Applied Mathematics and Computation 205 (2008) 792–798

Contents lists available at ScienceDirect

Applied Mathematics and Computation

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

A new approach for function approximation incorporating adaptive particle swarm optimization and *a priori* information

Fei Han [a,b,*], Qing-Hua Ling [c]

[a] *School of Computer Science and Telecommunication Engineering, Jiangsu University, Zhenjiang, Jiangsu 212013, China*
[b] *Intelligent Computing Lab, Hefei Institute of Intelligent Machines, Chinese Academy of Sciences, P.O. Box 1130, Hefei Anhui 230031, China*
[c] *School of Electronics and Information, Jiangsu University of Science and Technology, Zhenjiang, Jiangsu 212003, China*

ARTICLE INFO

Keywords:
Feedforward neural networks
Function approximation
Particle swarm optimization
A priori information
Gradient-based learning algorithms

ABSTRACT

In this paper, a new approach coupling adaptive particle swarm optimization (APSO) and *a priori* information for function approximation problem is proposed to obtain better generalization performance and faster convergence rate. It is well known that gradient-based learning algorithms such as backpropagation (BP) algorithm have good ability of local search, whereas PSO has good ability of global search. Therefore, in the new approach, first, APSO encoding the first-order derivative information of the approximated function is applied to train network to near global minima. Second, with the connection weights produced by APSO, the network is trained with a gradient-based algorithm. Due to combining APSO with local search algorithm and considering *a priori* information, the new approach has better generalization performance and convergence rate than traditional learning ones. Finally, simulation results are given to verify the efficiency and effectiveness of the proposed approach.

1. Introduction

Most traditional supervised learning algorithms for feedforward neural networks (FNN) use the sum-of-square error criterion to derive the updated formulae [1]. These supervised learning algorithms have the following major drawbacks. First, they are apt to be trapped in local minima. Second, they have not considered the network structure features as well as the involved problem properties, thus their generalization capabilities are limited. Finally, since BP algorithms are the gradient-based type learning algorithms, they converge very slowly [2–6]. Although the network with too many synaptic weights can solve exactly the involved problems, it may suffer from overfitting, thus result in poor generalization performance. On the other hand, while a smaller network can not solve the problem accurately, it can obtain a better generalization capability [7,8]. So, in practical applications, a good compromise should be made between the generalization capability and the network complexity.

In literatures [9,10], two algorithms were proposed that are referred to as Hybrid-I and Hybrid-II methods, respectively. The Hybrid-I algorithm incorporated the first-order derivatives of the neural activation at hidden layers into the sum-of-square error cost function to reduce the input-to-output mapping sensitivity. On the other hand, the Hybrid-II algorithm incorporated the second-order derivatives of the neural activations at all layers into the sum-of-square error cost function to penalize the high frequency components in the training data. In literature [11], a modified hybrid algorithm (MHLA) was proposed according to Hybrid-I and Hybrid-II algorithms to improve the generalization performance.

* Corresponding author. Address: School of Computer Science and Telecommunication Engineering, Jiangsu University, Zhenjiang, Jiangsu 212013, China.
E-mail address: hanfei1976@163.com (F. Han).

0096-3003/$ - see front matter © 2008 Elsevier Inc. All rights reserved.
doi:10.1016/j.amc.2008.05.025

Obviously, gradient-based learning algorithm has good capability of local search. On the other hand, particle swarm optimization algorithm has good capability of global search. Therefore, global search can be combined with local search in a learning algorithm to improve the convergence performance. In recent years, particle swarm optimization has been used increasingly as an effective technique for search global minima [12,13]. Compared with genetic algorithm (GA) [14], PSO has some attractive characteristics. Firstly, PSO has memory, that is, the knowledge of good solutions is retained by all particles, whereas in GA, previous knowledge of the problem is destroyed once the population changes. Secondly, PSO has constructive cooperation between particles, that is, particles in the swarm share their information [15]. Finally, PSO is rapidly converging towards an optimum, simple to compute, easy to implement and free from the complex computation in genetic algorithm (e.g., coding/decoding, crossover and mutation) [16].

In this paper, a new approach for function approximation based on APSO and *a priori* information is proposed. In order to overcome the drawbacks from gradient-based algorithm for FNN, the single-hidden layered feedforward neural networks (SLFN) is trained by APSO first to near the global minima, and then the network is trained again by a local search algorithm – MHLA. Moreover, the first-order derivative information of the approximated function is incorporated in APSO in the new approach. Due to combining APSO with local search algorithm and considering the *a priori* information, the new approach has better generalization performance and convergence rate. Finally, simulation results are given to verify the efficiency and effectiveness of the proposed learning approach.

In addition, what should be stressed is that since a neural network with single nonlinear hidden layer is capable of forming an arbitrarily close approximation of any continuous nonlinear mapping [17–19], our discussion will be limited to such networks.

2. Particle swarm optimization

The particle swarm optimization (PSO) is an evolutionary computation technique developed by Dr. Eberhart and Dr. Kennedy in 1995 [20,21], inspired by social behavior of bird flocking. PSOA is an optimization tool based on population, and the system is initialized with a population of random solutions and can search for optima by updating generations.

PSO is a kind of algorithm to search for the best solution by simulating the movement and flocking of birds. The algorithm works by initializing a flock of birds randomly over the searching space, where every bird is called as a "particle". These "particles" fly with a certain velocity and find the global best position after some iteration. At each iteration, each particle can adjust its velocity vector, based on its momentum and the influence of its best position (P_b) as well as the best position of its neighbors (P_g), and then compute a new position that the "particle" is to fly to. Supposing the dimension of searching space is D, the total number of particles is n, the position of the ith particle can be expressed as vector $X_i = (x_{i1}, x_{i2}, \ldots, x_{iD})$; the best position of the ith particle searching until now is denoted as $P_{ib} = (p_{i1}, p_{i2}, \ldots, p_{iD})$, and the best position of the total particle swarm searching until now is denoted as vector $P_g = (p_{g1}, p_{g2}, \ldots, p_{gD})$; the velocity of the ith particle is represented as vector $V_i = (v_{i1}, v_{i2}, \ldots, v_{iD})$. Then the original PSOA [20,21] is described as:

$$v_{id}(t+1) = v_{id}(t) + C_1 * \ \mathrm{rand}() * [p_{id}(t) - x_{id}(t)] + C_2 * \ \mathrm{rand}() * [p_{gd}(t) - x_{id}(t)], \tag{1}$$

$$x_{id}(t+1) = x_{id}(t) + v_{id}(t+1) \quad 1 \leqslant i \leqslant n \ 1 \leqslant d \leqslant D, \tag{2}$$

where c_1, c_2 are the acceleration constants with positive values; rand() is a random number between 0 and 1; w is the inertia weight. In addition to the c_1, and c_2 parameters, the implementation of the original algorithm also requires to place a limit on the velocity ($v_{\max}$). After adjusting the parameters w and $v_{\max}$, the PSO can achieve the best search ability.

The adaptive particle swarm optimization (APSO) algorithm is based on the original PSO algorithm, firstly proposed by Shi & Eberhart in 1998 [22,23]. The APSO can be described as follows:

$$v_{id}(t+1) = w * v_{id}(t) + C_1 * \ \mathrm{rand}() * [p_{id}(t) - x_{id}(t)] + C_2 * \ \mathrm{rand}() * [p_{gd}(t) - x_{id}(t)], \tag{3}$$

$$x_{id}(t+1) = x_{id}(t) + v_{id}(t+1) \quad 1 \leqslant i \leqslant n \ 1 \leqslant d \leqslant D, \tag{4}$$

where w is a new inertial weight. This algorithm by adjusting the parameter w can make w reduce gradually as the generation increases. In the searching process of the PSO algorithm, the searching space will reduce gradually as the generation increases. So the APSO algorithm is more effective, because the searching space reduces step by step, not linearly, so the parameter w here also reduces step by step.

3. APSO encoding *a priori* information from the approximated function

Before presenting new modified cost function, we first make the following mathematical notation. Assume that x_k and y_i denote the k th element of the input vector and the i th element of the output vector, respectively; $w_{j_l j_{l-1}}$ denotes the synaptic weight from the j_l th neuron at the l th layer to the j_{l-1} th neuron at $(l-1)$ th layer; $h_{j_l} = f_l(\hat{h}_{j_l})$ is the activation function of the j_l th element at the l th layer with $\hat{h}_{j_l} = \sum_{j_{l-1}} w_{j_l j_{l-1}} h_{j_{l-1}}$. The t_i and y_i denote the target and actual output values of the i th neuron at output layer, respectively.

In the course of approximating a function, the FNN can approximate it more accurately when *a priori* information containing the function properties is encoded into the network. In this paper, *a priori* information containing the first-order derivatives of the approximated function is considered.