Available online at www.sciencedirect.com
ScienceDirect
Pattern Recognition 40 (2007) 80–98
PATTERN RECOGNITION
THE JOURNAL OF THE PATTERN RECOGNITION SOCIETY
www.elsevier.com/locate/patcog

A cooperative constructive method for neural networks for pattern recognition

Nicolás García-Pedrajas*, Domingo Ortiz-Boyer

Department of Computing and Numerical Analysis, University of Córdoba, Spain

Received 9 November 2005; received in revised form 9 June 2006; accepted 26 June 2006

Abstract

In this paper, we propose a new constructive method, based on cooperative coevolution, for designing automatically the structure of a neural network for classification. Our approach is based on a modular construction of the neural network by means of a cooperative evolutionary process. This process benefits from the advantages of coevolutionary computation as well as the advantages of constructive methods. The proposed methodology can be easily extended to work with almost any kind of classifier.

The evaluation of each module that constitutes the network is made using a multiobjective method. So, each new module can be evaluated in a comprehensive way, considering different aspects, such as performance, complexity, or degree of cooperation with the previous modules of the network. In this way, the method has the advantage of considering not only the performance of the networks, but also other features.

The method is tested on 40 classification problems from the UCI machine learning repository with very good performance. The method is thoroughly compared with two other constructive methods, cascade correlation and GMDH networks, and other classification methods, namely, SVM, C4.5, and k nearest-neighbours, and an ensemble of neural networks constructed using four different methods.

Keywords: Constructive algorithms; Pattern classification; Evolutionary computation; Neural networks; Cooperative coevolution

1. Introduction

The automatic design of artificial neural networks has two basic tasks: parametric learning and structural learning. In structural learning, both architecture and parametric information must be learned through the process of training. Basically, we can consider three models of structural learning: constructive algorithms, destructive or pruning algorithms, and evolutionary computation.

Constructive algorithms [1] start with a small network and train this network until it is unable to continue learning, then new components are added to the network. This process is repeated until a satisfactory solution is reached. Destructive methods, also known as pruning algorithms [2], start with a big network, that is able to learn but usually over-fits the training data, and try to remove the connections and nodes that are not useful. A major problem with pruning methods is the assignment of credit to structural components of the network in order to decide whether a connection or node must be removed. Constructive methods have several advantages [1] that are very useful for neural network automatic design: (i) the initial network is easy to specify; (ii) small solutions are searched first and preferred if bigger ones are not able to improve the performance; (iii) with respect to pruning algorithms they are faster as small networks are tested first and do not need to estimate the relevance of nodes or connections. This estimation is time consuming and can only be approximated from a computational point of view.

On the other hand, most constructive algorithms implement a greedy approach which may reach suboptimal solutions in many cases. In this paper, we present an algorithm based on cooperative coevolution that allows the simultaneous evolution of many different modules, avoiding in part the problems of greedy approaches.

* Corresponding author. Tel.: +34 957211032; fax: +34 957218630.
E-mail addresses: npedrajas@uco.es (N. García-Pedrajas), dortiz@uco.es (D. Ortiz-Boyer).

0031-3203/$30.00 © 2006 Pattern Recognition Society. Published by Elsevier Ltd. All rights reserved.
doi:10.1016/j.patcog.2006.06.024

Cooperative coevolution [3] is a recent paradigm in the area of evolutionary computation focused on the evolution of coadapted subcomponents without external interaction. In cooperative coevolution a number of species are evolved together. Cooperation among the individuals is encouraged by rewarding them based on how well they cooperate to solve a target problem. The work on this paradigm has shown that cooperative coevolutionary models present many interesting features, such as specialisation through genetic isolation, generalisation and efficiency [4]. Cooperative coevolution approaches the design of modular systems in a natural way, as the modularity is part of the model.

Modularity is the "integration of functionally related structures and the dissociation of unrelated structures" [5]. Modularity combines parts that must be reusable, interchangeable, and functionally separate [6]. In this paper, we propose a new approach to network design algorithms based on cooperative coevolution that aims at a modular constructive creation of neural networks. Most of the previous models for constructive algorithms are based on the addition of hidden nodes or whole layers to the networks. In this way, the network grows until the stopping criterion is reached. Nevertheless, these procedures of adding new elements to the network limit the number of reachable network architectures.

The proposed approach takes advantage of the cooperative coevolutionary paradigm. At each stage we will have a population of previously constructed modular networks as well as a population of new modules that are able to combine with the previously evolved networks, producing better networks. The two populations coevolve together. This approach has several advantages, the most interesting being:

- the evolutionary approach increases the number of reachable architectures within the search space;
- the cooperative approach allows the solution of the problem in a modular way;
- there is no need to define new objective functions for the training of hidden units. The new modules learn how to cooperate with the previously added modules of the network.

This approach allows a broader search. Instead of adding new modules to a previously fixed network, we evolve a new population of modules that can be combined to the "frozen" populations of modules whose evolution has stagnated. When a population of modules has reached a local optimum it is frozen, and a new population is created. This new population cooperates with the previous ones. This process is continued until the addition of a new population is not able to improve the performance of the system. A second population of networks, each network being a combination of a module from every subpopulation of modules, keeps track of the best combinations of modules so far.

The second important aspect of our work is the use of multiobjective optimisation for the evaluation of the fitness of the new modules that are added to the network during the constructive evolution. In this way, each new module is thoroughly evaluated from different points of view. Moreover, we can add specific objectives if we have some a priori knowledge to bias the structure of the constructed network to a specific set of architectures. The multiobjective evaluation has been successfully applied to the evolution of networks [7] and ensembles of neural networks [8].

This paper is organised as follows: Section 2 reviews some related work; Section 3 describes the cooperative method for constructing neural networks; Section 4 describes the multiobjective evaluation of modules and networks; Section 5 shows the experimental setup and Section 6 shows the results of experiments carried out; Section 7 studies different aspects of the model; and finally Section 8 states the conclusions of our work and future research lines.

2. Related work

The constructive approach is one of the methods of automatically designing the structure of a neural network that has received the most attention in the literature. One of the major reasons for this interest in constructive algorithms is the drawbacks of the other two alternative standard methods: regularisation and pruning techniques. Regularisation methods impose certain conditions on the network training so that unnecessary weights should be driven to 0 during the learning process. However, regularisation techniques are not able to determine the size of the networks. Moreover, the balance between error and penalty terms is very delicate. This balance is usually controlled by a regularisation parameter which is very difficult to adjust. Other approaches incorporate Bayesian methods [9,10], regularisation is then achieved by using appropriate priors [11] that favour small network weights and the regularisation parameter is automatically set. Nevertheless, the relationship between generalisation error and Bayesian evidence is not clear, and some of the hypotheses needed in Bayesian regularisation [1] are not always fulfilled in real problems.

Pruning algorithms suffer form several limitations that prevent their application in real world problems, namely [12]: (i) the size of the initial network may be difficult to determine, (ii) the determination of the relevance of a connection and/or a node remains an open problem, and (iii) the computational cost is excessive due to the necessity of repeated pruning and retraining processes.

Many constructive methods have been proposed in the literature since the pioneer model of cascade correlation [13]. Most of these methods are reviewed in Refs. [1,14]. Over the last few years there have been several papers focused on alternative approaches to constructive methods. These methods try to improve the flexibility of exploring the space of neural network topologies [15], the incremental feature of the learning process [16], or to propose new training strategies [12]. Potter [17] developed a genetic cascade-correlation algorithm where a genetic algorithm