PARTICLE SWARM OPTIMIZATION FOR NEURAL NETWORK LEARNING ENHANCEMENT

HAZA NUZLY BIN ABDULL HAMED

A project report submitted in partial fulfillment of
the requirements for the award of the
degree of Master of Science (Computer Science)

Faculty of Computer Science and Information System
Universiti Teknologi Malaysia

JUNE 2006

“To my beloved mother Azizah binti Jaffar and my siblings, Acik, Ajo and Nazri, thanks for your encouragement, support and understanding. To all my lecturers and friends, nice knowing you all and always remember our sweet memory”

ACKNOWLEDGEMENTS

Alhamdulillah, it is with Allah S.W.T will that finally I get to finish this project. In completing this project, I was contacting many people and they have contributed toward my understanding. First of all, I wish to express my sincere appreciation to my supervisor, Associate Professor Dr. Siti Mariyam Binti Shamsuddin for her attention and generous help throughout the length of this study. I am very thankful for her kind guidance since the first day of this study. I am also very thankful to my co-supervisor, Associate Professor Dr. Naomie Binti Salim for her advices and motivation. Not forgotten to my evaluators, Dr. Siti Zaiton binti Mohd Hashim and Dr. Norazah Bte Yusof, I am also grateful for their helpful suggestions.

Here, I would like to give my special thanks to Prof AP Engelbrecht from University of Pretoria, South Africa, for his guidance on how to apply Particle Swarm Optimization in neural network and his constructive comments on my C++ programs. Thank you to all my lecturers, staff, course mates and friends in the Faculty of Computer Science and Information System, Universiti Teknologi Malaysia for their help and support.

ABSTRACT

Backpropagation (BP) algorithm is widely used to solve many real world problems by using the concept of Multilayer Perceptron (MLP). However, major disadvantages of BP are its convergence rate is relatively slow and always being trapped at the local minima. To overcome this problem, Genetic Algorithm (GA) has been used to determine optimal value for BP parameters such as learning rate and momentum rate and also for weight optimization. In Backpropagation Neural Network (BPNN), there are many elements to be considered such as the number of input, hidden and output nodes, learning rate, momentum rate, bias, minimum error and activation/transfer functions. All these elements will affect the speed of neural network learning. Although GA is successfully improved BPNN learning, there are still some issues such as longer training time to produce the output and usage of complex functions in selection, crossover and mutation calculation. In this study, the latest optimization algorithm, Particle Swarm Optimization (PSO) is chosen and applied in feedforward neural network to enhance the learning process in terms of convergence rate and classification accuracy. Two programs have been developed; Particle Swarm Optimization Feedforward Neural Network (PSONN) and Genetic Algorithm Backpropagation Neural Network (GANN). The results show that PSONN give promising results in term of convergence rate and classification compared to GANN.

ABSTRAK

Algoritma Rambatan Balik banyak digunakan dalam menyelesaikan pelbagai masalah dengan menggunakan konsep Multilapisan Perceptron. Walau bagaimanapun, masalah utama Algoritma Rambatan Balik ialah kadar penumpuan yang lambat dan selalu terperangkap dalam minima setempat. Untuk mengatasi masalah ini, Algoritma Genetik telah digunakan untuk menentukan nilai optimal bagi Algoritma Rambatan Balik seperti kadar pembelajaran, kadar momentum serta mencari pemberat terbaik. Dalam Rangkaian Neural menggunakan Rambatan Balik, terdapat banyak elemen yang perlu dipertimbangkan seperti jumlah nod input, nod tersembunyi, nod output, kadar pembelajaran, kadar momentum, bias, ralat minimum dan fungsi penggerak. Semua elemen ini akan memberi kesan terhadap kelajuan pembelajaran Rangkaian Neural. Walaupun Algoritma Genetik berjaya meningkatkan keupayaan pembelajaran bagi Rangkaian Neural menggunakan Rambatan Balik, masih terdapat beberapa masalah seperti latihan untuk mengeluarkan output mengambil masa yang lama dan penggunaan fungsi yang komplek seperti pengiraan pemilihan, penyilangan dan mutasi. Dalam kajian ini, algoritma pengoptima yang terkini iaitu Pengoptima Partikal Berkumpulan telah dipilih dan digunakan dalam Rangkaian Neural untuk meningkatkan keupayaan proses pembelajaran dari segi masa penumpuan dan ketepatan pengkelasan. Dua program telah dibangunkan iaitu Rangkaian Neural Kehadapan menggunakan Pengoptima Partikal Berkumpulan dan Rangkaian Neural Rambatan Balik menggunakan Algoritma Genetik. Hasil kajian menunjukkan bahawa Rangkaian Neural Kehadapan menggunakan Pengoptima Partikal Berkumpulan memberikan keputusan yang lebih baik dari segi masa penumpuan dan ketepatan pengkelasan berbanding Rangkaian Neural Rambatan Balik menggunakan Algoritma Genetik.

TABLE OF CONTENT

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS

? t	-	Time interval
c_1	-	Acceleration constants for gbest
c_2	-	Acceleration constants for pbest
h	-	Learning rate
a	-	Momentum rate

LIST OF ABBREVIATIONS

2D	-	Two Dimensional
ACO	-	Ant Colony Optimization
ANN	-	Artificial Neural Network
BP	-	Backpropagation
EA	-	Evolutionary Algorithms
EP	-	Evolutionary Programming
ES	-	Evolutionary Strategies
GA	-	Genetic Algorithm
GANN	-	Genetic Algorithm Backpropagation Neural Network
GP	-	Genetic Programming
KLSE	-	Kuala Lumpur Stock Exchange
MLP	-	Multilayer Perceptron
MPPSO	-	Multi-phase Particle Swarm Optimization
MSE	-	Mean Squared Error
NN	-	Neural Network
PDP	-	Parallel Distributed Processing
PSO	-	Particle Swarm Optimization
PSONN	-	Particle Swarm Optimization Feedforward Neural Network
SI	-	Swarm Intelligence
SSE	-	Sum of Squared Error
UAV	-	Unmanned Ariel Vehicle

LIST OF APPENDICES

CHAPTER 1

INTRODUCTION

1.1 Introduction

An Artificial Neural Network (ANN) or commonly referred as Neural Network (NN) is an information processing paradigm that is inspired by the way biological nervous systems process the information. The computation is highly complex, nonlinear and parallel. Many applications have been developed using NN algorithm and most of the applications are on predicting future events based on historical data. Processing power in ANN allows the network to learn and adapt, in addition to making it particularly well suited to tasks such as classification, pattern recognition, memory recall, prediction, optimization, and noise filtering (Luger, 2002).

The primary significance for a NN is the ability of the network to learn from its environment and to improve its performance through learning (Haykin, 1999). Learning is a process of modifying the weights and biases to the neurons and continued until a preset condition is met such as defined error function. Learning process is usually referred as training process in NN. The objective of training process is to classify certain input data patterns to certain outputs before testing with another group of related data. The backpropagation (BP) algorithm is commonly used learning algorithm for training NN (Zweiri *et al*., 2003). BP algorithm is used in NN learning process for supervised or associative learning. Supervised learning learns based on the target value or the desired outputs. During training, the network tries to match the outputs with the desired target values. Other algorithm that usually use is Genetic Algorithm (GA) which is one of the famous evolutionary technique in NN learning.

With the latest research in softcomputing, Swarm Intelligence (SI) technique was introduced in 1995 by James Kennedy who is a social psychologist and Russell C. Eberhart, Associate Dean for Research, Purdue School of Engineering and Technology. SI is a bio-inspired technique and the latest an artificial intelligence technique based around the study of collective behaviour in decentralized and self-organized systems. SI is defined as any attempt to design algorithms or distributed problem-solving devices inspired by the collective behaviour of the social insect colonies and other animal societies (Bonabeau *et al*., 1999). The idea of SI came from systems that can be found in nature, including ant colonies, bird flocking and animal herding that can be effectively applied to computationally intelligent system. SI systems are typically made up from a population of agents interacting locally with one another and with their environment and local interactions between such nodes often lead to the emergence of global behaviour. There are two major techniques in SI which are Ant Colony Optimization (ACO) and Particle Swarm Optimization (PSO). The ACO algorithm is a probabilistic technique for solving computational problems which can be reduced to finding good paths through graphs. They are inspired by the behaviour of ants in finding paths from the colony to food. While PSO is a technique where several particles (solutions) interacting between each other to find the best solutions. In this study, PSO was chosen as experiment learning algorithm in NN.

1.2 Problem Background

The most familiar technique in NN learning is called Backpropogation (BP) algorithm. BP is widely used to solve many real world problems by using the concept of Multilayer Perceptron (MLP) training and testing. However, the major disadvantages of BP are its convergence rate relatively slow (Zweiri *et al.*, 2003) and being trapped at the local minima. Since BP learning is basically a hill climbing technique, it runs the risk of being trapped in local minima where every small change in synaptic weight increases the cost function. But somewhere else in the weight space there exist another set of synaptic weight for which the cost function is smaller than the local minimum in which the network is stuck. It is clearly undesirable to have the learning process terminate at a local minimum. There are many solutions proposed by many NN researcher to overcome the slow converge rate problem. Many powerful optimization algorithms have been devised, most of which have been based on simple gradient descent algorithm as explain by C.M. Bishop (1995) such as conjugate gradient decent, scaled conjugate gradient descent, quasi-Newton BFGS and Levenberg-Marquardt methods. The classical solutions are by improving the program codes and upgrading the machine's hardware. Lately, latest solutions proposed by NN researcher try to guide the learning so that the converge speed become faster. The guidelines to select better functions, learning rate, momentum rate and activation functions. Genetic Algorithm (GA) is one of the algorithms proposed to determine the learning rate and momentum rate and will produce a set of weight that can be used for testing related data. Table 1 briefly described the finding from several researchers in order to increase learning speed (Fnaiech *et al.,* 2002), avoid from trapped into local minima (Wyeth *et al.,* 2000) and better classification result.

Table 1.1: Approaches for increasing the learning speed in NN (Fnaiech *et al*., 2002).

Problems	**Cases**	**Finding**
Local minima	Choosing better parameters	There are several parameters that can be adjusted to improve training convergence. The two most relevant

		are momentum and learning rate. If the error is considered to lie over a multi-dimension energy surface, learning rate is considered to designate the step sizes across this surface to reach a global minimum. Momentum, on the other hand, tries to push the process through any local minima. A learning rate too large result in the global minima being stepped across while too small will cause the convergence time is prohibitively large. With the momentum, too large a value sees the process oscillate, while too small can result in local minima entrapment. By using better optimization algorithm such as GA, it avoid from trap into local minima
	The weight updating procedure	Distinguish the online and batch method where the weight changes are accumulated over some number of learning examples before the weights are actually changes.
	The choice of the optimization criterion	A modified form of the optimization error using combination of linear and nonlinear errors can decrease the learning iteration number and also learning time. More sophisticated error measures can be use in order to achieve a better NN learning.

	The use of adaptive parameters	The use of an adaptive slope of the activation function or global adaptation of the learning rate and/or momentum rate can increase the convergence speed in some applications.
	Estimation of optimal initial conditions	Network always start with random initial weight values. Finding initial weight that is better starting values than pure random-values can considerably improve the convergence speed.
	Reducing the size of the problem	By pre-processing of the data. For example by employing future extraction algorithms or the projection.
	Estimation of optimal NN structure	Usually the NN structure is evaluated by trials and error approaches. Starting with optimal NN structure for example the optimal number of the hidden layers and their corresponding number of neuron is a very helpful task in the speeding process
	Application of more advanced algorithms	Several heuristic optimization algorithms have been proposed to improve the convergence speed but unfortunately, some of these algorithms are computationally very expensive and required large storage
Classification result	Define good network architecture	In ANN, there are many elements to be considered such as number of

		input, hidden and output node, learning rate, momentum rate, bias, minimum error, activation/transfer function and optimization algorithm. All of these elements will affect the learning and classification results.

According to Song *et al.* (2004), because of the convenience of realization and promising optimization ability in various problems, PSO algorithm has been paid more and more attention to by researchers. Lee *et al.* (2005) have used PSO and GA for excess return evaluation in stock market. Based on their experiment, it is proven that PSO algorithm is better compared to GA. PSO can reach the global optimum value with less iteration, keep equilibrium versus GA and shows the possibility to solve the complicated problem using only basic equation without crossover, mutation and other manipulation as in GA. The application for stock trading using PSO also has been done by Nenortaite *et al.* (2004) where it shows good profit accumulation results. Another study by Zhang *et al.* (2000) applied two real problems in medical domain which are breast cancer and heart disease to feed-forward ANN with PSO called Particle Swarm Optimization Feed-forward Neural Network (PSONN). The result shows that PSONN has better accuracy in classified data compared to other algorithms. Al-kazemi *et al.* (2002) was conducted a study on Multi-phase Particle Swarm Optimization (MPPSO) where it evolves multiple groups of particle. Based on the previous researcher works, it gives a good credit to PSO. This study is conducted to prove the effectiveness PSO-based neural network and compared to GA-based neural network based on several universal data for classification problem.

1.3 Problem Statement

In BP, there are many elements to be considered such as the number of input, hidden and output nodes, learning rate, momentum rate, bias, minimum error and activation/transfer function. All these elements will affect the convergence of NN learning. As mentioned before, GA can be used to determine some parameters and provide the best pattern of weight in order to enhance the BP learning. In this study, the Swarm Intelligence technique called Particle Swarm Optimization is employed to see the convergence speed and the classification accuracy of feedforward neural network learning. In order to evaluate the performance, two programs called Particle Swarm Optimization Feedforward Neural Network (PSONN) and Genetic Algorithm Backpropagation Neural Network (GANN) have been developed.

The hypothesis of this study can be stated as:

How efficient is the PSO algorithm for neural network learning enhancement compared to GA?

1.4 Project Aim

This project aims to determine the efficiency of PSO that applied in NN compared to GA-based neural network in term of convergence rate and correct classification. Three datasets are used to validate the above algorithm.

1.5 Objectives

Few objectives have been identified in this study:

a) To develop and apply PSO algorithm in NN.
b) To analyze the effectiveness of the PSO in NN learning.

c) To compare the results between PSONN and GANN in terms of convergence rate and classification result.

1.6 Project Scope

This project is focusing on PSO technique to enhance neural network learning. The scopes of this project are as follows:

a) Three dataset which are XOR, Cancer and Iris have been used to get the results for both algorithms.
b) The PSO program has been developed and applied to feedforward neural network using Microsoft Visual C++ 6.0.
c) The GA program has been developed and applied to backpropagation neural network using Microsoft Visual C++ 6.0.

1.7 Significance of Project

The performance between PSO-based neural network and GA-based neural network is analysed, thus we can determine which method is better for neural network learning. This is important to identify the suitable technique for future study and can be implemented in real world application.

1.8 Organization of Report

This report consists of five chapters. Chapter 1 presents the introduction of the study, problems background, the hypothesis, objectives and project scope. Chapter 2 gives literature reviews on the NN, PSO, and GA. Project methodology is discussed in Chapter 3 and Chapter 4 discusses the experimental results. The conclusion and suggestions for future work are explained in Chapter 5.

REFERENCES

Al-kazemi, B. and Mohan, C.K. (2002). Training Feedforward Neural Network Using Multi-phase Particle Swarm Optimization. *Proceeding of the 9th International Conference on Neural Information Processing*. New York.

Ben Kröse, Patrick van der Smagt (1996). *An Introduction to Neural Networks*. The University of Amsterdam.

Bonabeau, E., Dorigo, M and Theraulaz, G. (1999). *Swarm Intelligence: From Natural to Artificial System*. Oxford University Press. New York.

Charytoniuk, W. and Chen, M.S (2000). Neural Network Design for Short-term Load Forecasting. *International Conference on Electric Utility Deregulation and Restructuring and Power Technologies 2000*. 4-7 April 2000. City University, London, 554-561.

Cho, T-H., Conners, R.w. and Araman, P.A. (1991). Fast Back-Propagation Learning Using Steep Activation Functions and Automatic Weight Reinitialization. *Proceeding of Decision Aiding For Complex Systems Conference*. 13-16 October 1991. Charlottesville, VA USA: IEE, 1587-1592.

C.M. Bishop (1995). *Neural Networks for Pattern Recognition*. Chapter 7, pp.253-294. Oxford University Press.

Eberhart, R. and Shi, Y. (2001). Particle Swarm Optimization: Developments, Application and Resources. *IEEE*: 81-86.

Ferguson, D. (2004). *Particle Swarm*. University of Victoria, Canada.

Fnaiech, F., Abid, S., Ellala, N. and Cheriet, M. (2002). A Comparative Study of Fast Neural Network Learning Algorithms. *IEEE International Conference on Systems, Man and Cybernetics 2002. IEEE SMC*. 24-29.

Haykin, S. (1999). *Neural Network. A Comprehensive Foundation.* 2nd Ed: Prentice Hall.

Jones M.T (2005). *AI Application Programming*. 2nd Ed. Hingham, Massachusetts: Charles River Media Inc.

Kennedy, J. and Eberhart, R (2001). *Swarm Intelligence*. Inc., San Francisco, CA: Morgan Kaufmann Publishers Inc.

Kim, G-H., Yoon, J-E., An, S-H., Cho, H-H. and Kang, K-I. (2004). Neural Network Model Incorporating A Genetic Algorithm in Estimating Construction Cost. Building and Environment. 39(11): 1333-1340.

Lee, J. S., Lee, S., Chang, S. and Ahn, B. H. (2005). A Comparison of GA and PSO for Excess Return Evaluation in Stock Markets. *Springer-Verlag*: 221-230.

Luger, George F., (2002). *Artificial Intelligence: Structures and Strategies for Complex Problem Solving*. 4th ed.. Addison-Wesley/Pearson Education Limited, Harlow, England. 417-473.

Mohankrishnan, N., Lee, W.S. and Paulik, M.J.(1996). Multi-layer neural network classification of on-line signatures. *IEEE 39th Midwest symposium on Volume 2*, 18-21 August 1996: 831 - 834.

Nenortaite, J. and Simutis, R. (2004). Stock Trading System Based on the Particle Swarm Optimization Algorithm. *Springer-Verlag*: 843-850.

Randall, S. S. and Naheel, A. S. (2001). Data Mining Using a Genetic Algorithm-Trained Neural Network. *International Journal of Intelligent Systems in Accounting, Finance & Management*.10, 201–210.

Rumelhart, D.E. and McClelland, J.L. (1986). *Parallel Distributed Processing: Explorations in The Microstructure of Cognition*. Vol 1. MIT press, Cambridge, MA.

Saeed Khan, M.K., Al-Khatib, W.G. and Moinuddin, M. (2004). Automatic classification of speech and music using neural networks. *Proceedings of the 2nd ACM international workshop on Multimedia databases*. Washington DC, USA: 94 - 99.

Shi, Y. (2004). Particle Swarm Optimization. *IEEE Neural Network Society:* 8-13.

Siti Mariyam Hj. Shamsuddin (2004). *Lecture Note Advanced Artificial Intelligence: Number of Hidden Neurons*. Universiti Teknologi Malaysia. Unpublished.

Song, M. P. and Gu, G.C. (2004). Research on Particle Swarm Optimization: A Review. *IEEE*: 2236-2241.

Van den Bergh, F. (1999). Particle Swarm Weight Initialization In Multi-Layer Perceptron Artificial Neural Networks. *Accepted for ICAI.* Durban, South Africa.

Van den Bergh, F. and Engelbrecht, A. P. (2000). Cooperative Learning in Neural Networks using Particle Swarm Optimizers. *South African Computer Journal.* (26):84-90.

Wu C.H., Epidemiology A. and Chang T.C. (1991). Protein classification using a neural network database system. *Proceedings of the conference on Analysis of neural network applications*. Fairfax, Virginia, United States: 29 - 41

Wyeth, G., Buskey, G. and Roberts, J. (2000). *Flight Control Using an Artificial Neural Network*. University of Queensland.

Yao, X. (1993). A review of evolutionary artificial neural networks. *International Journal of Intelligent Systems*. 4: 203--222.

Yao, X. (1999). Evolving artificial neural networks. *Proceedings of the IEEE*. vol. 87, no. 9: 1423 -1447.

Zhang, C., Shao, H. and Li, Y. (2000). Particle Swarm Optimization for Evolving Artificial Neural Network. *IEEE*: 2487-2490.

Zweiri, Y.H., Whidborne, J.F. and Sceviratne, L.D. (2002). A Three-term Backpropagation Algorithm. *Neurocomputing*. 50: 305-318.