Optimizing Weights of Artificial Neural Networks using Genetic Algorithms

Kinjal Jadav, Mahesh Panchal

Abstract- Artificial Neural Networks have a number of properties which make them suitable to solve complex pattern classification problems. Their applications to some real world problems has been adopted by the lack of a training algorithm. This algorithms finds a nearly globally optimal set of weights in a relatively short time. Back propagation is one of the training algorithm of the Artificial neural network. However, training the neural networks using backpropagation algorithm may cause two main drawbacks: trapping into local minima and converging slowly. In view of these limitations of back-propagation neural networks, global search technique such as Genetic algorithm have been presented to overcome these shortcomings. Genetic algorithms are a class of optimization procedures which are good at exploring a large and complex space in an intelligent way. It finds values close to the global optimum. Hence, they are well suited to the problem of training and optimize weights of Artificial Neural Networks. In this paper the use of Genetic algorithms to optimize weights of Artificial Neural Networks is shown.

Index terms - Artificial Neural Networks, Back propagation algorithm, Genetic algorithms.

I. INTRODUCTION

An Artificial Neural Network (ANN) is an information processing paradigm that is based on the function on human brain. The power and usefulness of artificial neural networks have been demonstrated in several applications including speech synthesis, diagnostic problems, medicine, business and finance, robotic control, signal processing, computer vision and many other problems that fall under the category of pattern recognition and classification. ANNs can identify and learn correlated patterns between input data sets and corresponding target values. ANNs can be used to predict input data after the training. It follows the learning process of the human brain. It can process problems involving non-linear and complex data. ANNs can also deal with the imprecise and noisy data. The majority of the studies rely on a gradient algorithm, such as Backpropagation algorithm to obtain the weights of the model and to learn the Artificial neural network.

Back propagation is a topology of artificial neural network; it adjusts the network’s weights and biases by calculating the gradient of the error. Usually, back propagation neural networks are applied with random initial weight. Training the neural networks with random initial weights may cause two main drawbacks: trapping into local minima and converging slowly [1]. In view of these limitations of back-propagation neural networks, global search technique such as Genetic algorithm have been presented to overcome these limitations. Genetic Algorithms (GA) are adaptive search and optimization techniques developed to mimic some of the processes observed in natural evolution. Genetic Algorithms have been proposed as one of the potential candidates for optimization of weight of Artificial neural networks. So far a number of works compare the evaluation between back propagation neural network and genetic algorithm for training neural networks, both are techniques for optimization and learning.

As the Back propagation algorithm suffers from many problems, various researchers have been made various attempts to solve these problems using genetic algorithms. Therefore, this study tried to find the optimal initial weights of artificial neural network via genetic algorithm so that the predictor could enhance the ability of predicting the risk.

The remainder of this paper is organized as follows. Section II introduces the artificial neural networks, section III describes the Learning methods, section IV introduces the genetic algorithms, section V shows Related work, section VI describes the Proposed method. In section VII, the advantages of the proposed method is given and finally conclusion is summarized in section VIII.

II. ARTIFICIAL NEURAL NETWORKS

An Artificial Neural Network (ANN) is an information processing paradigm that is based on the function on human brain. The power and usefulness of artificial neural networks have been demonstrated in several applications including speech synthesis, diagnostic problems, medicine, business and finance, robotic control, signal processing, computer vision and many other problems that fall under the category of pattern recognition. There are many different types of neural networks are available : Single layer feed forward network, Multilayer feed forward network, Recurrent network. Among them Multi layer neural networks are the most popular which are extremely successful in pattern reorganization problems[2].Multi layer networks consists of an input layer, a hidden layer and an output layer are made up of no. of highly interconnected processing elements (neurons) working in parallel to solve a specific problem as in Figure 1. In this network data flows through the network in one direction only, from input to output; hence this type of network is called a
feed-forward network. Each connection has a weight factor and these weights are adjusted in a training process. Each neuron has its own unique threshold value. The input units represent the raw information that is fed into the input layer of the network. The activity of each hidden unit is determined by the activities of the input units and the weights on the connections between the input and the hidden units. The behavior of the output units depends on the activity of the hidden units and the weights between the hidden and output units. Each output unit takes as input, a weighted sum of the outputs from the previous layer. It applies a nonlinear or activation function to the weighted input. The behavior of an ANN depends on both the weights and input-output function (transfer function) that is specified for the units.

III. LEARNING METHODS

Learning rules are algorithm for slowly alerting the connections weights to achieve a desirable goal such as minimization of an error function. Learning methods in neural networks can be broadly classified into three basic types: [3]

- Supervised learning
- Unsupervised learning
- Reinforced learning

In Supervised Learning, a supervisor is present during learning process and presents expected output. Every input pattern is used to train the network. Learning process is based on comparison, between network's computed output and the correct expected output, generating "error". The "error" generated is used to change network parameters that results in improved performance. In Unsupervised Learning, there is no supervisor. The expected or desired output is not presented to the network. The system learns of it own by discovering and adapting to the structural features in the input patterns. In Reinforced Learning, a supervisor is present but does not present the expected or desired output but only indicated if the computed output is correct or incorrect. The information provided helps the network in its learning process. A reward is given for correct answer computed and a penalty for a wrong answer. Among all of them, supervised learning is most widely used.

III.1 Backpropagation algorithm

Backpropagation is the most widely used supervised learning method and is a learning rule for multi-layered Neural Networks. Back-Propagation networks are fully connected, layered, feed forward networks, in which activations flow from the input layer through the hidden layer(s) and then to the output layer. In Back propagation, the network is trained using data for which inputs as well as desired outputs are known. In order to train a neural network to perform some task, the weight of each unit must be adjusted, in such a way that the error between the desired output and the actual output is reduced[4]. The algorithm gives a prescription for adjusting the initially randomized set of synaptic weights.

Training algorithm of backpropagation include four stages as:

- Initialize the weight.
- Feed forward.
- Back propagation of errors.
- Updating of the weights and biases.

In Artificial Neural Networks, the training begins by constructing a network with the desired number of hidden and output units and initializing all network weights to small random values. There are two phases involved to train such a network. In first, the inputs are propagated forward to compute...
the outputs for each output node. Then, each of these outputs are subtracted from its desired output, causing an error [an error for each output node. In the second phase, each of these output errors is passed backward and the weights are updated. These two phase are repeated until the network performs acceptably well or the termination criteria is reached. Leaning of Artificial neural network using back propagation algorithm causes two drawbacks : trapping into local minima and converging slowly[1]. So to overcome such drawbacks of back propagation neural network, Global search techniques such as Genetic algorithm, particle swarm optimization have been presented. Both are techniques for optimization and learning.

IV. GENETIC ALGORITHM

Genetic Algorithms developed in 1970 by John Holland, are adaptive search and optimization techniques developed to mimic some of the processes observed in natural evolution. Genetic Algorithms have been proposed as one of the potential candidates for optimization of weight parameters of neural network. Genetic Algorithms perform directed random searches through a given set of alternatives to find the best alternative with respect to given criteria of fitness [5]. It closely resembles the natural process of regeneration, reproduction, inheritance and evolution. Genetic algorithms are typically used for problems that cannot be solved efficiently with traditional techniques. GA seems to be useful for searching very general spaces and optimization problems. Each solution generated in GA is called a chromosome (or an individual). Each chromosome is made up of genes, which are the individual elements (alleles) that represent the problem. The collection of chromosomes is called a population. The internal representation of the chromosome is known as its genotype. This can be either bit strings or gray codes or hexadecimal codes. The real world representation of the genotype is known as the phenotype. Genetic Algorithms work with population of individual strings, each representing a possible solution to the problem considered. Each string is assigned a fitness value accessing how good the solution is, to that particular problem. The string having high fitness values, participate in reproduction yielding new strings by cross over. The least fit individuals are discarded out. A whole new set of population, containing characteristics, which are better than their ancestors, are generated by selecting the high fit individuals. Progressing in this way, after many generations, the entire population inheriting the best characteristics is formed. If the Genetic Algorithm is well implemented, the most promising areas of search space are explored, with the population having fitness values increasing towards the global optimum. A population is said to have converged if 95% of the individuals constituting the population share the same fitness value [6].

GA Operators :

Crossover, mutation and selection are the primary operations of a genetic algorithm. Crossover or the recombination operator forms offspring(s) or new individuals by combining certain portions of two individuals (parents) currently in the population. There are various types of recombination methods such as single point crossover, two point crossover, order based crossover, cycle crossover etc. In Crossover, Two parents are selected. A crossover point is chosen at random. Offspring 1 is built by combing the left side genes of the parent 1 with the right side genes of the parent 2 taking crossover point as the median. Offspring 2 is built by combing the right side genes of the parent 1 with the left side genes of the parent 2 taking crossover point as the median[7].

Mutation is a variation operator that induces a change in the genotype of the individual which in-turn gets reflected in the phenotype. The most common way of implementing mutation is to select a bit at random and flip (change) its value. Selection is the process of rating the fitness of each individual and preferentially selecting the best individual. This criterion ensures that only the fittest individual is passed on to the next generation.

A typical genetic algorithm consists of the following steps:

1. Creating an initial population.
2. Evaluating each individual(chromosome) in the population.
3. Repeating step 2 until the termination condition is reached, or an optimal solution is obtained.
4. Return the best individual as the solution.

V. RELATED WORK

Garima Singh and Laxmi Srivastava[8] introduced a model for Genetic Algorithm-Based Artificial Neural Network for Voltage Stability Assessment. In which, genetic algorithm based back propagation neural network (GABPNN) has been proposed for voltage stability margin estimation which is an indication of the power system’s proximity to voltage collapse. The proposed approach utilizes a hybrid algorithm that integrates genetic algorithm and the back propagation neural network. The proposed algorithm aims to combine the capacity of GAs in avoiding local minima and at the same time fast execution of the BP algorithm. This algorithm uses the real encoding method. In this paper the performance of the proposed GABPNN approach has been compared with the most commonly used gradient based BP neural network by estimating the voltage stability margin at different loading conditions in 6-bus and IEEE 30-bus system. Based on this analysis it could be concluded that the value of this hybrid approach is that GA requires no gradient information so less susceptible than backpropagation to local variations in the error surface. As well as GA based neural network learns faster, at the same time it provides more accurate voltage stability margin estimation as compared to that based on BP algorithm. So that the proposed approach provides acceptably good generalization ability during testing and found computationally efficient in VSM estimation.
Yu-Tzu Chang, Jinn Lin [9] introduced a model for Optimization of the Initial Weights of Artificial Neural Networks via Genetic Algorithm Applied to Hip Bone Fracture Prediction. This paper aims to find the optimal set of initial weights to enhance the accuracy of artificial neural networks (ANNs) by using genetic algorithms (GA). In this paper three-layer (one hidden layer) ANNs models with back-propagation training algorithms were adopted. Here, the genetic algorithm should be further modified to improve the performance because of the data of hip fracture cases were highly nonlinear and complex. This study used the real coded method for describing the chromosomes and Roulette wheel selection operator for creating the mating pool of chromosome for reproduction as well as uses four point crossover method. The fitness function here was the back-propagation neural network. The mutation operator used in this study was non uniform mutation. Based on this analysis it could be concluded that the genetic algorithm obtained a good result on modeling data and on testing data for small range of initial parameter. Thus, study for using simple GA has been proved to be effective for improving the accuracy of artificial neural networks.

Marwan. A. Ali, Mat Sakim. H. A [10] introduced a new method of learning Multilayer Neural Network (NN) using Genetic Algorithms (GAs) techniques. In which the evolutionary techniques based on GAs are studied and employed for the Model Reference Adaptive Control (MRAC) scheme of different plants. In this method the GAs are used for selecting an optimal number of hidden nodes for the neural controller, as well as training the network to minimize the error between the output of the plant and the output of the model reference. They also perform some of the simulation examples to demonstrate how the hidden nodes are adapted through the generation until they reach their optimal integer number, which depends on the complexity of the controlled plant. In this work Variable chromosome representation was introduced with real-coding operators of GAs because of the limitations of traditional binary coding. Moreover, a hybrid selection method plus elitism strategies are used for reproduction process of the GAs. So that the applicability of GAs with real-coding has made its existing operators to be used without any modification while handling these variable strings length. From the analysis, it could be concluded that the proposed neural genetic controller could control different plants to follow the desired model with an acceptable accuracy. It also shows the power of GAs to find the optimal hidden nodes for this controller without reducing its performance.

Subhra Rani Patra, R. Jehadeesan [11] introduces a method which provides the construction of Genetic Algorithm based Neural Network for parameter estimation of Fast Breeder Test Reactor (FBTR) Subsystem. The network was implemented to predict Primary and Secondary Sodium temperatures of Intermediate Heat Exchanger of Fast Breeder Test Reactor. The network has been trained with 92 training samples using both Standard Back Propagation algorithm and Genetic Algorithm based Back Propagation algorithm. In this method the training data have been normalized to be in the binary form for speedy training of the network. In this model about 90% of the data has been used in the training set and the rest of the data has been used for validation of the network model. From the analysis it could be concluded that Genetic Algorithm based Neural Network is a useful method for prediction of parameters in Nuclear Reactor Subsystems with less number of iterations compared to Back Propagation algorithm providing acceptably good generalization ability and faster convergence. Thus a lot of time can be saved using this model without sacrificing the appreciable computational accuracy. Thus, this has been proved to be a quite straight forward approach to improve the capability of parameter estimation using Neural Network.

VI. PROPOSED METHOD

The sequential steps of Proposed method to optimize the weights of Artificial Neural Networks using different parent selection methods of Genetic algorithms is summarized as follows:
1. Read the data file of the problem.
2. Initialize structure of the ANN, that is, input-hidden-output nodes for determining the number of weights by considering number of inputs and number of outputs according to given problem.
3. Decide the size of Initial population and the encoding method for genetic optimization of weights.
4. Extract weight values for chromosome.
5. Train the ANN and get the output, than compute its total mean square error between actual and target outputs.
6. Determine fitness value of chromosome.
7. Repeat the steps 4-6 for every chromosome.
8. Apply parent selection methods to select the best chromosome from the population.
9. Perform the crossover and mutation operations to generate new offsprings.
10. Get the new population.
11. Repeat the steps 4-10 until stopping conditions reach.

VII. ADVANTAGES OF PROPOSED METHOD

In the previous methods, the roulette wheel selection method is used which have a problem when the fitness values differ very much. If the best chromosome fitness is 90%, its circumference occupies 90% of Roulette wheel, and then other chromosomes have too few chances to be selected. It results in premature convergence and a loss of diversity which is definitely not advantageous for the optimization process. There is no guarantee that the best solution will be copied.

In the proposed method, the tournament selection method with elitism is used. As it is known that an ideal selection strategy should be such that it is able to adjust its selective pressure and population diversity. In tournament selection, Selection pressure is easily adjusted by changing the tournament size. it is also efficient to code, works on parallel architectures and allows the selection pressure to be easily...
adjusted. In tournament selection, the fitness difference provides the selection pressure, which drives GA to improve the fitness of the succeeding genes. So it is shown that method is more efficient, generated very fit populations generation and leads to an optimal solution. It also uses the rank selection when there is so much difference between the fitness value of chromosome to avoid the premature convergence. It preserves diversity and hence leads to a successful search. In effect, potential parents are selected and a tournament is held to decide which of the individuals will be the parent.

In the proposed method, it uses the genetic algorithm to optimize the weights of ANN. So that it can efficiently optimize the weights of ANN. For determining the fitness value of chromosome it compute its total mean square error between actual and target output to determine the fitness value of the chromosome instead of using the backpropagation algorithm as in the previous work to determine the fitness value.

VIII. CONCLUSION AND FUTURE WORK

From the above analysis it could be conclude that GA requires no gradient information so less susceptible than backpropagation. Instead of starting with a single candidate, They operate in a population of possible solution candidates in parallel and iteratively operate on it using some sort of heuristics. Therefore, genetic algorithm can efficiently optimize the weights of the Artificial Neural Network. It is also concluded that the weights and biases are trained satisfactorily compared to the traditional ANN and the relative difference between the traditional ANN and GA-ANN is also satisfactorily. Our future work is as follows: First, to use different encoding methods to improve efficiency of genetic algorithms. Secondly, it can use the Tournament selection method with Elitism as well as whenever needed it can use the Rank based selection method along with the tournament selection to select the best individual from the given population.

REFERENCES


