Bio-inspired Aging Model Particle Swarm Optimization Neural Network Training for Solar Radiation Forecasting

Eduardo Rangel[1], Alma Y. Alanís[1], Luis J. Ricalde[2],
Nancy Arana-Daniel[1], and Carlos López-Franco[1]

[1] CUCEI, Universidad de Guadalajara, Apartado Postal 51-71, Col. Las Aguilas,
C.P. 45080, Zapopan, Jalisco, Mexico
almayalanis@gmail.com

[2] UADY, Faculty of Engineering, Av. Industrias no Contaminantes por Periferico
Norte, Apdo. Postal 115 Cordemex, Merida, Yucatan, Mexico

Abstract. This paper deals with a novel training algorithm for a neural network architecture applied to solar radiation time series prediction. The proposed training algorithm is based in a novel bio-inspired aging model-particle swarm optimization (BAM-PSO). The BAM-PSO based algorithm is employed to update the synaptic weights of the neural network. The size of the regression vector is determined by means of the Cao methodology. The proposed structure captures efficiently the complex nature of the solar radiation time series. The proposed model is trained and tested using real data values for solar radiation.

1 Introduction

The limited existing reserves of fossil fuels and the harmful emissions associated with them have led to an increased focus on renewable energy applications recently. Among renewable energy sources, solar energy is one of the most important techniques. However, in practice the integration of solar energy into the existing electricity supply system is a real challenge because its availability mainly depends on meteorological conditions, which cannot directly be changed by human intervention. For this reason it is important to have a reliable estimation of solar radiation.

Integration of solar radiation forecast and output power is a good way to improve the performance in scheduling for microgrids. Solar radiation forecasting is not an easy task; solar radiation has a stochastic nature with high rate of change. Solar radiation time series present highly nonlinear behavior with no typical patterns and a weak seasonal character [1]. Several methods have been proposed to accomplish solar radiation forecasting like numerical weather prediction systems, statistical approaches and artificial neural networks using feedforward or recurrent structures ([2], [3]). Soft Computing methods are more suitable for short term predictions; these methods are based on time series historical data in order to build a mathematical model which approximates the input-output relationship.

E. Bayro-Corrochano and E. Hancock (Eds.): CIARP 2014, LNCS 8827, pp. 682–689, 2014.

On the other hand, in recent years, there have been considerable interest in nature-inspired optimization algorithms in computer sciences. Bio-inspired optimization algorithms achieve exceptional results because their mathematical models are closely related to one or other characteristic of natural behavior [6]. The Particle Swarm Optimization algorithm has been one of the most promising methods to solve modern engineering problems. In PSO, an effect called premature convergence appears when most (or all) of the particles within a swarm compromises their ability to explore and stay close to a local minima. Particle Swarm Optimization with an Aging Leader and Challengers (ALC-PSO) [7], has been the first approach to include aging processes to intend to alleviate premature convergence; however, this approach only includes a leader-ship oriented and not swarm related, even more important, the aging dynamics is linear and bounded to static predefined values. Our novel PSO variant: Particle Swarm Optimization with Bio-inspired Aging Model (BAM-PSO) use a mathematical model that describes the aging dynamics of the immune cells to propose a mean to control the existence of each particle within the swarm in order to avoid premature convergence effect. Therefore, the main contribution of this paper is the use of BAM-PSO to train an artificial neural network in order to perform a time series forecasting applied for solar radiation in a microgrid.

The paper is organized as follows: first the neural identification scheme is proposed; then, the BAM-PSO algorithm is introduced to train the neural identifier; after that, the proposed scheme is applied for time series prediction of solar radiation using experimental information obtained from a photovoltaic tower which is part of a microgrid benchmark. Finally some important conclusions are stated.

2 Neural Identification

In this paper for the neural model identification the well-known RMLP (Recurrent Multi-Layer Perceptron) is selected, then the neural model structure definition reduces to dealing with the following issues: selecting the inputs to the network and 2) selecting the internal architecture of the network. The structure selected in this paper is NNARX [8] (acronym for Neural Network AutoRegressive eXternal input); the output vector for the artificial neural network is defined as the regression vector of an AutorRegressive eXternal input linear model structure (ARX) [9]. It is common to consider a general nonlinear system; however, for many control applications is preferred to express the model in an affine form, which can be represented by the following equations

$$y\left(k+1\right)=f\left(y\left(k\right),y\left(k-1\right),...,y\left(k-q+1\right)\right) \tag{1}$$

where q is the dimension of the regression vector. In other words, a nonlinear mapping f exists, for which the present value of the output $y\left(k+1\right)$ is uniquely defined in terms of its past values $y\left(k\right),...,y\left(k-q+1\right)$ and the present values of the input $u\left(k\right)$. Considering that it is possible to define:

$$\phi\left(k\right)=\left[y\left(k\right),...,y\left(k-q+1\right)\right]^{T}$$

which is similar to the regression vector of a ARX linear model structure [8], then the nonlinear mapping f can be approximated by a neural network defined as

$$y\left(k+1\right)=\varphi\left(\phi\left(k\right),w^{*}\right)+\varepsilon$$

where w^{*} is an ideal weight vector, and ε is the modeling error; such neural network can be implemented on predictor form as

$$\widehat{y}\left(k+1\right)=\varphi\left(\phi\left(k\right),w\right) \tag{2}$$

where w is the vector containing the adjustable parameters in the neural network. The neural network structure, used in this work is depicted below in Fig. 2, which contains sigmoid units only in the hidden layer; the output layer is a linear one. The used sigmoid function $S\left(\bullet\right)$ is defined as a logistic one, as follows:

$$S\left(\varsigma\right)=\frac{1}{1+\exp\left(-\beta\varsigma\right)},\qquad\beta>0 \tag{3}$$

where ς is any real value variable.

3 Bio-inspired Aging Model-Particle Swarm Optimization Neural Network Training Algorithm

3.1 Particle Swarm Optimization

The PSO algorithm consists of an iterative adaptation of set of multidimensional vectors called particles that communicate information among them as a swarm which provides a set of candidate solutions for an objective function. This objective function is normally multimodal and high dimensional. The mathematical model of the described algorithm reads as follows

$$v_{ij}(t+1)=v_{ij}(t)+c_{1}R_{1}(p_{ij}(t)-x_{ij}(t))+c_{2}R_{2}(p_{gj}(t)-x_{ij}(t)) \tag{4}$$

$$x_{ij}(t+1)=x_{ij}(t)+v_{ij}(t) \tag{5}$$

where i $(i=1,2,...,S)$ is the ith particle of a swarm that satisfies $S\in\mathbb{R}^{D}$, and j $(j=1,2,...,D)$ is the jth element of dimension problem D. Also t represents the iteration counter, R_1 and R_2 are random, normalized and uniformly distributed values. c_1 and c_2 represent the social and cognitive parameter, x_{ij} is the particle ij position for the iteration t; $x_{ij}(t+1)$ is the particle ij position for $t+1$ iteration, $v_{ij}(t)$ is the particle ij velocity for t iteration. $p_{ij}(t)$ represents the local best position for particle ij in iteration t and $p_{gj}(t)$ represents the global best position for entire swarm in iteration t. In order to improve the PSO algorithm performance, an inertial weight vector is introduced in a variation of PSO [6]; this variation, introduce a dynamic constriction factor for velocity vectors in order to concentrate the swarm around the most promising solution. Resulting in the following equations:

$$v_{ij}(t+1)=\omega*v_{ij}(t)+c_{1}R_{1}(p_{ij}(t)-x_{ij}(t))+c_{2}R_{2}(p_{gj}(t)-x_{ij}(t)) \tag{6}$$

where ω represents the inertial weight factor; this quotient can either be predefined as a constant value with previous knowledge of the problem, or can be adjusted dynamically as a decreasing linear function as follows:

$$\omega(t) = \omega_{up} - \frac{(\omega_{up} - \omega_{low})}{T_{\max}} t \tag{7}$$

where $\omega(t)$ is the time-dependent inertial weight, ω_{up} is the inertial weight upper boundary, ω_{low} is the inertial weight lower boundary. The iteration counter is represented by t, $T_{\max}$ is the iteration total. In 2013 other variant for PSO, named PSO with an Aging leader and Challengers (ALC-PSO) has been proposed in [7]. The mathematical model of ALC-PSO is very similar that of the original PSO except that the velocity is adjusted by satisfying the following [7]:

$$v_{ij}(t+1) = \omega * v_{ij}(t) + c_1 R_1(p_{ij}(t) - x_{ij}(t)) + c_2 R_2(leader - x_{ij}(t)) \tag{8}$$

where *leader* is the particle holding the best solution found by the swarm at iteration t.

3.2 Particle Swarm Optimization with Biologic Aging Model

The steps involved in the proposed BAM-PSO variant algorithm are shown in Fig. 1 and are explained as follows:

Step 1: Initialization. The initial positions of all particles are generated randomly within the n-dimensional search space, with velocities initialized to 0. The best particle among the swarm is selected as the *leader*. The age of the *leader* is initialized to 0 and the lifespan of the *leader* is set to a known initial value.

Step 2: Velocity and position updating. Every particle follows the velocity update rule shown in Eq. (8) and the position update rule shown in Eq. (5) to adjust its velocity and position respectively.

Step 3: Update $p_{ij}(t)$ and the *leader* . For particle i $(i = 1, 2, \ldots, S$), if the newly generated position x_{ij} is better than p_{ij} , then x_{ij} becomes the new p_{ij} . In addition, if the best position obtained in this iteration is better than the *leader* , then the *leader* is updated to be the best position in this iteration. In this sense, this step is similar to that of the conventional PSO, but the *leader* represents the best solution generated by particles during its lifetime.

Step 4: Lifespan control. After the positions of all particles are updated, the leading power of the *leader* is evaluated and the lifespan of the particles is adjusted according to each particle performance

Step 5: Generating a challenger. A new particle is generated and is used to challenge a particle whose lifespan is exhausted.

Step 6: Evaluating the challenger. The leading power of the newly generated challenger is evaluated. If the challenger has better performance, it replaces the old particle.

Step 7: Terminal condition check. If the number of fitness evaluations (FEs) is larger than the predefined maximum evaluation number (max_eval), the algorithm terminates. Otherwise go to Step 2 for a new round of iteration.

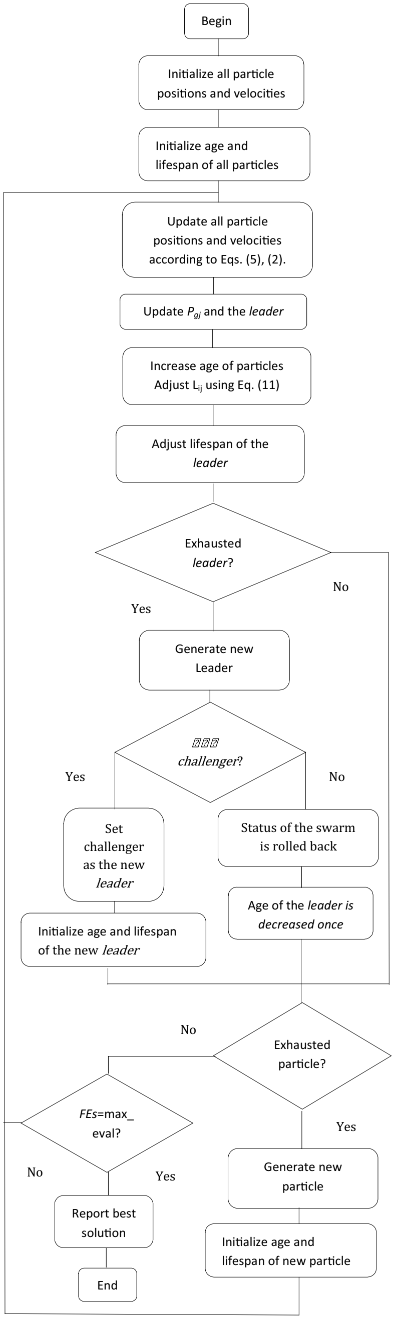
Begin
Initialize all particle positions and velocities
Initialize age and lifespan of all particles
Update all particle positions and velocities according to Eqs. (5), (2).
Update P_{gj} and the *leader*
Increase age of particles Adjust L_{ij} using Eq. (11)
Adjust lifespan of the *leader*
Exhausted *leader*?
Yes
No
Generate new Leader
□□□ *challenger*?
Yes
No
Set challenger as the new *leader*
Status of the swarm is rolled back
Initialize age and lifespan of the new *leader*
Age of the *leader is decreased once*
Exhausted particle?
No
Yes
FEs=max_ eval?
No
Yes
Generate new particle
Report best solution
Initialize age and lifespan of new particle
End

Fig. 1. Flow chart of proposed PSO algorithm

This BAM-PSO algorithm is used in this paper to train the neural network in order to perform solar radiation forecasting, as depicted in Fig. 2.

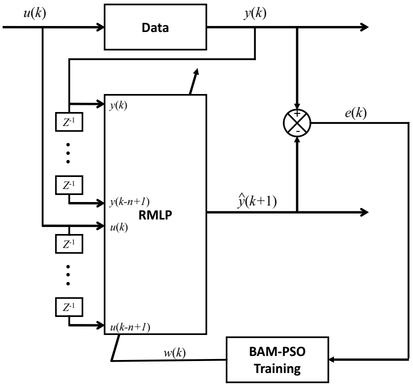

Fig. 2. RMLP trained with an BAM-PSO training algorithm

4 Experimental Results

In this section, we implement a neural network predictor for solar radiation, on the basis of a BAM-PSO training algorithm. As first stage it is necessary to determine the optimal dimension of the regression vector; once we did this, we select the number of hidden units for both hidden layers. The training is performed using hourly data from the first two weeks and the testing has been made using the subsequent data. Experimental data has been taken from a microgrid facility which includes a photovoltaic tower, as is shown in Fig. 3. The neural network used is a RMLP trained with an BAM-PSO, whose structure is as the presented in Fig. 2; the hidden layer has 12 units with logistic sigmoid activation functions (3), whose β was fixed in 1 and the output layer is composed of just one neuron, with a linear activation function. The BAM-PSO design parameters have been selected as: iterations $t = 10000$, $c1 = c2 = 2.0$, $\omega_{up} = 0.9$, $\omega_{low} = 0.4$, $swarm = 3$, and for the aging model $lifespan = 3$.

The initial values for neural weights are randomly selected. The length of the regression vector is 12 because that is the order of the system, which is determined using the Cao methodology. The training is performed off-line, using a series-parallel configuration; for this case the delayed output is taken from the solar radiation measurements. The mean square error (MSE) reached in training is 6.4793×10^{-3} in 2000 iterations the mean and standard variation for relative error are $9.3 \times 10^{-3} W/m^2$ and $7.8964 \times 10^{-4} W/m^2$.

Simulation results are presented as follows: Fig. 4 displays the solar radiation time series forecasting and Fig. 5 includes the MSE for training.

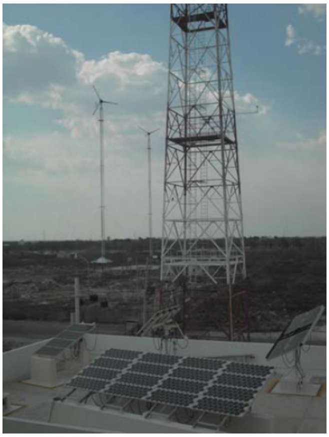

Fig. 3. Photovoltaics modules in a microgrid facility

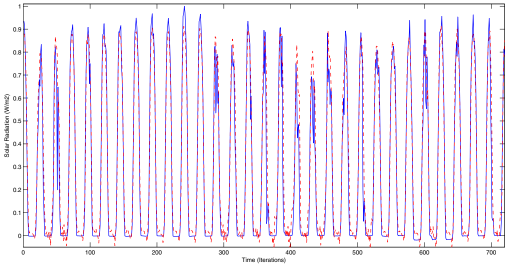

Fig. 4. Neural identification results (measured signal in blue solid line and neural forecasting signal in red dashed line)

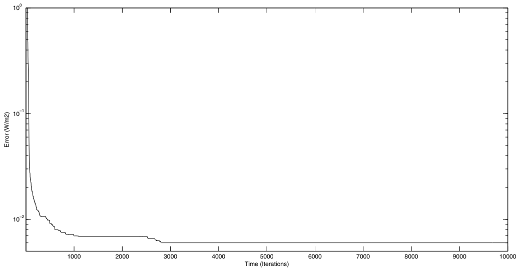

Fig. 5. Mean square error for training

5 Conclusions

This paper proposes the use of a RMLP trained with a novel BAM-PSO learning algorithm, to predict hourly solar radiation with good results as shown in results. The proposed method has a compact structure but taking into account the dynamic nature of the system which behavior is required to predict. The proposed neural identifier proves in our experiments to be a model that captures very well the complexity associated with solar radiation forecasting.

Acknowledgement. Authors thank the support of CONACYT Mexico, through Projects CB-103191Y, 106838Y, 156567Y and INFR229696.

References

1. Alanis, A.Y., Simetti, C., Ricalde, L.J., Odone, F.: A Wind Speed Neural Model with Particle Swarm Optimization Kalman Learning. In: Proceedings of the 9th International Symposium on Intelligent Automation and Control, Puerto Vallarta, Mexico (2012)
2. Cardenas-Barrera, J.L., Castillo-Guerra, E., Meng, J., Chang, L.: Multi-step-ahead, Short-Term Prediction of Wind Speed Using a Fusion Approach. In: Ruiz-Shulcloper, J., Sanniti di Baja, G. (eds.) CIARP 2013, Part II. LNCS, vol. 8259, pp. 83–91. Springer, Heidelberg (2013)
3. Landassuri-Moreno, V.M., Bustillo-Hernández, C.L., Carbajal-Hernández, J.J., Sánchez-Fernández, L.P.: Single-Step-Ahead and Multi-Step-Ahead Prediction with Evolutionary Artificial Neural Networks. In: Ruiz-Shulcloper, J., Sanniti di Baja, G. (eds.) CIARP 2013, Part I. LNCS, vol. 8258, pp. 65–72. Springer, Heidelberg (2013)
4. Alanis, A.Y., Arana-Daniel, N., Lopez-Franco, C.: Neural-PSO second order sliding mode controller for unknown discrete-time nonlinear systems. In: Proceedings of the IEEE International Joint Conference on Neural Networks, pp. 1–6 (2013)
5. Ge, S.S., Zhang, J., Lee, T.H.: Adaptive neural network control for a class of MIMO nonlinear systems with disturbances in discrete-time. IEEE Transactions on Systems, Man and Cybernetics, Part B 34(4), 1630–1645 (2004)
6. Kennedy, J., Eberhart, R.C.: Particle swarm optimization. In: Proceedings of the IEEE International Joint Conference on Neural Networks, pp. 1942–1948 (1995)
7. Wei-Neng, C., Jun, Z., Ying, L., Chen, N., Zhi-Hui, Z., Henry, S.-H.C., Yun, L., Yu-Hui, S.: Particle swarm optimization with an aging leader and challengers. IEEE Transactions on Evolutionary Computation 17(2), 241–258 (2013)
8. Norgaard, M., Poulsen, N.K., Ravn, O.: Advances in Derivative-Free State Estimation for Nonlinear Systems. Technical Report IMM-REP-1988-15 (revised edition), Technical University of Denmark (2000)
9. Alanis, A.Y., Sanchez, E.N., Loukianov, A.G.: Discrete time adaptive backstepping nonlinear control via high order neural networks. IEEE Transactions on Neural Networks 18(4), 1185–1195 (2007)
10. Cao, L.: Practical method for determining the minimum embedding dimension of a scalar time series. Physica D: Nonlinear Phenomena 110(1), 43–50 (1997)