Expert Systems With Applications 56 (2016) 368–384

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

Optimization of neural networks through grammatical evolution and a genetic algorithm

Lídio Mauro Lima de Campos[a,*], Roberto Célio Limão de Oliveira[a], Mauro Roisenberg[b]

[a] Faculty of Computer Engineering, Federal University of Pará, Rua Augusto Corrêa 01, Guamá, CEP:66075-100, Belém Caixa postal 479, Pará Brazil
[b] Informatic and Statistic Department, Federal University of Santa Catarina, PO. Box 476, Trindade, Florianópolis 88040-900, Santa Catarina, Brazil

ARTICLE INFO

Article history:
Received 17 August 2015
Revised 18 January 2016
Accepted 8 March 2016
Available online 16 March 2016

Keywords:
Evolutionary computation
Neural networks
Grammatical evolution

ABSTRACT

This paper proposes a hybrid neuro-evolutive algorithm (NEA) that uses a compact indirect encoding scheme (IES) for representing its genotypes (a set of ten production rules of a Lindenmayer System with memory), moreover has the ability to reuse the genotypes and automatically build modular, hierarchical and recurrent neural networks. A genetic algorithm (GA) evolves a Lindenmayer System (L-System) that is used to design the neural network's architecture. This basic neural codification confers scalability and search space reduction in relation to other methods. Furthermore, the system uses a parallel genome scan engine that increases both the implicit parallelism and convergence of the GA. The fitness function of the NEA rewards economical artificial neural networks (ANNs) that are easily implemented. The NEA was tested on five real-world classification datasets and three well-known datasets for time series forecasting (TSF). The results are statistically compared against established state-of-the-art algorithms and various forecasting methods (ADANN, ARIMA, UCM, and Forecast Pro). In most cases, our NEA outperformed the other methods, delivering the most accurate classification and time series forecasting with the least computational effort. These superior results are attributed to the improved effectiveness and efficiency of NEA in the decision-making process. The result is an optimized neural network architecture for solving classification problems and simulating dynamical systems.

1. Introduction

Artificial Neural Networks (ANNs) are applied in classification (Rivero, Dorado, Rabual, & Pazos, 2010), control systems (Li, Jia, Liu, & Ding, 2014), prediction (Donate, Sanchez, & de Miguel, 2012), and many other problems (Czajkowski, Patan, & Szymański, 2014). As the network architecture depends on the class of problem to be solved (classification, time series forecasting, or reinforcement learning), the full potential of an ANN can be exploited by optimizing its architecture and training. The architecture is designed by heuristic trial-and-error methods that select the transfer function and training algorithm for adjusting the synaptic weights. These heuristics affect the learning and generalization capacities of the network and are usually performed by experts, who sequentially train different topologies to determine the best architecture for a specific problem. To mitigate these deficiencies, researchers have proposed automatic procedures for these processes.

Neuro-evolutive algorithms (NEAs) design and/or train ANNs through evolutionary algorithms (EAs). Bio-inspired algorithms (BIOAs) have gained popularity as efficient solvers of nonlinear optimization problems (Krömer, Platoš, & Snášel, 2014). In this paper, we investigate whether a new biologically inspired NEA can provide an efficient automatic design tool for ANNs.

As discussed by (Stanley & Miikkulainen, 2003) , the increasing complexity of evolutionary computation demands more sophisticated methods than direct mapping from genotype to phenotype , (Dasgupta & McGregor, 1992; Donate et al., 2012; Miller, Todd, & Hegde, 1989). An indirect encoding system (IES) allows a more compact and scalable representation than a direct encoding scheme (DES) (Ahmadizar, Soltanian, & AkhlaghianTab, 2015; Hornby & Pollack, 2002; Lee, Seo, & Sim, 2007; Soltanian, Tab, Zar, & Tsoulos, 2013; Stanley, D'Ambrosio, & Gauci, 2009).

Building an efficient encoding scheme for repetitive and recurrent structures is a challenging task for IES-based NEAs. The studies of (Ahmadizar et al., 2015; Dasgupta & McGregor, 1992; Donate et al., 2012; Lee et al., 2007; Miller et al., 1989; Niska, Hiltunen, Karppinen, Ruuskanen, & Kolehmainen, 2004; Sanchez & Melin, 2014; Soltanian et al., 2013; Tsoulos, Gavrilis, & Glavas, 2008) were limited to multilayer perceptron neural

---

* Corresponding author. Tel.: +55 91 3201 7901.
E-mail addresses: lidiomauro@hotmail.com, lidio@ufpa.br (L.M.L. de Campos), limao@ufpa.br (R.C.L. de Oliveira), mauro@inf.ufsc.br (M. Roisenberg).

http://dx.doi.org/10.1016/j.eswa.2016.03.012
0957-4174/

networks (MLPs). However, dynamical systems cannot be precisely estimated by MLPs, and they require recurrent neural networks (RNNs). NEAs evolving RNNs are rarely reported in the literature (Beer & Gallagher, 1992; Hornby & Pollack, 2002). The present paper proposes that the brain manipulates recurrent subnetworks with specialized functions. Such architecture is more similar to biological neural networks than feedforward neural networks and realizes more complex computation.

Our approach is inspired by two natural biological mechanisms: genetic encoding and the evolution of genetic coding. As is well-known, neuron development is governed by the genetic information encoded in deoxyribonucleic acid (DNA), and ultimately generates the final shape of the brain. During biological development, in the complex process that links the DNA code to its phenotype, the same gene is used in many contexts. This compactness minimizes the information required to describe complex individuals. On the other hand, evolution describes the temporal changes in the genetic code (DNA). Among the several mechanisms underlying these evolutionary changes, natural selection is very important.

The two natural processes described above are hybridized such that the DNA contained in cells can also spawn cells. On the other hand, the changes in DNA are passed onto later generations. Motivated by these natural processes, we propose an artificial hybrid system that abstracts these natural mechanisms at an acceptable level of complexity.

To this end, we propose a new NEA, a biologically inspired artificial hybrid system called Artificial Development and Evolution of ANNs (ADEANN). The ADEANN integrates two components. The first is a generative representation that represents genotypes (a set of production rules of a Lindenmayer system) by a compact IES. The IES also conducts and controls the process of mapping the genotypes to the phenotypes (complex neural morphologies). To mimic the DNA encoding scheme and enable scalability, our IES leverages the phenotype representation to a smaller genotype. Thus, the search process is carried out in a lower-dimensional solution space. In addition, our IES implements the organizational principles of hierarchy, modularity, and gene reuse (allowing compact representation of complex phenotypes). The second component is a genetic algorithm (GA), a simplified representation of natural evolution. In local search problems based on GAs, a bit string is called a chromosome (the genotype). Each bit on the chromosome is a gene, and a gene set represents the parameters of a function to be optimized. Each string is assigned a fitness that indicates the quality of its encoded solution (the phenotype). To improve the biological realism of GA, the GA in our approach evolves the generative representation. The evolutionary process can be regarded as the temporal genetic changes in the hypothetical DNAs of a population of individuals, regulated by an artificial selection mechanism. The above biological inspiration underlies the originality of our approach. To our knowledge, we report the first attempt to generate recurrent neural networks from combined metaphors.

The main contribution of our method is the genotype representation by our proposed IES. Using a compact DNA encoding, we codify a parametric Lindenmayer system (L-system) with memory, which implements the principles of organization, modularity, repetition, and hierarchy to achieve complex neural architectures (multilayer and recurrent networks).(Hornby & Pollack, 2002; Lee et al., 2007) adopted L-systems. Although Lee et al. (2007), used DNA encoding, their study was restricted to feedforward neural networks, whereas our approach is extended to recurrent networks. In the IES used by Hornby and Pollack (2002), the genotypes encode twenty rewrite rules of an L-system. Our DNA encoding system encodes a parametric L-system with memory using 10 production rules. Therefore, our IES is more compact than Hornby and Pollack (2002)'s method, and reduces the search space of all feasible solutions. In addition, the memory mechanism in our approach enables the reuse of phenotypic structures (rewrite of nodes and connections) at different stages of development. Such reuse is an important capability of NEAs.

Our ADEANN also utilizes expert knowledge of the problem to more efficiently search the infinite space of topologies, thus minimizing the expert's effort in the optimization. The penalty approach implemented by the fitness function (Eq. 5) automatically rewards the economical ANNs with stronger generalization and extrapolation capacities. Our L-system generates ANN topologies without requiring additional substrate configurations for the given problem. In this sense, our model automates the methodology of (Stanley et al., 2009).

In addition, we investigate whether increasing or decreasing the chromosome length affects the crossover rate, and consequently the number of valid production rules. Our experiments confirm interference between the chromosome length, crossover rate, and production rules. The variable associations are discussed in Section 6.1.

The next section presents state-of-the-art methodologies and the findings of other authors. Section 3 presents the theoretical foundation of our work. The components of ADEANN and system operation are detailed in Section 4. Section 5 describes the materials and methods and Section 6 presents the general experimental design and analyzes the computational results. A general discussion and final conclusions are presented in Sections 7 and 8.

2. State-of-the-art techniques

This section discusses existing research on NEAs. Most NEAs use DESs, which specify every connection and node in the genotype that will appear in the phenotype (ANN) . In the approaches proposed by Dasgupta and McGregor (1992); Miller et al. (1989) a multilayer perceptron is encoded as a directed graph, and its genotype is represented by an adjacency matrix . The ANN architecture is evolved by the ANN itself. In other words, the chromosome encodes all details of the network architecture. The weights of the neural network can then be optimized by gradient-based search approaches. These methods are simply implemented, but the size of the connectivity matrix scales as the square of the number of nodes. Thus, this representation blows up as the number of nodes increases.

Stanley and Miikkulainen (2002) proposed neuro-evolution of augmenting topologies (NEAT). In NEAT, the DES incorporates a few biologically plausible entities, and alters both the weighting parameters and structures of the networks. The flexible genetic encoding of NEAT describes a network by a list of edge genes. Each edge gene denotes an edge between two node genes, and specifies the in-node, out-node, and weight of the edge. The data structure of NEAT, which represents the genotype, grows linearly with the number of edges between two nodes.

Donate et al. (2012) proposed a new approach for automatic ANN design, and applied it to time-series forecasting using a GA. Their method alters both the weighting parameters and network structures. The objective was to improve the accuracy of time-series forecasting. However, they restricted their approach to MLP, which is unsuitable for simulating temporal series forecasting.

Sanchez and Melin (2014) developed a novel multi-objective optimization for a hierarchical genetic algorithm (MOHGA) based on the micro-GA approach. Their method, which optimizes modular neural networks (MNNs), is used in iris recognition. The MOHGA automatically divides the data into granules or submodules, and selects the data for the training and testing phases. It determines the number of granules or submodules and the percentage of the training data to improve the results. The applicability of MOHGA was confirmed in human recognition. The MMNs delivered