THE IMPACT OF VMAX ACTIVATION FUNCTION IN PARTICLE SWARM OPTIMIZATION NEURAL NETWORK

LEE YIEW SIANG

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

Faculty of Computer Science and Information Systems
Universiti Teknologi Malaysia

JUNE 2008

ABSTRACT

Back propagation (BP) Network is the most common technique in Artificial Neural Network (ANN) learning. However, major disadvantages of BP are its convergence rate is relatively slow and always being trapped at the local minima. Therefore, latest optimization technique, Particle Swarm Optimization (PSO) is chosen and applied in feed forward neural network to enhance the network learning. In conventional PSO, maximum velocity, *Vmax* serves as a constraint that controls the maximum global exploration ability PSO can have. By setting a too small maximum velocity, maximum global exploration ability is limited and PSO will always favor a local search no matter what the inertia weight is. By setting a large maximum velocity, PSO can have a large range of exploration ability. Therefore, in this study, different activation functions will apply in the PSO *Vmax* function in order to control global exploration of particles and increase the convergence rate as well as correct classification. The preliminary results show that *Vmax* hyperbolic tangent function give promising results in term of convergence rate and classification compared to *Vmax* sigmoid function and standard *Vmax* function.

ABSTRAK

Kaedah Rambatan Balik banyak digunakan dalam menyelesaikan pelbagai masalah dengan menggunakan konsep Multilapisan Perceptron. Walaubagaimanapun, masalah utama Algoritma Rambatan Balik ialah kadar penumpuan yang lambat dan selalu terperangkap dalam minima setempat. Jadi, algoritma pengoptima yang terkini iaitu Pengoptima Partikal Berkumpulan telah dipilih dan digunakan dalam Rangkaian Neural untuk meningkatkan keupayaan proses pembelajaran. Maksimum pergerakan, *Vmax* berfungsi sebagai faktor penting untuk mengehadkan pergerakan partikal. Dengan menetapkan nilai maksimum pergerakan yang kecil, maksimum pergerakan partikal adalah dihadkan dan Pengoptima Partikal Berkumpulan cenderung membuat carian di tempat sepusat tanpa mengira nilai awal pemberat. Manakala dengan menetapkan nilai maksimum pergerakan yang besar, Pengoptima Partikal Berkumpulan dapat membuat carian di tempat yang lebih luas. Jadi, dalam kajian ini, pelbagai jenis fungsi akan digunakan bersama dengan maksimum pergerakan untuk meningkatkan keputusan dari segi masa penumpuan dan ketepatan pengkelasan. Hasil kajian awalan menunjukkan Rangkaian Neural Kehadapan dengan menggunakan Pengoptima Partikal Berkumpulan dan fungsi maksimum pergerakan tangent memberikan keputusan yang lebih baik dari segi masa penumpuan dan ketepatan pengkelasan berbanding dengan fungsi maksimum pergerakan sigmoid dan fungsi maksimum pergerakan.

TABLE OF CONTENT

CHAPTER 1

INTRODUCTION

1.1 Introduction

An artificial neural network consists of a number of very simple and highly interconnected processors, also called neurons which are analogous to the biological neurons in the brain. The neurons are connected by weighted links passing signals from one neuron to other. Each neuron receives a number of input signals through its connection and weights are the basic means of long term memory in ANNs [1,2]. ANN like people, learns by example and is configured for a specific application such as classification, pattern matching, pattern recognition, function approximation or data mining through a learning process [3].

Back propagation (BP) is the most widely used algorithm for training multilayer feed forward neural network. The algorithm uses gradient descent technique to adjust the connection weights between neurons in order to minimize the system error between the actual output and desired target output [2]. One of the major drawbacks of back propagation learning is its slow convergence [3].

An activation function is a nonlinear function that, when applied to the net input of a neuron, determines the output of that neuron. Its domain must generally be all real numbers, as there is no theoretical limit to what the input net can be. The range of the activation function (values it can output) is usually limited. The most common limits are 0 to 1, while some range from –1 to 1 [4].

Activation function plays an important role in Multilayer Perceptrons (MLP). Not only determining the decision borders, but the value of the activation function also determines the total signal strength the neuron will produce and receive. In turn, it will affect almost all aspects of solving the problem in hand like the quality of the network initial state, speed of conversion and the efficiency of the synaptic weights updates. As a result, a careful selection of the activation function has a huge impact on the MLP classification performance [5]. Table 1.1 shows different activation functions in various network types and their corresponding input and output relation [6].

Table 1.1: Different Activation Functions in Various Network Types

Name	Input/Output relation	Network type
Hard limiting	$f(x) = 0 \quad \text{if } x < 0$ $= 1 \quad \text{if } x \geq 0$	Back propagation
Symmetrical hard limiting	$f(x) = 0 \quad \text{if } x < 0$ $= 1 \quad \text{if } x \geq 0$	Perceptron
Linear	$f(x) = x$	ADALINE
Log-sigmoid	$f(x) = \dfrac{1}{1+e^{-x}}$	Back propagation, RBF
Hyperbolic tangent	$f(x) = tanh(x)$ $= \dfrac{e^{x} - e^{-x}}{e^{x} + e^{-x}}$	Back propagation
Logarithmic	$f(x) = \begin{cases} \ln(x+1) & \text{if } x \geq 0 \\ -\ln(-x+1) & \text{if } x < 0 \end{cases}$	Back propagation
Sigmoid Positive Linear	$f(x) = 0, \quad \text{if } x < 0$ $= x, \quad \text{if } x \geq 0$	Back propagation
Algebraic sigmoid	$f(x) = \dfrac{x}{\sqrt{1+x^2}}$	Back propagation
Competitive	$f(x) = 1$, neuron with maximum x $= 0$, for all other neurons	LVQ

Recently, swarm intelligence particularly particle swarm optimization (PSO) has been introduced to enhance the BP network [3,7,8]. The particle swarm optimization (PSO) is an evolutionary computation technique developed by Eberhart and Kennedy [3,7,8], inspired by social behavior of bird flocking. Similar to the genetic algorithm (GA), the PSO algorithm is an optimization tool based on population, and the system is initialized with a population of random solutions and can search for optima by the updating of generations. In the PSO algorithm, the potential solutions, called as particles, are obtained by ‘‘flowing’’ through the problem space by following the current optimum particle [3,7,8]. PSO has been successfully applied in many areas such as function optimization, artificial neural network training, fuzzy system control, and other areas [7,8].

1.2 Problem Background

There are many research have been done using different network structures and error functions to enhance the BP learning. However, in this section, we will cover on three issues. The first part will discuss activation functions, second part will discuss Particle Swarm Optimization (PSO) and third part will discuss the PSO-Based Neural Network of the study.

In 1991, [9] show that the majority of current models use a logistic function. A logistic function is a continuous function whose range is bounded. One advantage of this function is that its derivative is easily found. However, logistic function slows learning in the basic back propagation algorithm. Therefore, two of the alternatives which are hyperbolic tangent and scaled arctangent along with their derivatives are proposed.

According to [9], *tanh* function has additional advantage of equalizing training over layers compared with sigmoid functions. Through experiment, sigmoid functions never reach theoretical minimum or maximum. It is certainly reasonable to use the extremes of 0.0 and 1.0 as inputs to a network. But, it is ineffective to train a network to achieve extreme value as its output.

Another study in [10] shows comparison performance of different activation functions such as semilinear, semiquadratic and logarithmic-exponential function in MLP networks. The results showed that the data composed of Gaussians give far well than the logistic sigmoid function or the proposed semiquadratic function.

As well, an experiment with inverse tangent function to accelerate back propagation learning is presented in [11]. Simulation results with different categories of problems have shown that an inverse tangent function had improved the learning speed and reduced the possibility of being trapped in local minima.

On the other hand, in [12], the convergence rate of improved error with sigmoid function is much faster compared to the improved error with arctangent function. This result is contradictory to the expected result that arctangent should yield faster convergence speed as this function has higher derivatives values compare to sigmoid function.

There are many studies that have used evolutionary techniques like PSO to enhance the BP learning. Optimizations of nonlinear function using particle swarm methodology have been introduced by Russell Eberhart and James Kennedy in 1995. This paper reviews the PSO concept and discusses application of the algorithm to train artificial neural network weights. The results show that original *gbest* version performs best in terms of median number of iterations to convergence, while the *lbest* version with a neighborhood of two is most resistant to local minima [13].

In order to improve the performance of the particle swarm optimizer, a new parameter, called inertia weight is introduced into the original particle swarm optimizer. Simulations have been done to illustrate the significant and effective impact of this new parameter on the particle swarm optimizer. Through the experiment, it is concluded that the PSO with the inertia weight in the range of [0.9,1.2] will have a better performance and has a bigger chance to find the global optimum within a reasonable number of iterations. Furthermore, a time decreasing inertia weight is introduced which brings in a significant improvement on the PSO performance [14].

In 1998, impact of the inertia weight and maximum velocity allowed on the performance of PSO are investigated. A number of experiments have been done with different inertia weights and different values of maximum velocity allowed. It is concluded that when *Vmax* is small (less than 2), then an inertia weight of approximately 1 is chosen. While when *Vmax* is not small (bigger than 3), then an inertia weight w = 0.8 is a good choice. When the knowledge regarding the selection of *Vmax* is lacking, it is a good choice to set *Vmax* equal to *Xmax* and an inertia weight w = 0.8 for a good starting point [15].

In 1999, empirical study of PSO has been done with four different benchmark functions with asymmetric initial range settings are selected as testing functions. The experimental results illustrated the advantages and disadvantages of PSO. Under testing cases, the PSO always converges very quickly towards the optimal positions but may slow its convergence when it near to a minimum. The experimental results show that PSO is a promising optimization method and by using an adaptive weight, the performance can be improved [16].

According to [17], an evolutionary system for evolving feed-forward ANN called PSONN has been proposed. In order to evaluate the ability of PSONN in evolving ANN, it was applied to two real problems in the medical domain such as breast cancer and heart disease. The results show that ANN evolved by PSONN yield good accuracy and generalization ability.

A comparison between the efficiency of Particle Swarm Optimization Feedforward Neural Network (PSONN) and Genetic Algorithm Backpropagation Neural Network (GANN) is presented in [18].Two programs have been developed and the results show that PSONN give promising results in term of convergence rate and classification compared to GANN.

In addition, significant research have been done in [19] on training Artificial Neural Networks (ANN) with Particle Swarm Optimization (PSO). The results show that ANN requires the user to choose a network topology, inputs, and transfer functions for a network before training. However, PSO has overcome these limitations due to its capability in optimizing the weights. Any network parameter may be thrown into the mix along with the network weights to be optimized.

Recently in 2007, a hybrid algorithm combining particle swarm optimization (PSO) algorithm with back-propagation (BP) algorithm to train the weights of feed forward neural network (FNN) are investigated in [20]. Three kinds of algorithm to evolve the weights of the feed forward neural network with two layered structures are used. Assuming that the hidden transfer function is sigmoid function, and the output transfer function is a linear activation function. The experimental results show that the proposed hybrid PSO–BP algorithm is better than the Adaptive Particle swarm optimization algorithm (APSOA) and BP algorithm in terms of speed and accuracy.

Different researchers approach in activation functions, PSO and PSONN are depicted in table 1.2 below.

Table 1.2 : Various Studies of Activation Functions, PSO and PSONN

Title/ Researcher /Year	Brief Description
Efficient activation functions for the back-propagation neural network. (Kenue, 1991)	Majority of current models use a logistic function. Advantage of this function is that its derivative is easily found. However, slows learning in the basic back propagation algorithm caused two alternatives activation functions which are hyperbolic tangent and scaled arctangent along with their derivatives are proposed.
Why Tanh ? Choosing a Sigmoidal Function. (Kalman and Kwasny, 1992)	This paper shows *tanh* has the additional advantage of equalizing training over layers compared with sigmoid functions.
Visual comparison of performance for different activation functions in MLP networks. (Filip et.al.,1994)	Different activation functions such as semilinear, semiquadratic and logarithmic-exponential transfer function are compared. Data composed of Gaussians gives results far better than the logistic sigmoid function or the proposed semiquadratic function.
A Note on Activation function in Multilayer Feed forward Learning. (Kamruzzaman et.al.,2002)	In this paper, inverse tangent function is presented to accelerate back propagation learning. Simulation results with different categories of problems show inverse tangent function improved learning speed and reduce the chance of being trapped in local minima.

An Improved Error Signal of Backpropagation With Different Activation Function (Faridatul Azna Ahmad Shahabudin, 2003)	Through experiment, convergence rate of improved error with sigmoid function is much faster compared to the improved error with arctangent function. This result is contradictory to the expected result that arctangent should yield faster convergence speed as this function has higher derivatives values compare to sigmoid function.
A new optimizer using particles swarm theory (James Kennedy and R. Eberhart, 1995)	This paper reviews the PSO concept and discusses application of the algorithm to the training of artificial neural network weights. Results show that original *gbest* version performs best in terms of median number of iterations to convergence, while the *lbest* version with a neighborhood of two is most resistant to local minima.
A modified particle swarm optimizer (Y Shi and R. Eberhart, 1998)	To improve the performance of the particle swarm optimizer, a new parameter, called inertia weight is introduced into the original particle swarm optimizer. Simulations have been done to illustrate the significant and effective impact of this new parameter on the particle swarm optimizer.
Parameter selection in particle swarm optimization (Y. Shi and R. Eberhart, 1998).	Impact of the inertia weight and maximum velocity allowed (*Vmax*) on the performance of PSO are investigated. A number of experiments have been done with different inertia weights and different values of maximum velocity allowed.

Empirical study of particle swarm optimization (Y. Shi and R. Eberhart,1999)	Empirical study of Particle Swarm Optimization had been done and the advantages and disadvantages of PSO are investigated. Under all testing cases, the PSO always converges very quickly towards the optimal positions but may slow its convergence spend when near a minimum. The experimental results show that PSO is a promising optimization method and using an adaptive weight can improve the PSO performance.
Particle Swarm Optimization for Evolving Artificial Neural Network. (Zhang, C., Shao, H. and Li, Y, 2000)	An evolutionary system for evolving feed-forward ANN called PSONN has been proposed. In order to evaluate the ability of PSONN in evolving ANN, it was applied to two real problems in the medical domain such as breast cancer and heart disease. The results show that ANN evolved by PSONN has good accuracy and generalization ability.
Particle Swarm Optimization For Neural Network Learning Enhancement. (Haza Nuzly Bin Abdull Hamed, 2006)	Comparison between the efficiency of Particle Swarm Optimization Feedforward Neural Network (PSONN) and Genetic Algorithm Back propagation Neural Network (GANN) has been made. Results show that PSONN give promising results in term of convergence rate and classification compared to GANN.
Enhancement Of Elman Recurrent Network Learning With Particle Swarm Optimization (Mohamad Firdaus bin Ab Aziz, 2006)	Comparison between Elman Recurrent Network with Backpropagation (ERNBP) and Elman Recurrent Network with Particle Swarm Optimization (ERNPSO) to seek the performance of both networks. The results show that ERNPSO gives promising outcomes in terms of classification accuracy and convergence rate compared to ERNBP.

A Hybrid Particle Swarm and Neural Network Approach for Reactive Power Control (Paulo et. al., 2006)	Standard methods like back-propagation require the user to choose a network topology, inputs, and transfer functions for a network before training. PSO overcomes these limitations because it is blind to what it is optimizing. Any network parameter may be thrown into the mix along with the network weights to be optimized.
Particle Swarm Optimization for Evolving Artificial Neural Network (Zhang et. al.,2007)	A hybrid algorithm combining particle swarm optimization (PSO) algorithm with back-propagation (BP) algorithm to train the weights of feed forward neural network (FNN) is proposed. Results show that the proposed hybrid PSO–BP algorithm is better than the Adaptive Particle swarm optimization algorithm (APSOA) and BP algorithm in convergent speed and convergent accuracy.

1.3 Problems Statement

To date, no study has been conducted on investigating the effectiveness of implementing of PSO *Vmax* with activation functions in PSO-Based ANN. Hence, this study will explore the significant of implementing *Vmax* activation functions in PSO-Based ANN.

The hypothesis of this study can be stated as:

PSO Vmax Activation functions could affect the classification result of PSO based NN

1.4 Project Aim

The aim of this project is to explore the effectiveness of *Vmax* activation functions in PSO-Based NN learning in terms of convergence and classifications rate.

1.5 Objectives

Few objectives have been identified in this study:

i) To propose *Vmax* activation functions in PSO-Based NN Learning.
ii) To investigate the effectiveness of different types of *Vmax* activation function in PSO-based ANN.
iii) To compare and validate the efficiency of *Vmax* activation function in PSO-Based ANN.

1.6 Project Scope

The scopes of this project are defined as follows:

i) Four types of datasets are used to compare with previous study.

ii) The PSO program will be developed using Microsoft Visual C++ 6.0.

iii) Two activation functions which are Sigmoid function and Hyperbolic tangent function will be used and apply together with Vmax function to compare the effectiveness on PSO-Based NN.

1.7 Significance of Project

The project will investigate the impact of *Vmax* activation function in PSO-Based ANN. This is important to identify the effectiveness of PSO-Based NN with different *Vmax* activation functions for future study.

1.8 Organization of Project

This report consists of five chapters. Chapter 1 presents the introduction of the study, problems background, hypothesis, objectives and project scope. Chapter 2 gives literature reviews on the Neural Network, Particle Swarm Optimization and different activation functions in previous study. Chapter 3 will discuss the project methodology and Chapter 4 discusses the experimental results. The conclusion and suggestions for future work are explained in Chapter 5.

REFERENCES

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

2. Michael Negnevitsky. A *Guide To Intelligent System*, 2nd ed. Addison-Wesley/Pearson Education.165-189 (2004)

3. Andries P.Engelbrecht. *An Introduction of Computational Intelligence*, Second Edition, John Wiley & Sons, Ltd, (2007).

4. Timothy Masters. *Practical Neural Network Recipes in C++*. Morgan Kaufmann is an imprint of Academic Press (81-85) (1993).

5. Moshe, L., Allan,P. Multilayer Feedforward Networks With A Non-Polynomial Activation Function Can Approximate Any Function. *Proc. of Sixth Int. Symp. on Computer Science, Thailand* (1992).

6. Ananda Rao,M., Srinivas,J. *Neural Network Algorithms and Applications*. 3nd ed. McGraw-Hill, (2006).

7. James, K., Eberhart,R., Shi,Y. *Swarm intelligence*, Morgan Kaufmann Publishers, (2001).

8. James, K., Eberhart,R. Particle swarm optimization. *Proceedings, IEEE Intemational Conf. on Neural Networks, Perth, Australia.* Vol. IV, pp. 1942-1948.(2001)

9. Kalman, B.L., and Kwasny, S.C.Why Tanh? Choosing a Sigmoidal Function. *International Joint Conference on Neural Networks*, Baltimore, MD(1992).

10. Piekniewski, F., Rybicki, L.: Visual Comparison of Performance for Different Activation Functions in MLP Networks. *In: IJCNN 2004 & FUZZ-IEEE,* Vol. 4. Budapest 2947-2953 (2004).

11. Kamruzzaman, J. Aziz, M. A Note on Activation function in Multilayer Feedforward Learning. *Faculty of Information Technology, Monash University, Gippsland campus Victoria-3842, Australia* (2002).

12. Faridatul Azna Ahmad Shahabudin. An Improved Error Signal Of Back propagation With Different Activation Function In Electricity Load Fore-casting.*Prosiding Simposium Kebangsaan Sains Matematik ke-XI.* (2003).

13. James, K., Eberhart, R. A new optimizer using particles swarm theory, in: *Proc. of Sixth Int. Symp. on Micro Machine and Human Science,* Nagoya, Japan 39–43 (1995).

14. Shi,Y., Eberhart, R. A modified particle swarm optimizer. *IEEE International Conf. An Evolutionary Computation, Anchorage, Alaska. USA,* pp. 69-73. (May 1998).

15. Shi,Y., Eberhart, R. Parameter Selection in Particle Swarm Optimization. *Proc. Seventh Annual Conf. on Evolutionary Programming.* pp.591-601, (March 1998).

16. Shi,Y., Eberhart, R. Empirical study of particle swarm optimization. *Proceeding of the 1999 Congress on Evolulionary Computation*. CEC 99.Vol. 3.(1999).

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

18. Haza Nuzly Bin Abdull Hamed. Particle Swarm Optimization For Neural Network Learning Enhancement.*Thesis University Technology Malaysia* (2006).

19. Paulo, F., Kyle, W. A Hybrid Particle Swarm and Neural Network Approach for Reactive Power Control. *IEEE*: 2132-2145 (2006).

20. Zhang, J.R. , Zhang, J. , Lyu, M. A hybrid particle swarm optimization–back-propagation algorithm for feed forward neural network training. *Applied Mathematics and Computation* 185 (2007) 1026–1037 (2007).

21. Gao.L, Zhou.C, Gao H.B, Shi Y.R. Combining Particle Swarm Optimization and Neural Network for diagnosis of Unexplained Syncope*, ICIC 206, LNBI 4115*, pp 174-181 (2006).

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

23. Benda,L. Unsupervised learning with Particle Swarm Optimization on a real E-puck. *IEEE International Conf. An Evolutionary Computation, USA,* pp. 32-45. May (2007).

24. Mohamad Firdaus Bin Ab Aziz. Enhancement Of Elman Recurrent Network Learning with Particle Swarm Optimization. *Thesis University Technology Malaysia* (2006).

25. Ricardo.G. *Lecture Note Intelligent Sensor System.* Wright State University. Unpublished (2002).

26. Ahmed T. (2004). *Adaptive Particle Swarm Optimizer for Dynamic Environments.* The University Of Texas: Master Thesis.

27. Lin L. I. (2005). *Particle Swarm Optimization for Solving Constraint Satisfaction Problems*. Simon Fraser University: Master Thesis.

28. Stone M: Cross-Validate choice and assessment of statistical predictions. *Journal of the Royal Statistical Society* 1974, 36:111-147.

29. Kohavi R: Wrappers for performance enhancement and oblivious decision graphs. *PhD thesis*. Stanford University; 1995.

30. Donald J: *Statistics-A Self Teaching Guide* , Fourth Edition, John Wiley & Sons, Inc. 127-152 (1997).

31. Definition and description of t-test. Available from <http://en.wikipedia.org/wiki/Student's_t-test>. (Accessed on 8 june 2008).

32. Demo t-test calculation. Available from http://europe.isixsigma.com/library/content/c070613a.asp> (Accessed on 8 june 2008).

33. Knowledge of t-test. Available from <http://www.socialresearchmethods.net/kb/stat_t.php> (Accessed on 8 june 2008).

34. Table of t-statistics. Available from <http://davidmlane.com/hyperstat/t_table.html> (Accessed on 8 june 2008).

35. Andrew.W, Moore.G. *Cross-validation for detecting and preventing overfitting*. School of Computer Science, Carnegie Mellon University. Unpublished (2001).

36. Table of Critical Values for T. Available from <http://www.jeremymiles.co.uk/misc/tables/t-test.html> (Accessed on 12 june 2008).