รายการอ้างอิง

1 McCulloch, W.S., and Pitts, W. A Logical Calculus of the Ideas Immanent in Neurous Activity. Bulletin of Mathmatical Biophysics. 5 (1943) : 115-133.

2 Stern, H.s. Neural Network in Applied Statistics. Technometrics. 38 (1996): 205-214.

3 Laurene Fausett. Fundamental of Neural Networks. Florida Institute of Technology, Prentice Hall International (n.d.).

4 Rosenblatt, F. The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain. Psychological Review. 65 (1959): 386-408.

5 Minsky, M., and Papert, S. Perceptron. Cambridge, MA: MIT Press, 1969.

6 Schlkoff, R.J. Artificial Neural Network. New York: McGraw-Hill, 1997.

7 Donald R. Tveter. Backpropagator's Review. Available from: http://gannoo.uce.ac.uk/bpr/art1.html

8 Fahlman, S.E., and Lebiere, C. The Cascade-Correlation Learning Architecture. Advances in Neural Information Processing Systems 2. (1990): 524-532.

9 J.R. Chen and P. Mars. Stepsize Variation Methods for Accelerating the Back-Propagation Algorithm. IJCNN-90-WASH-DC. 1 (1990): 601-604.

10 Schiffmann, W., Joost, M., and Werner, R. Optimization of the Backpropagation Algorithm for Training Multilayer Perceptrons. Available from: ftp://archive.cis.ohio-state.edu/pub/neuroprose/schiff.bp_speedup.ps.Z

11 Eduardo Somtag. On the Recognition Capabilities of Feed Forward Nets. Available from: ftp://archive.cis.ohio-state.edu/pub/neuroprose/Somtag.recognition.ps.Z

12 Yoshio Izui and Alex Pentland. Speeding up Back-Propagation. IJCNN-90-WASH-DC. 1 (1990): 639-642.

13 Tom Tollenaere. Super SAB:Fast Adaptive Back-Propagation with Good Scaling Properties. Neural Networks. 3 (n.d.): 561-573.

14 Jonathan Richard Shewchuk. An Introduction to the Conjugate Gradient Method without the Agonizing Pain. Available from: ftp://archive.cis.ohio-state.edu/pub/neuroprose

15 Fahlman, S.E. Faster-Learning Variations on Back-Propagation. Proceedings of the 1988 Connectionist Models Summer School. (1989): 38-51.

16 Lutz Prechelt. Investigation of the Cascade Family of Learning Algorithms. Neural Networks, 10 (1997): 885-896.

17 Merten Joost and Wolfrann Schiffmann. Speeding Up Backpropagation Algorithms by Using Cross-Entropy Combined with Pattern Normalizaiton, The University of Karlsruhe, Germany. (n.d.)

18 Dogan Alpsan., et al. Efficiency of Modified Backpropagation and Optimization Methods on a Real-work Medical Problem. Neural Networks. 8 (n.d.): 945-962.

19 Moshe Leshno., et al. Multilayer Feedforward Networks with a Nonpolynomial Activation Function Can Approximate any Function. Neural Networks, 6(1993): 861-867.

20 Barry L. Kalman and Stan C. Kwasny. TRAINREC: A System for Training Feedforward and Simple Recurrent Networks Efficiently and Correctly. Available from: ftp://archive.cis.ohio-state.edu/pub/neuroprose

21 D. Anguita, G.Parodi and R.Zunino. Speed Improvement of the Back-Propagation on Current Generation Workstations, Proceedings of the World Congress on Neural Networking. 1 (1993): 165-168.

22 David Elliott. A better Activation Function for Artificial Neural Networks. Available from: ftp://archive.cis.ohio-state.edu/pub/neuroprose

23 Eduardo D.Somtag. On the Recognition Capabilities of Feedforward Nets. Availabe from: ftp://archive.cis.ohio-state.edu/pub/neuroprose/Somtag.recognition.ps.Z

24 Danial L. Chester. Why Two Hidden Layers are Better Than One. IJCNN-90-WASH-DC. 1 (1990): 265-268.

25 Patrick M. Shea and Felix. Operation Experience with a Neural Network in the Detection of Explosives in Checked Airline Luggage. IJCNN San Diego. 2(1990): 175-178.

26 Alvin Surkan and J.Clay Singleton. Neural Networks for Bend Rating Improved by Multiple Hidden Layers. IJCNN San Diego. 2 (1990): 157-165.

27 B. Jacques de Villiers and Etienne Barnard. Backpropagation Neural Networks with One and Two Hidden Layers. IEEE Transactions on Neural Networks. 4(1992): 136-141.

28 Shin'ichi Tamura and Masahiko Tateishi. Capabilities of a Four-Layered Feedforward Neural Network Four Layers Versus Three. IEEE Transaction on Neural Networks. 8 (1997).

29 Michael P.Perrone and Leon N.Cooper. When Networks Disagree:Ensemble Methods for Hybrid Networks. Available from: ftp://archive.cis.ohio-state.edu/pub/neuroprose

30 William Finnoff, Ferdinand Hergert and Hans Georg Zimmermann. Improving Model Selection by Nonconvergent Methods. Neural Networks. 6 (1993): 771-783.

31 Joachim Utans and John Moody. Selecting Neural Network Architectures via the Prediction Risk: Application to Corporate Bond Rating Prediction. Available from: ftp://archive.cis.ohio-state.edu/pub/neuroprose

32 Bhat N.V. and T.J. McAvoy. Use of Neural Nets for Dynamic Modeling and Control of Chemical Process Systems. Computer Chem Eng. 14 (1990): 573-582.

33 Willis, M.J.Massimo, et al. Artificial Neural Networks in Process Engineering. IEEE Proceedings. 139 (1991).

34 Dawson,M. and Schopflocher,D. Modifying the Generalized Delta Rule to Train Networks of Non-Monotonic Processor for Pattern Classification. Connection Science. 1 (1991): 19-31.

35 Nahas, E.P., Henson, M.A. and Seborg, D.E. Nonlinear Internal Model Control Strategy for Neural Networks Models. Computers Chem. Eng. 16 (1992): 1039-1057.

36 Pollard, et al. Improving Distillation Column by using Neural Network. Available from: http://www..Latter.Demon.Co.UK/Technical/Neuralnetworks/Page8.Html

37 สุรพล คำสุภา. การสร้างแบบจำลองกระบวนการไม่เชิงเส้นโดยการใช้ข่ายงานนิวรอน. วิทยานิพนธ์ภาควิศวกรรมเคมี, จุฬาลงกรณ์มหาวิทยาลัย 2538.

38 R.Baratti, G.Uacca and A.Servida. Neural Network Modeling of Distillation Columns. Hydrocarbon Processing. (1995): 35-38.

39 C.J.Quek., R.Balasubramanian and G.P.Rangaiah. Consider Using Soft Analyzers to Improve SRU Control. Hydrocarbon Processing. (2000): 101-106.

40 K.W. Lee and H.N. Lam. Proceedings of the Second New Zealand Two-Stream Int'l Conference on Artificial Neural Networks and Expert Systems (ANNES '95). Dept. of Mechanical Eng. University of Hong Kong. 1995.

41 Zvi Boger. Proceedings of the Second New Zealand Two-Stream Int'l Conference on Artificial Neural Networks and Expert Systems (ANNES '95). Dept. of Mechanical Eng. University of Hong Kong. 1995.

42 A.L. Riddle. Neural Networks Help Optimize Solvent Extraction. Hydrocarbon Processing. 77 (1998).

43 Srdjan Nesic and Miran Vrhovac. A Neural Network Model for CO2 Corrosion of Carbon Steel. The Journal of Corrosion Science and Engineering (JCSE). 1(1999).

44 A.Barsamian and J.Macias. Inferential Property prediction using neural network. Hydrocarbon Processing. (77) 1998: 107-115.

45 H.H.Chen, M.T. Manry and Hema Chandrasekaran. A Neural Network training Algorithm Utilizing Multiple Sets of Linear Equations. Neurocomputing. 25(1999): 55-72.

46 K.B. McAuley and J.F.MacGregor. On-Line Inference of Polymer Properties is an Industrial Polyethylene Reactor. AICHE Journal. 37 (1991): 825-835.

47 Laurene Fausett. Fundamentals of Neural Network. Florida Institute of Technology, Prentice Hall International 1994.

48 R.Neelakantan and J.Guver. Applying Neural Network. Hydrocarbon Processing. (1998): 91-96

ภาคผนวก

ภาคผนวก ก.

โปรแกรมแม็ทแลบ(Matlab)

โปรแกรมแม็ทแลบ (Matlab) เป็นโปรแกรมที่ใช้ในการคำนวณทางคณิตศาสตร์ขั้นสูงและใช้แก้ปัญหาที่เป็นแมทริกซ์และเวคเตอร์ การเขียนโปรแกรมด้วยแม็ทแลบจะง่ายกว่าโปรแกรมภาษาอื่น ๆ เนื่องจากโปรแกรมแม็ทแลบได้รวมคุณสมบัติที่ดีของโปรแกรมภาษาอื่นเข้ามาเช่นการที่ไม่ต้องประกาศชื่อและชนิดของตัวแปรที่เหมือนกับโปรแกรมภาษาเบสิก และการเขียนโปรแกรมที่เป็นโครงสร้างเข้าใจง่าย ที่ใช้ในโปรแกรมภาษาซีและภาษาปาสคาล

แม็ทแลบเป็นภาษาคอมพิวเตอร์ชั้นสูง (high-level language) สำหรับการคำนวณทางเทคนิคที่ประกอบด้วยการคำนวณเชิงตัวเลข กราฟิกที่ซับซ้อนและการจำลองแบบเพื่อให้มองเห็นภาพพจน์ได้ง่ายและชัดเจน ชื่อของแม็ทแลบ (MATLAB) ย่อมาจาก matrix laboratory เดิมโปรแกรม MATLAB ได้เขียนขึ้นมาเพื่อใช้ในการคำนวณทางแมทริกซ์ ที่พัฒนาจากโครงงานที่ชื่อ LINKPACK และ EISPACK

แม็ทแลบได้พัฒนามาด้วยการแก้ปัญหาที่ส่งมาจากหลายๆ ผู้ใช้เป็นระยะเวลาหลายปีจึงทำให้โปรแกรมแม็ทแลบมีฟังก์ชันต่างๆ ให้เลือกใช้มากมาย ในบางมหาวิทยาลัยได้ใช้โปรแกรม MATLAB เป็นหลักสูตรพื้นฐานในการศึกษาทางด้านคณิตศาสตร์ วิศวกรรม และวิทยาศาสตร์แขนงต่างๆ ตลอดจนในด้านอุตสาหกรรมมได้ใช้โปรแกรมแม็ทแลบเป็นเครื่องมือสำหรับใช้ในงานวิจัย พัฒนาและวิเคราะห์

1. ความสามารถของโปรแกรมแม็ทแลบ

โปรแกรมแม็ทแลบเป็นโปรแกรมที่สามารถโต้ตอบกับผู้ใช้งานแบบทันทีทันใดและได้มีการพัฒนามาอย่างต่อเนื่อง ในการศึกษาโปรแกรมแม็ทแลบจะถูกใช้ไปในการวิจัยและการสอนทางคณิตศาสตร์ ในทางอุตสาหกรรมโปรแกรมแม็ทแลบจะใช้ในการวิจัยทางวิศวกรรม และการแก้ปัญหาคณิตศาสตร์เช่น การควบคุมกระบวนการแบบอัตโนมัติ และการวิจัยสัญญาณของกระบวน การทำงานของโปรแกรมแม็ทแลบจะเป็นฟังก์ชันของคำสั่งที่อยู่ในรูปของโปรแกรม "M-File" ซึ่งฟังก์ชันเหล่านี้เป็นเครื่องมือที่ใช้ในการหาคำตอบเรียกว่าทูลบ็อกซ์ (Toolbox) โดยโปรแกรมแม็ทแลบจะมีทูลบ็อกซ์ในแต่ละสาขา เช่น

ก. การคำนวณและการแก้ปัญหาด้วยคุณสมบติพิเศษของแมทริกซ์

ข. การคำนวณโพลีโนเมียล

ค. การจัดการเกี่ยวกับเวคเตอร์ และการวิเคราะห์ข้อมูล

ง. การจัดการเกี่ยวกับการแสดงผลกราฟ

จ. การติดต่อสื่อสาร (communication toolbox)

ฉ. การแก้ปัญหาทางสถิติ (statistics toolbox)

ช. เวฟเลท(wavelet)

ซ. การประมวลผลภาพ (image processing toolbox)

ฌ. การแก้ปัญหาด้านระบบควบคุม (control system toolbox)

ญ. โครงข่ายนิวรัล (neural networks toolbox)

ฎ. ฟัซซี่ลอจิก (fuzzy logic toolbox)

ฏ. การประมวลผลสัญญาณ (Signal processing toolbox)

ฐ. เครื่องมือในการสร้างแบบจำลองในการควบคุมแบบโมเดลพรีดิกทีฟ (Model Predictive Control Toolbox)

ฑ. การออกแบบระบบควบคุมไม่เชิงเส้น (Nonlinear Control Design Toolbox)

ฒ. การออฟติไมซ์เซชัน (Optimization Toolbox)

2. ลักษณะเด่นของโปรแกรมแม็ทแลบ

2.1 มีฟังก์ชันคณิตศาสตร์ให้เลือกใช้ในการคำนวณมากมายตลอดจนเราสามารถสร้างฟังก์ชันขึ้นมาใช้งานได้เองในสาขาที่ต้องการ

2.2 อัลกอริธึมพัฒนาได้ไม่ยุ่งยาก สามารถแก้ไขปัญหาทางด้านคณิตศาสตร์ที่มีความซับซ้อนได้ง่ายและรวดเร็วกว่าโปรแกรมภาษาอื่นๆ เช่น ภาษาซี (C), ฟอร์แทรน (Fortran), ภาษาเบสิก (Basic) เป็นต้น

2.3 มีโครงสร้างแบบจำลอง (simulink) ซึ่งเป็นเครื่องมือที่สามารถนำไปสร้างบล็อกไดอะแกรมเพื่อใช้ทดสอบ และประเมินผลระบบพลศาสตร์ (dynamic) ต่างๆ ก่อนนำไปใช้งานจริง

2.4 สามารถวิเคราะห์และตรวจสอบข้อมูลได้ง่ายและรวดเร็ว

2.5 นำไปใช้ในงานทางด้านกราฟิกได้เป็นอย่างดีทั้งในด้านการแสดงภาพตั้งแต่สองมิติ รวมทั้งภาพสามมิติในรูปแบบพื้นผิวและระดับสูงต่ำ (contour) ตลอดจนสามารถนำภาพมาต่อกัน และเก็บไว้เพื่อที่จะสร้างเป็นภาพเคลื่อนไหวได้อีกด้วย

2.6 ประยุกต์ใช้ในการสร้างรูปแบบปฏิสัมพันธ์ทางภาพ (graphical user interface) ได้โดยการเลือกใช้วัตถุ (object) และเมนูต่างๆ โดยโปรแกรมแม็ทแลบจะมีเครื่องมือให้เลือกใช้ เช่น เมนู,รายการ,ปุ่มกด เพื่อให้ผู้ใช้สามารถเลือกนำไปใช้ในการทำงานปฏิสัมพันธ์กันระหว่างผู้ใช้กับเครื่องคอมพิวเตอร์

2.7 ทำการประมวลผลร่วมกับโปรแกรมอื่นได้ เช่น ฟอร์แทรน (Fortran) , (บอร์แลนด์)Borland ,ซี หรือ ซีพลัสพลัส หรือ ไมโครซอฟท์วิชวลซี (C/C++ Microsoft visual C++) และ วัทคอมซี (Watcom C/C++) ด้วยการเขียนฟังก์ชันที่เป็นเม็กซ์ไฟล์โดยโปรแกรมแม็ทแลบจะเรียกใช้ รูทีนจากโปรแกรมภาษาซีและฟอร์แทรน

2.8 โปรแกรมแม็ทแลบ เป็นระบบที่มีส่วนของข้อมูลพื้นฐานเป็นอาร์เรย์ที่ไม่ต้องการมิติ ทำให้โปรแกรมแม็ทแลบสามารถทำการแก้ปัญหาทางเทคนิคต่างๆ ได้มากใช้เวลาในการประมวลผลน้อย และดีกว่าโปรแกรมภาษาซีและฟอร์แทรน

2.9 โปรแกรมแม็ทแลบ มีเครื่องมือสำหรับการสร้างแบบจำลองของโครงข่ายนิวรัล (neural network) ซึ่งมีหลายอัลกอริธึมให้เลือกสำหรับการสร้างข่ายงาน ในงานวิจัยนี้ได้ศึกษาการสร้างข่ายงานนิวรัลด้วยโปรแกรมแม็ทแลบโดยใช้อัลกอริธึมแบบการกระจายความผิดพลาดย้อนกลับแบบ Levenberg -Marquardt

3. โครงสร้างของแม็ทแลบ

โครงสร้างของโปรแกรมแม็ทแลบประกอบด้วย 5 ส่วนใหญ่ๆ คือ

3.1 ภาษาโปรแกรมแม็ทแลบ (The MATLAB language)

แม็ทแลบเป็นโปรแกรมภาษาชั้นสูงที่ใช้ควบคุม ฟังก์ชัน โครงสร้างข้อมูลอินพุท/เอาท์พุท และลักษณะโปรแกรม object-oriented programming(เป็นแนวความคิดในการเขียนโปรแกรมให้ความสนใจกับสิ่งต่างในโปรแกรม เช่น ปุ่มกด , เมนู) ทำให้การเขียนโปรแกรมไม่ยุ่งยากเมื่อเทียบกับการเขียนโปรแกรมด้วยภาษาอื่นๆ เช่น ซี , ฟอร์แทรน หรือเบสิก เป็นต้น

3.2 สถาปัตยกรรมในการทำงานของแม็ทแลบ (The MATLAB working environment)

แม็ทแลบจะมีกลุ่มของเครื่องมือที่เป็นประโยชน์สำหรับการทำงานของผู้ใช้โปรแกรมหรือโปรแกรมเมอร์ ประโยชน์คือการจัดการตัวแปรในเวิร์คสเปซ (workspace) การนำข้อมูลหรือการผ่านค่าตัวแปรเข้า/ออก และกลุ่มเครื่องมือต่างๆนี้ก็จะใช้สำหรับพัฒนา,จัดการ,ตรวจสอบความผิดพลาดของโปรแกรม (debugging) ที่ได้เขียนขึ้น

3.3 ฟังก์ชันในการคำนวณทางคณิตศาสตร์ (The MATLAB mathematical function library)

แม็ทแลบ จะมีไลบรารีทั่วไปที่ใช้ในการคำนวณอย่างกว้างเช่น sine ,cosine และฟีชคณิตเชิงซ้อน โดยสามารถนำไปประยุกต์ใช้เป็นฟังก์ชันหรือไลบรารีเพิ่มเติมขึ้นจากไลบรารีที่ใช้กันโดยทั่วไป เช่นฟังก์ชันในการหา eigenvalues และ eigenvectors การแยกตัวประกอบและส่วนประกอบของเมตริกซ์ด้วยวิธีต่างๆการวิเคราะห์ข้อมูล การหาความน่าจะเป็น และการแก้ปัญหาระบบของสมการเชิงเส้นที่เป็นพื้นฐานของสาขาวิชาต่างๆ เป็นต้น ทำให้โปรแกรมแม็ทแลบมีฟังก์ชันสำหรับใช้งานค่อนข้างมากและครอบคลุมรายละเอียดของการคำนวณในสาขาวิชาต่างๆได้มากขึ้น

3.4 กราฟ (Handle Graphics)

ระบบกราฟิกของแม็ทแลบจะประกอบด้วยคำสั่งชั้นสูงสำหรับการพล็อตกราฟโดยมีพื้นฐานอยู่บนแนวความคิดที่ว่าทุกสิ่งบนหน้าต่างรูปภาพของโปรแกรมแม็ทแลบจะเป็นวัตถุซึ่งมีเอกลักณ์เฉพาะตัว Handle graphics จะประกอบด้วยคำสั่งชั้นสูงให้ได้เลือกใช้ในการสร้างปฏิสัมพันธ์กับผู้ใช้ (graphic user interface) บนพื้นฐานการประยุกต์ใช้งาน นอกจากนี้โปรแกรมแม็ทแลบยังมีฟังก์ชันที่ใช้สำหรับการแสดงภาพสองมิติ ภาพสามมิติและการสร้างภาพเคลื่อนไหว

3.5 ระบบอินเทอร์เฟสของโปรแกรมแม็ทแลบ (The MATLAB Application Program Interface (API))

API จะใช้เพื่อสนับสนุนการติดต่อจากภายนอกโดยใช้โปรแกรมที่เป็นเม็กซ์ไฟล์ ฟังก์ชันในแม็ทแลบจะเรียกใช้ตามปกติจากโปรแกรมภาษาซีและฟอร์แทรน หรืออาจกล่าวได้ว่า API เป็นไลบรารีที่เขียนด้วยโปรแกรมภาษาซีและฟอร์แทรนที่มีการเชื่อมต่อกับโปรแกรมแม็ทแลบด้วยไฟล์ที่เป็นเม็กซ์ฟังก์ชันอีก

4. ข้อจำกัดของโปรแกรมแม็ทแลบ

เนื่องจากโปรแกรมแม็ทแลบเป็นโปรแกรมที่ใช้สำหรับการคำนวณทางคณิตศาสตร์และกราฟิกที่ซับซ้อนดังนั้นจึงจำเป็นต้องใช้เครื่องคอมพิวเตอร์ที่มีความเร็วสูง คอมพิวเตอร์พีซีที่เหมาะสมกับโปรแกรมแม็ทแลบคือมีระบบประมวลผล (CPU) รุ่นเพนเทียมขึ้นไป หน่วยความจำ (RAM) ควรมีอย่างต่ำ 32 เมกกะไบต์ ส่วนพื้นที่จัดเก็บข้อมูล (Hardisk) ควรมีเนื้อที่ว่างเกิน 80 เมกกะไบต์

5. การเขียนโปรแกรมด้วยคำสั่งในโปรแกรมแม็ทแลบ

โปรแกรมแม็ทแลบจะรับคำสั่งทีละ 1 บรรทัดในพื้นที่หน้าต่างของโปรแกรม หลังจากรับคำสั่งแล้วโปรแกรมแม็ทแลบจะทำการประมวลผลและแสดงผลทางหน้าต่างการทำงานและสามารถสร้างไฟล์ที่ประกอบไปด้วย

ชุดคำสั่งจะเก็บอยู่ในรูป "ชื่อไฟล์.M" คือมีนามสกุลของไฟล์เป็น .M หรือเรียกว่าเอ็มไฟล์ (M-files) การประมวลผลจะทำการประมวลผลทีละคำสั่งตามลำดับก่อนหลัง การเขียนเอ็มไฟล์มี 2 รูปแบบคือ สคริปไฟล์และฟังก์ชันไฟล์

การเขียนโปรแกรมแม็ทแลบด้วยเอ็มไฟล์ (M-File) นี้จะมีรูปแบบการเขียนโปรแกรมเป็น 2 รูปแบบ คือ

สคริปท์ เอ็มไฟล์ (Script M-files) และเอ็มไฟล์ฟังก์ชัน (M-files functions) ทั้ง สคริปท์ เอ็มไฟล์ (Script M-files) และเอ็มไฟล์ฟังก์ชัน (M-files functions) จะคล้ายคลึงกันคือเป็นในไฟล์ที่มีนามสกุล .m ซึ่งสร้างจากหน้าต่างโปรแกรมเอดิเตอร์ (editor) โปรแกรมที่เป็นเอ็มไฟล์ ฟังก์ชันจะทำการผ่านเข้า/ออกค่าตัวแปรต่างๆให้กับแม็ทแลบเวิร์คสเปซ (MATLAB workspace) เฉพาะค่าผลลัพธ์ที่เป็นเอาท์พุทของคำสั่งเท่านั้น ส่วนโปรแกรมที่เป็นสคริปท์ เอ็มไฟล์ จะทำการผ่านเข้า/ออกค่าตัวแปรต่างๆทุกตัวภายในโปรแกรมให้กับแม็ทแลบเวิร์คสเปซ ในหัวข้อนี้จะอธิบายถึงโครงสร้างและกฎการเขียนเอ็มไฟล์ดังนี้

ก. สคริปท์ไฟล์ (Script M-files)

การกำหนดค่าตัวแปรและเขียนคำสั่งหรือฟังก์ชันต่างๆที่ต้องการประมวลผลทีละคำสั่งลงในเท็กซ์ไฟล์ (text file)ที่เป็นไฟล์นามสกุล .m แล้วสามารถทำการประมวลผลได้ด้วยการพิมพ์ชื่อไฟล์โดยไม่ต้องตามด้วยนามสกุล .m

สคริปไฟล์เป็นไฟล์ซึ่งเป็นลำดับคำสั่ง คำอธิบายในสคริปไฟล์สามารถเขียนได้โดยใช้ "%" นำหน้าข้อความที่อธิบายนี้จะไม่มีผลต่อการทำงานของสคริปไฟล์ ใช้โอปะเรเตอร์ในสคริปไฟล์ได้แก่ ยกกำลัง (^) คูณ (*) หาร(/) บวก(+) และลบ(-)

```
% Program 'example.m'
% This program compute the matrix inverse
A=[1 5 ; 3 7];
B=[3 4 ; 1 2];
C=A*B;
D=inv(A)
```

หลังจากพิมพ์โปรแกรมเสร็จเรียบร้อยแล้ว ให้ทำการเก็บไฟล์โดยทำการบันทึก(save)เป็นเอ็มไฟล์และทำการพิมพ์ชื่อไฟล์โดยไม่ต้องตามด้วยนามสกุล .m เช่นบันทึกไว้ในชื่อ example และเมื่อต้องการประมวลผลให้พิมพ์คำว่า example หลังพร็อมพ์ (>>) ที่หน้าจอคำสั่งแล้วโปรแกรมจะแสดงผลลัพธ์ที่ได้ออกมาที่หน้าจอคำสั่ง

```
>> example
D=
    -0.8750        0.6250
     0.3750       -0.1250
```

ข. เอ็มไฟล์ ฟังก์ชัน (M-files)

โปรแกรมแม็ทแลบได้สร้างเอ็มไฟล์ฟังก์ชันที่เป็นเครื่องมือในการคำนวณด้านคณิตศาสตร์และสาขาอื่นๆ ทั้งนี้เพื่อความสะดวกในการเขียนโปรแกรม ตัวอย่างเช่น ฟังก์ชัน inv ,abs, angle,conv และ sqrt เป็นต้น ในบางครั้งฟังก์ชันโปรแกรมแม็ทแลบสร้างขึ้นไม่สามารถแก้ปัญหาได้โดยตรง แต่สามารถนำฟังก์ชันต่างๆเหล่านี้มาใช้ประกอบกับการเขียนโปรแกรมเอ็มไฟล์ใหม่ได้เพื่อการประยุกต์ใช้งาน ตัวอย่างเช่น

```
function    y = fliplr(x)
if ndims(x)~=2 error('X must be a 2-D matrix.'); end
[m,n] =size(x)
y = X(:,n:-1:1)
```

6. โครงสร้างโปรแกรมพื้นฐาน

6.1 โครงสร้างแบบมีเงื่อนไข

กระบวนการทำงานแบบมีเงื่อนไขในโปรแกรมแม็ทแลบจะเป็นการควบคุมให้มีการเลือกการทำงานในโมดูลหนึ่งจากหลายๆ โมดูลตามเงื่อนไขที่กำหนด การทำงานในลักษณะนี้เรียกว่าโครงสร้างแบบ if โครงสร้างแบบมีเงื่อนไขนี้จะแบ่งออกเป็น

6.1.1 เลือกทำหรือไม่ มีโครงสร้างดังนี้

```
if  Condition
        Module 1
end
```

6.1.2 เลือกทำอย่างใดอย่างหนึ่งจาก 2 โครงสร้างที่มี ตัวอย่างเช่น

```
if Condition
    Module 1
else
    Module 2
end
```

ถ้าหากเงื่อนไขเป็นจริงจะกระทำโมดูล 1 แต่ถ้าเงื่อนไขเป็นเท็จก็จะกระทำโมดูล 2

6.1.3 ถ้าโปรแกรมที่ต้องการเขียนมีความซับซ้อนและต้องตัดสินใจก็จะใช้เงื่อนไขนี้คือ เลือกอย่างใดอย่าง หนึ่งจากหลายๆ สเตทเมนท์ ซึ่งมีโครงสร้างดังนี้

```
if Condition 1
  Module 1
elseif  Condition 2
  Module 2
elseif Condition N
  Module N
else
  Module M
end
```

6.2 โครงสร้างการทำงานแบบซ้ำ

โครงสร้างการทำงานแบบซ้ำจะมี อยู่ 2 แบบ คือการทำงานซ้ำที่ได้กำหนดตัวแปรควบคุมจำนวนรอบ ซึ่งเรียกว่า โครงสร้าง for และการทำงานที่ใช้เงื่อนไขควบคุมการวนรอบซึ่งเรียกว่า โครงสร้าง while โครงสร้างทั้งสองนี้อธิบายได้ดังนี้

```
while Condition
        Module
End
```

คำสั่งนี้จะกระทำ Module จนกว่าเงื่อนไขจะเป็นเท็จจึงจะข้ามไปที่คำสั่ง end

6.3 โครงสร้าง switch – case

โครงสร้างนี้เป็นโครงสร้างแบบมีเงื่อนไขที่กระทำตามรูปแบบสเตทเมนท์ switch ที่กำหนดในแต่ละ case การประมวลผล case แรกจนถึง case สุดท้ายจะสัมพันธ์กับสเตทเมนท์ switch เสมอ

```
switch Condition
case  Condition 1
        Module 1
case  Condition 2
        Module 2
otherwise
end
```

7. เครื่องมือในการสร้างข่ายงานนิวรัล (Neural Network Toolbox)

การออกแบบและจำลองข่ายงานนิวรัลที่ถูกสร้างโดยโปรแกรมแม็ทแลบสามารถนำมารวมกับการใช้งานเครื่องมืออื่นของแม็ทแลบและซิมมูลิงก์ได้ (other matlab toolbox and simulink) โปรแกรมข่ายงานนิวรัลนี้จะช่วยสนับสนุนงานวิจัยและประยุกต์ใช้กับงานทางด้านวิทยาศาสตร์,วิศวกรรมศาสตร์

ผู้สร้างเครื่องมือในการสร้างข่ายงานนิวรัล(Neural Network Toolbox) คือ Emeritus Howard Demuth, เป็นผู้ช่วยสอนที่ มหาวิทยาลัย Idaho (Idaho university) และ Mark Beale และ Martin Hagan เป็นผู้เขียนการออกแบบข่ายงานนิวรัล

เทคโนโลยีข่ายงานนิวรัล (neural network technology) ใช้เพื่อแก้ปัญหาในทางวิทยาศาสตร์ วิศวกรรมศาสตร์ และทางธุรกิจ ข่ายงานนิวรัลเหมาะกับการแก้ปัญหาเพราะข่ายงานนิวรัลจะทำการฝึกเพื่อหาวิธีทางแก้ปัญหาโดยการจดจำรูปแบบหรือการแบ่งประเภทข้อมูลและการทำนายค่าของสิ่งที่ต้องการในอนาคต

ข่ายงานนิวรัลเป็นเครื่องมือที่ดีมากในการประยุกต์ใช้กับการวิเคราะห์ที่ยากหรือมีความเป็นไปไม่ได้ เช่น การจดจำรูปแบบ (pattern recognition) และ การสร้างแบบจำลองและการควบคุมระบบที่ไม่เป็นเชิงเส้น

พฤติกรรมของข่ายงานนิวรัลจะหาได้จากวิธีการคำนวณน้ำหนักแต่ละน้ำหนักที่เชื่อมต่อกันระหว่างนิวรัลหนึ่งไปสู่อีกนิวรัลหนึ่ง น้ำหนักจะถูกปรับอย่างอัตโนมัติโดยวิธีการฝึกข่ายงานตามอัลกอริธึมแต่ละแบบจนกระทั่งค่าความผิดพลาดจะอยู่ในเกณฑ์ที่กำหนด

7.1 เครื่องมือข่ายงานนิวรัล

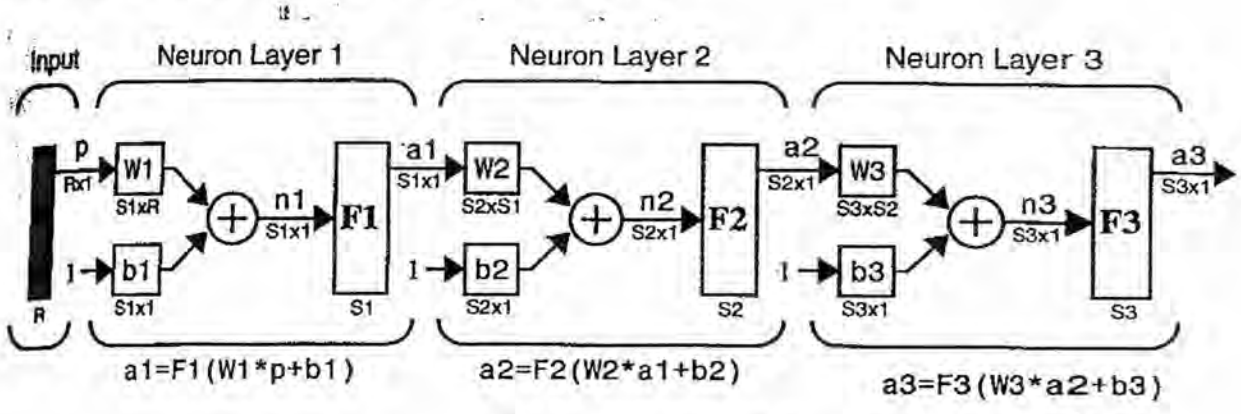
เครื่องมือแบบจำลองข่ายงานที่สามารถสร้างได้ในแม็ทแลบจะมีความยืดหยุ่นในการออกแบบ ซึ่งไม่จำกัดจำนวนของอินพุท สามารถรวมจำนวนอินพุทกับฟังก์ชันดีเลย์ (delay) สามารถถูกฝึกได้หลายวิธีแล้วแต่การออกแบบ ขั้นตอนการกำหนดค่าเริ่มต้น (initialize), ขั้นตอนการเรียนรู้ (learning), ขั้นตอนการฝึกข่ายงาน (training) และขั้นตอนการทดสอบประสิทธิภาพ (performance)

แต่ละฟังก์ชันของข่ายงานนิวรัลถูกประยุกต์ใช้กับอัลกอริธึมของการเรียนรู้ (learning rule) ได้หลายแบบ ฟังก์ชันดังต่อไปนี้เป็นฟังก์ชันที่ใช้ในข่ายงานนิวรัล

init	การจำลองข่ายงานเริ่มต้น (initialize network)
sim	การจำลองข่ายงานในขั้นสุดท้าย (simulate network)
train	การฝึกข่ายงานนิวรัล (train network)
adapt	การเรียนรู้ในการปรับค่าน้ำหนัก (adaptive learning)
disp	การแสดงผลของข่ายงาน (display network properties)

7.2 องค์ประกอบหลักของเครื่องมือข่ายงานนิวรัล

เครื่องมือข่ายงานนิวรัลมีหลายประเภทซึ่งสามารถนำไปประยุกต์ใช้ได้ในการสร้างข่ายงานนิวรัล ฟังก์ชันดังต่อไปนี้เป็นตัวอย่างฟังก์ชันที่นำไปสร้างข่ายงานนิวรัล



รูปที่ ก.1 แสดงข่ายงานนิวรัลแบบป้อนไปข้างหน้า 3 ชั้น

network	create a custom neural network

newc	create a competitive layer
newcf	create a cascade-forward backpropagation network
newff	create a feed forward backpropagation network
newfftd	create a feed-forward input-delay backpropagation network
newgrnn	design a generalized regression neural network
newhop	create a Hopfield recurren network
newlin	create a linear layer
newlind	design a linear layer
newlvq	create a learning vector quantization network
newp	create a perceptron network
newpnn	design a probabilistic network
newrb	design a radial basis network
newrbe	design an exact radial basis network
newsom	create a self – organizing map

7.2.1 แบบจำลองข่ายงานนิวรัล

โดยทั่วไป ข่ายงานนิวรัลสามารถแบ่งประเภทออกเป็นแบบมีการชี้นำ (Supervised) และแบบไม่มีการชี้นำ (Unsupervised) ความแตกต่างกันระหว่างข่ายงานที่มีการชี้นำและไม่มีการชี้นำขึ้นอยู่กับการรู้ชุดข้อมูลที่ใช้ในการฝึกข่ายงานนิวรัลนั้นคือถ้าข้อมูลมีค่าเป้าหมายอยู่ด้วยจะเป็นการเรียนรู้แบบมีการชี้นำ , ถ้าชุดข้อมูลไม่มีค่าเป้าหมายอยู่ด้วยจะเป็นการเรียนรู้แบบไม่มีการชี้นำ

ก. ข่ายงานที่มีการชี้นำ (supervised networks)

ข่ายงานที่มีการชี้นำจะถูกฝึกให้ทำนายค่าเอาท์พุทออกมาโดยมีการป้อนชุดข้อมูลอินพุทและค่าเป้าหมายเข้าไป โดยจะมีการปรับค่าน้ำหนักจนกระทั่งค่าเอาท์พุทที่ออกมาจากข่ายงานมีค่าใกล้เคียงกับค่าเป้าหมายที่ให้เข้าไป ดังนั้นจากสิ่งนี้ข่ายงานแบบนี้จึงเหมาะกับการสร้างแบบจำลองและการควบคุมระบบทางพลศาสตร์ (modeling and controlling dynamic systems) ,การแบ่งประเภทข้อมูลที่เป็นสัญญาณรบกวน(classifying noisy data). และการทำนายค่าในเหตุการณ์อนาคต (predicting future events)

- ข่ายงานนิวรัลแบบป้อนไปข้างหน้า (feed-forward networks) มีวิธีการเชื่อมต่อจากชั้นอินพุทไปสู่ชั้นเอาท์พุท ซึ่งปกติจะใช้ข่ายงานนี้สำหรับการทำนาย , การจดจำรูปแบบ (pattern recognition), การหาความสัมพันธ์ไม่เป็นเชิงเส้น (nonlinear funciton fitting) และอื่นๆ

- Radial basis networks จัดอยู่ในข่ายงานนิวรัลป้อนไปข้างแบบไม่เชิงเส้นที่เร็วอีกข่ายงานหนึ่ง ซึ่งจะประกอบด้วยการเปลี่ยนแปลงของ radial basis networks รวมทั้ง generalized regression และ probabilistic neural networks

- Recurrent networks ใช้เป็นรูปแบบที่ย้อนกลับเข้าไปฝึกข่ายงาน สถาปัตยกรรมแบบ recurrent network รวมทั้ง Elman และ Hopfield networks

- Learning vector quantization (LVQ) เป็นเครื่องมือที่ใช้สำหรับแบ่งประเภทของรูปแบบที่ไม่สามารถแบ่งเป็นเชิงเส้น

-

ข. ข่ายงานแบบไม่มีการชี้นำ (unsupervised networks)

ข่ายงานแบบไม่มีการชี้นำจะทำการฝึกข่ายงานโดยการปล่อยให้ข่ายงานปรับตัวเองอย่างต่อเนื่องด้วยอินพุทค่าใหม่ ข่ายงานนิวรัลแบบไม่มีการชี้นำสามารถหาความสัมพันธ์ภายในของชุดข้อมูลอินพุท

- Associative learning rules จะสร้างการเรียนรู้ระหว่างชุดข้อมูลอินพุทและชุดข้อมูลเอาท์พุท เพราะว่าข่ายงานชนิดนี้เป็นการเปลี่ยนแปลงเฉพาะจุด ดังนั้นจึงใช้ข่ายงานชนิดนี้ในการสร้างบล็อคสำหรับข่ายงานที่สมบูรณ์มากกว่า
- Self-organizing network เรียนรู้การตรวจสอบความสัมพันธ์ระหว่างค่าอินพุทและปรับการตอบสนองตามอินพุทที่เข้ามาใหม่ ดังนั้นอินพุทจึงแบ่งประเภทเป็นเวกเตอร์อินพุท และเรียนรู้การกระจายตัวของชุดข้อมูล

7.3 ฟังก์ชันการฝึกข่ายงานและการเรียนรู้ (Training and Learning Functions)

ฟังก์ชันการฝึกข่ายงานและการเรียนรู้คือกระบวนการทางคณิตศาสตร์ที่ใช้ปรับค่าน้ำหนักและค่าไบอัสของข่ายงานอย่างอัตโนมัติ ฟังก์ชันการฝึกข่ายจะเป็นอัลกอริธึมแบบโกลบอล (Global algorithm) นั้นคืออัลกอริธึมที่สามารถใช้ได้กับทุกโปรแกรมที่เขียนขึ้นจากโปรแกรมแม็ทแลบและมีการเรียกใช้ฟังก์ชันนั้น

ในการนำมาประยุกต์ใช้กับการสร้างข่ายงานนิวรัลสามารถนำฟังก์ชันการเรียนรู้หรือการฝึกข่ายงานมาประยุกต์ใช้ได้ตามที่ต้องการ ตัวอย่างเช่น Levenberg-Marquardt training algorithm คืออัลกอริธึมแบบการกระจายความผิดพลาดย้อนกลับชนิดหนึ่งที่เร็วกว่าแบบดั้งเดิมแต่ว่าจะใช้หน่วยความจำมาก

7.3.1 ฟังก์ชันการฝึกข่ายงานของเครื่องมือข่ายงานนิวรัล

trainbfg	BFGS quasi-Newton backpropagation
trainbr	Bayesian regularization
traincgb	Powell-Beale conjugate gradient backpropagation
traingp	Polak-Rigiere conjugate gradient backpropagation
traingd	gradient descent backpropagation
traingda	gradient descent with adaptive learning
traingdx	gradient descent with momentum and adaptive learning
trainlm	Levenberg-Marquardt backpropagation
trainoss	one-step secant backpropagaiton
trainrp	resilient backpropagation(Rprop)
trainscg	scaled conjugate gradient backpropagation
trainwb	by-weight-and-bias training

เครื่องมือนี้ยังรวมฟังก์ชันการเรียนรู้ดังต่อไปนี้

learngd	gradient descent
learnh	Hebb
learnhd	Hebb with decay
learnis	instar
learnk	kohonen
learnlvq1	LVQ1
learnlvq2	LVQ2
learnp	perceptron
learnpn	normalized perceptron
learnsom	self-organizing map
learnwh	Widrow-Hoff

7.3.2 ฟังก์ชันกระตุ้น (Transfer function)

ฟังก์ชันกระตุ้นจะเป็นการแปลงค่าสัญญาณให้เป็นเอาท์พุทของแต่ละชั้นของข่ายงานนิวรัลทั้งที่เป็นแบบเชิงเส้นหรือเป็นแบบไม่เป็นเชิงเส้น ฟังก์ชันกระตุ้นในเครื่องมือของแม็ทแลบมีดังต่อไปนี้ดังต่อไปนี้

hard limit

logistic

sigmoid

linear

radial basis

triangular basis

7.3.3 ฟังก์ชันในการแสดงผล (Visualization Functions)

เครื่องมือข่ายงานนิวรัลจะมีฟังก์ชันในการแสดงผลดังต่อไปนี้

Performance plots of training history

Hinton diagrams

Error surface plots

Self-organizing maps

Network visualization and Simulink

7.4 การลดช่วงข้อมูล (Pre and Post-Processing Function)

การฝึกข่ายงานนิวรัลจะมีประสิทธิภาพมากถ้ามีขั้นตอนการลดช่วงข้อมูล (pre-processing step) กับข้อมูลอินพุทและค่าเป้าหมาย และการขยายข้อมูล (Post-processing) จะใช้วิเคราะห์ประสิทธิภาพของข่ายงาน ซึ่งฟังก์ชันลดและขยายช่วงข้อมูลมีดังนี้ (pre and post-processing) จะมีดังต่อไปนี้

7.4.1 หลักการวิเคราะห์ส่วนประกอบ (Principal Component Analysis (PCA)

PCA algorithm,prepca สามารถใช้ลดช่วงของอินพุทเวกเตอร์

7.4.2 การวิเคราะห์หลังการฝึกข่ายงาน (Post-training Analysis)

ฟังก์ชัน postreg ใช้เป็นการวิเคราะห์ regression ระหว่างการตอบสนองของข่ายงานและเป้าหมายที่สัมพันธ์กัน

7.4.3 การลดช่วงข้อมูลระหว่างค่าต่ำสุดและค่าสูงสุด (Scale Minimum and Maximum)

ฟังก์ชันที่ใช้คือ ฟังก์ชัน premnmx ใช้สเกลค่าอินพุทและค่าเป้าหมายให้อยู่ในช่วงค่า [-1,1]

7.4.4 การลดช่วงข้อมูลโดยใช้ค่าเฉลี่ยและค่าเบี่ยงเบนมาตรฐาน (Scale Mean and Standard Deviation)

ฟังก์ชันที่ใช้คือ ฟังก์ชัน prestd ในการสเกลข้อมูลโดยใช้ค่าเฉลี่ยและค่าเบี่ยงเบนมาตรฐาน

7.5 ฟังก์ชันในการทดสอบประสิทธิภาพ (Performance Functions)

ฟังก์ชันในการทดสอบประสิทธิภาพของข่ายงานจะใช้เพื่อการคำนวณและการแสดงถึงประสิทธิภาพของข่ายงานระหว่างการฝึกข่ายงาน นั่นก็คือการวัดค่าความผิดพลาดของข่ายงาน ฟังก์ชันในการหาประสิทธิภาพมีดังต่อไปนี้

mae	mean absolute error
mse	mean squared error
msereg	mean squared error with regularization
sse	sum squared error

7.6 ฟังก์ชันที่ใช้ทั่วไป (Generalization Function)

ปัญหาส่วนใหญ่ที่ใช้ในการออกแบบข่ายงานนิวรัลคือการฝึกข่ายงานมากเกินไป (overfitting) ซึ่งจะเกิดขึ้นเมื่อข่ายงานมีการจดจำชุดข้อมูลการฝึกมากเกินไปจนทำให้ไม่มีการเรียนรู้กับชุดข้อมูลที่เข้ามาใหม่ เครื่องมือข่ายงานนิวรัลรวมวิธีแก้ไขไว้ 2 วิธีเพื่อจะหลีกเลี่ยงการฝึกข่ายงานมากเกินไป คือ การทำให้อยู่ในรูปทั่วไป (regularization) และ การหยุดในช่วงเริ่มต้น (early stopping)

การทำให้อยู่ในรูปทั่วไป (regularization) เกี่ยวข้องกับการเพิ่มเทอมของฟังก์ชัน squared error performance (SSE) เทอมนี้จะช่วยทำให้ข่ายงานมีการตอบสนองที่ราบเรียบขึ้น เครื่องมือข่ายงานนิวรัลรวม

ฟังก์ชันการฝึกข่ายงาน Bayesian regularization (trainbr) ซึ่งจะเลือกหาเทอมออฟติไมซ์ของการทำให้อยู่ในรูปทั่วไปดีที่สุด

การหยุดในช่วงเริ่มต้นคือเทคนิคในการใช้ชุดข้อมูล 3 ชุด ได้แก่ชุดข้อมูลในการฝึกข่ายงาน (training set) ชุดในการทดสอบแบบจำลอง (validation set) และชุดในการทดสอบใช้งานจริง (test set) เทคนิคนี้จะช่วยในการสร้างน้ำหนักและค่าไบอัสของข่ายงานให้อยู่ในช่วงที่ทำให้ค่าความผิดพลาดของชุดในการทดสอบแบบจำลองน้อยที่สุด

7.6 เครื่องมือในการสร้างข่ายงานนิวรัลสามารถใช้กับซิมมูลิงค์ (Simulink support)

คำสั่งที่ใช้คือ gensim เป็นการสร้างบล็อคข่ายงานอย่างอัตโนมัติโดยการใช้ร่วมกับซิมมูลิงค์

ตารางที่ ก.1 การเปรียบเทียบความเร็วของอัลกอริธึมที่เป็นแบบชี้นำ

FUNCTION	TECHNIQUE	TIME	EPOCHS	FLOPS
TRAINBP	BACKPROPAGATION	259.1	4123	21603435
TRAINBPX	FASTER BACKPROP	42.4	570	2971696
TRAINLM	LEVENBERG-MARQUARDT	3.3	5	315769
SOLVERB	RADIAL BASIS FUNCTION	1.9	5	38305

ดังนั้นโปรแกรมแม็ทแลปเป็นโปรแกรมที่ใช้งานง่ายและสะดวกในการเขียนโปรแกรมเนื่องจากมีฟังก์ชันเครื่องมือในการสร้างงานนิวรัลไว้ให้ใช้มากมาย โปรแกรมแม็ทแลบจึงเหมาะสมที่จะนำมาใช้ในการเขียนโปรแกรมในงานวิจัยนี้ ซึ่งงานวิจัยนี้มีการเขียนโปรแกรมข่ายงานนิวรัลแบบป้อนไปข้างหน้าหลายชั้น (multi-layered feed forward network) ซึ่งใช้การเรียนรู้แบบการกระจายความผิดพลาดย้อนกลับ ชนิด Levenberg-Marquardt (trainlm) ซึ่งเป็นอัลกอริธึมที่สามารถเรียนรู้ได้รวดเร็วมาก

ภาคผนวก ข.
ตัวอย่างโปรแกรม

โปรแกรมที่เขียนสำหรับงานวิจัยนี้คือ โปรแกรมสำหรับสร้างแบบจำลองข่ายงานนิวรัลและโปรแกรมในการทดสอบกับข้อมูลจริง โดยการนำชุดข้อมูลอินพุท และเป้าหมายรวมทั้งหมด 3 ชุดในการหาแบบจำลองที่เหมาะสมจะใช้ชุดข้อมูลชุดที่ 1 สำหรับในการฝึกข่ายงานและชุดข้อมูลชุดที่ 2 สำหรับการทดสอบข่ายงาน และชุดสุดท้ายสำหรับชุดทดสอบในการใช้งานจริง

โปรแกรมสำหรับสร้างแบบจำลองของกระบวนการจะมีรายละเอียดดังต่อไปนี้

1. ส่วนของการติดต่อกับผู้ใช้งาน (user interface)

เป็นส่วนของการติดต่อกันระหว่างผู้ใช้งานกับโปรแกรมโดยมีการกำหนดตัวแปรโกลบอล (global variable) ซึ่งเป็นตัวแปรที่ใช้ตลอดทั้งโปรแกรม ซึ่งในการเขียนโปรแกรมจะขึ้นคำว่า global ตามด้วยชื่อตัวแปร

2. ส่วนของฟังก์ชันในการเรียนรู้ข่ายงานนิวรัล (Levenberg - Marquardt)

ฟังก์ชันในการเรียนรู้ข่ายงานนิวรัลได้กล่าวถึงดังในรายละเอียดของภาคผนวก ก. ซึ่งในหัวข้อนี้จะกล่าวถึงการนำฟังก์ชันไปใช้งานในการเรียนรู้

2.1 ส่วนของการกำหนดชุดข้อมูลในการสร้างข่ายงานและทดสอบข่ายงานโดยมีทั้งหมด 3 ชุด

ชุดที่ 1 ชุดข้อมูลในการฝึกข่ายงานนิวรัล (training set) ประกอบด้วยชุดข้อมูลอินพุทและเป้าหมายในการสร้างแบบจำลองของข่ายนิวรัล

ชุดที่ 2 ชุดข้อมูลในการฝึกทดสอบข่ายงานนิวรัล (validation set) ประกอบด้วยชุดข้อมูลอินพุทและเป้าหมาย ซึ่งจะไม่มีการปรับค่าน้ำหนักและค่าไบอัสของข่ายงาน ซึ่งจะใช้ค่าความผิดพลาดระหว่างค่าเอาท์พุทและค่าเป้าหมายเป็นตัวแปรในการพิจารณาว่าแบบจำลองพอใจหรือไม่ในการนำไปใช้งานจริง

ชุดที่ 3 ชุดข้อมูลในการทดสอบการใช้งานจริง (test set) ประกอบด้วยชุดข้อมูลอินพุท หรือมีชุดข้อมูลเป้าหมาย ซึ่งจะมีการปรับค่าน้ำหนักเมื่อมีข้อมูลเป้าหมายเข้ามาคู่กับค่าอินพุทโดยปรับตามความผิดพลาดระหว่างค่าเอาท์พุทและค่าเป้าหมาย

2.2 การเริ่มต้นในการเรียนรู้

การเริ่มต้นในการเรียนรู้จะมีการกำหนด

- ค่าความผิดพลาด SSE (sum square error) = 0

- จำนวนชั้นในข่ายงานนิวรัลโดยในโปรแกรมนี้กำหนดให้จำนวนชั้นมีมากที่สุด 3 ชั้น ได้แก่ ชั้นซ่อน 1 ชั้นหรือ 2 ชั้น และชั้นเอาท์พุท 1 ชั้น

- การกำหนดจำนวนอินพุทและเอาท์พุทที่ใช้ในการฝึกข่ายงาน และกำหนดจำนวนนิวรัลในแต่ละชั้นของชั้นซ่อน

ข้อมูลชุดอินพุทจะมีการถ่ายข้อมูลลงในตัวแปร Xin ส่วนข้อมูลเป้าหมายจะถ่ายลงในตัวแปร Xout หลังจากทำการลดขนาดช่วงข้อมูลให้อยู่ในช่วง 0 – 1 แล้วส่งข้อมูลอินพุทและเป้าหมายที่ทำการลดขนาดช่วงข้อมูลไปคำนวณผลรวมของค่าอินพุทตามฟังก์ชันมูลฐานแล้วผ่านฟังก์ชันกระตุ้นในการหาสัญญาณเอาท์พุทของชั้นอินพุทและส่งค่าเอาท์พุทไปเป็นอินพุทให้กับชั้นซ่อน ในชั้นซ่อนจะกระทำในวิธีเดียวกันจนถึงชั้นเอาท์พุทซึ่งจะได้ค่าผลลัพธ์ของข่ายงาน

2.3 การคำนวณค่าความผิดพลาด SSE Error

เป็นการเปรียบเทียบค่าเป้าหมายกับค่าเอาท์พุทของข่ายงาน ซึ่งเก็บไว้ในตัวแปร SSE (ค่าความผิดพลาดของชุดข้อมูลฝึกข่ายงาน) , SSEtest (ค่าความผิดพลาดในการทดสอบข่ายงาน (validation))

2.4 การปรับค่าน้ำหนักและค่าไบอัส

การเริ่มปรับค่าน้ำหนักในชั้นจะเริ่มจากชั้นเอาท์พุทแล้วย้อนกลับมาสู่ชั้นฮิดเดนถัดลงมาเรื่อยๆ โดยปรับค่าน้ำหนัก W1,W2, W3,b1,b2,b3 (ถ้ามีจำนวนชั้น 2 ชั้น จะปรับค่า W1,W2,b1,b2 และถ้ามีจำนวนชั้น 3 ชั้น จะปรับค่า W1,W2,W3,b1.b2,b3) ซึ่งการปรับค่าน้ำหนักจะขึ้นอยู่กับค่าความแตกต่างของค่าเป้าหมายกับค่าเอาท์พุท

2.5 การสิ้นสุดการเรียนรู้

การเรียนรู้ถูกกำหนดให้สิ้นสุดเมื่อ

2.5.1 จำนวนรอบที่ต้องการให้เรียนรู้ (Iteration) ถึงตามที่กำหนด

2.5.2 ค่าความผิดพลาดต่ำสุด (SSE goal) ถึงที่กำหนดไว้

2.5.3 ค่าเกรเดียนต่ำสุด (Minimum gradient) ถึงที่กำหนดไว้

2.5.4 ค่าโมเมนตัมสูงสุด (Maximum momentum) ถึงที่กำหนดไว้

2.6 การแสดงประสิทธิภาพของข่ายงาน

การแสดงประสิทธิภาพของข่ายงานจะแสดงเป็น

- กราฟระหว่างค่า SSE Error ที่จำนวนรอบต่างๆ ของชุดฝึกข่ายงาน (training set)
- กราฟระหว่างค่า SSE Error ที่จำนวนรอบต่างๆ ของชุดข้อมูลทดสอบข่ายงาน (validation set)
- กราฟระหว่างค่า SSE Error ที่จำนวนรอบต่างๆ ของชุดข้อมูลที่ใช้งานจริง (test set)
- กราฟระหว่างค่าเป้าหมายกับค่าเอาท์พุทที่ได้จากข่ายงาน(ชุดข้อมูลการฝึกข่ายงาน , training set)
- กราฟระหว่างค่าเป้าหมายกับค่าเอาท์พุทที่ได้จากข่ายงาน (ชุดข้อมูลที่ใช้ทดสอบข่ายงาน , validation set) ที่จำนวนรอบถึงค่าที่กำหนดไว้
- กราฟระหว่างค่าเป้าหมายกับค่าเอาท์พุทที่ได้จากข่ายงาน (ชุดข้อมูลที่ใช้ทดสอบข่ายงาน , validation set) ที่ให้ค่าความผิดพลาดต่ำสุดระหว่างชุดข้อมูลเป้าหมายกับค่าเอาท์พุท
- กราฟระหว่างค่าเป้าหมายกับค่าเอาท์พุทที่ได้จากข่ายงาน (ชุดข้อมูลที่ใช้ทดสอบกับการใช้งานจริง , test set)

ตัวอย่างโปรแกรมที่ 1 โปรแกรมสำหรับปฏิสัมพันธ์กับผู้ใช้งานสำหรับการทำนายค่าอัตราการไหลของพอลิเมอร์ในเครื่องปฏิกรณ์

```
function NN_setup1(action);
%%%% NN_setup1 is graphic user interface for create neural network configuration%%%%
%%%% define global variable%%%%
global fig1
global SSEtest  % sum square error of test set
global hout     % output
global SSE      % sum square error of training set
global hout2    % output of test set
```

```
global W1       %  weight at layer 1
global b1       % bias at layer 1
global W2       % weight at layer 2
global b2       % bias at layer 2
global W3       % weight at layer 3
global b3       % bias at layer 3
global dw1      % diff weight at layer 1
global db1      % diff bias at layer 1
global dw2      % diff weight at layer 2
global db2      % diff weight at layer 2
global dw3      % diff weight at layer 3
global db3      % diff bias at layer 3
global Xout     % target of Training set
global Xout2    % target of Test set
global ep       % epoach
global tr       % tr
global TG       % target vector
global PD       % output vector
global Nwname   % Network name
global Nmlayer  % Number  of  layer
global Ndinp    % Number of input vector
global NdHd1    % Number of neuron at hidden layer 1
global NdHd2    % Number of neuron at hidden layer 2
global Ndout    % Number of neuron at output layer
global ActHd1   % Activation function at hidden layer 1
global ActHd2   % Activation function at hidden layer 2
global ActOP    % Activation function at output layer
global Mingrad  % Minimum gradient
global Momentum  % Initial momentum
```

```
global IPFile          % Input file
global IPPath          % Input path
global OPFile          % Output file
global OPPath          % Output path
global Epoupd          % Epoach update
global Testset         % Number of Test set
global Netwname        % Network name
global RMSMin          % Minimum sum square error
global Itemin          % Iteration at minimum sum square error
global w1min           % w1 at minimum SSE
global w2min           % w2 at minimum SSE
global w3min           % w3 at minimum SSE
global b1min           % b1 at minimum SSE
global b2min           % b2 at minimum SSE
global b3min           % b3 at minimum SSE
if nargin < 1 ,
  action ='initialise';
end;
if strcmp(action,'initialise'),
  close('Select Network');
  clear all
    file='';
    path='.';
%%%%initialize variable%%%%
Nwname ='No Name Network';
Nminp  ='No Data';
Nmlayer= '3';
ndhd1  ='No Hidden Node';
NdHd2  ='No Hidden Node';
```

```
Ndout  ='No Output Node';
Ite    = 100;
Momentum = 0.001;
IncMom =10;
DecMom =0.1;
MaxMom =1e10;
SSETg  =0.02;
Mingrad =0.0001;
Epoupd =25;
%%%%Configuration windows user interface%%%%
FIGURENAME = 'NN setup parameter';
FIGURETAG = 'NN_setup1';
FIGURESIZE = [10 10 780 500];
a = figure('Color',[0.8 0.8 0.8], ...
  'PaperType','a4letter', ...
  'Name', FIGURENAME, ...
  'NumberTitle', 'off', ...
  'Tag', FIGURETAG, ...
  'Position', FIGURESIZE, ...
  'Resize', 'on');
fig1 = a;
set(a, 'HandleVisibility', 'callback');
pos_l=2;
pos_w=FIGURESIZE(3)-4;
pos_h=FIGURESIZE(4)-4;
pos_t=2;
h_f_adv_background = uicontrol('Parent',a, ...
  'BackgroundColor',[0.701961 0.701961 0.701961], ...
  'Position',[pos_l pos_t pos_w pos_h], ...
```

```
  'Style','frame', ...
  'Tag','f_adv_background');
pos_w=280;
pos_l=FIGURESIZE(3)-(pos_w+2);
pos_h=FIGURESIZE(4)-2*4;
pos_t=4;
h_f_adv_side = uicontrol('Parent',a, ...
  'BackgroundColor',[0.701961 0.701961 0.701961], ...
  'Position',[pos_l pos_t pos_w pos_h], ...
  'Style','frame', ...
  'Tag','f_adv_side');
pos_l=4;
pos_w=FIGURESIZE(3)-8-pos_w-2;
pos_h=FIGURESIZE(4)-8;
pos_t=4;
h_f_advopt = uicontrol('Parent',a, ...
  'BackgroundColor',[0.701961 0.701961 0.701961], ...
  'Position',[pos_l pos_t pos_w pos_h], ...
  'Style','frame', ...
  'Tag','f_advopt');
bgc = get(h_f_advopt, 'BackgroundColor');
pos_frame=get(h_f_advopt, 'Position');
% Create static text for Load Input Data File and path (button)%
pos_h = 30;
pos_l = pos_frame(1) +4;
pos_w = 150;
pos_t = 450;
IPFilename = uicontrol('Parent',a, ...
  'Style','push',...
```

```
    'BackgroundColor',bgc, ...
    'Callback','NN_setup1 Lo_input',...
    'Position',[pos_l pos_t pos_w pos_h], ...
    'String','Load Data Input File',...
    'Visible','on');

% Create Static Text for Filename
pos_l = pos_frame(1) +260;
shwIPFile=uicontrol('Parent',a,...
    'Style','Text',...
    'BackgroundColor',bgc,...
    'Position',[pos_l+40 pos_t 100 15], ...
    'HorizontalAlignment','Left',...
    'String','Input File', ...
    'Tag','Tag_shwIPFile',...
    'Visible','on');
InpPath = uicontrol('Parent',a,...
    'Style','Text',...
    'BackgroundColor',bgc,...
    'Position',[pos_l-80 pos_t 120 15], ...
    'HorizontalAlignment','Right',...
    'String','Please Load Data ', ...
    'Tag','Tag_Inppath',...
    'Visible','on');
% Buttons Load Output Data File
pos_t = pos_t-50;
pos_l = pos_frame(1) +4;
OPFilename = uicontrol('Parent',a, ...
    'Style','push',...
```

```
  'BackgroundColor',bgc, ...
  'Callback','NN_setup1 Lo_output',...
  'Position',[pos_l pos_t pos_w pos_h], ...
  'String','Load Data Output File',...
  'Visible','on');
pos_l = pos_frame(1) +260;
shwOPFile = uicontrol('Parent',a,...
  'Style','Text',...
  'BackgroundColor',bgc,...
  'Position',[pos_l+40 pos_t 100 15], ...
  'HorizontalAlignment','Left',...
  'String','Output File', ...
  'Tag','Tag_shwOPFile',...
  'Visible','on');
OutPath = uicontrol('Parent',a,...
  'Style','Text',...
  'BackgroundColor',bgc,...
  'Position',[pos_l-80 pos_t 120 15], ...
  'HorizontalAlignment','Right',...
  'String','Please Load Data ', ...
  'Tag','Tag_Outpath',...
  'Visible','on');
% Create Network name edit box
pos_w1= 250;
pos_w2= 120;
pos_w3= 70;
pos_h = 20;
pos_t =pos_t-40;
pos_l = pos_frame(1) + 6;
```

```
b = uicontrol('Parent',a, ...
  'BackgroundColor',bgc, ...
  'HorizontalAlignment','left', ...
  'Position',[pos_l pos_t pos_w1 pos_h], ...
  'String','NetWork Name:', ...
  'Style','text'); ...

pos_l = pos_frame(1) +184;
uicontrol('Parent',a, ...
  'BackgroundColor',[1 1 1], ...
  'HorizontalAlignment','right', ...
  'Position',[pos_l pos_t pos_w2 pos_h], ...
  'Style','edit',...
  'String','No Network',...
  'UserData',[char(Nwname)],...
  'Tag','Tag_Nwname')
%Create Number of Layer edit box
pos_t = pos_t-20;
pos_l = pos_frame(1)+ 6;
b = uicontrol('Parent',a, ...
  'BackgroundColor',bgc, ...
  'HorizontalAlignment','left', ...
  'Position',[pos_l pos_t pos_w1 pos_h], ...
  'String','Number of Layer:', ...
  'Style','text');

pos_l = pos_frame(1) +184;
uicontrol('Parent',a, ...
  'BackgroundColor',[1 1 1], ...
```

```
  'HorizontalAlignment','right', ...
  'Position',[pos_l pos_t pos_w2 pos_h], ...
  'Style','edit',...
  'Enable','on',...
  'Callback','NN_setup1 CallNmlayer',...
  'String','3',...
  'Tag','Tag_Nmlayer');

%Creat Number of input node edit box
pos_t = pos_t-20;
pos_l = pos_frame(1)+ 6;
b = uicontrol('Parent',a, ...
  'BackgroundColor',bgc, ...
  'HorizontalAlignment','left', ...
  'Position',[pos_l pos_t pos_w1 pos_h], ...
  'String','Number of Input Node:', ...
  'Style','text');

pos_l = pos_frame(1) +184;
uicontrol('Parent',a, ...
  'BackgroundColor',[1 1 1], ...
  'HorizontalAlignment','right', ...
  'Position',[pos_l pos_t pos_w2 pos_h], ...
  'Style','edit',...
  'Enable','on',...
  'Tag','Tag_Nminp');

%Create Number node of hidden layer 1 edit box
pos_t = pos_t-20;
```

```
pos_l = pos_frame(1)+ 6;
b = uicontrol('Parent',a, ...
  'BackgroundColor',bgc, ...
  'HorizontalAlignment','left', ...
  'Position',[pos_l pos_t pos_w1 pos_h], ...
  'String','Node of Hidden Layer 1 :', ...
  'Style','text'); ...
 pos_l = pos_frame(1) +184;
NdHd1E=uicontrol('Parent',a, ...
   'BackgroundColor',[1 1 1], ...
   'HorizontalAlignment','right', ...
   'Position',[pos_l pos_t pos_w2 pos_h],
   'Style','edit',...
   'String','5',...
   'enable','on',...
   'Tag','Tag_NdHd1');
pos_l = pos_frame(1)+ 330;
b = uicontrol('Parent',a, ...
  'BackgroundColor',bgc, ...
  'HorizontalAlignment','left', ...
  'Position',[pos_l pos_t pos_w3 pos_h], ...
  'String','Activation Fn:', ...
  'Style','text'); ...
 pos_l = pos_frame(1) +400;
ActHd1E=uicontrol('Parent',a, ...
   'BackgroundColor',[1 1 1], ...
   'HorizontalAlignment','right', ...
   'Position',[pos_l pos_t pos_w3 pos_h], ...
   'Style','popup',...
```

```
    'String','Linear|Sigmoid|tanh',...
    'enable','on',...
     'Tag','Tag_ActHd1');
%Create Number node of hidden layer2 edit box
pos_t = pos_t-20;
pos_l = pos_frame(1)+ 6;
b = uicontrol('Parent',a, ...
   'BackgroundColor',bgc, ...
   'HorizontalAlignment','left', ...
   'Position',[pos_l pos_t pos_w1 pos_h], ...
   'String','Node of Hidden Layer 2 :', ...
   'Style','text'); ...
pos_l = pos_frame(1) +184;
NdHd2E=uicontrol('Parent',a, ...
    'BackgroundColor',[1 1 1], ...
    'HorizontalAlignment','right', ...
    'Position',[pos_l pos_t pos_w2 pos_h], ...
    'Style','edit',...
    'enable','off',...
    'Tag','Tag_NdHd2');
pos_l = pos_frame(1)+ 330;
b = uicontrol('Parent',a, ...
   'BackgroundColor',bgc, ...
   'HorizontalAlignment','left', ...
   'Position',[pos_l pos_t pos_w3 pos_h],
   'String','Activation Fn:', ...
   'Style','text'); ...
pos_l = pos_frame(1) +400;
ActHd2E=uicontrol('Parent',a, ...
```

```
    'BackgroundColor',[1 1 1], ...
    'HorizontalAlignment','right', ...
    'Position',[pos_l pos_t pos_w3 pos_h],...
    'Style','popup',...
    'String','Linear|Sigmoid|tanh',...
    'enable','off',...
    'Tag','Tag_ActHd2');

% Create Number node of Output Layer edit box
pos_t = pos_t-20;
pos_l = pos_frame(1)+ 6;
b = uicontrol('Parent',a, ...
   'BackgroundColor',bgc, ...
   'HorizontalAlignment','left', ...
   'Position',[pos_l pos_t pos_w1 pos_h], ...
   'String','Node of Output Layer :', ...
   'Style','text'); ...
pos_l = pos_frame(1) +184;
uicontrol('Parent',a, ...
    'BackgroundColor',[1 1 1], ...
    'HorizontalAlignment','right', ...
    'Position',[pos_l pos_t pos_w2 pos_h],
    'Style','edit',...
    'String','1',...
    'Tag','Tag_Ndout');
pos_l = pos_frame(1)+ 330;
b = uicontrol('Parent',a, ...
   'BackgroundColor',bgc, ...
   'HorizontalAlignment','left', ...
```

```
    'Position',[pos_l pos_t pos_w3 pos_h], ...
    'String','Activation Fn:', ...
    'Style','text');
pos_l = pos_frame(1) +400;
uicontrol('Parent',a, ...
    'BackgroundColor',[1 1 1], ...
    'HorizontalAlignment','right', ...
    'Position',[pos_l pos_t pos_w3 pos_h], ...
    'Style','popup',...
    'String','Linear|Sigmoid|tanh',...
    'Tag','Tag_ActOP');


%Create maximum iteration edit box
pos_t = pos_t-20;
pos_l = pos_frame(1)+ 6;
b = uicontrol('Parent',a, ...
    'BackgroundColor',bgc, ...
    'HorizontalAlignment','left', ...
    'Position',[pos_l pos_t pos_w1 pos_h], ...
    'String','Maximum Number of Iteration:', ...
    'Style','text'); ...
pos_l = pos_frame(1) +184;
uicontrol('Parent',a, ...
    'BackgroundColor',[1 1 1], ...
    'HorizontalAlignment','right', ...
    'Position',[pos_l pos_t pos_w2 pos_h], ...
    'Style','edit',...
    'String','1000',...
```

```
  'Tag','Tag_ite');

%Create Momentum edit box
pos_t = pos_t-20;
pos_l = pos_frame(1)+ 6;
b = uicontrol('Parent',a, ...
  'BackgroundColor',bgc, ...
  'HorizontalAlignment','left', ...
  'Position',[pos_l pos_t pos_w1 pos_h], ...
  'String','Momentum:', ...
  'Style','text'); ...
pos_l = pos_frame(1) +184;
uicontrol('Parent',a, ...
  'BackgroundColor',[1 1 1], ...
  'HorizontalAlignment','right', ...
  'Position',[pos_l pos_t pos_w2 pos_h], ...
  'Style','edit',...
  'String','0.85',...
  'Tag','Tag_Momentum');

% Creat Multiplier for increasing momentum edit box
pos_t = pos_t-20;
pos_l = pos_frame(1)+ 6;
b = uicontrol('Parent',a, ...
  'BackgroundColor',bgc, ...
  'HorizontalAlignment','left', ...
  'Position',[pos_l pos_t pos_w1 pos_h], ...
  'String','Multiplier for increasing Momentum:', ...
  'Style','text'); ...
```

```
pos_l = pos_frame(1) +184;
uicontrol('Parent',a, ...
  'BackgroundColor',[1 1 1], ...
  'HorizontalAlignment','right', ...
  'Position',[pos_l pos_t pos_w2 pos_h], ...
  'Style','edit',...
  'String','10',...
  'Tag','Tag_IncMom');

% Create Multiplier for decreasing momentum edit box
pos_t = pos_t-20;
pos_l = pos_frame(1)+ 6;
b = uicontrol('Parent',a, ...
  'BackgroundColor',bgc, ...
  'HorizontalAlignment','left', ...
  'Position',[pos_l pos_t pos_w1 pos_h], ...
  'String','Multiplier for Decreasing Momentum:', ...
  'Style','text'); ...
pos_l = pos_frame(1) +184;
uicontrol('Parent',a, ...
  'BackgroundColor',[1 1 1], ...
  'HorizontalAlignment','right', ...
  'Position',[pos_l pos_t pos_w2 pos_h], ...
  'Style','edit',...
  'String','0.1',...
  'Tag','Tag_DecMom');

% Create Maximum momentum edit box
pos_t = pos_t-20;
```

```
pos_l = pos_frame(1)+ 6;
b = uicontrol('Parent',a, ...
  'BackgroundColor',bgc, ...
  'HorizontalAlignment','left', ...
  'Position',[pos_l pos_t pos_w1 pos_h], ...
  'String','Maximum Momentum:', ...
  'Style','text'); ...
pos_l = pos_frame(1) +184;
uicontrol('Parent',a, ...
  'BackgroundColor',[1 1 1], ...
  'HorizontalAlignment','right', ...
  'Position',[pos_l pos_t pos_w2 pos_h], ...
  'Style','edit',...
  'String','1e20',...
  'Tag','Tag_MaxMom');

% Create Sum squared error goal edit box
pos_t = pos_t-20;
pos_l = pos_frame(1)+ 6;
b = uicontrol('Parent',a, ...
  'BackgroundColor',bgc, ...
  'HorizontalAlignment','left', ...
  'Position',[pos_l pos_t pos_w1 pos_h], ...
  'String','Sum Squared Error Goal:', ...
  'Style','text'); ...
pos_l = pos_frame(1) +184;
uicontrol('Parent',a, ...
  'BackgroundColor',[1 1 1], ...
  'HorizontalAlignment','right', ...
```

```
  'Position',[pos_l pos_t pos_w2 pos_h], ...
  'Style','edit',...
  'String','1e-20',...
  'Tag','Tag_SSETg');

% Create Minimum gradient edit box
pos_t = pos_t-20;
pos_l = pos_frame(1)+ 6;
b = uicontrol('Parent',a, ...
  'BackgroundColor',bgc, ...
  'HorizontalAlignment','left', ...
  'Position',[pos_l pos_t pos_w1 pos_h], ...
  'String','Minimum Gradient:', ...
  'Style','text'); ...
pos_l = pos_frame(1) +184;
uicontrol('Parent',a, ...
  'BackgroundColor',[1 1 1], ...
  'HorizontalAlignment','right', ...
  'Position',[pos_l pos_t pos_w2 pos_h], ...
  'Style','edit',...
  'String','1e-20',...
  'Tag','Tag_Mingrad');

% Create epoach between updating display edit box
pos_t = pos_t-20;
pos_l = pos_frame(1)+ 6;
b = uicontrol('Parent',a, ...
  'BackgroundColor',bgc, ...
  'HorizontalAlignment','left', ...
```

```
  'Position',[pos_l pos_t pos_w1 pos_h], ...
  'String','Epoach between Updating Display:', ...
  'Style','text'); ...
pos_l = pos_frame(1) +184;
uicontrol('Parent',a, ...
  'BackgroundColor',[1 1 1], ...
  'HorizontalAlignment','right', ...
  'Position',[pos_l pos_t pos_w2 pos_h], ...
  'Style','edit',...
  'String','1',...
  'Tag','Tag_Epoupd');

% Creat test set edit box
pos_t = pos_t-20;
pos_l = pos_frame(1)+ 6;
b = uicontrol('Parent',a, ...
  'BackgroundColor',bgc, ...
  'HorizontalAlignment','left', ...
  'Position',[pos_l pos_t pos_w1 pos_h], ...
  'String','Test Set Cases:', ...
  'Style','text'); ...

pos_l = pos_frame(1) +184;
uicontrol('Parent',a, ...
  'BackgroundColor',[1 1 1], ...
  'HorizontalAlignment','right', ...
  'Position',[pos_l pos_t pos_w2 pos_h], ...
  'Style','edit',...
  'String','0',...
```

```
  'Tag','Tag_Testset');
pos_l=40;
pos_w=70;
pos_h=30;
pos_t=pos_t-40;

% Create Save,Cancel, Help Button
pos_l=20;
uicontrol('Parent',a, ...
  'BackgroundColor',[0.701961 0.701961 0.701961], ...
  'Callback','NN_setup1', ...
  'Interruptible', 'off', ...
  'Position',[pos_l pos_t pos_w pos_h], ...
  'String','Run Training', ...
  'Tag','RunTrn',...
  'Callback','NN_setup1 Runtrain');

pos_l=120;
Save_but = uicontrol('Parent',a, ...
  'BackgroundColor',[0.701961 0.701961 0.701961], ...
  'Callback','NN_setup1 Save', ...
  'Interruptible', 'off', ...
  'Position',[pos_l pos_t pos_w pos_h], ...
  'String','SAVE', ...
  'Tag','Save_but');

pos_l=220;
Can_but= uicontrol('Parent',a, ...
  'BackgroundColor',[0.701961 0.701961 0.701961], ...
```

```
  'Callback','Sel_network', ...
  'Interruptible', 'off', ...
  'Position',[pos_l pos_t pos_w pos_h], ...
  'String','Cancel', ...
  'Tag','Can_but');

pos_l = 320;
help_but= uicontrol('Parent',a, ...
  'BackgroundColor',[0.701961 0.701961 0.701961], ...
  'Callback','gui_cb LoadData', ...
  'Interruptible', 'off', ...
  'Position',[pos_l pos_t pos_w pos_h], ...
  'String','Help', ...
  'Tag','help_but');

% Update Data Run Neural Button
pos_t =450;
uicontrol('Parent',a, ...
  'BackgroundColor',bgc, ...
  'ForegroundColor',[1,0,0], ...
  'HorizontalAlignment','left', ...
  'Fontsize',13,...
  'Position',[570 pos_t 200 30], ...
  'Style','text',...
  'String','TRAINING RESULT');
pos_w1 =150;
pos_w2 = 120;
pos_h =20;
pos_l = 510;
```

```
h = 40;
w = 125;
pos_t = pos_t-h;
uicontrol('Parent',a, ...
  'BackgroundColor',bgc, ...
  'HorizontalAlignment','left', ...
  'Position',[pos_l pos_t pos_w1 pos_h], ...
  'String','RMS Error(Trainning Set):', ...
  'Style','text');
pos_t = pos_t-h;
uicontrol('Parent',a, ...
  'BackgroundColor',bgc, ...
  'HorizontalAlignment','left', ...
  'Position',[pos_l pos_t pos_w1 pos_h], ...
  'String','RMS Error(Test Set):', ...
  'Style','text');
pos_t = pos_t-h;
uicontrol('Parent',a, ...
  'BackgroundColor',bgc, ...
  'HorizontalAlignment','left', ...
  'Position',[pos_l pos_t pos_w1 pos_h], ...
  'String','Iteraion:', ...
  'Style','text');
pos_t = pos_t-h;
uicontrol('Parent',a, ...
  'BackgroundColor',bgc, ...
  'HorizontalAlignment','left', ...
  'Position',[pos_l pos_t pos_w1 pos_h], ...
  'String','% Complete:', ...
```

```
  'Style','text');
pos_t = pos_t-h;
uicontrol('Parent',a, ...
  'BackgroundColor',bgc, ...
  'HorizontalAlignment','left', ...
  'Position',[pos_l pos_t pos_w1 pos_h], ...
  'String','Momentum:', ...
  'Style','text');
pos_t = pos_t-h;
uicontrol('Parent',a, ...
  'BackgroundColor',bgc, ...
  'HorizontalAlignment','left', ...
  'Position',[pos_l pos_t pos_w1 pos_h], ...
  'String','Iteration at Min RMS(Test)',...
  'Style','text');
pos_t = pos_t-h;
uicontrol('Parent',a, ...
  'BackgroundColor',bgc, ...
  'HorizontalAlignment','left', ...
  'Position',[pos_l pos_t pos_w1 pos_h], ...
  'String','Min RMS Error(Test)',...
  'Style','text');
pos_l =pos_l+w ;
pos_t =450-h
uicontrol('Parent',a, ...
  'BackgroundColor',[1 1 1], ...
  'HorizontalAlignment','right', ...
  'Position',[pos_l pos_t pos_w2 pos_h], ...
  'Style','edit',...
```

```
  'String','',...
  'Tag','Tag_RMStrn');
pos_t = pos_t-h;
uicontrol('Parent',a, ...
  'BackgroundColor',[1 1 1], ...
  'HorizontalAlignment','right', ...
  'Position',[pos_l pos_t pos_w2 pos_h], ...
  'Style','edit',...
  'String','',...
  'Tag','Tag_RMStest');
pos_t = pos_t-h;
uicontrol('Parent',a, ...
  'BackgroundColor',[1 1 1], ...
  'HorizontalAlignment','right', ...
  'Position',[pos_l pos_t pos_w2 pos_h], ...
  'Style','edit',...
  'String','',...
  'Tag','Tag_iterat');
pos_t = pos_t-h;
uicontrol('Parent',a, ...
  'BackgroundColor',[1 1 1], ...
  'HorizontalAlignment','right', ...
  'Position',[pos_l pos_t pos_w2 pos_h], ...
  'Style','edit',...
  'String','',...
  'Tag','Tag_complete');
pos_t = pos_t-h;
uicontrol('Parent',a, ...
  'BackgroundColor',[1 1 1], ...
```

```
  'HorizontalAlignment','right', ...
  'Position',[pos_l pos_t pos_w2 pos_h], ...
  'Style','edit',...
  'String','',...
  'Tag','Tag_mom');
pos_t = pos_t-h;
uicontrol('Parent',a, ...
  'BackgroundColor',[1 1 1], ...
  'HorizontalAlignment','right', ...
  'Position',[pos_l pos_t pos_w2 pos_h], ...
  'Style','edit',...
  'String','',...
  'Tag','Tag_IteMin');
pos_t = pos_t-h;
uicontrol('Parent',a, ...
  'BackgroundColor',[1 1 1], ...
  'HorizontalAlignment','right', ...
  'Position',[pos_l pos_t pos_w2 pos_h], ...
  'Style','edit',...
  'String','',...
  'Tag','Tag_MinRMS');
%%% create menu bar %%%
menu1=uimenu(a,'Label','Show Results...','Callback','');
uimenu(menu1,'Label','Network Information','Callback','NN_setup1 NWinform');
uimenu(menu1,'Label','Case Outputs/Targets(Train Set)','Callback','NN_setup1 OutTarTrn');
uimenu(menu1,'Label','Case Outputs/Targets(Test Set)','Callback','NN_setup1 OutTarTest');
uimenu(menu1,'Label','Case Outputs/Targets(Train Set)ax','Callback','NN_setup1 OutTarTrnax');
uimenu(menu1,'Label','Case Outputs/Targets(Test Set)ax','Callback','NN_setup1 OutTarTestax');
uimenu(menu1,'Label','Outputs/Targets (Train+Test)','Callback','NN_setup1 OutTarTot');
```

```
uimenu(menu1,'Label','RMS Error History vs Iteration(Trn)','Callback','NN_setup1 RMSError');
uimenu(menu1,'Label','RMS Error History vs Iteration(Test)','Callback','NN_setup1 RMStest');
uimenu(menu1,'Label','Node Weight and Deltas','Callback','NN_setup1 WgDelta');
uimenu(menu1,'Label','Scatter Comparison of Normalized Targets vs. Net
Outputs','Callback','NN_setup1 Cmpout');
uimenu(menu1,'Label','Outputs/Targets Information','Callback','NN_setup1 OutTarInfm');
uimenu(menu1,'Label','Outputs/Targets Min Error(Test)','Callback','NN_setup1 OutTarMinErr');


%save handles of important UI objects
hndlList =  [IPFilename OPFilename shwIPFile shwOPFile InpPath OutPath NdHd1E...
     NdHd2E ActHd1E ActHd2E];
set(a,'UserData',hndlList);
%Hide window from command line
set(a,'HandleVisibility','callback');

% create action for Object-Oriented program
elseif strcmp(action,'Lo_input'),
  hndlList=get(gcf,'UserData');
  [fileIP,pathIP]=uigetfile('*.TXT','Get Data File',50,50);
  if strcmp(fileIP,'') |fileIP==0,
     set(hndlList(3),'String','Nodata file loaded.');
  else
     set(hndlList(3),'String',fileIP);
     set(hndlList(5),'String',pathIP);
  end;
elseif strcmp(action,'Lo_output'),
  hndlList=get(gcf,'UserData');
```

```
    [fileOP,pathOP]=uigetfile('*.TXT','Get Data File',50,50);
    if strcmp(fileOP,'') |fileOP==0,
      set(hndlList(4),'String','Nodata file loaded.');
    else
      set(hndlList(4),'String',fileOP);
      set(hndlList(6),'String',pathOP);
    end;
elseif strcmp(action,'CallNmlayer'),
    hndlList=get(gcf,'UserData');
    a = findobj(0,'tag','NN_setup1');
    Nmlayer  = (get(findobj(a,'Tag','Tag_Nmlayer'),'String'));
     if strcmp(Nmlayer,'3'),
      set(hndlList(7),'Enable','on');
      set(hndlList(8),'Enable','off');
      set(hndlList(9),'Enable','on');
      set(hndlList(10),'Enable','off');
    elseif strcmp(Nmlayer,'4'),
      set(hndlList(7),'Enable','on');
      set(hndlList(8),'Enable','on');
      set(hndlList(9),'Enable','on');
      set(hndlList(10),'Enable','on');
    else
      errordlg('Invalid Number Layer(Number Hidden Layer <= 2 Layer');
    end;
elseif strcmp(action,'Runtrain')
    a = findobj(0,'tag','NN_setup1');
    Nwname = (get(findobj(a,'tag','Tag_Nwname'),'String'));
    aname=Nwname
    Nmlayer  = str2num(get(findobj(a,'Tag','Tag_Nmlayer'),'String'));
```

```
Ndinp  = str2num(get(findobj(a,'tag','Tag_Nminp'),'String'));
NdHd1  = str2num(get(findobj(a,'Tag','Tag_NdHd1'),'String'));
NdHd2  = str2num(get(findobj(a,'Tag','Tag_NdHd2'),'String'));
Ndout  = str2num(get(findobj(a,'Tag','Tag_Ndout'),'String'));
Ite    = str2num(get(findobj(a,'Tag','Tag_ite'),'String'));
Momentum= str2num(get(findobj(a,'Tag','Tag_Momentum'),'String'));
IncMom =  str2num(get(findobj(a,'Tag','Tag_IncMom'),'String'));
DecMom =  str2num(get(findobj(a,'Tag','Tag_DecMom'),'String'));
MaxMom =  str2num(get(findobj(a,'Tag','Tag_MaxMom'),'String'));
SSETg  =  str2num(get(findobj(a,'Tag','Tag_SSETg'),'String'));
Mingrad=  str2num(get(findobj(a,'Tag','Tag_Mingrad'),'String'));
Epoupd =  str2num(get(findobj(a,'Tag','Tag_Epoupd'),'String'));
Testset =  str2num(get(findobj(a,'Tag','Tag_Testset'),'String'));
ActHd1N = findobj(a,'Tag','Tag_ActHd1');
if get(ActHd1N,'Value')==1,
   ActHd1 ='purelin';
elseif get(ActHd1N,'Value')==2,
   ActHd1 ='logsig';
else
   ActHd1 = 'tansig';
end;

ActHd2N = findobj(a,'Tag','Tag_ActHd2');
if get(ActHd2N,'Value')==1,
   ActHd2 ='purelin';
elseif get(ActHd2N,'Value')==2,
   ActHd2 ='logsig';
else
   ActHd2 = 'tansig';
```

```
end;

ActOPN  = findobj(a,'Tag','Tag_ActOP');
if get(ActOPN,'Value')==1,
   ActOP ='purelin';
elseif get(ActOPN,'Value')==2,
   ActOP ='logsig';
else
   ActOP = 'tansig';
end;
IPFile= get(findobj(a,'Tag','Tag_shwIPFile'),'String');
IPPath= get(findobj(a,'Tag','Tag_Inppath'),'String');
if isempty(findstr(IPFile,'.txt' )),
   errordlg('Input File Format is not correct.');
else
   IPFile= strrep(IPFile,'.txt','');
end;
OPFile= get(findobj(a,'Tag','Tag_shwOPFile'),'String')
OPPath= get(findobj(a,'Tag','Tag_Outpath'),'String')
if isempty(findstr(OPFile,'.txt' )),
   OPFile=errordlg('Output File Format is not correct.');
else
   OPFile=strrep(OPFile,'.txt','');
end;
if NdHd1 >0 ,
   if Nmlayer==4,

[hout,SSE,hout2,SSEtest,W1,b1,W2,b2,W3,b3,Xout,Xout2,ep,tr,TG,PD,dw1,dw2,dw3,db1,db2,db3,...
        w1min,b1min,w2min,b2min,w3min,b3min]=trainNN1(IPFile,OPFile,IPPath,OPPath,...
```

```
            Ndinp,NdHd1,Ndout,ActHd1,ActOP,Ite,Momentum,IncMom,DecMom,...
            MaxMom,SSETg,Mingrad,Epoupd,Testset,NdHd2,ActHd2)
        elseif Nmlayer==3
          [hout,SSE,hout2,SSEtest,W1,b1,W2,b2,Xout,Xout2,ep,tr,TG,PD,dw1,dw2,db1,db2,...
            w1min,b1min,w2min,b2min]=trainNN1(IPFile,OPFile,IPPath,OPPath,...
             Ndinp,NdHd1,Ndout,ActHd1,ActOP,Ite,Momentum,IncMom,DecMom,...
            MaxMom,SSETg,Mingrad,Epoupd,Testset);
        end;
%           k=ep;
%           t=tr;

    else
       errordlg('Invalid Number Hidden layer')
    end;
elseif strcmp(action,'Save'),
    a = findobj(0,'tag','NN_setup1');
    Nwname = (get(findobj(a,'tag','Tag_Nwname'),'String'));
    aname=Nwname
    Nmlayer  = (get(findobj(a,'Tag','Tag_Nmlayer'),'String'));
    Ndinp  = (get(findobj(a,'tag','Tag_Nminp'),'String'));
    NdHd1  = (get(findobj(a,'Tag','Tag_NdHd1'),'String'));
    NdHd2  = (get(findobj(a,'Tag','Tag_NdHd2'),'String'));
    Ndout  = (get(findobj(a,'Tag','Tag_Ndout'),'String'));
    Ite    = (get(findobj(a,'Tag','Tag_ite'),'String'));
    Momentum=(get(findobj(a,'Tag','Tag_Momentum'),'String'));
    IncMom = (get(findobj(a,'Tag','Tag_IncMom'),'String'));
    DecMom = (get(findobj(a,'Tag','Tag_DecMom'),'String'));
    MaxMom = (get(findobj(a,'Tag','Tag_MaxMom'),'String'));
    SSETg  = (get(findobj(a,'Tag','Tag_SSETg'),'String'));
```

```
Mingrad= (get(findobj(a,'Tag','Tag_Mingrad'),'String'));
Epoupd = (get(findobj(a,'Tag','Tag_Epoupd'),'String'));
Testset =(get(findobj(a,'Tag','Tag_Testset'),'String'));
ActHd1N = get(findobj(a,'Tag','Tag_ActHd1'),'Value');
ActHd2N = get(findobj(a,'Tag','Tag_ActHd2'),'Value');
ActOPN  = get(findobj(a,'Tag','Tag_ActOP'),'Value');
IPFile= get(findobj(a,'Tag','Tag_shwIPFile'),'String');
OPFile= get(findobj(a,'Tag','Tag_shwOPFile'),'String');
IPPath= get(findobj(a,'Tag','Tag_Inppath'),'String');
OPPath= get(findobj(a,'Tag','Tag_Outpath'),'String');
Mom= get(findobj(a,'Tag','Tag_mom'),'String');
RMSMin=get(findobj(a,'Tag','Tag_MinRMS'),'String');
Itemin=get(findobj(a,'Tag','Tag_IteMin'),'String');
[Savefile,Savepath]=uiputfile('*.mat','Save As',50,50);
Savename=[Savepath Savefile]
if NdHd2>0
  save(Savename,'Nwname','Nmlayer','Ndinp','NdHd1','NdHd2','Ndout','Momentum',...
    'IncMom','DecMom','MaxMom','SSETg','Mingrad','Epoupd','Testset','ActHd1N','ActHd2N',...
'ActOPN','IPFile','OPFile','IPPath','OPPath','hout','SSE','hout2','SSEtest','W1','b1','W2','b2','W3','b
3',...
'dw1','dw2','dw3','db1','db2','db3','Xout','Xout2','ep','tr','TG','PD','Ite','RMSMin','Itemin','Mom',...
    'w1min','b1min','w2min','b2min','w3min','b3min')
else
  save(Savename,'Nwname','Nmlayer','Ndinp','NdHd1','NdHd2','Ndout','Momentum',...
    'IncMom','DecMom','MaxMom','SSETg','Mingrad','Epoupd','Testset','ActHd1N','ActHd2N',...
    'ActOPN','IPFile','OPFile','IPPath','OPPath','hout','SSE','hout2','SSEtest','W1','b1','W2','b2',...
    'dw1','dw2','db1','db2','Xout','Xout2','ep','tr','TG','PD','Ite','RMSMin','Itemin','Mom',...
```

```
    'w1min','b1min','w2min','b2min')

  end;

elseif strcmp(action,'OutTarTrn'),
    title1='Outputs/Target MFR1 Slurry vs Pattern number(Trainset)'
    title2='Outputs/Target MFR1 Slurry vs Pattern number(Trainset+Test Set)'
    ylabelmfr1='Output/Target MFR1(g/10 min)'
    legendmfr1='Output MFR1'
    legendmfr2='Target MFR1'
    figure(2)
    subplot(211)
    plot(1:length(hout),hout,'-',1:length(Xout),Xout,'--')
    xlabel('Pattern ')
    ylabel(ylabelmfr1)
    title(title1);
    legend(legendmfr1,legendmfr2);
    subplot(212)
    Out=[hout;hout2];
    Tar =[Xout;Xout2];
     plot(1:length(Out),Out,'-',1:length(Tar),Tar,'--')
     xlabel('Pattern ')
     ylabel(ylabelmfr1)
     title(title2);
     legend(legendmfr1,legendmfr2);
elseif strcmp(action,'OutTarTest'),
    title1='Outputs/Target MFR1 Slurry vs Pattern number(Test set)'
    title2='Outputs/Target MFR1 Slurry vs Pattern number(At Min RMS Error (Test Set))'
    ylabelmfr1='Output/Target MFR1(g/10 min)'
```

```
 legendmfr1='Output MFR1'
 legendmfr2='Target MFR1'
figure(2)
subplot(211)
if Testset==0
    errordlg('No Test Set Pattern')
else
    plot(1:length(hout2),hout2,'-',1:length(Xout2),Xout2,'--')
    xlabel('Pattern ')
    ylabel(ylabelmfr1)
    title(title1);
    legend(legendmfr1,legendmfr2);
 end;
 subplot(212)
 if Testset==0
    errordlg('No Test Set Pattern')
 else
     load c:\rmstest2.txt
     rmstest2=reshape(rmstest2,size(rmstest2,1)/2,2);
     i=1:1:size(rmstest2,1)
     Output=rmstest2(:,1);
         Target=rmstest2(:,2);
  clear rmstest2
  plot(i,Output,'-',i,Target,'--')
  xlabel('Pattern')
  ylabel(ylabelmfr1)
  title(title2);
  legend(legendmfr1,legendmfr2);
end;
```

```
elseif strcmp(action,'OutTarTrnax'),
    title1='Outputs/Target MFR1 Slurry vs Pattern number(Trainset)'
    title2='Outputs/Target MFR1 Slurry vs Pattern number(Trainset+Test Set)'
    ylabelmfr1='Output/Target MFR1(g/10 min)'
    legendmfr1='Output MFR1'
    legendmfr2='Target MFR1'
    figure(2)
    subplot(211)
    plot(1:length(hout),hout,'-',1:length(Xout),Xout,'--')
    xlabel('Pattern ')
    ylabel(ylabelmfr1)
    title(title1);
    legend(legendmfr1,legendmfr2);
    axis([1,length(hout),0,40]);
    subplot(212)
    Out=[hout;hout2];
    Tar =[Xout;Xout2];
    plot(1:length(Out),Out,'-',1:length(Tar),Tar,'--')
    xlabel('Pattern ')
    ylabel(ylabelmfr1)
    title(title2);
    legend(legendmfr1,legendmfr2);
    axis([1,length(Out),0,40]);
elseif strcmp(action,'OutTarTestax'),
    title1='Outputs/Target MFR1 Slurry vs Pattern number(Test set)'
    title2='Outputs/Target MFR1 Slurry vs Pattern number(At Min RMS Error (Test Set))'
    ylabelmfr1='Output/Target MFR1(g/10 min)'
    legendmfr1='Output MFR1'
    legendmfr2='Target MFR1'
```

```
figure(2)
subplot(211)
if Testset==0
    errordlg('No Test Set Pattern')
else
    plot(1:length(hout2),hout2,'-',1:length(Xout2),Xout2,'--')
    xlabel('Pattern ')
    ylabel(ylabelmfr1)
    title(title1);
    legend(legendmfr1,legendmfr2);
    axis([1,length(hout2),0,40]);
end;
subplot(212)
if Testset==0
    errordlg('No Test Set Pattern')
else
  load c:\rmstest2.txt
  rmstest2=reshape(rmstest2,size(rmstest2,1)/2,2);
  i=1:1:size(rmstest2,1)
  Output=rmstest2(:,1);
  Target=rmstest2(:,2);
  clear rmstest2
  plot(i,Output,'-',i,Target,'--')
  xlabel('Pattern')
  ylabel(ylabelmfr1)
  title(title2);
  legend(legendmfr1,legendmfr2);
  axis([1,length(Output),0,40]);
end;
```

```
elseif strcmp(action,'OutTarTot'),
  title1='Outputs/Target MFR2 Slurry vs Pattern number(Trainset)'
  title2='Outputs/Target MFR2 Slurry vs Pattern number(Trainset+Test Set)'
  ylabelmfr1='Output/Target MFR2(g/10 min)'
  legendmfr1='Output MFR2'
  legendmfr2='Target MFR2'
  Out=[hout;hout2];
  Tar =[Xout;Xout2];
  figure(2)
  plot(1:length(Out),Out,'-',1:length(Tar),Tar,'--')
  xlabel('Pattern ')
  ylabel(ylabelmfr1)
  title(title2);
  legend(legendmfr1,legendmfr2);
elseif strcmp(action,'OutTarMinErr'),
   if Testset==0
      errordlg('No Test Set Pattern')
   else
      load c:\rmstest2.txt
      rmstest2=reshape(rmstest2,size(rmstest2,1)/2,2);
      i=1:1:size(rmstest2,1)
      Output=rmstest2(:,1);
      Target=rmstest2(:,2);
      clear rmstest2
      figure(2)
      plot(i,Output,'-',i,Target,'--')
         xlabel('Pattern')
      ylabel('Outputs/Target')
        title('Outputs/Target vs Pattern number(At Min RMS Error (Test Set))');
```

```
    legend('Output','Target');
  end;
elseif strcmp(action,'RMSError'),
  load c:\rmserr1.txt
  [C,D]=size(rmserr1)
  i=rmserr1(:,1)
  SSE=rmserr1(:,2)
  clear C D
  clear rmserr1
  load c:\rmstest1.txt
  i=rmstest1(:,1)
  SSEtest=rmstest1(:,2)
  clear rmstest1
  figure(2)
  subplot(211)
  plot(i,SSE,'-')
  xlabel('Iteration ')
  ylabel('RMS Error')
  title('Root Mean Square of Train Set');
  subplot(2,1,2)
  if Testset==0
     errordlg('No Test Set Pattern')
  else
     plot(i,SSEtest,'-')
     xlabel('Iteration ')
     ylabel('RMS Error')
     title('Root Mean Square of Test Set');
  end;
elseif strcmp(action,'RMStest'),
```

```
  if Testset==0
     errordlg('No Test Set Pattern')
  else
     load c:\rmstest1.txt
     i=rmstest1(:,1)
        SSE=rmstest1(:,2)
     clear rmstest1
     figure(2)
     plot(i,SSE,'-')
     xlabel('Iteration ')
     ylabel('RMS Error')
     title('Root Mean Square of Test Set');
  end;
elseif strcmp(action,'WgDelta'),
  figurePos=[20 30 400 400];
  figNumber=figure('NumberTitle','off','name','Network Weight and Current Adjustment
Deltas',...
    'color',[.8 .8 .8],'Position',figurePos);
  currfig=figNumber;
  category='Network Weight and Current Adjustment Deltas';
  ttlStr=' ';
  Wg=[];dWg=[];wg1=[];wg2=[];wg3=[];dwg1=[];dwg2=[];dwg3=[];
  wg1 =reshape(W1,size(W1,1)*size(W1,2),1)
  wg2 =reshape(W2,size(W2,1)*size(W2,2),1)
  dwg1 =reshape(dw1,size(dw1,1)*size(dw1,2),1)
  dwg2 =reshape(dw2,size(dw2,1)*size(dw2,2),1)
  if NdHd2> 0
     wg3 =reshape(W3,size(W3,1)*size(W3,2),1)
     dwg3 =reshape(dw3,size(dw3,1)*size(dw3,2),1)
```

```
end
Wg = [wg1;b1;wg2;b2;wg3;b3];
dWg = [dwg1;db1;dwg2;db2;dwg3;db3];
Wgmat=[];
 if Nmlayer==3
  NdHd2=0
end;
for i =1 : (Ndinp+1)
  for j =1:NdHd1
    Wgmat=[Wgmat;1   i   j];
  end
end;
if NdHd2>0
 for i = 1:(NdHd1+1)
     for j=1:NdHd2
        Wgmat=[Wgmat;2   i   j];
      end
 end
 for i = 1:(NdHd2+1)
     for j=1:(Ndout)
         Wgmat=[Wgmat;3   i   j];
     end
 end
 else
   for i = 1:(NdHd1+1)
       for j=1:(Ndout)
           Wgmat=[Wgmat;2   i   j];
        end
     end
```

```
    end
    [a,b]=size(Wg)
    [c,d]=size(dWg)
    [e,f]=size(Wgmat)
    blank=repmat(blanks(10),size(Wg,1),1)
    Wgformat=[num2str(Wgmat) blank num2str(Wg)  blank num2str(dWg)];
     Title={['Targets and Node Information']
     ['Network Name'  '  ' Nwname]
     ['Iteration' ' '  num2str(ep)]
     ['_____']
     ['Layer Node Connect   Weight        Delta Weight']
     ['_____']
     [(Wgformat)      ]
     ['_____']};
     funhelp;
     h = uicontrol(currfig, 'style', 'list','units','norm',...
     'position', [left bottom .9 .75],...
     'FontName','Courier', ...
     'FontSize',10, ...
     'string', Title, 'callback', 'NN_setup1 OutTarInfm',...
     'BackGroundColor', [1 1 1],...
     'Max', 1);
elseif strcmp(action,'Cmpout'),
  for i = 1:size(TG,2);
     Num_fig=5;
     figure(Num_fig);
     hold on;
     plot([0 1],[0 1],'r:');
     plot(PD(:,i),(TG(:,i))','o');
```

```
    hold off;
  end;
elseif strcmp(action,'NWinform'),
  a = findobj(0,'tag','NN_setup1');
  Itemin =  (get(findobj(a,'Tag','Tag_IteMin'),'String'))
  RMSMin =  (get(findobj(a,'Tag','Tag_MinRMS'),'String'))
  mom  =  (get(findobj(a,'Tag','Tag_mom'),'String'))
  SSEtn  = (get(findobj(a,'Tag','Tag_RMStrn'),'String'));
  SSEtst  = (get(findobj(a,'Tag','Tag_RMStest'),'String'));
  figurePos=[20 30 400 400];
  figNumber=figure('NumberTitle','off','name','Neural Network Information... ',...
    'color',[.8 .8 .8],'Position',figurePos);
  currfig=figNumber;
  category='Neural Network Information';
  ttlStr=' ';
  Title={['Network name:' '      ' Netwname]
    ['Number of Layers:' '      ' num2str(Nmlayer)]
    ['Input Layer:' ]
    ['   ' 'Nodes:' '              ' num2str(Ndinp)]
    ['   ' 'Transfer Fuction:' '  ' 'Linear']
    ['Hidden Layer1:' ]
    ['   ' 'Nodes:' '              ' num2str(NdHd1)]
    ['   ' 'Transfer Fuction:' '  ' num2str(ActHd1)]
    ['Hidden Layer2:' ]
    ['   ' 'Nodes:' '              ' num2str(NdHd2)]
    ['   ' 'Transfer Fuction:' '  ' num2str(ActHd2)]
    ['Output Layer:' ]
    ['   ' 'Nodes:' '              ' num2str(Ndout)]
    ['   ' 'Transfer Fuction:' '  ' num2str(ActOP)]
```

```
        ['Training Information']
        ['    ' 'Iteration:' '          ' num2str(ep)]
            ['    ' 'Training Error:' '    ' num2str(SSEtn)]
        ['    ' 'Testset Error:' '     ' num2str(SSEtst)]
        ['    ' 'Learn Rate:' '        ' num2str(Mingrad)]
        ['    ' 'Momentum:' '          ' num2str(mom)]
        ['Input Node File:' '     ' IPPath IPFile '.TXT']
        ['Output Node File:' '    ' OPPath OPFile '.TXT']
        ['Training Patterns:' '   ' num2str(size(hout,1))]
        ['Test Patterns:' '       ' num2str(size(hout2,1))]
        ['Iteration at Min RMS(Test):' '    ' Itemin]
        ['Minimum RMS(Test Set):' '    ' RMSMin]};
    funhelp;
    h = uicontrol(currfig, 'style', 'list','units','norm',...
    'position', [left bottom .9 .75],...
    'FontName','Courier', ...
    'FontSize',10, ...
    'string', Title, 'callback', 'NN_setup1 OutTarInfm',...
    'BackGroundColor', [1 1 1],...
    'Max', 1);
elseif strcmp(action,'OutTarInfm')
    figurePos=[20 30 400 400];
    figNumber=figure('NumberTitle','off','name','Targets/Outputs Information...',...
      'color',[.8 .8 .8],'Position',figurePos);
    currfig=figNumber;
    category='Targets/Outputs Information';
    ttlStr=' ';
    targets = [Xout' Xout2'];
    outputs = [hout' hout2'];
```

```
    a=targets';
    b=outputs';
    ai=[];
    for j=1:size(a,2)
        for i = 1:size(a,1)
         au=[i j a(i,j) b(i,j)];
         ai =[ai
              au];
        end
    end
    Title={'Targets and Node Information'
      ['Network Name'  '   ' Nwname]
      ['Iteration' ' '   num2str(ep)]
      '-------------------------------------------------------------'
      'Number      Node     Targets      Outputs'
      num2str(ai)
      '-------------------------------------------------------------'};
funhelp;
h = uicontrol(currfig, 'style', 'list','units','norm',...
'position', [left bottom .9 .75],...
'FontName','Courier', ...
'FontSize',10, ...
'string', Title, 'callback', 'NN_setup1 OutTarInfm',...
'BackGroundColor', [1 1 1],...
'Max', 1);
end;
```

ตัวอย่างโปรแกรมที่ 2 สำหรับอัลกอริธึมการเรียนรู้แบบ Levenberg-Marquardt ที่ใช้ในการฝึกข่ายงานและทดสอบแบบจำลอง

```
function
[a,b,c,d,e,f,g,h,i,j,k,m,n,o,p,q,r,s,t,u,v,x,z1,z2,z3,z4,z5,z6]=trainNN1(IPFile,OPFile,IPPath,OPPath,...
            Ndinp,NdHd1,Ndout,ActHd1,ActOP,Ite,Momentum,IncMom,DecMom,...
          MaxMom,SSETg,Mingrad,Epoupd,Testset,NdHd2,ActHd2)
 % Transfer data from variable of NN_setup to variable of trainNN1 program
 act_1=ActHd1;
 TP=[Epoupd Ite SSETg Mingrad Momentum IncMom DecMom MaxMom];
 S1=NdHd1;
 if nargin==18
    act_2=ActOP;
    S2=Ndout;
 elseif nargin ==20
    act_2 =ActHd2;
    S2=NdHd2
    act_3 =ActOP;
    S3=Ndout;
 end
%Load Data file to variable
 FileInput = [IPFile '.TXT'];
 FileOutput = [OPFile '.TXT'];
 load(FileInput);
 load(FileOutput);
 inpdat=eval(IPFile);
 outpdat=eval(OPFile);
 [A1,B1]=size(inpdat); % A = number of record, B= number of input
 [A2,C1]=size(outpdat);
rowtrnIP = A1-Testset;
 rowtrnOP = A2-Testset;
 % Set Data Training File
```

```
inpdata = inpdat(1:rowtrnIP,1:Ndinp-1);
outpdata = outpdat(1:rowtrnOP,1:Ndout);
[A,B]=size(inpdata);
[A,C]=size(outpdata);
 % Set Data Test File
inptest = inpdat(rowtrnIP+1:A1,1:Ndinp-1);
outtest = outpdat(rowtrnOP+1:A2,1:Ndout);
[D,E]=size(inptest);% D = number of record. E= number of input
[D,F]=size(outtest);
Xin = inpdata';
Xout = outpdata';
% Pre-normalize Input and Output of Training Data
[X,minp,maxp,T,mint,maxt]=premnmx2(Xin,Xout);
Xadd=delaysig(T,1,1);
X= [X;Xadd];
P=[X];
T2=[];
clear inpdata outpdata
Xin=Xin';
Xout=Xout';
if Testset >=1
   Xin2 = inptest';
   Xout2 = outtest';
   % Pre-normalize Input and Output of Test Set Data
   [mr,mc] = size(Xin2);
   [nr,nc] = size(Xout2);
   oneQp = ones(1,mc);
   oneQt = ones(1,nc);
   X2 = (Xin2-minp*oneQp)./((maxp-minp)*oneQp)*(1-0)+0;
```

```
    T2 = (Xout2-mint*oneQt)./((maxt-mint)*oneQt)*(1-0)+0;
    X2add=delaysig(T2,1,1);
    X2=[X2;X2add];%
    P2=[X2];
   [R3,Q3]=size(X2);
  end
  Xin2=Xin2'
  Xout2=Xout2'
%initialize network architecture
if nargin == 18
     %initff is used to initialize the weights and biases
     [W1,b1,W2,b2]=initff(P,S1,act_1,T,act_2);
     %training by Levenberg-Marquardt Algorithms
   [W1,b1,W2,b2,ep,tr,dw1,dw2,db1,db2,A2,w1min,w2min,b1min,b2min]=trainlm1(W1,b1,act_1,...
     W2,b2,act_2,P,T,TP,P2,T2);
   %Output value
   [A11,A1] = simuff(P,W1,b1,act_1,W2,b2,act_2);
   [A11min,A1min] = simuff(P2,w1min,b1min,act_1,w2min,b2min,act_2);
 elseif nargin ==20
   [W1,b1,W2,b2,W3,b3]=initff(P,S1,act_1,S2,act_2,S3,act_3);
   [W1,b1,W2,b2,W3,b3,ep,tr,dw1,dw2,dw3,db1,db2,db3,A2,w1min,w2min,w3min,b1min,b2min,...
     b3min]=trainlm1(W1,b1,act_1,W2,b2,act_2,W3,b3,act_3,P,T,TP,P2,T2);
   [A11,A12,A1] = simuff(P,W1,b1,act_1,W2,b2,act_2,W3,b3,act_3);
   [A11min,A12min,A1min] =
simuff(P2,w1min,b1min,act_1,w2min,b2min,act_2,w3min,b3min,act_3);
 end;
 %Calculate SSE error
 SSE=sumsqr(A1-T);
```

```
% Post-normalization Output of Training Data
hout=postmnmx2(A1,mint,maxt);
hout=hout';
Xout =Xout;
clear A B C
%results
 % Post-normalization Output of Test Set Data
if Testset>1
   SSETest=sumsqr(A2-T2);
   hout2=postmnmx2(A2,mint,maxt);
   hout2=hout2';
   SSEMin=sumsqr(A1min-T2);
   houtmin=postmnmx2(A1min,mint,maxt);
   houtmin=houtmin';
   fid2= fopen('c:\rmstest2.txt','wt');
   Outmat=[houtmin;Xout2];
   fprintf(fid2,'%15.5f\n',Outmat);
   fclose(fid2)
  clear D E F
end

%Record variable for plot comparison of Normalized Targets vs. Net outputs
PD=[A1';A2'];
TG=[T';T2'];
% Send variable to NN_setup
if nargin ==18
     a= hout; b= SSE;c=hout2;d=SSETest;e=W1;f=b1;g=W2;h=b2;...
     i=Xout;j=Xout2;k=ep;m=tr;n=TG;o=PD;p=dw1;q=dw2;r=db1;s=db2;
     t=w1min;u=b1min;v=w2min;x=b2min;
```

```
elseif nargin ==20
   a= hout; b= SSE;c=hout2;d=SSETest;e=W1;f=b1;g=W2;h=b2;...
   i=W3;j=b3;k=Xout;m=Xout2;n=ep;o=tr;p=TG;q=PD;r=dw1;s=dw2;t=dw3;u=db1;...
   v=db2;x=db3;z1=w1min;z2=b1min;z3=w2min;z4=b2min;z5=w3min;z6=b3min;
end;
```

ตัวอย่างโปรแกรมที่ 3 สำหรับการทดสอบข่ายงานที่ได้จากโปรแกรมที่ 2 ด้วยชุดข้อมูลที่ใช้ทดสอบออนไลน์จริง

```
function
[a,b,c,d,e,f,g,h,i,j,k,m,n,o,p,q,r,s,t,u,v,x,z1,z2,z3,z4,z5,z6]=trainNN1(IPFile,OPFile,IPPath,OPPath,...
        Ndinp,NdHd1,Ndout,ActHd1,ActOP,Ite,Momentum,IncMom,DecMom,...

MaxMom,SSETg,Mingrad,Epoupd,IPFiTn,OPFiTn,TestsetTn,W1r1,b1r1,W2r1,b2r1,W3r1,b3r1,...
        NdHd2,ActHd2)
% Transfer data from variable of NN_setup to variable of trainNN1 program
act_1=ActHd1;
TP=[Epoupd Ite SSETg Mingrad Momentum IncMom DecMom MaxMom];
S1=NdHd1;
 if isempty(NdHd2)
   act_2=ActOP;
   S2=Ndout;
 else
   act_2 =ActHd2;
   S2=NdHd2;
   act_3 =ActOP;
   S3=Ndout;
 end
%Load Data file to variable
```

```
IPFi= strrep(IPFiTn,'.txt','');
OPFi= strrep(OPFiTn,'.txt','');
FileInput = [IPFile '.TXT']
FileOutput = [OPFile '.TXT'];
load(FileInput);
load(FileOutput);
load(IPFiTn);
load(OPFiTn);
inpdat=eval(IPFile);
outpdat=eval(OPFile);
inpTndat =eval(IPFi);
outpTndat=eval(OPFi);
h2c2_1=inpTndat(:,1);
h2c2_1=h2c2_1';
h2c2_1=delaysig(h2c2_1,1,1);
h2c2_1=h2c2_1';
inpTndat=[h2c2_1 inpTndat];
h2c2_2=inpdat(:,1);
h2c2_2=h2c2_2';
h2c2_2=delaysig(h2c2_2,1,1);
h2c2_2=h2c2_2';
inpdat=[h2c2_2 inpdat];
[A1,B1]=size(inpdat);
[A2,C1]=size(outpdat);% A = number of record, B= number of input
inpdata = inpdat;
outpdata = outpdat;
[A,B]=size(inpdata);
[A,C]=size(outpdata);
rowtrnIP = A1;
```

```
rowtrnOP = A2;
%Data Test Set and Training
inpdata = inpdat(1:rowtrnIP,1:(Ndinp-1));
outpdata = outpdat(1:rowtrnOP,1:Ndout);
[A,B]=size(inpdata);
[A,C]=size(outpdata)
[R1,R2]=size(inpTndat)
[R3,R4]=size(outpTndat)
rwtninpfi = R1-TestsetTn;
rwtnoutpfi =R3-TestsetTn;
inpTndata=inpTndat(1:rwtninpfi,1:(Ndinp-1));
outpTndata = outpTndat(1:rwtnoutpfi,1:Ndout);
% Traing Data and Test Set Data
Xin = inpdata;
Xout = outpdata;
clear inpdata outpdata
XinTn = inpTndata;
XoutTn = outpTndata;
outadd = delaysig(XoutTn',1,1);
outadd=outadd';
XinTn =[XinTn outadd];
set(findobj(gcf,'Tag','Tag_Nmpat'),'String',num2str(rowtrnIP));
outputon = Xout(1,1:Ndout);
out = Xout(1,1:Ndout);
% Initialize Variable
Xinmat=[];
Xoutmat=[];
XinT1=[];
XoutT1=[];
```

```
P2=[];
T2=[];
output=[];
out1=  Xout(1,1:Ndout);
for i= 1:rowtrnIP
   set(findobj(gcf,'Tag','Tag_Nopat'),'String',num2str(i));
   if  Xout(i,:)>zeros(1,1:Ndout)
      out = Xout(i,1:Ndout);
   end;
   Xinst = [Xin(i,:) outputon];
   Xinmat =[Xinmat;Xinst];
   Xoutmat =[Xoutmat;out];
   if Xout(i,:)>zeros(1,1:Ndout)
      XinT1 = [XinTn;Xinmat];
      XoutT1 = [XoutTn;Xoutmat];
      % Pre-Normalized Data
      [Xup,minp,maxp,Tup,mint,maxt]=premnmx2(XinT1',XoutT1');
      if isempty(NdHd2)
            [W1,b1,W2,b2,ep,tr,dw1,dw2,db1,db2,A2,w1min,w2min,b1min,b2min]=trainlm1
(W1r1,b1r1,act_1,...
            W2r1,b2r1,act_2,Xup,Tup,TP,P2,T2);
            W1r1=w1min;
            b1r1=b1min;
            W2r1=w2min;
            b2r1=b2min;
            [A11,A3] = simuff(Xup,W1r1,b1r1,act_1,W2r1,b2r1,act_2);
            [aa1,bb1]=size(A3)
      else
         % training Levenberg-Marquardt Algorithm
```

```
[W1,b1,W2,b2,W3,b3,ep,tr,dw1,dw2,dw3,db1,db2,db3,A2,w1min,w2min,w3min,b1min,b2min,...

b3min]=trainlm1(W1r1,b1r1,act_1,W2r1,b2r1,act_2,W3r1,b3r1,act_3,Xup,Tup,TP,P2,T2);
            W1r1=w1min;
            b1r1=b1min;
            W2r1=w2min;
            b2r1=b2min;
            W3r1=w3min;
            b3r1=b3min;
            [A11,A12,A3] = simuff(Xup,W1r1,b1r1,act_1,W2r1,b2r1,act_2,W3r1,b3r1,act_3);
           [aa2,bb2]=size(A3)
        end;
      [kk,ii]=size(Xup)
      [aa,bb]=size(A3)
      [cc,dd]=size(Tup)
      SSEtn=sumsqr(A3-Tup);
      % Post-Normalized Data
      outtn=postmnmx2(A3,mint,maxt);
      outtn=outtn'
      A3=A3';
      [A1,B1]=size(A3);
      outt1 = A3(A1,B1);
    else
      [mr,mc] = size(Xinst');
      [nr,nc] = size(Xout);
      oneQp = ones(1,mc);
      oneQt = ones(1,nc);
      X = (Xinst'-minp*oneQp)./((maxp-minp)*oneQp)*(1-0)+0;
```

```
    if isempty(NdHd2)
      [A11,A1a] = simuff(X,W1r1,b1r1,act_1,W2r1,b2r1,act_2);
    else
      [A11,A12,A1a] = simuff(X,W1r1,b1r1,act_1,W2r1,b2r1,act_2,W3r1,b3r1,act_3);
    end
    SSEon=sumsqr(A1a-out);
    [A1,B1]=size(A1a);
    outt1 = A1a(A1,B1);
  end;
  outputon=postmnmx2(outt1,mint,maxt);
  outt1=outt1';
  output=[output;outputon];
end;
outon=postmnmx2(A1a,mint,maxt);
 %results
PD=[A3];
TG=[Tup'];
Xout2=[XoutTn;Xout];
% return variable  back to NN_setup
 a= output; b= SSEon;c=outtn;d=SSEtn;e=W1;f=b1;g=W2;h=b2;...
 i=W3;j=b3;k=Xout;m=Xout2;n=ep;o=tr;p=TG;q=PD;r=dw1;s=dw2;t=dw3;u=db1;,...
 v=db2;x=db3;z1=w1min;z2=b1min;z3=w2min;z4=b2min;z5=w3min;z6=b3min;
```

ตัวอย่างโปรแกรมที่ 4 สำหรับสร้างชุดข้อมูลสำหรับแบบจำลองทางคณิตศาสตร์ของเครื่องปฏิกรณ์ถังกวนต่อเนื่อง

```
function modelou(action)
% set Operating Condition of the Continuous Stirred Tank Reactor
clear all
close all
% set initial  value of each parameter
```

```
Q = 100;        % l/min
CAf = 1;        % mol/l
Tf = 350;   %K
Tcf = 350;  %K
V = 100;    %l
hA = 700000;% cal/min.k
K0 = 7.2*10^10; %/min
E_R = 9950; % K .E_R=E/R
DelH = -200000; % cal/mol
D = 1000;       %g/l
Dc = 1000;          %g/l
Cp = 1;         %cal/g.k
Cpc = 1;        %cal/g.k
qc0 = 103.41;   %l/min
T =440.2 ;      %K
CA= 0.0836;     %mol/l
Tcf_sp=350;     % Temp cooling initial
Tsp = 440.2;    % Temp reactor
CAsp = 0.0836;  % Concentration of A material
% set initial  parameter
t = 0.5;
%set time and step time
time = 20;
dt = 0.01;
nt = time/dt+1;
tm = (0:dt:time);
norm =menu('Select change of coolant temp', '+/-15% change of  coolant flow')
disp('')
if norm ==1
```

```
    Tcf_L =Tcf;
    tit = ' +/-15% change of  coolant flow'
elseif norm==2
end;
Tcf(1)= 350;
i=1;
qcc=[];
for i=1:1:nt,
    qc =(2*rand(1)-1)*0.15*qc0+qc0;
    if i == 1:10:nt
      Tcf=(2*rand(1)-1)*0.1*Tcf(1)+Tcf(1);
    end;
    if i==1
      T(i)  = 440.2;
      CA(i) = 0.0836;
   else
      T(i)=T(i-1)+(Q/V*(Tf-T(i-1))+(-DelH*K0*CA(i-1)/(D*Cp)*exp(-E_R/T(i-1)))+Dc*Cpc*qc/
(D*Cp*V)*(1-exp(-hA/(qc*Dc*Cpc)))*(Tcf(1)-T(i-1)))*dt;
      CA(i)=CA(i-1)+(Q/V*(CAf-CA(i-1))-K0*CA(i-1)*exp(-E_R/T(i-1)))*dt ;
   end;
   qcc=[qcc;qc];
end;
% Create Input / Target Data for vary qcc and record CA by no change flow qcc
inputtrn=qcc(1:1500,1),
outputtrn=CA(1:1500,1)
inputtest=qcc(1501:2001,1);
outputtest=CA(1501:2001,1)
figure(2)
subplot(211)
```

```
plot(tm,qcc,'r-')
title(tit)
xlabel('time(min)')
ylabel('Reactor coolant flow qc(l/min)')
axis
subplot(2,1,2)
plot(tm,CA,'r-')
xlabel('time (min)')
ylabel('Concentration :CA (mol/l)')
axis
[Savefile,Savepath]=uiputfile('*.MAT','Save As input file',50,50);
Savename=[Savepath Savefile]
save(Savename,'qcc')
[SavefileO,SavepathO]=uiputfile('*.MAT','Save As output file',50,50);
Savename1=[SavepathO SavefileO]
save(Savename1,'CA')
```

ตัวอย่างโปรแกรมที่ 5 โปรแกรมสำหรับทดสอบความถูกต้องของโปรแกรมที่ใช้สำหรับสร้างข่ายงานและทดสอบข่ายงานนิวรอน โดยศึกษาการทำนายความเข้มข้นของสาร A ของเครื่องปฏิกรณ์ถังกวนแบบต่อเนื่อง

```
function
[a,b,c,d,e,f,g,h,i,j,k,m,n,o,p,q,r,s,t,u,v,x,z1,z2,z3,z4,z5,z6]=trainNN1(IPFile,OPFile,IPPath,OPPath,...
        Ndinp,NdHd1,Ndout,ActHd1,ActOP,Ite,Momentum,IncMom,DecMom,...
        MaxMom,SSETg,Mingrad,Epoupd,Testset,NdHd2,ActHd2)
act_1=ActHd1;
TP=[Epoupd Ite SSETg Mingrad Momentum IncMom DecMom MaxMom];
S1=NdHd1;
if nargin==18
   act_2=ActOP;
   S2=Ndout;
```

```
elseif nargin ==20
   act_2 =ActHd2;
   S2=NdHd2
   act_3 =ActOP;
   S3=Ndout;
end
load(IPFile);
load(OPFile);
qcc=qcc';
CAA=CA
inpdat=delaysig(qcc,1,2);
inp=delaysig(CAA,1,3);
inpdat=[qcc;inpdat;inp];
inpdat=inpdat';
outpdat=CA';
[A1,B1]=size(inpdat);
[A2,C1]=size(outpdat);
rowtrnIP = A1-Testset;
rowtrnOP = A2-Testset;
%Data Training File
inpdata = inpdat(1:rowtrnIP,1:Ndinp-1);
outpdata = outpdat(1:rowtrnOP,1:Ndout);
[A,B]=size(inpdata);
[A,C]=size(outpdata);
% Data Test File
inptest = inpdat(rowtrnIP+1:A1,1:Ndinp-1);
outtest = outpdat(rowtrnOP+1:A2,1:Ndout);
[D,E]=size(inptest);% D = number of record, E= number of input
[D,F]=size(outtest);
```

```
% Training
Xin = inpdata';
Xout = outpdata';
[X,minp,maxp,T,mint,maxt]=premnmx2(Xin,Xout);
Xadd=delaysig(T,1,1);
X= [X;Xadd];
P=[X];
T2=[];
clear inpdata outpdata
Xin=Xin';
Xout=Xout';
if Testset >=1
    Xin2 = inptest';
    Xout2 = outtest';
    [mr,mc] = size(Xin2);
    [nr,nc] = size(Xout2);
    oneQp = ones(1,mc);
    oneQt = ones(1,nc);
    X2 = (Xin2-minp*oneQp)./((maxp-minp)*oneQp)*(1-0)+0;
   T2 = (Xout2-mint*oneQt)./((maxt-mint)*oneQt)*(1-0)+0;
   X2add=delaysig(T2,1,1);
  X2=[X2;X2add];%
  P2=[X2];
  [R3,Q3]=size(X2);
end
Xin2=Xin2'
Xout2=Xout2'
%initialize network architecture
if nargin == 18
```

```
    %initff is used to initialize the weights and biases
    [W1,b1,W2,b2]=initff(P,S1,act_1,T,act_2);
    %training parameters
    [W1,b1,W2,b2,ep,tr,dw1,dw2,db1,db2,A2,w1min,w2min,b1min,b2min]=trainlm1(W1,b1,act_1,...
     W2,b2,act_2,P,T,TP,P2,T2);
    %Output value
    [A11,A1] = simuff(P,W1,b1,act_1,W2,b2,act_2);
    [A11min,A1min] = simuff(P2,w1min,b1min,act_1,w2min,b2min,act_2);
elseif nargin ==20
    [W1,b1,W2,b2,W3,b3]=initff(P,S1,act_1,S2,act_2,S3,act_3);
    [W1,b1,W2,b2,W3,b3,ep,tr,dw1,dw2,dw3,db1,db2,db3,A2,w1min,w2min,w3min,b1min,b2min,...
    b3min]=trainlm1(W1,b1,act_1,W2,b2,act_2,W3,b3,act_3,P,T,TP,P2,T2);
    [A11,A12,A1] = simuff(P,W1,b1,act_1,W2,b2,act_2,W3,b3,act_3);
    [A11min,A12min,A1min] =
simuff(P2,w1min,b1min,act_1,w2min,b2min,act_2,w3min,b3min,act_3);
end;
%Calculate SSE error
SSE=sumsqr(A1-T);
% Post-Normalized Outputs for Training Set
hout=postmnmx2(A1,mint,maxt);
hout=hout';
Xout =Xout;
clear A B C
% Post-Normalize Output for Test Set
if Testset>1
   SSETest=sumsqr(A2-T2);
   hout2=postmnmx2(A2,mint,maxt);
   hout2=hout2';
   SSEMin=sumsqr(A1min-T2);
```

```
houtmin=postmnmx2(A1min,mint,maxt);
houtmin=houtmin';
fid2= fopen('c:\rmstest2.txt','wt');
Outmat=[houtmin;Xout2];
fprintf(fid2, %15.5f\n',Outmat);
fclose(fid2)
clear D E F
end
%plot comparison of Normalized Targets vs. Net outputs
PD=[A1';A2'];
TG=[T';T2'];
if nargin ==18
   a= hout; b= SSE;c=hout2;d=SSETest;e=W1;f=b1;g=W2;h=b2;...
   i=Xout;j=Xout2;k=ep;m=tr;n=TG;o=PD;p=dw1;q=dw2;r=db1;s=db2;
   t=w1min;u=b1min;v=w2min;x=b2min;
elseif nargin ==20
   a= hout; b= SSE;c=hout2;d=SSETest;e=W1;f=b1;g=W2;h=b2;...
   i=W3;j=b3;k=Xout;m=Xout2;n=ep;o=tr;p=TG;q=PD;r=dw1;s=dw2;t=dw3;u=db1;,...
   v=db2;x=db3;z1=w1min;z2=b1min;z3=w2min;z4=b2min;z5=w3min;z6=b3min;
end;
```

ภาคผนวก ค.

การใช้โปรแกรม "Neural Network"

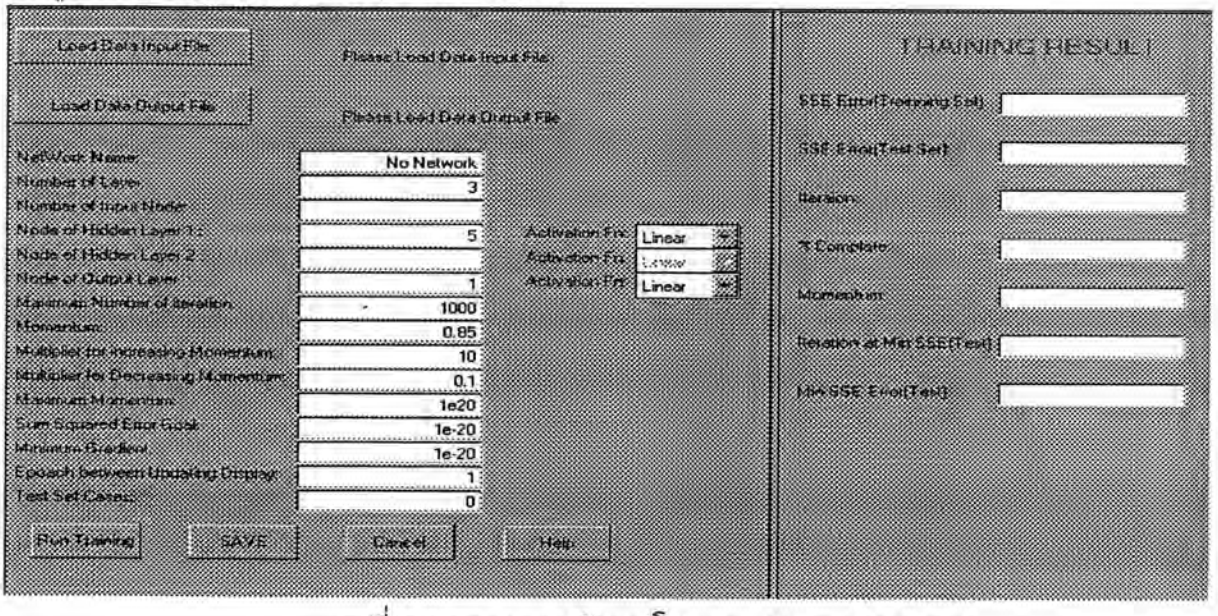
การสร้างแบบจำลองของกระบวนการโดยการใช้ข่ายงานนิวรัลจะใช้การเขียนด้วยโปรแกรมแม็ทแลบ ดังนั้นจะต้องประมวลผลอยู่บนโปรแกรมแม็ทแลบบนคอมพิวเตอร์รุ่นเพนเทียม (Pentium) ขึ้นไป ซึ่งจะมีรายละเอียดดังต่อไปนี้

1. การเริ่มต้นเข้าสู่โปรแกรม

ให้เรียกโปรแกรมแม็ทแลบ ซึ่งขึ้นเป็นตัวพร็อมพ์ (prompt) และให้พิมพ์คำว่า cd e:\prgneural เพื่อเปลี่ยนไดเรกทอรีไปที่มีโปรแกรมอยู่ และให้พิมพ์คำว่า "sel_network"

```
>>> cd e:\prgneural
>>> sel_network
```

หลังจากพิมพ์เรียบร้อยแล้วจะขึ้นหน้าจออินเทอร์เฟสเพื่อให้ติดต่อกับผู้ใช้งานโดยปรากฏบนจอภาพแสดงดังรูปที่ ค.1 ประกอบด้วย new , load และ exit



รูปที่ ค.1 แสดงเมนูหลักของโปรแกรม Neural Network

new ใช้สำหรับการสร้างแบบจำลองใหม่

load ใช้สำหรับการเรียกแบบจำลองที่สร้างมาแล้ว

exit ใช้เพื่อออกจากโปรแกรม

2 . การกำหนดโครงสร้างของข่ายงานนิวรัลและการกำหนดพารามิเตอร์ต่างๆ สำหรับการเรียนรู้

จากเมนูหลักกำหนดโครงสร้างของข่ายงานนิวรัลโดยการป้อนค่าดังต่อไปนี้

Load Data Input :	การกำหนดชุดข้อมูลอินพุท
Load Data Output :	การกำหนดชุดข้อมูลเป้าหมาย
Network Name :	ให้ตั้งชื่อข่ายงานนิวรัล
Number of Layer:	ให้ป้อนจำนวนชั้นของข่ายงานโดย ถ้า ป้อน 3 หมายถึง ชันของข่ายงานประกอบไปด้วย ชั้นอินพุท ชั้นซ่อน 1 ชั้น และชั้นเอาท์พุท ถ้าป้อน 4 จะเพิ่มชั้นซ่อนอีก 1 ชั้น Number of Input Node : ให้ป้อนจำนวนอินพุทที่ใช้ฝึกข่ายงาน
Node of Hidden Layer 1:	ให้ป้อนจำนวนนิวรัลในชั้นซ่อนที่ 1
Node of Hidden Layer 2:	ให้ป้อนจำนวนนิวรัลในชั้นซ่อนที่ 2 สำหรับกรณีที่จำนวนชั้นมีทั้งหมด 4 ชั้น
Node of Output:	ให้ป้อนจำนวนเอาท์พุทที่ต้องการ
Activation Function:	การกำหนดฟังก์ชันกระตุ้นของแต่ละชั้น ซึ่งจะมีให้เลือก คือ linear , sigmoid , tansig
Maximum Number of Iteration :	จำนวนที่ต้องการวนรอบซ้ำกันในการฝึกข่ายนิวรัล ถ้าข่ายงานมีการปรับค่าน้ำหนักถึงค่านี้จะหยุดการฝึกข่ายงาน
Momentum :	ค่าเริ่มต้นของค่าโมเมนตัมที่กำหนดให้สำหรับข่ายงาน
Multiplier for increasing Momentum :	ตัวคูณสำหรับการปรับค่าโมเมนตัมเพิ่มขึ้น
Multiplier for decreasing Momentum :	ตัวคูณสำหรับการปรับค่าโมเมนตัมลดลง

Maximum Momentum :	ค่าโมเมนตัมสูงสุดที่กำหนดไว้ ถ้าข่ายงานมีการปรับค่าน้ำหนักถึงค่านี้จะหยุดการฝึกข่ายงาน
Sum Squared Error Goal :	ค่าเป้าหมายของค่าความผิดพลาด ถ้าข่ายงานมีการปรับค่าน้ำหนักถึงค่านี้จะหยุดการฝึกข่ายงาน
Minimum Gradient :	ค่าเกรเดียนต่ำสุด ถ้าข่ายงานมีการปรับค่าน้ำหนักถึงค่านี้จะหยุดการฝึกข่ายงาน
Epoach between Updating Display :	การแสดงข้อมูลที่หน้าจอในรอบของการประมวลผล
Test Set Cases :	จำนวนชุดข้อมูลที่ใช้ทดสอบข่ายงานซึ่งจะนับจากชุดหลังสุดลงมาหาชุดหน้าสุด
Save :	การบันทึกข่ายงานนิวรัล
Run training :	การประมวลผลการฝึกข่ายงานนิวรัล ให้กดหลังจากป้อนค่าในทุกพารามิเตอร์เรียบร้อยแล้ว
Cancel :	การกลับไปสู่หน้าจอที่ผ่านมา

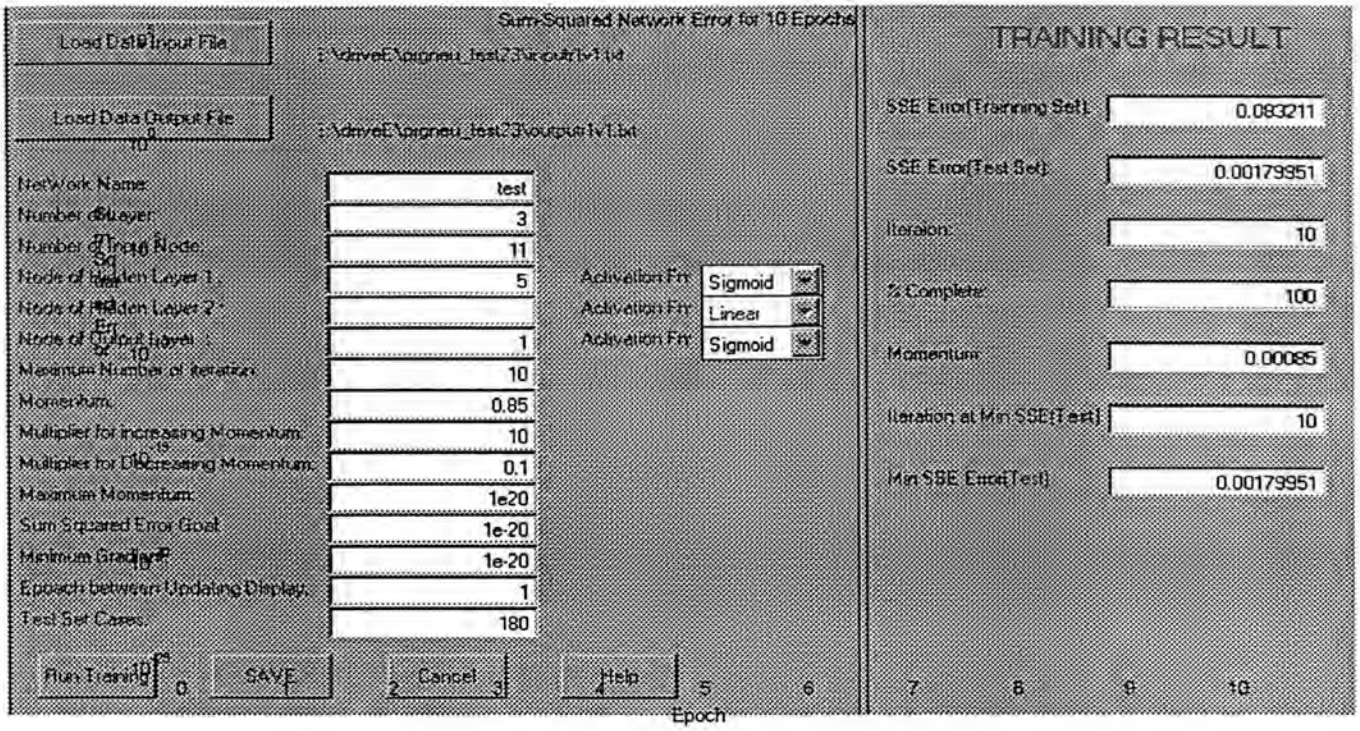
3 การเลือกชุดข้อมูล

ชุดข้อมูลที่ใช้ในการฝึกข่ายงานและใช้ในการทดสอบจะเป็นเท็กซ์ไฟล์ (text file) โดยคอลัมน์จะเป็นจำนวน อินพุทหรือเป้าหมาย และแถวจะเป็นจำนวนชุดข้อมูล (pattern)

4 การฝึกข่ายงานนิวรัล

เมื่อได้กำหนดโครงสร้างของข่ายงานนิวรัล เรียบร้อยแล้วให้ข่ายงานทำการเรียนรู้ตามที่ได้กำหนดไว้ โดยกดปุ่ม run training โปรแกรมจะทำการกำหนดค่าน้ำหนักเริ่มต้นโดยวิธีแบบสุ่ม และทำการประมวลผลในการจำลองข่ายงานนิวรัลให้ได้ค่าเอาท์พุทใกล้เคียงกับค่าเป้าหมายโดยทำการปรับค่าน้ำหนักเพื่อให้ได้ค่าความผิดพลาดอยู่ในช่วงที่ยอมรับได้

หลังจากที่ได้มีการฝึกข่ายงาน (run training) แล้ว ข่ายงานนิวรัลจะเริ่มเรียนรู้ตามพารามิเตอร์ต่างๆ ที่กำหนดไว้โดยใช้อัลกอริธึมแบบ Levenberg - Marquardt ซึ่งจะมีการปรับค่าน้ำหนัก ไบอัส และค่าโมเมนตัม และค่าเกรเดียนโดยอัตโนมัติตามชุดข้อมูลอินพุทและเอาท์พุท ซึ่งจะมีการแสดงการเปลี่ยนค่าของความผิดพลาด (SSE Error)ของชุดฝึกข่าย



รูปที่ ค.2 แสดงผลการฝึกข่ายงานทุกๆครั้งที่มีการแสดงผลที่หน้าจอ (update display)

ค่าความผิดพลาดของข่ายงาน (SSE Error) ของชุดทดสอบข่ายงานในระหว่างการเรียนรู้ซึ่งจะแสดงผลการฝึกข่ายงานในทางด้านขวามือของรูปที่ ค.2 โดยมีการเปลี่ยนหน้าจอทุกๆรอบของค่าที่กำหนดไว้ใน Epoach between Updating Display

SSE Error (Training set) : ค่าความผิดพลาดของชุดฝึกข่ายงาน ณ จำนวนรอบนั้นๆ (iteration)

SSE Error (Test set): ค่าความผิดพลาดของชุดทดสอบข้อมูล (validation set) ณ จำนวนรอบนั้นๆ

Iteration: รอบที่ประมวลผลอยู่ในขณะนั้น

%Complete: เปอร์เซนต์ที่ทำการวนรอบไปแล้ว

Momentum ค่าโมเมนตัมที่ถูกปรับสำหรับข่ายงาน ณ จำนวนรอบนั้นๆ

Iteration at Min SSE (Test) : จำนวนรอบที่ทำให้ค่าความผิดพลาดมีค่าต่ำสุดของชุดข้อมูลที่ใช้ในการทดสอบ (validation set)

Min SSE Error (Test) : ค่าความผิดพลาดต่ำสุดของชุดข้อมูลที่ใช้ในการทดสอบ (validation set)

5. การทดสอบการเรียนรู้ของข่ายงานนิวรัล (Cross validation)

เมื่อการเรียนรู้ของฝึกข่ายงานถึงจุดสิ้นสุดแล้วนั้นคือจะได้ข่ายงานนิวรัลที่จำลองกระบวนการที่เหมาะสมแล้วจะนำชุดข้อมูลที่ใช้ทดสอบป้อนเข้าไปเพื่อทำนายค่าเอาท์พุทออกมาแล้วหาค่าความผิดพลาดว่าอยู่ในช่วงที่ยอมรับได้หรือไม่ซึ่งโปรแกรมจะทำให้อัตโนมัติ (โดยจะต้องป้อนค่า Testset>0 ก่อนทำการประมวลผล) และหน้าจอจะแสดงผลค่า SSE Error ของชุดทดสอบหลังจากการประมวลผลในแต่ละรอบที่มีการเรียนรู้

6. การแสดงผลในการทดสอบประสิทธิภาพของข่ายงาน

6.1 Network Information

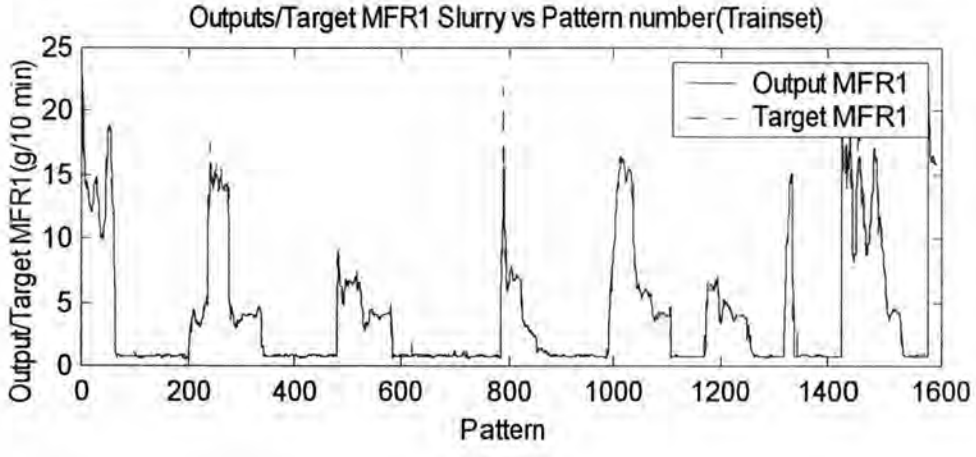
เป็นการแสดงรายละเอียดต่างๆของข้อมูลพารามิเตอร์ที่กำหนดไว้และผลของค่าความผิดพลาดของข่ายงานดังแสดงในรูปที่ ค.3

```
Network name:
Number of Layers:       3
Input Layer:
   Nodes:               11
   Transfer Fuction:    Linear
Hidden Layer1:
   Nodes:               5
   Transfer Fuction:    logsig
Hidden Layer2:
   Nodes:
   Transfer Fuction:    purelin
Output Layer:
   Nodes:               1
   Transfer Fuction:    logsig
Training Information
   Iteration:           10
   Training Error:      0.083211
   Testset Error:       0.00179951
   Learn Rate:          1e-020
   Momentum:            0.00085
Input Node File:        E:\driveE\prgneu_test23\inputr1v1.TXT
Output Node File:       E:\driveE\prgneu_test23\outputr1v1.TXT
Training Patterns:      1610
Test Patterns:          180
Iteration at Min SSE(Test):     10
Minimum SSE(Test Set):      0.00179951
```

รูปที่ ค. 3 แสดง Network Information

6.2 Case Outputs/Targets (Training set)

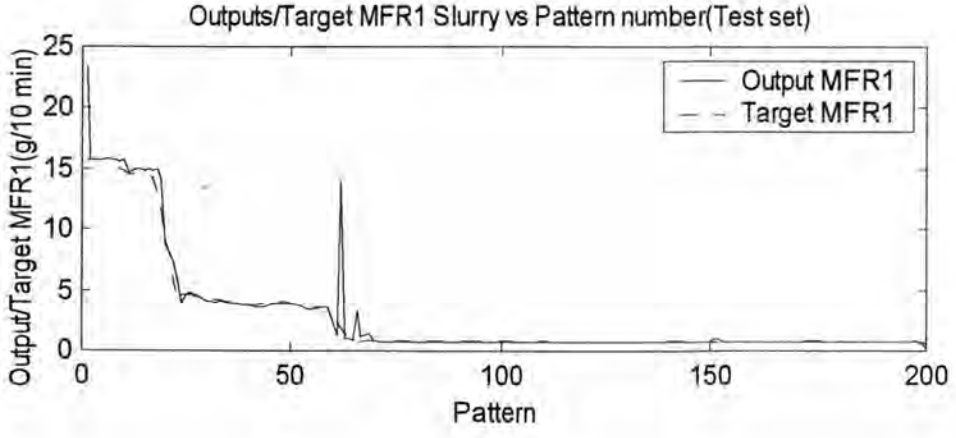
เป็นการแสดงการพล็อตกราฟระหว่างค่าเอาท์พุทที่ได้จากข่ายงานและค่าเป้าหมายของชุดฝึกข่ายงาน ดังแสดงในรูปที่ ค.4



รูปที่ ค.4 แสดง Case Outputs/Targets(Training Set)

6.3 Case Outputs/Targets (Test set)

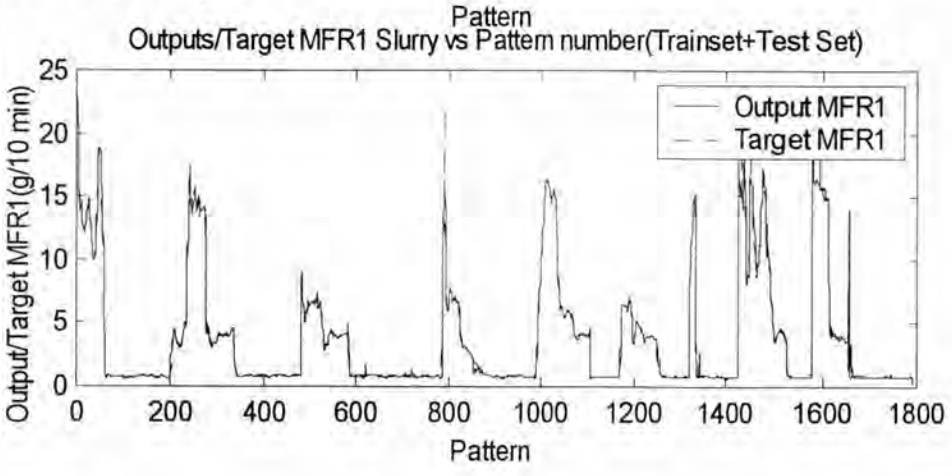
เป็นการแสดงการพล็อตกราฟระหว่างค่าเอาท์พุทที่ได้จากข่ายงานและค่าเป้าหมายของชุดทดสอบข่ายงาน ดังแสดงในรูปที่ ค.5



รูปที่ ค.5 แสดง Case Outputs/Targets (Test set)

6.4 Outputs/Targets(Train+Test)

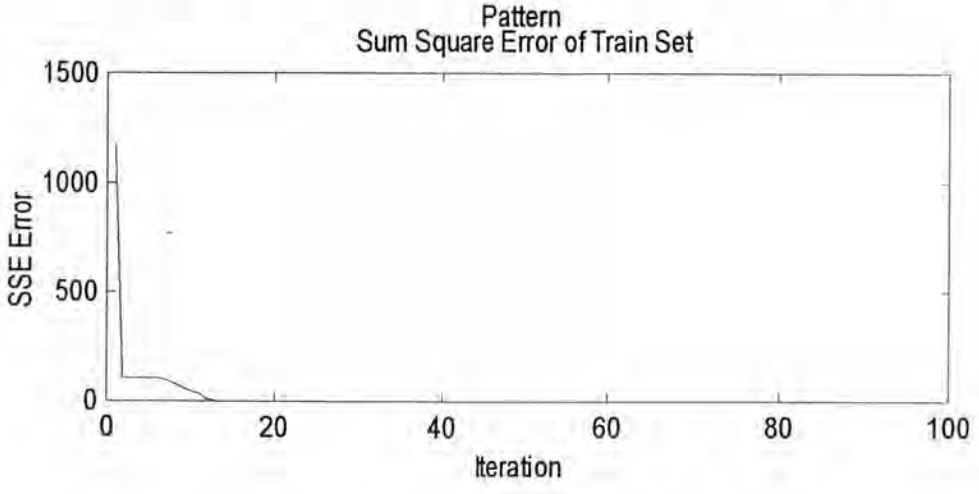
เป็นการแสดงการพล็อตกราฟระหว่างค่าเอาท์พุทที่ได้จากข่ายงานและค่าเป้าหมายของชุดฝึกข่ายงาน และชุดทดสอบข่ายงาน ดังแสดงในรูปที่ ค.6



รูปที่ ค.6 แสดง Case Outputs/Targets (Train+Test set)

6.5 SSE Error history vs Iteration (Trn)

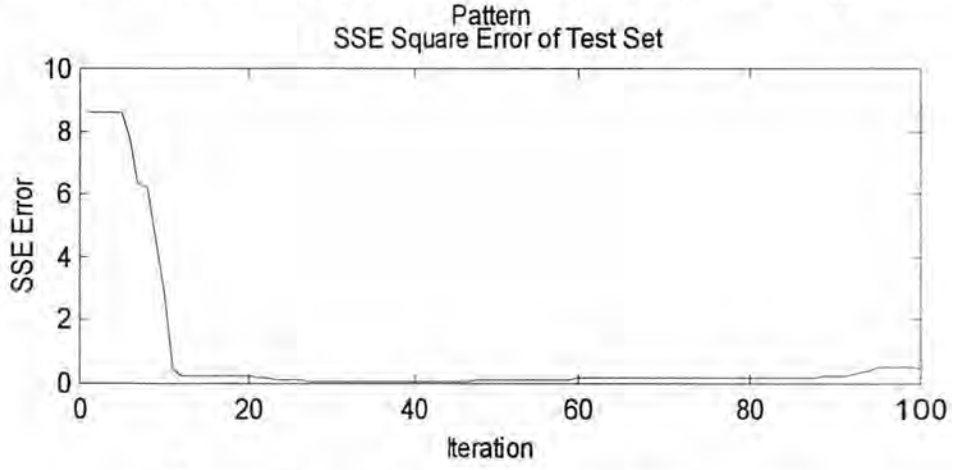
เป็นการแสดงกราฟระหว่างค่า SSE ของชุดฝึกข่ายงานที่รอบต่างๆ ดังแสดงในรูปที่ ค.7



รูปที่ ค.7 แสดง SSE Error history vs Iteration (Train set)

6.6 SSE Error history vs Iteration (Test)

เป็นการแสดงกราฟระหว่างค่า SSE ของชุดทดสอบข่ายงานที่รอบต่างๆ ดังแสดงในรูปที่ ค.8



รูปที่ ค.8 แสดง SSE Error history vs Iteration (test set)

6.7 Node Weight and Deltas

การแสดงข้อมูลของค่าน้ำหนักและค่าน้ำหนักที่เปลี่ยนแปลงไปในแต่ละรอบที่ทำการฝึกข่ายงานสำหรับการเชื่อมโยงระหว่างชั้นหนึ่งไปอีกชั้นหนึ่ง ดังแสดงในรูปที่ ค.9

Network Weight and Current Adjustment Deltas

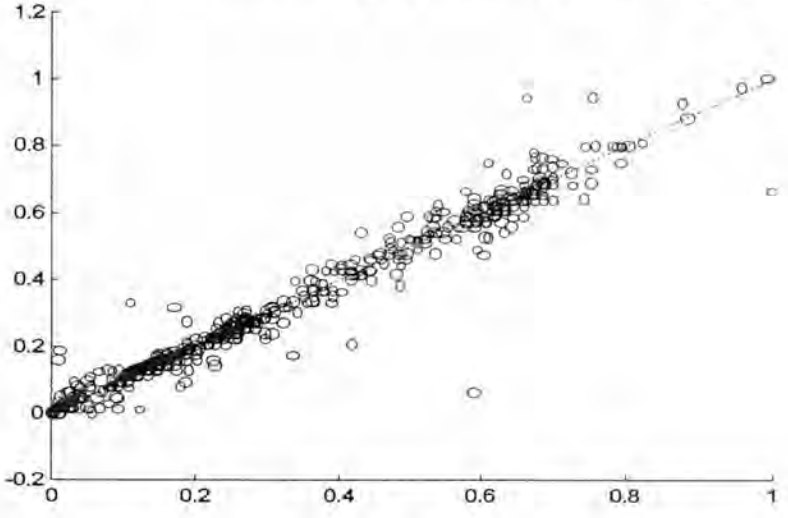
Targets and Node Information
Network Name test
Iteration 30

Layer	Node	Connect	Weight	Delta Weight
1	1	1	47.41507	0.033963
1	1	2	9.123639	1.4948
1	1	3	-51.65534	0.020407
1	1	4	9.908125	1.5371
1	1	5	-71.66579	-1.1427
1	2	1	76.98609	1.0746
1	2	2	-23.06497	-0.81278
1	2	3	150.6386	0.81249
1	2	4	-12.73572	-0.15352
1	2	5	30.68298	0.46877
1	3	1	34.19767	0.13361
1	3	2	2.798065	-0.089505
1	3	3	4.003575	-0.22043
1	3	4	-2.095444	0.024544
1	3	5	2.556609	0.065331
1	4	1	19.99733	-0.5892
1	4	2	3.698743	-0.098908
1	4	3	0.1380198	-0.14134
1	4	4	-0.7587532	-0.076226
1	4	5	1.565066	0.12846
1	5	1	-6.167321	0.1501
1	5	2	-21.05036	-0.070246
1	5	3	11.64629	0.6668
1	5	4	3.148045	-0.06557
1	5	5	-9.465825	0.17554

รูปที่ ค.9 แสดง Node Weight and Deltas

6.8 Scatter Comparison of Normalized Targets vs Net Outputs

เป็นกราฟแสดงการเปรียบเทียบการกระจายตัวของค่าเป้าหมายและเอาท์พุทที่ยังไม่ผ่านการสเกลกลับมาเป็นค่าที่ใช้งานคือมีค่าอยู่ระหว่างช่วง 0 -1(normalized or pre-processing) ดังแสดงในรูปที่ ค.10



รูปที่ ค.10 แสดง Scatter comparision Normalized Targets vs Net Outputs

6.9 Outputs/Targets Information

เป็นการแสดงผลข้อมูลเปรียบเทียบค่าเอาท์พุทที่ได้จากข่ายงานกับเป้าหมาย ดังแสดงในรูปที่ ค.11

Targets and Node Information

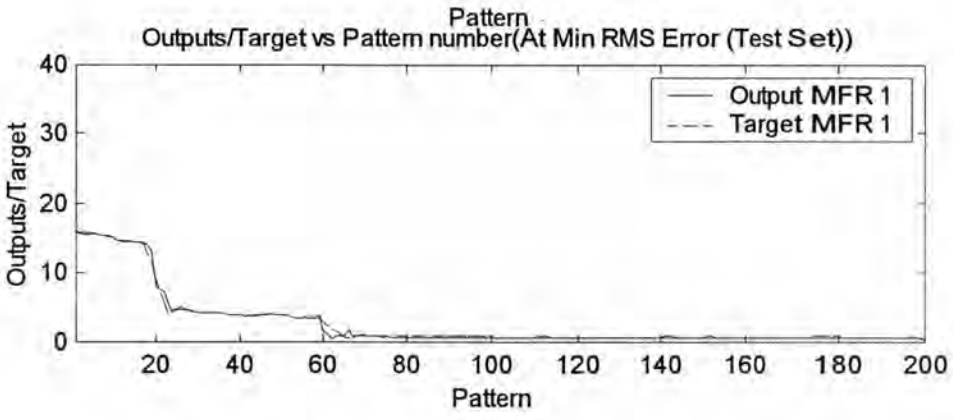
Network Name No Network
Iteration 100

Number	Node	Targets	Outputs
1	1	23.4	23.192166
2	1	21.7	20.609111
3	1	18.9	19.389832
4	1	17.1	17.711758
5	1	16.133	16.167731
6	1	15.167	14.659501
7	1	14.2	14.708861
8	1	14.4	13.850095
9	1	14.6	14.266205
10	1	13.85	14.57794
11	1	13.1	13.627268
12	1	12.925	12.982173
13	1	12.75	12.636486
14	1	12.575	12.466674
15	1	12.4	12.419397
16	1	12.375	12.253836
17	1	12.35	12.0524
18	1	12.325	12.252225

รูปที่ ค.11 แสดง Targets/Outputs Information

6.10 Outputs/Targets Min Error test

เป็นการแสดงกราฟเปรียบเทียบระหว่างค่าเอาท์พุทที่ได้จากข่ายงานกับค่าเป้าหมายที่จำนวนชุดข้อมูลต่าง ณ จำนวนรอบที่ทำให้เกิดค่าความผิดพลาดระหว่างค่าเอาท์พุทและค่าเป้าหมายต่ำที่สุดดังแสดงในรูปที่ ค.12



รูปที่ ค.12 แสดง Outputs/Targets Min Error Test

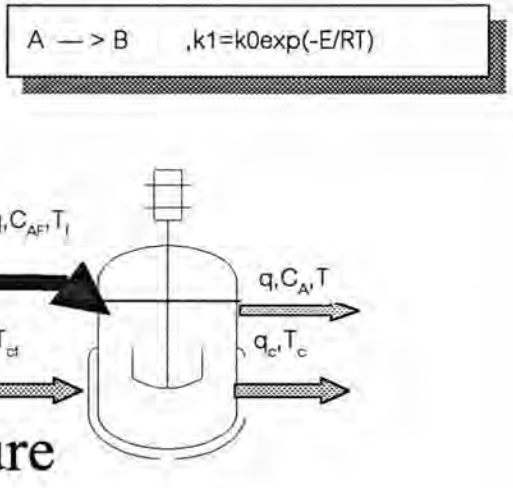
ภาคผนวก ง.

แบบจำลองทางคณิตศาสตร์ของเครื่องปฏิกรณ์ถังกวนแบบต่อเนื่องชนิดคายความร้อน

ง.1 แบบจำลองทางคณิตศาสตร์ของเครื่องปฏิกรณ์ถังกวนแบบต่อเนื่องชนิดคายความร้อน

เครื่องปฏิกรณ์ถังกวนชนิดคายร้อนอันดับหนึ่งเป็นระบบที่นำมาทดสอบโปรแกรม

แบบจำลองนี้เป็นปฏิกิริยาคายความร้อนแบบผันกลับไม่ได้อันดับหนึ่งในเครื่องปฏิกรณ์ถังกวนแบบต่อเนื่องดังแสดงในรูปที่ ง.1 เป็นดังนี้



รูปที่ ง.1 ระบบเครื่องปฏิกรณ์ถังกวนต่อเนื่อง

<u>สมมติฐาน</u>

- ความหนาแน่นของสารในระบบมีค่าคงที่
- การสูญเสียความร้อนออกจากระบบมีค่าน้อยมาก
- ปริมาตรของน้ำหล่อเย็นที่ไหลเวียน(V_c) ในขดลวดมีปริมาตรคงที่
- เกิดการผสมกันอย่างสมบูรณ์แบบภายในถังกวนและภายในขดลวด(อุณหภูมิทุกจุดมีค่าเท่ากัน) นั้นคือ อุณหภูมิภายในถังกวน, T และอุณหภูมิภายในขดลวด, T_c ในทุกๆจุดมีค่าเท่ากัน
- ความเฉื่อยเชิงความร้อน (thermal inertia) สำหรับมวลของผนังโลหะมีค่าน้อยมาก

กระบวนการนี้เป็นระบบการควบคุมความเข้มข้นของสาร A, C_A ที่แสดงการเกิดปฏิกิริยาคายความร้อนแบบไม่ย้อนกลับ (irreversible exothermic reaction) ซึ่งทำให้อุณหภูมิของระบบมีการเปลี่ยนแปลงไปตามเวลาดังแสดงในรูป ง.1 (Nahas, 1992) ปฏิกิริยาเคมีในระบบเกิดจากสารตั้งต้น A เปลี่ยนไปเป็นผลิตภัณฑ์ B ซึ่งให้ค่าความร้อนของการเกิดปฏิกิริยา (hea of reaction, ΔH) ออกมา ในการระบายความร้อนที่เกิดจากปฏิกิริยาเคมีนั้นใช้น้ำหล่อเย็นไหลผ่านขดลวด (cooling coil) ที่ขดรอบถังกวนโดยมีอัตราการไหลเชิงปริมารเป็น q_c และมีอุณหภูมิขาเข้าเป็น T_{Cf} สมการที่ใช้อธิบายแบบจำลองของระบบประกอบด้วยสมการดังต่อไปนี้

ตารางสภาวะปฏิบัติการ CSTR ที่สภาวะคงตัว

ลำดับที่	ตัวแปร	คำอธิบาย	สภาวะ	หน่วย
1	q_c	อัตราการไหลของสายป้อน	100	L/min
2	C_{Af}	ความเข้มข้นเริ่มต้นของสาร A	1	Mol/l
3	T_f	อุณหภูมิของสายป้อน	350	K
4	q	อัตราการไหลของสายผลิตภัณฑ์	100	L/min
5	C_A	ความเข้มข้นของผลิตภัณฑ์	0.0836	Mol/l
6	T	อุณหภูมิของระบบ	440.2	K
7	q_c	อัตราการไหลของน้ำหล่อเย็น	103.41	L/min
8	T_{cf}	อุณหภูมิของน้ำหล่อเย็นขาเข้า	350	K
9	T_c	อุณหภูมิของน้ำหล่อเย็นขาออก	แปรผัน	K
10	V	ปริมาตรถังกวน	100	L
11	ρ	ความหนาแน่นของสารในถังกวน	1000	g/l
12	ρ_c	ความหนาแน่นของน้ำหล่อเย็น	1000	g/l
13	UA	สัมประสิทธิ์การถ่ายเทความร้อนรวม	700000	Cal/min.K
14	K_0	สัมประสิทธิ์อัตราการเกิดปฏิกิริยา	$7.2*10^{10}$	Min^{-1}
15	E/R	พลังงานกระตุ้นต่อค่าคงที่ของก๊าซ	9950	K
16	$-\Delta H$	ความร้อนจากการเกิดปฏิกิริยา	200000	Cal/min

สมการอนุรักษ์มวล

$$\frac{d(\rho V)}{dt} = \rho_c q_c - \rho q$$
$$\frac{d(V)}{dt} = q_c - q$$
$$\frac{Adh}{dt} = q_c - q \qquad \text{...(ง.1)}$$

ถ้าอัตราการไหลในสายป้อนกลับผลิตภัณฑ์มีค่าเท่ากันแล้ว (q_c=q) มวลในถังกวนจะอยู่ในระดับคงที่

สมการอนุรักษ์มวลย่อย

$$\frac{d(C_A V)}{dt} = q_c C_{Af} - qC_A - rV$$
$$\frac{VdC_A}{dt} + \frac{C_A dV}{dt} = q_c C_{Af} - qC_A - rV$$
$$\frac{VdC_A}{dt} + C_A(q_c - q) = q_c C_{Af} - qC_A - rV$$
$$\frac{VdC_A}{dt} = q_c(C_{Af} - C_A) - rV$$
$$\frac{dC_A}{dt} = \frac{q_c}{V}(C_{Af} - C_A) - r \qquad \text{...(ง.2)}$$

โดยที่อัตราการเกิดปฏิกิริยาเคมีต่อหนึ่งหน่วยปริมาตร คือ

$$r = kC_A \qquad \text{...(ง.3)}$$

และสัมประสิทธิ์อัตราการเกิดปฏิกิริยาเป็นไปตามความสัมพันธ์ของอารีเนียส (Arrehenius reaction)

$$k = k0\ \exp(-E/RT) \qquad \text{...(ง.4)}$$

โดยมี k_0 = แฟกเตอร์การชน(frequency factor), E คือพลังงานกระตุ้น (activation energy), และ R เป็นค่าคงที่ของก๊าซ (gas constant)

สมการอนุรักษ์พลังงานในถังกวน

ในที่นี้พลังงานความร้อนที่เกิดจากการกวนจะไม่นำมาพิจารณา

$$\rho C_p V\frac{dT}{dt} = (q_c\rho C_p T_f - q\rho C_p T) + -\Delta HrV - UA(T - T_c)$$

$$\frac{d(VT)}{dt} = (q_c T_f - qT) + -\frac{\Delta HkC_A V}{\rho C_p} - \frac{UA}{\rho C_p}(T - T_c) \qquad ...(ง.5)$$

โดยที่ ΔH คือความร้อนของปฏิกิริยา (heat of reaction), และ U คือสัมประสิทธิ์การถ่ายเทความร้อนทั้งหมด (overall heat transfer coefficient)

สมการสมดุลพลังงานในขดลวด (coil energy equation)

$$\frac{dT}{ct} = \frac{q}{V}(T_{c'} - T) + \frac{-\Delta Hk_0 C_a e^{\frac{E}{RT}}}{\rho C_p} - \frac{UA}{\rho C_p V}(T - T_c) \qquad ..(ง.6)$$

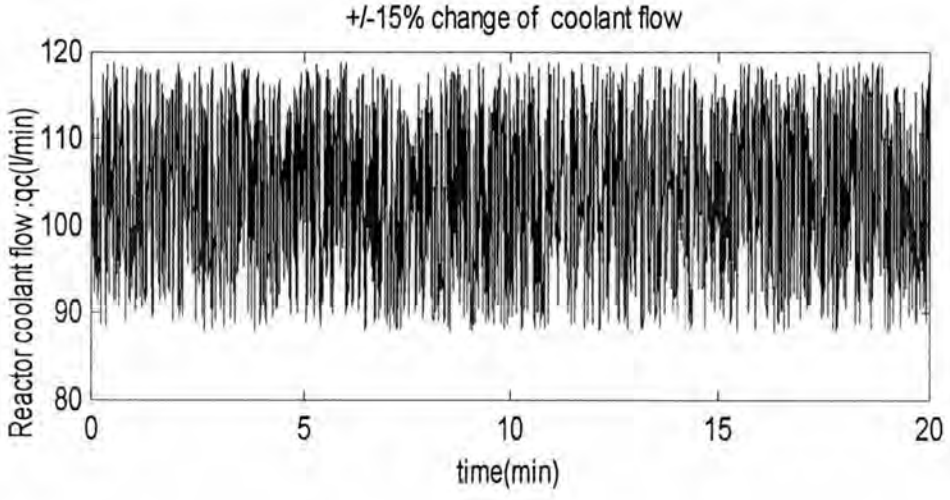
อัตราการถ่ายเทความร้อนของกระบวนการในถังกวนที่อุณหภูมิ T และน้ำหล่อเย็นที่อุณหภูมิ Tc แสดงดังสมการ

$$Q = A(T\text{-}Tc) \qquad ...(ง.7)$$

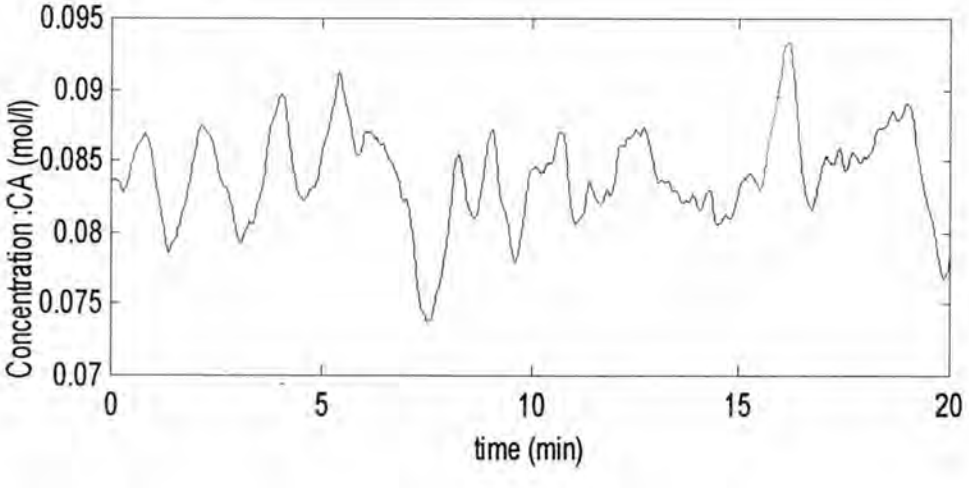
เมื่อ Q คืออัตราการถ่ายเทความร้อน และ A คือพื้นที่การถ่ายเทความร้อน

ดังนั้นดังแบบจำลองคณิตศาสตร์ข้างต้นจะนำไปจำลองเพื่อหาค่าอินพุท และเอาท์พุทของกระบวนการ แล้วนำค่าอินพุทและเอาท์พุทนี้ไปให้ข่ายงานนิวรอนได้เรียนรู้เพื่อให้ข่ายงานสร้างแบบจำลองขึ้นมาทดสอบการทำงานของโปรแกรม

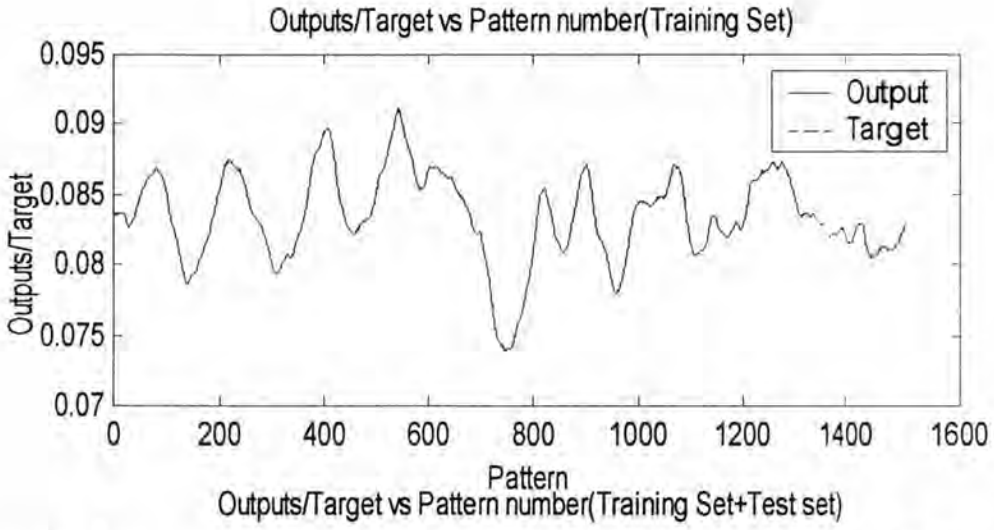
ง.2 ผลการทดสอบแบบจำลองถังกวนต่อเนื่องชนิดคายความร้อนเพื่อหาความเข้มข้นของผลิตภัณฑ์ (แบบจำลองที่ใช้ในการทดสอบโปรแกรม)



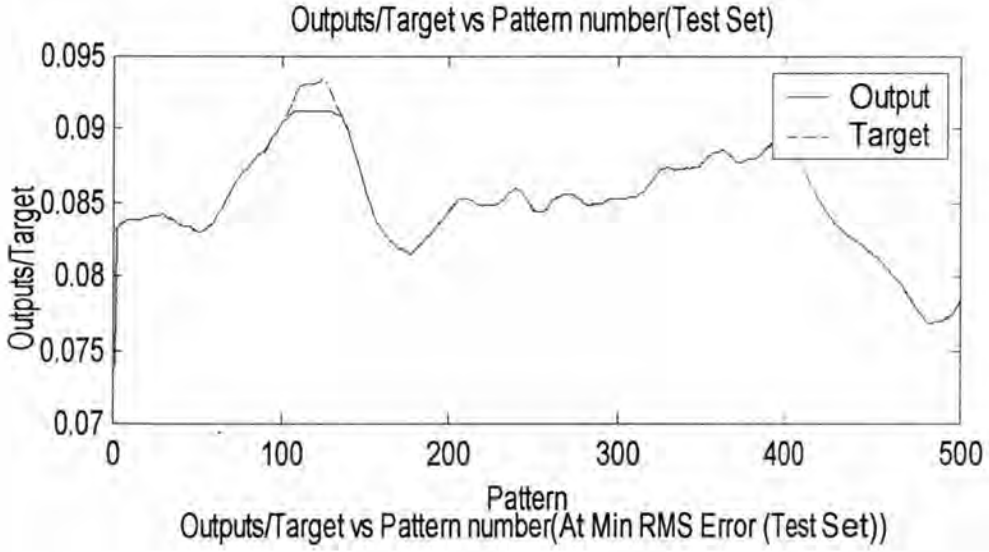
รูปที่ ง.2 การเปลี่ยนแปลงของอัตราการไหลของน้ำหล่อเย็น +/- 15% ของค่าที่สภาวะคงตัว



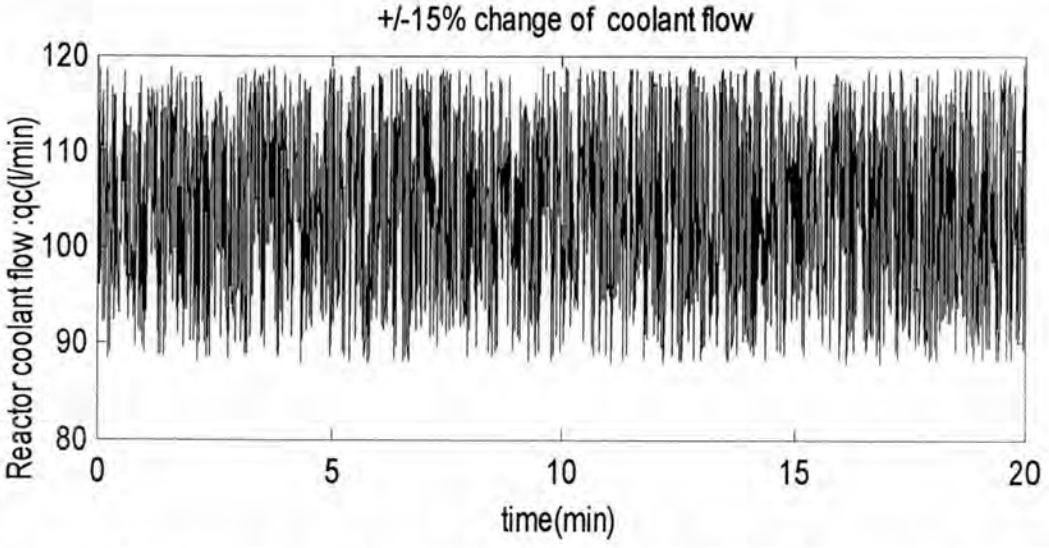
รูปที่ ง.3 ความเข้มข้นของผลิตภัณฑ์หลังจากทำการเปลี่ยนแปลงอัตราการไหลของน้ำหล่อเย็น +/- 15%



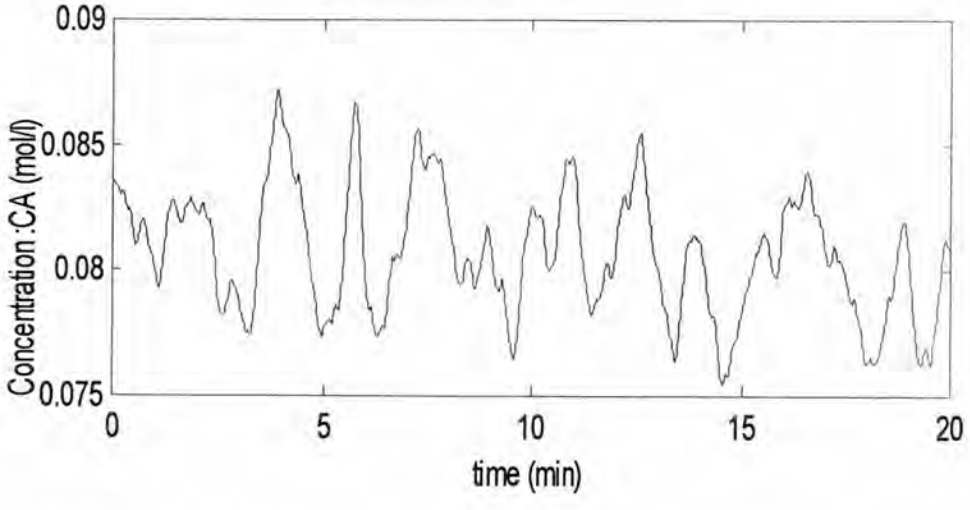
รูปที่ ง.4 การเรียนรู้กระบวนการทำนายค่าความเข้มข้นของในเครื่องปฏิกรณ์ถังกวนแบบต่อเนื่องโดยให้มีอัตราการไหล ของน้ำหล่อเย็น +/- 15% ของค่าที่สภาวะคงตัว (โครงสร้าง 6-5-1 , ฟังก์ชันกระตุ้น: ซิกมอยด์ , จำนวนรอบ 1000 รอบ, ความผิดพลาดของชุดฝึกข่ายงาน : 1.0209e-10)



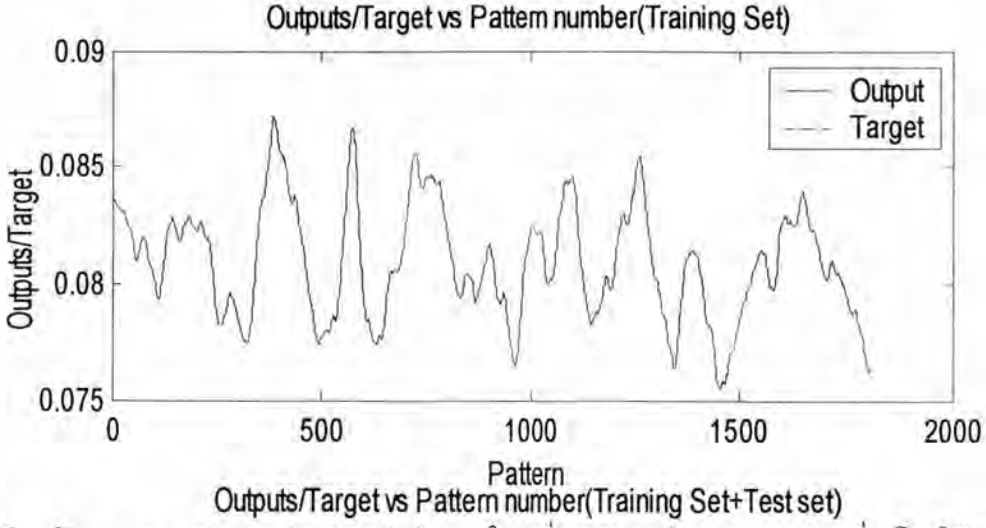
รูปที่ ง.5 การทดสอบกระบวนการทำนายค่าความเข้มข้นของในเครื่องปฏิกรณ์ถังกวนแบบต่อเนื่องโดยให้มีอัตราการไหลของน้ำหล่อเย็น +/- 15% ของค่าที่สภาวะคงตัว (โครงสร้าง 6-5-1 , ฟังก์ชันกระตุ้น: ซิกมอยด์ , จำนวนรอบ : 1000 รอบ, ความผิดพลาดของชุดทดสอบที่ต่ำที่สุด 5.04673e-11 ที่จำนวนรอบที่ 1000)



รูปที่ ง.6 การเปลี่ยนแปลงของอัตราการไหลของน้ำหล่อเย็น +/- 15% ของค่าที่สภาวะคงตัวและมีการรบกวน จาก อุณหภูมิของน้ำหล่อเย็น +/- 10% ของค่าที่สภาวะคงตัว

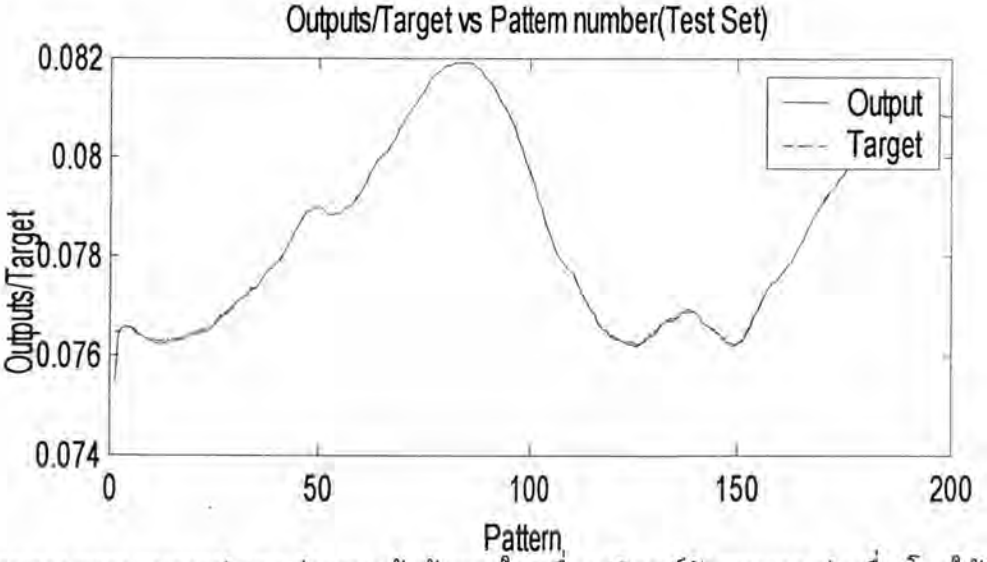


รูปที่ ง.7 ความเข้มข้นของผลิตภัณฑ์หลังจากทำการเปลี่ยนแปลงอัตราการไหลของน้ำหล่อเย็น +/- 15% และมีการรบกวนจาก อุณหภูมิของน้ำหล่อเย็น +/- 10% ของค่าที่สภาวะคงตัว



Outputs/Target vs Pattern number(Training Set+Test set)

รูปที่ ง.8 การเรียนรู้กระบวนการทำนายค่าความเข้มข้นของในเครื่องปฏิกรณ์ถังกวนแบบต่อเนื่องโดยให้มีอัตราการไหล ของน้ำหล่อเย็น +/- 15% ของค่าที่สภาวะคงตัว และมีการรบกวนจาก อุณหภูมิของน้ำหล่อเย็น +/- 10% ของค่าที่สภาวะคงตัว (โครงสร้าง 6-5-1 , ฟังก์ชันกระตุ้น: ซิกมอยด์ , จำนวนรอบ 1000 รอบ, ความผิดพลาดของชุดผึกข่ายงาน : 2.5097e-8)



รูปที่ ง.9 การทดสอบกระบวนการทำนายค่าความเข้มข้นของในเครื่องปฏิกรณ์ถังกวนแบบต่อเนื่องโดยให้มีอัตราการไหลของน้ำหล่อเย็น +/- 15% ของค่าที่สภาวะคงตัวและมีการรบกวนจาก อุณหภูมิของน้ำหล่อเย็น +/- 10% ของค่าที่สภาวะคงตัว (โครงสร้าง 6-5-1 , ฟังก์ชันกระตุ้น: ซิกมอยด์ , จำนวนรอบ : 1000 รอบ, ความผิดพลาดของชุดทดสอบที่ต่ำที่สุด 1.40939e-8 ที่จำนวนรอบที่ 1000)

จากผลการจำลองการทดสอบกระบวนการทำนายค่าความเข้มข้นของในเครื่องปฏิกรณ์ถังกวนแบบต่อเนื่องโดยให้มีอัตราการไหลของน้ำหล่อเย็น +/- 15% ของค่าที่สภาวะคงตัว (จากรูปที่ ง.4 และ ง.5) กระบวนการทำนายค่าความเข้มข้นของสาร A ในเครื่องปฏิกรณ์ถังกวนแบบต่อเนื่องโดยให้มีอัตราการไหลของน้ำหล่อเย็น +/- 15% ของค่าที่สภาวะคงตัวและถูกรบกวนจาก อุณหภูมิของน้ำหล่อเย็น +/- 10% ของค่าที่สภาวะคงตัว (รูปที่ ง.8 และ รูปที่ ง.9) จะพบว่าค่าความเข้มข้นของสาร A ที่ได้จากการทำนายโดยใช้ข่ายงานนิวรอนกับค่าความเข้มข้นของสาร A ที่ได้จากการคำนวณโดยใช้แบบจำลองทางคณิตศาสตร์ของเครื่องปฏิกรณ์ถังกวนแบบต่อเนื่องมีค่าใกล้เคียงกันทั้งชุดฝึกข่ายงานและชุดทดสอบข่ายงาน

ดังนั้นจากการนำข้อมูลจากการคำนวณในแบบจำลองทางคณิตศาสตร์ของเครื่องปฏิกรณ์ถังกวนแบบต่อเนื่องไปเป็นข้อมูลสำหรับฝึกข่ายงานเพื่อใช้ในการทดสอบการเขียนโปรแกรม ผลปรากฏว่ามีความถูกต้องสูงมาก โดยเปรียบเทียบจากค่าที่ได้ความเข้มข้นที่ได้จากการคำนวณในแบบจำลองคณิตศาสตร์และค่าความเข้มข้นที่ได้จากการทำนายจากข่ายงานนิวรอนโดยใช้โปรแกรม "Neural Network" ดังนั้นจึงสามารถนำโปรแกรมนี้ไปใช้กับข้อมูลในการวิจัยได้

ภาคผนวก จ.
สรุปฟังก์ชัน ตัวแปร และเครื่องหมาย

รวบรวมฟังก์ชัน ตัวแปรและเครื่องหมายที่ใช้ในการเขียนโปรแกรมในงานวิจัยนี้ ในลักษณะแยกกลุ่มเฉพาะงาน สำหรับรายละเอียดฟังก์ชันอื่นสามารถศึกษาได้ในหน้าต่างคำสั่ง

เครื่องหมายพิเศษ

=	กำหนดค่า
[ ]	สร้างเวกเตอร์และเมตริกซ์
...	กระทำคำสั่งยังบรรทัดต่อไป
,	แยกเอเลเมนต์ภายในเมตริกซ์
;	แยกเมตริกซ์หรือคำสั่ง หรือไม่ให้พิมพ์ค่าแสดงผลที่หน้าจอ
%	หมายเหตุ
	สร้างเวกเตอร์

ค่าเฉพาะ

NaN	ไม่ใช่จำนวน

เครื่องหมายทั่วไป

>> หรือ ป	เครื่องหมายพร็อมพ์ในหน้าต่างคำสั่ง
help	คำสั่งที่ใช้ในการอธิบายฟังก์ชันต่างๆ ในแม็ทแลบ
demo	ให้ประมวลผลการทำงานของโปรแกรมแม็ทแลบที่เป็นตัวอย่างอย่างย่อ
who	ให้บอกตัวแปรที่อยู่ในหน่วยความจำ
whos	ให้บอกตัวแปรที่อยู่ในหน่วยความจำพร้อมขนาด
what	ให้บอกเอ็มไฟล์(M-files) ที่มีอยู่ในแผ่นดิสก์ของเมตริกซ์หนึ่ง

size	บอกขนาดแถวและคอลัมน์
length	ความยาวเวกเตอร์
clear	ให้เคลียร์ พื้นที่ใช้งาน
^C	ยกเลิก , เบรค ออกจากโปรแกรมที่กำลังประมวลผลอยู่
quit	ให้หยุดโปรแกรม
exit	ออกจากแม็ทแลบ

ตัวดำเนินการทางเมตริกซ์

+	บวก
-	ลบ
*	คูณ
/	หาร
^	ยกกำลัง
'	ทรานสโพส (transpose)

ตัวดำเนินการทางสเกลาร์

+	บวก
-	ลบ
*	คูณ
/	หาร
^	ยกกำลัง
'	ทรานสโพส (transpose)

ตัวดำเนินการเปรียบเทียบและตรรกะ

<	น้อยกว่า
<=	น้อยกว่าหรือเท่ากับ

>	มากกว่า
>=	มากกว่าหรือเท่ากับ
==	เท่ากับ
~=	ไม่เท่ากับ
&	และ
~	ไม่
|	หรือ

ข้อความและชุดตัวอักษร

num2str	เปลี่ยนตัวเลขเป็นข้อความ string)
int2str	เปลี่ยนเลขจำนวนเต็มเป็นข้อความ (string)
strcmp	เปรียบเทียบตัวแปรชุดตัวอักษร

คำสั่งที่ใช้กับกราฟ

plot	พลอตกราฟเชิงเส้น x-y
title	หัวข้อของกราฟ
xlabel	คำอธิบายแกน x
ylabel	คำอธิบายแกน y
grid	วาดเส้นกริด
text	ข้อความที่วางบนตำแหน่งใดๆ
gtext	ข้อความที่วางไว้ที่ตำแหน่งเม้าส์ชี้
axis	ปรับสเกลบนแกน
hold	คงกราฟที่สร้างไว้บนหน้าจอ
shg	แสดงกราฟบนหน้าจอ
clg	เคลียร์กราฟบนหน้าจอ
subplot	แยกกราฟเป็นหลายกราฟบนหน้าจอ
print	ส่งกราฟไปยังเครื่องพิมพ์
prtsc	พิมพ์สิ่งที่ปรากฏบนหน้าจอ

คำสั่งที่ใช้ในหน้าต่างคำสั่ง

if	คำสั่งให้ปฏิบัติคำสั่งอย่างมีเงื่อนไข
elseif	ใช้กับ if
else	ใช้กับ if
end	หยุด if,for ,while
for	ให้กระทำคำสั่งเป็นจำนวน n ครั้ง
while	สร้างโครงสร้างแบบลูป
break	ออกจาก for และ while loops
return	กลับจากฟังก์ชัน
pause	หยุดชั่วขณะจนกว่าจะกดคีย์ใดๆ

ฟังก์ชันที่ใช้ในการเปรียบเทียบและตรรกะ

exist	เช็คว่ามีตัวแปรอยู่หรือไม่
isempty	ตรวจเมตริกซ์ว่าง
isstr	ตรวจตัวแปรชุดตัวอักษร

คำสั่งที่ใช้ในการจัดไฟล์ในดิสก์

delete	การลบไฟล์
dir	ไดเรกทอรีของไฟล์บนดิสก์
chdir	เปลี่ยนไดเรกทอรี่
load	โหลดตัวแปรออกจากไฟล์
save	เก็บตัวแปรบนไฟล์
type	แสดงรายการฟังก์ชันหรือไฟล์

เมตริกซ์พิเศษ

ones	เมตริกซ์คงที่หนึ่ง
zeros	เมตริกซ์ศูนย์
rand	เอเลเมนต์แบบสุ่ม

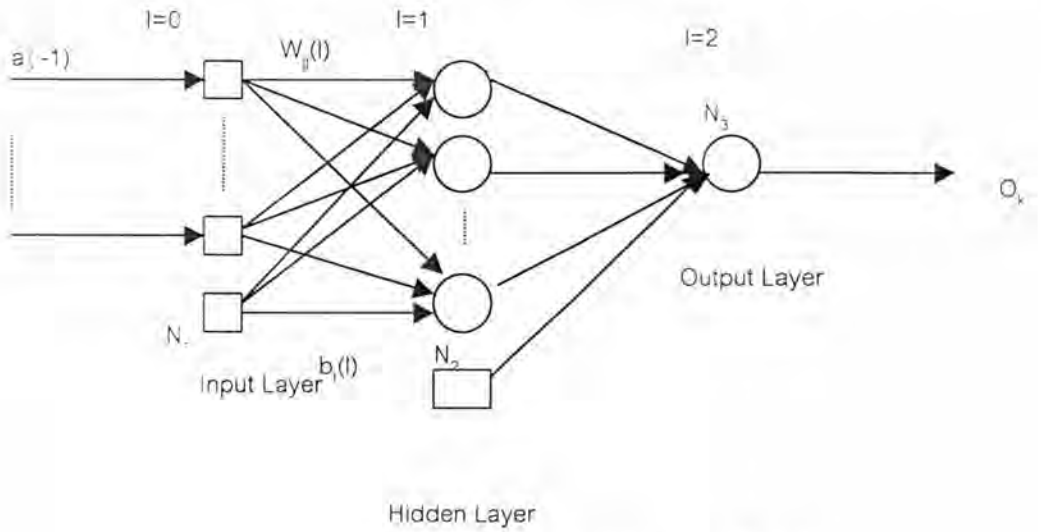
คำสั่งที่ใช้ทางสถิติ

max	ค่าสูงสุด
min	ค่าต่ำสุด
mean	ค่าเฉลี่ย
sum	ผลรวม
std	ค่าเบี่ยงเบนมาตรฐาน

สำหรับฟังก์ชันที่ใช้ในการเขียนโปรแกรมทางข่ายงานนิวรอนสำหรับการสร้างข่ายงานสามารถศึกษาได้ในภาคผนวก ก. หรือศึกษาคำสั่งเพิ่มเติมได้ใน หนังสือ Neural Network Toolbox for Use with MATLAB ,The MATHWORKS Inc.)

ภาคผนวก ฉ
อัลกอริธึมการกระจายความผิดพลาดย้อนกลับ

ข่ายงานนิวรัลแบบกระจายย้อนกลับมีฟังก์ชันมูลฐานเป็นฟังก์ชันมูลฐานเชิงเส้น และมีฟังก์ชันกระตุ้นเป็นฟังก์ชันซิกมอยด์มีสมการไดนามิกดังสมการ (ฉ.1) และ (ฉ.2) ตามลำดับ ดังแสดงในรูปที่ ฉ.1



รูปที่ ฉ.1 แสดงแบบจำลองนิวรัลในชั้นที่ l

$a_i(l-1)$ = ค่าอินพุทที่ i ของชั้นอินพุท

O_i = ค่าเอาท์ที่ต้องการในชั้นเอาท์พุทตัวที่ i

$a_i(l)$ = $f(a_i)$ = ค่าเอาท์พุทจริงของนิวรัล i

$u_i(l)$ = ค่าผลรวมของค่าน้ำหนักกับอินพุทโดยใช้การคำนวณแบบฟังก์ชันมูลฐาน

w_{ij} = ค่าน้ำหนักที่เชื่อมต่อระหว่างนิวรัล i ในชั้นที่ l-1 กับนิวรัล j ในชั้นที่ l

E = ค่าความผิดพลาด (global error)

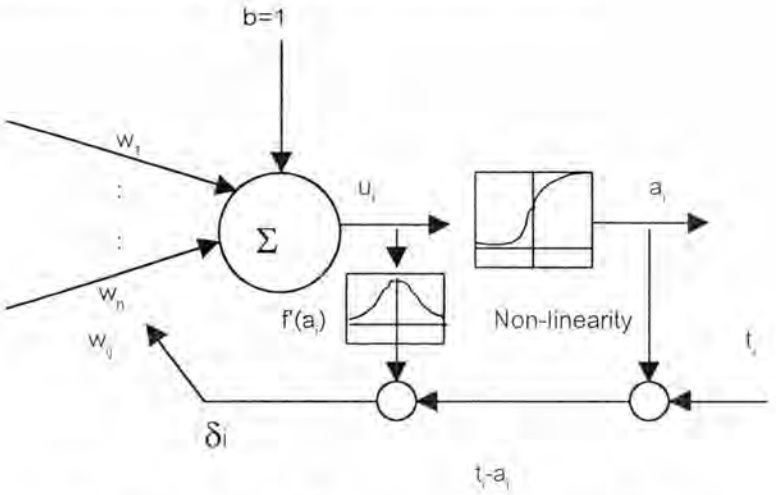
η = ค่าอัตราการเรียนรู้

$b_i(l)$ = ค่าไบอัสของชั้นที่ i

$$u_i(l) = \sum_{j=1}^{N} w_{ij}(l)a_j(l-1) + b_i(l) \qquad \text{...(ฉ.1)}$$

$$a_i(l) = f(u_i(l)) \quad 1 \le i \le N_l, \quad 1 \le l \le L \qquad \text{...(ฉ.2)}$$

การเรียนรู้ของอัลกอริธึมการกระจายความผิดพลาดย้อนกลับ



รูปที่ ฉ.2 แสดงการเรียนรู้การกระจายความผิดพลาดย้อนกลับ

วัตถุประสงค์ของอัลกอริธึมก็คือต้องการหาค่าน้ำหนักให้ถูกต้องเพื่อที่สามารถจำลองแบบกระบวนการนั้นให้สามารถทำนายค่าเอาท์พุทจากอินพุทที่รับเข้าไปได้อย่างถูกต้อง ดังนั้นอัลกอริธึมการกระจายย้อนกลับเป็นอัลกอริธึมหนึ่งที่ต้องการลดค่าผิดพลาดกำลังสองระหว่างค่าเป้าหมายกับผลลัพธ์จากข่ายงาน โดยใช้วิธีการออพติไมซ์แบบเกรเดียนท์เดสเซนท์ซึ่งเป็นวิธีการลดความผิดพลาด (optimization method) ซึ่งแสดงในสมการ ฉ.3

สำหรับข้อมูลแต่ละชุด

minimize:

$$E = \frac{1}{2}\sum_{i=1}^{N_i}\{t_i - a_i(L)\}^2 \qquad \ldots(ฉ.3)$$

subject to : $t_{max} > t_i > t_{min}$

$a_{max} > a_i(L) > a_{min}$

เมื่อ E คือค่าความผิดพลาด(global error) ที่ชั้นเอาท์พุท ได้มาจากผลรวมกำลังสองของความแตกต่างของเอาท์พุทที่ได้จากการคำนวณกับค่าเป้าหมายในชั้นเอาท์พุท

t_i คือค่าเป้าหมาย

t_{max} คือค่าสูงสุดของเป้าหมาย

t_{min} คือค่าต่ำสุดของเป้าหมาย

$a_i(L)$ คือค่าเอาท์พุทของข่ายงาน

a_{max} คือค่าสูงสุดของเอาท์พุทของข่ายงาน

a_{min} คือค่าต่ำสุดของเอาท์พุทของข่ายงาน

เกรเดียนท์เดสเซ็นท์คือการลดความผิดพลาดให้ต่ำที่สุดโดยการปรับค่าน้ำหนัก ดังนั้นถ้าทราบค่าอนุพันธ์ย่อย (partial derivative) ของค่าผิดพลาดเทียบกับค่าน้ำหนักแต่ละค่าจะทำให้รู้ทิศทางที่น้ำหนักไปในทิศทางที่ลดค่าผิดพลาด นั้นคือ $\partial E/\partial w_{ij}$ เป็นค่าเกรเดียนท์ในทิศทางลบของพื้นผิวความผิดพลาด เครื่องหมายลบแทนด้วยทิศทางเกรเดียนลดความผิดพลาดดังนั้นวิธีการปรับค่าน้ำหนักใหม่ในทางลดความผิดพลาดให้ต่ำสุด ซึ่งควรจะเป็นความผิดพลาดจริง (global error) มากกว่าที่จะเป็นความผิดพลาดเฉพาะที่ (local error) สำหรับอัลกอริธึมการกระจายย้อนกลับจะทำการปรับค่าน้ำหนัก (w_{ij}) เพื่อให้ค่า E มีค่าต่ำสุด สมการการเรียนรู้แบบเกรเดียนท์แสดงดังในสมการ ฉ.4

ค่าน้ำหนักใหม่ = ค่าน้ำหนักเดิม + ค่าน้ำหนักที่เปลี่ยนไปในทางลดความผิดพลาด

$$W_{ij}^{m+1}(l) = W_{ij}^{m}(l) + \Delta W_{ij}(l) \qquad ...(ฉ.4)$$

อนุพันธ์ของค่าผิดพลาดจะใช้วิธีการตามเทคนิคของกฎลูกโซ่ (chain rule) ของชุดของข้อมูลการฝึก (iteration) จาก $-\partial E/\partial w_{ij}$ เป็นค่าเกรเดียนท์ในทิศทางลบของพื้นผิวความผิดพลาด

$$\Delta w_{ij}^{m}(l)\alpha \; - \frac{\partial E}{\partial w_{ij}^{m}(l)} \qquad ...(ฉ.5)$$

$$\Delta w_{ij}^{m}(l) = -\eta \frac{\partial E}{\partial w_{ij}^{m}(l)}$$

$$= -\eta \frac{\partial E}{\partial a_i^{m}(l)} \frac{\partial a_i^{m}(l)}{\partial w_{ij}^{m}(l)} \qquad ...(ฉ.6)$$

จากสมการ ฉ.6 นำมาแยกพิจารณาทีละเทอมจะได้ว่า

จาก $a_i^{m}(l) = f(u_i^{m}(l))$

$$\frac{\partial a_i^{m}(l)}{\partial w_{ij}^{m}(l)} = \frac{\partial f(u_i^{m}(l))}{\partial w_{ij}^{m}(l)}$$

และจาก $u_i(l) = \sum_{j=1}^{N} w_{ij}(l) a_j(l-1) + b_i(l)$

$$\frac{\partial a_i^{m}(l)}{\partial w_{ij}^{m}(l)} = \frac{\partial f(\sum_{j=1}^{N} w_{ij}^{m}(l) a_j^{m}(l-1) + b_i(l))}{\partial w_{ij}^{m}(l)}$$

$$= f'(u_i^{m}(l) a_j^{m}(l-1)) \qquad ...(ฉ.7)$$

กำหนดให้สัญญาณความผิดพลาด, $\delta_i(l)$ เป็น

$$\delta_i^m(l) = -\frac{\partial E}{\partial a_i^m(l)} \qquad ...(ฉ.8)$$

โดย

สัญญาณความผิดพลาดสำหรับชั้นเอาท์พุท

$$\delta_i^m(l) = -\frac{\partial E}{\partial a_i^m(l)} = -\frac{\partial(\frac{1}{2}\sum_i (ti - ai(l))^2)}{\partial a_i^m(l)} = t_i - a_i(l)$$

ชั้นเอาท์พุท =ชั้นที่ L จะได้ว่า

$$\delta_i(L) = (t_i - a_i(L)) \quad ,i = 1,2,...,N_L \qquad ...(ฉ.9)$$

ดังนั้นค่าน้ำหนักที่เปลี่ยนแปลงจะได้ว่า

$$\Delta w_{ij}^m(l) = \eta \delta_i^m(l) f(u_i^m(l) a_j^m(l-1)) \qquad ...(ฉ.10)$$

สัญญาณความผิดพลาดสำหรับชั้นซ่อนแต่ละชั้น

$$\Delta w_{ij}^m(l) = -\eta \frac{\partial E}{\partial w_{ij}^m(l)}$$

$$\Delta w_{ij}^m(l) = -\eta \frac{\partial E}{\partial a_i^m(l)} \frac{\partial a_i^m(l)}{\partial u_i^m(l)} \frac{\partial u_i^m(l)}{\partial w_{ij}^m(l)}$$

$$= -\eta \frac{\partial E}{\partial a_i^m(l)} \frac{\partial a_i^m(l)}{\partial u_{ij}^m(l)} a_i^m(l-1)$$

$$\delta_i^m(l) = -\frac{\partial E}{\partial a_i^m(l)} \frac{\partial a_i^m(l)}{\partial u_i^m(l)}$$

$$= -\frac{\partial(\frac{1}{2}\sum_i (t_i - a_i(L))^2)}{\partial a_i^m(l)} f'(u_i^m(l))$$

$$= -\left[\sum_i (t_i - a_i(L) \frac{\partial(t_i - a_i(l+1)}{\partial a_i^m(l)}\right] f'(u_i^m(l))$$

$$= -\left[\sum_i (t_i - a_i(L) \frac{\partial(t_i - a_i(l+1))}{\partial u_i^m(l+1)} \frac{\partial u_i^m(l+1)}{\partial a_j^m(l)}\right] f'(u_i^m(l)$$

$$= -\left[\sum_{i}(t_i - a_i(L))\frac{\partial(t_i - f(u_i(l+1))}{\partial u_i^m(l+1)} w_{ij}(l+1)\right] f'(u_j^m(l))$$

$$= \left[\sum_{i}(t_i - a_i(l+1))f'(u_i(l+1))w_{ij}(l+1)\right] f'(u_j^m(l))$$

$$= \left[\sum_{i}\delta_i(l+1)w_{ij}(l+1)\right] f'(u_j^m(l))$$

$$\delta_j(l) = \sum_{i=1}^{N_{l+1}}(\delta_i(l+1)W_{ij}(l+1))a_j(l))(1-a_j(l)) \quad j = 1,2,\ldots,N_l$$

ดังนั้นค่าน้ำหนักที่เปลี่ยนแปลงในชั้นซ่อนจะได้ว่า

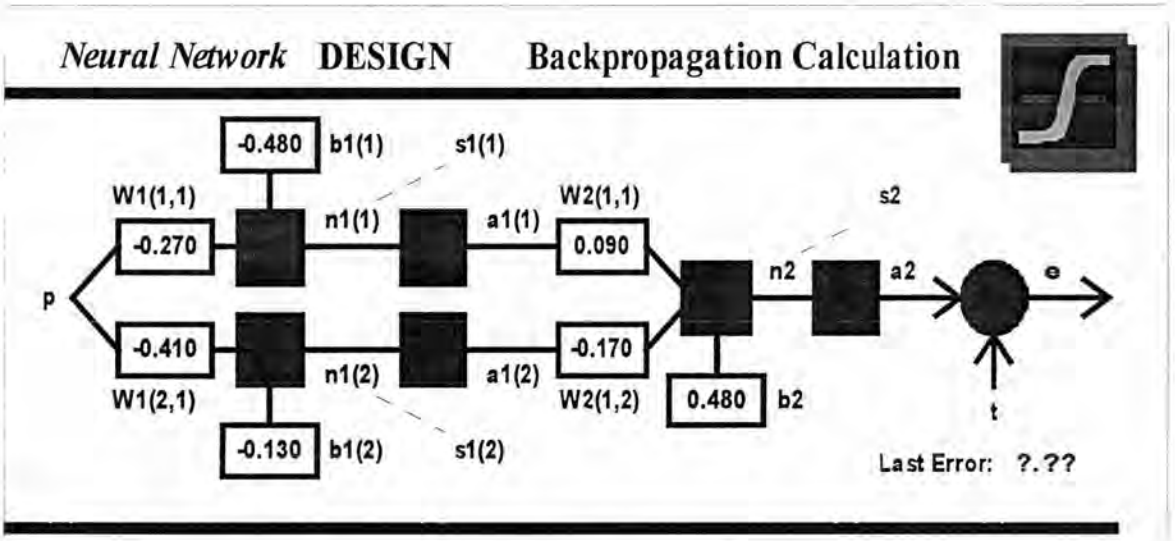
$$\Delta w_{ij}^m(l) = \eta\delta_i^m(l)f'(u_i^m(l))a_j^m(l\text{-}1) \qquad \ldots(ฉ.11)$$

ภาคผนวก ซ

ตัวอย่างวิธีการคำนวณอัลกอริธึมแบบการกระจายความผิดพลาดย้อนกลับ

ตัวอย่างที่ 3.1 วิธีการคำนวณแบบข่ายงานที่มีการกระจายความผิดพลาดย้อนกลับ

ตัวอย่างเป็นข่ายงานแบบป้อนไปข้างหน้า 2 ชั้นโดยชั้นอินพุทมี 1 อินพุท และชั้นซ่อนมี 1 ชั้น จำนวน 2 นิวรัล และชั้นเอาท์พุท 1 นิวรัล



ขั้นตอนของการเรียนรู้อัลกอริธึมการกระจายย้อนกลับสามารถคำนวณได้ดังต่อไปนี้

กำหนดชุดข้อมูลมี 1 ชุดคือ

อินพุท: $p = 5$

เป้าหมาย: $t = 1 + \sin(p*pi/4) = 0.293$

ขั้นตอนที่ 1 การกำหนดค่าเริ่มต้นสำหรับพารามิเตอร์ของเครือข่ายนิวรัล

กำหนดค่า อัตราการเรียนรู้ (learning rate): $\eta = 0.1$

ความผิดพลาดต่ำสุด : $E_{min} = 0.02$

น้ำหนักเริ่มต้น (weight) : $W1(1,1) = -0.270$

$W1(2,1) = -0.410$

$W2(1,1) = -0.090$

$W2(1,2) = -0.170$

ไบอัส : $b1(1) = -0.480$

b1(2) = -0.13

b2 = 0.480

ค่าผิดพลาดเริ่มต้น : E = 0

เริ่มการเรียนรู้ด้วยการวนรอบครั้งที่1(q=iteration) q=1 เพื่อใช้เป็นจุดเริ่มต้นในกระบวนการประมวลผล

ขั้นตอนที่ 2 กระบวนการประมวลผล

ขั้นตอนที่ 2.1 การคำนวณไปข้างหน้า (Feedforward Calculation) ดังสมการ 3.53

$$u_i(l) = \sum_{j=1}^{N} w_{ij}(l)a_j(l-1) + b_i(l)$$
$$a_i(l) = f(u_i(l)) \quad 1 \le i \le N_l, \quad 1 \le l \le L \qquad ...(3.53)$$

คำนวณฟังก์ชันมูลฐานชั้นอินพุท n1(1) = (0.270*5)+(-0.48*1) = -1.83

คำนวณฟังก์ชันกระตุ้นแบบซิกมอยด์ชั้นอินพุท a1(1) = 1/(1+e(-(-1.83))) = 0.138

คำนวณฟังก์ชันมูลฐานชั้นซ่อนที่ 1 n1(2) = (-0.410*5)+(-0.13*1) = -2.18

คำนวณฟังก์ชันกระตุ้นแบบซิกมอยด์ชั้นซ่อนที่ 1 a1(2) = 1/(1+e(-(-2.18))) = 0.102

คำนวณฟังก์ชันมูลฐานชั้นเอาท์พุท n2 = (0.09*0.138)+(-0.17*0.102) +(0.480*1) = 0.475

คำนวณฟังก์ชันกระตุ้นแบบเชิงเส้นชั้นเอาท์พุท a2 = 0.475

คำนวณค่าความผิดพลาดชั้นเอาท์พุท e = t-a2 = 0.293-0.475 = -0.182

ขั้นตอนที่ 2.2 การกระจายความผิดพลาดย้อนกลับ (Backpropagation of Error)

สัญญาณความผิดพลาดสำหรับชั้นเอาท์พุทจากสมการ 3.55

$$\delta_i(L) = (t_i - a_i(L)) \quad ,I = 1,2,...,N_L$$
$$e = -0.182$$

สัญญาณความผิดพลาดสำหรับชั้นซ่อนแต่ละชั้น จากสมการ 3.56

$$\Delta w_{ij}^{m}(l) = \eta \delta_i^{m}(l) f'(u_i^{m}(l) a_i^{m}(l-1)) \qquad ...(3.51)$$

$$\delta_i(l) = \sum_{j=1}^{N_{l+1}} (\delta_j(l+1) W_{ij}(l+1) a_i(l+1)(1 - a_i(l+1))) \quad ,i = 1,2,....N$$

s2 = -2*-0.182*0.475/0.475 = 0.365

s1(1) = (0.365*0.09*0.138*(1-0.138)) = 0.004

s1(2) = (0.365*-0.17*0.102*(1-0.102)) = -0.006

ขั้นตอนที่ 3 การปรับค่าน้ำหนักที่เชื่อมระหว่างชั้นเอาท์พุทกับชั้นซ่อน หรือชั้นอินพุทกับชั้นซ่อนดังสมการ 3.57 :

$$W_{ij}(l)^{new} = W_{ij}(l)^{old} + \eta \delta_j(l)\, a_i(l-1)$$

W1(1,1) = -0.27+0.1*0.004*5 =-0.263

b1(1) = -0.480+0.1*0.004*1 =-0.480

W1(2,1) = -0.41+0.1*-0.006*5 =-0.413

b1(2) = -0.130+0.1*-0.006*1 =-0.131

W2(1,1) = 0.09+0.1*0.365*0.138 = 0.095

W2(1,2) = -0.170+0.1*0.365*0.102 =-0.166

b1(2) = 0.480+0.1*0.365*1 = 0.517

ขั้นตอนที่ 4 ตรวจสอบจำนวนชุดข้อมูลการเรียนรู้

ข้อมูลเรียนรู้จนครบแล้ว (p=p) ให้ไปดำเนินการขั้นตอนที่ 5

ขั้นตอนที่ 5 ตรวจสอบความผิดพลาดกับค่าต่ำสุดที่ตั้งไว้

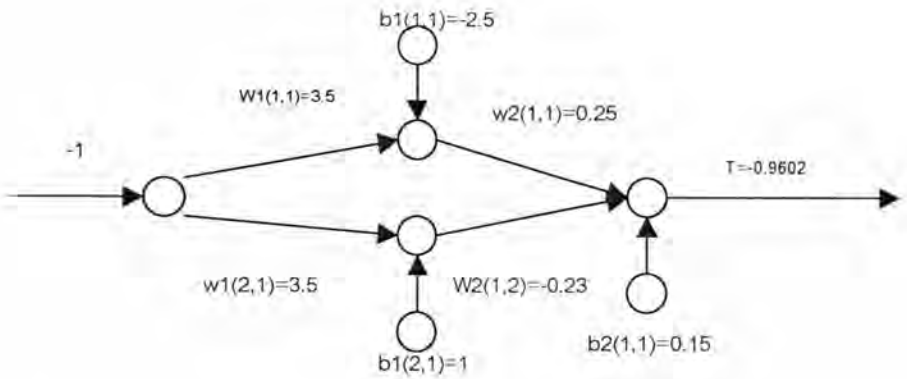
ตรวจสอบ $E < E_{min}$: -0.182 > 0.02

ถ้า $E > E_{min}$ แล้วให้ E=0 และ p=1 เริ่มวงจรการเรียนรู้ใหม่ตามขั้นตอนที่2

ถ้า $E < E_{min}$ แล้วให้หยุดการฝึกข่ายงาน

ตัวอย่างที่ 3.2 วิธีการคำนวณแบบข่ายงานที่มีการกระจายย้อนกลับแบบ Levenberg-Marquardt

ข่ายงานแบบป้อนไปข้างหน้า 2 ชั้น โดยชั้นอินพุทมี 1 อินพุท และชั้นซ่อน มี 2 นิวรัล และชั้นเอาท์พุท มี 1 เอาท์พุท



กำหนดให้ อินพุท: P = 1

เป้าหมาย : t = -0.9602

ฟังก์ชันกระตุ้นชั้นที่ 1 : ฟังก์ชันซิกมอยด์

ฟังก์ชันกระตุ้นชั้นที่ 2 : ฟังก์ชันซิกมอยด์

น้ำหนัก: w1(1,1) = 3.5000

w1(1,2) = 3.5000

b1(1,1) = -2.5000

b1(1,2) =1.0000

w2(1,1) = 0.25

w2(1,2) = -0.23

b2(1,1) = 0.15

จำนวนรอบของการฝึกข่ายงาน: me = 1

ค่าความผิดพลาดรวม (sum square error): eg = 0.02

เกรเดียนท์ต่ำสุด (gradian minimum): grad_min = 1e-6

โมเมนตัมเริ่มต้น(initial momentum) : mu_init = 1*e-3

แฟกเตอร์โมเมนตัมที่เพิ่มขึ้น (increase momentum):mu_inc = 10

แฟกเตอร์โมเมนตัมที่ลดลง (decrease momentum):mu_dec = 0.1

โมเมนตัมสูงสุด (maximum momentum): mu_max = 1e10

จากสมการที่ 3.71

$$\Delta w=\Delta\phi = N_{\phi} = -(J_s^T J_s+uI)^{-1}J_s^T e \qquad ..(3.71)$$

จาก $w(new) = w(old)+ \Delta w$

คำนวณหาค่าเอาท์พุทที่ได้จากการฝึกข่ายงานโดยใช้การคำนวณแบบป้อนไปข้างหน้า

u1(1,1) = (3.5*-1)+(-2.5*1) = -6

a1(1,1) =1/(1+e(-(-6))) = 0.0025

u1(2,1) = (3.5*-1)+ (1*1) = -2.5

a1(2,1) = 1/(1+e(-(-2.5))) = 0.0759

u2(1,1) = (0.25*0.0025)+(-0.23*0.0759)+(0.15*1) = 0.133

a2(1,1) = 1/(1+e(-0.133)) = 0.5332

e = (t-a2) = -0.9602-0.5332 = -1.4934

คำนวณหา J_s

$$js=[j1 \quad j2 \quad j3 \quad j4]=\begin{bmatrix} \frac{\partial f_1}{\partial p_1} & \frac{\partial f_1}{\partial p_2} & \frac{\partial f_1}{\partial p_3} & \frac{\partial f_1}{\partial p_4} \end{bmatrix}$$

j1 = [a1.*(1-a1).*(w2'*-(a2.(1-a2)))]' .* p'

j2 = [a1.*(1-a1).*(w2'*-(a2.(1-a2)))]'

j3 = [-(a2.*(1-a2))]' .* [a1]'

j4 =[-(a2.*(1-a2))]'

$$j1 = ((\begin{bmatrix} 0.0025 \\ 0.0759 \end{bmatrix} .* \begin{bmatrix} 1 - 0.0025 \\ 1 - 0.0759 \end{bmatrix}) .* (\begin{bmatrix} 0.25 \\ -0.23 \end{bmatrix} * [(0.5332 \ .* (1 - 0.5332 \ ))])' .* [-1]'$$

$$j1 = \begin{bmatrix} -0.0002 \\ 0.0040 \end{bmatrix}' .* [-1]' = [0.0002 \quad -0.004]$$

$$j2 = ((\begin{bmatrix} 0.0025 \\ 0.0759 \end{bmatrix} .* \begin{bmatrix} 1 - 0.0025 \\ 1 - 0.0759 \end{bmatrix}) .* (\begin{bmatrix} 0.25 \\ -0.23 \end{bmatrix} * [(0.5332 \ .* (1 - 0.5332 \ ))])'$$

$$j2 = [-0.0002 \quad 0.004]$$

$$j3 = [-(0.5332 \ .* (1 - 0.5332 \ ))]' .* \begin{bmatrix} 0.0025 \\ 0.0759 \end{bmatrix}' = -0.2489 \ .* [0.0025 \quad 0.0759]$$

$$j3 = [-0.0006 \quad -0.0189]$$

$$j4 = [-0.2489]$$

$$j = [0.0002 \quad -0.004 \quad -0.0002 \quad 0.004 \quad -0.0006 \quad -0.0189 \quad -0.2489]$$

$$\Delta w = -\frac{j'e}{(j'*j + ul)}$$

$$je = [0.0002 \quad -0.004 \quad -0.0002 \quad 0.004 \quad -0.0006 \quad -0.0189 \quad -0.2489]'[-1.4934]$$

$$je = \begin{bmatrix} -0.0002 \\ 0.0060 \\ 0.0002 \\ -0.006 \\ 0.0009 \\ 0.0282 \\ 0.3717 \end{bmatrix}$$

$$j'j = \begin{bmatrix} 0.0002 \\ -0.004 \\ -0.0002 \\ 0.004 \\ -0.0006 \\ -0.0189 \\ -0.2489 \end{bmatrix} [0.0002 \quad -0.004 \quad -0.0002 \quad 0.004 \quad -0.0006 \quad -0.0189 \quad -0.2489]$$

$u = 1e-3$

$$I = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} ; \Delta w = \begin{bmatrix} 0.0036 \\ -0.0946 \\ -0.0036 \\ 0.0946 \\ -0.0145 \\ -0.4452 \\ -5.8687 \end{bmatrix}$$

$$dw1 = \begin{bmatrix} 0.0036 \\ -0.0946 \end{bmatrix} , db1 = \begin{bmatrix} -0.0036 \\ 0.0946 \end{bmatrix} , dw2 = [0.0145 \quad -0.4452] , db2 = -5.8687$$

$$neww1 = \begin{bmatrix} 3.5 + 0.0036 \\ 3.5 - 0.0946 \end{bmatrix} \quad newb1 = \begin{bmatrix} -2.5 - 0.0036 \\ 1 + 0.0946 \end{bmatrix} , neww2 = [0.25 + 0.0145 \quad -0.23 - 0.4452] , newb2 = 0.15 - 5.8687$$

$$neww1 = \begin{bmatrix} 3.5 + 0.0036 \\ 3.5 - 0.0946 \end{bmatrix} \quad newb1 = \begin{bmatrix} -2.5 - 0.0036 \\ 1 + 0.0946 \end{bmatrix} , neww2 = [0.25 + 0.0145 \quad -0.23 - 0.4452] , newb2 = 0.15 - 5.8687$$

$$neww1 = \begin{bmatrix} 3.5036 \\ 3.4054 \end{bmatrix} \quad newb1 = \begin{bmatrix} -2.5036 \\ 1.0946 \end{bmatrix} \quad ,neww2 = \begin{bmatrix} 0.2355 & -0.6752 \end{bmatrix} \quad ,newb2 = -5.7187$$

หมายเหตุ .* , . / คือการกระทำแบบสเกลาร์นำมาคูณ/หารกัน

* คือการกระทำแบบเมตริกซ์นำมาคูณกัน

' คือเมตริกซ์ทรานสโพสท์

ประวัติผู้เขียน

นางสาวนงลักษณ์ พลรักษา เกิดเมื่อวันที่ 12 พฤศจิกายน พ.ศ.2518 สำเร็จการศึกษาในระดับชั้นมัธยมศึกษาปีที่ 5 จากโรงเรียนสตรีวิทยา 2 เมื่อ พ.ศ. 2535 สำเร็จการศึกษา ปริญญาวิศวกรรมศาสตรบัณฑิต สาขาวิศวกรรมเคมี จากมหาวิทยาลัยศรีนครินทรวิโรฒ เมื่อ พ.ศ. 2539 ปัจจุบันทำงานในตำแหน่งวิศวกรส่วนปฏิบัติการ บริษัท บางกอกโพลีเอททิลีนจำกัด (มหาชน) ตั้งแต่ปี 2539 จนถึงปัจจุบัน