Performance Enhancement of Data Classification using Selectively Cloned Genetic Algorithm for Neural Network

Devinder Kaur[*]
EECS Department, University of Toledo, 2801 W. Bancroft
Toledo, OH 43606, USA
E-mail: devinder.kaur@utoledo.edu

Praneeth Nelapati
EECS Department, University of Toledo, 2801 W. Bancroft
Toledo, OH 43606, USA
E-mail: pnelapa@rockets.utoledo.edu

Received: 10-10-2009
Accepted: 15-10-2010

Abstract

The paper demonstrates performance enhancement using selective cloning on evolutionary neural network over the conventional genetic algorithm and neural back propagation algorithm for data classification. Introduction of selective cloning improves the convergence rate of the genetic algorithm without compromising on the classification errors. The selective cloning is tested on five data sets. The Iris data problem is used as a bench-mark to compare the selective cloning technique with the conventional GA and the back-propagation algorithm. For comparative analysis, same neural network architecture is used for both the back propagation and the genetic algorithms. The selective cloning approach is based on the schema theorem. By using selective cloning, it has been shown that GA is 27.78% more efficient than the conventional GA and 83.33% more efficient than the back propagation approach. The results of selective cloning on other data sets are also discussed.

Keywords: Neural evolution, selective cloning, performance enhancement

1. Introduction

Data Classification and Pattern recognition systems are of great significance in the real world. A neural network is a very efficient tool to classify the patterns appropriately. The neural network stores the knowledge in the form of interconnection strength (weights) between the neurons. The architecture of the neural network is guided by the classification problem and then the network is trained using back propagation to adjust the weights of the network. The neural network can be trained with a back propagation algorithm or an evolutionary technique. However the back propagation algorithm may converge to a set of sub optimal solutions which it cannot escape, so it cannot always guarantee an optimal solution [1]. On the other hand, the evolutionary techniques, for instance genetic algorithm, using crossover and mutation operators ensure that the obtained solution is an optimal one. For this reason a neural network trained with genetic algorithm is a better alternative for the data classification problem. The genetic algorithm is based

Published by Atlantis Press

on the Darwinian theory of survival of the fittest. There is a greater probability that parents with higher fitness are given the opportunity to reproduce, thus improving the fitness of the population in successive generations. However, since the parents are picked up randomly there is a small probability that certain offsprings may have lower fitness than their parents. This may increase the number of generations before a required level of fitness in the population is achieved. The selective cloning technique being proposed in the paper ensures that all the offsprings in the successive generations have a higher or equal fitness than their parents. The selective cloning ensures that the useful 'Schema' instances (the instances that are required to be present in a chromosome population for the group to have higher fitness) are present in all the offsprings.

2. Data Classification

The proposed technique is tried on the following five data sets. These data sets have been taken from the machine learning repository [2].

2.1. *Iris Data Set*

The Iris data set is a multivariate data set [3]. It is the collection of 50 samples each of three plant species namely Iris Setosa, Iris Versicolor and Iris Verginica. The data set consists of sepal length, sepal width, petal length and petal width of Iris flower as the inputs, and the corresponding species of the plant as output. Given the sepal and petal lengths and widths of a given plant, it is required to classify it correctly into one of the three species.

2.2. *Lenses Data Set:*

The Lenses data set is a collection of 24 instances of data which suggest the type of contact lenses to be prescribed. Based on the age of the patient (young, pre-presbyopic, presbyopic), spectacle prescription (myope or hypermetrope), astigmatic condition(true or false) and tear production rate (reduced or normal), the patient is suggested the type of contact lens (hard contact lenses, soft contact lenses or no contact lenses).

2.3. *Balloon Data Set:*

The Balloon data set is the result of a cognitive psychology experiment. It consists of 16 observations. The balloon condition (inflated or not) has to be determined, based upon the four parameters: color of the balloon (yellow or purple), size of the balloon (large or small), act (stretch or dip), and age of the person acting on the balloon (adult or child).

2.4. *Balance Scale Data Set:*

The data set has 51 instances. Every instance is classified as having the balance scale tip to the right, tip to the left, or be balanced. The attributes are the left weight of the balance, the left distance of the balance, the right weight of the balance, and the right distance of the balance. If (left-distance * left-weight) and (right-distance * right-weight) are equal, it is balanced.

2.5. *Cars Data Set:*

Car evaluation database was derived from a simple hierarchical decision model. A car has to be listed into one of the classes (unacceptable, acceptable, good, very good) based on its characteristic features: maintenance (v. high, high, med, low), number of doors (2, 3, 4, 5 and more), capacity (2, 4 or more), luggage boot (small, med, big) and safety (low, med, high).

These data sets are used to test the efficiency of the proposed selective cloning method over the conventional genetic algorithm. The following sections first model, the iris classification problem on neural network with back propagation and then evolve the neural network using genetic algorithm which is further enhanced by selectively cloned GA. The GA and selectively cloned GA are then modeled for other data sets.

3. Neural Network

Neural networks have been successfully used for data classification [4]. Various algorithms are developed for neural network learning including back propagation [5], [6], [7], [8] and genetic algorithms [9]. In this paper, we are proposing a new algorithm that improves the genetic algorithm. Fig. 1 shows the neural network for the Iris data classification. It has three layers; one input layer, a hidden layer and an output layer. The input layer has four neurons that correspond to the four plant characteristics of sepal length, sepal width, petal length and petal width. The hidden layer has three neurons. The output layer has three neurons corresponding to the three plant species, namely Iris Setosa, Iris Versicolor and Iris Verginica. The neural network of 4 data sets (Iris, Lenses, Balloons, Balance Scale) have four input parameters, while the cars data set has five input parameters. The Lenses and the Balance Scale data have three output parameters: hence they share neural network architecture similar to Iris data set. The Balloon data set has only two output parameters and hence the network architecture of balloon data set is modeled with four input, three hidden and two output neurons.

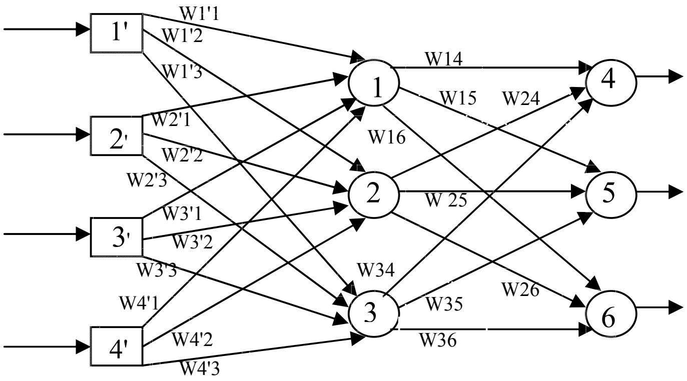


Fig 1: The Neural Network for the Iris Data Classification

A neural network stores the knowledge in the form of weights [10]. The weights are to be properly adjusted so that the Iris plants are classified properly. The activation function used for the neural network is the sigmoid function. The sigmoid function can functionally be represented as

$$\text{Sigmoid}(x) = 1/(1+e^{-x})$$

4. Neural Back Propagation

The back propagation is one of the most common algorithms to train a neural network [11], [12]. The back propagation algorithm for the Iris data problem uses the generalized delta rule including the momentum term to ensure better performance [13], [14], [15].

$$\Delta w_{jk}(p) = \beta \cdot \Delta w_{jk}(p-1) + \alpha \cdot y_j(p) \cdot \delta_k(p)$$

β is the momentum term

$\Delta W_{jk}(p-1)$ is the weight correction (between j'th neuron and k'th neuron present in the adjacent layers) in the p'th iteration

α is the learning rate parameter

$y_j(p)$ is the output of the j'th neuron in the p'th iteration

$\delta_k(p)$ is the error gradient of k'th neuron in the p'th-iteration

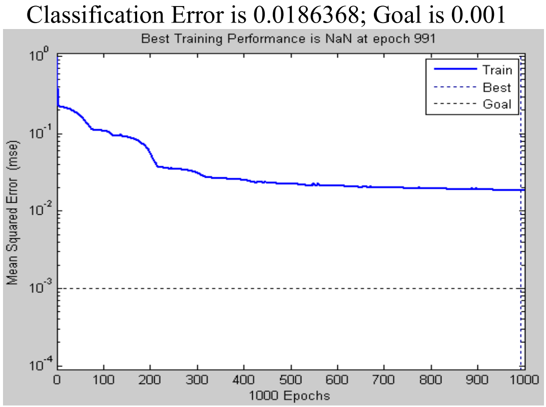


Fig. 2: Learning Rate of a Back Propagation

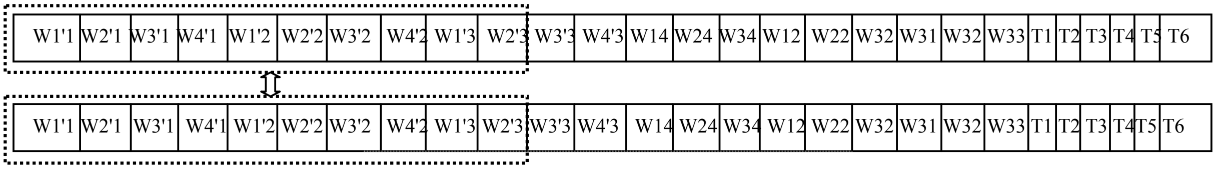
Fig. 2 shows the convergence of the back propagation algorithm [1]. After 1000 epochs it achieves classification error of 0.018. The slope of the convergence graph is very low, indicating that the learning of the network is very slow.

5. Genetic algorithm

In this section, the genetic algorithm is used to find an optimized neural network for the Iris data classification. Initially, the neural network of Fig. 1 is represented as a chromosome. Fig. 3 represents the chromosome for the neural network of Fig. 1. The neural network of Fig. 1 has 21 weights and 6 thresholds. T1, T2, T3, T4, T5, T6 are the thresholds of the neurons 1, 2, 3, 4, 5, 6 respectively. The chromosome which represents the neural network has 27 parameters or genes. The chromosomes for Lenses Data, Balloon data, Balance scale data and Cars data have 27, 23, 27 and 48 genes respectively. Initially a population of 100 such chromosomes will be generated. Using Genetic algorithms, the best chromosomes will be evolved,

W1'1	W2'1	W3'1	W4'1	W1'2	W2'2	W3'2	W4'2	W1'3	W2'3	W3'3	W4'3	W14	W24	W34	W12	W22	W32	W31	W32	W33	T1	T2	T3	T4	T5	T6

Fig 3: Chromosome for the Iris Data Classification



W1'1	W2'1	W3'1	W4'1	W1'2	W2'2	W3'2	W4'2	W1'3	W2'3	W3'3	W4'3	W14	W24	W34	W12	W22	W32	W31	W32	W33	T1	T2	T3	T4	T5	T6
W1'1	W2'1	W3'1	W4'1	W1'2	W2'2	W3'2	W4'2	W1'3	W2'3	W3'3	W4'3	W14	W24	W34	W12	W22	W32	W31	W32	W33	T1	T2	T3	T4	T5	T6

Fig. 4: Random crossover operator

which represent the best neural networks with weights that minimize the classification error [11], [16]. Chromosomes with high fitness have higher probability of getting selected as parents. Such selected parent chromosomes are operated with crossover operator and mutation operator to produce the next generation. Each successive generation of chromosomes is presented with the training data to see how many of them are classified correctly. Classification error with successive generations decreases. Fig. 5 shows the performance graph for the GA. The best trained network is obtained within 20 generations.

5.1. *Effects of Inappropriate Crossover Operator*

The crossover operator has a significant effect on the performance of the genetic algorithm. Improper crossover operator can destroy the essential schema instances being passed from one generation to the next. To observe this effect, a random crossover operator is chosen which implies the position and the length of crossover operator are chosen randomly. Fig. 4 shows the random crossover operator that interchanges the first ten genes of the parent chromosomes.

This crossover operator is used to generate the next generation of chromosomes for the iris data classification. Fig. 6 shows the destructive effect of this crossover operator. As shown in Fig.6, the initial generations had some best individuals that minimized the error to a low value. However, after the application of the crossover operator, the error started increasing and finally converged at 0.15.

One of the solutions to overcome this problem is to randomly check for various crossover lengths and positions and then decide the best crossover operator. This is computationally intensive. Even after finding the proper crossover operator, not all the offsprings have better fitness than their parent chromosomes. This is because the parent chromosomes are chosen randomly, where the probability of getting selected is directly proportional to the fitness of the chromosome. As a result some of the selected parents can be of lower fitness and thus result in less fit offspring. This results in degradation of the convergence rate. This drawback is overcome by selective cloning being proposed in section 6.

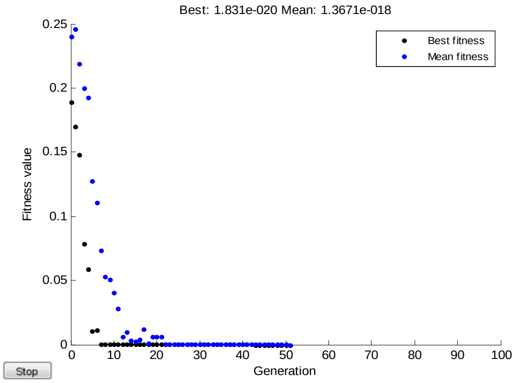


Fig 5: Convergence of a GA

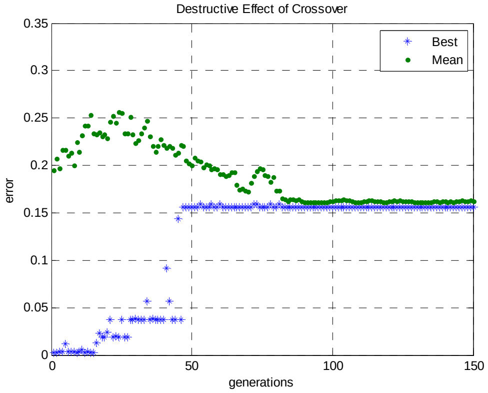


Fig. 6: Destructive effect of improper crossover operator

6. Selective Cloning

Selective cloning is a technique of selecting the parent over the offspring, when the fitness of the parent is superior to the fitness of the offspring. This approach can guarantee a definite and a faster convergence. The selective cloning is based on the schema theorem which is explained in 6.1. A schema is a set of bit strings of ones, zeros and asterisks, where each asterisk can assume the value of one or zero. For instance consider a schema A {0 * * 1}. The instances of schema A are {0001, 0101, 0011, 0111}. According to the schema theorem, one can predict the presence of a schema in next chromosome generation [1] [17] [18]. For faster convergence, the fittest schema traces must be increased from generation to generation. Let the fitness function be directly proportional to the instances of given schema 'A' in any generation. The negative effect of random crossover can be corrected using selective cloning. In case of selective cloning, when a random crossover length and position are chosen, the crossover operator can still break some of the schema instances of A and may leave the rest undisturbed. Thus, the offspring that contains the instances of A are of higher fitness and hence passed to the next generation. The offsprings that do not contain instances of A show poor fitness than the parents and hence instead of passing the offspring to the next generation, the parents are passed on to the next generation (cloning). This approach guarantees the increase in population containing instances of A from generation to generation and thus guarantees definite increase in fitness of the generation and hence convergence, irrespective of crossover operator length and positioning. Thus, by using this technique we can choose any crossover operator to obtain an optimal solution. The mutation operator ensures that search space is not narrowed down.

6.1. Mathematical background

This section explains mathematically why selective cloning ensures higher fitness of successive generations and is independent of the choice of random crossover operator.

Let 'm' be the total chromosome population in any generation.

The fitness criterion is directly proportional to the number of instances of schema H.

In the ith generation,

$m_H(i)$ = Total chromosome population containing instances from schema H.

$m_H'(i) = m - m_H(i)$ = Total chromosome population that do not contain instances from schema H.

Then by applying crossover and mutation operators, offsprings are generated.

$M_H(i)$ = Offspring chromosome population containing instances from schema H.

$M_H'(i) = m-M_H(i)$ = Offspring chromosome population that does not contain instances from schema H.

The crossover and mutation operators work in such a way that, some of the offspring of $m_H(i)$ population may lack instances from schema H and remaining possess the instances from schema H as given in equation 1. Fig. 7 shows the creation of new generation of offsprings.

$$m_H(i) \rightarrow a*M_H(i)+b*M_H'(i) \qquad (1)$$

where a+b =1, and a, b ϵ [0,1].

a = fraction of the total offspring population, where both the parents and the offsprings have the instances of schema H.

b = fraction of the total offspring population, where the offsprings lack the instances of schema H but parents have instances of schema H.

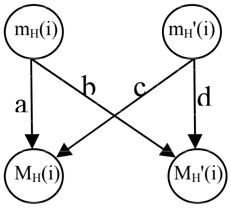


Fig 7: Creation of next generation of chromosomes

$$m_H'(i) \rightarrow c*M_H(i)+d*M_H'(i) \qquad (2)$$

where c+d = 1, and c, d ϵ [0,1].

c = fraction of the total offspring population, where the parents lack the instances of schema H and the offspring have the instances of schema H.

d = fraction of the total offspring population, where both the parents and the offspring lack the instances of schema H.
Now in selective cloning approach, the fitness of offspring is compared with the fitness of parents: Thus, b* $M_H'(i)$ population has less fitness when compared to its parents and hence parents are passed on to the next generation.

c*$M_H(i)$ population has higher fitness than their parents and hence the offspring replace their parents in the next generation.

Therefore using the selective cloning, instances of schema H in next generation is given by equation 3.

$$m_H(i+1)= m_H(i)+ c*M_H(i) \qquad (3)$$

The worst case scenario is that when c=0, which is a rare case, and then there is no increase in the instances of schema H in next generation as shown in equation 4.

$$m_H(i+1)= m_H(i) \qquad (4)$$

Whenever c >0, the instance of schema H in the next generation is given by equation 5.

$$m_H(i+1) \geq m_H(i) \qquad (5)$$

This ensures that selective cloning ensures fitness improvement and so convergence.

7. GA With Selective Cloning

In the selectively cloned GA, initially a population of 100 chromosomes is created. The fitness of these chromosomes is calculated, the fitness criterion being the classification error. By using the crossover and mutation, a next generation of chromosomes is created. The fitness of new set of chromosomes is calculated and compared with fitness of its parents. The fittest out of the two chromosomes is passed to the next generation. This process is repeated until the desired error level or the specified number of generations is achieved. As shown in Fig. 8, the selective cloning approach made the genetic algorithm converge within 12 generations which is faster than the conventional genetic algorithm, which converged in 18 generations as was shown in Fig.5. Hence GA with selective cloning is more appropriate for data classification. The choice of crossover operator has a little effect on the convergence of selectively cloned GA. The dependence of selectively cloned genetic algorithm on crossover operator is discussed in sub section 7.1.

7.1. Effect of the choice of Crossover operator on Selective Cloning

In this section, it has been shown that with selective cloning, the negative effect of the random crossover operator is minimized. It has mathematical foundation as shown in 6.1.

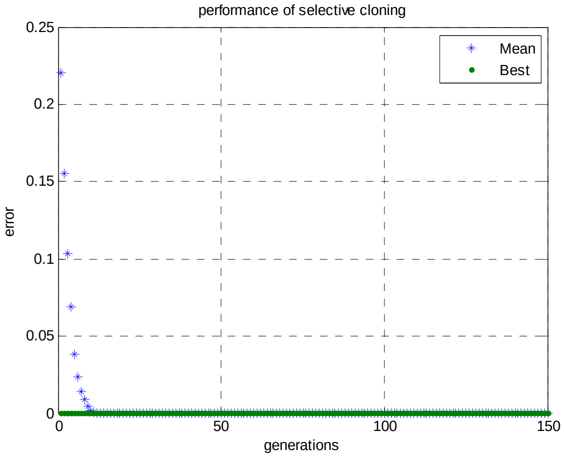


Fig 8: Performance of Selectively Cloned GA

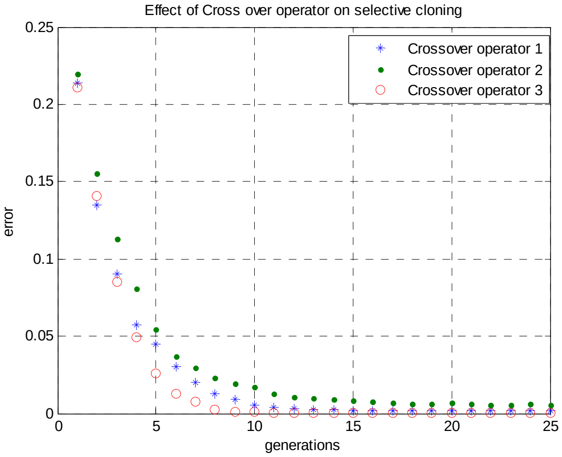


Fig 9: Convergence of a GA

Published by Atlantis Press

To demonstrate this, three different cross over operators are introduced individually in the selectively cloned genetic algorithm.

The first crossover operator is shown in Fig. 4. It interchanges first ten genes (w1 to w10) of the parent chromosomes. The second crossover operator interchanges the genes from w2 to w6 and w21 to w24 between the parents. The third crossover operator swaps genes from w16 to w22 between the parents. Using these crossover operators, three selectively cloned genetic algorithms are developed and their performance is compared. Fig. 9 shows the performance graph of the three selectively cloned genetic algorithms. The genetic algorithm with crossover operator number one converged in 12 generations. The genetic algorithm with crossover operator number two converged in 12 generations. The genetic algorithm with crossover operator number three converged in 9 generations. The difference in the number of generations required to converge is very small. Thus the performance of the selectively cloned genetic algorithm is minimally affected by the choice of the crossover operator.

7.2. Programming Model

In this section, the programming model of the selectively cloned genetic algorithm is discussed. The selectively cloned genetic algorithm is an extension to the genetic algorithm. Three steps that are used in the genetic algorithm are also used in the selectively cloned genetic algorithm.

1. Initial Population Generation: A set of randomly generated chromosomes represent the initial population.
2. Fitness Evaluation: The fitness of the chromosomes is evaluated based upon the required criterion. In the Iris data classification, the required criterion is to train the neural network to attain minimum classification error. Thus, the fitness criterion is chosen to be minimum mean square error.

3. Creation of Next Generation of Chromosomes: Using the crossover and mutation operators, a new set of chromosome population is created from the existing set of chromosomes. The size of this new population remains fixed from generation to generation. Both the genetic algorithm and the selectively cloned genetic algorithm use these three steps for finding solutions in the search space. Additionally the selectively cloned genetic algorithm compares the fitness of the parent chromosomes with the fitness of the offsprings. The chromosome with a better fitness is allowed to be passed to the next generation. This ensures that the required schema instances are present in all the chromosomes and hence ensures faster convergence.

8. Results

The efficiency of the back propagation, conventional GA and the selectively cloned genetic algorithm are compared for the given Iris classification. Table 1 shows the results of this comparison.

Table 1: Comparison of Selectively Cloned Neural Network, Genetic Algorithm and Conventional Neural Network Back Propagation Algorithm for Iris Data Classification.

Technique Used	Number of epochs required for convergence	Best error that could be achieved at the end of convergence	Convergence efficiency in epochs compared to back propagation	Time Required for the convergence	Time efficiency when compared to back propagation
Back Propagation Algorithm	**1000 epochs**	**0.018**	**-**	**6.739 sec**	**-**
Genetic Algorithm	**18 generations**	**$1.9\text{x}10^{-21}$**	**55.55%**	**1.45 sec**	**78.4%**
Genetic Algorithm with selective cloning	**12 generations**	**0**	**83.33%**	**0.272 sec**	**95.96%**

Published by Atlantis Press

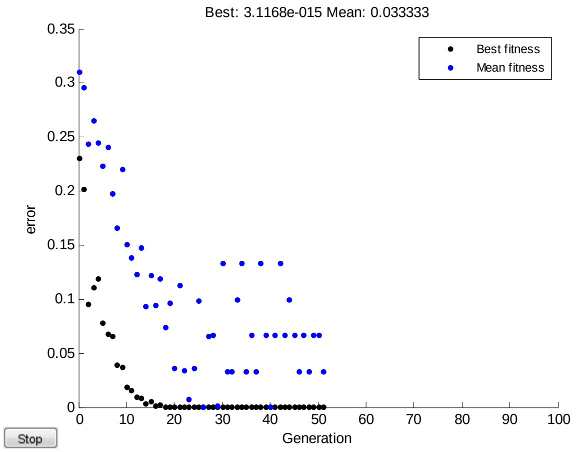


Fig 10 (a) Performance of GA for Balance Scale data set

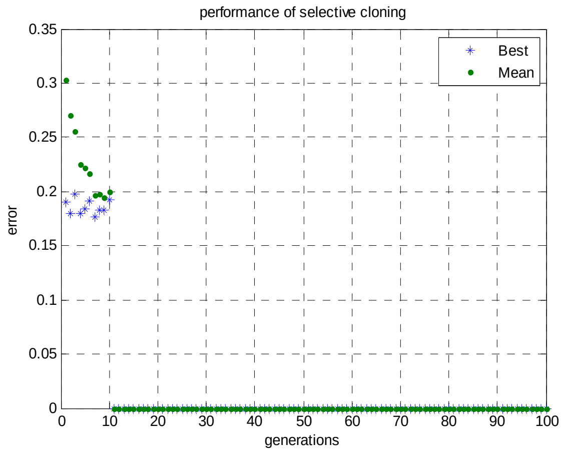


Fig 10 (b) Performance of Selectively Cloned GA for Balance Scale data set

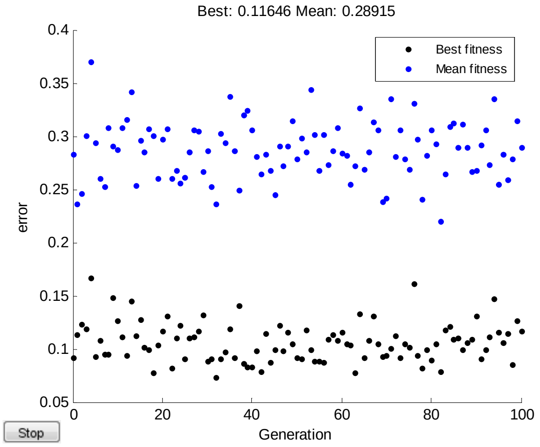


Fig 11(a) Performance of GA for Balloon data set

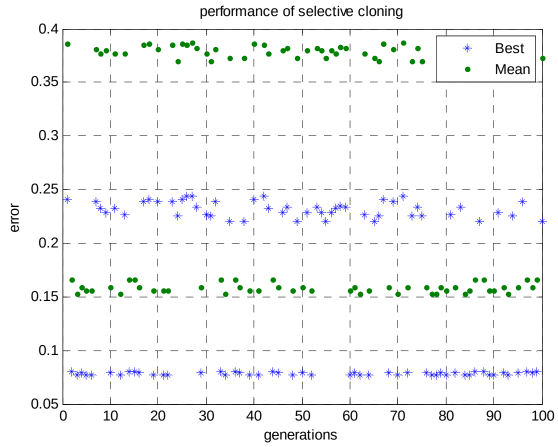


Fig 11 (b) Performance of Selectively Cloned GA for Balloon data set

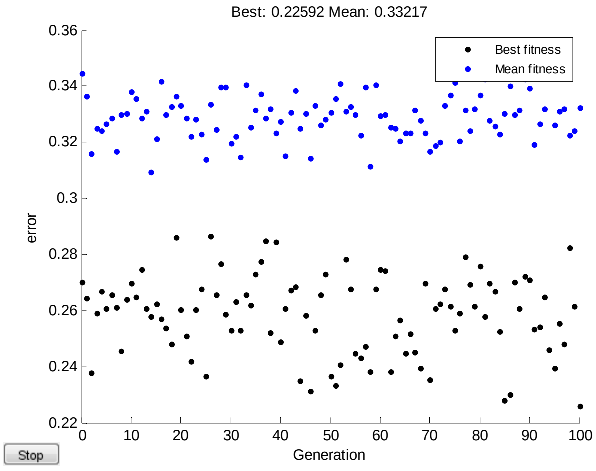


Fig 12(a) Performance of GA for Cars data set

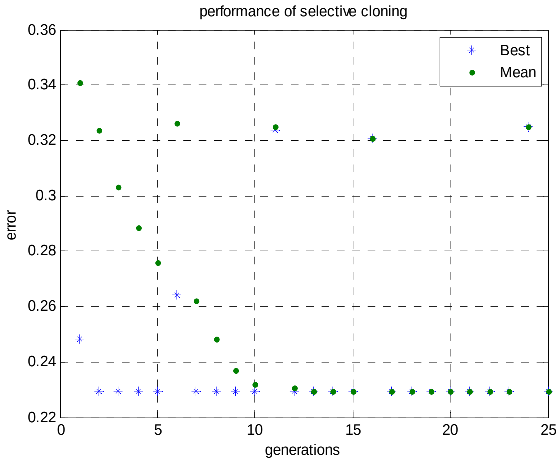


Fig 12(b) Performance of Selectively Cloned GA for Cars data set

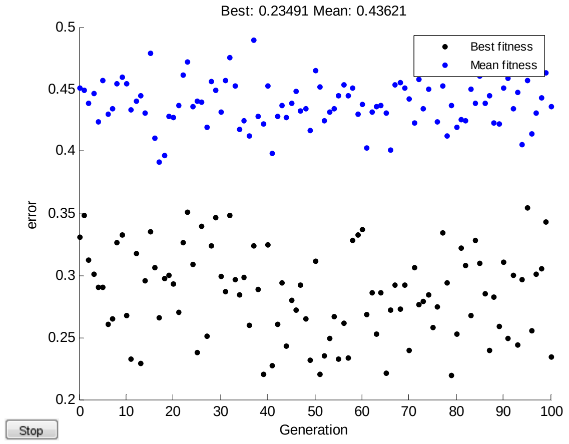


Fig 13 (a) Performance of GA for Lenses data set

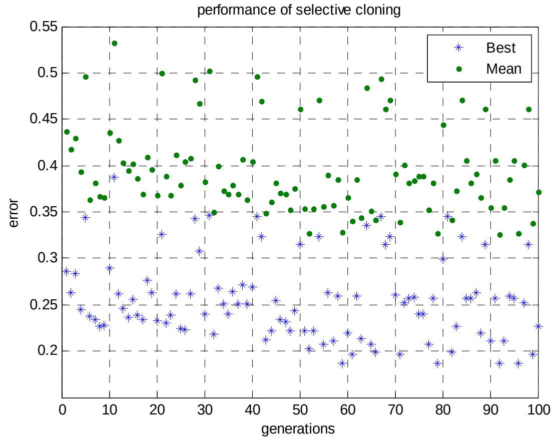


Fig 13 (b) Performance of Selectively Cloned GA for Lenses data set

The back propagation has a final error of 0.018. The conventional GA converged in 18 generations, with an error of 1.9*10-21.The selectively cloned GA was far superior and converged in 12 generations with an error of almost 0. Thus selectively cloned GA has a convergence efficiency of 83.33% which is far superior to GA which is 55.55% efficient in comparison to back propagation.

Considering the clock time required for convergence, the back propagation converged in 6.7 seconds, while the conventional GA converged in 1.45 seconds and selectively cloned GA converged in 0.272 seconds.

For conventional GA:

Clock time saved with respect to back propagation = 6.739-1.45=5.289 sec. Time efficiency of conventional GA = (5.289/6.739)% = 78.4%. Similarly, time efficiency of selectively cloned GA = 95.96%.

Thus, the selectively cloned GA is 27.78% faster than the conventional GA.

8.1. Other Data Sets

Balance Scale Data Set:

Fig. 10a shows the performance of the conventional and Fig. 10b shows the performance of selectively cloned genetic algorithms for this data set. The selectively cloned GA converged in 10 generations with a mean error value of zero, while the conventional GA converged in 25 generations with a mean error value of 0.1.

Balloon Data Set:

Fig. 11a and Fig. 11b show the performance of the conventional and selectively cloned genetic algorithms for Balloon data set. The selectively cloned GA has similar mean performance as the conventional GA for the Balloon data set. However, the selectively cloned GA had best individuals in the error range of 0.08 to 0.16, while the conventional GA had best individuals oscillated with a mean error value of 0.12.

Cars Data Set:

Fig. 12a and Fig. 12b show the performance of the conventional and selectively cloned genetic algorithms for car data set. The selectively cloned GA converged in 10 generations with an error of 0.23, while the conventional GA had an error of 0.33 at the end of 100 generations.

Lenses Data Set:

Fig. 13a and Fig. 13b show the performance of the conventional and selectively cloned genetic algorithms for this data set. The selectively cloned GA had a mean error of 0.32 and the conventional GA had a mean error of 0.45 at the end of 100 generations. However, for selectively cloned GA, the mean error reduced from 0.45 to 0.32, while that of conventional GA stayed at 0.45

9. Conclusion

It has been shown that the selectively cloned GA algorithm improves the convergence time and reduces the error for the data classification problems. The selectively cloned GA improves the instances of schema in successive generations of GA and reduces the negative effects of random crossover operators.

10. References

1. Michael Negnevitsky, *Artificial Intelligence-A Guide to Intelligent Systems*, 2nd edition, (Addison Wesley, 2005).
2. UCI machine learning repository(www.ics.uci.edu/~mlearn/)
3. Fisher R.A., *The use of multiple measurements in taxonomic problems*, Annual Eugenics, 7, Part II. (1936), pp. 179-188.
4. S. Haykin, *Neural networks - a comprehensive foundation*, 2nd edition, (Prentice Hall, USA, 1999).
5. Bo Yang, Xiaohong Su and Yadong Wang, *Back propagation based on selective attention for fast convergence of training neural network*, in Fifth World Congress on Intelligent Control and Automation (WCICA). Vol. 3 (2004), pp. 2009-2013.
6. Ming Chen and Zhengwei Yao, *Classification Techniques of Neural Networks Using Improved Genetic Algorithms*, in IEEE International Conference on Genetic and Evolutionary Computing. (2008), pp. 115 - 119.
7. R.A. Dilruba, N. Chowdhury, F.F. Liza and C.K. Karmakar, *Data Pattern Recognition using Neural Network with Back-Propagation Training,* in International Conference on Electrical and Computer Engineering, ICECE (Dec 2006) , pp. 451 – 455.
8. A. Fazih, J. Chedjou, K. Kyamakya, *Cellular Neural Networks-Based Genetic Algorithm for Optimizing the Behavior of an Unstructured Robot*, in International Journal of Computational Intelligence Systems, Vol. 2, Issue 2, June 2009, pp. 124-131.
9. Colin R. Reeves and Jonathan E. Rowe, *Genetic algorithms: principles and perspectives: a guide to GA theory,* (Kluwer Academic Publishers, 2003).
10. Gowrishankar and P.S. Satyanarayana, *Neural Network Based Traffic Prediction for Wireless Data Networks*, in International Journal of Computational Intelligence Systems (IJCIS), Vol.1 issue 4 (2008), pp. 379-389.
11. Takeshi Kamio, Shinichiro Tanaka and Mititada Morisue, *Back-propagation Algorithm for Logic Oriented Neural Networks",* in IEEE-INNS-ENNS International Joint Conference on Neural Networks, Vol. 2 (2000), pp. 123-128.
12. M. Ananda Rao and J. Srinivas, *Neural networks: algorithms and applications,* (Alpha Science International Ltd, 2003).
13. G. D. Magoulas, M. N. Vrahatis and G. S. Androulakis, *Improving the Convergence of the Back-propagation Algorithm Using Learning Rate Adaptation Methods,* Neural Computation, ISSN 0899-7667, Vol. 11, No. 7, (1999).
14. Shaobin Chai and Yong Zhou, *A Study on How to Help Back-propagation Escape Local Minimum",* in Third International Conference on Natural Computation, ICNC (2007), pp. 64-68.
15. Bing Gong, *A Novel Learning Algorithm of Back-Propagation Neural Network,* in IEEE International Conference on Control, Automation and Systems Engineering (July 2009), pp. 411-414.
16. S Forrest, *Genetic algorithms: principles of natural selection applied to computation, (*Science, New Series, Vol. 261, issue 5123, Aug 1993), pp. 872-878.
17. John Holland, *Adaptation in Natural and Artificial Systems*, (MIT Press, Apr 1992).
18. Zhi-Hua Cui and Jian-Chao Zeng, *Schema theorem of real-coded nonlinear genetic algorithm*, in IEEE International Conference on Machine Learning and Cybernetics, Vol. 3, (2002), pp. 142 – 143.