 ISSN : 0976-8491 (Online) | ISSN : 2229-4333 (Print)

An Implementation for Combining Neural Networks and Genetic Algorithms

[1]Md. Mijanur Rahman, [2]Tania Akter Setu
[1,2]Dept. of Computer Science and Engineering, JatiyaKabiKaziNazrulIslam University, Bangladesh

Abstract
Neural Network (NN) and Genetic Algorithm (GA) are two very known methodology for optimizing and learning. Each having its own strengths and weakness. These two have generally evolved along separate paths. Recently there have been attempts to combine the two technologies. This research is devoted to implement a method for combining genetic algorithm with neural network (GANN). A collaborative approach has been used in this research. To integrated GA and NN into a single system, a population of neural networks is evolved, i.e., the goal of the proposed system is to find the optimal neural network solution. In collaborative system GA and NN work parallel; for optimization it is very necessary to work with parallel.Several MATLAB functions and tools have been used to implement the proposed GANN method. The development and experiments demonstration of GANN is done on MATLAB 7.0.12. The proposed method consists of neural learning by the backpropagation algorithm and applies evolutionary, genetic operators. The learning behavior of the algorithm was tested on a simple pattern recognition example and it was able to prove its performance. The new ideas, concepts and processes of GANN bring new life in the field of Artificial Intelligenceresearch.

Keyword
Backpropagation, Collaborative Approach, Genetic Algorithm, Neural Network, Optimization

I. Introduction

Neural Networks (NN) and Genetic Algorithms (GA) have powerful problem solving ability. They are based on simple technical principles, but take advantages of their mathematical characteristics. Neural networks used as a learning system and genetic algorithms as an optimization system. The choice of the basic parameter (network topology, learning rate, initial weights) is the requirement of the success of the training process. Genetic algorithms are global search methods that are based on principles like selection, crossover and mutation. The Neural network is use for the training the system and genetic algorithm optimized the training parameter of the network system.

The idea of combining GA and NN came up first in the late 80s, and it has generated anintense field of research in the 1980s [1]. Both are autonomous computing methods, but we combine them to set the different parameters of the methods. The problem with neural networks is that a number of parameterhas to be set before any training can begin. However, there are no clear rules how to setthese parameters. Yet these parameters determine the success of the training.

By combining genetic algorithms with neural networks (GANN), the genetic algorithm isused to find these parameters. The inspiration for this idea comes from nature. In real life, thesuccess of an individual is not only determined by his knowledge and skills which he/she gained through experience (the neural network training), it also depends on his/her genetic heritage (setby the genetic algorithm). GANN applies a natural algorithm that proved tobe very successful on this planet. It created human intelligence from scratch.The topic of this research is the question of how exactly GA and NN can be combined, i.e.especially how the neural network should be represented to get good results from the geneticalgorithm.

A brief introduction to the foundations of neural networks and genetic algorithms have discussed is this paper. The different combining methods and techniques for combining Genetic Algorithm and Neural Network (GANN) have been implemented. The research defines how to optimize neural network by using genetic algorithm. The combining process is applied on a function, where genetic algorithm is used to set parameters for training the neural network.

II. Genetic Algorithm

The biological metaphor for genetic algorithms is the evolution of the species by survival of the fittest, as described by Charles Darwin. In a population of animals or plants, a new individual is generated by the crossover of the genetic information of two parents. A genetic algorithm (GA) tries to simulate the natural evolution process. Its purpose is to optimize a set of parameters. In the original idea, proposed by John Holland [2], the genetic information is encoded in a bit string of fixed length, called the parameter string or individual. A possible value of a bit is called an allele [3]. A variety of computational models based on evolutionary processes have been proposed, and the most popular models are those known as genetic algorithms [4].

A genetic algorithm (GA) is great for finding solutions to complex search problems and it is efficient in reducing computation time for a huge heap-space [5]. In 1989, Dr. David Goldberg offered the definition of GA as "Genetic algorithms are search algorithms based on the mechanics of natural selection and natural genetics". This method combines Darwinian style survival of the fittest among binary string "artificial creatures" with a structured, yet randomized information exchange [6]. A genetic algorithm requires a few base components:

- A representation of a candidate solution.
- A way of calculating the correctness of a candidate solution, commonly referred to as the fitness function.
- A mutation function that changes a candidate solution slightly, in an attempt to obtain a better candidate.

The basic process for a genetic algorithm is given below and summarized in Fig. 1:

A. Evaluation

Each member of the population is then evaluated; calculate a 'fitness' for that individual. The fitness value is calculated by how well it fits with the desired requirements. These requirements could be simple, 'faster algorithms are better', or more complex, 'stronger materials are better but they shouldn't be too heavy'.

B. Selection
Constantly improve the populations overall fitness. Selection helps to do this by discarding the bad designs and only keeping the best individuals in the population. There are a few different selection methods but the basic idea is the same, make it more likely that fitter individuals will be selected for the next generation.

C. Crossover
Create new individuals by combining aspects of the selected individuals during crossover. The hope is that by combining certain traits from two or more individuals, an even 'fitter' offspring will be created which will inherit the best traits from each of it's parents.

D. Mutation
Add a little bit randomness into the populations' genetics otherwise every combination of solutions that can create would be in the initial population. Mutation typically works by making very small changes at random to an individual's genome.

E. And Repeat!
Now for the next generation we can start again from step two until we reach a termination condition.

There are a few reasons to terminate a genetic algorithm from continuing it's search for a solution. The most likely reason is that the algorithm has found a solution which is good enough and meets a predefined minimum criteria. Other reasons for terminating could be constraints such as time or money.

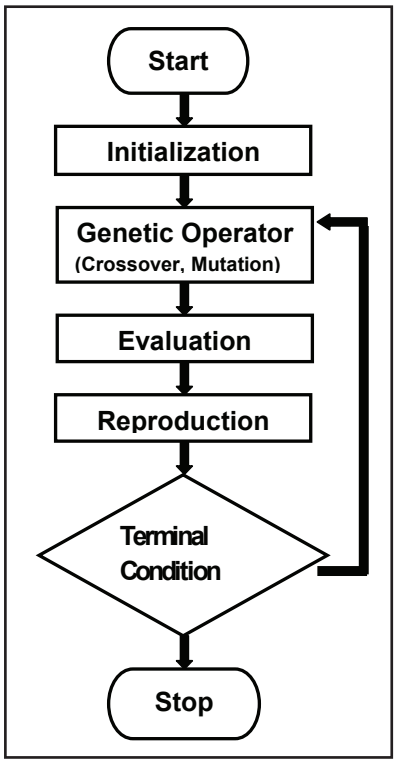

Fig. 1: Flowchart of Genetic Algorithm.

III. Neural Network
An Artificial Neural Network (ANN) is information processing system that is inspired by the way biological nervous systems, such as how does the brain process information. It is composed of a large number of highly interconnected processing elements (also called neurons) working with each other to solve specific problems.

The learning process is essentially an optimization process in which the parameters of the best set of connection coefficients (weighs) for solving a problem are found and includes the following basic steps [7]:

- Present the neural network with a number of inputs (Vectors each representing a pattern)
- Check how closely the actual output generated for a specific input matches the desired output.
- Change the neural network parameters (weights) to better approximate the outputs.

Multilayer feedforward neural networks have been the preferred neural network architectures for the solution of pattern classification and recognition problems, through a learning process [8]. A multilayer neural network, consists of three layers, input, hidden and output layers (see Fig. 2), can approximate any smooth, measurable function between input and output vectors by selecting a suitable set of connecting weights and transfer functions.

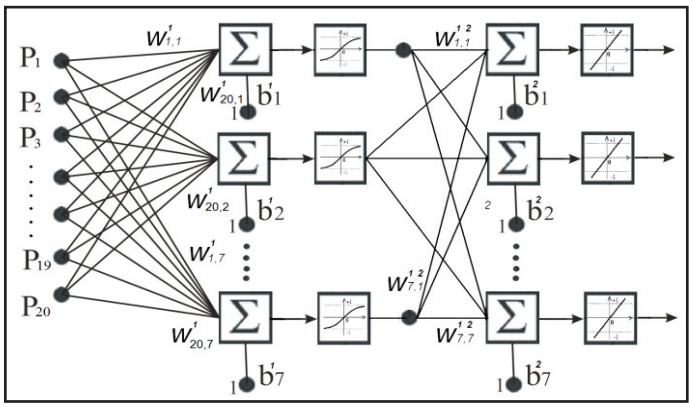

Fig. 2: The architecture of a feed forward neural network with 20 neurons in input, 7 in hidden layer and 7 in output layers.

Backpropagation training algorithm is a supervised learning algorithm for multilayer feed forward neural network, given by the flowchart shown in Fig. 3. The training of a neural network by back-propagation algorithm involves three stages [9]:

1. The feed forward of the input training pattern (feed forward pass)
2. The calculation and back propagation of associated error (backward pass) and
3. The adjustment of the weights

IV. Encoding The Network
The general idea of combining GA and NN is illustrated in Fig. 4 [1]. Information about the neural network is encoded in the genome of the genetic algorithm. At the beginning, a number of random individuals are generated. The parameter strings have to be evaluated, which means a neural network has to be designed according to the genome information. Its performance can be determined after training with back-propagation. Some GANN strategies rely only on the GA to find an optimal network; in these, no training takes place. Then, they are evaluated and ranked. The fitness evaluation may take more into consideration than only the performance of the individual. Some approaches take the network size into account in order to generate small networks. Finally, crossover and mutation create new individuals that replace the worst - or all - members of the population. This general procedure is quite straight-forward. The problem of combining GA and NN lies in the encoding of the network.

ISSN : 0976-8491 (Online) | ISSN : 2229-4333 (Print)

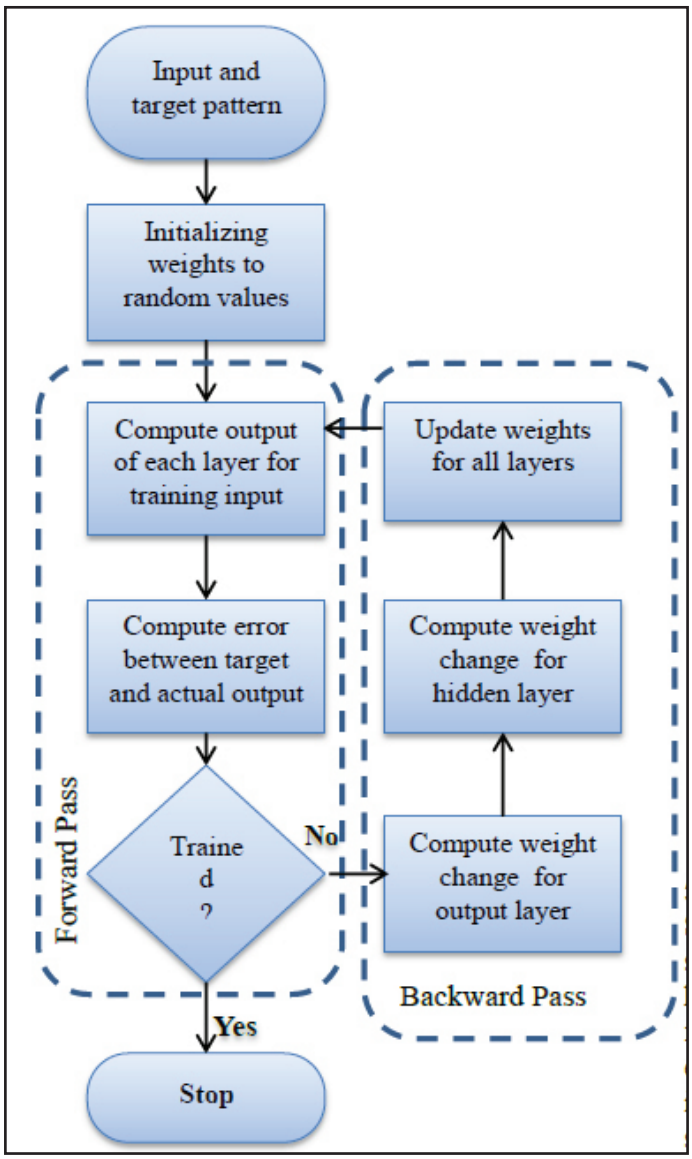

Fig. 3: Flowchart of Backpropagation Algorithm for Training Neural Network

The new ideas and concepts of GA and NN bring new life into Artificial Intelligence research.

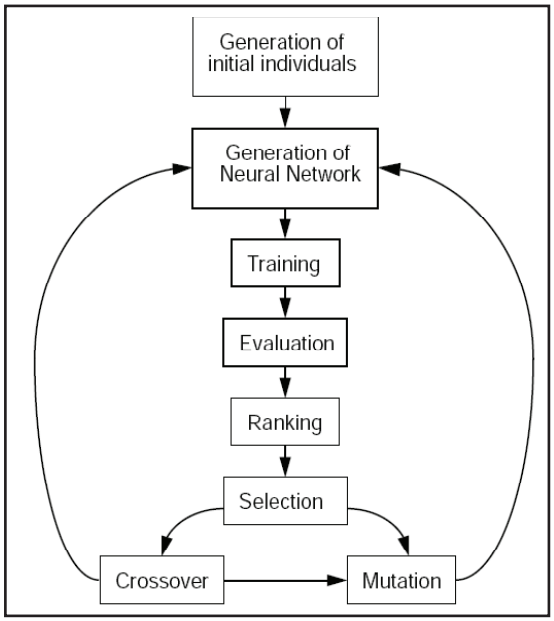

Fig. 4: The Principle Structure of a GANN System

V. GANN Approaches

Since first attempts to combine GA and NN started in the late 1980s, other researchers have joined the movement and created a flood of journal articles, technical reports etc. A broad variety of problems have been investigated by different GANN approaches, such as face recognition [10], animate [11], classification of the normality of the thyroid gland [12], color recipe prediction [13] and many more. Also, a variety of different encoding strategies have been implemented. This chapter tries to structure this information with focus on feed-forward networks with back-propagation. It does not emphasize on the complete review of single approaches, it rather attempted to gather more general information and find general patterns.

Researchers have combined NNs and GAs in a number of different ways. Schaffer et al. have noted that these combinations can be classified into one of two general types - supportive combinations in which the NN and GA are applied sequentially, and collaborative combinations in which they are applied simultaneously [4].

A. Supportive and Collaborative Approach

In a supportive approach, the GA and the NN are applied to two different stages of the problem. The most common combination is to use a GA to preprocess the data set that is used to train a NN. For instance, the GA may be used to reduce of the data space by eliminating redundant or unnecessary features. In the supportive approach, neural networks and evolutionary algorithms are used independently at different stages of the problem.

Some common ways in which these sequential collaborations are implemented include input feature selection of neural network training data using evolutionary optimization, pre-processing of training data using evolutionary algorithm, and analysis of neural network representations using genetic algorithms.

In the collaborative approach, evolutionary algorithms and neural networks are used in a single system to evolve a population of neural networks best suited for a given task. One of the popular methods is to use it as a means to learn artificial neural network connection weights that are coded, as binary or real numbers, in a genetic algorithm string [14-15]. The second one is to use the genetic algorithm to evolve and select the artificial neural network architecture, together or independently from the evolution of weight [14].

A collaborative approach has been used in this research. To integrated GA and NN into a single system, a population of neural networks is evolved. In other words, the goal of the proposed system is to find the optimal neural network solution. In collaborative system GA and NN work parallel, for optimization it is very necessary to work with parallel.

B. Combining Process

In the combining process, the genetic algorithm is used to train the neural network. To integrate the NN with GA, choose their structure or design related aspects like the function of their neurons, to select the ANN topology and we can use genetic programming to define network. The combining process is below [16].

- Using GA to train NN: In this case it is no matter how the network is connected. The weight of the network use as a population of GA by making the string and GA evaluate the fitness.
- Using GA to select NN topologies: By using genetic algorithm one can evaluate the how neurons are connected with one

another in a network.
- Using GP to define networks: Some researchers have used genetic programming (GP) techniques to represent networks. We can also use GAs to find the optimum configuration of the network and its associated parameters.

GAs areDirected Random Searches. They search, starting from random points, and slowly converge to a solution. There are several reasons, but the most important is simply that the evaluation algorithm will train the network, no matter how it is connected. The following steps are worked to train the neural network with genetic algorithm:
- Step-1: Set weights in the network, are joined to make one string
- Step-2: Define the network
- Step-3: Use this string in the GA as a member of the population
- Step-4: Train the weight until obtain the target output.

C. BP Neural Network Optimized by Genetic Algorithm

A genetic algorithm can be applied to optimizing a neural network in a variety of ways. Yao has indicated three main approaches of optimization; such as, the evolution of weights, the evolution of topology and the evolution of learning rules [15].

1. Evolution of connection weights

In the first, the GA is used as the learning rule of the NN. The genetic code is a direct encoding of the neural network, with each weight being represented explicitly. The populations of the GA are all NNs with the same basic topology, but with different weight values. Mutation and crossover thus affect only the weights of the individuals.

2. Evolution of Architectures

In the second approach, the GA is used to select general structural parameters and the neural learning is used separately to train the network and determine its fitness. This includes evolution of both the topology and activation functions of each node.

3. Evolution of Learning Rules

Evolving learning rules does not refer simply to adapting learning algorithm parameters, e.g., learning rate, momentum, etc.; but to adapting the learning functions themselves.

The overall optimization process of BP neural network with genetic algorithm is shown in Fig. 5 and summarized below.

(i). Define Neural Network

The structure of a BP neural network is determined by the input and output parameters. A feed forward neural network proposed to combine with Genetic algorithm has 10 input and 10 output and 2 hidden neurons.

(ii). Weight Initialization

Weightsand bias valueshave been initialized randomly. Then the Mean Square Error function has been used to calculate Error (Error =target output-actual output).

(iii). Genetic Algorithm Optimization

The genetic algorithm is exploited to optimize the weights and thresholds of the BP neural network. Each individual of the population consists of all weights and thresholds of a network and the individual fitness value is calculated by the individual fitness function. Subsequently, the genetic algorithm finds the best fitness value corresponding to the individual by means of the selection, crossover, and mutation.

Whenthe error is greater than minimum error then GA use MSE function as a handle function until the minimum error will be acquire, the weight will update by GA and generation continuously. Weight value will be optimized by genetic algorithm until minimum error is found.

(iv). Network Prediction

Next, the BP neural network obtains the optimal initial weights and threshold provided by the genetic algorithm and predicts the function output after the network has been trained.

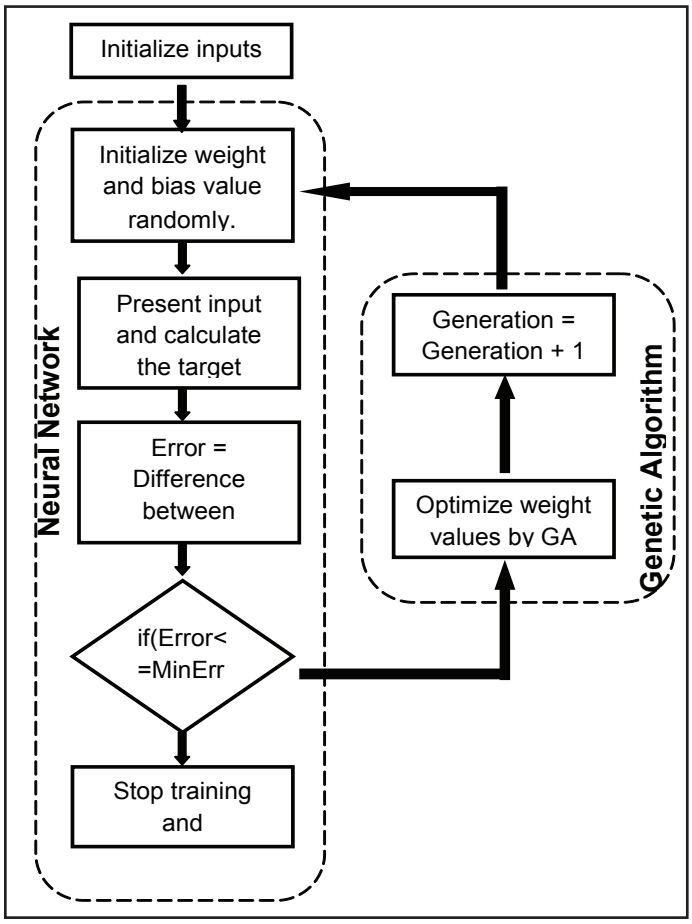

Fig. 5: Flowchart of BP Neural Network Optimization by Genetic Algorithm

VI. Experiments and Conclusions

In this research work, the genetic algorithm trains the neural network by using a function. The proposed approach used several steps to optimize the neural network weight selection with genetic algorithm. The proposed network has two neurons in the hidden layer and takes the inputs, from 1 to 10 and fixed the target cos(inputs2). The GA function requires a function handle as an input argument to which it passes a 1×N vector, where N is the number of variables in the system to be optimized. For the neural network, the weights and biases are M×1 vector. These are optimized by using genetic algorithm. Weight and biases are selected by the network randomly. MSE function calculate the error correspond to the output and target output. Until the minimum error is found, the GA function continuously optimized the weights and bias values of the network. The proposed method can make 100 numbers of generations to optimize the network. Several MATLAB functions

and tools have been used to implement the proposed method. The development and experiments demonstration of GANN is done on MATLAB 7.0.12.

A hybrid learning method for neural network structure is described in this research. The proposed method consists of neural learning by the backpropagation algorithm and applies evolutionary, genetic operators. The learning behavior of the algorithm was tested on a simple pattern recognition example. It was able to prove its potential and it may exhibit good performance. In future, this research can be extended in image and speech recognition applications.

References

[1] Philipp Koehn,"Combining Genetic Algorithms and Neural Networks: The Encoding Problem", MS Thesis, The University of Tennessee, Knoxville, December 1994.

[2] J.H. Holland,"Adaption in natural and artificial systems", University ofMichigan Press, Ann Harbor, 1975.

[3] David W. White,"GANNet: A genetic Algorithm for Searching Topology and Weight Spaces in Neural Network Design", Dissertation at the University of Maryland, 1993.

[4] Talib S Hussain,"Methods of Combining Neural Networks and Genetic Algorithms", Queen's University, hussain@qucis.queensu.ca, 1997.

[5] Davis L,"Handbook of Genetic Algorithms", Van Nostrand Reinhold, 1991.

[6] RichaMahajan, GaganpreetKaur,"Neural Networks using Genetic Algorithms", 2013.

[7] SergiosTheodorios, KonstantinosKoutroumbas,"Pattern Recognition", Cambridge: Academic Press, 1999.

[8] Christos, S., Dimitrios, S.,"Neural Networks", http://www.emsl.pnl.gov:2080/docs/cie/neural/neural.homepage.html.

[9] Md. MijanurRahman,"Development of Speech Recognition System for Continuous Bangla Speech", PhD Thesis, Jahangirnagar University, Bangladesh, 2014.

[10] P. J. B. Hancock,"Gannet: Genetic design of a neural network for face recognition", In: Parallel Problem Solving from Nature 2, pp. 292-296, H.P. Schwefel and R.Maenner, Springer Verlag, 1991.

[11] VittorioManiezzo,"Genetic Evolution of the Topology and Weight Distribution of Neural Networks", In: IEEE Transactions of Neural Networks, Vol. 5, No. 1, pp. 39-53, 1994.

[12] W. Schiffmann, M. Joost, R. Werner,"Application of Genetic Algorithms to the Construction of Topologies for Multilayer Perceptrons", In: Proceedings of the International Joint Conference on Neural Networks and Genetic Algorithms, pp. 675-682, Innsbruck, 1993.

[13] J.M. Bishop, M.J. Bushnell,"Genetic Optimisation of Neural Network Architectures for Colour Recipe Prediction", in: Proceedings of the International Joint Conference on Neural Networks and Genetic Algorithms, pp. 719-725, Innsbruck, 1993.

[14] Z. Michalewicz, Genetic Algorithms + Data Structures =Evolutionary Programs, Springer-Verlag, New York, 1994.

[15] D. Fogel, L. Fogel, V. Porto,"Evolving neural networks", Biol. Cybernet, Vol. 63, pp. 487–493, 1990.

[16] "Applying Genetic Algorithms to Neural networks", [Online] Available: http://www.rgu.ac.uk/files/chapter15%20-%20eanns.pdf.

Md. Mijanur Rahman Received his B.Sc and M.Sc Degree in Computer Science and Engineering from Islamic University, Bangladesh. He also Received his PhD. Degree from Jahangir Nagor University, Bangladesh in 2015. He worked as Associate Professor in Dept. of Computer Science and Engineering, Jatiya Kabi Kazi Nazrul Islam University, Bangladesh. His Research interest in Pattern Recognization, Bangla Speech Processing, Digital image Processing.

Tania Akter Setu Received her B.Sc. Engg degree in Computer Science & Engineering from Jatiya kabi Kazi Nazrul Islam University Trishal, Mymenshingh, Bangladesh in 2011. Now she is a MS researcher in Jatiya kabi Kazi Nazrul Islam University.Her interest in Digital Image processing, Neural network, Genetic Algorithm.