ABCM Symposium Series in Mechatronics - Vol. 2 - pp.751-756

LEARNING OF B-SPLINE NEURAL NETWORK USING NEW PARTICLE SWARM APPROACHES

Leandro dos Santos Coelho
Pontifical Catholic University of Paraná, LAS/PPGEPS/CCET/PUCPR, Production and Systems Engineering Graduate Program
Imaculada Conceição, 1155, Zip code 80215-901, Curitiba, Paraná, Brazil
leandro.coelho@pucpr.br

Renato Krohling
Lehrstuhl Elektrische Steuerung und Regelung
Fakultät für Elektrotechnik und Informationstechnik
sd Universität Dortmund, D-44221 Dortmund, Germany
krohling@est.e-technik.uni-dortmund.de

Abstract*. New approaches of particle swarm optimisation algorithm based on Gaussian and Cauchy distributions to adjust the control points of B-spline neural networks are proposed. B-spline networks are trained by gradient-based methods, which may fall into local minimum during the learning procedure. To overcome the problems encountered by the conventional learning methods, particle swarm optimisation — a swarm intelligence methodology — can provide a stochastic search for global optimisation of B-spline networks for nonlinear system identification. Simulation results show the potential of the proposed optimisation with particle swarm of B-spline networks for the identification of Rössler system.*

Keywords*: neural networks, B-splines, nonlinear system identification, particle swarm optimization.*

1. Introduction

The problem of nonlinear function approximation and system identification has been studied in many fields of Science by many researchers and can be solved by various mathematical approaches. The conception of mathematical models for the representation of complex systems of time series is an excellent procedure and with practical applications. However, in general, the construction of adjusted mathematical models for the engineering intentions is not a simple task. In the last decades, diverse conceptions of algorithms for modeling and identification of complex dynamics systems have been proposed in literature, such as: frequency methods, techniques based on estimates of Wiener models, Hammerstein, bilinear and Volterra, nonlinear regression, wavelets and recursive identification (Ljung 1987; Haber and Unbehauen 1990). An excellent approach, between much others, for mathematical representation of dynamics systems with complex or chaotic behavior it is of the neural networks.

A relevant approach is to find the best approximation with respect to certain class of basis functions for neural networks representation. In this case, there are many possible choices of basis functions, such as radial basis function, associate memory networks, wavelets, and B-spline function.

The main advantage of the B-spline functions over other radial functions e.g., the Bezier curve, is the local control of the curve shape, as the curve only changes in the vicinity of a few control points that have been changed (Newmann and Sproull, 1979). A B-spline neural network (BSNN) consists of the piecewise polynomials with a set of local basis functions to model an unknown function for which a finite set of input-output samples are available. The performance of the identification depends largely of an optimisation algorithm for the training procedure of BSNN in order to avoid any possible local minima. In this Letter, we propose a modified particle swarm optimisation (PSO) approach to train the BSNNs. Simulation results for the identification of a chaotic temporal series show the feasibility and effectiveness of the proposed approach.

The paper is organized as follows. Basic concepts of B-spline neural networks are presented in section 2. In section 3, the design procedure of new PSO approaches is proposed. In the section 4, the simulation results of identification of a nonlinear system are presented and discussed. The conclusions and futures works are commented in the section 5.

2. B-splines neural networks

BSNN is introduced as a class of one-hidden-layer feedforward neural networks composed of B-spline functions. Each basis function is composed of q polynomial segments. There exists a simple and stable recursive relationship for evaluating the membership of a B-spline basis function of order k,

$$N_q^j(x) = \left(\frac{x - \lambda_{j-q}}{\lambda_{j-1} - \lambda_{j-q}} \right) N_{q-1}^{j-1}(x) + \left(\frac{\lambda_j - x}{\lambda_j - \lambda_{j-q+1}} \right) N_{q-1}^{j}(x) \tag{1}$$

$$N_1^j(x) = \begin{cases} 1 & if\,(x \in I_j) \\ 0 & otherwise \end{cases} \tag{2}$$

where $N_q^j(\cdot)$ is defined as the *j*-th univariate basis function of order *q* and λ_j the *j*-th knot and I_j is the *j*-th interval. A output of neural network is

$$\hat{o}_k = f(x_k) = \sum_{j=1}^{p} w_j N_q^j(x_k) \tag{3}$$

where x_k and $\hat{o}_k$ are the inputs and output of network, respectively, w_j is the weight attached to the *j*-th basis function and $N_q^j(\cdot)$ is given by the recursive form (2). The index *j* is associative with the region of local support $\lambda_{(j-q)} \le x \le \lambda_{(j)}$, whereas the index *q* indicates the order of the basis functions (Harris *et al*., 1993).

The quality of approximation depends on the placement of knots of B-spline functions. The objective of optimisation of BSNNs by PSO is determination of the knots of each B-spline basis functions. However, the number of basis functions in this work is choice of user.

3. Particle swarm optimisation for training BSNN

PSO is a kind of evolutionary algorithm based on a population of individuals and motivated by the simulation of social behaviour instead of the survival of the fittest individual. It is a population-based evolutionary algorithm. Similar to the other population-based evolutionary algorithms, PSO is initialised with a population of random solutions. Unlike the most of the evolutionary algorithms, each potential solution (individual) in PSO is also associated with a randomised velocity, and the potential solutions, called *particles*, are then "flown" through the problem space.

Each particle keeps track of its coordinates in the problem space, which are associated with the best solution (fitness) it has achieved so far. This value is called *pbest*. Another "*best*" value that is tracked by the *global* version of the particle swarm optimiser is the overall best value, and its location, obtained so far by any particle in the population. This location is called *gbest*. The particle swarm optimisation concept consists of, at each time step, changing the velocity (accelerating) of each particle flying toward its *pbest* and *gbest* locations (global version of PSO). Acceleration is weighted by random terms, with separate random numbers being generated for acceleration toward *pbest* and *gbest* locations, respectively. The procedure for implementing the global version of PSO is given as follows (Krohling *et al*., 2002):

Step 1. *Initialisation*: Initialise a population (array) of particles with random positions and velocities in the *n* dimensional problem space.
Step 2. *Evaluation*: For each particle, evaluate its fitness value.
Step 3. *Comparison* 1: Compare each particle's fitness evaluation with the particle's *pbest*. If current value is better than *pbest*, then set *pbest* value equal to the current value and the *pbest* location equal to the current location in *n*-dimensional space.
Step 4. *Comparison* 2: Compare fitness evaluation with the population's overall previous best. If current value is better than *gbest*, then reset *gbest* to the current particle's array index and value.
Step 5. *Updating*: Change the velocity and position of the particle according to equations (4) and (5), respectively:

$$\boldsymbol{v_i} = w \cdot \boldsymbol{v_i} + c_1 \cdot d(\,) \cdot (\boldsymbol{p_i} - \boldsymbol{x_i}) + c_2 \cdot D(\,) \cdot (\boldsymbol{p_g} - \boldsymbol{x_i}) \tag{4}$$

$$\boldsymbol{x}_i = (\boldsymbol{x}_i + \boldsymbol{v}_i)\ . \tag{5}$$

Step 5. *Stop criterion*: Loop to step 2 until a criterion is met, usually a sufficiently good fitness or a maximum number of iterations (generations).

where $\boldsymbol{x}_i = [x_{i1}, x_{i2}, ..., x_{in}]^{\mathrm{T}}$ stands for the position of the *i*-th particle, $\boldsymbol{v}_i = [v_{i1}, v_{i2}, ..., v_{in}]^{\mathrm{T}}$ stands for the velocity of the *i*-th particle and $\boldsymbol{p}_i = [p_{i1}, p_{i2}, ..., p_{in}]^{\mathrm{T}}$ represents the best previous position (the position giving the best fitness value) of the *i*-th particle. The index *g* represents the index of the best particle among all the particles in the group. Variable *w* is the inertia weight, c_1 and c_2 are positive constants; *d*( ) and *D*( ) are two random functions in the range [0, 1]. Particles' velocities on each dimension are clamped to a maximum velocity *Vmax*. If the sum of accelerations would cause the velocity on that dimension to exceed *Vmax*, which is a parameter specified by the user, then the velocity on that dimension is limited to *Vmax*. *Vmax* is an important parameter. It determines the resolution with which the regions

around the current solutions are searched. If *Vmax* is too high, the PSO facilitates global search, and particles might fly past good solutions. If *Vmax* is too small, on the other hand, the PSO facilitates local search, and particles may not explore sufficiently beyond locally good regions.

The first part in equation (4) is the momentum part of the particle. The inertia weight *w* represents the degree of the momentum of the particles. The second part is the "cognition" part, which represents the independent thinking of the particle itself. The third part is the "social" part, which represents the collaboration among the particles. The constants c_1 and c_2 represent the weighting of the "cognition" and "social" parts that pull each particle toward *pbest* ($\boldsymbol{p_i}$) and *gbest* ($\boldsymbol{p_g}$) positions.

In this work, new approaches to PSO, named fast PSO are proposed which are based on the studies of mutation operators in fast evolutionary programming (Krohling *et al*., 2002). The aim is to modify the equation (4) of the conventional PSO (case 1) with $d(\ )$ and $D(\ )$ based on uniform distribution to use it with Gauss or Cauchy distribution in the range [0, 1].

4. Identification of a Rössler' chaotic system

Chaos' theory studies pertinent phenomenon to the nonlinear dynamic systems (any process that evolve with the time) and that presents complex behavior to be treat mathematically. The theory of the chaos studies the unexpected phenomenon apparently, in the search of hidden standards and simple laws that conduct the complex behaviours. However, this study if became effectively reasonable from the decade of 1960, when the computers had started to possess reasonable graphical capacity and of processing, giving to the physicists and mathematicians the power to discover answers for basic questions of the science in general way, that before were obscure.

The nonlinear systems, had appeared from the chaos' theory that supplies to an explanation, many times adequate, (through formulas and equations), to many behaviors current in the nature, such as: natural phenomenon (populations, turbulence, fluid movement, and cloud formation), complex behaviors in electric circuits, behavior of stock exchange and economy, nonlinear systems and variant in the time, telecommunications, system control, dynamic behavior of the cardiac beatings, among others.

The behavior of the chaotic systems can present the great sensitivity in relation to the initial conditions that are applied. Although, to be a difficult task to describe the behaviors of a chaotic system, in probabilistic terms this situation can be treated and some paradigms have been presented in literature for this purpose.

In this work, it considers the nonlinear Rössler system (Rössler, 1976). The german scientist O. Rössler proposed a chaotic attractor composed by nonlinear differential equations given by,

$$\frac{dx}{dt} = -y - z \tag{6}$$

$$\frac{dy}{dt} = x + ay \tag{7}$$

$$\frac{dz}{dt} = b + (x - c)z \tag{8}$$

which exhibits a chaotic behavior at the popular parameters values (a, b, c) = (0,36; 0,40; 4,50). This behavior is illustrated in figure 1.

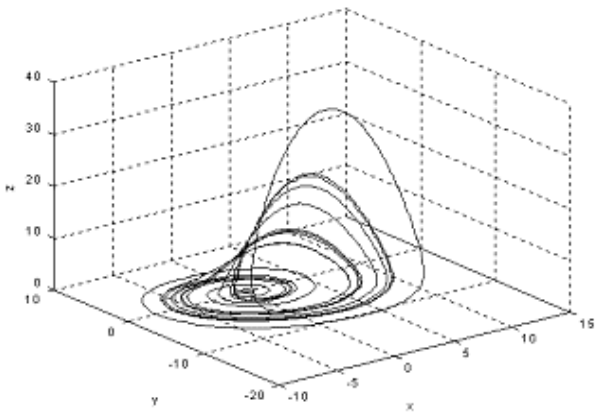


Figure 1. Rossler' attractor system for (a, b, c) = (0.36, 0.40, 4.50).

The processes identification is a procedure to identify a model of an unknown process, for intentions of forecast and/or understanding of the dynamic behavior of the dynamic system. A model describes reality in some way, and system identification is the theory of how mathematical models for dynamical systems are constructed form observed data. Typically, a parameterized set of models, a model structure, is hypothesized and data is used to find the best model within this set according to some criterion. The choice of model structure is guided by prior knowledge or assumptions about the system that generated the data. When little prior knowledge is available it is common to use a black-box model. A black-box model is a standard flexible structure and it can be used to approximate a large variety of different systems. Neural network models have proven to be successful non-linear black-box model structures in many applications.

The objective of this work is the identification of *x*-coordinate time series by BSNN using PSO. A measure of the BSNN accuracy can be found by evaluating the mean squared error (*MSE*),

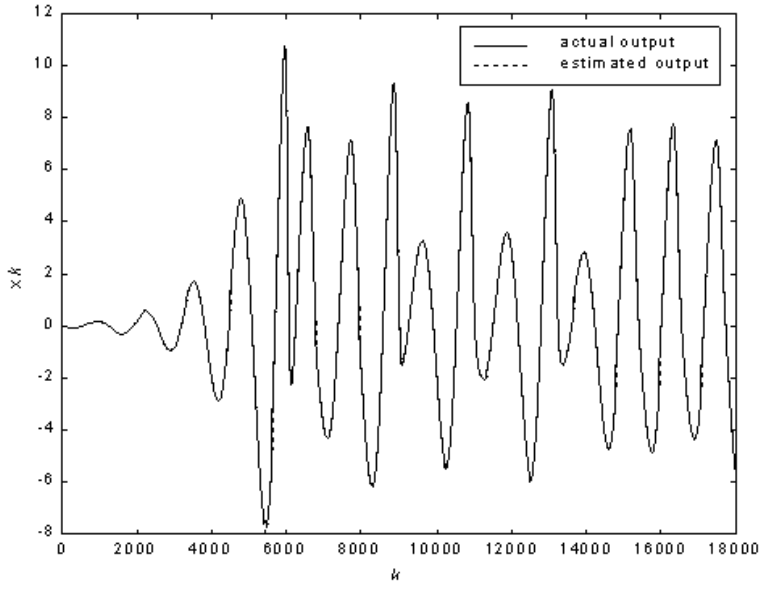
$$MSE = \frac{\sum_{i=1}^{m} (o_k - \hat{o}_k)^2}{m} \tag{9}$$

where *i* is the index of the *m* points over which the *MSE* is computed, o_j is the actual output (*x-coordinate*) of the system at the *j*-th input vector, and $\hat{o}_j$ is the estimated output of the BSNN at the *j*-th vector input. The objective of PSO is the minimization of *MSE*.

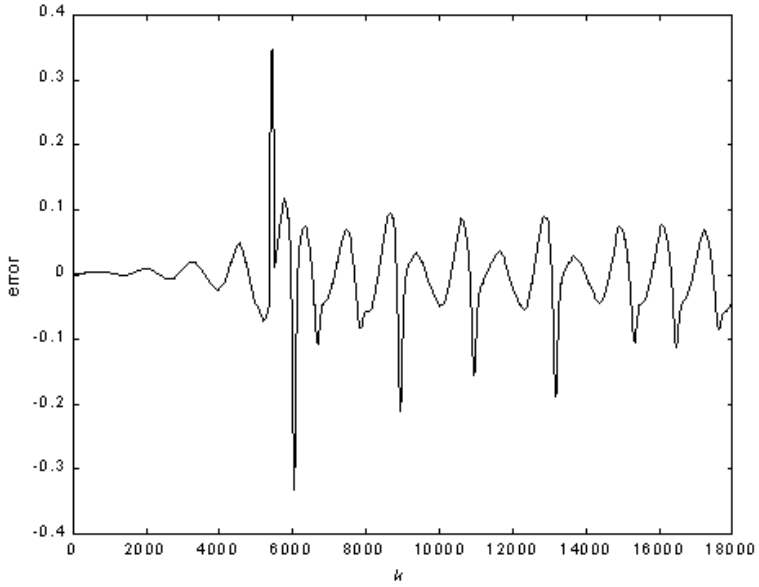
Based on previous experience with particle swarm optimisation (trial and error, mostly) led us to set the acceleration constants c_1 and c_2 equal to 0.2; *Vmax* set to 20% of the dynamic range of the variable on each dimension, *w*=0.3 and the number of particles to 10. The stop criterion of PSO is 15 generations. The range of space search of BSNN parameters by PSO is [0.001 1]. The inputs of BSNN are [x_{k-1} x_{k-2} x_{k-3}] and the estimated output is $\hat{o}_j = x_{k+1}$ (Nonlinear AutoRegressive model in serie-parallel conception). The data of estimation phase of process mathematical model are constituted of samples 1 to 1000. However, the samples 1001 to 1500 are utilised in validation phase of the obtained mathematical model.

PSO approaches optimise three knots of order 2 for each input of BSNN, e.g., the total of optimised parameters is 9. The matrix weights *w* of basis function is obtained by pseudo-inverse of Penrose-Moore.

PSO approaches are compared in statistical terms and the results are summarised in Tables 1 and 2. Table 1 shows that the PSO(1) to PSO(7) were outperformed by PSO(9) based on mean, maximum and standard deviation of *MSE*. It also shows that the PSO(9) was quite robust was able to deal with BSNN optimisation. However, the best result (minimum) of *MSE* was presented by PSO(8). Figure 2 shows the best result obtained by PSO(8) with *MSE* equal to 0,002453 and 0,002231 for estimation and validation phases, respectively. Tables 1 and 2 shows that PSO(4) was trapped in to local optima and failed to explore new areas.



(a) time series



(b) estimation error

Figure 2. Best result for BSNN optimization (prediction one step ahead) using PSO(8).

Table 1. Results obtained in estimation phase by the minimisation of *MSE* using PSO (it is adopted the particle with best *MSE* after the accomplishment of 10 experiments)

PSO	distribution of generation of random numbers		*MSE* (estimation phase)			
type	*d*( )	*D*( )	mean	minimum (best)	maximum (worst)	standard deviation
1	Uniform	Uniform	0.003426	0.002719	0.004165	0.000654
2	Cauchy	Uniform	0.003284	0.002805	0.004285	0.000580
3	Uniform	Cauchy	0.002940	0.002708	0.003234	0.000207
4	Cauchy	Cauchy	0.004616	0.002643	0.012163	0.004219
5	Gaussian	Uniform	0.002871	0.002597	0.003084	0.000195
6	Uniform	Gaussian	0.003142	0.002562	0.004126	0.000611
7	Gaussian	Gaussian	0.002917	0.002737	0.003185	0.000191
8	Gaussian	Cauchy	0.004302	**0.002453**	0.007781	0.002036
9	Cauchy	Gaussian	**0.002715**	0.002516	**0.002884**	**0.000142**

Table 2. Results obtained in validation phase by the minimisation of *MSE* using PSO (it is adopted the particle with best *MSE* after the accomplishment of 10 experiments)

PSO	distribution of generation of random numbers		*MSE* (validation phase)			
type	*d*( )	*D*( )	mean	minimum (best)	maximum (worst)	standard deviation
1	Uniform	Uniform	0.002579	0.002473	0.002665	**0.000085**
2	Cauchy	Uniform	0.002714	0.002552	0.002858	0.000110
3	Uniform	Cauchy	0.002540	0.002446	**0.002613**	0.000079
4	Cauchy	Cauchy	0.004151	0.002355	0.010869	0.003756
5	Gaussian	Uniform	0.002597	0.002363	0.002789	0.000170
6	Uniform	Gaussian	0.002547	0.002291	0.002917	0.000252
7	Gaussian	Gaussian	0.002593	0.002490	0.002757	0.000101
8	Gaussian	Cauchy	0.002687	**0.002231**	0.003036	0.000308
9	Cauchy	Gaussian	**0.002470**	0.002288	0.002624	0.000129

5. Conclusion and future works

In the last years of interest in the development of strategies of nonlinear identification of chaotic systems reappeared. This interest is motivated by diverse factors, such as: (i) advances of the nonlinear systems theory, causing applicable methodologies of project to an extension of control nonlinear problems; (ii) development of efficient identification methods for the treatment of empirical nonlinear models; (iii) continued development of the capacities of software and the hardware, becoming possible the incorporation of complex nonlinear models in control systems design.

The behavior of the chaotic systems can present the great sensitivity in relation to the initial conditions that are applied. Although, to be a difficult task to describe the behavior of a chaotic system, in probabilistic terms this situation can be treated and some paradigms have been presented in literature for this purpose. The behavior of a chaotic system can be evaluated through the configuration of nonlinear identification methodologies. The methodologies that can be used include the fuzzy systems, neural networks, genetic algorithms, among others approaches.

In this work, new approaches of particle swarm optimisation algorithm based on Gaussian and Cauchy distributions to adjust the control points of B-spline neural networks were proposed. B-spline networks are trained by gradient-based methods, which may fall into local minimum during the learning procedure.

PSO using uniform probability distribution suffers form easy entrapment when the particles lie in a locally optima region. In this Letter any new PSO approaches has been introduced for BSNN learning. In this context, new PSO approaches using the combination of other probabilities distribution theories, e.g., Gaussian and Cauchy seem to be promising to escape form local minima. The use of Cauchy distribution in PSO could be useful to escape of local minima, while the Gauss distribution could provide a faster convergence in local searches.

Simulation results show the potential of PSO for the BSNNs optimisation in systems identification. Wolpert and Macready (1997) have already proved that there is no generic optimisation algorithm that will work better in general on all problems. In this context, more works need be done to test the new PSO approaches on benchmark optimisation problems.

6. References

Haber, R. and Unbehauen, H., 1990, “Structure Identification of Nonlinear Dynamic Systems — A Survey on Input/Output Approaches”, Automatica, Vol. 26, No. 4, pp. 651-677.

Harris, C. J., Moore, C. G. and Brown, M., 1993, Intelligent Control: Aspects of Fuzzy Logic and Neural Nets, Word Scientific.

Krohling, R. A., Coelho, L. S. and Shi, Y., 2002, “Cooperative Particle Swarm Optimisation for Robust Control System Design”, 7th Word Conference on Soft Computing in Industrial Applications, Granada, Spain.

Ljung, L., 1987, System Identification: Theory for the User, Prentice-Hall: New York.

Newmann, W. M. and Sproull, R. F., 1979, Principles of Interactive Computer Graphics, Mc-Graw-Hill, Inc.

Rössler, O. E., 1976, “An Equation for Continuous Chaos”, Physical Letters, 35A, pp. 397-398.

Wolpert, D. H. and Macready, W. G., 1997, “No Free Lunch Theorems for Optimization”, IEEE Transactions on Evolutionary Computation, Vol. 1, No. 1, pp. 67-82.

Yao, X. and Liu, Y., 1996, “Fast Evolutionary Programming”, Proceedings of 5th Annual Conference on Evolutionary Programming, San Diego, CA, pp. 451-460.

7. Responsibility notice

The authors are the only responsible for the printed material included in this paper.