I J C T A, 9(37) 2016, pp. 393-401
© International Science Press

A Genetic Algorithm Based Learning In Constructive Neural Network for Pattern Classification

Mochitha Vijayan[*] and S.S. Sridhar[**]

Abstract: Constructive Neural Networks (CoNN) offer an interesting framework for pattern classifications problems. It provides an optimal way to determine the topology of a multilayer perceptron network trainable with supervised learning algorithms. This paper focuses on a simple CoNN with dynamic structure using Genetic Algorithm. Conventional neural network training involves topology to be defined before training, which results in an exhaustive search for solutions. CoNN learning algorithms eliminate the need for choosing the architecture before hand, and provide a mean for constructing networks whose size is dependent on the problem in hand. We apply Genetic Algorithm (GA) on the input data set subsequently until we get a best set of template strings. Data extracted from real world problems frequently contains rogue values. Rogue values decreases the quality of data and will affect the training process leading to inaccurate NN models. The template string helps to remove rogue values from the actual training set reducing the size of the input data set and thereby making learning easier. CoNN along with GA classifies the patterns in Iris dataset, Pima Indians Diabetes dataset and Wine dataset available in UCI machine learning repository with faster convergence and less space.

Keywords: Artificial Neural Network, Constructive Neural Network, Genetic Algorithm, Pattern classification.

1. INTRODUCTION

Artificial Neural Networks (ANNs) are models for computation inspired by biological neural networks. Artificial neural networks are generally presented as systems of processing elements called "neurons" massively interconnected by trainable connections called weights. These numeric weights that can be tuned based on experience, making neural nets adaptive to the given data and capable of learning. ANN algorithms involve training the numeral connection weights through a systematic procedure. Learning in ANN refers to the search for an optimal network topology and weights so as to accomplish a specific task. ANNs are capable of generalization, refitting and performing computation in parallel resembling the human brain. A "layer" in ANN has common functionality. The choice of number of neurons in input, hidden and output layers and their functionality depend on the task and the learning algorithm used. The input layer neuron count is usually same as the number of attributes of the training patterns, and, the output layer neuron count is same as the number of categories sought. Determination of the hidden layer neuron count is dependent on the inherent complexity of the task and number of training examples available.

A wide range of ANN architectures and algorithms have been put forward by researchers, among which Constructive Neural Networks (CoNN) offer a winning framework for pattern classifications problems. CoNN algorithms present an optimal way to determine the architectural structure of a Multi Layer Perceptron network trainable with learning algorithms to determine appropriate weights[1][2][3]. These algorithms initially start with a small network and allow the network to grow dynamically by adding and training neurons as needed until a satisfactory solution is found [4]. The architectural adaptation process is continued till the training algorithm finds a near optimal topology that gives suitable solution for the problem. Traditional neural network learning involves modification of interconnection weights on a network architecture

[*] Student, Department of Computer Science and Engineering, SRM University, Chennai, India. *Email: mochithavijayan_vijayakumar@srmuniv.edu.in*

[**] Professor, Department of Computer Science and Engineering, SRM University, Chennai, India. *Email: sridhar.s@ktr.srmuniv.ac.in*

defined prior to the training. Determining the network architecture in such a case is a challenge, which requires an expensive trial-and-error process. Instead of learning on a pre-specified network topology, a constructive algorithm learns the topology in a way specific to the problem. Constructive learning is capable of automatically fitting network size to the data without overspecializing, but yields better generalization for arbitrary pattern classification and thus has faster convergence in learning [5].

Constructive Neural Networks provide an optimal way to build minimal networks for the classification of patterns. They are based on elementary Threshold Logic Units (TLU), which tools a hard -limiting function. It starts with single TLU and add TLUs if necessary. CoNN became popular in two subtasks.

- When current topology fail to achieve the result, then it incrementally adds one or more threshold neurons to the network.
- Using different types of training algorithms to train the newly added threshold neuron(s).

The input data set selection method of NN is based on the template strings produced by Genetic Algorithm (GA). In the field of artificial intelligence, GA is a search heuristic based on the process of natural selection that mimics the biological evolution. This heuristic is routinely used to generate useful solutions to optimization and search problems. GA generates solutions to optimization problems using techniques inspired by natural evolution, such as inheritance, crossover, mutation and selection.

Template strings generated by GA is matched with the actual data set. This template string helps to remove rogue values from the actual training set reducing the size of the input data set and thereby making learning easier. CoNN along with GA classifies the patterns in Iris dataset, Pima Indians Diabetes dataset and Wine dataset available in UCI machine learning repository with faster convergence and with less spatial requirements.

The rest of the paper is divided into six sections. Section-II presents a literature review. Section-III discusses the proposed system. Section-IV describes the methodology being used. Section-V illustrates the practical applicability of the proposed system in comparison with existing architectures on various datasets and analyzes their performance and Section-VI is the conclusion. Section-VII is the future work.

2. LITERATURE REVIEW

Everett Fall, et al, has presented a new approach in designing neural networks [6]. The structure of the proposed NN is not stringently defined (each neuron may receive input from any other neuron). Instead, the initial network structure can be generated at random, and traditional methods of training, such as back propagation, are replaced or augmented by a genetic algorithm (GA). The weight of each input neuron is encoded to serve as the genes for the GA. By means of the training data provided to the supervised topology, the contribution of each neuron in creating a desired output serves as a selection strategy. Each of the neurons is then modified to store and recall past weightings for possible future use. The proposed NN can mimic the flexibility of biological networks in adapting to unexpected data. A simple network is trained to identify vertical and horizontal lines as a proof of concept.

Dr. S.S. Sridhar, et al, presents several CoNN learning algorithms that enable the NN architecture to be constructed along with the learning process [2]. This paper explores several algorithms like MTower, MPyramidal, MTiling, MSequential, MPerceptroncascade, MUpstart, Dist AI and algorithms such as Pocket algorithm along with ratchet modification and Barycentric correction procedure commonly used for training individual TLU's which have been proved to converge on Multi-categories. Constructive learning aims at automatically designing a network topology of appropriate size, given the full training data set. Constructive algorithms offer an attractive approach to automated design of neural networks for pattern classification. They eliminate the need for choosing the architecture before hand, and provide a mean for constructing networks whose size is dependent on the problem in hand. This paper has focused on a family of such algorithms that incrementally construct networks of threshold neurons that deal with multi categories.

Rajesh Parekh, et al, provides convergence proofs for several constructive learning algorithms [4]. These algorithms offer a winning approach for incremental construction of potentially near-minimal neural network architecture for pattern classification tasks. These algorithms help to overcome the need for ad-hoc and often inappropriate choice of network topology for the algorithms that search for a suitable weight setting in a-priori fixed network architecture. The convergence proof for the Tower, Pyramidal, Tiling, and Sequential Learning (except Upstart, Perceptron cascade) relies on the assumption that the pattern attributes values are either binary or bipolar. This paper explores multi-category extensions of several CoNN learning algorithms for classification tasks where the input patterns may take on real valued attributes.

Dr. S.S. Sridhar, et al, proposes a new Tiling architecture for Constructive Neural Networks [1]. In the study they proposed a new MTiling architecture with unsupervised learning strategy on binary pattern datasets for achieving better performance in terms of generalization capability, faster convergence and fewer connections thereby less storage requirement.

Philipp Koehn introduces the idea of the genetic algorithm paradigm applied to neural networks in the task of weight as well as architecture optimization [7]. Since genetic algorithms operate on simple fitness, i.e. quality evaluations, they can be applied to a larger class of problems. The application of GANN architecture optimization is feasible, if the goal is a good topology for a large class of similar problems. In these cases, the high computation time is justified, since it has to be performed only once. Then, for a particular instance of the problem the optimized architecture can be used.

Joao R. Bertini, et al, makes a comparative study of two TLU training algorithms [8]. CoNN algorithms enable the architecture of a NN to be constructed as an intrinsic part of the learning process. These algorithms are dependent on the TLU training algorithm employed. Generally they use a perceptron based algorithm such a Pocket or Pocket with Ratchet Modification (PRM) for training individual node added to the network, during the learning process. PRM is a practical choice for training a TLU but its cost of computation can be reasonably high when there are many training instances; it takes a long time to find an optimal solution. Inference based on results of the experiments shows that there are domains whose characteristics favour the geometric approach employed by BCP and others does not. Depending on the domain, BCP makes a better option than PRM to be used as a part of constructive neural methods.

3. SYSTEM STUDY

Constructive neural network architectures use the multi-category extensions of CoNN learning algorithms for classification tasks. Data extracted from real world problems frequently contains rogue values. Rogue values decreases the quality of data and will affect the training process leading to inaccurate NN models. That is, the quality of the data may be reduced by errors or deviations produced in the data collection phase, as a consequence of human error in translating information or due to limitations in the tolerance of the measurement equipment. This may result in errors in the values of attributes or in the instances.

In general, rogue values may bias the learning process, making it more difficult for learning algorithms to form accurate NN models for the data. This is the issue addressed in this work.

Proposed work presents a CoNN which provides an optimal way to determine the architecture of a multilayer neural network trainable with supervised learning algorithms for pattern classification tasks. Instead of feeding the network with input dataset that contains rogue values, we remove the rogue values prior to the training process. This can lead training to the right direction resulting an accurate CoNN model which have faster convergence to the solution and requires less space. Figure 1 is the block diagram of the proposed system.

4. METHODOLOGY

A. *Template String Generation Using GA:* A genetic algorithm (or GA) is a search technique used in computing to find true or approximate solutions to optimization and search problems. GAs are a particular class of evolutionary algorithms that use techniques inspired by evolutionary biology such as inheritance, mutation, selection, and crossover (also called recombination).

The evolution usually starts from a population of randomly generated individuals and happens in generations. Here the inputs from the dataset are considered as initial population.

In each generation, the fitness of every individual in the population is evaluated, multiple individuals are selected from the current population (based on their fitness), and modified to form a new population. The contribution of each attribute of a data (individual) in making the data lie in an acceptable range act as a fitness strategy. The new population is used in the next iteration of the algorithm. The algorithm terminates when the required number of template strings are achieved.

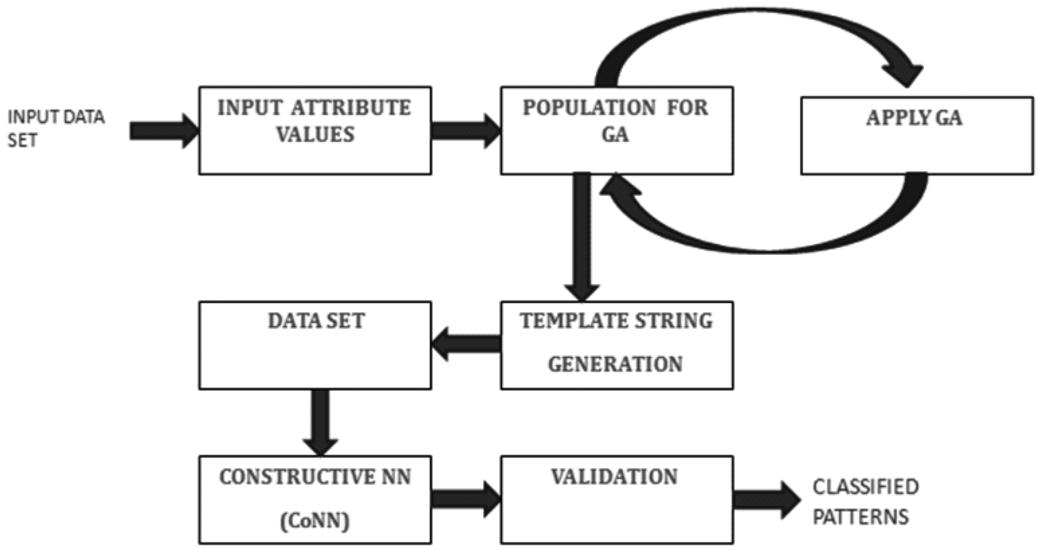

Figure 1: Block diagram of proposed system

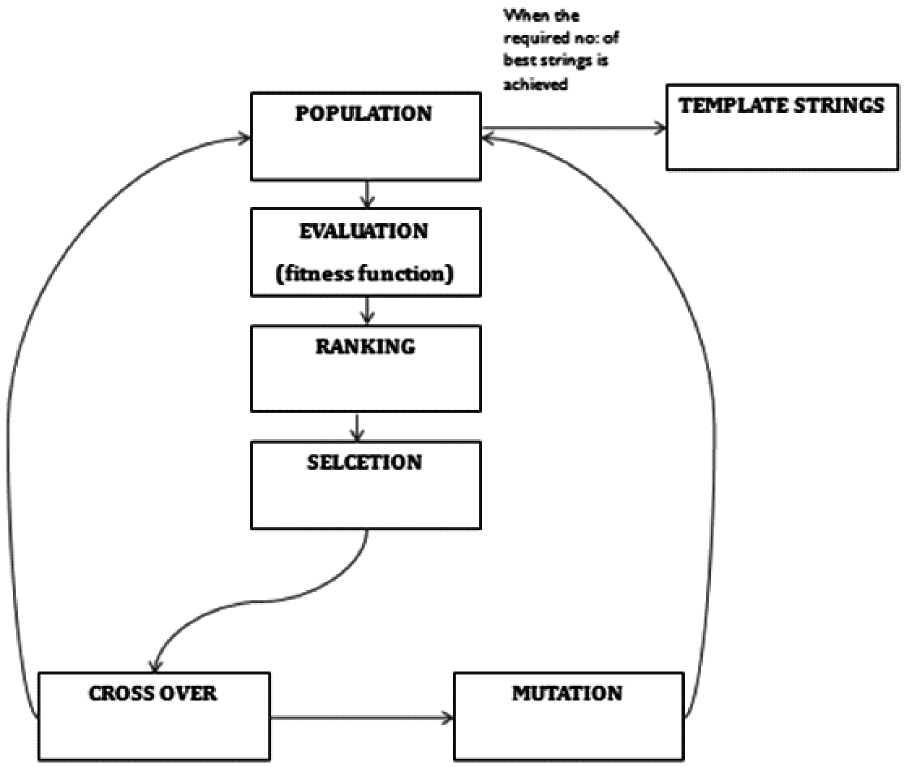
When the required no: of best strings is achieved
POPULATION
TEMPLATE STRINGS
EVALUATION (fitness function)
RANKING
SELCETION
CROSS OVER
MUTATION

Figure 2: Demonstrates the GA process

B. *Refining The Actual Dataset Using Template String:* Template strings are considered as the bench mark datasets. The actual dataset is refined using the template string generated using GA. Only those datasets that matches with the template strings is being refined. This process of refinement will remove the rogue values thereby reducing the number of datasets for training the network.

C. *Constructive Neural Network Learning Algorithm:* In order to classify a set of input patterns into one of two classes, a single perceptron or TLU shall be trained. This perceptron computes a hard-limiting function of weighted sum of inputs. The Multi-category Tiling algorithm incrementally constructs networks of threshold neurons to handle multi categories as discussed here.

This algorithm constructs a layered network of TLUs (Threshold Logic Units). The bottom-most layer receives inputs from each of the X input neurons. The neurons in each subsequent layer receive inputs from the neurons in the layer immediately below itself; in this process each layer maintains a master neuron (output of that layer). In addition to master neurons, ancillary neurons are added to the layers for faithful representation, if classification is not done properly. Figure 3 is a CoNN constructed using MTiling algorithm [2].

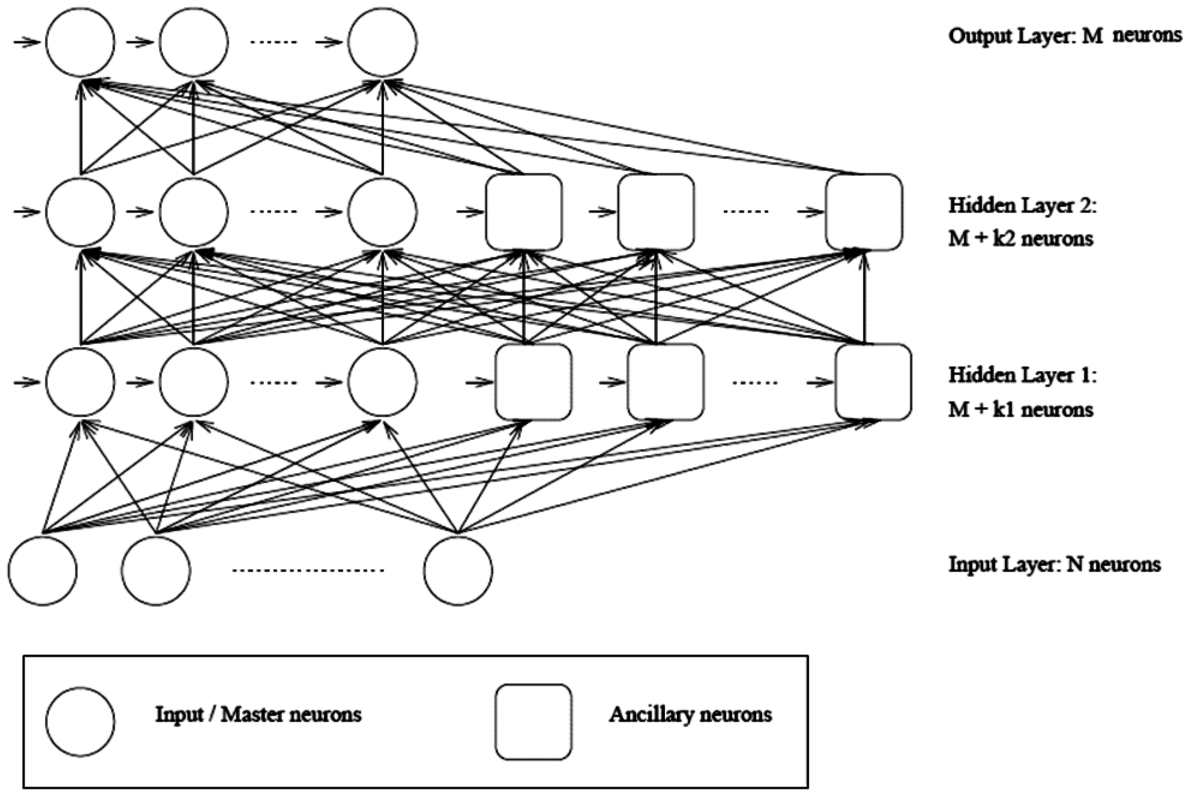

Figure 3: A CoNN using MTiling algorithm

Algorithm

(a) Initialize a layer of O master neurons each connected to X input neurons and train them.

(b) If the desired classification is achieved by master neurons, then stop.

(c) else, if the current layer is not faithful,

Then, add ancillary neurons to the current layer to make it faithful as follows, Else go to step d.

1. Identify the output vectors for which maximum number input patterns map to (an output vector is said to be unfaithful if it is generated by input patterns belonging to different classes)

2. Identify the set of patterns that generate the output vector identified in step c (1) above. This set of patterns will form the training set for ancillary neurons.
3. Add and train a set of n ($1 <= n <= O$) ancillary neurons where n is the number of target classes represented in the set of patterns identified above.
4. Repeat the last three steps till the output layer representation of the patterns is faithful.

(d) The new layer of master neurons are trained, which are connected to each neuron in the previous layer, then go to step b.

A simple perceptron learning algorithm is used for training individual TLU's.

5. EXPERIMENTAL RESULTS

The performance of the existing CoNN and the new CoNN with GA is analyzed in this section. The real-world datasets **Iris**, **Pima Indian diabetic** and **Wine** are available at the UCI Machine Learning Repository [9].

All the datasets are initially normalized to distribute the data evenly and scale it into an acceptable range for the network [10]. Patterns in each normalized dataset were divided as training and evaluation set for both the cases. The normalized attribute values of each training data set are fed to the CoNN to make it learn. Equal number of datasets is used for training the existing CoNN and the new CoNN with GA, and equal number of datasets is used for evaluation in both the cases. The parameters like the number of hidden layers, number of neurons in each hidden layer, number of weight connections, generalization capability, number of ancillary neurons , total number of neurons and time taken for training the CoNN are analyzed based on the results obtained during the evaluation.

Table 1, 2 and 3 summarizes the results of experiments done to test the performance of CoNN and CoNN with GA.

On applying Iris dataset as in Table-I, it was found that in CoNN with GA has fewer weight connections thereby less storage requirement, time taken is also reduced when compared with the other. The performance graph is shown in Figure 4.

On applying Pima Indian diabetic dataset as given in Table 2, it was found that CoNN and CoNN with GA behaved in a similar manner for all the parameters except for the number of misclassifications and time taken. CoNN with GA has reduced the number of misclassifications and the time taken. The performance graph is shown in Figure 5.

On applying Wine dataset as in Table 3, it was found that the total weight connections are considerably reduced in CoNN with GA when compared with other thereby minimizing its space requirements; time taken is also being reduced. The performance graph is shown in Figure 6.

Table 1
Performance on iris dataset

	CoNN	*CoNN (GA)*
Misclassifications	3	1
Hidden Layers	2	1
Hidden Layer Neurons	8	4
Ancillary Neurons	2	1
Total Neurons	15	11
Total Weight Connections	44	28
Time In ms	1705	1674

Table 2
Performance on pima indian diabetic dataset

	CoNN	*CoNN (GA)*
Misclassifications	5	4
Hidden Layers	2	2
Hidden Layer Neurons	6	6
Ancillary Neurons	2	2
Total Neurons	16	16
Total Weight Connections	39	39
Time In ms	1325	1245

Table 3
Performance on wine dataset

	CoNN	*CoNN (GA)*
Misclassifications	4	2
Hidden Layers	2	2
Hidden Layer Neurons	10	8
Ancillary Neurons	4	2
Total Neurons	26	24
Total Weight Connections	105	80
Time In ms	1625	1368

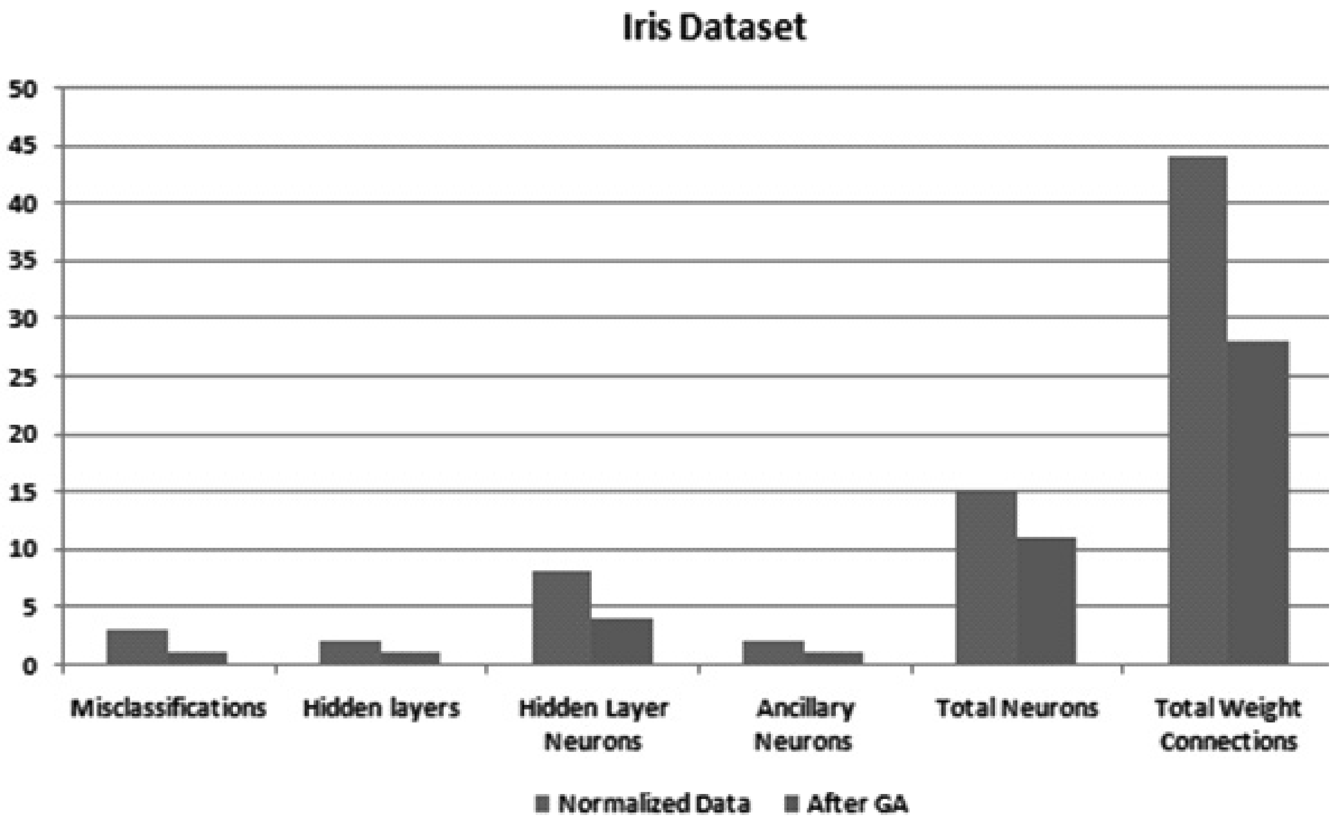

Figure 4: Performance on Iris dataset

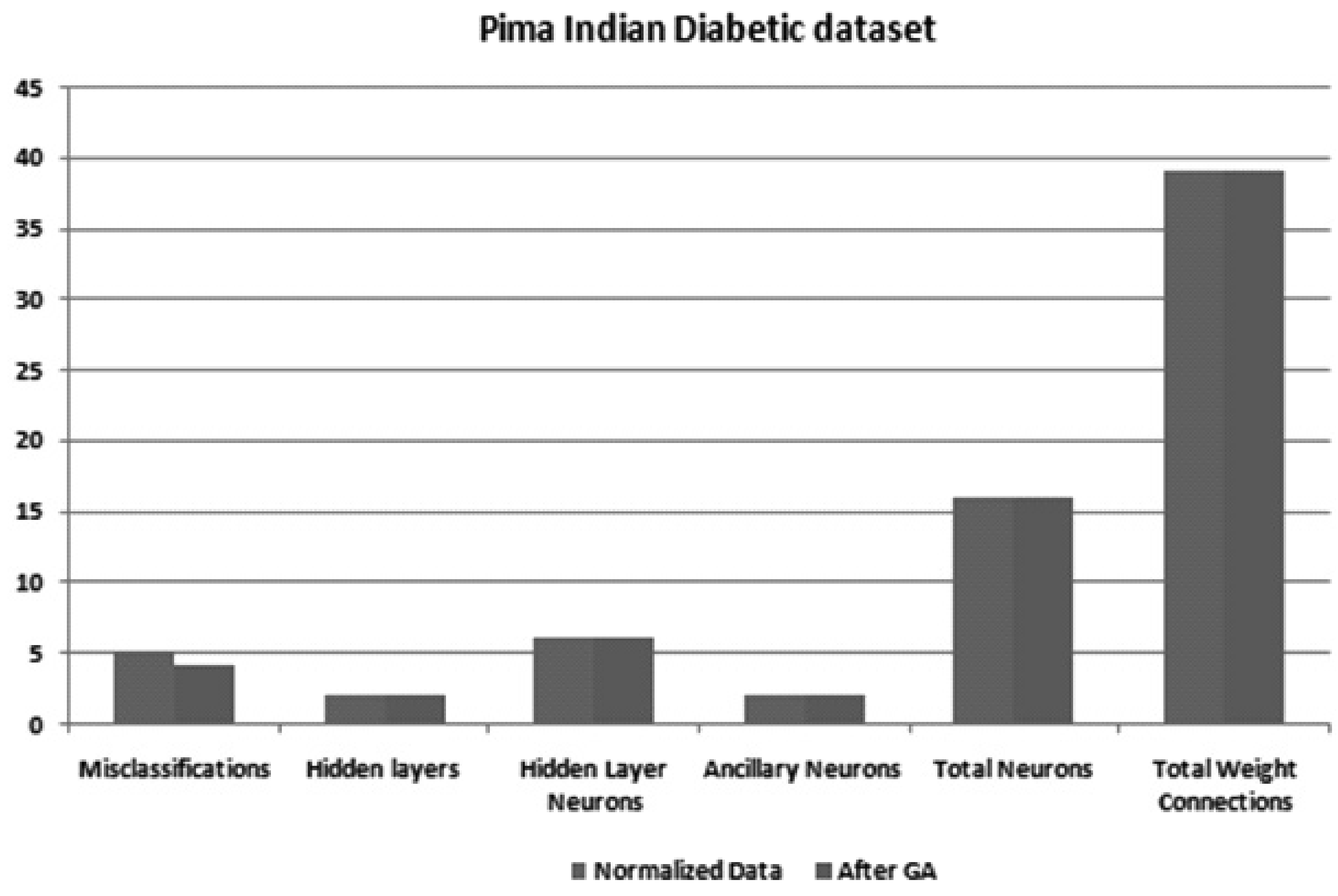

Figure 5: Performance on Pima Indian Diabetic dataset

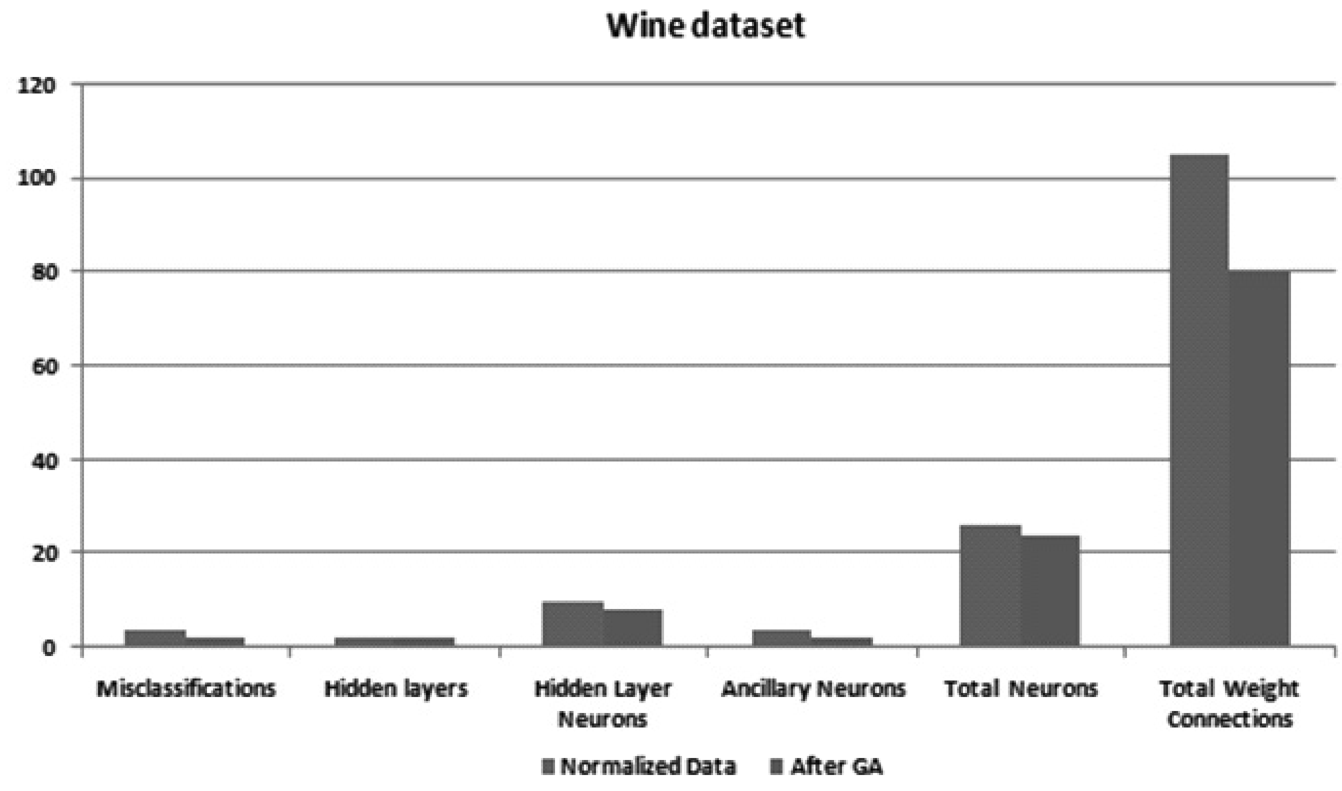

Figure 6: Performance on Wine dataset

6. CONCLUSION

In this work, we reduced the time and spatial requirements of a CoNN by applying GA to the input data set prior to the training process. The template string generated by GA removed rogue values from the actual training set reducing the size of the input data set and thereby making learning easier. This lead the constructive training algorithms to generate an accurate CoNN model which has faster convergence to the solution with less space.

7. FUTUTE WORK

Following are some of the future research directions

1. The performance of CoNN with GA in terms of parameters like the network size, training time, generalization capability and convergence properties on various other datasets of UCI repository shall be done.
2. Other learning algorithms shall be experimented for improving the performance of CoNN with GA.
3. Various pre-processing techniques shall be applied to input dataset for the performance improvement.

References

1. Dr. S.S. Sridhar and Dr. M. Ponnavaikko, "*New Constructive Neural Network Architecture for Pattern Classification"*, Journal of Computer Science 5 (11): 843-848, 2009 ISSN 1549-3636.
2. Dr S.S. Sridhar and Dr M. Ponnavaikko, "*Analysis of Constructive Neural Network Learning Algorithms for Pattern Classification*", 2009 IEEE International Advance Computing Conference (IACC 2009) Patiala, India, 6–7 March 2009.
3. Maria do Carmo Nicoletti, João R. Bertini Jr., David Elizondo, Leonardo Franco, and José M. Jerez*,"*Constructive Neural Network Algorithms for Feed forward Architectures Suitable for Classification Tasks*"
4. Rajesh Parekh, Jihoon Yang and Vasanth Honavar, "*Constructive Neural Network Learning Algorithm For Multi-Category Real valued Pattern Classification*", Springer-Verlag Berlin Heidelberg 2009
5. Justin Fletcher and Zoran ObradoviC, "*Constructively Learning a Near-Minimal Neural Network Architecture*", IEEE, 1994
6. Everett Fall and Hsin-Han Chiang, "*Neural Networks with Dynamic Structure Using a GA-based Learning Method"*, 2015 IEEE 12th International Conference on Networking, Sensing and Control Howard Civil Service International House, Taipei, Taiwan, April 9-11, 2015
7. Philipp Koehn *, "Combining Genetic Algorithms and Neural Networks: The Encoding Problem*", A Thesis Presented for the Master of Science, Degree,The University of Tennessee, Knoxvill, December 1994
8. Joao R. Bertini, Maria do Carmo Nicoletti and Estevam R. Hruschka Jr, "*A Comparative Evaluation of Constructive Neural Networks Methods using PRM and BCP as TLU Training Algorithms*", 2006 IEEE International Conference on Systems, Man, and Cybernetics,October 8-11, 2006, Taipei, Taiwan
9. P. Murphy and D. Aha, "*Repository of machine learning databases*", Dept. Inform. Comput. Sci., Univ. California,Irvine,CA, http://www.ics.uci.edu/AI/ML/MLDBRepository.html,1994
10. Lou Mendelsohn, "*Preprocessing Data for Neural Networks*"