

IMPROVED CUCKOO SEARCH BASED NEURAL NETWORK LEARNING ALGORITHMS FOR DATA CLASSIFICATION

ABDULLAH

A thesis submitted in
fulfillment of the requirement for the award of the
Doctor of Philosophy in Information Technology

Faculty of Computer Science and Information Technology
Universiti Tun Hussein Onn Malaysia

AUGUST, 2014

ABSTRACT

Artificial Neural Networks (ANN) techniques, mostly Back-Propagation Neural Network (BPNN) algorithm has been used as a tool for recognizing a mapping function among a known set of input and output examples. These networks can be trained with gradient descent back propagation. The algorithm is not definite in finding the global minimum of the error function since gradient descent may get stuck in local minima, where it may stay indefinitely. Among the conventional methods, some researchers prefer Levenberg-Marquardt (LM) because of its convergence speed and performance. On the other hand, LM algorithms which are derivative based algorithms still face a risk of getting stuck in local minima.

Recently, a novel meta-heuristic search technique called cuckoo search (CS) has gained a great deal of attention from researchers due to its efficient convergence towards optimal solution. But Cuckoo search is prone to less optimal solution during exploration and exploitation process due to large step lengths taken by CS due to Levy flight. It can also be used to improve the balance between exploration and exploitation of CS algorithm, and to increase the chances of the egg's survival.

This research proposed an improved CS called hybrid Accelerated Cuckoo Particle Swarm Optimization algorithm (HACPSO) with Accelerated particle Swarm Optimization (APSO) algorithm. In the proposed HACPSO algorithm, initially accelerated particle swarm optimization (APSO) algorithm searches within the search space and finds the best sub-search space, and then the CS selects the best nest by traversing the sub-search space. This exploration and exploitation method followed in the proposed HACPSO algorithm makes it to converge to global optima with more efficiency than the original Cuckoo Search (CS) algorithm.

Finally, the proposed CS hybrid variants such as; HACPSO, HACPSO-BP, HACPSO-LM, CSBP, CSLM, CSERN, and CSLMERN are evaluated and compared with conventional Back propagation Neural Network (BPNN), Artificial Bee Colony Neural Network (ABCNN), Artificial Bee Colony Back propagation algorithm (ABC-BP), and Artificial Bee Colony Levenberg-Marquardt algorithm (ABC-LM). Specifically, 6 benchmark classification datasets are used for training the hybrid Artificial Neural Network algorithms. Overall from the simulation results, it is realized that the proposed CS based NN algorithms performs better than all other proposed and conventional models in terms of CPU Time, MSE, SD and accuracy.

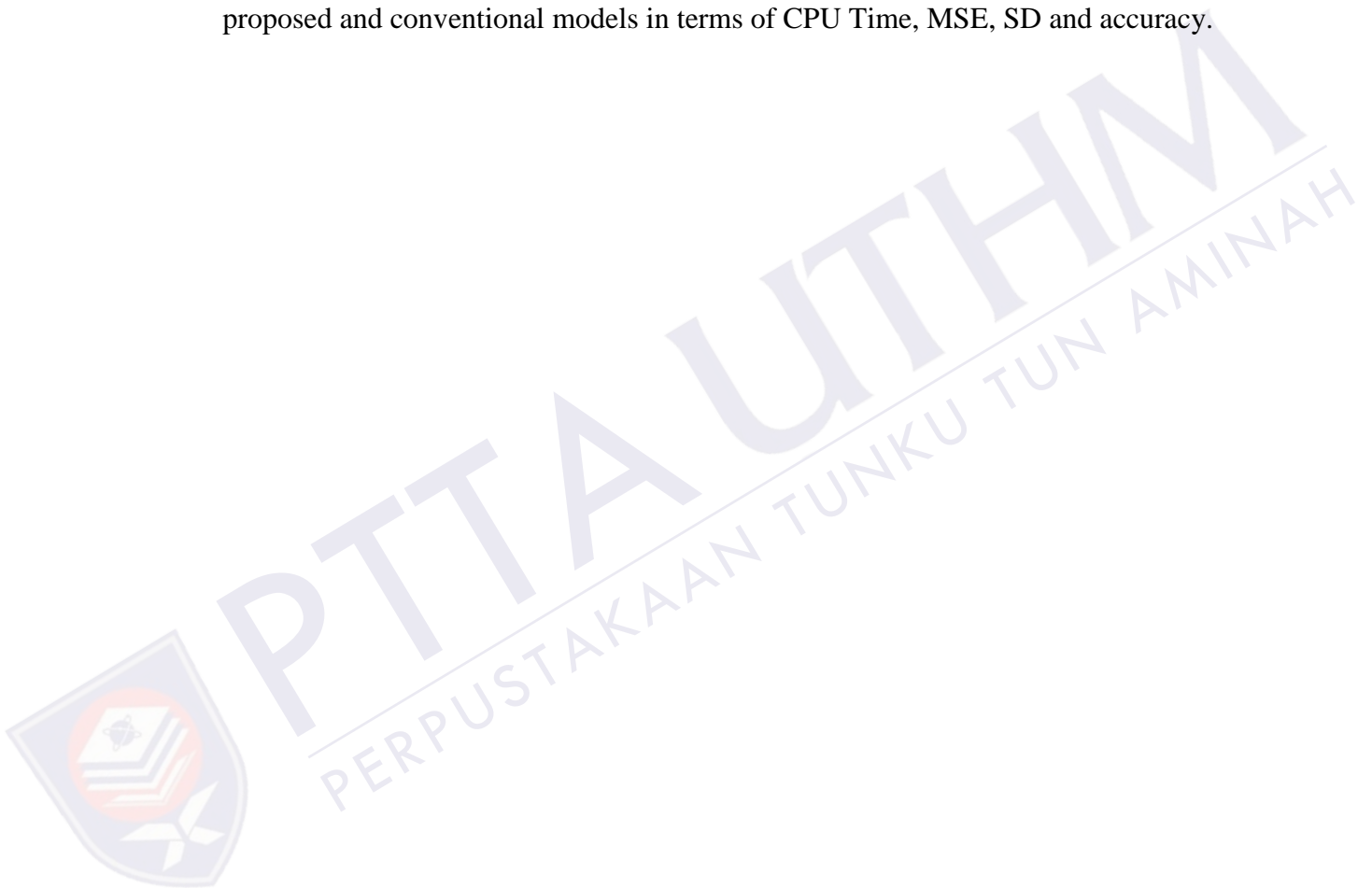


TABLE OF CONTENTS

DECLARATION	ii
DEDICATION	iii
ACKNOWLEDGEMENT	iv
ABSTRACT	v
TABLE OF CONTENTS	vii
LIST OF TABLES	x
LIST OF FIGURES	xi
LIST OF ALGORITHMS	xiv
LIST OF SYMBOLS AND ABBREVIATIONS	xv
LIST OF APPENDICES	xviii
LIST OF PUBLICATIONS	xix
CHAPTER 1 INTRODUCTION	1
1.1 Background of the Research	1
1.2 Problem Statements	4
1.3 Aims of the Research	5
1.4 Objectives of the Research	6
1.5 Scope of the Research	6
1.6 Significance of the Research	7
1.7 Thesis Outline	8
CHAPTER 2 LITERATURE REVIEW	10
2.1 Introduction	10
2.2 Artificial Neural Network (ANN)	11
2.3 Multilayer Perceptron (MLP)	12
2.4 Recurrent Neural Network (RNN)	14
2.4.1 Elman Recurrent Network (ERN)	15
2.4.2 Previous improvements on RNN	17
2.5 Back Propagation Neural Network (BPNN)	19

2.6	The BP Training Algorithm Limitation	21
2.7.1	Delta Bar Delta Rule	23
2.7.2	Learning Rate (η)	23
2.7.3	Momentum Coefficient (α)	24
2.7.4	Parameters Selection	26
2.7.5	Improved Error Function	27
2.7.6	Improved Activation Function	27
2.8	Second Order Methods	29
2.8.1	Related works on Second Order Methods	30
2.8.2	Levenberg-Marquardt (LM) Algorithm	31
2.8.3	Improvement on LM Training NN	33
2.9.1	Meta-heuristic Techniques	37
2.10	Cuckoo Search via Levy Flight (CS) Algorithm	39
2.10.1	Levy Flight in Cuckoo Search	40
2.11	Artificial Bee Colony (ABC) Algorithm	43
2.12	Particle Swarm Optimization (PSO) Algorithm	46
2.13	Accelerated Particle Swarm Optimization Algorithm	47
2.14	Research Gap Analysis on BPNN	48
2.15	Summary	50
CHAPTER 3	RESEARCH METHODOLOGY	52
3.1	Introduction	52
3.2	The Proposed CSLM Algorithm	54
3.3	The Proposed CSBP Algorithm	61
3.4	The Proposed CSERN Algorithm	67
3.5	The Proposed CSBPERN Algorithm	72
3.6	The Proposed CSLMERN Algorithm	77
3.7	The Proposed HACPSO algorithm	85
3.8	The Proposed HACPSO-BP Algorithm	89
3.9	The proposed HACPSO-LM Algorithm	95
3.10	Data Collection	101
3.11	Variable Selection	102
3.12	Data Pre-Processing	102
3.13	Data partitioning	103
3.14	Neural Network Architecture	103

3.15	Performance Comparison and Accuracy Measurement	104
3.16	Summary	105
CHAPTER 4	RESULTS AND SIMULATION	106
4.1	Introduction	106
4.2	Preliminaries	107
4.3	Wisconsin Breast Cancer Classification Problem	108
4.4	IRIS Classification Problem	113
4.5	Thyroid Benchmark Classification Problem	117
4.6	Diabetes Classification Problem	121
4.7	Glass Classification Problem	125
4.8	Australian Credit Card Classification Problem	129
4.9	Result and Discussion	133
4.10	Summary	134
CHAPTER 5	CONCLUSION AND FUTURE WORK	135
5.1	Introduction	135
5.2	Research Summary	135
5.3	Research Contributions	138
5.4	Future works	138
	REFERENCES	140
	Table A.1 Gantt-Chart of Project Activities	151
	APPENDIX	152
	VITAE	176



LIST OF TABLES

4.1	Summary of algorithms performance for breast cancer classification problem	110
4.2	Summary of algorithms performance for Iris Benchmark classification Problem.	114
4.3	Summary of algorithms performance for Thyroid Benchmark classification Problem	118
4.4	Summary of algorithms performance for Diabetes Benchmark classification Problem.	122
4.5	Summary of algorithms performance for Glass Benchmark classification Problem	126
4.6	Summary of algorithms performance for Card Benchmark classification Problem.	130

LIST OF FIGURES

2.1	Multilayer Perceptron (MLP)	12
2.2	The MLP the error function	13
2.3	Cuckoo search Variants	43
2.4	Previous Research On BPNN Algorithm	49
3.1	Flowchart of the Research process	53
3.2	The Proposed CSLM Algorithm	55
3.3	Proposed flowcharts for (CSBP)	64
3.4	Proposed flowcharts for (CSERN)	68
3.5	Proposed flowcharts for (CSLMERN)	78
3.6	Proposed flowcharts for (HACPSO-BP)	95
4.1	Comparison of average MSE, and SD for breast cancer classification problem.	111
4.2	MSE Convergence comparison performances for breast cancer classification problem of the proposed techniques	112
4.3	MSE Convergence comparison performances for breast cancer classification problem of the proposed techniques	112
4.4	MSE Convergence comparison performances for breast cancer classification problem of the standerd algorithms	113
4.5	Comparison of average MSE and SD for Iris classification Problem	115
4.6	MSE Convergence comparison performances for Iris classification Problemof the proposed techniques.	116

4.7	MSE Convergence comparison performances for Iris classification Problem of the proposed techniques.	116
4.8	MSE Convergence comparison performances for Iris classification Problem of the standard algorithms	117
4.9	Comparison of average MSE and SD for Thyroid classification Problem	119
4.10	MSE Convergence comparison performances for Thyroid classification Problem of the proposed algorithms	120
4.11	MSE Convergence comparison performances for Thyroid classification Problem of the proposed algorithms	120
4.12	MSE Convergence comparison performances for Thyroid classification Problem of the standard algorithms	121
4.13	Comparison of average MSE and SD for Diabetes classification Problem	123
4.14	MSE Convergence comparison performances for Diabetes classification Problem of the proposed techniques	124
4.15	MSE Convergence comparison performances for Diabetes classification Problem of the proposed techniques	124
4.16	MSE Convergence comparison performances for Diabetes classification Problem of the standard algorithms	125
4.17	Comparison of average MSE and SD of the algorithms for Glass classification Problem	127
4.18	MSE Convergence comparison performances for Glass classification Problem of the proposed techniques	128
4.19	MSE Convergence comparison performances for Glass classification Problem of the proposed techniques	129
4.20	MSE Convergence comparison performances for Glass classification Problem of the standard algorithms	129

4.21	Comparison of average MSE and SD for Credit Card classification Problem	131
4.22	MSE Convergence comparison performances for Credit Card classification Problem of the proposed techniques	132
4.23	MSE Convergence comparison performances for Credit Card classification Problem of the proposed techniques	132
4.24	MSE Convergence comparison performances for Credit Card classification Problem of the standard algorithms	133



LIST OF ALGORITHMS

- (i) Hybrid Accelerated Cuckoo Particle Swarm Optimization (HACPSO) algorithm.
- (ii) Hybrid Accelerated Cuckoo Particle Swarm Optimization Back propagation (HACPSO-BP) algorithm.
- (iii) Hybrid Accelerated Cuckoo Particle Swarm Optimization Levenberg-Marquardt (HACPSO-LM) algorithm.
- (iv) Cuckoo Search Back Propagation (CSBP) algorithm.
- (v) Cuckoo Search Levenberg-Marquardt (CSLM) algorithm.
- (vi) Cuckoo Search Recurrent Elman Network (CSERN) algorithm.
- (vii) Cuckoo Search Back Propagation Elman Recurrent network (CSBPERN) algorithm.
- (viii) Cuckoo Search Levenberg-Marquardt Elman Recurrent network (CSLMERN) algorithm.
- (ix) Conventional Back Propagation Neural Network (BPNN) algorithm.
- (x) Artificial Bee Colony Neural Network (ABCNN) algorithm.
- (xi) Artificial Bee Colony Back Propagation (ABC-BP) algorithm.
- (xii) Artificial Bee Colony Levenberg-Marquardt (ABC-LM) algorithm.

LIST OF SYMBOLS AND ABBREVIATIONS

N	Number of output nodes in the output layer
T_i	Desired output of the i^{th} output unit
Y_i	Network output of the i^{th} output unit
δ_k	Is the error for the output layer at k^{th} node
δ_j	Is the error for the hidden layer at j^{th} node
h_j	Output of the j^{th} hidden node
O_i	Output of the i^{th} input node
η	Learning rate
i, j	Subscripts i , and j , corresponding to input, hidden
k	Subscript k , corresponding to output nodes
w_{jk}	Weight on the link from hidden node j to output node k
	Weight on the link from input node i to hidden node j
v_i^{t+1}	Is the velocity vector
x_i^t	Is the position vector
α	Is the learning parameter or accelerating constant
ϵ_n	Is the random vector drawn from $N(0, 1)$
g^*	Global best
x^{new}	New value obtained
x^{old}	Old value in the data
x^{max}	Maximum of the old data range
x^{min}	Minimum of the old data range
U	The Upper normalization bound
L	The Lower normalization bound
T_i	Predicts data
A_i	Actual data
n	Total number of inputs patterns

X_i	The observed value
\bar{X}_i	Mean value of the observed value
ANN	Artificial Neural Network
ALM	Adaptive Learning Rate and Momentum
AF	Activation Function
ACO	Ant Colony Optimization
ABPNN	Ant Back Propagation Neural Network
ABC	Artificial Bee Colony algorithm
APSO	Adaptive Particle Swarm Optimization
APSO-BP	Adaptive Particle Swarm Optimization Back Propagation
ABC-LM	Artificial Bee Colony algorithm Levenberg-Marquardt
BP	Back Propagation Neural Network
BPNN	Back Propagation Neural Network
BPFM	Back Propagation with Fixed Momentum
BPALM	Back Propagation with Adaptive Learning Rate and Momentum.
BPERN	Back Propagation Elman Recurrent Network
BPGD-AG	Back Propagation Gradient Descent with Adaptive Gain
BPGD-AGAMAL	Back Propagation Gradient Descent with Adaptive Gain and adaptive momentum and learning rate
BPTT	Back Propagation through Time
CSBP	Cuckoo Search Back Propagation
CSLM	Cuckoo Search Levenberg-Marquardt.
CSERN	Cuckoo Search Recurrent Elman Network
CSBPERN	Cuckoo Search Back Propagation Elman Recurrent network
CSLMERN	Cuckoo Search Levenberg-Marquardt Elman Recurrent network
CG	Conjugate Gradient
DO	Dissolved oxygen
ERNN	Elman Recurrent Neural Network
ERN	Elman Recurrent Network
ERNPSO	Elman recurrent network particle swarm optimization
FFNN	Feed Forward Neural Network

FBP	Firefly Back propagation algorithm
FCRNN	Fully Connected Recurrent Neural Network
GDAM	Gradient Descent with Adaptive Momentum
GN	Gauss Newton
HACPSO	Hybrid Accelerated Cuckoo Particle Swarm Optimization
HACPSO-BP	Hybrid Accelerated Cuckoo Particle Swarm Optimization Back propagation
HACPSO-LM	Hybrid Accelerated Cuckoo Particle Swarm Optimization Levenberg-Marquardt
IT	Information Technology
IWS	Initial Weight Selection
LM	Levenberg-Marquardt
MLP	Multilayer Perceptron
MLPNN	Multilayer perceptron Neural Network
MSE	Mean Squared Error
OBP	Optical Back Propagation
PSO-BP	Particle Swarm Optimization Back Propagation
PSO	Particle Swarm Optimization
PCA	Principle Component analysis
QN	Quasi Newton
RNN	Recurrent Neural Network
SLPNN	Single Layer Perceptron Neural Network
SI	Swarm Intelligence
TN	Total nitrogen
TP	Total phosphorus

LIST OF APPENDICES

APPENDIX	TITLE	PAGE
A	Table A.1 Gantt-Chart of Project Activities	151
B	Table B.1 Algorithms Performance for Breast Cancer Classification Problem	152
B	Table B.2 Algorithms Performance for IRIS Classification Problem	156
B	Table B.3 Algorithms Performance for Diabetes Classification Problem	160
B	Table B.4 Algorithms Performance for Glass Classification Problem	164
B	Table B.5 Algorithms' Performance for Card Classification Problem	168
B	Table B.6 Algorithms Performance for Thyroid Classification Problem	172

LIST OF PUBLICATIONS

1. N. M. Nawawi, **Abdullah Khan**, M. Z. Rehman (2013). *A New Optimized Cuckoo Search Recurrent Neural Network (CSRNN) Algorithm*. In: ROVISIP-2013, Proceedings in LNEE Journal of Springer, Penang, Malaysia.
2. M. Z. Rehman, N. M. Nawawi, **Abdullah Khan** (2013). Countering the problem of oscillations in Bat-BP gradient trajectory by using momentum. In: DaEng 2013, Proceedings in LNEE Journal of Springer. Kuala Lumpur, Malaysia. (indexed by ISI Proceedings, EI-Compendex, SCOPUS, EI Compendex, MetaPress, Springer link).
3. N. M. Nawawi, **Abdullah Khan**, M. Z. Rehman (2013). *CSBPRNN: A New Hybridization Technique Using Cuckoo Search to Train Back Propagation Recurrent Neural Network*. In: DaEng 2013, Proceedings in LNEE Journal of Springer. Kuala Lumpur, Malaysia. (indexed by ISI Proceedings, EI-Compendex, SCOPUS, EI Compendex, MetaPress, Springer link).
4. N. M. Nawawi, **Abdullah Khan**, M. Z. Rehman (2013). A New Back-propagation Neural Network optimized with Cuckoo Search Algorithm. B. Murgante et al. (Eds.): ICCSA 2013, Part I, LNCS 7971, pp. 413--426. Springer, Heidelberg.
5. N. M. Nawawi, **Abdullah Khan**, M. Z. Rehman (2013). A New Cuckoo Search based Levenberg-Marquardt (CSLM) Algorithm. B. Murgante et al. (Eds.): ICCSA 2013, Part I, LNCS 7971, pp. 438--451. Springer, Heidelberg.
6. N. M. Nawawi, **Abdullah Khan**, M. Z. Rehman (2013). A New Levenberg-Marquardt based Back-propagation Algorithm trained with Cuckoo Search. In: ICEEI-2013, Proceedings in Elsevier Technology Journal.
7. N. M. Nawawi, **Abdullah Khan**, M. Z. Rehman (2013). *CSLM: Levenberg-Marquardt based Back Propagation Algorithm Optimized with Cuckoo Search*, J. ICT Res. Appl., Vol. 7, No. 2, 2013, 105-119.

8. Nazri Mohd. Nawi, M. Z. Rehman, M. I. Ghazali, M. N. Yahya, **Abdullah Khan** (2014). *Hybrid Bat-BP: A New Intelligent tool for Diagnosing Noise-Induced Hearing Loss (NIHL) in Malaysian Industrial Workers*, J. Applied Mechanics and Materials, Trans Tech Publications, Switzerland, vol. 465-466, pp. 652—656, 2014.
9. N. M. Nawi, M. Z. Rehman, **Abdullah Khan**(2013). *The Effect of Bat Population in Bat-BP Algorithm*. In: ROVISIP-2013, Proceedings in LNEE Journal of Springer, Penang, Malaysia.
10. N. M. Nawi, M. Z. Rehman, **Abdullah Khan** (2013). *A New Bat Based Back-Propagation (BAT-BP) Algorithm*. In: ICSS-2013, Proceedings in LNEE Journal of Springer, Wroclaw, Poland.
11. N. M. Nawi, **Abdullah Khan**, M. Z. Rehman (2014). *Data Classification Using Metaheuristic Cuckoo Search Technique for Levenberg-Marquardt Back Propagation (CSLM) Algorithm*. ICoMEIA 2014, Penang.
12. N. M. Nawi, M. Z. Rehman, **Abdullah Khan** (2014). *WS-BP: A New Wolf Search based Back-propagation Algorithm*. ICoMEIA 2014, Penang.
13. N. M. Nawi, **Abdullah Khan**, M. Z. Rehman (2014). *Comparison of CSBP, CSRNN, and CSLM algorithms on Parity bit Classification datasets*. SCDM 2014.
14. N. M. Nawi, **Abdullah Khan**, M. Z. Rehman (2014). *CSLMEN: A New Cuckoo Search Levenberg-Marquardt Elman Network*. SCDM 2014.
15. N. M. Nawi, **Abdullah Khan**, M. Z. Rehman (2014). *Researching the Effect of Back Propagation Based Cuckoo Search on Data Classification*. Accepted in IRECOS Journal for 2014.
16. N. M. Nawi, M. Z. Rehman, **Abdullah Khan** (2013). *Verifying the Accuracy of GDAM algorithm on Multiple Classification Problems*. In: InteliSys 2013, 11 - 12 October 2013, Jakarta, Indonesia.
17. N. M. Nawi, **Abdullah Khan**, M. Z. Rehman (2013). *CSLMEN: A New Optimized Method for Training Levenberg-Marquardt Elman Network Based Cuckoo Search Algorithm*. In: InteliSys 2013, 11 - 12 October 2013, Jakarta, Indonesia.

CHAPTER 1

INTRODUCTION

1.1 Background of the Research

Classification is one of the main data mining processes that maps or sort items into a group of related predefined objects or classes. Data sorting however is a difficult practice, especially if it is carried out by hand. Data sorting is a vital step to gather useful information about an association's working style. Data classification engages categorizing information to predefined levels or an object. For instance, to classify relevant data available (i.e. file type, working platform, normalizing file size in megabytes or gigabytes, to their appropriate groups or classes etc) a classification procedure is required which has to be done automatically (Chandra *et al.*, 2011).

Machine learning techniques provide us one of the best classification approaches to perform the data organization in a systematic manner. The main goal of Machine Learning is to answer the rising levels of computerization in the information creation process, substituting a vast amount of time burly human activity with techniques that improve accuracy or efficiency by finding out and utilizing the irregularities in the training data. Many techniques are available under the machine learning and the most important objective of machine learning research is to automatically identify the complex patterns and construct intelligent decisions based on the data provided (Chandra *et al.*, 2011).

Artificial Neural Networks (ANN) is an intelligent Machine Learning technique modeled on the human brain and includes processing units known as

artificial neurons that can be trained to perform complex calculations. Unlike conventional techniques in which an output is based on the input it gets, an ANN can be trained to learn and guess patterns without having the prior information about the form of the function (Zheng and Gong, 1992; Kosko, 1992; Basheer and Hajmeer, 2000; Krasnopolsky and Chevallier, 2003; Coppin, 2004). Multilayered ANN architectural models are most efficient when trained on complex patterns. Usually, multilayered networks are classified as Feed-Forward Networks and Recurrent Neural Networks with respect to the path of their connections (Haykin, 1994; Guler *et al.*, 2005).

Recurrent Neural Network (RNN) can achieve incredibly non-linear vibrant mappings and thus have temporally complete applications, where multilayer feed forward networks are restrained to perform static mappings (Elman, 1990; Gupta *et al.*, 2000; Saad *et al.*, 1998). Fully recurrent networks use unrestricted fully interrelated architectures and learning algorithms that can deal with time varying inputs or outputs in a non-linear manner. Therefore, this research's primary focus is on partially recurrent networks, where connections are mainly feed forward, and they comprise a carefully selected set of feedback associates. The reappearance allows the system to memorize past history from the precedent without complicating the learning extremely (Guler and Ubeyli *et al.*, 2005). One example of partially RNN is Elman which is set up as a usual feed forward network (Elman, 1990).

Back-Propagation Neural Network (BPNN) is an ANN algorithm, used for recognizing a mapping function among a known set of inputs and outputs. The conventional BPNN method is very sensitive to the parameters such as initial weights as indicated by Kolen and Pollack (1991). Usually small random values are used to initialize weights in BPNN but an unsuitable weight value will cause the BPNN to be trapped in the local minima or face slow learning rate; whereas initializing large random weights can cause premature saturation. Therefore, careful selection of the initial weights is required to speed-up the slow learning process (Hyder *et al.*, 2009). The Elman network which utilizes the gradient descent technique for convergence also has the problem of initial weights and slow convergence. This is because the gradient descent may get stuck in local minima,

where it may stay indefinitely and the global minima of the error function will never be achieved (Ahmed *et al.*, 2001; Wen *et al.*, 2000; Nawi *et al.*, 2011).

A number of research studies have recently attempted to improve the convergence of the back propagation and Elman RNN. Second order optimization method such as; Levenberg-Marquardt (LM) has been used for network training (Levenberg, 1944; Marquardt, 1963; Nawi *et al.*, 2010; Nawi *et al.*, 2011; Yan, *et al.* 2009; Qing, *et al.*, 2010; Bogdan and Wilamowski *et al.*, 2007; Hagan and Menhaj, 1994). But, as LM also follows gradient descent, therefore it can get stuck in local minima for indefinite time and global minima will never be achieved (Karaboga *et al.*, 2011).

Seeing the problem of convergence in deterministic methods, many nature inspired meta-heuristic methods have been used to solve non-linear optimization problems, such as; artificial bee colony (ABC) algorithm (Karaboga, and Basturk 2007), particle swarm optimization (PSO) algorithm (Zhang, *et al.*, 2007), bat algorithm (BA) (Yang, 2011; Yang, and Gandomi, 2012), firefly algorithm (FA) (Yang, 2010; Gandomi *et al.*, 2013), and krill herd (KH) algorithm (Gandomi, and Alavi, 2012). It has been found through experimentation that meta-heuristic techniques are highly suitable for finding the optimal solution. Since these methods are known to have the capability of avoiding the local minima, therefore they are used in selecting the best weights and biases for the BPNN and Elman RNN networks (Ozturk and Karaboga, 2011). However, the algorithms still have a problem of slow convergence to global minima which has rendered them useless in real environments where the time constraint exists.

Developed by Yang and Deb in 2009, a novel meta-heuristic search technique called cuckoo search (CS) has gained a great deal of attention from researchers due to its efficient convergence towards optimal solution. CS imitates the cuckoo behavior of laying its eggs in other birds nest in such a way that the other bird is not able to find out the difference between the foreign eggs and its own. CS algorithm uses levy flight which makes possible to reach the global optimal solution efficiently (Yang and Deb, 2009; Yang and Deb, 2010; Tuba and Subotic *et al.*, 2011).

The Cuckoo search is prone to less optimal solution during exploration and exploitation process owing to large step lengths taken by CS due to Levy flight (Zheng and Zhou, 2012). Therefore, to improve the balance between exploration and exploitation of CS algorithm and to increase the chances of the egg's survival which will lead to converge to global optimal solution, this research proposed an improved CS algorithm hybridized with Accelerated particle Swarm Optimization (APSO) algorithm.

The proposed hybrid Accelerated Cuckoo Particle Swarm Optimization (HACPSO) is used to train different variants of neural networks such as: simple Back propagation (Rumelhart and Hinton *et al.*, 1986), feed forward neural network (FFNN), and Levenberg-Marquardt Back propagation Neural Network (Shereef and Baboo, 2011; Ozturk and Karaboga, 2011) by selecting the best weights and bias for avoiding the local minima and improving the convergence to global minimum.

1.2 Problem Statements

Back-Propagation Neural Network (BPNN) is a supervised learning Artificial Neural Network (ANN) algorithm that has been successfully applied in wide assortment of applications (Coppin, 2004). Nevertheless the back propagation is very sensitive to the parameters such as initial weights (Kolen and Pollack, 1991). Usually small random values are used to initialize weights. However, starting with unsuitable weight values will cause it to be trapped in the local minima or leads towards slow convergence. Since, it uses gradient descent and can easily get stuck in local minima where it may stay indefinitely (Nawi and Ransing *et al.*, 2011; Ahmed *et al.*, 2001; Wen *et al.*; 2000), a number of research studies have attempted to improve the convergence of the back propagation. Second order optimization methods such as quasi-Newton, and Levenberg-Marquardt (LM) have also been used for neural networks training (Hagan and Menhaj, 1994; Yusak and Tanoto *et al.*, 2011; Wilamowski *et al.*, 2007; Yan and Hui *et al.*, 2009; Qing *et al.*, 2010).

Among the conventional optimization methods researchers prefer Levenberg-Marquardt (LM) because of its convergence speed and performance. On the other hand, LM algorithm is derivative based having a risk of getting stuck local minima. To deal with this problem, global search optimized techniques, have gained great attention by researchers since they are known to have the capability of avoiding local minima (Ozturk and Karaboga, 2011).

Recently, nature inspired meta-heuristic based global search techniques have become popular in finding the global optimal solution (Yang, 2010). Cuckoo Search (CS) is a recently proposed meta-heuristic algorithm by Yang and Deb in 2009. CS is found to be quite efficient during convergence towards optimal solution but it is prone to less optimal solution during exploration and exploitation process due to large step lengths by Levy flight (Zheng and Zhou, 2012). Therefore, a modification of CS algorithm is needed to improve the balance between exploration and exploitation of CS algorithm, and to increase the chances of the cuckoo egg's survival.

1.3 Aims of the Research

The research aims to develop improved Cuckoo Search (CS) based learning techniques to train Multilayer Perceptrons (MLP) for searching optimal solution during exploration and exploitation. The hybrid techniques will use accelerated particle swarm optimization (APSO) algorithm to search within the search space first and finds the best sub-search space. Then APSO finds the best solution or best nest from sub-search spaces and share this information with CS. The proposed algorithm is further integrated with different variants of neural networks such as; Levenberg-Marquardt Back propagation Neural Network (Shereef and Baboo, 2011; Ozturk and Karaboga, 2011), Simple Back propagation (Rumelhart and Hinton *et al.*, 1986), Elman RNN, Back propagation Elman RNN, and Levenberg-Marquardt Elman RNN. The performance of the proposed techniques are compared with conventional Back propagation (BP), Artificial Bee Colony Neural Network (ABCNN), Artificial Bee Colony Back propagation algorithm (ABC-BP), and Artificial Bee Colony

Levenberg-Marquardt algorithm (ABC-LM) (Ozturk and Karaboga, 2011; Nandy *et al.*, 2012; Karaboga and Ozturk, 2009; Rumelhart *et al.*, 1986) on selected benchmark classification problems from UCI Machine Learning Repository.

1.4 Objectives of the Research

This research encompasses the following four objectives in order to achieve the research aims;

- i. To implement and simulate the effect of levy flight in Cuckoo Search algorithm based on first and second order Neural Network algorithms in avoiding local minima towards converging to global minimum.
- ii. To develop a hybrid algorithm known as HACPSO which further enhances CS by introducing APSO in finding the best solution or best nest from sub-search space and share this information to CS.
- iii. To implement and assess the performance of the proposed HACPSO algorithm in (ii) to train the first and second order Neural Network to provide a better search direction for the network.
- iv. To assess the performances of the proposed algorithms in terms of accuracy and mean square error comparing with conventional algorithms on selected benchmark classification problems.

1.5 Scope of the Research

This research is focused on enhancing the Cuckoo Search algorithm, with different variant of Neural Network in order to improve accuracy, network convergence, and to avoid local minimum. Initially Cuckoo Search algorithm was explored with first and second order ANN such as Back Propagation (BP) Elman Recurrent Network (ERN), Back propagation Elman Recurrent Network (BPERN), Levenberg-Marquardt Back propagation (LMBP), and Levenberg-Marquardt Elman Recurrent

Network (LMERN). Later, the proposed CS algorithm is integrated with Accelerated Particle Swarm Optimization (APSO) which is applied to first and second order Network. All these Networks were exploited on some selected benchmark datasets from University California Irvine Machine Learning Repository (UCIMLR) were employed in order to verify the efficiency of the proposed algorithm.

1.6 Significance of the Research

This research provides the following contributions to knowledge in the fields of meta-heuristic based learning algorithm for BPNN. In meta-heuristic learning algorithms, the performance highly depends on exploration and exploitation procedures:

- i. Cuckoo Search with Levy flight algorithm has been simulated and implemented with different variants of Neural Network in order to remove the oscillations in the gradient path and to avoid local minima problem.
- ii. An improved Hybrid Accelerated Cuckoo Particle Swarm Optimization (HACPSO) algorithm is proposed to improve the balance between exploration and exploitation of CS algorithm, and to increase the chances of the egg's survival by intelligently selecting the best search space and the optimal solution in CS. This exploration method followed in the proposed HACPSO algorithm makes it to converge to global optimal solution with more efficiency than the original Cuckoo Search (CS) algorithm.
- iii. In the first phase, this Research investigates the accuracy performance of the proposed proposed HACPSO algorithm for training first order neural network, such as FFNN, BPNN, and second order network (LMBP) and compared with ABCNN, ABC-BP, ABC-LM, and conventional BPNN.

- iv. Cuckoo Search with Levy flight and HCAPSO and their variants are finally compared by means of simulation on some selected benchmark classification problems taken from UCI Machine Learning Repository.

1.7 Thesis Outline

This thesis is divided into five chapters including the introduction and conclusion chapters. The outline of each chapter is given below;

Besides providing an outline of the thesis, Chapter 1, contains the overview on background of the Research, scope of the Research, objectives, aims, and significance of the Research.

Chapter 2 consists of some efficient learning methods for BPNN algorithm. The BPNN algorithm is one of the finest and widely used learning algorithms for Artificial Neural Network (ANN). However, BP algorithm has problem of slow convergence and local minima. This chapter reviews some of the fundamental theory about ANN, BPNN, Levenberg-Marquardt (LM) algorithm, and Recurrent Neural Networks (RNN). This chapter also sheds some light on the previous improvements proposed by various researchers on improving the training efficiency of these networks. Some of the fundamental theories about the meta-heuristics, such as; Cuckoo Search (CS) algorithm, Artificial Bee Colony (ABC) algorithm, Particle Swarm Optimization (PSO) and Accelerated Particle Swarm Optimization (APSO) algorithms are also reviewed. The problem of slow convergence in Cuckoo Search (CS) due to large step lengths taken by levy flight and the poor communication among cuckoos in CS are identified in the Chapter 2.

Chapter 3 presents the main contribution of this research known as HACPSO to answer the slow convergence and poor communication abilities of cuckoos. The proposed algorithms based on CS with different variants of ANN are also presented and discussed in the Chapter 3. Finally, Chapter 3 discusses the research methodology used to carry out this Research in a systematic manner.

In Chapter 4, the new algorithms developed in Chapter 3 are further validated in terms of simulations on selected benchmark problems for UCI Machine Learning Repository. The performance evaluation is carried out based on accuracy and Mean Squared Error (MSE) and all the proposed algorithms are compared with conventional BPNN, ABCNN, ABC-BP, and ABC-LM algorithms.

Finally in Chapter 5, research contributions are summarised and recommendations for future work are proposed for further continuation in this field.



CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

In the past two decades, much advancement has been done in the Information Technology (IT) field. Previously, accessing large information was a time consuming process but due to recent developments in Artificial Intelligence (AI) accessing large information has become much easier and less time consuming. Artificial Neural Network is powerful technique in solving complex and non-linear problems. The main reason for ANN being usually used is the ability to present some properties such as learning from training data.

This chapter is organized in the following manner. Section one gives an introduction to the most popular Artificial Neural network (ANN) architecture called multilayer perceptron (MLP) (Fung *et al.*, 2005). Further in this chapter, the Elman Recurrent Neural Network (ERNN) and other training algorithms for the Elman network are discussed. Section two gives a deep review of Back Propagation Neural Network (BPNN) algorithm, which is one of the most novel and widely used algorithms for training neural networks. This chapter also highlights the limitations of the conventional BPNN training algorithm. Last but not least, this chapter also discusses some improvements and modifications of the BPNN learning algorithm which were done in the past. To improve the performance of the conventional BPNN this Chapter also focuses on some well-known meta-heuristic techniques used in this research, such as Cuckoo Search (CS) via levy flight, particle swarm optimization

(PSO), Accelerated Particle Swarm Optimization (APSO), and Artificial Bee Colony (ABC) algorithms.

2.2 Artificial Neural Network (ANN)

Artificial Neural Network (ANN) is an analytical procedure modeled on the learning of human cognitive system and neurological function of the brain. ANN is one of the most popular approaches used extensively in machine learning, which involves the development of the algorithms that enables the computer to learn (Negnevitsky, 2005). ANN works by processing information like human neurons in the brain and is composed of small interconnected processing elements known as nodes (neurons), which can be trained to perform complex computations (Chen and Pei, 2008). ANN is a powerful set of adaptive learning techniques that can be trained to store, recognize, estimate and adapt to new patterns without having the prior information of the function it receives (Popescu *et al.*, 2009).

ANN can demonstrate a surprising number of characteristics of human brain, which has the ability to learn from training through examples fed to it (Elhag and Wang, 2007). Due to its ability to solve complex time critical problems, it has been widely used in the engineering fields such as biological modeling, financial forecasting, weather forecasting, decision modeling, control system, health and medicine, ocean and space exploration etc (Zhen and Gong, 1992; Basheer and Hajmeer, 2000; Kosko, 1992; Krasnopolsky and Chevallier, 2003; Coppin, 2004; Lee, 2008). One of the basic types of ANN is feed forward Neural Network (FFNN) which is capable of approximating generic classes of function which includes continuous and discrete values. And the most frequently used FFNN for pattern classification is the Multilayer Perceptron Neural Network (MLP) which is trained to generate a spatial output pattern in response to an input spatial one (Haykin, 1994).

2.3 Multilayer Perceptron (MLP)

Multilayer Perceptron (MLP) is the most popular Artificial Neural Networks (ANN) architecture used due to its simple construction (Popescu *et al.*, 2009). MLP is also known as Multilayer Feed forward Neural Network (MLFNN), and consists of more than one layer of nodes and is appropriate for large number of applications (Fung *et al.*, 2005). The MLP network comprises of a set of nodes (neurons) that represent the input layer one or more hidden layers of computation, and output layer of nodes that calculate the output of the network. In MLP, nodes in any layer of the network are connected to all other neurons in the adjacent layer. The input signal propagates in forward direction from left to right through the network. Figure 2.1 shows the MLP network structure which consist of three layers; one input, one hidden, and one output layer.

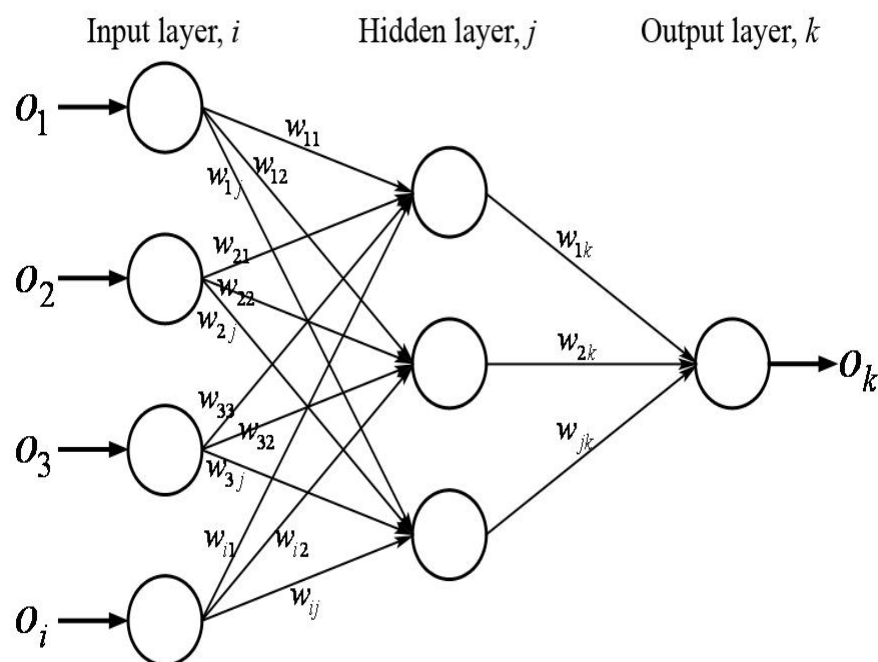


Figure 2.1: Multilayer Perceptron (MLP)

MLP training in feed forward neural network is also known as supervised learning process and can be interpreted as an example of an optimization method. The purpose of the learning process is to find a weight vector W^+ which minimizes the difference between the actual and predicted output which can be defined as following:

$$\text{Err} = \frac{1}{2} \sum_{i=1}^n (T_i - Y_i)^2 \quad (2.1)$$

where;

- n : Number of output nodes in the output layer.
- T_i : Desired output of the i^{th} output unit.
- Y_i : Network output of the i^{th} output unit.

Error function can be visualized in one dimensional weight space as shown in Figure 2.2.

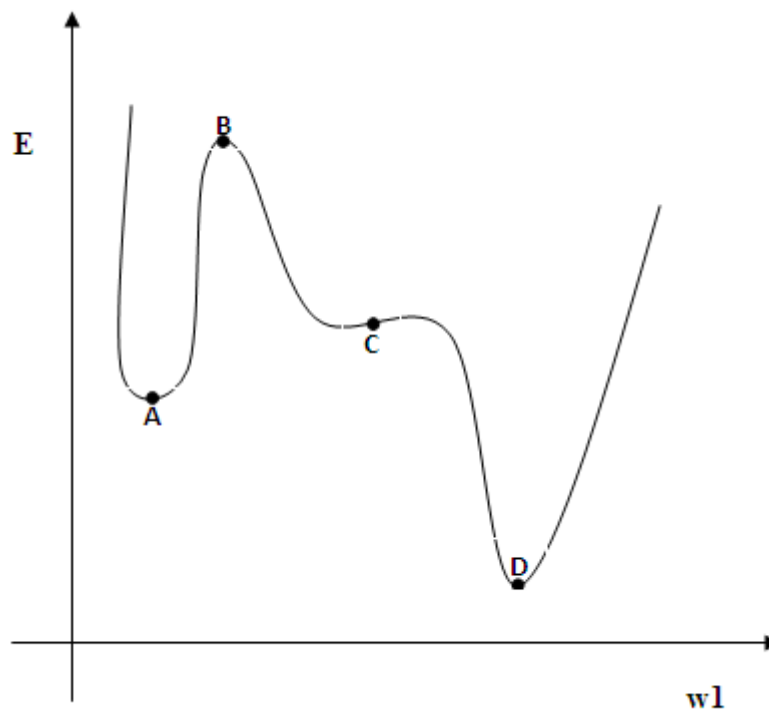


Figure 2.2: The MLP error function

Figure 2.2 shows that the MLP error function is a non-linear function of the weights and have many minimum, which satisfies the following equation:

$$\nabla \text{Err}(w) = 0 \quad (2.2)$$

where, $\nabla \text{Err}(w)$ denotes the gradient of the error, E with respect to weights. Figure 2.2, shows schematic error function for a single parameter w_1 , showing four stationary points. The point at which the value of the error function is smallest (Point

D) is called global minima, while point A is called local minima. There may be also other points, which satisfy the error condition in Equation (2.2) for instance local maxima (point B) or saddle point (point C) (Nawi, 2007).

Generally the MLP network is trained with one of the most popular and traditional algorithm known as back propagation neural network (BPNN) algorithm which will be discussed in detail in Section 2.5.

Although, MLP is stable and popular but its mapping performance is fixed, therefore, the network is not suitable for processing temporal pattern. Many attempts have been made to use the MLP to classify temporal patterns by transforming the temporal domain into a spatial domain. An alternate neural network approach is to use recurrent neural network (RNN) which has the memory to train on the past history. (Gupta and Mcavoy, 2000; Saad *et al.*, 1998). RNN is discussed in more detail in the next section.

2.4 Recurrent Neural Network (RNN)

Recurrent Neural Network (RNN) is an alternate neural network architectural approach which has the ability to store past inputs to a node in its memory for future use. Unlike, multi-layer feed forward neural network (MLFNN), it can be trained on temporal data easily (Ubeyli, 2008a).

Recently, various types of RNNs have been suggested and they may be classified as partially recurrent or fully recurrent networks. RNN can carry out highly nonlinear dynamic mappings, therefore RNNs have been used in a number of interesting applications including associative memories, pattern classification control, optimization forecasting, and generalization of pattern sequence (Ubeyli 2008a, 2008c). In partially recurrent network, partial recurrence is produced by feedback delay hidden unit output or the output of the network as additional input units. The partially recurrent network whose links are mostly feed forward, also

contain a careful chosen set of feedback connections. One instance of such network is Elman which is usually set-up as a normal feed forward neural network (Elman, 1990).

2.4.1 Elman Recurrent Network (ERN)

Among some feasible network architectures, the feed forward and recurrent neural networks (RNN) are commonly used (Haykin, 1994). In a feed forward neural network the signals are transmitted only in one direction, starting from the input layer, consequently through the hidden layers to the output layer. A recurrent neural network (RNN) has local feedback connections to some of the previous layers. It is different from feed forward network architecture in the sense that there is at least one feedback loop. Thus in RNN, there can exist one layer with feedback connections as well as there can also be neurons with self-feedback link where the output of a neuron is fed back into itself as the input (Kazemy *et al.*, 2007).

Thus, the partially recurrent network where the connections are largely feed forward consists of carefully chosen set feedback association. The recurrence allows the network to memorize output from the past without complicating the learning greatly (Ubeyli, 2008). One of the popular networks in the partially recurrent network is Elman Recurrent Network (ERN). An ERN is a network which in principle is set up as a normal feed forward network with a feedback connection from the hidden layer to the input layer. It means that all the neuron in one layer is connected with all neurons in the next layer called context layer which is a special container of hidden layer (Elman, 1990; Guler and Ubeyli *et al.*, 2005).

The nodes in the context layer receive copy of the output of the hidden neurons. The output of every hidden node is copied into a specific neuron in the context layer (Ubeyli *et al.*, 2005). The value of the context neuron is used as extra input for all the neurons in the hidden layer as time delay. Therefore the Elman

network has an open memory of one time delay (Elman, 1990; Ubeyli, 2008; Kazemy *et al.*, 2007).

Like MLFNN, the strength of all association among nodes is represented with weights. Initially, all the weight values are represented randomly and changed during the training process. In the Elman RNN the weights from the hidden layer to the context layer is set to one, and kept fixed because the values of the context nodes have to be copied accurately (Elman, 1990).

A simple ERN is mostly used with one input layer, one hidden or state layer, and one output layer. Each layer will have its own index variable: k for output nodes, j and l for hidden, and i for input nodes. In a feed forward network, the input vector X is propagated through a weight layer V .

$$net_j(t) = \sum_i^n X_i(t)V_{ji} + b_j \quad (2.3)$$

where, n is the number of inputs is, b_j is a bias, and f is an output function. In a simple recurrent network, the input vector is similarly propagated through a weight layer, but also combined with the previous state activation through an additional recurrent weight layer, U ;

$$Y_j(t) = f(net_j(t)) \quad (2.4)$$

$$net_j(t-1) = \sum_i^n X_i(t)V_{ji} + \sum_l^m Y_l(t-1)U_{jl} + b_j(t) \quad (2.5)$$

where, m is the number of 'state' nodes. The output of the network in both cases is determined by the state and a set of output weights W ;

$$Y_k(t-1) = g(net_j(t-1)) \quad (2.6)$$

$$net_k(t) = \sum_j^m Y_j(t-1)W_{kj} + b_k \quad (2.7)$$

where, g is an output function and W_{kj} represents the weights from hidden to output layer.

In the next section, we will discuss the previous improvements by various researchers on Recurrent Neural Networks (RNN) to achieve faster convergence to global minima.

2.4.2 Previous improvements on RNN

There has been a lot of research on dynamic system modeling with recurrent neural networks (RNN). The Recurrent Neural Networks (RNN) has an inside feedback connection within the network which allows it to hold past presented pattern. This ability of dynamic modeling system makes this kind of neural network more superior than the conventional feed forward neural network because the system outputs are function of both the current inputs as well as their inner states (Barbounis *et al.*, 2006; Peng *et al.*, 2007).

As a part of supervised learning technique, different training algorithms were established for training the RNNs, such as BPNN through time (Ahmad *et al.*, 2004) and second order optimization LM algorithm (Toha *et al.*, 2008; Guler *et al.*, 2005). However the existence of feedback loops in the network architecture, the calculation of the gradient becomes more complex which makes the BPNN procedure computationally more intricate. In addition the error surface of the recurrent network is more complex as compared to the static network. Therefore, the training is more likely to be trapped in to local minima (Peng *et al.*, 2007).

Ahmad and Ismail *et al.* (2004) investigated a new method using Fully Connected Recurrent Neural Network (FCRNN) and Back Propagation Through Time (BPTT) algorithm to observed the difference of Arabic alphabetic like “alif” to “ya” and to improve the people’s knowledge and understanding of Arabic words using the proposed technique. The experimental results showed that the proposed method has better performance and can achieve high rate of convergence.

Peng *et al.*, (2007) trained RNN by integrating it with particle swarm optimization (PSO) and BPNN algorithm. The PSO-BP algorithm provides the optimal weights for identifying the frequency dependent impedance of power electronic system such as rectifiers, inverter and AC-DC converter. The experimental results showed that the proposed method successfully identified the impedance characteristics of the three phase inverter system, not only it can systematically help avoiding the training process getting trapped in local minima but also has better performance as compared to both simple BPNN and PSO algorithms.

Aziz and Hamed *et al.*, (2008) carried out a research to improve the performance in ERN training with PSO algorithm. To discover the classification accuracy and convergence rate, the proposed algorithm is compared with simple ERN and BPNN algorithms. Based on the simulation results, the proposed Elman recurrent network with particle swarm optimization (ERNPSO) algorithm has better performance than the Back Propagation Elman Recurrent Network (BPERN) in terms of classification accuracy. However, in terms of convergence time the BPERN is much better than the proposed ERNPSO algorithm.

Cheng and Shen (2010) proposed an improved ERN to calculate radio propagation loss with three dimensional parabola method in order to decrease calculation time and to improve approximation performance of the network. Based on the results, the improved ERNs showed high performance in predicting propagation loss than the simple ERN.

Wang and Gao *et al.* (2011) used the ERN to compute the Total Nitrogen (TN) Total Phosphorus (TP) and Dissolved Oxygen (DO) at three different sites of lake Taihu during the period of water diversion. The conceptual form of the ERN for different parameters was used by mean of the principle component analysis (PCA), to train and validate on daily dataset. The values of TS, TP, and DO was calculated by the model were ultimately related to their respective values. The simulated results showed that the PCA can efficiently accelerate the input parameters for the ERN and

can precisely compute and forecast the water quality parameters during the period of water diversion, but still it is not free from local minimum problem.

Tanoto and Ongsakul *et al.*, (2011) proposed LM algorithm based on Elman and Jordan recurrent neural networks to forecast annual peak load of Java Madura Bali interconnection from 2009 to 2011. The research was carried out to check the performance of the proposed LM based recurrent networks with respect to their forecasting accuracy over the given time period. From the simulations, it is clear that the proposed LM based recurrent neural networks have better results.

Normally, most of the ERNs and FFNN are generally trained with back propagation (BP) algorithms, which will be explained in the next section.

2.5 Back Propagation Neural Network (BPNN)

Back propagation Neural Network (BPNN) is one of the most widely used supervisor learning model for updating the MLP weights during the training process. It was proposed in order to solve the problem of a single layer perceptron which fails to solve XOR patterns (Minsky and Papert, 1969). The BPNN algorithm has been individually calculated by many researchers working in different fields. Werbos (1974) presented the basic idea of BP algorithm while working on his doctoral thesis called back propagation of error. Parker (1985) redeveloped the BPNN algorithm and called it the learning logic algorithm. Finally Rumelhart *et al.*, (1986) rediscovered the algorithm and since then the technique is commonly used. For this reason, the BPNN can be viewed as the standard method of complex patterns learning ANN technique. The BPNN learns by calculating the error of the output layer to find the error in the hidden layers. This capability makes it highly appropriate to be applied on problem in which no relation is set up between the output and the input.

The BPNN algorithm uses gradient descent method which requires careful selection of parameters such as network topology, initial weights, biases, learning

rate, and activation function. An inappropriate use of these parameters can lead to slow network convergence or even network stagnancy (Zawei and Althoefer, 2005). The issues of convergence in back propagation is really important and to solve these problems different techniques were developed. And new modification is given in the Section 2.7. However this section explains the traditional implementations of the BPNN algorithm which is known as batch or offline BPNN. The procedure for conventional back propagation is given as follows:

- i. Initialize all weights and present input patterns to the neural network.
- ii. Identify desired outputs for each input pattern.
- iii. All the input is then propagated forward through the network, until the output layer.
- iv. Calculate error by comparing the network output with the desired output using Equation (2.1).
- v. The error is propagated backward through the network is used to adjust the weights using the following equation. The error for each unit k in the output layer

$$\delta_k = Err_k(1 - Err_k)(T_k - Err_k) \quad (2.8)$$

- vi. The error is propagated backward to compute the error specifically, for each unit j at the hidden layer. Using Equation (2.9);

$$\delta_j = Err_j(1 - Err_j) \sum_k \delta_k w_{jk} \quad (2.9)$$

- vii. The weights in the links connecting to output nodes, (w_{jk}) , and hidden nodes, (w_{ij}) , are then modified based on the gradient descent method as following:

$$\nabla w_{jk} = w_{jk} + \eta \delta_k h_j \quad (2.10)$$

$$\nabla w_{ij} = w_{ij} + \eta \delta_j o_i \quad (2.11)$$

where;

δ_k	:	Is the error for the output layer at k^{th} node
δ_j	:	Is the error for the hidden layer at j^{th} node
h_j	:	Output of the j^{th} hidden node
O_i	:	Output of the i^{th} input node
η	:	Learning rate
i, j, k	:	Subscripts i, j , and k , corresponding to input, hidden and output nodes, respectively
w_{jk}	:	Weight on the link from hidden node j to output node k
w_{ij}	:	Weight on the link from input node i to hidden node j

Since, BPNN algorithm uses gradient descent (GD) to update weights therefore, BPNN cannot be guaranteed to reach global minima of the error function. This limitation of the BPNN will be explained in the next section.

2.6 The BP Training Algorithm Limitation

The traditional BP has been successfully applied in some real problems such as, predictions, pattern recognitions, and classifications. Despite providing many successful applications, BP faces several limitations which need to be solved. Since BP algorithm used gradient descent (GD) optimization technique to update weights, one of the limitations of these methods is that it does not guarantee to reach global minima of the error function (refer to Figure 2.2). BP contains slow learning convergence and can easily get trapped in local minima and fails to find the global best solution (Bi *et al.*, 2005; Wang *et al.*, 2004).

Although the gradient descent method is an iterative process to find the parameters that can minimize an error measure, the error surface normally possess properties that make this method too slow to converge. When the derivative of the weight is small in magnitude, the error surface is relatively smooth along a weight

dimension, thus the weight value is adjusted by a small amount and many procedures are required to make major reduction in error (Nawi, 2007).

Another cause for the slow speed of convergence of the gradient descent is that the negative gradient vector direction may not point directly toward the minimum of the error surface (Nawi, 2007). It is also noted that the neuron saturation in the hidden layer is closely associated with many local minima problems. When such saturation occurs, neuron in the hidden layer will lose their sensitivity to the input signals and propagated chain is blocked severely and in some situations, the network can no longer be trained. Moreover, the BP algorithm convergence behavior depends on the selection of network architecture, initial weights, biases, learning rate, momentum coefficient and the activation function.

In the last decade, a significant numbers of different learning algorithms have been introduced by researchers in order to overcome those limitations of BP algorithm. The next section will discuss some improvements of BP proposed by previous researchers.

2.7 Improvements on BP Algorithm

In recent years, with many improvements in the research and applications, researchers have been investigating the problems associated with the learning efficiency and convergence rate of the BP algorithm. Much work has been done to improve the general capacity of the network. Several learning acceleration techniques have been proposed as modification to the original BP algorithm. The research on BP falls in three categories:

- i. Heuristic technique which include variation of learning rate, using momentum and gain tuning of the activation function.
- ii. Second Order optimization techniques.

iii. Hybridization Techniques.

Based on the first category, various accelerating techniques have been proposed. This technique consists of Delta Bar Delta Rule, and involves varying the learning rate, momentum coefficient, and gain value of the activation function.

2.7.1 Delta Bar Delta Rule

The Delta- Bar- Delta rule, developed by Rich Sutton (1986) consists of a weights and learning rate update rule. The weights update rule is the same as the Delta Bar Delta rule in Equation (2.12);

$$W(t + 1) = w(t) - \alpha(t + 1) \frac{\delta x(t)}{\delta w(t)} \quad (2.12)$$

Jacobs (1988) has introduced a Delta- Bar- Delta rule modification, which consist of weight update rule and learning update rule. From his research, it is noted that if the consecutive change of the weights $W(t + 1)$ and $w(t)$ in the opposite direction the weights value is oscillating, than the learning rate (α), for that weight should be decremented. Similarly, if the consecutive derivative of the weights has the same direction than the learning rate for that weight should be increased. From the research, it is found the Delta -Bar-Delta shows a faster rate of convergence than the gradient descent.

2.7.2 Learning Rate (η)

One of the major issues with the BP algorithm is the fixed Learning rate η . In BP, it is very crucial to find the optimal value of η that can cause great reduction in the network error value. The reliability of the training process depends closely on the

choice of η . Various methods have been developed to find out a better learning rate. However, these techniques are generally based on heuristics and do not present the best learning rate (Kandil *et al.*, 1993).

Yu *et al.* (1995) proposed dynamic optimization of the learning rate using derivative information. It was shown that relatively large or less learning rates may change the training of BP algorithm and may be lead to failure of learning process.

While, Ye (2001) stated that the constant learning rate of the back propagation fails to improve the search for the optimal weight combination. Furthermore, Yemeni and Hong (2009) proposed an auto adapted learning rate, although the adjustment of the network weights is associated with error gradient during the training. When the training has fallen into a flat area, error gradient is closed to zero. Then the learning rate is large and the change of weights will be still slow, which may cause slow convergence to the target error.

Thota *et al.* (2013) proposed optimal learning rate for the stabilized and fast convergence of the BP learning algorithm. It was shown that the consistency of the total system mostly depends on the choice of η value. A small value of η results in slow learning while a great value of η results in fast learning but may also face oscillations which leads to no learning at all. It states that 0.02 values is found to be the optimal learning rate value for minimum error, correct classification, and incorrect classification occurrence. Overall, it can be concluded that the small value of learning rate may cause slow convergence, and large value of the learning rate may lead towards oscillations.

2.7.3 Momentum Coefficient (α)

Another efficient approach related to the speedup of the convergence and stabilized training process is by adding some momentum coefficient to the network.

REFERENCES

- Ahmad, A., Ismail, S., & Samaon, D. (2004). *Recurrent neural network with backpropagation through time for speech recognition*. Paper presented at the Communications and Information Technology, 2004. ISCIT 2004. IEEE International Symposium on.
- Ahmed, W. A. M., Saad, E., & Aziz, E. (2001). Modified back propagation algorithm for learning artificial neural networks. *Radio Science Conference, 2001. NRSC 2001. Proceedings of the Eighteenth National, 1*, 345-352.
- Ampazis, N. P., S. J. (2000). Levenberg-Marquardt algorithm with adaptive momentum for the efficient training of feedforward networks. *Neural Networks, IJCNN 2000, 1*, 126-131
- Aziz, M., Hamed, H. N. A., & Shamsuddin, S. M. H. (2008). Augmentation of Elman Recurrent Network Learning with Particle Swarm Optimization. *Modeling & Simulation, 2008. AICMS 08. Second Asia International Conference on*, 625-630.
- Barbounis, T. G., Theocharis, J. B., Alexiadis, M. C., & Dokopoulos, P. S. (2006). Long-term wind speed and power forecasting using local recurrent neural network models. *Energy Conversion, IEEE Transactions on*, 21(1), 273-284.
- Basheer, I. A., & Hajmeer. (2000). Artificial Neural Networks: fundamentals, computing, design and application. *Journal of Microbiological Methods 43 (1)*, 03-31.
- Battiti, R. (1992). First-and second-order methods for learning: between steepest descent and Newton's method. *Neural computation*, 4(2), 141-166.
- Becker, S., & Le Cun, Y. (1988). *Improving the convergence of back-propagation learning with second order methods*. Paper presented at the Proceedings of the 1988 connectionist models summer school.
- Bishop, C. M. (1995). Neural networks for pattern recognition.
- Chandra, E. E., & Rajeswari, J. J. . (2011). A Survey on Data Classification using Machine Learning Techniques. *International Journal Of Engineering Science & Technology*, 3(10), 7397-7401.

- Chandrasekaran, K., & Simon, S. P. (2012). Multi-objective scheduling problem: Hybrid approach using fuzzy assisted cuckoo search algorithm. *Swarm and Evolutionary Computation*, 5, 1-16.
- Cheng, F., & Shen, H. (2010). An Improved Recurrent Neural Network for Radio Propagation Loss Prediction. *Intelligent Computation Technology and Automation (ICICTA), 2010 International Conference on*, 1, 579-582.
- Coppin, B. (2004). Artificial Intelligence Illuminated. *Jones and Bartlet illuminated Series, USA*, 291-324.
- Cortez, P., Cerdeira, A., Almeida, F., Matos, T., & Reis, J. (2009). Modeling wine preferences by data mining from physicochemical properties. *Decision Support Systems*, 47 (4).
- Deng, W. J., Chen, W. C., & Pei, W. (2008). Back-propagation neural network based importance-performance for determining critical service attributes. *Journal of Expert Systems and Applications* 34(2), 1-26.
- Dunham, M. H. (2002). Data Mining: Introductory and Advanced Topics.
- Eberhart, J. K. R. C. (1995). particle Swarm Optimization. *proce of IEEE International Conforences on Neural Network Piscataway NJ.,* 1942-1948.
- Elhag, T. M. S., & Wang, Y.-M. (2007). Risk Assessment for Bridge Maintenance Projects: Neural Networks versus Regression Techniques. *Journal of Computing in Civil Engineering*, 21 (6), 769-775.
- Elman, J. L. (1990). Finding structure in time. *Cognitive Science*, 14(2), 179-211.
- Evet, I. W., & Spiehler, E. J. (1988). Rule induction in forensic science. in (Eds.). Knowledge Based Systems. *Halsted Press*, 152-160.
- Fisher, R. A. (1936). The use of multiple measurements in taxonomic problems. *Annual Eugenics* 7, 179-188.
- Fkirin, M. A., Badwai, S. M., & Mohamed, S. A. (2009). Change Detection Using Neural Network in Toshka Area. *NSRC-2009,* 1-10.
- Fung, C. C., Iyer, V., Brown, W., & Wong, K. W. (2005). Comparing the Performance of Different Neural Networks Architectures for the Prediction of Mineral Prospectivity. *Proceedings of the Proceedings of 2005 International Conference on Machine Learning and Cybernetics*, 394-398.

- Gandomi, A., Yang, X.-S., Talatahari, S., & Alavi, A. (2013). Firefly algorithm with chaos. *Communications in Nonlinear Science and Numerical Simulation*, 18(1), 89-98.
- Gandomi, A. H., & Alavi, A. H. (2012). Krill herd: a new bio-inspired optimization algorithm. *Communications in Nonlinear Science and Numerical Simulation*, 17(12), 4831-4845.
- Gandomi, A. H., Yun, G. J., Yang, X.-S., & Talatahari, S. (2013). Chaos-enhanced accelerated particle swarm optimization. *Communications in Nonlinear Science and Numerical Simulation*, 18(2), 327-340.
- Güler, N. F., Übeyli, E. D., & Güler, İ. (2005). Recurrent neural networks employing Lyapunov exponents for EEG signals classification. *Expert Systems with Applications*, 29(3), 506-514.
- Gupta, L., McAvoy, M., & Phegley, J. (2000). Classification of temporal sequences via prediction using the simple recurrent neural network. *Pattern Recognition*, 33(10), 1759-1770.
- Hagan, M. T., & Menhaj, M. B. (1994). Training feedforward networks with the Marquardt algorithm. *Neural Networks, IEEE Transactions on*, 5(6), 989-993.
- Haykin, S. (1994). *Neural Network : A Comprehensive Foundation*. Macmillan, New York.
- Hyder, M. M., Shahid, M. I., Kashem, M. A., & Islam, M. S. (2009). Initial Weight Determination of a MLP for Faster Convergence. *Journal of Electronics and Computer Science*, 10.
- Nocedal, J. S. W. (1999). *Numerical Optimization*. Springer, Berlin.
- Jacobs, R. A. (1988). Increased Rate of convergence Through Learning Rate Adaptation:. *Neural Network*, 1, 561-573.
- Jeng-Bin, L., & Yun-Kung, C. (2005). A Novel Back-propagation Neural Network Training Algorithm Designed by an Ant Colony Optimization. *Transmission and Distribution Conference and Exhibition: Asia and Pacific, 2005 IEEE/PES*, 1-5.
- Jiang, M., Gielen, G., Zhang, B., & Luo, Z. (2003). Fast learning algorithms for feedforward neural networks. *Applied Intelligence*, 18(1), 37-54.

- Jin, W., Li, Z. J., Wei, L. S., & Zhen, H. (2000). The improvements of BP neural network learning algorithm. *Signal Processing Proceedings, 2000. WCCC-ICSP 2000. 5th International Conference on*, 3, 1647-1649.
- Karaboga, D., & Basturk, B. (2007). Artificial bee colony (ABC) optimization algorithm for solving constrained optimization problems. *Foundations of Fuzzy Logic and Soft Computing*, 789-798.
- Karaboga, D., & Ozturk, C. (2009). Neural networks training by artificial bee colony algorithm on pattern classification. *Neural Network World*, 19(3), 279-292.
- Kazemy, A., Hosseini, S.A., & Farrokhi, M. (2007). Second order diagonal recurrent neural network. *In: Proceedings of the IEEE International Symposium on Industrial Electronics (ISIE)*, 251--256.
- Kennedy, J. F., Kennedy, J., & Eberhart, R. C. (2001). Swarm intelligence.
- Kermani, B. G., Schiffman, S. S., & Nagle, H. T. (2005). Performance of the Levenberg–Marquardt neural network training method in electronic nose applications. *Sensors and Actuators B: Chemical*, 110(1), 13-22.
- Kolen, J. F., & Pollack, J. B. (1991). Back Propagation Is Sensitive To Initial Conditions. *in R. P. Lippmann, J. E. Moody & D. S. Tjupetzky (Eds.). Advances in Neural Information Processing Systems. Denver*, 860-867.
- Kosko, B. (1992). Neural Network and Fuzzy Systems (1st Edition ed.). *Prentice Hall of India*.
- Kotsiantis, S., Kanellopoulos, D., & Pintelas, P. (2006). Data preprocessing for supervised learning. *International Journal of Computer Science*, 1(2), 111-117.
- Krasnopolsky, V. M., & Chevallier. (2003). Some Neural Network applications in environmental sciences. Part II. *advancing computational efficiency of environmental numerical models. Neural Networks*, 16 (3), 335-348.
- Thota, L.S. S. B. C. (2013). Optimum Learning rate for Classification Problem with MLP in Data Mining. *International Journal of Advance in Engineering & Technology (IJAET)*, 6(1), 35-44.
- Layeb, A., & Boussalia, S. R. (2012). A Novel Quantum Inspired Cuckoo Search Algorithm for Bin Packing Problem. *International Journal of Information Technology & Computer Science*, 4(5).

- Lee, T. L. (2008). Back-propagation neural network for the prediction of the short-term storm surge in Taichung harbor. *Taiwan. Engineering Applications of artificial Intelligence* 21 (1) 63-72.
- Levenberg, K. (1944). A method for the solution of certain problems in least squares. *Quarterly of applied mathematics*, 2, 164-168.
- Li, J.-B., & Chung, Y.-K. (2005). *A novel back-propagation neural network training algorithm designed by an ant colony optimization*. Paper presented at the Transmission and Distribution Conference and Exhibition: Asia and Pacific, 2005 IEEE/PES.
- Lourakis, M. I. (2005). A brief description of the Levenberg-Marquardt algorithm implemented by levmar.
- Marquardt, D. (1963). An algorithm for least square estimation of nonlinear parameter ,. *SIAM Journal on applied Mathematics* 11(2), 431-441.
- Marquardt, D. W. (1963). An algorithm for least-squares estimation of nonlinear parameters. *Journal of the Society for Industrial & Applied Mathematics*, 11(2), 431-441.
- Minsky, M. L., & Papert, S. (1969). *Perceptrons; An Introduction to Computational Geometry*. MIT Press.
- Møller, M. F. (1993). A scaled conjugate gradient algorithm for fast supervised learning. *Neural networks*, 6(4), 525-533.
- Mitchell, R. J. (2008). On Simple Adaptive Momentum. *CIS-2008*, 01-06.
- Mo,L.(2011). An APSO optimized BP neural network. *Electronics, Communications and Control (ICECC), 2011 International Conference on*, 1297-1300.
- Mohd Nawawi, N., Ransing, R., & Abdul Hamid, N. (2011). BPGD-AG: A New Improvement Of Back-Propagation Neural Network Learning Algorithms With Adaptive Gain. *Journal of Science and Technology*, 2(2).
- Kandil N, K. K., R.V.Patel, V.K. Sood,. (1993). Optimum Learning Rate for Back Propagation Neural Network. *Electrical and Computer Engineering*, , 1, 465-468.
- Nandy, S., Sarkar, P. P., & Das, A. (2012a). Analysis of a Nature Inspired Firefly Algorithm based Back-propagation Neural Network Training. *IJCA vol 43(22)*.

- Nandy, S., Sarkar, P. P., & Das, A. (2012b). Training a Feed-forward Neural Network with Artificial Bee Colony Based Backpropagation Method. *arXiv preprint arXiv:1209.2548*.
- Nawi, N., Ghazali, R., & Salleh, M. (2010). The Development of Improved Back-Propagation Neural Networks Algorithm for Predicting Patients with Heart Disease. In R. Zhu, Y. Zhang, B. Liu & C. Liu (Eds.), *Information Computing and Applications* (Vol. 6377, pp. 317-324): Springer Berlin Heidelberg.
- Nawi, N., Ransing, R. S., Salleh, M., Ghazali, R., & Hamid, N. (2010). An Improved Back Propagation Neural Network Algorithm on Classification Problems. In Y. Zhang, A. Cuzzocrea, J. Ma, K.-i. Chung, T. Arslan & X. Song (Eds.), *Database Theory and Application, Bio-Science and Bio-Technology* (Vol. 118, pp. 177-188): Springer Berlin Heidelberg.
- Nawi, M. N. (2007). Computational Issues in Process Optimisation using Historical Data, PhD. Disseration, Swansea University.
- Nawi, N. M., Ransing, M. R., & Ransing, R. S. (2007). An improved Conjugate Gradient based learning algorithm for back propagation neural networks. *International Journal of Computational Intelligence* 4 (1), 46-55.
- Negnevitsky, M. (2005). Artificial Intelligence a Guide to Intelligent Systems. 2 Harlow, England: Addison Wesley.
- Ng, W. W. Y., Yeung, D. S., & Tsang, E. C. C. (2006, 13-16 Aug. 2006). *Pilot Study on the Localized Generalization Error Model for Single Layer Perceptron Neural Network*. Paper presented at the Machine Learning and Cybernetics, 2006 International Conference on.
- Norhamreeza A. H., Nawi. N. M., Rozaida Ghazali, Najib M S. (2011). Accelerating Learning Performance of Back Propagation Algorithm by Using Adaptive Gain Together with Adaptive Momentum and Adaptive Learning Rate on Classification Problems. *Second International Conference, UCMA 2011, Daejeon, Korea, April 13-15, 2011. Proceedings, Part II*, 151, 559-570.
- Otair, M. A., & Salameh, W. A. (2005). *Speeding up back-propagation neural networks*. Paper presented at the Proceedings of the 2005 Informing Science and IT Education Joint Conference.

- Ozturk, C., & Karaboga, D. (2011). Hybrid artificial bee colony algorithm for neural network training. *Evolutionary Computation (CEC), 2011 IEEE Congress on*, 84-88.
- Parker D., (1985) Learning -logic: Technicle report TR-47.
- Peng, X., Venayagamoorthy, G. K., & Corzine, K. A. (2007). Combined Training of Recurrent Neural Networks with Particle Swarm Optimization and Backpropagation Algorithms for Impedance Identification. *Swarm Intelligence Symposium, 2007. SIS 2007. IEEE*, 9-15.
- Popescu, M.-C., Balas, V. E., Perescu-Popescu, L., & Mastorakis, N. (2009). Multilayer perceptron and neural networks. *WSEAS Trans. Cir. and Sys*WSEAS Trans. Cir. and Sys, 8(7), 579-588.
- Prechelt, I. (1994). A set of neural network benchmark problem and benchmarking rule (Web :<ftp://ira.uka.de/pub/neuron/problem1.tar.gz>). *Technical report 21/94*, 1-4.
- Pyle, D. (1999). *Data preparation for data mining* (Vol. 1): Morgan Kaufmann.
- Quinlan, J. R. (1987). Simplifying Decision Trees. *Man-Machine Studies* 27, 221-234.
- Quinlan, J. R., Compton, P. J., Horn, K. A., & Lazurus, L. (1986). Inductive knowledge acquisition: A case study. *Second Australian Conference on Applications of Expert Systems. Sydney*.
- Rajabioun, R. (2011). Cuckoo optimization algorithm. *Applied soft computing*, 11(8), 5508-5518.
- Ransing, M. R. (2002). Issues in learning Cause and effect Relationships from Example: with particular emphasis on casting process. *University fo Wales Swansea*, .
- Rehman, M., Mohd Nawi, N., & Ghazali, M. I. (2012). Predicting noise-induced hearing loss (NIHL) and hearing deterioration index (HDI) in Malaysian industrial workers using GDAM algorithm. *Journal of Engineering and Technology*, 3.
- Rehman, M. Z., Nazri, M. N. (2011). The Effect of Adaptive Momentum in Improving the Accuracy of Gradient Descent Back Propagation Algorithm on Classification Problems. *CCIS Journal of Software Engineering and Computer Systems. Springer Heidelberg*, 179 (6), 380-390.

- Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1985). Learning Internal Representations by error propagation. *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*
- Rummelhart, D., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature*, 323(9), 533-535.
- Saad, E. W., Prokhorov, D. V., & Wunsch, D. C. (1998). Comparative Study of stock trend prediction using time delay, recurrent and probabilistic neural networks. *Neural Networks, IEEE Transactions on*, 9(6), 1456-1470.
- Shereef, I. K., & Baboo, S. S. A New Weather Forecasting Technique using Back Propagation Neural Network with Modified Levenberg-Marquardt Algorithm for Learning. *International Journal of Computer Science*, 8.
- Smith, J. W., Everhart, J. E., Dickson, W. C., Knowler, W. C., & Johannes, R. S. . (1988). Using the ADAP learning algorithm to forecast the onset of diabetes mellitus. Symposium on Computer Applications and Medical Care. 261-265, IEEE Computer Society Press.
- Sudarshan Nandy, P. P. S., Achintya Das. (2012). Analysis of a Nature Inspired Firefly Algorithm based Back-propagation Neural Network Training. *International Journal of Computer Applications (0975 – 8887)*, 43(22).
- Mamat, R., Herawan, T., & Deris, M. M. (2013). MAR: Maximum Attribute Relative of soft set for clustering attribute selection. *Knowledge-Based Systems*, 52, 11-20.
- Sun, Y. J., Zheng, S., Miao, C. X., & Li, J. M. (2007). Improved BP Neural Network for Transformer Fault Diagnosis. *Journal of China University of Mining and Technology* 17 (1), 138-142.
- Sutton, R. S. (1986). Two problem with Back Propagation and other steepest descent learning procedure for network. *proceeding of the annual Conference of the Cognitive Science Society*, 823-831.
- Swanston, D. J., Bishop, J. M., & Mitchell, R. J. (1994). Simple adaptive momentum: New algorithm for training multilayer Perceptrons. *IEEE Journal of Electronic Letters* 30 (18), 1498-1500.
- Tanoto, Y., Ongsakul, W., & OP Marpaung, C. (2011). Levenberg-Marquardt Recurrent Networks for Long-Term Electricity Peak Load Forecasting.

- TELKOMNIKA (Telecommunication, Computing, Electronics and Control)*, 9(2), 257-266.
- Toha, S. F., & Tokhi, M. O. (2008). MLP and Elman recurrent neural network modelling for the TRMS. *Cybernetic Intelligent Systems, 2008. CIS 2008. 7th IEEE International Conference on*, 1-6.
- Tuba, M., Subotic, M., & Stanarevic, N. (2011). Modified cuckoo search algorithm for unconstrained optimization problems. *Proceedings of the 5th European conference on European computing conference*, 263-268.
- Übeyli, E. D. (2008b). Recurrent neural networks employing Lyapunov exponents for analysis of doppler ultrasound signals. *Expert Systems with Applications*, 34(4), 2538-2544.
- Übeyli, E. D. (2008c). Recurrent neural networks with composite features for detection of electrocardiographic changes in partial epileptic patients. *Computers in Biology and Medicine*, 38(3), 401-410.
- Valian, E., Mohanna, S., & Tavakoli, S. (2011). IMPROVED CUCKOO SEARCH ALGORITHM FOR FEEDFORWARD NEURAL NETWORK TRAINING. *International Journal of Artificial Intelligence & Applications*, 2(3).
- Vazquez, R. A. (2011). Training spiking neural models using cuckoo search algorithm. *Evolutionary Computation (CEC), 2011 IEEE Congress on*, 679-686.
- Walton, S., Hassan, O., Morgan, K., & Brown, M. (2011). Modified cuckoo search: a new gradient free optimisation algorithm. *Chaos, Solitons & Fractals*, 44(9), 710-718.
- Wang, H., Gao, Y., Xu, Z., & Xu, W. (2011). *An recurrent neural network application to forecasting the quality of water diversion in the water source of Lake Taihu*. Paper presented at the Remote Sensing, Environment and Transportation Engineering (RSETE), 2011 International Conference on.
- Wang, X. G., Tang, Z., Tamura, H., Ishii, M., & Sun, W. D. (2004). An improved backpropagation algorithm to avoid the local minima problem. *Neurocomputing*, 56 455-460.
- Watkins, D. (1997). Clementine's Neural Network Technical Overview. Technical Report.

- Werbos, P. (1974). Beyond regression: new tools for prediction and analysis in the behavioral sciences. Harvard University.
- Wilamowski, B., Cotton, N., & Hewlett, J. (2007). Neural network trainer with second order learning algorithms. *Intelligent Engineering Systems, 2007. INES 2007. 11th International Conference on*, 127-132.
- Wolberg, W. H., & Mangasarian, O. L. (1990). Multisurface method of pattern separation for medical diagnosis applied to breast cytology. *National Academy of Sciences*, 9193-9196.
- Yu, X. H et al. . (1995). Dynamic learning rate Optimization of the Back Propagation algorithm. *IEEE Trans. Neural Network*, 6(669-677).
- Xin-She, Y., & Deb, S. (2009). Cuckoo Search via Levy flights. *Nature & Biologically Inspired Computing, 2009. World Congress on*, 210-214.
- Xingbo, S., & Pingxian, Y. (2008, 18-20 Oct. 2008). *BP Neural Networks with Improved Activation Function and Its Application in the Micrographs Classification*. Paper presented at the Natural Computation, 2008. ICNC '08. Fourth International Conference on.
- Xue, Q., Yun, F., Zheng, C., Liu, Y., Wei, Y., Yao, Y., et al. (2010). Improved LMBP algorithm in the analysis and application of simulation data. *Computer Application and System Modeling (ICCA SM), 2010 International Conference on*, 6, 545--547.
- Yan, J., Cao, H., Wang, J., Liu, Y., & Zhao, H. (2009). *Levenberg-Marquardt algorithm applied to forecast the ice conditions in Ningmeng Reach of the Yellow River*. Paper presented at the Natural Computation, 2009. ICNC'09. Fifth International Conference on.
- Yang, X.-S. (2010a). Firefly algorithm, Levy flights and global optimization *Research and Development in Intelligent Systems XXVI* (pp. 209-218): Springer.
- Yang, X.-S. (2010b). *Nature-inspired metaheuristic algorithms*: Luniver Press.
- Yang, X.-S. (2011). Bat algorithm for multi-objective optimisation. *International Journal of Bio-Inspired Computation*, 3(5), 267-274.
- Yang, X.-S., & Deb, S. (2013). Multiobjective cuckoo search for design optimization. *Computers & Operations Research*, 40(6), 1616-1624.

- Yang, X.-S., Deb, S., & Fong, S. (2011). Accelerated particle swarm optimization and support vector machine for business optimization and applications. *Networked digital technologies*, 53-66.
- Yang, X.-S., & Gandomi, A. H. (2012). Bat algorithm: a novel approach for global engineering optimization. *Engineering Computations*, 29(5), 464-483.
- Yang, X. S. (2010). *Engineering optimization: an introduction with metaheuristic applications*: Wiley. com.
- Ye, Y. C. (2001). Application and Practice of the Neural Networks. *Taiwan: Scholars Publication*.
- Yu, C. C., & Liu, B. D. . (2002). A Backpropagation algorithm with adaptive learning rate and momentum coefficient. *IJCNN*, 1218 – 1223, Honolulu.
- Yuemei, X., & Hong, Z. (2009). Study on the Improved BP Algorithm and Application. *Proceedings of the Information Processing, 2009. APCIP 2009. Asia-Pacific Conference on*, 7-10.
- Zhang, J.-R., Zhang, J., Lok, T.-M., & Lyu, M. R. (2007). A hybrid particle swarm optimization–back-propagation algorithm for feedforward neural network training. *Applied Mathematics and Computation*, 185(2), 1026-1037.
- Zhang, Y., Wang, L., & Wu, Q. (2012). Modified Adaptive Cuckoo Search (MACS) algorithm and formal description for global optimisation. *International Journal of Computer Applications in Technology*, 44(2), 73-79.
- Zheng, H., Meng, W., & Gong, B. (1992). Neural Network and its Application on Machine fault Diagnosis. *ICSYSE 1992*, 576-579.
- Zheng, H., & Zhou, Y. (2012). A novel cuckoo search optimization algorithm based on Gauss distribution. *Journal of Computational Information Systems*, 8(10), 4193-4200.
- Zhou, Y., & Zheng, H. (2013). A novel complex valued cuckoo search algorithm. *The Scientific World Journal*, 2013.
- Zaweri Y.H And Althoefer L.D.,(2005). Stability analysis of the three term Back propoagation algorithm. *Neural network* V18(10) pp.1341-1347.