.80	20.33	WR2-17	1.168	3258	17.31	22.85
WR1-18	0.859	3359	16.40	21.66	WR2-18	1.509	4176	13.17	19.05
WR1-19	1.344	4528	12.18	21.62	WR2-19	0.977	4416	19.83	19.83
WR1-20	0.635	3351	15.82	25.05	WR2-20	1.274	5796	11.95	28.03
เฉลี่ย		3206.1	16.54	22.45	เฉลี่ย		5566.6	16.31	22.43

ตารางที่ 4.5 ผลการทดลองแบบจำลองเครือข่ายประสาทสำหรับพยากรณ์ในฤดูร้อนเมื่อเปลี่ยนดัชนีอินพุตและค่า MSE เป้าหมาย

เป้าหมายของค่าความคลาดเคลื่อนกำลังสองเฉลี่ย (mean square error : MSE) เท่ากับ 0.001									
อินพุตแบบไบนารี					อินพุตแบบไบโพลา				
model	MSEเริ่มต้น	epoch	MAPE-peak	MAPE-valley	model	MSEเริ่มต้น	epoch	MAPE-peak	MAPE-valley
SR1-1	1.029	247	17.25	23.35	SR2-1	1.29	1580	19.11	23.75
SR1-2	1.346	275	17.83	23.73	SR2-2	1.36	572	21.02	22.94
SR1-3	1.733	155	15.53	21.04	SR2-3	1.20	383	17.15	21.71
SR1-4	1.017	284	19.72	25.25	SR2-4	1.25	701	15.51	20.02
SR1-5	1.251	952	17.34	19.77	SR2-5	1.48	371	18.28	22.05
SR1-6	1.068	290	15.01	18.16	SR2-6	0.89	495	18.12	22.94
SR1-7	1.602	174	17.26	21.56	SR2-7	1.83	1718	18.24	21.31
SR1-8	1.203	265	14.13	20.54	SR2-8	1.58	294	19.25	20.01
SR1-9	1.661	208	17.06	28.21	SR2-9	1.31	195	18.52	21.40
SR1-10	1.635	204	16.39	20.84	SR2-10	1.28	578	19.84	22.42
เฉลี่ย		305.4	16.75	22.25	เฉลี่ย		688.6	18.50	21.85
เป้าหมายของค่าความคลาดเคลื่อนกำลังสองเฉลี่ย (mean square error : MSE) เท่ากับ 0.0001									
อินพุตแบบไบนารี					อินพุตแบบไบโพลา				
model	MSEเริ่มต้น	epoch	MAPE-peak	MAPE-valley	model	MSEเริ่มต้น	epoch	MAPE-peak	MAPE-valley
SR1-11	1.185	7013	22.20	24.05	SR2-11	1.041	2021	13.23	20.50
SR1-12	1.267	3317	17.52	15.41	SR2-12	1.208	1065	19.85	18.35
SR1-13	1.462	2242	19.66	19.07	SR2-13	1.030	2917	14.64	16.87
SR1-14	1.153	2469	19.02	19.95	SR2-14	1.164	703	17.29	18.49
SR1-15	1.084	1393	15.70	16.00	SR2-15	0.867	874	15.75	19.92
SR1-16	1.205	2343	18.25	17.01	SR2-16	1.015	725	15.68	14.97
SR1-17	0.912	1412	18.27	25.78	SR2-17	1.333	2922	16.81	23.92
SR1-18	1.252	3663	15.07	17.84	SR2-18	0.849	1131	15.91	21.74
SR1-19	1.254	1538	19.38	16.83	SR2-19	0.995	1523	15.45	19.38
SR1-20	1.145	1840	18.32	20.03	SR2-20	1.425	1293	20.36	18.31
เฉลี่ย		2723	18.34	19.19	เฉลี่ย		1517.4	16.50	19.25

ตารางที่ 4.6 ผลการทดลองแบบจำลองเครือข่ายประสาทสำหรับพยากรณ์ในฤดูฝนเมื่อเปลี่ยนดัชนีอินพุตและค่า MSE เป้าหมาย

เป้าหมายของค่าความคลาดเคลื่อนกำลังสองเฉลี่ย (mean square error : MSE) เท่ากับ 0.001									
อินพุตแบบไบนารี					อินพุตแบบไบโพลา				
model	MSEเริ่มต้น	epoch	MAPE-peak	MAPE-valley	model	MSEเริ่มต้น	epoch	MAPE-peak	MAPE-valley
FR1-1	0.688	137	6.10	11.19	FR2-1	0.913	399	6.81	12.27
FR1-2	0.890	297	7.31	9.56	FR2-2	0.618	796	6.55	9.87
FR1-3	0.919	408	6.31	11.07	FR2-3	0.955	296	5.80	9.96
FR1-4	1.024	265	7.67	12.72	FR2-4	1.061	1100	5.87	10.89
FR1-5	0.877	393	5.75	10.63	FR2-5	1.399	382	7.43	10.18
FR1-6	1.002	431	6.61	8.88	FR2-6	0.910	460	7.22	12.62
FR1-7	0.909	438	5.78	10.24	FR2-7	1.047	723	5.95	10.16
FR1-8	0.981	654	5.48	10.31	FR2-8	1.134	804	6.62	12.67
FR1-9	0.824	372	7.54	11.95	FR2-9	0.927	776	7.08	11.06
FR1-10	1.062	570	6.38	10.24	FR2-10	0.963	417	6.30	10.56
เฉลี่ย		396.5	6.49	10.68	เฉลี่ย		615.3	6.56	11.02
เป้าหมายของค่าความคลาดเคลื่อนกำลังสองเฉลี่ย (mean square error : MSE) เท่ากับ 0.0001									
อินพุตแบบไบนารี					อินพุตแบบไบโพลา				
model	MSEเริ่มต้น	epoch	MAPE-peak	MAPE-valley	model	MSEเริ่มต้น	epoch	MAPE-peak	MAPE-valley
FR1-11	0.819	1522	7.51	13.30	FR2-11	0.880	4665	6.22	10.34
FR1-12	1.052	1710	5.97	11.01	FR2-12	1.385	2532	7.31	10.17
FR1-13	1.077	1561	6.22	10.18	FR2-13	1.159	2321	7.40	12.49
FR1-14	1.209	2406	6.42	10.37	FR2-14	1.092	1557	6.39	9.38
FR1-15	1.094	1860	7.21	10.88	FR2-15	1.092	1768	6.04	11.86
FR1-16	1.012	2137	6.26	12.15	FR2-16	1.088	3329	7.18	9.82
FR1-17	1.068	2812	6.41	11.92	FR2-17	1.073	5725	7.36	9.25
FR1-18	0.874	1987	7.01	8.45	FR2-18	0.809	5079	7.93	11.89
FR1-19	0.932	2058	6.27	10.36	FR2-19	0.849	2124	6.27	8.72
FR1-20	1.347	2798	6.50	10.61	FR2-20	0.966	3041	4.95	10.35
เฉลี่ย		2085	6.57	10.92	เฉลี่ย		3214.1	6.71	10.43

ตารางที่ 4.7 ผลการทดลองสุ่ม(trial) หาแบบจำลองเครือข่ายประสาทที่เหมาะสมสำหรับใช้ในการพยากรณ์ความต้องการไฟฟ้าแต่ละฤดูกาล

ฤดูหนาว					ฤดูร้อน					ฤดูฝน				
model	MSE เริ่มต้น	epoch	MAPE peak	MAPE valley	model	MSE เริ่มต้น	epoch	MAPE peak	MAPE valley	model	MSE เริ่มต้น	epoch	MAPE peak	MAPE valley
WR1-1	0.887	606	14.16	14.64	SR1-1	1.029	247	14.28	7.82	FR1-1	1.461	674	7.99	8.89
WR1-2	6.920	1359	8.98	21.64	SR1-2	1.346	275	18.27	12.40	FR1-2	0.813	2245	9.49	10.92
WR1-3	1.197	539	11.29	22.65	SR1-3	1.733	155	19.99	7.52	FR1-3	1.111	529	10.10	10.28
WR1-4	1.013	1333	7.80	19.58	SR1-4	1.017	284	35.80	13.93	FR1-4	0.909	585	7.38	10.94
WR1-5	1.494	389	5.38	22.95	SR1-5	1.251	952	14.81	3.26	FR1-5	0.804	496	9.35	10.09
WR1-6	0.692	523	11.00	21.06	SR1-6	1.068	290	19.82	16.43	FR1-6	0.625	248	9.05	9.25
WR1-7	1.274	661	10.55	22.73	SR1-7	1.602	174	35.89	17.13	FR1-7	1.333	705	8.26	9.34
WR1-8	1.560	1044	10.58	15.77	SR1-8	1.203	265	13.42	3.46	FR1-8	0.978	743	7.84	11.04
WR1-9	1.207	511	8.55	19.53	SR1-9	1.661	208	20.99	17.65	FR1-9	1.063	358	6.79	9.00
WR1-10	0.906	313	11.41	20.36	SR1-10	1.635	204	14.69	8.34	FR1-10	0.576	1247	8.78	8.79
WR1-11	1.033	909	6.39	22.71	SR1-11	1.380	126	18.78	17.86	FR1-11	1.205	532	5.52	9.39
WR1-12	0.497	357	11.79	11.71	SR1-12	1.632	499	20.76	12.77	FR1-12	0.913	560	8.47	9.12
WR1-13	1.129	720	6.18	25.92	SR1-13	1.345	189	24.13	14.37	FR1-13	0.986	2124	7.43	9.34
WR1-14	1.002	506	16.31	16.63	SR1-14	1.580	485	31.42	16.47	FR1-14	0.740	639	11.46	12.06
WR1-15	1.268	1218	11.45	26.01	SR1-15	1.592	339	14.47	8.83	FR1-15	1.010	532	5.57	11.15
WR1-16	0.953	1410	9.62	22.06	SR1-16	1.520	536	12.90	14.36	FR1-16	1.227	337	7.86	8.07
WR1-17	0.691	536	9.36	16.89	SR1-17	1.114	327	33.27	18.40	FR1-17	1.245	480	8.84	9.64
WR1-18	0.724	511	7.94	19.86	SR1-18	1.509	252	16.99	4.42	FR1-18	1.153	555	8.10	11.33
WR1-19	0.916	812	8.37	13.60	SR1-19	1.654	1384	24.72	10.98	FR1-19	1.406	661	5.93	9.75
WR1-20	0.695	582	11.74	16.94	SR1-20	1.188	418	12.59	5.31	FR1-20	1.336	437	8.20	7.95

4.3 โปรแกรมคอมพิวเตอร์เครือข่ายประสาทร่วมกับหลักวิธีถ่ายทอดพันธุกรรม

เนื่องจาก โปรแกรมคอมพิวเตอร์ที่เขียนขึ้นมา เพื่อหาเครือข่ายประสาทให้สามารถพยากรณ์ความต้องการไฟฟ้า นั้น มีการกำหนดค่าน้ำหนักและไบอัสเริ่มต้น เป็นแบบสุ่ม ดังนั้นจึงสามารถสร้างได้หลายรูปแบบ และถ้าหากจะกำหนดค่าน้ำหนักเริ่มต้นให้เป็นค่าคงที่ใดๆ ก็อาจจะทำให้ต้องใช้จำนวนครั้งที่สอนมาก จึงจะได้แบบจำลองที่มีสมรรถนะดีตามต้องการ ดังนั้น การนำหลักวิธีถ่ายทอดพันธุกรรม (GA) มาทำการค้นหาคำตอบ และไบอัสที่มีเหมาะสม แล้วนำไปใช้เป็นค่าน้ำหนักและไบอัสเริ่มต้นของเครือข่ายประสาท จะทำให้กระบวนการสอนและเรียนรู้ ลู่เข้าหาคำตอบ หรือการปรับค่าน้ำหนักเป็นไปอย่างรวดเร็วยิ่งขึ้น และยังอาจจะทำให้ได้ ค่าน้ำหนักของแบบจำลองเครือข่ายประสาท ที่เป็นค่าออปติไมซ์ (optimize) ที่สุดในบริเวณของคำตอบ (solution space) อีกด้วย

การเขียนโปรแกรมคอมพิวเตอร์เลียนแบบ GA จะประกอบด้วยส่วนต่างๆ คือ

(1) การสร้างประชากร (population) ของคำตอบที่เป็นไปได้ ซึ่งในที่นี้ คือ ค่าน้ำหนักและไบอัส ของเครือข่ายประสาท

(2) การกระทำครอสโอเวอร์ (crossover)

(3) การกระทำมิวเตชัน (mutation)

(4) การหาค่าความเหมาะสม (fitness)

(5) การคัดเลือกประชากรรุ่นใหม่ (offspring selection)

(6) การตรวจสอบเงื่อนไขหยุดสร้างคำตอบ (stopping criteria)

4.3.1 การสร้างประชากรคำตอบเริ่มต้น

เนื่องจากเราต้องการใช้ GA ในการค้นหาค่าน้ำหนักและไบอัสเริ่มต้นให้แก่เครือข่ายประสาท ดังนั้น จำนวนค่าน้ำหนักและไบอัสที่จะต้องหาจึงขึ้นกับ รูปแบบของเครือข่ายที่ใช้ ในที่นี้ เครือข่ายเป็นแบบที่มีชั้นอินพุต ขนาด 15 นิวรอน ชั้นซ่อนหนึ่งชั้นมีขนาด 7 นิวรอน และชั้นเอาต์พุตมีขนาด 2 นิวรอน จะได้ค่าน้ำหนักและไบอัสทั้งหมดเป็น 128 หน่วย ในที่นี้จะเลือกใช้โครโมโซมของคำตอบที่มีแต่ละยีนส์ เป็นเลขจำนวนจริง (floating point) ที่อยู่ระหว่าง –1 กับ 1 ซึ่งเป็นค่าที่เหมาะสมของน้ำหนักและไบอัส แต่เนื่องจาก คำสั่งใน MatLab จะสร้างเลขจำนวนจริงที่อยู่ระหว่าง 0 กับ 1 จึงต้องเขียนโปรแกรมเพิ่มเติม และทำการสุ่มค่าน้ำหนักและไบอัสที่เรียงต่อกันเป็นอันดับ ในโครโมโซมที่แทนด้วยเมตริกซ์แถว และหากกำหนดขนาดประชากร (population size) หรือ pop_size เป็นจำนวน 10 ชุด จะได้เมตริกซ์แทนคำตอบเริ่มต้น เป็นขนาด 10 x 128 รูปแบบคำสั่งแสดงในรูปที่ 4.9

```
popsize=10;
wgh=zeros(popsize,128);
rand('state',sum(100*clock));
for i=1:popsize
    w1=rand(1,128);
    wgh(i,1:128)=2*(w1-min(w1))/(max(w1)-min(w1)) - 1;
 end
```

รูปที่ 4.9 ชุดคำสั่งที่ใช้ในการสร้างประชากรคำตอบที่เป็นค่าน้ำหนักและไบอัส

4.3.2 การกระทำครอสโอเวอร์

การกระทำครอสโอเวอร์ จะทำให้เกิดประชากรใหม่ อันเนื่องจากการไขว้ยีนส์ระหว่างโครโมโซม ซึ่งโครโมโซมที่จะเลือกมากระทำการครอสโอเวอร์จะได้จากการสุ่ม ดังนั้นในโปรแกรมจะทำการสุ่มหาโครโมโซมที่จะนำมาครอสโอเวอร์ ซึ่งจะได้โครโมโซมที่ต้องการใช้เป็นคู่ (mating) ดังแสดงชุดคำสั่งในรูปที่ 4.10

```
pc=0.8;
mask=rand(1,popsize);
    for i=1:popsize
        if mask(i)>=pc
           mask1(i)=0;
        else
           mask1(i)=mask(i);
        end
    end
    for i=1:popsize
        if mask1(1,i)==0
           father(i,1:128)=0;
        else
           father(i,1:128)=wgh(i,1:128);
        end
    end
    j=1;
    for i=1:popsize
        if father(i,:)~=0
           father1(j,1:128)=father(i,:);
           j=j+1;
        else
        end
    end
```

รูปที่ 4.10 ชุดคำสั่งในการสุ่มเลือกโครโมโซมมากระทำครอสโอเวอร์

สำหรับการกระทำครอสโอเวอร์ ในที่นี้จะเลือกแบบหนึ่งจุดตัด (1- cut point) โดยที่จุดตัดจะได้จากการสุ่มหา ในที่นี้จะกำหนดให้สุ่มค่าตำแหน่งรวม 127 ตำแหน่ง แล้วหากตำแหน่งใดได้ค่าเป็น 64 จะใช้ตำแหน่งนั้นในการครอสโอเวอร์ แสดงชุดคำสั่งในการครอสโอเวอร์ในรูปที่ 4.11

```
cutpt=randperm(127);
   crosspt=cutpt(64);
   row=size(father1,1);
   row1=floor(row/2);
   sonc=zeros(row1*2,128);
   j=1;
   if row1==1
      sonc(j,1:128)=[father1(1,1:crosspt),father1(2,crosspt+1:128)];
      sonc(j+1,1:128)=[father1(2,1:crosspt),father1(1,crosspt+1:128)];
   elseif row1>=2
      for i=1:2:row1*2
         sonc(j,1:128)=[father1(i,1:crosspt),father1(i+1,crosspt+1:128)];
         sonc(j+1,1:128)=[father1(i+1,1:crosspt),father1(i,crosspt+1:128)];
         j=j+2;
      end
   else
   end
```

รูปที่ 4.11 ชุดคำสั่งที่ใช้ในการหาจุดตัดไขว้ยีนส์และการกระทำครอสโอเวอร์

4.3.3 การกระทำมิวเตชัน

เป็นลักษณะของการกลายพันธุ์ ซึ่งหมายถึงมีการเปลี่ยนค่าของยีนส์ (alleles) ซึ่งยีนส์ที่จะเปลี่ยนค่า จะไม่แน่นอนว่าจะเป็นตำแหน่งใด (locus) และจะเปลี่ยนเป็นอะไร ดังนั้นในการเขียนโปรแกรม จะต้องสุ่มตำแหน่งของยีนส์ที่จะกระทำมิวเตชัน ดังแสดงในรูปที่ 4.12

```
pm=0.1;
    ngenes=128*popsize;
    maskmute=rand(1,ngenes);
    maskmute1=zeros(1,ngenes);
    for i=1:ngenes
        if maskmute(i)>=pm
           maskmute1(i)=0;
             else
           maskmute1(i)=maskmute(i);
       end
    end
```

รูปที่ 4.12 ชุดคำสั่งที่ใช้ในการกำหนดตำแหน่งที่จะกระทำมิวเตชัน

เมื่อได้ตำแหน่งของยีนส์ที่จะกระทำมิวเตชันแล้ว ต่อไปจะเขียนโปรแกรมให้ส่งทอดตำแหน่งของยีนส์ที่จะมิวเตชันไปยังประชากรคำตอบทั้งหมด แล้วทำการมิวเตชันค่าของน้ำหนักและไบอัสในตำแหน่งที่ตรงกัน ด้วยค่าน้ำหนักหรือไบอัสใหม่ที่ได้จากการสุ่ม ซึ่งมีชุดคำสั่งดังแสดงในรูปที่ 4.13

```
wghext=reshape(wgh',1,ngenes);
    wghext1=zeros(1,ngenes);
    for i=1:ngenes
        if maskmute1(i)~=0
           wnew(i)=rand(1,1);
           wnew1(i)=2*wnew(i)-1;
           wghext1(1,i)=wnew1(i);
        else
           wghext1(1,i)=wghext(1,i);
        end
    end
    wghnew=reshape(wghext1,128,popsize);
    wghnew1=wghnew';
[M,N] = size(wghnew1);
sonm=zeros(size(wghnew1));
    P = 0;
    for i = 1:M
        if wghnew1(i,:)==wgh(i,:)
           sonm(i,:)=0;
        else
           sonm(i,:)=wghnew1(i,:);
        end
    end
    for i = 1:M
        if sonm(i,:)~=zeros(1,N)
           P=P+1;
           k(P) = i;
        end
    end
      for j = 1:length(k)
        sonm1(j,:) = sonm(k(j),:);
    end
```

รูปที่ 4.13 ชุดคำสั่งในการกระทำมิวเตชัน

4.3.4 การหาค่าความเหมาะสม (fitness)

ประชากรคำตอบทั้งหมด ทั้งรุ่นพ่อแม่ (parents) และรุ่นลูก (offspring) ไม่ว่าจะเกิดจากการครอสโอเวอร์หรือมิวเตชัน จะต้องนำไปทดสอบด้วยการใช้เป็นค่าน้ำหนักและไบอัสเริ่มต้นในเครือข่ายประสาทที่กำหนด เมื่อผ่านการคำนวณจากเครือข่ายประสาทแล้วจะให้เอาต์พุต ซึ่งนำไปเปรียบเทียบกับค่าเอาต์พุตเป้าหมายที่ต้องการ หาค่าความผิดพลาดเป็น

ค่าเฉลี่ยของความผิดพลาดกำลังสอง (MSE) แต่เนื่องจากค่าที่ได้จะมีค่าน้อย จึงจะใช้ส่วนกลับของ MSE เป็นฟังก์ชันความเหมาะสม (fitness function) นั่นคือ เราต้องการหาค่าน้ำหนักและไบอัส มีค่าน้อยที่สุด แสดงว่า ค่าความเหมาะสมจะต้องมีค่ามากที่สุด ชุดคำสั่งในการหาค่าความเหมาะสมของประชากรทั้งรุ่นพ่อแม่และรุ่นลูก เรียงอันดับจากมากไปน้อย แสดงในรูปที่ 4.14

```
mserr=[mserr1 mserr2 mserr3];
    fitness1=1./mserr;
    sortmse=sort(mserr);
    fitness=1./sortmse;
```

รูปที่ 4.14 ชุดคำสั่งที่ใช้คำนวณหาค่าความเหมาะสมเรียงอันดับจากมากไปน้อย

4.3.5 การคัดเลือกประชากรรุ่นใหม่

หลังจากที่คำนวณหาค่าความเหมาะสม และจัดอันดับความเหมาะสมจากมากไปน้อยแล้ว จะทำการคัดเลือกประชากรรุ่นใหม่ ที่จะใช้ในกระบวนการ GA รุ่นถัดไป หากว่าค่าความเหมาะสมของประชากรรุ่นนี้ยังไม่ดีพอตามต้องการ ซึ่งประชากรรุ่นใหม่นี้ จะคัดเลือกจากทั้งรุ่นพ่อแม่และรุ่นลูก (elitist selection) โดยเลือกจากประชากรที่มีค่าความเหมาะสมสูงสุดเรียงลงมาหาค่าความเหมาะที่น้อยตามจำนวนประชากรที่ต้องการ ซึ่งชุดคำสั่งที่ใช้ แสดงในรูปที่ 4.15

```
weight=[wgh;sonc;sonm1];
    [M,N] = size(weight);
    for j = 1:M
        for k = 1:M
            if (fitness(j) == fitness1(k))
               nrow(j) = k;
            end
        end
    end
    numbers=popsize;
     for i = 1:numbers
        wgh(i,:) = weight(nrow(i),:);
     end
```

รูปที่ 4.15 ชุดคำสั่งที่ใช้ในการคัดเลือกประชากรรุ่นใหม่

4.3.6 การตรวจสอบเงื่อนไขหยุดสร้างคำตอบ (stopping criteria)

การวิวัฒนาการ (evolution) กำเนิดประชากรคำตอบ จากรุ่นหนึ่งไปอีกรุ่นหนึ่ง จะให้ประชากรคำตอบที่มีความเหมาะสมมากขึ้น ดังนั้น จะต้องกำหนดเงื่อนไขเพื่อหยุดวิวัฒนาการ เมื่อเห็นว่าสมควร อย่างใดอย่างหนึ่ง หรือหลายอย่าง ในที่นี้จะกำหนดได้สองอย่าง คือ เมื่อมีค่าความเหมาะสมของคำตอบเป็นดังที่คาดไว้ หรือ จำนวนรุ่นที่วิวัฒนาการ (generation) ดังนั้น เมื่อมีการวิวัฒนาการไปหนึ่งรุ่น จะต้องมีการตรวจสอบเงื่อนไข ว่า ค่าความเหมาะสมเป็นไปดังที่กำหนดแล้วหรือไม่ หรือ จำนวนรุ่นที่กำหนดให้วิวัฒนาการ ถึงค่าสูงสุด (maximum generation) ที่ตั้งไว้แล้วหรือยัง หากสอดคล้องกับเงื่อนไขแล้ว จึงจะหยุดการวิวัฒนาการ แล้วให้คำตอบเป็นประชากรรุ่นสุดท้าย ที่มีค่าความเหมาะสมสูงที่สุด ชุดคำสั่งแสดงในรูปที่ 4.16

```
[R,C]=size(wgh);
    dd=0;
    d=0;
    for i=2:R
       if wgh(i,:)==wgh(1,:)
          d=i;
       else
          dd=1;
       end
    end
      if (dd~=1)& (fitness >=10.0)
       n=generation;
    else
    end
```

รูปที่ 4.16 ชุดคำสั่งที่แสดงถึงเงื่อนไขการหยุดวิวัฒนาการ

4.4 การทดลองใช้เครือข่ายประสาทร่วมกับหลักวิธีถ่ายทอดพันธุกรรม

เนื่องจาก GA เป็นเป็นการหาค่าคำตอบ โดยการส่งทอด (mapping) ค่าคำตอบที่สุ่มขึ้นมาในพื้นที่เข้ารหัส (coding space) ไปยังพื้นที่ของคำตอบ (solution space) ซึ่งอาจจะมีทั้งพื้นที่ของคำตอบที่เป็นไปได้ (feasible area) กับพื้นที่ของคำตอบที่เป็นไปไม่ได้ (infeasible area) ทั้งนี้ขึ้นกับองค์ประกอบต่างๆ ได้แก่ จำนวนประชากรคำตอบเริ่มต้น (pop_size) อัตราการมิวเตชัน (Pm) อัตราการครอสโอเวอร์ (Pc) และจำนวนของรุ่นของวิวัฒนาการ (generation) ดังนั้น ในการที่จะค้นหาคำตอบได้เหมาะสมที่สุด จะต้องกำหนดค่าเหล่านี้ให้แก่ GA ซึ่งแล้วแต่ปัญหา

สำหรับงานวิจัยนี้ เราไม่ได้ต้องการให้ GA หาค่าน้ำหนักและไบอัสที่ดีที่สุด จนสามารถนำไปใช้เป็นค่าน้ำหนักและไบอัสของเครือข่ายประสาทได้เลยทันที หากแต่ต้องการใช้ GA ค้นหาค่าน้ำหนักและไบอัสเริ่มต้น แทนที่การใช้สุ่มหาแบบเดิม ซึ่งเครือข่ายประสาทเคยใช้อยู่ ดังนั้น การกำหนดเงื่อนไขการหยุดค้นหาคำตอบ โดยการตั้งค่าของความเหมาะสมไว้ จึงไม่จำเป็น เพราะอาจจะทำให้ GA ใช้เวลานานในการค้นหาคำตอบที่เหมาะสม ตามต้องการ ซึ่งเป็นสิ่งที่ไม่จำเป็น เงื่อนไขที่จะกำหนดให้การวิวัฒนาการสิ้นสุด ควรจะเป็น จำนวนรุ่นของวิวัฒนาการ และ/หรือ การที่ประชากรรุ่นใหม่ที่สร้างขึ้นโดย GA มีค่าความเหมาะสมเท่ากัน

สิ่งที่เป็นปัญหา คือ การกำหนดจำนวนประชากร ค่าของอัตราการมิวเตชัน และอัตราการครอสโอเวอร์ โดยทั่วไป อัตราการมิวเตชัน จะมีค่าน้อยๆ ในย่านของ 0.1 % - 1 % และอัตราการครอสโอเวอร์ก็สามารถแปรค่าได้ในช่วงกว้างตั้งแต่ 10 % จนถึง 90 % ขึ้นกับวิธีการคัดเลือกประชากรรุ่นใหม่ ในที่นี้เราใช้วิธี elitist หรือการขยายขนาดของพื้นที่คัดเลือก (enlarged sampling space) ดังนั้น จึงไม่ต้องกังวลว่า ค่าอัตราการครอสโอเวอร์ กับ อัตราการมิวเตชันจะมากเกินไปจนทำให้เกิดการสุ่มค่ามากเกินเหตุ (random perturbation) [7]

อย่างไรก็ตาม จะทำการทดลองเพื่อให้มั่นใจในการกำหนดจำนวนประชากร อัตราการมิวเตชัน และอัตราการครอสโอเวอร์ โดยที่จะหาค่าน้ำหนักและไบอัสเริ่มต้นสำหรับเครือข่ายประสาทที่ใช้ในการพยากรณ์ความต้องการไฟฟ้าแต่ละฤดูกาล แล้วตรวจดูว่า ควรจะกำหนดขนาดของประชากร อัตราการครอสโอเวอร์ และ อัตราการมิวเตชันเท่าใด จึงจะเหมาะสมกับปัญหา ซึ่งจะดูจากผลการนำน้ำหนักและไบอัส ที่ได้ ไปใช้ในกระบวนการสอนของเครือข่ายประสาท แล้วให้ผลการพยากรณ์แม่นยำที่สุด ผลการทดสอบในฤดูหนาว ฤดูร้อน และฤดูฝน แสดงในตารางที่ 4.8 ตารางที่ 4.9 และ ตารางที่ 4.10 ตามลำดับ

จะเห็นได้ว่า การใช้จำนวนประชากรคำตอบจำนวนมากขึ้น จะต้องใช้จำนวนรุ่นในการวิวัฒนาการมากขึ้น จึงจะได้ค่าความเหมาะสมที่กำหนด โดยที่ กำหนดให้ GA สิ้นสุดเมื่อประชากรรุ่นใหม่ มีค่าความเหมาะสมเท่ากันหมด

จากผลการทดลอง จะเห็นว่า เมื่อใช้ขนาดประชากรมากขึ้น จะได้ค่าความเหมาะสมของคำตอบสูงขึ้น ขณะที่ค่าอัตราการครอสโอเวอร์สูงขึ้น ไม่ได้ทำให้ค่าความเหมาะสมเพิ่มขึ้นแต่อย่างใด และจำนวนรุ่นของวิวัฒนาการ ที่ใช้ใน GA เพื่อให้คำตอบมีความเหมาะสม จะไม่ขึ้นกับค่าอัตราการครอสโอเวอร์ ถ้าหากกำหนดให้ความเหมาะสมของคำตอบที่ได้จาก GA ในแต่ละกรณีมีค่าเท่ากัน ซึ่งหมายถึง ค่าน้ำหนักและไบอัสเริ่มต้นของเครือข่ายประสาทมีพื้นฐานเดียวกัน อย่างไรก็ตาม การกำหนดเงื่อนไขเช่นนั้น จะส่งผลให้ต้องใช้จำนวนรุ่นของวิวัฒนาการ มากยิ่งขึ้น ซึ่งใช้เวลาการค้นหาคำตอบที่เหมาะสม ใช้เวลานานเกินไป จึงอาจจะต้องกำหนดจำนวน

ตารางที่ 4.8 การทดลองใช้ GA หาน้ำหนักและไบอัสเริ่มต้นของเครือข่ายประสาทที่ใช้พยากรณ์ความต้องการไฟฟ้าในฤดูหนาว

การทดลอง	ขนาดประชากร	Pm	Pc	generation	fitness	MAPE-peak	MAPE-valley	MAPE-forecast
GAW_1	10	0.01	0.3	114	3.68	7.55	21.50	14.77
GAW_2			0.4	91	2.74	12.82	18.24	13.05
GAW_3			0.5	46	2.48	12.17	18.55	11.48
GAW_4			0.6	63	3.75	8.38	19.69	12.01
GAW_5			0.7	48	2.53	7.56	21.59	13.09
GAW_6			0.8	52	2.66	5.95	11.87	6.09
GAW_7			0.9	28	1.39	4.73	15.55	6.36
average				63				
GAW_8	20	0.01	0.3	266	5.45	6.62	16.75	8.55
GAW_9			0.4	227	6.14	8.77	27.38	17.72
GAW_10			0.5	207	7.06	8.85	17.66	8.81
GAW_11			0.6	193	8.70	9.05	15.59	11.91
GAW_12			0.7	78	4.23	6.39	18.36	7.60
GAW_13			0.8	185	7.02	9.04	15.31	15.11
GAW_14			0.9	98	4.91	5.85	10.53	7.75
average				179				

ตารางที่ 4.9 การทดลองใช้ GA หาน้ำหนักและไบอัสเริ่มต้นของเครือข่ายประสาทที่ใช้พยากรณ์ความต้องการไฟฟ้าในฤดูร้อน

การทดลอง	ขนาดประชากร	Pm	Pc	generation	fitness	MAPE-peak	MAPE-valley	MAPE-forecast
GAS_1	10	0.01	0.3	172	2.35	15.37	11.95	13.93
GAS_2			0.4	57	0.98	20.53	10.89	15.64
GAS_3			0.5	56	1.81	14.61	12.16	11.39
GAS_4			0.6	41	0.76	36.33	9.18	28.62
GAS_5			0.7	92	1.57	27.35	8.80	22.12
GAS_6			0.8	89	1.36	12.55	4.63	8.90
GAS_7			0.9	47	0.43	15.04	5.13	11.00
average				79				
GAS_8	20	0.01	0.3	385	8.26	17.54	14.48	15.16
GAS_9			0.4	139	1.72	26.13	5.08	16.57
GAS_10			0.5	150	2.62	20.74	13.11	16.87
GAS_11			0.6	170	0.96	18.73	18.76	22.45
GAS_12			0.7	135	1.55	12.11	9.08	9.42
GAS_13			0.8	155	3.88	28.04	12.27	21.75
GAS_14			0.9	137	1.21	22.63	10.17	13.96
average				182				

ตารางที่ 4.10 การทดลองใช้ GA หาน้ำหนักและไบอัสเริ่มต้นของเครือข่ายประสาทที่ใช้พยากรณ์ความต้องการไฟฟ้าในฤดูฝน

การทดลอง	ขนาดประชากร	Pm	Pc	generation	fitness	MAPE-peak	MAPE-valley	MAPE-forecast
GAF_1	10	0.01	0.3	162	4.27	5.30	11.93	7.78
GAF_2			0.4	104	3.58	9.23	13.12	11.20
GAF_3			0.5	191	4.27	9.26	9.55	8.94
GAF_4			0.6	82	1.64	7.08	9.97	9.42
GAF_5			0.7	63	3.58	5.55	9.93	8.70
GAF_6			0.8	23	1.67	8.57	11.96	10.16
GAF_7			0.9	52	1.68	5.47	14.29	9.44
average				97				
GAF_8	20	0.01	0.3	148	4.68	9.86	9.95	11.08
GAF_9			0.4	220	9.80	7.58	8.59	8.90
GAF_10			0.5	123	4.81	6.76	14.08	10.46
GAF_11			0.6	51	1.64	8.44	9.85	11.20
GAF_12			0.7	87	3.84	8.52	9.22	8.60
GAF_13			0.8	55	3.45	7.80	7.76	8.46
GAF_14			0.9	120	5.43	7.35	9.32	8.40
average				115				

รุ่นสูงสุด ที่จะให้ GA ทำงาน ไว้ด้วย ซึ่งหากว่า การค้นหาคำตอบยังไม่ได้ค่าความเหมาะสมที่กำหนด ภายในจำนวนรุ่นของวัฒนาการสูงสุดที่กำหนดแล้ว ควรจะสิ้นสุดการค้นหา

นอกจากนี้ จำนวนประชากรเริ่มต้นที่กำหนดขึ้น ที่ใช้ในพื้นที่เข้ารหัส (coding space) ไม่น่าจะมีผลต่อการสิ้นสุดของ GA เพราะว่า หากค่าประชากรที่สุ่มสร้างขึ้นมา เมื่อถูกส่งทอดผ่านไปยังพื้นที่ของคำตอบ (solution space) แล้ว ไปตกในพื้นที่ของคำตอบ ที่ไม่น่าเป็นไปได้ (infeasible area) หรือ พื้นที่ของคำตอบที่ผิดพลาด (illegal area) ก็จะต้องมีการกระทำของ GA ต่อไปอีก จนกว่าจะได้ประชากรคำตอบที่อยู่ใน พื้นที่คำตอบที่เป็นไปได้ (feasible area) การกำหนดให้ใช้จำนวนประชากรมากเกินไป จะมีผลให้ GA ใช้เวลานาน จึงจะลู่เข้าหาคำตอบที่เหมาะสม

กล่าวโดยสรุปแล้ว ค่าอัตราการครอสโอเวอร์ และอัตราการมิวเตชัน ควรจะใช้ค่ามาก เพื่อเพิ่มสมรรถนะในการค้นหาคำตอบที่เหมาะสมของ GA และจำนวนประชากรที่เหมาะสม น่าจะเป็น 10 ชุด เงื่อนไขการสิ้นสุดของ GA ควรเป็นจำนวนรุ่นสูงสุดของการวิวัฒนาการ และเมื่อประชากรรุ่นใหม่มีค่าความเหมาะสมเท่ากัน และนอกจากนี้ ควรจะได้ทำการทดลองหาคำตอบหลายครั้ง (trial) เพื่อ ดูผลลัพธ์ในเชิงสถิติ ของการลู่เข้าหาคำตอบด้วย ผลการทดลองรัน (run) จำนวน 10 ครั้ง เมื่อใช้ประชากรเริ่มต้นจำนวน 10 ชุด จำนวนรุ่นของการวิวัฒนาการสูงสุดไม่เกิน 1000 รุ่น โดยใช้อัตราการมิวเตชัน 0.1 และอัตราการครอสโอเวอร์ 0.8 ในการค้นหาค่าน้ำหนักและไบอัส ที่จะนำไปใช้ในเครือข่ายประสาท สำหรับการพยากรณ์ความต้องการไฟฟ้าในฤดูหนาว ฤดูร้อน และฤดูฝน แสดงในตารางที่ 4.11 ตารางที่ 4.12 และ ตารางที่ 4.13 ตามลำดับ

ตารางที่ 4.11 ผลการทดลองพยากรณ์โดยแบบจำลองเครือข่ายประสาทที่ใช้น้ำหนักและไบอัสเริ่มต้นจาก GA สำหรับความต้องการไฟฟ้าในฤดูหนาว จำนวน 10 ครั้ง

Model Code	generation for getting optima	fitness	MAPE-peak	MAPE-valley	MAPE-forecast
GW_1	1000	5.76	4.68	14.06	8.11
GW_2	1000	2.76	5.91	20.12	7.06
GW_3	1000	3.52	9.52	17.17	9.37
GW_4	1000	3.06	8.15	17.84	8.96
GW_5	1000	4.10	7.28	17.87	11.20
GW_6	1000	4.40	4.52	13.56	7.09
GW_7	1000	2.51	7.74	10.58	9.87
GW_18	1000	4.02	10.02	15.62	9.67
GW_9	1000	3.42	9.48	18.33	12.42
GW_10	1000	4.47	7.28	18.55	8.65
average		3.80			9.24

ตารางที่ 4.12 ผลการทดลองพยากรณ์โดยแบบจำลองเครือข่ายประสาทที่ใช้น้ำหนักและไบอัส เริ่มต้นจาก GA สำหรับความต้องการไฟฟ้าในฤดูร้อน จำนวน 10 ครั้ง

Model Code	generation for getting optima	fitness	MAPE-peak	MAPE-valley	MAPE-forecast
GS_1	1000	1.40	18.47	3.51	11.97
GS_2	1000	1.14	14.21	5.81	10.89
GS_3	1000	2.24	18.39	9.51	11.60
GS_4	1000	1.72	17.59	8.22	12.16
GS_5	1000	2.01	16.92	9.92	10.62
GS_6	1000	2.82	19.59	3.05	13.62
GS_7	1000	0.62	16.64	10.35	14.64
GS_8	1000	0.89	16.21	5.49	10.81
GS_9	1000	1.64	15.93	9.57	12.55
GS_10	1000	2.19	22.83	10.58	18.18
average		1.67			12.70

ตารางที่ 4.13 ผลการทดลองพยากรณ์โดยแบบจำลองเครือข่ายประสาทที่ใช้น้ำหนักและไบอัส เริ่มต้นจาก GA สำหรับความต้องการไฟฟ้าในฤดูฝน จำนวน 10 ครั้ง

Model Code	generation for getting optima	fitness	MAPE-peak	MAPE-valley	MAPE-forecast
GF_1	1000	3.93	9.07	8.56	9.10
GF_2	1000	4.32	8.16	8.10	8.85
GF_3	1000	3.27	5.07	11.18	8.45
GF_4	1000	3.01	7.68	10.69	9.08
GF_5	1000	3.42	6.98	12.45	10.16
GF_6	1000	3.31	8.89	9.30	9.95
GF_7	1000	6.65	5.61	8.87	7.69
GF_8	1000	2.20	6.99	11.11	8.83
GF_9	1000	4.77	8.14	9.69	8.83
GF_10	1000	3.16	6.47	8.90	8.49
average		3.80			8.94

ซึ่งจากผลการทดลอง จะเห็นว่า การที่ GA มีจำนวนรุ่นของวิวัฒนาการจำนวนมากถึง 1000 รุ่น ไม่ได้หมายความว่า จะได้ค่าความเหมาะสมของคำตอบสูงขึ้น เสมอไป นั่นแสดงให้เห็นว่า กระบวนการสุ่มค้นหาคำตอบ โดยใช้ค่าของจำนวนรุ่นในการวิวัฒนาการเป็นดัชนีกำหนดขอบเขตการทำงานของ GA อาจจะไม่สามารถจะทำให้ได้ ค่าน้ำหนักและไบอัสที่สามารถนำไปใช้สำหรับเครือข่ายประสาท ซึ่งจากผลการพยากรณ์ความต้องการไฟฟ้า พบว่าจะมีค่าความผิดพลาดสูงพอสมควร เมื่อเทียบ กับ กรณีกำหนดเงื่อนไขการทำงานของ GA ให้สิ้นสุดเมื่อค่าความเหมาะสมของประชากรรุ่นลูก มีค่าเท่ากันหมด นอกจากนี้ เมื่อพิจารณาค่าความเหมาะสมของคำตอบจาก GA ที่ได้จากผลการทดลองแต่ละฤดูกาล พบว่า ในฤดูร้อน จะมีค่าความเหมาะสมเฉลี่ยต่ำ คือ 1.67 ในขณะที่ในฤดูหนาว และฤดูฝน มีค่าความเหมาะสมเฉลี่ยสูงกว่าและมีค่าเท่ากัน คือ 3.80 ผลการพยากรณ์ในฤดูร้อนได้ค่าความผิดพลาดของการพยากรณ์ คือ MAPE เป็น 12.70 % ซึ่งสูงกว่า ฤดูหนาว และฤดูฝน ซึ่งมี MAPE เพียง 9.24 % และ 8.94 % เท่านั้น ซึ่งแสดงว่า ค่าความเหมาะสมของคำตอบจาก GA มีผลต่อสมรรถนะของเครือข่ายประสาท ในการพยากรณ์ความต้องการไฟฟ้าด้วย

ค่าน้ำหนักและไบอัสที่ได้จากการทดลอง GA ในตารางที่ 4.11 ถึงตารางที่ 4.13 (ทุกแบบจำลอง) เมื่อทดลองนำไปใช้เป็นค่าน้ำหนักและไบอัสเริ่มต้น แก่เครือข่ายประสาท แล้วทำการสอนให้เครือข่ายประสาทที่มีการเรียนรู้แบบแพร่ความผิดพลาดย้อนกลับ ด้วยอัตราการเรียนรู้เริ่มต้น 0.05 ค่าโมเมนตัม 0.8 เป้าหมายการเรียนรู้เป็น MSE เท่ากับ 0.001 จะได้ผลการพยากรณ์และจำนวนครั้งที่ใช้ในการสอน ดังตารางที่ 4.14

จากผลการทดลอง จะเห็นว่า การใช้ GA มาช่วยในการหาค่าน้ำหนักเริ่มต้นแก่เครือข่ายประสาท แล้วทำการสอนให้เครือข่ายเรียนรู้ จนถึงเป้าหมาย MSE = 0.001 จะต้องใช้จำนวนครั้งในการสอนมากถึงประมาณ 7500 ครั้ง จึงจะเสร็จสิ้น โดยที่เครือข่ายประสาทนี้ เมื่อนำไปใช้ในการพยากรณ์ความต้องการไฟฟ้าในฤดูหนาว ฤดูร้อน และฤดูฝน จะได้ค่าเฉลี่ยของเปอร์เซ็นต์ความผิดพลาดสัมบูรณ์ (MAPE) เป็น 7.10 % 10.02 % และ 7.76 % ตามลำดับ เหตุที่ค่าพยากรณ์ได้ในฤดูร้อนมีค่า MAPE สูงกว่า อาจจะเกิดเนื่องจาก เนื่องจาก ค่าน้ำหนักและไบอัส เริ่มต้นที่ใช้ในการสอนมีค่าความเหมาะสมต่ำที่สุด ทั้งๆที่ จำนวนครั้งในการสอนเป็น 7589 ครั้ง เช่นเดียวกับในฤดูฝน

ตารางที่ 4.14 ผลการทดลองใช้ผลจาก GA มาเป็นน้ำหนักและไบอัสเริ่มต้นแก่เครือข่ายประสาท

Model Code	summer		fall		winter	
	Epochs	MAPE-forecast	Epochs	MAPE-forecast	Epochs	MAPE-forecast
NG_1	8025	13.71	1068	8.89	6896	9.10
NG_2	7439	11.83	6434	10.07	8192	7.10
NG_3	2299	11.08	6044	8.23	5213	8.14
NG_4	7589	10.02	3159	7.76	4943	11.62
NG_5	2218	11.42	3576	9.80	5303	10.76
NG_6	801	13.23	5401	9.55	3616	9.32
NG_7	11061	10.88	40000	8.06	4602	12.39
NG_8	16873	10.30	3930	9.90	3505	9.03
NG_9	5546	13.47	1380	8.49	2838	13.25
NG_10	6153	15.86	3377	8.05	3297	9.08

มหาวิทยาลัยเชียงใหม่ Chiang Mai University