A Novel Hybrid Learning Algorithm for Artificial Neural Networks

Ranadhir Ghosh, BE (Comp. Sc.), MIT

School of Information Technology
Faculty of Engineering and Information Technology
Griffith University - Gold Coast Campus

Submitted in fulfillment of the requirements of the degree of Doctor of Philosophy
December 2002

Abstract

Last few decades have witnessed the use of artificial neural networks (ANN) in many real-world applications and have offered an attractive paradigm for a broad range of adaptive complex systems. In recent years ANN have enjoyed a great deal of success and have proven useful in wide variety pattern recognition or feature extraction tasks. Examples include optical character recognition, speech recognition and adaptive control to name a few. To keep the pace with its huge demand in diversified application areas, many different kinds of ANN architecture and learning types have been proposed by the researchers to meet varying needs.

A novel hybrid learning approach for the training of a feed-forward ANN has been proposed in this thesis. The approach combines evolutionary algorithms with matrix solution methods such as singular value decomposition, Gram-Schmidt etc., to achieve optimum weights for hidden and output layers. The proposed hybrid method is to apply evolutionary algorithm in the first layer and least square method (LS) in the second layer of the ANN. The methodology also finds optimum number of hidden neurons using a hierarchical combination methodology structure for weights and architecture. A learning algorithm has many facets that can make a learning algorithm good for a particular application area. Often there are trade offs between classification accuracy and time complexity, nevertheless, the problem of memory complexity remains. This research explores all the different facets of the proposed new algorithm in terms of classification accuracy, convergence property, generalization ability, time and memory complexity.

Statement of Originality

This work has not previously been submitted for a degree or diploma in any university. To the best of my knowledge and belief, this thesis contains no material previously published or written by another person except where due acknowledgement is made in the thesis itself.

Signature: ……………………………………….

(RANADHIR GHOSH)

Date: …………………………………………

Acknowledgement

I would like to express my gratitude to all those who gave me the opportunity to complete this thesis. I want to thank the School of Information Technology, Griffith University, Gold Coast Campus for giving me the opportunity to commence this thesis in the first instance, to do the necessary research work and to provide all the resources that were required.

I am deeply indebted to my supervisor Dr Brijesh Verma whose help, stimulating suggestions and encouragement helped me in all the time of research for and writing this thesis. Also I would like to thank my associate supervisor Dr Vallipuram Muthukkumarasamy and faculty member of our school Dr Michael Blumenstein for their valuable suggestions many a times.

It will be incomplete without acknowledging the guidance of my dear brother Samir whose guidance helped me to overcome all the difficult phases during the period. Also, I would like to give my special thanks to my wife, my beloved Guddi whose patient love enabled me to complete this work.

Table of Contents

List of Figures vii

List of Tables ix

Abbreviation xii

1 Introduction 1

1.1 Background history 5
1.2 Biological Neuron 8
1.3 ANN taxonomy 9
1.3.1 Processing unit 10
1.3.2 Combination function 10
1.3.3 Activation function 11
1.3.3.1 Identity function $g(x) = x$ 12
1.3.3.2 Binary step function 12
1.3.3.3 Sigmoid function 13
1.3.3.4 Bipolar sigmoid function 13
1.3.4 Genetic algorithm 15
1.3.4.1 Outline of the basic genetic algorithm 16
1.3.4.2 Application of genetic algorithm 17
1.3.4.2.1 Genetic programming 18
1.3.4.2.2 Evolving neural networks 19
1.3.4.2.3 Other areas 19
1.4 Motivation and aim of the research 19
1.5 Organization of the Thesis 21

2 Literature Review 24

2.1 Algorithm notations 26
2.2 Algorithm Description 27
2.2.1 Hebb rule 27
2.2.2 Hamming net algorithm 28
2.2.3 Kohonen self-organizing maps 30
2.3 Gradient based learning 31
2.3.1 Frequency of updating weights in gradient descent algorithm 31
2.3.2 First Order gradient based learning 32
2.3.2.1 Perceptron rule 32
2.3.2.2 Pocket algorithm with ratchet 33
2.3.2.3 Delta rule 35
2.3.2.3.1 The μ-LMS algorithm 35
2.3.2.3.2 The α-LMS algorithm 36
2.3.2.4 MADALINE rule I 37
2.3.2.5 MADALINE rule II 38
2.3.2.6 Back propagation algorithm 39
2.3.2.7 Modification of EBP 41
2.3.2.7.1 Adding momentum factor 41
2.3.2.7.2 Exponential smoothing 43
2.3.2.7.3 Adaptive control 44
2.3.2.7.4 VM 46
2.3.2.8 Nonlinear backpropagation (NLEBP) 48
2.3.2.9 Successive over relaxation EBP algorithm 50
2.3.2.10 RP propagation 51
2.3.2.11 Quick propagation 53
2.3.3 Second order gradient based learning 55
2.3.3.1 Conjugate gradient descent 55
2.3.3.2 Quasi newton 56
2.3.3.3 Levenberg-Marquardt 58
2.3.3.3.1 Single output network 58
2.3.3.3.2 Small networks 58
2.3.3.3.3 Sum-squared error function 59

2.3.3.4 Radial bias Function (RBF) network 60
2.4 Evolutionary based learning algorithm 62
2.4.1 Exhaustive search 63
2.4.2 Genetic hill climber 63
2.4.3 L- systems 63
2.4.4 Stochastic hill climber 64
2.4.5 Simulated annealing 64
2.4.6 Genetic algorithm 64
2.4.6.1 Comparison between GA and gradient based training 66
2.4.6.2 Evolution of architecture using GA 69
2.4.6.3 Direct encoding Scheme 70
2.4.6.4 Indirect encoding scheme 71
2.4.6.5 Evolution of node transfer function 71
2.4.6.6 Prados's GenLearn method 71
2.4.6.7 GMU GANNET 72
2.4.6.8 EPNet 73
2.4.7 Hybrid Method 75
2.4.7.1 Layer wise EBP based Kaczmarz algorithm 75
2.4.7.2 Global Extended Kalman Filter Algorithm (GEKF) 77
2.4.7.3 Decoupled extended Kalman filter algorithm (DEKF) 78
2.4.7.4 Local linearized least squares algorithm (LLLS) 78
2.4.7.5 Precise linear sigmoidal network 79
2.4.7.6 Reduced complexity based on Jacobian deficiency 81
2.4.7.7 Linearly constrained least square method (LCLS) 84
2.4.7.8 Regularizer for recursive least squared algorithm 85
2.4.8 Problems still unresolved 87
2.4.8.1 Local minima 87
2.4.8.2 Computation of gradients 88
2.4.8.3 Paralysis problem 89
2.4.8.4 Generalization and convergence problem 89
2.4.8.5 Start state 90

3 Research Methodology 91

3.1 module details for finding weights 91
3.1.1 Common architecture details 92
3.1.2 Combination dynamics 93
3.1.2.1 T1 connection 98
3.1.2.2 T2 connection 99
3.1.2.3 T3 connection 100
3.1.3 Least square method 101
3.1.4 Efficiency order for the connection topology 105
3.1.4.1 Stopping criteria 108
3.1.5 Finding optimal number of hidden neurons (findArchitecture) 109
3.1.5.1 Combining EALS-T3 and findArchitecture modules 110
Experiment A (LI-EALS-T3) 111
Experiment B (BT-EALS-T3) 111

4 Experimental Results 113

4.1 IMPLEMENTATION DETAILS 114
4.1.1 Platform used 114
4.1.2 Data set details 115
4.2 Results 116
4.2.1 Experimental Results – Type I 116
4.2.1.1 Results with parameter variations 117
4.2.1.1.1 Results for EBP 118
4.2.1.1.2 Results for GAWLS 123
4.2.1.1.3 Results for EAWLS 127
4.2.1.1.4 Results for GALS-T1 132
4.2.1.1.5 Results for GALS-T2 134
4.2.1.1.6 Results for GALS-T3 138
4.2.1.1.7 Results for EALS-T1 142

4.2.1.1.8 Results for EALS-T2 145
4.2.1.1.9 Results for EALS-T3 148
4.2.1.2 Results with fixed parameters 153
4.2.1.2.1 Results from EBP 153
4.2.1.2.2 Results from EAWLS 153
4.2.1.2.3 Results from LI-EALS-T3 154
4.2.1.2.4 Results from BT-EALS-T3 154
4.2.2 Experimental results – Type II 155
4.2.2.1 Experiment 1 155
4.2.2.2 Experiment 2 155
4.2.2.3 Experiment 3 156
4.2.2.4 Experiment 4 157
4.2.2.5 Experiment 5 157
4.2.2.6 Experiment 6 159

5 Analysis and Discussion 161

5.1 Comparison analysis 161
5.1.1 Comparison of classification accuracy 161
5.1.2 Comparison of time complexity 164
5.1.3 Comparison of required number of hidden neurons 168
5.1.4 Comparison of memory complexity 169
5.1.5 Comparison of sensitivity analysis 170
5.2 Analysis of the special properties for the proposed learning algorithm 171
5.2.1 Analysis of convergence property for the new algorithm 171
5.2.2 Variations of hidden layer output for various weight range 175
5.2.3 Variations of inverse of R Matrix 176
5.2.4 Need of bias for the proposed algorithm (EALS-T3) 178
5.2.5 Is the hybrid nature of the proposed algorithm (EALS-T3) a combination of global and local search 179
5.2.6 Analysis of fitness breaking 181
5.2.7 Comparison of results for different T connections 183
5.2.8 Comparison of results for different GA strategies 185

6 Conclusion & Further research 187

6.1 Conclusion 187
6.2 Further research 190

Reference 191

List of Figures

Figure 1-1 Biological neuron 9
Figure 1-2 Artificial neuron 10
Figure 1-3 Identity function 12
Figure 1-4 Binary step function..... 12
Figure 1-5 Sigmoid function 13
Figure 1-6 Bipolar sigmoid function 13
Figure 1-7 Genetic operations on genotypes 17
Figure 1-8 Genetic programming as a solver 18
Figure 2-1 Hebb rule architecture 27
Figure 2-2 Hamming network 29
Figure 2-3 Kohonen self organizing network..... 30
Figure 2-4 Multi layer perceptron 40
Figure 2-5 Gaussian function used for RBF 61
Figure 2-6 Radial bias network 61
Figure 2-7 A flowchart for basic genetic algorithm 67
Figure 2-8 EPNet architecture 74
Figure 2-9 Local minima problem..... 88
Figure 2-10 Error surface considering only one-weight and bias vectors 89
Figure 2-11 Training set result for a function approaximator 90
Figure 2-12 Validation set result by the same network 90
Figure 3-1 Combination dynamics for findArchitecture and findWeight modules 91
Figure 3-2 A two layer ANN architecture for the proposed hybrid learning method..... 93
Figure 3-3 A general T (1/2/3) connection for GA and LA methods 97
Figure 3-4 T1- connection architecture 99
Figure 3-5 T2-connection architecture 100
Figure 3-6 T3-connection architecture 101
Figure 3-7 Flowchart for EALS-T3..... 110
Figure 5-1 Comparison of classification accuracy I 162
Figure 5-2 Comparison of classification accuracy II..... 162
Figure 5-3 Improvement of classification accuracy over EBP 163
Figure 5-4 Improvement of classification accuracy over EAWLS..... 163
Figure 5-5 Comparison of time complexity I 164
Figure 5-6 Comparison of time complexity II..... 164
Figure 5-7 Improvement of time complexity over EBP 165
Figure 5-8 Improvement of time complexity over EAWLS..... 165
Figure 5-9 Time complexity for the proposed algorithm for increasing number of hidden neurons 166
Figure 5-10 Time complexity for the proposed algorithm for increasing number of populations 166
Figure 5-11 Time complexity for the proposed algorithm for increasing number of training data pattern..... 167
Figure 5-12 Comparison of hidden neuron I 168
Figure 5-13 Comparison of hidden neuron II..... 168
Figure 5-14 Comparison of memory complexity 169
Figure 5-15 Increment of memory usage 170
Figure 5-16 Comparison of sensitivity analysis 171
Figure 5-17 Convergence of train RMS error 172
Figure 5-18 Convergence of test classification error..... 172
Figure 5-19 Stability of solution in local neighborhood..... 173
Figure 5-20 Convergence of RMS error with the lest square output..... 173
Figure 5-21 Convergence of classification error with the LS output 174
Figure 5-22 Comparisons on required # generations..... 174
Figure 5-23 Variations of hidden layer output for various range of weight initialization 175
Figure 5-24 Variations of inverse output for R Matrix with range element chosen randomly 176
Figure 5-25 Variations of inverse output for R Matrix with range element chosen was fixed 177
Figure 5-26 Effect of inverse function for number of different element 177
Figure 5-27 Solution region without the bias 179
Figure 5-28 Solution region with the bias 179

Figure 5-29 Percentage of improvement because of LS after EA output 180
Figure 5-30 Fitness distribution before LS 182
Figure 5-31 Fitness distribution after the LS 182
Figure 5-32 Rank profiling before/after the LS 183
Figure 5-33 Comparison of classification accuracy for different T connections 184
Figure 5-34 Comparison of time complexity for different T connections 184
Figure 5-35 Comparison of classification accuracy based on different genetic algorithm strategies 185
Figure 5-36 Comparison of classification accuracy based on different genetic algorithm strategies 186

List of Tables

Table 3-1 Variations of strategies depending on the GA methods 94
Table 3-2 Three different connection strategies for calling LS from GA/EA 97
Table 4-1 Description for all the algorithms 113
Table 4-2 Data set information........ 115
Table 4-3 EBP results with different #iteration and #hidden neurons for XOR dataset........ 118
Table 4-4 EBP results with different #iteration and #hidden neurons for 10 bit odd parity dataset........ 119
Table 4-5 EBP results with different #iteration and #hidden neurons for handwriting characters (CEDAR) dataset 120
Table 4-6 EBP results with different #iteration and #hidden neurons for breast cancer (Wisconsin) dataset 120
Table 4-7 EBP results with different #iteration and #hidden neurons heart disease (Cleveland) dataset........ 121
Table 4-8 EBP results with different #iteration and #hidden heart disease (Hungary) dataset121
Table 4-9 EBP results with different #iteration and #hidden heart disease (Switzerland) dataset 122
Table 4-10 GAWLS results with different #population and #hidden neurons for XOR datase123
Table 4-11 GAWLS results with different #population and #hidden neurons for 10 bit odd parity dataset 124
Table 4-12 GAWLS results with different #population and #hidden neurons for handwriting characters (CEDAR) dataset 124
Table 4-13 GAWLS results with different #population and #hidden neurons for breast cancer (Wisconsin) dataset 125
Table 4-14 GAWLS results with different #population and #hidden neurons heart disease (Cleveland) dataset 125
Table 4-15 GAWLS results with different #population and #hidden heart disease (Hungary) dataset........ 126
Table 4-16 GAWLS results with different #population and #hidden heart disease (Switzerland) dataset........ 127
Table 4-17 EAWLS results with different #population and #hidden neurons for XOR datase 127
Table 4-18 EAWLS results with different #population and #hidden neurons for 10 bit odd parity dataset 128
Table 4-19 EAWLS results with different #population and #hidden neurons for handwriting characters (CEDAR) dataset 129
Table 4-20 EAWLS results with different #population and #hidden neurons for breast cancer (Wisconsin) dataset 129
Table 4-21 EAWLS results with different #population and #hidden neurons heart disease (Cleveland) dataset 130
Table 4-22 EAWLS results with different #population and #hidden heart disease (Hungary) dataset........ 130
Table 4-23 EAWLS results with different #population and #hidden heart disease (Switzerland) dataset........ 131
Table 4-24 GALS-T1 results with different #population and #hidden neurons for XOR datase 132
Table 4-25 GALS-T1 results with different #population and #hidden neurons heart disease (Cleveland) dataset 132
Table 4-26 GALS-T1 results with different #population and #hidden heart disease (Hungary) dataset........ 133
Table 4-27 GALS-T1 results with different #population and #hidden heart disease (Switzerland) dataset 134
Table 4-28 GALS-T2 results with different #population and #hidden neurons for XOR datase 135
Table 4-29 GALS-T2 results with different #population and #hidden neurons for breast cancer (Wisconsin) dataset 135
Table 4-30 GALS-T2 results with different #population and #hidden neurons heart disease (Cleveland) dataset 136
Table 4-31 GALS-T2 results with different #population and #hidden heart disease (Hungary)

dataset 136
Table 4-32 GALS-T2 results with different #population and #hidden heart disease (Switzerland) dataset 137
Table 4-33 GALS-T3 results with different #population and #hidden neurons for XOR datase 138
Table 4-34 GALS-T3 results with different #population and #hidden neurons for 10 bit odd parity dataset 138
Table 4-35 GALS-T3 results with different #population and #hidden neurons for handwriting characters (CEDAR) dataset 139
Table 4-36 GALS-T3 results with different #population and #hidden neurons for breast cancer (Wisconsin) dataset 140
Table 4-37 GALS-T3 results with different #population and #hidden neurons heart disease (Cleveland) dataset 140
Table 4-38 GALS-T3 results with different #population and #hidden heart disease (Hungary) dataset 141
Table 4-39 GALS-T3 results with different #population and #hidden heart disease (Switzerland) dataset 141
Table 4-40 EALS-T1 results with different #population and #hidden neurons for XOR datase 142
Table 4-41 EALS-T1 results with different #population and #hidden neurons breast cancer (Wisconsin) dataset 143
Table 4-42 EALS-T1 results with different #population and #hidden neurons heart disease (Cleveland) dataset 143
Table 4-43 EALS-T1 results with different #population and #hidden heart disease (Hungary) dataset 144
Table 4-44 EALS-T1 results with different #population and #hidden heart disease (Switzerland) dataset 145
Table 4-45 EALS-T2 results with different #population and #hidden neurons for XOR datase 145
Table 4-46 EALS-T2 results with different #population and #hidden neurons for breast cancer (Wisconsin) dataset 146
Table 4-47 EALS-T2 results with different #population and #hidden neurons heart disease (Cleveland) dataset 147
Table 4-48 EALS-T2 results with different #population and #hidden heart disease (Hungary) dataset 147
Table 4-49 EALS-T2 results with different #population and #hidden heart disease (Switzerland) dataset 148
Table 4-50 EALS-T3 results with different #population and #hidden neurons for XOR datase 149
Table 4-51 EALS-T3 results with different #population and #hidden neurons for 10 bit odd parity dataset 149
Table 4-52 EALS-T3 results with different #population and #hidden neurons for handwriting characters (CEDAR) dataset 150
Table 4-53 EALS-T3 results with different #population and #hidden neurons for breast cancer (Wisconsin) dataset 150
Table 4-54 EALS-T3 results with different #population and #hidden neurons heart disease (Cleveland) dataset 151
Table 4-55 EALS-T3 results with different #population and #hidden heart disease (Hungary) dataset 152
Table 4-56 EALS-T3 results with different #population and #hidden heart disease (Switzerland) dataset 152
Table 4-57 EBP results with fixed parameters 153
Table 4-58 EAWLS results with fixed parameters (population 60) 153
Table 4-59 LI-EALS-T3 results with fixed parameters 154
Table 4-60 BT-EALS-T3 results with fixed parameters 154
Table 4-61 Experiment 1 155
Table 4-62 Experiment 2 156
Table 4-63 Variation of results for different ranges with matrix element chosen randomly 156
Table 4-64 Experiment 4 157
Table 4-65 Experiment 5 158

Table 4-66 Experiment 6 160

Abbreviation

AI :	Artificial intelligence
ANN :	Artificial neural network
BT-EALS-Tx :	Binary architecture search with EALS weight updating using Tx connection; $x \in \{1,2,3\}$
BT-GALS-Tx :	Binary architecture search with GALS weight updating using Tx connection; $x \in \{1,2,3\}$
CI :	Computational intelligence
EA :	Evolutionary algorithm
EALS :	Evolutionary algorithm with least square
EAWLS :	Evolutionary algorithm without least square
EBP :	Error back propagation
EP :	Evolutionary programming
GA :	Genetic algorithm
GALS :	Genetic algorithm with least square
GAWLS :	Genetic algorithm without least square
GV1 :	Is same as LI-EALS-T3
GV2 :	Is same as BT-EALS-T3
LI-EALS-Tx :	Linear architecture search with EALS weight updating using Tx connection; $x \in \{1,2,3\}$
LI-GALS-Tx :	Linear architecture search with GALS weight updating using Tx connection; $x \in \{1,2,3\}$
LS :	Least square
MLP :	Multi layered perceptron
NP :	Non polynomial