Linear Network Representation Applied to evolving of the Architecture and Weights of Neural Network

Sally Ali Abdul Lateef
Al-Mustansiryia University College of Educate

Abstract.

Evolutionary computation is a class of global search techniques based on the learning process of a population of potential solutions to given problem that has been successfully applied to a variety of problems. In this paper a new approach to design of the neural networks based on evolutionary computation is present. A linear chromosome representation of the network are used by genetic operators, which allow the evolution of the architecture and weights simultaneously without the need of local weights optimization. This paper describes the approach, the operators and reports results of the application of this technique to several binary classification problems.

1. Introduction

The design of a neural network is still largely performed using a lengthy process of trial and error definition of the topology, followed by the application of learning algorithm such as back propagation [Mor.93].The most ambitious combination attempts to evolve the architecture and weights simultaneously without a separate training process [Ham.97].Two common strategies to do this are the destructive and constructive algorithms.

Evolutionary computation is a class of global search techniques based on the evolution process of each individual [Fuj.94][Ang.94], representing a potential solution to a given problem. Typical evolutionary computation updates this population seeking for better regions of the search space using the operations of selection, recombination and mutation, inspired by biological evolution [Sch.04].

Approaches based on (Genetic Algorithms) usually represent the structure and weights of a neural network as a string of bits or a combination of bits and real numbers [Mor.93][Gru.06] and perform the crossover operation as if the network were a linear structure. However, neural networks cannot naturally be represented as binary vectors. They are oriented graphs, whose nodes are neurons and whose arcs are synaptic connections. Therefore, it is arguable that any efficient approach to evolve (Neural Networks) should use operators based on this structure[Ham.97]. An approach based on genetic programming [GP], has been largely limited by the lack of a good encoding mechanism, which is a first step in this direction, for example, in [Mor.93 and Fog.03] NNs have been represented as parse trees. This representation was proposed by Koza, which is recombined using a crossover operator that swaps sub-trees representing sub-nets. The graph like structure of NNs is not ideally represented directly with parse trees either.

This paper describes a new form of GAs which is suitable for evolution of artificial NNs.

It represents each individual with a linear chromosome of variable size and shape. This feature gives the power and ability of the new approach to passing most problems of representation inGAs.

The new approach allows the use of standard genetic operators (such as one-point crossover, two-point crossover and uniform crossover). In addition, it proposes a new combination of crossover and mutation operators. Applying any one from both standard or special genetic operators allows the evolution of topology and weights of NNs concurrently and very efficiently. The new method has been successfully applied to determination of architecture and weights of (three layers) feed forward networks.

2. Representation

In the new form, instead of the usual parse tree or the graph representation, an array of pointers is used to represent the chromosome. Each pointer represents a neuron and points to a linked list of nodes that describe its connections with the sending neurons in a previous layer.

Graph representation uses a linear genotype and a separate grid description to make it more natural. Our new(linear or two-dimensional) representation avoids this problem where no need to descriptions because it represents NNs directly . The linear chromosome consists of a number of genes which represent the neurons of network.This chromosome is divided into three sub chromosomes. The genes of the first sub chromosome represent the input neurons, the genes of the second sub chromosome represent the hidden neurons, and the genes of the third sub chromosome represent the output neurons. This chromosome uses an array of pointers structure (see figure (1)).

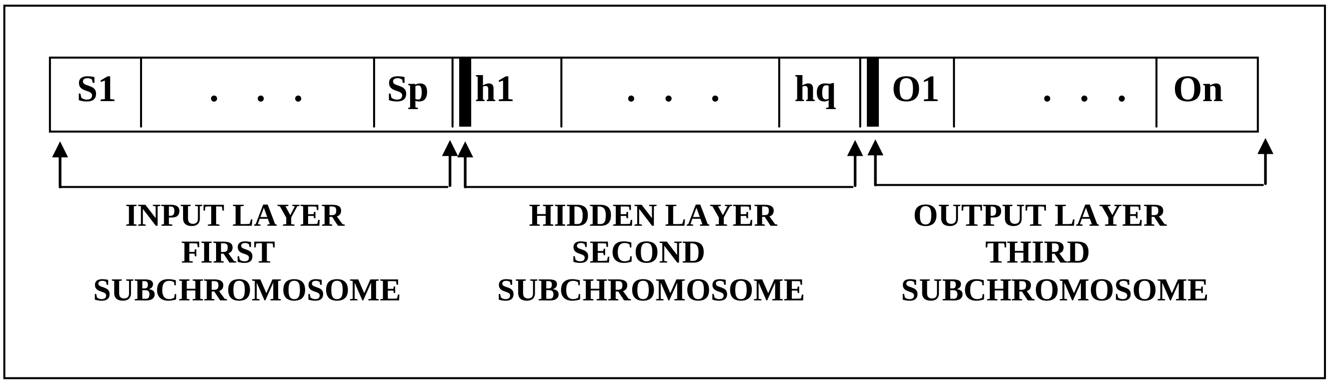

FIGURE (1) The general structure of the representation of three layeres NNs by our new form

In the standard GP the size of chromosome (parse tree) may grow excessively, but the new representation avoids this problem because the length of the chromosome is limited by the length of the array which equals the total sum of neurons in a particular neural network. Graph representation must represent the same number of neurons in all layers, but in our new form no need to do that. This characteristic can make the new method of representation more efficient in using the available memory. Like in standard GP there are three classes of neurons. The input class, the internal class and the output class. Also there are three types of nodes, in the linked lists pointed to by the neurons: function node, terminal node and body node . Each node consists of four fields. The function node fields are (see figure 2.a):

- The first field contains a value representing the number of nodes in the linked list.
- The second field contains a value representing the type of activation function.
- The third field contains a value representing the bias of the receiving neuron.
- The fourth field contains a pointer to next node.

The body node fields are (see figure 2.b):

- The first field contains a value representing the class of sending neuron.
- The second field contains the index of sending neuron.
- The third field contains a weight value of the link.
- The fourth field contains a pointer to the next node.

The terminal node fields are (see figure 2.c):

- The first field contains a value representing the the class of sending neuron.
- The second field contains the index of sending neuron.
- The third field contains the input value or a weight value of the link.
- The fourth field contains the nil pointer.

The input class must contain terminal nodes only, but the internal class and output class may consist of all types of nodes.

In the following are descriptions of the neurons in all parts of the chromosome, see figure (2): -

a. *The first sub chromosome* (Input neurons or input class*)*. The neurons in this part are representing the input layer, which contains input values. The node type in this part is a terminal node only (see figure 3).

b. *The second sub chromosome. (*Hidden neurons or internal class) The neurons in this part are representing hidden layer see figure (4). Each neuron in this sub chromosome points to a linked list of nodes that describe its connections with the sending neurons in the input layer. The nodes of this list consist of three types of nodes, the first is a function node; the last is a terminal node and otherwise is a body node (internal node).

c. *The third sub chromosome.* (Output neurons or output class)The neurons in this part represent output layer see figure (٥). Each neuron in this sub chromosome points to a linked list of nodes that describe its connections with the sending neurons in a previous layer. The nodes of this list consist of three types. The first is a function node; the last is a terminal node and otherwise is a body node (internal node).

The number of input neurons and output neurons depends on the problem (fixed number), and the number of hidden neurons is variable but subject to the condition that at least there is one neuron in the hidden layer connected with input and output layers.

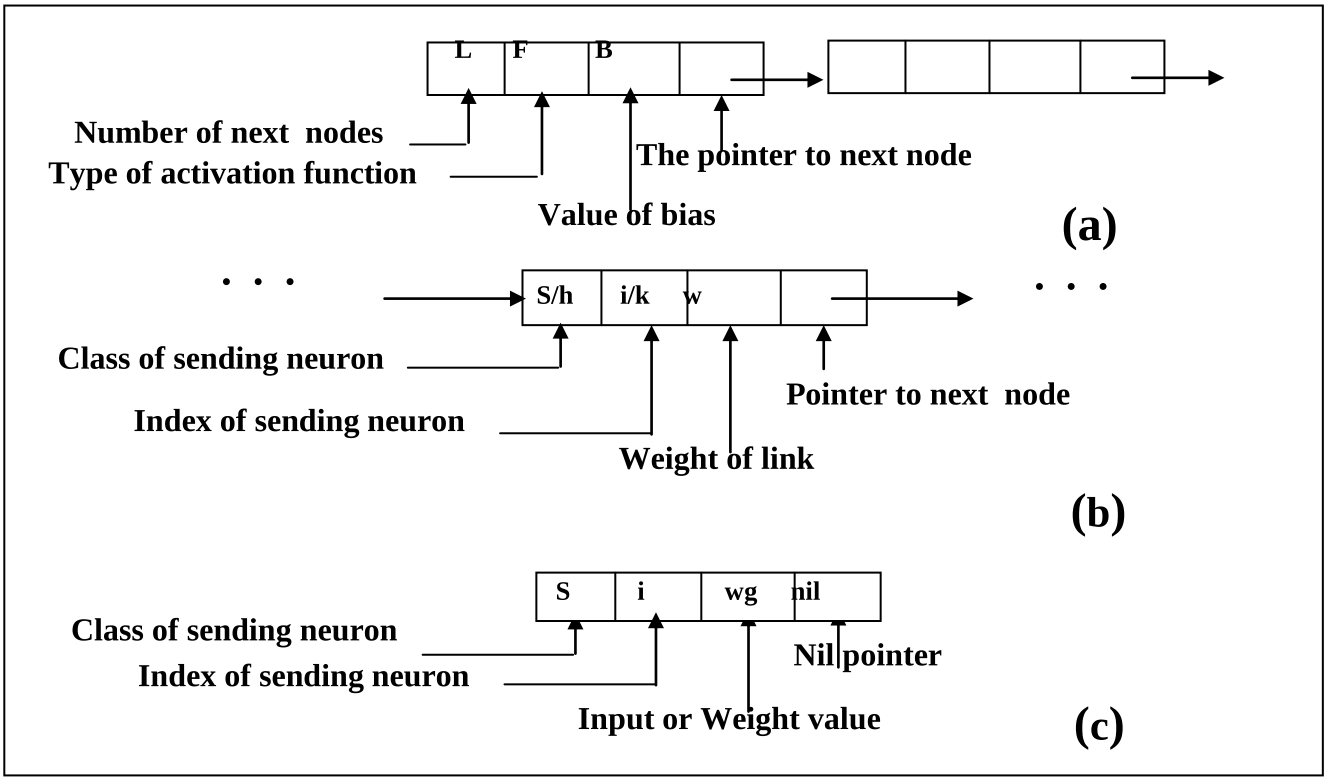

Figure (2) (a). Function node (b).body node (c).terminal node.

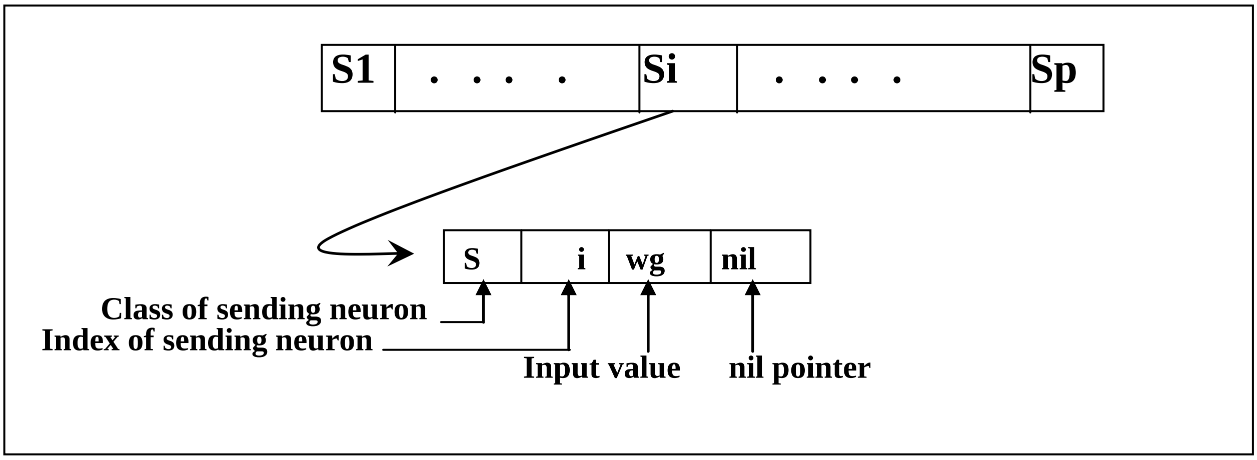

Figure (3) first sub chromosome (input layer)

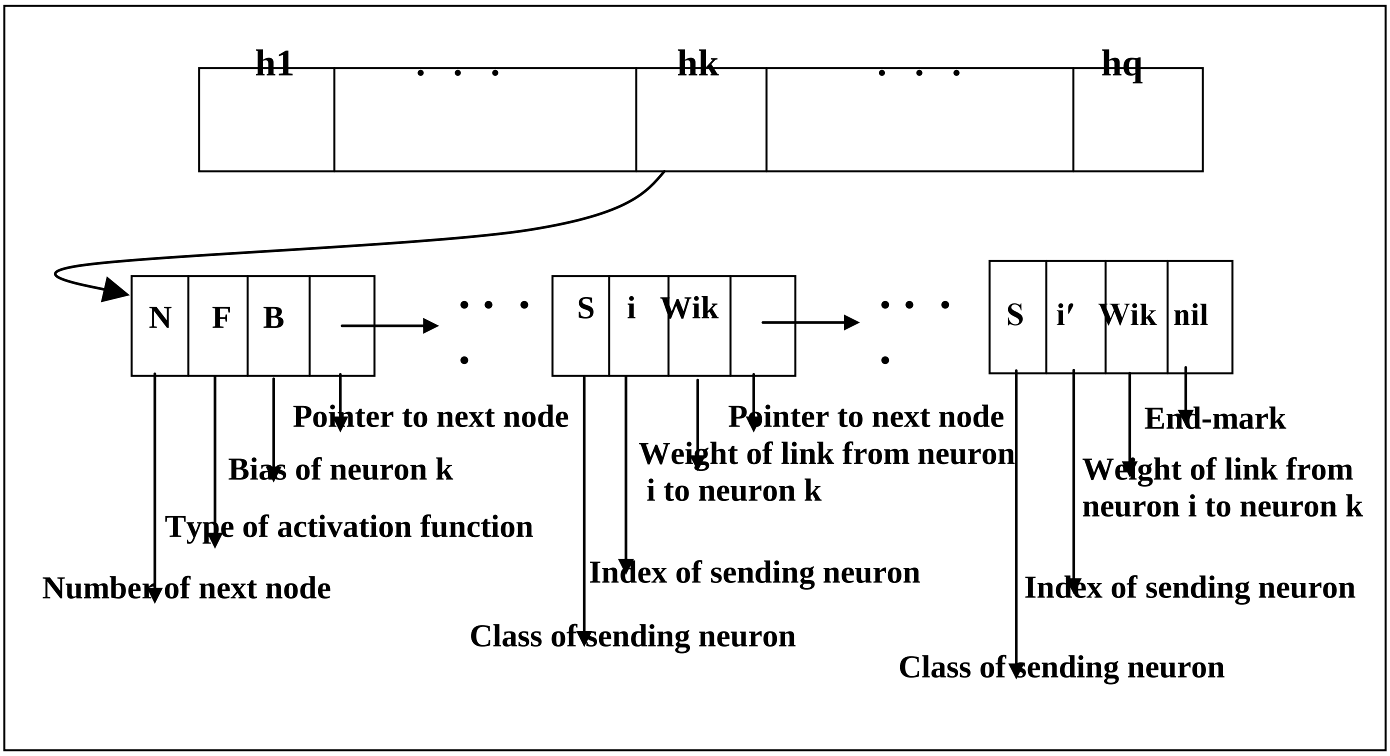

Figure (4) second sub chromosome (hidden layer)

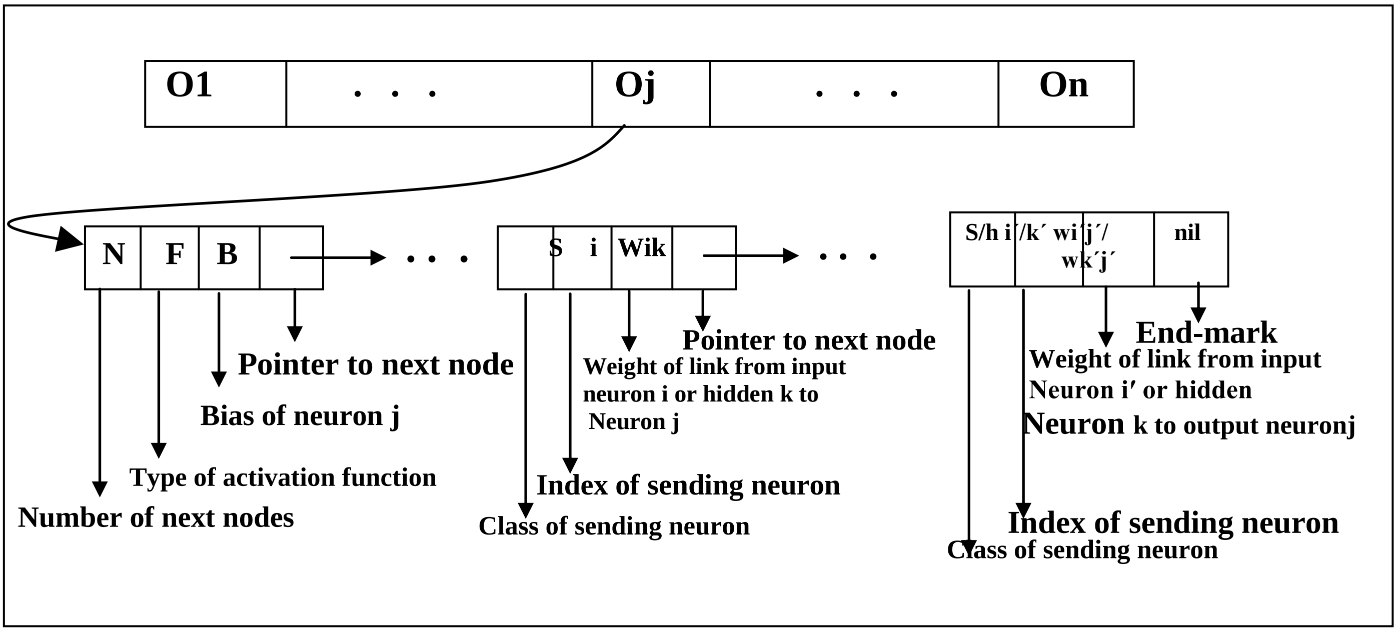

Figure (5) third sub chromosome (output layer)

3. Genetic operators

Genetic operators are performed on the last two subchromosomes.

3.1. Crossover operators

The possibility of the new representation is to make advantage of the linear structure of the chromosome and perform all types of crossover of the standard GA. The types of standard crossover are carried out as follows:

- One point crossover (1-x crossover) The crossover is applied by selecting randomly one point in both parents, and then replacing the block of neurons coming after the cut point(k1) in the first parent by the block of neurons coming after the cut point(k1)in the second parent(see figure5).
- Tow point crossover (2-x crossover) the crossover is applied by selecting randomly two points in both parents (k1 and K2), and then exchanging the blocks of neurons between k1 and K2. (See figure 6.)
- Uniform crossover (multi-point crossover) in the above methods replacement is performed on block level, but this method performs replacement on gene level of both parents. (See figure 7).

The genes of last two subchromosomes of parents are scanned from left to right and exchanged with probability Pc.

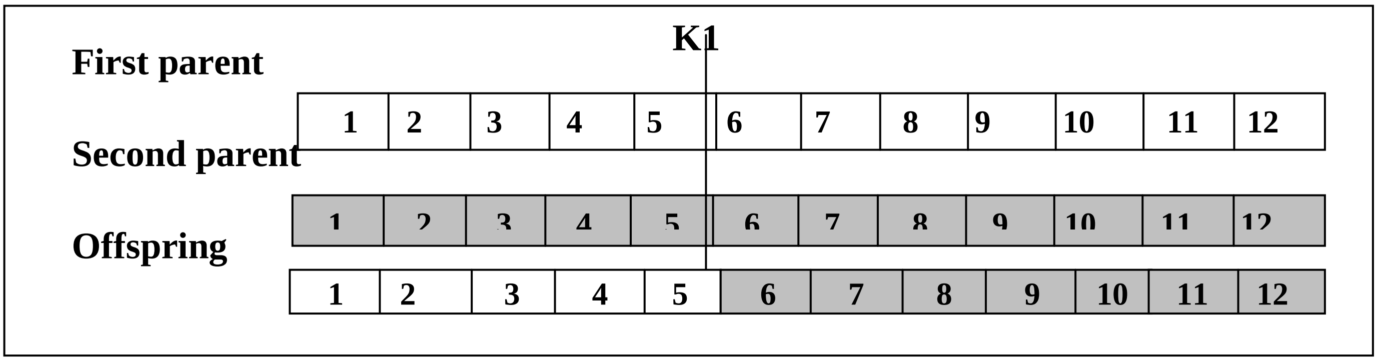

Figure (6) the one point crossover

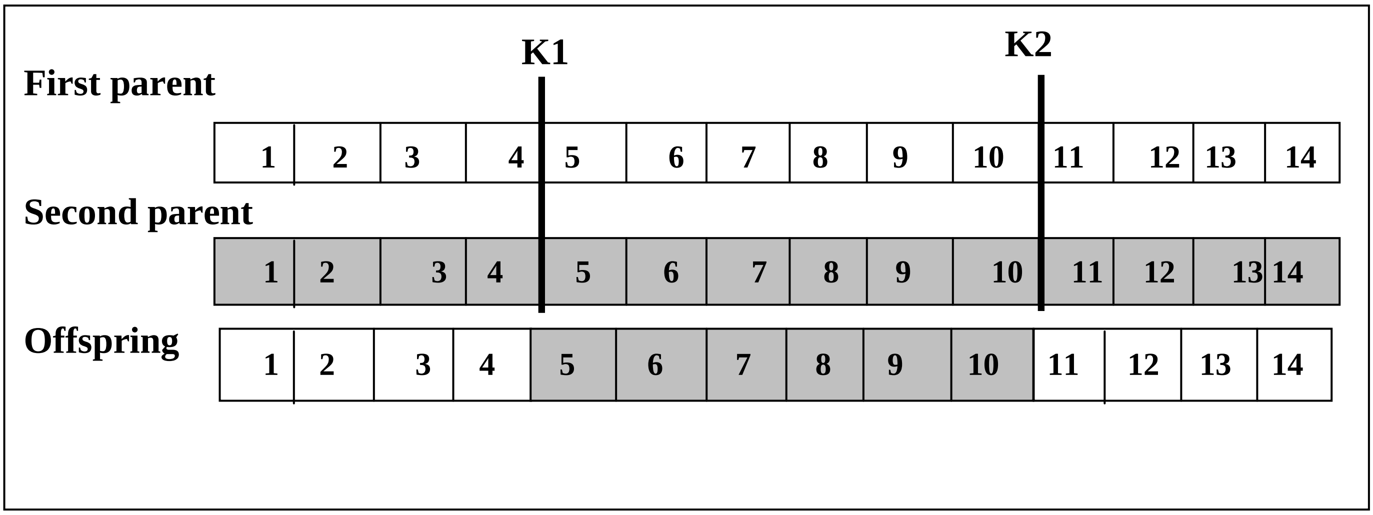

Figure (7) the two point crossover

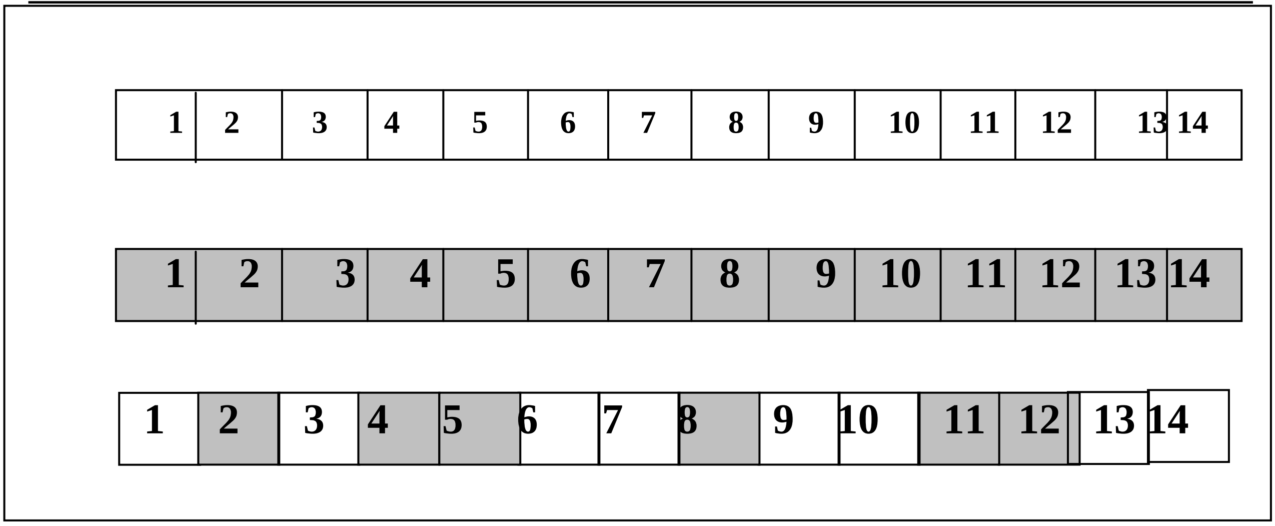

Figure (8) Uniform crossover (multi-point crossover)

3.2 Mutation operator

3.2.1 Changing mutation

The neurons of last two subcromosomes of the child are scaned from left to right and with small probability (Pm) we mutate the bias and weight in all nodes associated with the neuron. Each of the mutation values is generated randomly in the range [-α.. + α].

If a new weight is close to zero, the node is removed. If all weights are close to zero the hidden neuron is removed.

3.2.2 **Removing mutation**

The neurons of second subchromosome of the child are scanned from left to right and with small probability Pr we remove the neuron (remove all its associated nodes and set the neuron to nil).

Note

If all hidden neurons are removed, the child is refused.

4. Results

To show the performance of the method proposed, it was applied to a test suit of benchmarks present in the literature: N-bit odd parity problems (for N=2(XOR),3,4 and 5), where the network output must be 1 if there is an odd number of 1's in the input pattern, and 0 otherwise.

In all experiments a population of 100 chromosomes was evolved for maximum of 100 generations.

For each problem, we performed 50 runs with different random seeds. In the experiments, we used a generational genetic algorithm with: tournament selection (tournament size=2) and crossover and two types of mutation probabilities of 0.4 and 0.09 and 0.1 respectively.

The weights and biases were randomly initialized within the range [-1.0, +1.0].

The mutation operator used was the addition value to the weights during the evolution. The fitness function was the standard mean square error of the output of the network for all input patterns. The function set included a Sigmund activation function.

The results obtained of using the XOR-problem (N-party problems) when applied all type of crossover is shown in the table 1, table2, table3 and table4.

Table 1a-d.

Column 2 represents the average number of generations at which the solution. Column 3 and 4 show the number of hidden neurons and the number of connections of such solution, respectively. Column 5 shows the copulated effort, and column 6 shows the percentages of runs in which solutions were found.

The representation of a typical solution for the XOR problem and the corresponding.

4.1. The best solutions for (N-parity) problems:-

4.1.1. XOR problem:-

4.1. XOR problem:-

Table (1) the results of (XOR) problem for new method

The type of crossover	generations	Hidde neurons Min/avg./max	Connections Min/avg./max	effort	Solutions found
One point crossover	47	4-3.0-2	12 -7.5 - 6	8150	%100
Tow point crossover	45	4-3.0-2	12 -7.5 - 6	7556	%100
Uniform	40	4-3.0-2	12 -7.5 - 6	6975	%100

4.2. 3-Parity problem:-

Table (2) the results of 3-parity problem for new method

The type of crossover	generations	Hidde neurons Min/avg./max	Connections Min/avg./max	effort	Solutions found
One point crossover	60	5-4.0-3	20 -16.0 - 12	8675	%95
Tow point crossover	60	5-2.5-4	9 -7.5 - 6	8426	%96
Uniform crossover	60	5-2.5-2	9 -7.5 - 6	7896	%98

4.3. 4-parity problem:-

Table (3) the results of 4-parity problem for new method

The type of crossover	generations	Hidde neurons Min/avg./max	Connections Min/avg./max	effort	Solutions found
One point crossover	146	6-5.0-4	30 -25.0 - 20	14695	%55
Tow point crossover	138	5-4.5-4	25 -22.5 - 20	13690	%61
Uniform crossover	130	5-4.5-4	25 -4.5 - 20	13459	%65

4.4. 5-parity problem:-

Table (4) the results of 5-parity problem for new method

The type of crossover	generations	Hidde neurons Min/avg./max	Connections Min/avg./max	effort	Solutions found
One point crossover	193	8-7.0-6	48 -42.0 - 36	23280	%10
Tow point crossover	188	8-6.5-5	48 -39.5 - 30	23010	%12
Uniform crossover	179	7-6.0-5	42 -39.0 - 36	21985	%16

5. Further works

1. The results obtained so far are promising, but they were achieved without any optimization of relative probabilities of operators. Although some operators seem to perform better than others.

2. To keep diversity within the population other mutation operators should be explored.

3. The extension to recurrent neural networks is a natural one, allowing the method to be applied to a wide range of tasks.

6. Conclusions

In this paper, a new approach to the automatic design of neural networks based on evolutionary computation has been presented. New operators were introduced, which exploited a liner representation, in which a linear encoding is used in conjunction with an array representation.

The representation of the neural network in a liberalized form allowed the development of efficient forms of crossover operations and the introduction of strategy to reduce the complexity of solutions, whereas the array of pointer description allowed controlling the connectivity properties of the network.

The method was applied to evolve feedforward networks for a variety of binary classification problems showing promising results.

REFRENCES

[Ang,94]

Angeline P. J, Saunders G.,M and Pollack J.B.[1994].

An evolutionary algorithm that constructs recurrent neural networks. *IEEE Transactions on Neural Network*s, 5(1):54–65.

[Fog,03]

Fogel D.[2003].***Evolutionary computation:toward a new philosophy of machine Intelligence.*** IEEE Press, Piscataway, NJ, USA.

[Fuj,94]

Fujita S. and Nishimura H. [1994]. ***An evolutionary approach to associative memory in recurrent neural***

[Gru,06]

Gruau F.[2006]. ***Neural network synthesis using cellular encoding and the gene`tic algorithm.*** PhD thesis, Laboratoire de L'informatique du Parall_elisme, Ecole Normale Sup_eriere de Lyon, Lyon, France.

[Ham,97]

Crossover point (k1)

Hammel U., and Schwefel H.[1997]. ***Evolutionary computation: Comments on the history and current state.*** *IEEE Transactions on Evolutionary Computatio*n, 1(1):3–17, Apr.

[Kit,94]

Kitano H.[1994]. ***Neurogenetic learning: an integrated method of designing and training neural networks using genetic algorithms.*** Physica D, 75:225-238.

[Mor, 93]

Moriarty D. and Miikkulainen R.[1993]. ***Evolving complex Othello strategies using marker-based genetic encoding of neural networks.*** Technical Report AI93–206, Department of Computer Science, The University of Texas.

[Sch, 04]

Schwefel H.[2004].***Evolutionary computation: an overview.*** In *Proceedings of The third IEEE Conference on Evolutionary Computatio*n, pages 20–. IEEE Press.

التمثيل الأحادي للشبكة لتطوير معمارية وأوزان الشبكات العصبية

سالي علي عبد اللطيف

الجامعة المستنصرية/ كلية التربية

الخلاصة:-

الخوارزميات التطويرية هي صنف من تقنيات البحث العالمية تعتمد على نظريات التطور البيولوجي في المعالجة ، من خلال تطبيق العمليات التطويرية من اختيار(selection) وتزاوج(crossover) وطفرة(mutation)، على أفراد في مجتمع من حلول ممكنة لمسألة معطاة، هذه العمليات تطبق بشكل صدفي لتنتقي الفرد الأفضل إنجازية. يقترح هذا البحث طريقة جديدة للتشفير في البرمجة الجينية ملائمة جدا لتطوير الشبكات العصبية الاصطناعية ، إذ يشفير كل فرد على شكل كروموسوم خطي متغير الحجم والشكل، هذه الخصائص التي تميزت بها طريقة التشفير الجديدة اعطت القوة والقدرة على تجاوز معظم المشاكل التى عانت منها مسألة تمثيل الكروموسوم في الخوارزميات التطويرية طبقت الطريقة المقترحة بنجاح لتحديد معمارية واوزان شبكة ثلاثية الطبقة امامية الاتجاه لحل مجموعة من مسائل التصنيف.