An ANN's Evolved by a New Evolutionary System and Its Application

Chunkai Zhang and Huihe Shao

Automation Department, Shanghai Jiao Tong University
Shanghai, P. R. China, 200030
cgzhang812@sohu.com

Abstract

The paper presents a new evolutionary system for evolving artificial neural networks (ANN's). In the process of evolution, the network architecture and the node weights of ANN's are evolved alternately, and the evolution value of network architecture is related to the error value of ANN's evolved by node weights. An evolved ANN's has been used in modelling product quality estimator for a fractionator of the hydrocracking unit in the oil refining industry, the results show that it has good accuracy and generalisation ability.

1 Introduction

Design of a near optimal ANN's architecture can be formulated as a search problem in the architecture space where each point represents a type of architecture. Miller et al. indicated that the evolutionary algorithms are better candidates for searching architecture space than those constructive and pruning algorithms [1, 2]. GA was used to evolve ANN's, but the evolution of ANN's often suffers from the permutation problem [3]. This problem not only makes the evolution inefficient, but also makes crossover operators more difficult to produce highly fit offspring.

This paper proposes a new evolutionary system for evolving feedforward ANN's called PSONN, which combines the architectural evolution with weight learning. PSONN is based on the improved particle swarm optimisation (PSO) algorithm [4]. And in PSONN a strategy of evolving added nodes (EAN) and a partial training algorithm (PT) are used to maintain behavioural link between a parent and its offspring to prevent destruction of the behaviour already learned by the parent.

2 PSONN system

2.1 Particle swarm optimisation (PSO)

In PSONN system, the learning algorithm is the PSO algorithm. PSO is a population based optimisation algorithm that is motivated from the simulation of social behaviour. In PSO each individual flies in the search space with a velocity that is dynamically adjusted according to its own flying experience and its companions' flying experience. Compared with other evolutionary algorithms, such as GA, PSO algorithm possesses some attractive properties such as memory and constructive cooperation between individuals, so it has more chance to "fly" into the better solution areas more quickly and discover reasonable quality solution much faster [4]. Because there is not a selection operator in PSO, each individual in an original population has a corresponding partner in a new population. From the view of the diversity of population, this property is better than GA, so it can avoid the premature convergence and stagnation in GA to some extent.

2.2 PSONN system

In PSONN, evolving ANN's architectures and weight learning are alternated. This process can avoid a moving target problem resulted from the simultaneous evolution of both architectures and weights [5]. And the network architectures are adaptively evolved by PSO algorithm, starting from the parent's weights instead of randomly initialised weights, so this can preferably solve the problem of the noisy fitness evaluation that can mislead the evolution. In PSONN all the data sets are partitioned into three sets: a training set, a validation set, and a testing set. The training set is used to evolve the nodes of ANN's with a given network architecture, and the fitness evaluation is equal to the mean squared error E of ANN's. But in evolving the architecture of network, the fitness evaluation is different from previous work in evolving the nodes of ANN's since it is determined through a validation set which does not overlap with the train set. Such use of validation set in evolutionary learning system improves the generalisation ability of evolved ANN's and introduces little overhead in computation time.

The major steps of PSONN can be described as follows:

1) Generate an initial population of M networks. The number of hidden nodes for each network is uniformly generated at random within certain ranges, and the initial weights are uniformly distributed inside a small range.

2) Use the partially training (PT) algorithm to train each network in the population on the training set, which is as follows:

- Choose a network as a parent network, then randomly generate N-1 initial individuals as a population where each individual's initial weights uniformly generated at random within certain ranges, but their network architectures are the same as the parent network architecture. And then the parent network is added into the population. Here each individual in this population is to parameterise a whole group of g nodes in ANN's, this means that every component of each individual represents a connection weight.

- Employ the PSO algorithm to evolve this population until the best individual found is accepted or the maximum number of generations has been reached.
- The best individual that survived will join the network architecture evolution.

3) All survived networks form a new population. Evaluate the fitness values of every individual in this population. Here the mean squared error value *E* of each network on the validation set represents the fitness evaluation of each individual.

4) If the best network found is accepted or the maximum number of generations has been reached, stop and go to step 7). Otherwise continue.

5) Employ the PSO algorithm to only evolve the network architecture of each individual. Here each individual represents the number of the hidden nodes of a network.

6) When the network architecture of an individual changes, employ the strategy of evolving added nodes (EAN) to decide how to evolve its connection weights with the PT algorithm to. There are two choice:

- If some hidden nodes need to be added to this network, under the strategy of EAN, the PT algorithm only evolves the new added nodes to explain as much of the remaining output variance as possible. In this case the cost function that is minimised at each step of algorithm is the residual sum squared error that will remain after the addition of the new nodes, and the existing network nodes of the hidden layer are left unchanged during the search for the best new added nodes. Compared with the existing hidden nodes, the added nodes will represent or explain the finer details of this mapping that the entire network is trying to approximate between the inputs and outputs of the training data. This strategy can decrease the computation time for evolving the entire network and prevent destruction of the behaviour already learned by the parent.
- If some hidden nodes need to be deleted from a network, EAN strategy can remove the nodes in the reverse order in which they were originally added to the network, then the PT algorithm evolves the connection weights of the entire network, but sometimes a few of jump in fitness from the parent to the offspring is not avoided.

Then go to Step 3).

7) After the evolutionary process, train the best ANN’s further with the PT algorithm on the combined training and validation set until it “converges”.

In step 7), the generalisation ability of ANN’s can be further improved by training the best ANN’s with the PT algorithm on the combined training and validation set.

After evolving the architecture of networks every time, the strategy of evolving added nodes (EAN) and the partial training (PT) algorithm are used to optimise the connection weights of nodes with a given network architecture which has been evolved by the PSO algorithm. In other words, the purpose of this process is to evaluate the quality of this given network architecture and maintain the behavioural link between a parent and its offspring [6].

3 Experimental studies

In order to evaluate the ability of PSONN in evolving ANN’s, it was applied to product quality estimation where jet fuel endpoint variable is related to the thirteen variables. After 150 epoch off-line learning, the maximum absolute mean error averaged 30 runs for jet fuel endpoint in the tested 60 date sets is 1.1℃, and the root mean square error 0.27. From the simulation results, it is illustrated that the proposed neural network system completely satisfies the requirement of jet fuel endpoint measurement and possesses good generalisation ability.

4 Conclusion

PSONN is based on PSO algorithm, which is used to evolve both the architecture and weights of ANN's, this means that the network architecture is adaptively adjusted by PSO algorithm, and then PSO algorithm is used to evolve the nodes of ANN’s with a given architecture. This process is repeated until the best network is accepted or the maximum number of generations has been reached. It can effectively alleviate the noisy fitness evaluation problem and the moving target problem.

References

[1] R. K. Belew, J. MchInerney and N. N. Schraudolph. Evolving networks: Using GAs with connectionist learning. Computer Sci. Eng. Dept., Univ. California-San Diego, Tech. Rep. CS90-174 revised, Feb. 1991.

[2] N.Burgess. A constructive algorithm that converges for real-valued input patterns. Int. J. Neural Syst., vol 5, no. 1, pp. 59-66, 1994.

[3] R. K. Belew, J. MchInerney and N. N. Schraudolph. Evolving networks: Using GAs with connectionist learning. Computer Sci. Eng. Dept., Univ. California-San Diego, Tech. Rep. CS90-174 revised, Feb. 1991.

[4] Kennedy, J., and Eberhart, R. C. Particle swarm optimization. Proc. IEEE International Conference on Neural Networks, IEEE Service Center, Piscataway, NJ, pp. 39-43, 1995.

[5] X. Yao. A review of evolutionary artificial neural networks. Int. J. Intell. Syst., vol. 8, no. 4, pp. 539-567, 1993.

[6] J. R. McDonnell and D. Waagen. Evolving recurrent perceptrons for time-series modeling. IEEE Tran. Neural Networks, vol. 5, no. 1, pp. 24-38, 1994.

This work was supported by National 973 Fundamental Research Program of China.