

Neural Networks using Genetic Algorithms

Richa Mahajan

Student, Guru Nanak Dev University, Amritsar

Gaganpreet Kaur

Student, Guru Nanak Dev University, Amritsar

ABSTRACT

Combining neural network with evolutionary algorithms leads to evolutionary artificial neural network. Evolutionary algorithms like GA to train neural nets choose their structure or design related aspects like the functions of their neurons. Along basic concepts of neural networks and genetic algorithm this paper includes a flexible method for solving travelling salesman problem using genetic algorithm. This offers a solution which includes a genetic algorithm implementation in order to give a maximal approximation of the problem with the reduction of cost.

Keywords

Genetic algorithm, Neural network, Travelling Salesman problem.

Related Work

(D. Whitley, 1995) in “Genetic Algorithms and Neural Networks” has described that how the genetic algorithm can make a positive and competitive contribution in the neural network area. Also describes the different ways in which genetic algorithms have been used in conjunction with neural networks.

(Montana and L. Davis, 1989) in “Training feedforward neural networks using genetic algorithms” has explained that multilayered feedforward neural networks possess a number of properties which make them particularly suited to complex pattern classification problem. Along with they also explained the concept of genetics and neural networks.

(D. Arjona, 1996) in “Hybrid artificial neural network/genetic algorithm approach to on-line switching operations for the optimization of electrical power systems” had intended to present an approach to decision making in the operation of electric power systems that will use a simple genetic algorithm as a teacher for the process of supervised learning of a feedforward, backpropagation artificial neural networks.

(Phogat, 2012) in “Travelling Salesman Problem using Genetic Algorithm” had included a flexible method for solving the travelling salesman problem using genetic algorithm. In this problem TSP is used as a domain. TSP has long been known to be NP-complete and standard example of such problems. This paper offers a solution which includes a genetic algorithm implementation in order to give a maximal approximation of the problem with the reduction of cost. In genetic, crossover is a main operator for TSP. The algorithm crossover is as work proposed here intends to compare the efficiency of the new crossover operator with some existing crossover operators.

1. INTRODUCTION

Genetic algorithm and neural networks are both inspired by computation in biological system. A good deal of biological neural architecture is determined genetically.

Neural networks and genetic algorithms are two techniques for optimization and learning, each having its own strengths

and weaknesses. The two have generally evolved along separate paths. However, recently there have been attempts to combine the two technologies. [1][2]

This paper is having mainly seven sections. Section 2 and 3, helps the reader to understand the concept of genetic algorithm and neural network respectively. Section 4, illustrates how to apply genetic algorithm to neural networks. Section 5, depicts the implementation of genetic algorithm in travelling salesman problem. Section 6, explains the application area genetic algorithms / neural network and section 7, describes pros and cons.

2. Genetic Algorithm

Dr. David Goldberg, 1989 offered the following definition: "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.

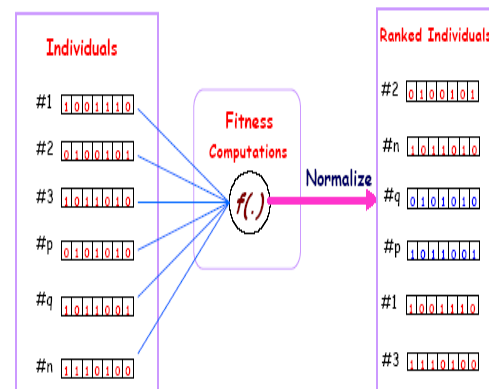


Figure 1: Fitness function [4]

Genetic algorithms consist of a population of binary bit strings, Initial values are determined randomly and evaluated. Each combination of ones and zeros is a possibility in the complex space that can be searched and the relation between them is found in an evaluation function that will return a "fitness" or ranking for that particular bit string. [3]

Genetic algorithms have three main operations:

- Reproduction (or Selection),
- Crossover
- Mutation.

a) Reproduction is a process in which individual strings are copied according to their fitness. Whose fitness value is more that is having more chances to survive in next generation.

b) Crossover is a process that can be divided in two steps. First, pairs of bit strings will be mated randomly to become the parents of two new bit strings. The second part consists of choosing a place (crossover site) in the bit string and exchanges all characters of the parents after that point. The

process tries to artificially reproduce the mating process where the DNA of two parents determines the DNA for the newly born.

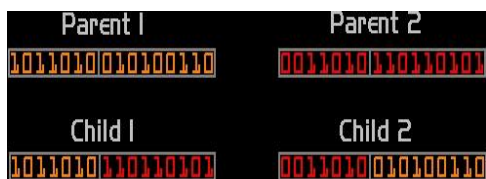


Figure 2: Crossover [4]

In figure 2, crossover site is 7 so after 7th bit the values of parent 1 and parent 2 get interchanged and results as child 1 and child 2.

c) Mutation is included, not because the previous process of reproduction and recombination are not sufficient, but because of the probability that a certain bit can't be changed by the previous operations due to its absence from the generation, either by a random chance or because it has been discarded. It only implies the change of a 0 for a 1 and vice versa.



Figure 3: Mutation [4]

In figure 3, mutation takes place at bit 7, as the value of bit 7 is changed from 1 to 0.

3. Neural Network

Let us review the basics of a neural network. A neural network is a computational model consisting of a number of connected elements, known as neurons. A neuron is a processing unit that receives input from outside the network and/or from other neurons, applies a local transformation to that input, and provides a single output signal which is passed on to other neurons and/or outside the network. Each of the inputs is modified by a value associated with the connection. This value is referred to as the connection strength, or weight, and roughly speaking, represents how much importance the neuron attaches to that input source. The local transformation is referred to as the activation function and is usually sigmoidal in nature.

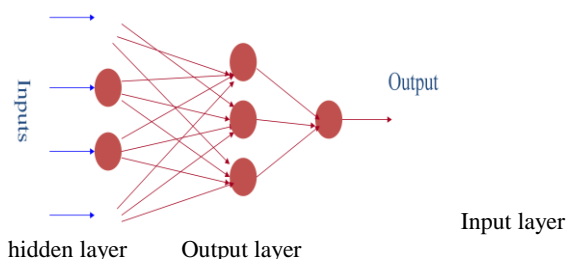


Figure 4: Neural Network [5]

An artificial neural network is composed of many artificial neurons that are linked together according to specific network architecture. The objective of the neural network is to transform the inputs into meaningful outputs. [5]

- The main elements or blocks of an artificial neural network are as follows:
 - a) The computing element (called an artificial neuron or simply neuron)
 - b) The connection pattern among the elements (structure or architecture)
 - c) The process used for training the neural network (learning algorithm)

- An important difference between ANNs and GANNs. In a genetic algorithm only those items of data that have value in predicting the outputs are retained as inputs to the system. A neural network, on the other hand, does not exclude irrelevant data inputs from the final system. It nullifies the effects of such data inputs by assigning a low weight to them in the decision process. [6]

3.1 How to apply GA to neural networks [7]

Combining Neural Nets with Evolutionary Algorithms leads to Evolutionary Artificial Neural Networks (EANNs). One can use Evolutionary Algorithms like the GA to train Neural Nets, choose their structure or design related aspects like the function of their neurons.

3.1.1 Using GA to Train Neural Network

First why one use GA to train Neural Networks: GA will train the network no matter how it is connected - whether it's a feed-forward or a feedback network. Furthermore, it can train general networks which are a mixture of the two types.

a) How to create a string or chromosome from simple neural network.

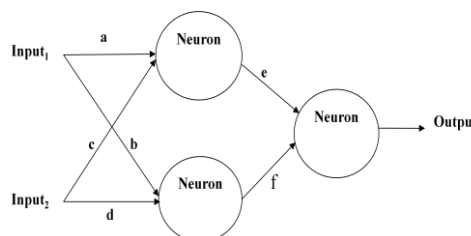


Figure 5:

Simple Neural Network

All the weights in the network are joined to make one string. This string is then used in the GA as a member of the population. Each string represents the weights of a complete network.

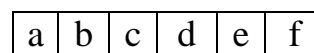


Figure 6: String or Chromosome

Figure 6 depicts the value of chromosome obtained from simple neural network as figure 5.

b) How to evaluate Fitness
 Fitness is measured by calculating the error (target – output) (i.e. fitness= 1/error) - the lower the error the higher the fitness.

Example:

The target for a network with a particular input is 1. The outputs are shown below, calculate their fitness.

Population member	Output
1	0.4
2	0.2
3	1.6
4	-0.9

One can complete the entities below by first calculating the error as described above. Then making all the errors positive and finally working out a fitness (low errors have a high fitness) by using fitness = 1 / error.

Population member	Output	Error (T-O)	Positive	Fitness
1	0.4	0.6	0.6ss	1.67
2	0.2	0.8	0.8	1.26
3	1.6	-0.6	0.6	1.67
4	-0.9	1.9	1.9	0.53

So members 1 and 3 (which are closest to the target) have the highest fitness.

3.1.2 Using GA to Select ANN Topology:

By using genetic algorithm one can evaluate the how neurons are connected with one another in a network.

Case 1: Simple Neural Network

Consider a simple neuron network. If there is a connection of one neuron with other neuron, it will be represented by 1 otherwise 0.

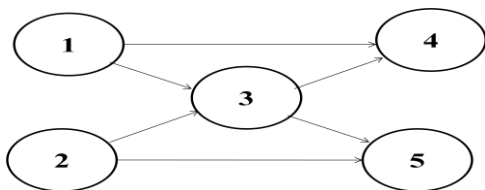


Figure 7: Simple Neural Network

From figure 7, consider the connections from neuron 1. These may be represented by the string shown below:

0 0 1 1 0

The first zero represents the fact that neuron 1 is not connected to itself. The second zero means that neuron 1 is not connected to neuron 2. The third digit, which is 1, means that neuron 1 is connected to neuron 3; and so on.

The complete network may be represented by the matrix shown in figure 8.

0 0 1 1 0	Neuron 1
0 0 1 0 1	Neuron 2
0 0 0 1 1	Neuron 3
0 0 0 0 0	Neuron 4
0 0 0 0 0	Neuron 5

Figure 8: Matrix representing the complete network.

Where matrix element M_{jk} is 0 if there is no connection between neuron j and k ; if the matrix element is a 1, then there is a connection.

It is possible to concatenate the matrix into one string from figure 8:

0 0 1 1 0 0 0 1 0 1 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0

Each string represents the connection pattern of a whole network.

Case 2: Neural network with weights

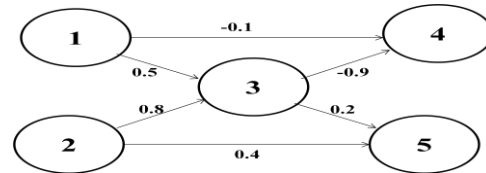


Figure 9:

Neural networks with weights

Figure 9, matrix representing network as:

Connections	Weights
0 0 1 1 0	+0.0 +0.0 +0.5 -0.1 +0.0
0 0 1 0 1	+0.0 +0.0 +0.8 +0.0 +0.4
0 0 0 1 1	+0.0 +0.0 +0.0 -0.9 +0.2
0 0 0 0 0	+0.0 +0.0 +0.0 +0.0 +0.0
0 0 0 0 0	+0.0 +0.0 +0.0 +0.0 +0.0

Figure 9, an alternative matrix representing as:

+0.0 +0.0 +0.0 +0.5 - 0.1 +0.0
 +0.0 +0.0 +0.0 +0.8 +0.0 +0.4
 +0.0 +0.0 +0.0 +0.0 - 0.9 +0.2
 +0.0 +0.0 +0.0 +0.0 +0.0 +0.0
 +0.0 +0.0 +0.0 +0.0 +0.0 +0.0

This corresponds to the string:

0 0 0 0.5 0.1 0 0 0 0 0.8 0 0.4 0 0 0 0 -0.9 0.2 0 0 0 0 0 0 0 0 0 0 0 0

In this case, a weight of zero simply means that no connection exists between these neurons.

4. Implementation of GA in Travelling Salesman Problem [8]

The genetic algorithms are more appropriately said to be an optimization technique based on natural evolution. They include the survival of the fittest idea algorithm. The idea is to first 'guess' the solutions and then combining the fittest solution to create a new generation of solutions which should be better than the previous generation. [10]

4.1 Solution of Tsp Using GA:

A genetic algorithm can be used to find a solution is much less time. Although it might not find the best solution but can find a near perfect solution for 100 city tour in less than a minute. There are basic steps for solving the TSP using a GA.

- First, create a group of many random tours in what is called a population. This algorithm uses a greedy initial population that gives preference to linking cities that are close to each other.
- Second, pick 2 of the better (shorter) tours as a parents in the population and combine them to make 2 new child tours. The new child tours are inserted into the population replacing two of the longer tours.

4.1.1 Methodology:

Simple GA works by randomly generating an initial population of strings, which is referred as gene pool and then applying (possibly three) operators to create new, and hopefully, better populations as successive generations.

- The first operator is reproduction where strings are copied to the next generation with some probability based on their objective function value.
- The second operator is crossover where randomly selected pairs of strings are mated, creating new strings.
- The third operator, mutation, is the occasional random alteration of the value at a string position.

The crossover operator together with reproduction is the most powerful process in the GA search. Mutation diversifies the search space and protects from loss of genetic material that can be caused by reproduction and crossover. So, the probability of applying mutation is set very low, whereas the probability of crossover is set very high.

4.1.2 Steps of Algorithms:

Step-1: Randomly create the initial population of individual string of the given TSP problem and create a matrix representation of the cost of the path between two cities.

Step-2: Assign the fitness to each chromosome in the population using fitness criteria measure.

$$F(x) = 1/x$$

Where, x represents the total cost of the string. The selection criteria depend upon the value of string if it is close to some threshold value.

Step-3: Create new off-spring population from two existing chromosomes in the parent population by applying crossover operator.

Step-4: Mutate the resultant off-springs if required.

NOTE: After the crossover off spring population has the fitness value higher than the parents.

Step-5: Repeat step 3 and 4 until one get an optimal solution to the problem.

4.1.3 Flow Chart:

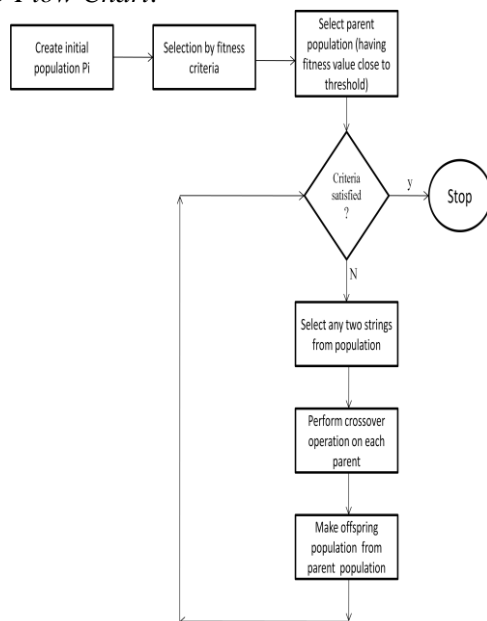


Figure 10: Flow Chart of GA.

Figure 10, shows the various steps are following while implementing genetic algorithms to travelling salesman problem. First create initial population, and then evaluate fitness of all the chromosomes by applying fitness function to it.

After calculating fitness, the chromosome having fitness value close to threshold will select as parent for next generation. If it satisfies the criteria then stop otherwise select any two strings from initial population and then perform crossover on it to form offspring and again send to check it satisfies criteria or not, if yes then stop otherwise continue this cycle until it satisfies.

Operations perform in GA:

- Genetic coding
- Fitness function
- Selection process
- Crossover operator

a) Genetic coding:

To apply GA for any optimization problem, one has to think a way for encoding solutions as feasible chromosomes so that the crossovers of feasible chromosomes result in feasible chromosomes. The techniques for encoding solutions vary by problem and, involve a certain amount of art. For the TSP, solution is typically represented by chromosome of length as the number of node in the problem. Each gene of a chromosome takes a label of node such that no node can appear twice in the same chromosome. There are mainly two representation methods for representing tour of the TSP – adjacency representation and path representation. Consider the path representation for a tour, which simply lists the label of nodes. For example, let {1, 2, 3, 4, 5} be the labels of nodes in a 5 node instance, then a tour {1, 3, 4, 2, 5, 1} may be represented as (1, 3, 4, 2, 5).

b) Fitness function:

The GA's are used for maximization problem. For the maximization problem the fitness function is same as the objective function. But, for minimization problem, one way of defining a 'fitness function' is:

$$F(x) = 1/f(x)$$

Where f(x) is an objective function. Since, TSP is a minimization problem; consider this fitness function, where f(x) calculates cost (or value) of the tour represented by a chromosome.

c) Selection Process:

In selection process, chromosomes are copied into next generation with a probability associated with their fitness value. By assigning to next generation a higher portion of the highly fit, Chromosomes, reproduction mimics the Darwinian survival-of-the-fittest in the natural world. In this paper author uses Elitism method for selection. Elitism is name of method, which first copies the best chromosome (or a few best chromosomes) to new population. The rest is done in classical way. Elitism can very rapidly increase performance of GA, because it prevents losing the best found solution.

d) Crossover Operator:

The search of the solution space is done by creating new chromosomes from old ones. The most important search process is crossover. Firstly, a pair of parents is randomly selected from the mating pool. Secondly, a point, called crossover site, along their common length is selected, then

before crossover point one use method of sequential constructive crossover operator and the information after the crossover site of the two parent strings are swapped, if a gene has already been copied into the off-spring then replace that gene by unvisited gene, thus creating two new children. [11]s

4.1.4 The Algorithm for Sequential Constructive Crossover Technique is as Follows:

Step-1: Start from the node p, the first node in parents P1 and P2.

Step-2: Sequentially search both of the parent chromosomes and consider the first legitimate node appeared after node 1 in both P1 and P2 Suppose the node x and node y are found in P1 and P2 respectively. Consider crossover point is selected after 2nd node in both parents P1 and P2.

Step-3: Now if $C_{px} < C_{py}$, select node x, otherwise node y as the next node and concatenate it to the partially constructed offspring chromosome.

Step-4: Now if one select node x as the next string in partially constructed offspring chromosome, copy the rest of the genes from parent P2 and otherwise copy it from P1.

Step-5: Suppose a gene has already been copied into the off-spring then replace that gene by unvisited gene.

Illustrate Crossover Technique:

Example is given as cost matrix in Table 1. Let a pair of selected chromosomes be P1: (1, 3, 6, 4, 5, 7, 2) and P2: (1, 5, 4, 2, 6, 3, 7) with values 365 and 349 respectively.

TABLE 1. COST MATRIX

N	1	2	3	4	5	6	7
1	100	75	99	9	35	63	8
2	51	100	86	46	88	29	20
3	50	5	100	16	28	35	28
4	20	45	11	100	59	53	49
5	86	63	33	65	100	76	72
6	36	53	89	31	21	100	52
7	58	31	43	67	52	60	100

Select first node in parents P1 and P2 as node 1.now the legitimate node after node 1 in both the parents p and p are node 3 and node 5 respectively. Consider crossover point after node 3 and node 5 in parents P1 and P2 respectively. Now calculate the value of $C_{13}=99$ and, accept 'node 5' as the next cost of the path as $C_{15}=35$. Since $C_{15} < C_{13}$ node in partially constructed chromosome. Node 5 is selected from the parent P2 so add the rest of the bits from the parent P1. Now the partially constructed chromosome is (1,5,6,4,5,7,2).Node 5 has already been copied into off-spring so replace 'node 5' by unvisited 'node 3'.Thus the complete off -spring will be (1,5,6,4,3,7,2) with value 263 which is less than value of both the parent chromosomes. The crossover is shown in the figure. Parents are shown as figure 11 and figure 12 while off-spring are shown as figure 13.

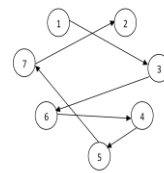


Figure 11, P1: (1, 3, 6, 4, 5, 7, 2)

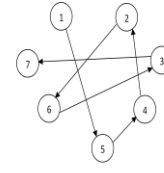


figure 12, P2: (1, 5, 4, 2, 6, 3, 7)

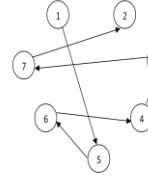
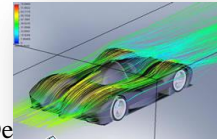


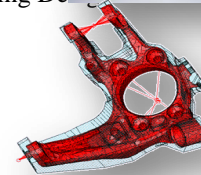
Figure 13, Offspring (1, 5, 6, 4, 3, 7, 2)

5. Application area of GA/ Neural network [9]

a) Automotive Design



b) Engineering Design



c) Robotics



d) Evolvable Hardware



e) Biomimetics Invention



6. Pros and Cons

6.1 Pros:

1. GA helps to generate better population from good parents, these results close to global optimum.
2. Important character of GA, it is robust.
3. They works well in various fields as:
 - a) In pattern matching
 - b) Speech recognition, text-to-speech
 - c) Machines that are able learn
 - d) Optical character recognition (OCR)
 - e) Fraudulent credit card detection (VISA)
 - f) Image compression

6.2 Cons:

1. It remains a 'black box' which once fed with inputs produces an output. However, their excellent result record might compensate for that deficiency.
2. A second drawback is that inputs have to be altered before being fed to the network.
3. It is fail to depict followings:
 - a) Which network (architecture) to use?
 - b) How many hidden layers?
 - c) How many neurons?
 - d) What activation functions should I use?
 - e) What cost function is the most appropriate?
 - f) Which training algorithm to apply?

7. Conclusion

This paper makes an effort to give a review with respect to neural networks, genetic algorithm and how they both work together. Genetic algorithm has three main operators: selection, mutation and crossover. Neural network is computational model having number of processing elements called neurons. These techniques are black box which once fed with inputs produces an output. As genetics and neural networks have a wide real-world application area, also suffers from various cons so in future it will try to work on these limitations.

References

- [1] D. Whitley, "Genetic Algorithms and Neural Networks", book title "Genetic Algorithms in Engineering and Computer Science", publisher "John Wiley, pp 191-201", august 1995.
- [2] D. Montana and L. Davis, "Training feedforward neural networks using genetic algorithms", published in "IJCAI'89 Proceedings of the 11th international joint conference on Artificial intelligence", pp 762-767 vol. 1, 1989.
- [3] D. Arjona, "A hybrid artificial neural network/genetic algorithm approach to on-line switching operations for the optimization of electrical power systems", appears in "Energy Conversion Engineering Conference", pp 2286 – 2290, vol.4, Aug 1996
- [4] M. Harrim, "Genetic Algorithms" <http://seminarprojects.com/attachment.php?aid=40499>
- [5] T. Hussain, "Methods of Combining Neural Networks and Genetic Algorithms", Queens University, 1997.
- [6] F. C. Su and W. L. Wu, "design and testing of a genetic algorithm neural network in the assessment of gait patterns", institute of biomedical engg. National cheng Kung University, pp 67-74, February 2000.
- [7] "Applying Genetic Algorithms to Neural networks", available from www4.rgu.ac.uk/files/chapter15%20-%20eanns.pdf
- [8] A. Phogat, "Travelling Salesman Problem using Genetic Algorithm", IMS Engineering College, NH -24, Adhyatmic Nagar, Dasna – Ghaziabad.
- [9] "15 Real-World Uses of Genetic Algorithms", Web page Available from <file:///C:/Users/Dell/Desktop/15%20Real-World%20Applications%20of%20Genetic%20Algorithms.htm>
- [10] Adewole Philip, Akinwale Adio Taofiki, Otunbanowo kehinde "A Genetic Algorithm for Solving Travelling Salesman Problem", (IJACSA) international journal of advanced computer science Applications, Vol. 2, No. 1, January 2011.
- [11] Koszelew, J., Piwonska, A.: Tuning Parameters of Evolutionary Algorithm in Travelling Salesman Problem: A Guided Tour of Combinatorial optimization.