Available online at www.sciencedirect.com
ScienceDirect
Applied Mathematics and Computation 185 (2007) 1026–1037
APPLIED MATHEMATICS AND COMPUTATION
www.elsevier.com/locate/amc

A hybrid particle swarm optimization–back-propagation algorithm for feedforward neural network training

Jing-Ru Zhang [a,b,*], Jun Zhang [a], Tat-Ming Lok [c], Michael R. Lyu [d]

[a] Intelligent Computing Lab, Hefei Institute of Intelligent Machines, Chinese Academy of Sciences, P.O. Box 1130, Hefei, Anhui 230031, China
[b] Department of Automation, University of Science and Technology of China, Hefei 230027, China
[c] Information Engineering Department, The Chinese University of Hong Kong, Shatin, Hong kong
[d] Computer Science and Engineering Department, The Chinese University of Hong Kong, Shatin, Hong kong

Abstract

The particle swarm optimization algorithm was showed to converge rapidly during the initial stages of a global search, but around global optimum, the search process will become very slow. On the contrary, the gradient descending method can achieve faster convergent speed around global optimum, and at the same time, the convergent accuracy can be higher. So in this paper, a hybrid algorithm combining particle swarm optimization (PSO) algorithm with back-propagation (BP) algorithm, also referred to as PSO–BP algorithm, is proposed to train the weights of feedforward neural network (FNN), the hybrid algorithm can make use of not only strong global searching ability of the PSOA, but also strong local searching ability of the BP algorithm. In this paper, a novel selection strategy of the inertial weight is introduced to the PSO algorithm. In the proposed PSO–BP algorithm, we adopt a heuristic way to give a transition from particle swarm search to gradient descending search. In this paper, we also give three kind of encoding strategy of particles, and give the different problem area in which every encoding strategy is used. The experimental results show that the proposed hybrid PSO–BP algorithm is better than the Adaptive Particle swarm optimization algorithm (APSOA) and BP algorithm in convergent speed and convergent accuracy.

Keywords: Particle swarm optimization algorithm; Feedforward neural network; BP algorithm; Adaptive particle swarm optimization

1. Introduction

In recent years, feedforward neural networks (FNN), in particular, two layered FNNs [12] have been widely used to classify nonlinearly separable patterns [31,26,22] and approximate arbitrary continuous functions

* Corresponding author. Address: Intelligent Computing Lab, Hefei Institute of Intelligent Machines, Chinese Academy of Sciences, P.O. Box 1130, Hefei, Anhui 230031, China.
E-mail addresses: jrzhang@iim.ac.cn (J.-R. Zhang), zhangjun@iim.ac.cn (J. Zhang), tmlok@ie.cuhk.edu.hk (T.-M. Lok), lyu@cse.cuhk.edu.hk (M.R. Lyu).

0096-3003/$ - see front matter
doi:10.1016/j.amc.2006.07.025

[20,25]. Currently, there have been many algorithms used to train the FNN, such as back-propagation algorithm (BPA), genetic algorithm (GA) [5,6], simulating annealing algorithm (SAA) [13,14], particle swarm optimization algorithm (PSO) [16,18], and so on.

The particle swarm optimization (PSO) is an evolutionary computation technique developed by Eberhart and Kennedy in 1995 [1,2], inspired by social behavior of bird flocking. And we can also say it to be a kind of algorithm based on social psychology. Similar to the genetic algorithm (GA), the PSO algorithm is an optimization tool based on population, and the system is initialized with a population of random solutions and can search for optima by the updating of generations. In 1998, Shi and Eberhart firstly introduced the inertia weights w into the previous PSO algorithm [3,17]. Through adjusting w, the performances of the PSO algorithm can be improved significantly. Researchers often use a kind of Adaptive particle swarm optimization (APSO) algorithm. The adaptive particle swarm optimization can be described as following, in different searching stages, the inertial weight w is changed adaptively.

Unlike the GA, the PSO algorithm has no complicated evolutionary operators such as crossover and mutation [21]. In the PSO algorithm, the potential solutions, called as particles, are obtained by "flowing" through the problem space by following the current optimum particles. Generally speaking, the PSO algorithm has a strong ability to find the most optimistic result, but it has a disadvantage of easily getting into a local optimum .After suitably modulating the parameters for the PSO algorithm, the rate of convergence can be speeded up and the ability to find the global optimistic result can be enhanced. The PSO algorithm's search is based on the orientation by tracing P_b that is each particle's best position in its history, and tracing P_g that is all particles' best position in their history, it can rapidly arrive around the global optimum. However, because the PSO algorithm has several parameters to be adjusted by empirical approach, if these parameters are not appropriately set, the search will become very slow near the global optimum.

Regarding the FNNs training, the mostly used training algorithm is the back-propagation (BP) algorithm, which is a gradient-based method. Hence some inherent problems existing in BP algorithm are also frequently encountered in the use of this algorithm. Firstly, the BP algorithm will easily get trapped in local minima especially for those non-linearly separable pattern classification problems or complex function approximation problem [7], so that back-propagation may lead to failure in finding a global optimal solution. Second, the convergent speed of the BP algorithm is too slow even if the learning goal, a given termination error, can be achieved. The important problem to be stressed is that the convergent behavior of the BP algorithm depends very much on the choices of initial values of the network connection weights as well as the parameters in the algorithm such as the learning rate and the momentum. To improve the performance of the original BP algorithm, researchers have concentrated on the following two factors: (1) selection of better energy function [8,9]; (2) selection of dynamic learning rate and momentum [10,11]. However, these improvements haven't removed the disadvantages of the BP algorithm getting trapped into local optima in essence. In particular, with FNN's structure becoming more complex, its convergent speed will be even slower. But if the search for the BP algorithm starts from near the optimum, and if the learning rate is adjusted small enough, how will the searching results be? The experiments in the sequel will give further analyses.

Genetic algorithm (GA) has been also used in training FNNs recently, but in training process, this algorithm needs encoding operator and decoding operator. Usually there are three kinds of complicated evolutionary operators with this algorithm, i.e., selection, crossover and mutation, it was found in experiments that when the structure of FNNs is simple, its training results may be better than the ones using the BP algorithm to train, when the structure of FNNs becomes complex and there are large training samples, the GA's convergent speed will become very slow, so that the convergent accuracy may be influenced by the slow convergent speed.

In this paper, we combined the adaptive particle swarm optimization (APSO) algorithm with the BP algorithm to form a hybrid learning algorithm for training FNNs. This hybrid uses the APSO algorithm to do global search in the beginning of stage, and then uses the BP algorithm to do local search around the global optimum P_g. In particular, this hybrid algorithm will be used to train the FNN weights for function approximation and classification problems, respectively compared with the APSO algorithm and the BP algorithm in convergent speed and generalization performance.