Genetic Encoding of Neural Networks using Attribute Grammars

Talib S. Hussain and Roger A. Browse

Department of Computing and Information Science
Queen's University, Kingston, Ont. K7L 3N6

hussain@qucis.queensu.ca

Abstract

The discovery of good neural network solutions to complex problems may be facilitated through the use of evolutionary computation techniques, such as genetic algorithms or genetic programming. One key issue in the development of any system which will evolve neural networks is how and what information about a neural network will be encoded in the genetic description that will be manipulated by the evolutionary processes. Several approaches have been taken to this encoding problem, including direct, structural, parametric, and grammatical encoding. We present a new grammatical encoding technique in which an attribute grammar is used to represent a class of neural networks. We propose that the resulting encoding offers several improvements over existing approaches.

1. Introduction

Neural networks and genetic algorithms are two computational techniques which have been actively and increasingly researched in recent years. Neural networks and genetic algorithms may be considered as complementary search mechanisms, and a number of researchers have proposed systems which combine both techniques to allow the evolution of neural networks (see Yao, 1993 for a more complete review).

In neural networks, computation is performed through the passing of signals within a structured arrangement of connected processing units, in response to given input signals. A neural network, in addition to its connectivity details, usually includes mechanisms which specify how weights on those connections may be changed over time in response to the inputs provided to the network.

In genetic algorithms, computation is performed through the creation of an initial population of individuals followed by the evaluation, synthesis, creation and elimination of individuals over successive generations until a satisfactory solution is found.

In a traditional system which evolves neural networks, a genetic algorithm performs a coarse search through the space of possible initial architectures, as limited by the genetic specification of the neural network model. A neural network with a given initial architecture performs a fine search through the space of possible connection weights as limited by the nature of the learning mechanisms. The resulting system performs a powerful, robust search.

2. Types of Genetic Encoding

The key issue in evolving neural networks is how to represent a neural network architecture in a manner that provides the genetic algorithm with a useful search space. This is termed a *genetic encoding* of a neural network. The nature of the genetic encoding has important consequences for the efficiency and robustness of the evolutionary search. We separate current genetic encoding techniques of neural networks into four different categories.

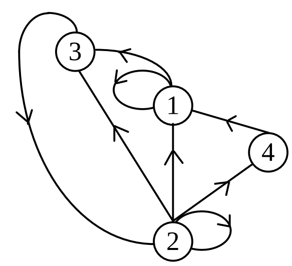
Within a *direct encoding*, the details of a neural network are described in the gene such that the gene may directly be used as a functioning neural network (e.g., The matrix-of-weights gene in Figure 1a, in which the row is the source node). Little or no initialization is required. The genetic algorithm thus operates at a level similar to that of the neural network learning rule. The genetic search space is the space of all possible functioning neural networks with a given number of nodes, and the number of degrees of freedom in that genetic search space is enormous. The search usually does not scale well to large problems.

0.03	0	-0.07	0
0.02	0.5	0.4	0.9
0	0.3	0	0
0.3	0	0	0

(a)

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

(b)



(c)

Figure 1: Direct encoding (a) and structural encoding (b) of a neural network architecture (c)

Within a *structural encoding*, the graphical structure of the network is described in the gene (e.g., The matrix-of-connections gene in Figure 1b). To derive a functioning neural network, some specific values need to be initialized according to formal, pre-determined rules. In particular, a functioning network may be formed only once weight values have been assigned to the connections. Consider, for example, an initialization rule which sets all weights to random values between -0.05 and +0.05. In this case, a given gene will define a specific connectivity pattern, but will represent a number of potential functioning neural networks with different initial weight configurations. Thus, the genetic search space is the space of all possible neural network structures with a given number of nodes, and is smaller than the space of all possible functioning neural networks with that number of nodes. Structural encoding based search is usually faster and more effective than direct encoding based search.

Within a *parametric encoding*, certain important aspects of a neural network architecture are represented by a fixed number of parameters (e.g., The gene of Figure 2 represents three parameters of a back-propagation network). The permissible values of those parameters form a genetic search space. Typically, the number of parameters is very low in comparison to the size of the network. This has three main consequences. Firstly, the steps required to derive a functional neural network are more complex and a number of assumptions must be made in the interpretation. Secondly, the search space is considerably smaller than those of direct and structural encoding and thus the search for a good solution is typically faster. Thirdly, if the parameters used have not been selected well, then the space of possible networks may not be robust and the genetic search may be ineffective for some problems.

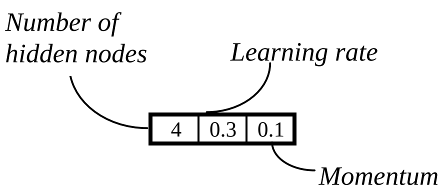
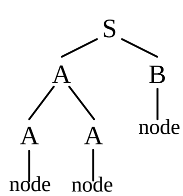


Figure 2: Parametric encoding of a back-propagation network

Finally, within a *grammatical encoding*, a neural network is represented as a sentence of a language described by a grammar. Two basic approaches to grammatical encoding have been examined in the literature. In *developmental grammatical encoding*, the gene actually describes the grammar rules that will be used to develop a specific neural network structure (Kitano, 1990). In *derivation grammatical encoding*, a single fixed grammar is designed and the gene contains a derivation sequence or tree (see Figure 3) which defines a specific network (Jacob and Rehder, 1993; Gruau, 1995). The level of network detail specified in the gene is determined by the nature of the production rules. Our research follows the second approach and examines the use of *attribute grammars* in creating a useful genetic encoding of neural networks.

1. S→AB
2. A→ AA
3. A→ node
4. B→ BB
5. B→ node

(a)



(b)

Figure 3: Context free grammar (a) and tree-based derivation encoding (b) for one possible network

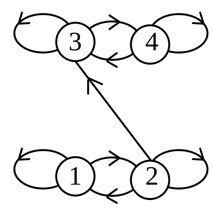
3. Structural Regularity

In neural network architectures, the degree of structural regularity, in the form of hierarchy and modularity in the connectivity, varies from model to model and is usually a fixed aspect of a network design. Such regularities have been shown by many researchers to be an important factor in the development of neural network architectures which scale well to large problems (Jacobs et al., 1991; Boers et al., 1993; Happel and Murre, 1994).

In a direct or structural encoding, few constraints are placed upon the variations in connectivity that the genetic algorithm may explore. For instance, in a connection matrix gene, arbitrary changes to the connectivity are possible. Thus, genetic exploration with such an encoding is not biased towards developing regularities in connectivity. For instance, in Figure 4, an example of structural encoding for a network with hierarchical and modular connectivity is given. Note that in the matrix, the 1's and 0's are distributed in a more organized arrangement than in the matrix of Figure 1b. However, both matrices are equally likely to be considered during the evolutionary process.

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

(a)



(b)

Figure 4: Structural encoding (a) of a network with hierarchical and modular connectivity (b)

In a parametric encoding, it is possible to encode a certain degree of structural regularity explicitly, but solutions that are developed will scale only as well as that degree of regularity permits. For instance, in a gene in which the only structural parameter is the number of nodes in the hidden layer of a back-propagation network, the layered structure is encoded by all genes. However, no variations to the single hidden layer architecture are possible.

Finally, in a grammatical encoding, structural regularities are readily represented. Grammars, by their very nature, enable the concise representation of structures which incorporate hierarchy and modularity. Current grammatical encoding techniques, however, have not exploited this property very well. The grammars developed tend to permit only limited structural regularity due to the difficulty in specifying grammar productions which allow complex connectivity manipulations. For instance, Gruau (1995) has presented the most advanced derivation grammatical encoding technique to date. However, in his approach, a given subtree in the genetic representation does not always have a clear structural meaning. Further, it is unclear whether his approach can be used to explicitly represent more popular neural network architectures. These conclusions are the findings of analyses that we have carried out in our research; space does not permit further substantiation of these claims.

4. Attribute Grammar Encoding

An attribute grammar (Knuth, 1968) is a context-free grammar in which symbols of the grammar are assigned semantic attributes and each production rule specifies not only how symbols are replaced but how their attributes are evaluated. Through the context-free component, an attribute grammar permits the explicit representation of hierarchical, recursive and modular structural design. Through the attribute evaluations, an attribute grammar permits the explicit representation of complex neural network design details.

Many branches of science, including linguistics and computer science, have long recognized the representational power of formal grammars in the description of complex structures and processes. By bringing the power of phrase structure grammars to bear in the specification of classes of neural networks, we have developed a technique in which semantically meaningful substructures may be explicitly represented. Through the use of attribute grammars, structural regularities may be more precisely specified to produce an interesting sub-class of all possible structures for the genetic algorithm to explore.

Given an attribute grammar which defines a class of neural networks, a genetic description of a particular neural network may be formed from the context-free derivation tree. That genetic description may be used naturally in a genetic programming evolutionary technique. To ensure that only valid genes are created during the evolutionary processes, tailored genetic operators must be used (Koza, 1994; Haynes et al., 1996). An example of the creation of a new individual from two parents, based on the grammar in Figure 3 and using a syntactically constrained crossover operator, is illustrated in Figure 5.

Three primary benefits are provided by using this *attribute grammar encoding* (Hussain and Browse, 1998ab). Firstly, the encoding is highly compact since a very complex network, as determined by the complexity of the attributes, is only represented by the concise context-free derivation tree. Secondly, the subtrees in a gene have a clear structural interpretation, and the evolutionary processes will tend to develop networks exhibiting a high degree of structural regularity. Thirdly, an attribute grammar may be written for many different classes of neural networks, including traditional models such as back-propagation. Thus, the technique has wide applicability in the design of systems that evolve neural networks.

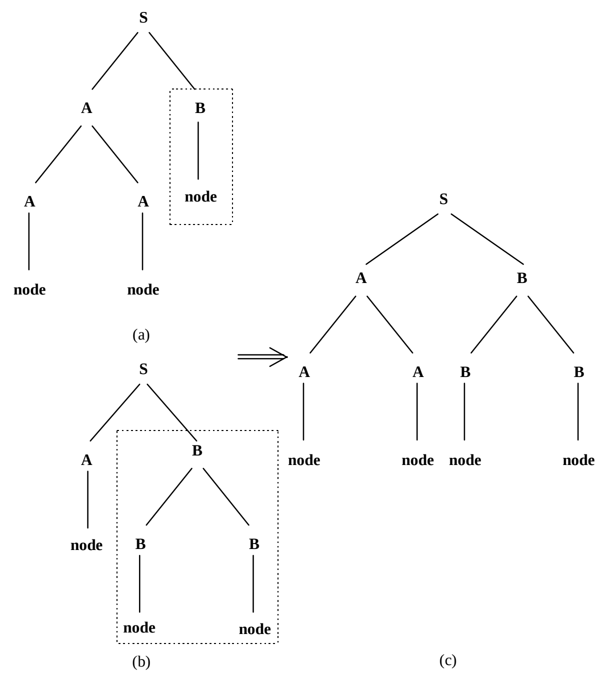


Figure 5: New individual (c) formed using syntactically constrained crossover on parents (a) and (b)

References

Boers, E.J.W., Kuiper, H., Happel, B.L.M & Sprinkhuizen-Kuyper, I.G. (1993) “Designing modular artificial neural networks,” Leiden University Tech Report #93-24.

Gruau, F. (1995) “Automatic definition of modular neural networks,” *Adaptive Behavior, 3*, p. 151-183.

Happel, B.L.M. & Murre, J.M.J (1994) “Design and evolution of modular neural network architectures,” *Neural Networks*, *7*, p.985-1004.

Haynes, T.D., Schoenefeld, D.A. & Wainwright, R.L. (1996) “Type inheritance in strongly typed genetic programming,” Chapter 18 in K.E. Kinnear, Jr. and P.J. Angeline (Eds.), *Advances in Genetic Programming 2*. Cambridge, Mass: MIT Press.

Hussain, T.S. & Browse, R.A. (1998a) “Network generating attribute grammar encoding,” *1998 IEEE International Joint Conference on Neural Networks*, May 4-9, 1998 in Anchorage, Alaska.

Hussain, T.S. & Browse, R.A.(1998b) “Basic properties of attribute grammar encoding,” Workshop at *Third Annual Genetic Programming Conference*, July 21, 1998 in Madison, Wisconsin.

Jacob, C. & Rehder, J. (1993) “Evolution of neural net architectures by a hierarchical grammar-based genetic system,” *ANNGA ’93*, p. 72-79.

Jacobs, R.A., Jordan, M.I., Nowlan, S.J. & Hinton, G.E. (1991) “Adaptive mixtures of local experts,” *Neural Computation*, *3*, p.79-87.

Kitano, H. (1990) “Designing a neural network using genetic algorithm with graph generation system,” *Complex Systems*, *4*, p.461-476.

Knuth, D. E. (1968) “The semantics of context-free languages,” *Mathematical Systems Theory*, *2*, p.127-145.

Koza, J.R. (1994) *Genetic Programming II: Automatic Discovery of Reusable Programs*. Cambridge, Mass: MIT Press.

Yao, X. (1993) “Evolutionary artificial neural networks,” *International Journal of Neural Systems*, *4*, p. 203-222.