Artificial Intelligence in Medicine 59 (2013) 157–167

Contents lists available at ScienceDirect

Artificial Intelligence in Medicine

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

Phased searching with NEAT in a Time-Scaled Framework: Experiments on a computer-aided detection system for lung nodules

Maxine Tan[a,b], Rudi Deklerck[a,b,*], Jan Cornelis[a], Bart Jansen[a,b]

[a] Department of Electronics and Informatics, Vrije Universiteit Brussel, Pleinlaan 2, B-1050 Brussel, Belgium
[b] Department of Future Media and Imaging, iMinds, Gaston Crommenlaan 8, 9050 Gent, Belgium

ARTICLE INFO

Article history:
Received 6 September 2012
Received in revised form 16 May 2013
Accepted 31 July 2013

Keywords:
Classifiers
Feature selection
Evolutionary computation
Artificial neural networks
Medical image analysis
Lung nodule detection

ABSTRACT

Objective: In the field of computer-aided detection (CAD) systems for lung nodules in computed tomography (CT) scans, many image features are presented and many artificial neural network (ANN) classifiers with various structural topologies are analyzed; frequently, the classifier topologies are selected by trial-and-error experiments. To avoid these trial and error approaches, we present a novel classifier that evolves ANNs using genetic algorithms, called "Phased Searching with NEAT in a Time or Generation-Scaled Framework", integrating feature selection with the classification task.
Methods and materials: We analyzed our method's performance on 360 CT scans from the public Lung Image Database Consortium database. We compare our method's performance with other more-established classifiers, namely regular NEAT, Feature-Deselective NEAT (FD-NEAT), fixed-topology ANNs, and support vector machines (SVMs) using ten-fold cross-validation experiments of all 360 scans.
Results: The results show that the proposed "Phased Searching" method performs better and faster than regular NEAT, better than FD-NEAT, and achieves sensitivities at 3 and 4 false positives (FP) per scan that are comparable with the fixed-topology ANN and SVM classifiers, but with fewer input features. It achieves a detection sensitivity of 83.0 ± 9.7% with an average of 4 FP/scan, for nodules with a diameter greater than or equal to 3 mm. It also evolves networks with shorter evolution times and with lower complexities than regular NEAT ($p = 0.026$ and $p < 0.001$, respectively). Analysis on the average and best network complexities evolved by regular NEAT and by our approach shows that our approach searches for good solutions in lower dimensional search spaces, and evolves networks without superfluous structure.
Conclusions: We have presented a novel approach that combines feature selection with the evolution of ANN topology and weights. Compared with the original threshold-based Phased Searching method of Green, our method requires fewer parameters and converges to the optimal network complexity required for the classification task at hand. The results of the ten-fold cross-validation experiments also show that our proposed CAD system for lung nodule detection performs well with respect to other methods in the literature.

1. Introduction – the context

Feature selection is a task of tremendous importance. The challenge is to select features that are important or *relevant* to the classification task. The exclusion of important features leads to sub-optimal results whereas the inclusion of *irrelevant* or *redundant* features adds unnecessary dimensions to the search space. The inclusion of every additional feature causes an exponential number of possible states in the search space. Various feature selection algorithms are proposed [1–6] and comparative studies are reported [7–10].

The challenges of designing a good classifier are not just related to the selection of relevant features; in designing an artificial neural network (ANN) classifier, frequently its topology is determined by trial-and-error experiments [11,12]. The addition of every connection imposes an extra weight-tuning requirement. Neuroevolution, the artificial evolution of ANNs using genetic algorithms (GAs), has shown great promise in solving complex learning tasks [13–18]. In early work by Harvey [19] and Cliff et al. [20], evolution to grow networks using an extended form of GAs known as SAGA is described. In work by Schlessinger et al. [21], the evolvability of ANNs within an artificial life simulation

---

* Corresponding author at: Department of Electronics and Informatics, Vrije Universiteit Brussel, Pleinlaan 2, B-1050 Brussel, Belgium. Tel.: +32 2 629 1682; fax: +32 2 629 2883.
E-mail address: rdeklerc@etro.vub.ac.be (R. Deklerck).

0933-3657/$ – see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.artmed.2013.07.002

is investigated. Specifically, the efficacies of five different types of structural mutations that incorporate different general principles thought to be important for network evolvability, are analyzed.

The evaluation of feature selection and classifier performance only yields meaningful results when applied on real life problems involving large data sets and complex tasks. Therefore we selected such a case in medical imaging, on which we have previous experience, namely lung nodule detection from computed tomography (CT) scans. In the most recent reports of the American Cancer Society for the United States, in 2012, it is estimated that lung cancer will account for about 28% of all cancer deaths [22]. Each year, more people die of lung cancer than of breast, colon and prostate cancer combined [22]. To date, many features are presented for lung nodule detection [11,23–27]. Most systems present feature selection and classification as two separate stages in the computer-aided detection (CAD) system [23,24].

Previously, in [27], we proposed a CAD system for lung nodule detection using a feature-deselective neuroevolution classifier called Feature-Deselective NEAT (FD-NEAT), which is a system that combines the feature selection process with the learning task during the training of the classifier. Now, we describe a novel method “Phased Searching with NEAT in a Time-Scaled Framework” that is based on the NeuroEvolution of Augmenting Topologies (NEAT) algorithm proposed by Stanley and Miikkulainen [28]. We describe the better properties of our method compared to the original threshold-based Phased Searching [29], namely fewer parameters have to be specified and better convergence to the optimal network complexity is obtained for the classification task at hand. We evaluate this method on the previously published lung CAD system described in [27,30], and will show that it outperforms regular NEAT and FD-NEAT, and achieves sensitivities at 3 and 4 false positives (FP) per scan that are comparable with the fixed-topology ANN and support vector machine (SVM) classifiers, but with fewer input features.

2. Literature review

The *NEAT algorithm* [28] is a neuroevolution method that differs from other neuroevolution approaches in three distinct ways, namely (1) it employs a method of crossover of different topologies using innovation numbers as historical markings; (2) it protects structural innovation using speciation; and (3) it incrementally grows from almost minimal structure. NEAT is designed to take advantage of structure as a way of minimizing the dimensionality of the search space of connection weights [28], i.e. NEAT starts almost minimally with the aim of evolving networks with *minimal structure.* NEAT's performance has been analyzed on various challenging tasks including a race car simulator domain (RARS) [31], a traditional board game (Go) [32], training a team of virtual robots in a modern video game (NERO) [33], and double pole balancing experiments [28].

Recently, Hypercube-based NEAT or *HyperNEAT*, which evolves large-scale ANNs with the principles of NEAT whilst utilizing the geometric regularities of the task domain at hand was proposed in [34–36].

Other methods in the literature currently exist that start with a randomly-selected feature set [37], and with all features connected/selected [38]. However, very few methods have been proposed that combine feature selection with neuroevolution. One of these algorithms is a feature-deselective version of NEAT called *FD-NEAT* [39]. FD-NEAT starts with the same topology as regular NEAT (with all inputs connected to all the outputs), but includes a mutation operator that enables discarding irrelevant or redundant inputs in the initial population of networks.

A feature-selective version of NEAT, *FS-NEAT* proposed by Whiteson et al. [40,41] starts with only one randomly-selected input connected to a randomly-selected output in the initial population of networks. The only difference between FS-NEAT and regular NEAT is in generating the initial population [40,41]. After the initial population is generated, FS-NEAT works exactly like regular NEAT, namely hidden nodes and links are added through mutation, and the additions that aid performance are likely to survive. Hence, feature selection is performed implicitly, with links emerging from useful inputs tending to survive.

In a recent paper by Loscalzo et al. [42], an embedded feature selection algorithm for reinforcement learning called Sample Aware Feature Selection-NEAT (*SAFS-NEAT*) is proposed. The method adopts sequential forward search using mutual information as an evaluation measure within the NEAT learning algorithm. In a Robot Auto Racing Simulator (RARS) racing simulation environment [43], SAFS-NEAT was shown to outperform FS-NEAT and regular NEAT in cases with many irrelevant inputs.

Further methods in the literature based on NEAT that combine feature selection with the learning task include *Blended Searching* by James and Tucker [44], whereby add and delete connection mutations are simultaneously enabled to evolve networks that find a balance between *complexification* and *simplification*.

In Yao's overview of *topological and weight evolving artificial neural networks* [45], algorithms for finding optimal network architectures fall into two broad categories, namely *constructive* and *destructive*. These dynamics can also be referred to as *complexification* and *simplification*, respectively. A *complexification* algorithm starts with minimal architecture, and incrementally adds new structure (nodes, connections and layers) when necessary during training. On the other hand, a *simplification* algorithm starts with an overabundance of topological structure and deletes unnecessary architecture (layers, nodes and connections) during training.

Green proposed an approach in an experimental write-up/technical report [29] called *Phased Searching* whereby the NEAT search performs better if it switches between pure complexifying and pure simplifying phases. It is put forward by Green [29] that this technique better preserves NEAT's ability to search complex topologies, and also allows the search to periodically remove or prune away redundant structures. The pruning process allows NEAT to explore a wider range of simple topologies, and also speeds up the search process by lowering the overall complexity of the genome population. Green also proposed that a specified pruning phase threshold value, depending on the type of experiment and domain, be specified to initiate or initialize the simplifying/pruning phase. In a basic optical character recognition (OCR) experiment, it was shown that Phased Searching outperforms pure complexification (regular NEAT) and Blended Searching, evolving networks with much lower complexities [29].

Phased Searching in itself yields several advantages: (1) removal of redundant structure; (2) simpler networks that are faster to evolve; (3) lowered species numbers or variations due to reduced network complexity; and (4) automated feature selection/deselection combined with the learning task. Our method, a variant of the Phased Searching method, has some extra advantages that will be described in Section 3.1.

2.1. Phased Searching removes redundant structure

It is shown in [29,39,44] that purely complexifying runs in NEAT (i.e. regular NEAT) evolve redundant structure. What constitutes redundant structure? This can easily be understood from an analogy with a real-world problem of a robot that moves around using sensors to avoid hitting objects in its path. Sensors that are not relevant to distance measurement, such as temperature sensors are *irrelevant* inputs. The minimal number of sensors it