Math. Model. Nat. Phenom.
Vol. 5, No. 7, 2010, pp. 132-138
DOI: 10.1051/mmnp/20105722

Hybrid Particle Swarm and Neural Network Approach for Streamflow Forecasting

A. Sedki* **and D. Ouazar**

Department of Civil Engineering, Mohammadia School of Engineering
University Mohammed V-Agdal, Rabat, Morocco

Abstract. In this paper, an artificial neural network (ANN) based on hybrid algorithm combining particle swarm optimization (PSO) with back-propagation (BP) is proposed to forecast the daily streamflows in a catchment located in a semi-arid region in Morocco. The PSO algorithm has a rapid convergence during the initial stages of a global search, while the BP algorithm can achieve faster convergent speed around the global optimum. By combining the PSO with the BP, the hybrid algorithm referred to as BP-PSO algorithm is presented in this paper. To evaluate the performance of the hybrid algorithm, BP neural network is also involved for a comparison purposes. The results show that the neural network model evolved by PSO-BP algorithm has a good predictions and better convergence performances.

Key words: artificial neural network, particle swarm optimization algorithm, back propagation, daily streamflows, catchment, semi-arid climate
AMS subject classification: 78M32

1. Introduction

Forecasting of streamflows are vital and important for flood caution, operation of flood-control reservoir, determination of river water potential, production of hydroelectric energy, allocation of domestic and irrigation water in drought seasons, and navigation planning in rivers. Two basic approaches are usually used to deal with; conceptual modelling and black box modelling. Conceptual models attempt to mathematically simulate the sub-processes and physical mechanisms that

*Corresponding author. E-mail: asedki@emi.ac.ma

govern the hydrological cycle. These models are generally non-linear, time-invariant, and deterministic, involving parameters that represent watershed characteristics [3]. Thus, it is often too complex, too demanding in terms of data and parameters requirements and requiring some degree of expertise and experience with the model [1]. In a black box approach a model is applied to identify a direct mapping between inputs and outputs without detailed consideration about the internal structure of the physical processes. One of the black box techniques, which has caught attention of hydrologists in recent past is artificial neural networks (ANN).

The most widely used neural network model is the multilayer percepton (MLP), in which the connection weight training is normally completed by a back propagation (BP) learning algorithm [6]. The idea of weight training in MLPs is usually formulated as minimization of an error function, such as mean square error (MSE) between the network output and the target output over all examples, by iteratively adjusting connection weights. The BP algorithm is gradient descent in essence. Despite its popularity as an optimization tool for neural network training, the BP algorithm has several drawbacks. For instance, the training convergence speed is very slow and easy entrapment in a local minimum. Furthermore, 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. These limitations cause the Back-propagation neural network (BPN) to become inconsistent and unpredictable on different applications [4], [8]. Therefore, it is necessary to develop an effective technique for optimizing BPN in order to improve its forecasting performance.

Recently some investigations into neural network training using Evolutionary algorithms (EAs) have been successfully employed to overcome the inherent limitations of the BP [4], [9]. EAs are considered to be capable to reduce the ill effect of the BP algorithm, because they do not require gradient and differentiable information. Among existing EAs, the most well-known branch is genetic algorithms (GAs). Though GA has many advantages over conventional methods, it also has some drawbacks, such as slow rate of convergence and a large number of simulations to arrive at an optimum solution.
Recently researchers reported that particle swarm optimization (PSO) [2] is showing some improvement on these issues, which use the local and global search capabilities of the algorithm, to find better quality solutions with less computational time [7].

In this paper, the hybrid PSO-BP algorithm is employed to train multilayer perceptrons for daily flow forecasting at a Semi-arid catchment in Morocco. This algorithm uses the PSO algorithm to do global search in the beginning of stage, and then uses the BP algorithm to do local search around the global optimum.

2. Adaptation of PSO to network training

The PSO algorithm is an adaptive algorithm based on a social-psychological metaphor. PSO is similar to the GA in that it begins with a random population matrix. The rows in the matrix are called particles (same as the GA chromosome). They contain the variable values and are not binary encoded. Each particle of the population has an adaptable velocity (position change), according to which it moves in the search space. Moreover, each particle has a memory, remembering the best position of the search space it has ever visited [2]. When the PSO algorithm is used in evolving weights of ANN, each particle represents all weights of a ANN structure. The representation of the connection weight matrix of the i-th particle is as follows:

$$W_i = \left\{ W_i^{[1]}, W_i^{[2]} \right\}, \tag{2.1}$$

where $W_i^{[1]}$ and $W^{[2]}$ represent the connection weight matrix of the i-th particle between the input layer and the hidden layer, and that between the hidden layer and the output layer, respectively. Moreover, the vector of the position of the previous best fitness value of any particle is represented by

$$P_i = \left\{ P_i^{[1]}, P_i^{[2]} \right\}, \tag{2.2}$$

where $P_i^{[1]}$ and $P_i^{[2]}$ represent the position of the previous best fitness value of the i-th particle, between the input layer and the hidden layer, and that between the hidden layer and the output layer, respectively.
The index of the best particle among all the particles in the population is represented by the symbol g. So the best matrix is represented by

$$P_g = \left\{ P_g^{[1]}, P_g^{[2]} \right\}, \tag{2.3}$$

where $P_g^{[1]}$ and $P_g^{[2]}$ represent the position of the best particle among all the particles, between the input layer and the hidden layer, and that between the hidden layer and the output layer, respectively. The velocity of the particle i is denoted by

$$V_i = \left\{ V_i^{[1]}, V_i^{[2]} \right\}. \tag{2.4}$$

If m and n represent the index of matrix row and column, respectively, the manipulation of the particles are as follows:

$$\begin{aligned} V_i^{[j]}(m,n) &= \chi\big[\omega V_i^{[j]}(m,n) + c_1 r_1 \big(P_i^{[j]}(m,n) - W_i^{[j]}(m,n)\big) && (2.5) \\ &\quad + c_2 r_2 \big(P_g^{[j]}(m,n) - W_i^{[j]}(m,n)\big)\big], && (2.6) \end{aligned}$$

and

$$W_i^{[j]} = W_i^{[j]} + V_i^{[j]}, \tag{2.7}$$

where j=1, 2; m=1, , M_j; n=1, , N_j; M_j and N_j are the row and column sizes of the matrices W, P, and V; c_1 and c_2 are two positive constants named as learning factors, r_1 and r_2 are random numbers in the range of (0,1). ω is called inertia weight and and χ is a constriction factor which is used, alternatively to ω to limit velocity. Eq. (2.7) is used to calculate the particle's new velocity according to its previous velocity and the distances of its current position from its own best position and the group's best position. Then, the particle flies toward a new position according to Eq. (2.8). Proper fine-tuning of the parameters c_1 and c_2, in Eq. (2.7), may result in faster convergence of the algorithm, and alleviation of the problem of local minima. The role of inertial weight ω, in Eq. (2.7), is to control the impact of the previous velocities on the current one. A large inertial weight facilitates global exploration (searching new areas), while a small weight tends to facilitate local exploration. Hence selection of a suitable value for the inertial weight ω usually helps in reduction of the number of iterations required to locate the optimum solution [5]. The variable ω is updated according to

$$\omega = \frac{\text{maxiter} - \text{iter}}{\text{maxiter}}, \tag{2.8}$$

where *iter* is the current iteration number and *maxiter* is the maximum number of allowable iterations.

3. Hybrid PSO-BP algorithm

The PSO-BP is an optimization algorithm combining the PSO with the BP for weight training of multi-layer neural network. The PSO algorithm is a global algorithm, which has a strong ability to find global optimal result, this PSO algorithm, however, has a disadvantage that the search around global optimum is very slow. The BP algorithm, on the contrary, has a strong ability to find local optimal result, but its ability to find the global optimal result is weak. The fundamental idea for this hybrid algorithm is that at the beginning stage of searching for the optimum, the PSO is employed to accelerate the training speed. When the fitness function value has not changed for some iterations, or value changed is smaller than a predefined number, the searching process is switched to gradient descending searching according to this heuristic knowledge.

4. Application Case

Data used in this paper are from the Ourika basin, located in semi-arid region of Marrakech and is the most important sub-catchment of Tensift basin drainage. The basin area is $503\ km2$, and its mean annual rainfall is approximately $800\ mm$. The daily streamflow at Aghbalou station is used for model investigation. The data contains information for a period of eight years (1996 to 2003). Furthermore, data from 1996 to 2002 constitute the training set and the 365 remaining data are used in the testing phase.

5. Results and discussion

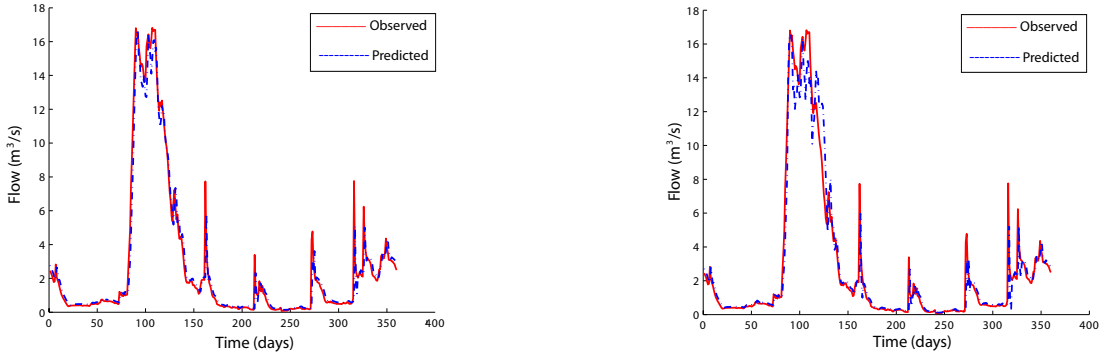
To forecast the runoff for day t, the input layer consists of 3 nodes representing the daily runoff values at times $t-1$, $t-2$ and $t-3$. Figure 1 shows comparison of results between the computed and observed flow during the testing period of 2003. Figure 1(a) shows the result of forecasting using PSO-PB algorithm. In addition Figure 1(b), shows the result of forecasting using PB algorithm. PSO-PB algorithm was trained by 30 generations, followed by a BP training. The results produced by the ANN based on PSO-BP algorithm are more close to the original data than those by BP neural network. To evaluate the models performance, three criteria are adopted: Mean Square Error (MSE), efficiency coefficient R^2 and mean relative error (MRE). Table 2 gives the performance comparisons for the two different models of the testing phase. Apparently, the hybrid algorithm generates more accurate predictions and exhibits better performance than the PB algorithm.
The convergence processes for ANN based on PSO-BP algorithm is shown in Figure 2. Figure 2a shows the best MSE corresponding to each generation during the PSO algorithm training. From this figure, we can see the variation of best MSE represented by the best particle with the number of generations of the PSO algorithm. After 10 generations of execution, the change of best MSE has become slower because of the around global optimum. After 30 generations the PSO-BP algorithm was switched to the gradient descending method based on BP algorithm to search for 6000 iterations. Figure 2b shows the MSE corresponding to each iteration during the PB training. This figure shows that, although PSO has greatly reduced the total MSE of the neural network, a more improvement of training performance could still be achieved by applying a back propagation weight adjustment procedure. Table 2 shows the training performance at various fitness evaluation times for both the standard PB and the hybrid PSO-PB. The maximum iteration number for each algorithm is set to 6000. The MSE reaches 0.0016 after 10 times of iteration, which is much faster than for BP algorithm, where the MSE does not reach 0.0016 in 6000 iterations and the minimum MSE available by PSO-BP can be 0.0013 in 4500 times of iteration. Thus, the PSO-BP algorithm presents better convergence performance than the BP algorithm.

6. Conclusion

This paper presents the application of a hybrid PSO algorithm for training ANNs to forecast the daily streamflows in a catchment located in a semi-arid climate in Morocco. The hybrid algorithm combining the PSO algorithm's strong ability in global search with PB algorithm's strong ability in local search. It is shown from the training and verification simulation that the streamflow forecasting results are more accurate and present better convergence performance when compared with the conventional BP neural network.

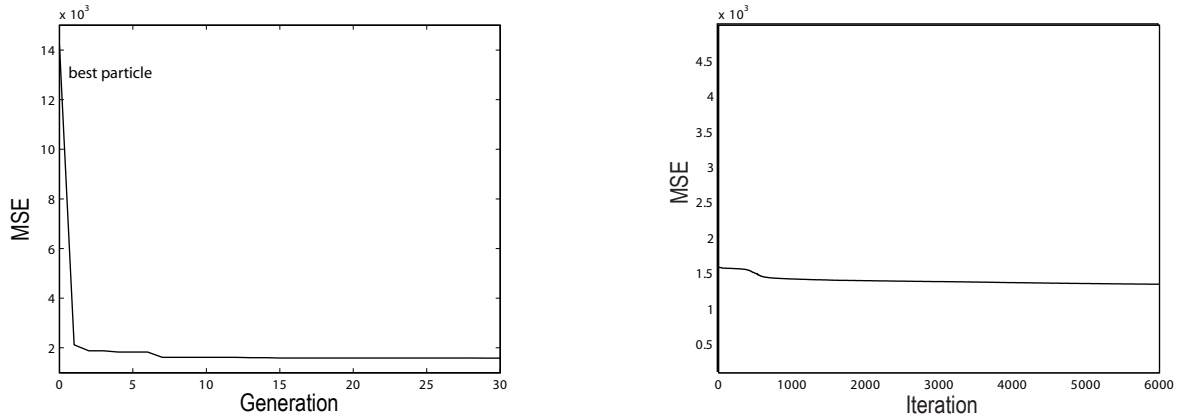
References

[1] Q. Duan, S. Sorooshian, V. Gupta. *Effective and efficient global optimization for conceptual rainfall runoff models.* Water Resour. Res, 28 (1992), 1015-1031



(a) PSO-BP algorithm (b) BP algorithm

Figure 1: Comparison between predicted and measured flow values at testing phases.



(a) Best MSE by PSO algorithm (b) MSE during BP training

Figure 2: Neural network training performance for PSO-BP algorithm.

Algorithm	R^2	MSE	MRE
PSO-BP	0.96	0.0019	0.0206
BP	0.93	0.0030	0.0284

Table 1: Comparison of PSO-BP and BP neural networks at testing phase.

Fitness valuation time	MSE	
	PSO-BP	BP
10	0.0016	0.0089
2000	0.0014	0.0023
4500	0.0013	0.0017
6000	0.0013	0.0017

Table 2: Convergence comparison of the PSO-BP and BP neural networks.

[2] R. C. Eberhart and J. Kennedy. *A new optimizer using particle swarm theory*Proc. of 6th Symp. on micro machine and human science, IEEE service center, Piscataway, N.J., (1995), 39-43.

[3] K.L. Hsu, H.V. Gupta, S. Sorooshian. *Artificial neural network modeling of the rainfall-rainoff process.* Water Resour. Res., 31 (1995), No. 10, 2517-2530.

[4] V. Maniezzo. *Genetic evolution of the topology and weight distribution of neural networks.* IEEE Transaction on Neural Networks, 5 (1994), 39–53.

[5] K.E. Parsopoulos, M.N. Vrahatis. *Recent approaches to global optimization problems through particle swarm optimization*. Natural Comput., 1 (2002), No. 23, 235-306.

[6] D.E. Rumelhart, G.E. Hinton, R.J. Williams. *Learning internal representation by error propagation.* In: Rumelhart, D.E., McClelland, J.L. (Eds.), Parallel Distributed Processing: Explorations in the Microstructure of Cognition, vol. 1. MIT Press, Cambridge, MA, (1986), 318-362.

[7] A. Salman, I. Ahmad, S. Al-Madani. *Particle swarm optimization for task assignment problem.* Microproc. and Microsyst., 26 (2002), No. 8, 363-371.

[8] R. S. Sexton, R. E. Dorsey, J. D. Johnson. *Toward global optimization of neural networks: A comparison of the genetic algorithm and back propagation.* Decision Support Systems, 22 (1998), 171–185.

[9] J.M. Yang, C.Y. Kao. *A robust evolutionary algorithm for training neural networks.* Neural Comput. Appl., 10 (2001), 214–230.