Bulletin of Environment, Pharmacology and Life Sciences
Bull. Env. Pharmacol. Life Sci., Vol 3 (2) January 2014: 132-139

Online ISSN 2277-1808
Journal's URL:http://www.bepls.com
CODEN: BEPLAD
Global Impact Factor 0.533
Universal Impact Factor 0.9804

ORIGINAL ARTICLE

Application of Fully Recurrent (FRNN) and Radial Basis Function (RBFNN) Neural Networks for Simulating Solar Radiation

Mehrdad Naderian, Hamid Barati, Mohammad Golashahi, Reza Farshidi
Department of Computer Engineering, Dezful Branch, Islamic Azad University, Dezful, Iran
*E-mail address: Mehrdad.naderian777@gmail.com

ABSTRACT
Measured maximum air temperature, minimum air temperature, relative humidity and sunshine hours values between 1990 and 2010 for Esfahan city (latitude N 32°67', longitude E 51°67', elevation 1550.4 m), Iran, were used for the estimation of global solar radiation (GSR) in future time domain using two types of Neural Networks: Fully Recurrent (FRNN) and Radial Basis Function (RBFNN). To achieve this, monthly mean air temperature, maximum air temperature, minimum air temperature, relative humidity and sunshine hours were used as inputs to the neural networks and the GSR used as output. Measured weather data from 1990 to 2006, was used in order to train the networks while the measured data from 2007 to 2010 was used for validating the trained networks. This study confirms the abilities of the RBFNNs and FRNNs to predict solar radiation values.
Keywords: *Radial Basis Function Neural Networks (RBFNNs), Fully Recurrent Neural Networks (FRNNs), Esfahan, Solar Energy.*

Received 14/10/2013 Accepted 20/12/2013

INTRODUCTION

Due to global political uncertainty and alarmingly increasing pollution levels in air, water, and soil, renewable energy resources are increasingly utilized [1]. Solar energy is a clean, inexhaustible, and a free source of energy. This source has served humankind for many centuries. For proper and efficient utilization of solar energy, the prediction of GSR is very important [2]. Several authors have used different formulations to predict GSR.

Rehman and Mohandes, used number of day, air temperature, and relative humidity as inputs to a multi-layer perceptron (MLP) neural networks in order to estimate daily GSR for Abha city in Saudi Arabia. Results showed a mean absolute percentage error (MAPE) of 4.49% [3].

Azadeh et al. used a MLP) neural network to predict monthly GSR by climatological and meteorological variables such as monthly mean maximum temperature, minimum temperature, relative humidity, vapor pressure, wind speed, duration of sunshine and total precipitation for six cities in Iran. The results showed an average MAPE and absolute fraction of variance (R2) of 6.70% and 94%, respectively [4].

In two different works, Mohandes et al. used latitude, longitude, altitude and sunshine duration as inputs of MLP and radial basis function (RBF) neural networks in order to predict GSR for 41 stations spread over Saudi Arabia [1, 2]. The data for 31 stations were used to train the neural networks and data from the other 10 stations were used for testing. The average MAPE for the MLP network was 12.61% while this value was 10.09% for RBF network. To predict the solar energy potential for 17 cities in Turkey, Sozan et al. used different meteorological and geographical factors (latitude, longitude, altitude, month, averages of sunshine duration and mean temperature) as inputs of the neural networks. The data for 11 stations were used to train the neural networks and data from the other 6 stations were used for testing. The results showed a maximum mean absolute percentage error (MAPE) and absolute fraction of variance (R2) of 6.7% and 99.89%, respectively [5]. In a different study which was done by Sozan et al., the solar potential for 12 cities spread over Turkey, was predicted using neural networks based on same meteorological and geographical factors. The data for 9 stations were used in training the neural networks and data from the other 3 stations were used for testing. The obtained results showed a

maximum mean absolute percentage error (MAPE) and absolute fraction of variance (R2) of 6.78% and 99.78%, respectively [6].
This study applies RBFNNs and FRNNs to predict monthly GSR. Esfahan city is considered as a case study in this work.

NEURAL NETWORKS

Redial Basis Function Neural Networks (RBFNNs)
The radial basis function (RBF) network is a popular type of network that is very useful for pattern classification problems. Figure 1 shows the structure of a RBF network which consists of three layers of neurons. The input layer neurons receive the input pattern (x_1 to x_N). The hidden layer neurons provide a set of activation functions that constitute an arbitrary "basis" for the input patterns in the input space to be expanded into the hidden space by way of non-linear transformation. At the input of each hidden neuron, the distance between the centre of each activation or basis function and the input vector is calculated. Applying the basis function to this distance produces the output of the hidden neuron.
The RBF network outputs y_1 to y_p are formed by the neurons in the output layer as weighted sums of the hidden layer neuron activations [7, 8].
The basis function is generally chosen to be a standard function which is positive at its centre x=0, and then decreases uniformly to zero on either side. A common choice is the Gaussian distribution function:

$$K(x) = \mathbf{exp}\left(-\frac{x^2}{2}\right) \tag{6}$$

This function can be shifted to an arbitrary centre, $x = c$, and stretched by varying its spread σ as follows:

$$K\left(\frac{x-c}{\sigma}\right) = \mathbf{exp}\left(-\frac{(x-c)^2}{2\sigma^2}\right) \tag{7}$$

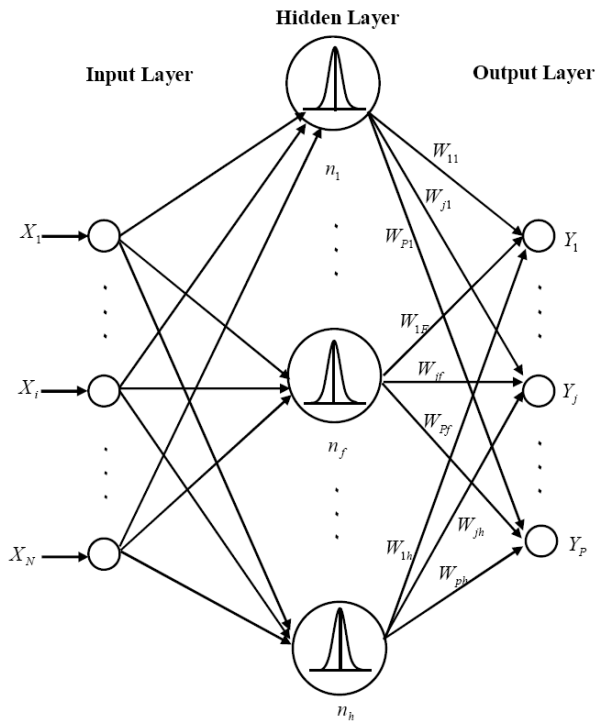

Figure.1. Topology of an RBF network.

The outputs of the RBF network y_j are given by:

$$y_j = \sum_{i=1}^{h} w_{ji} K\left(\frac{\|x - c_i\|}{\sigma_i}\right)_{\forall x} \tag{8}$$

where w_{ji} is the weight of the hidden neuron i to output j, c_i the centre of basis function i and σ_i the spread of the function. $\|x - c_i\|$ is the norm of $(x - c_i)$. There are various ways to calculate the norm. The most common is the Euclidean norm given by:

$$\|x - c_i\| = \sqrt{(x_1 - c_{i1})^2 + (x_2 - c_{i2})^2 + \cdots + (x_N - c_{iN})^2} \tag{9}$$

This norm gives the distance between the two points x and c_i in N-dimensional space. All points x that are the same radial distance from c_i give the same value of the norm. The purpose of training an RBF network is to determine the neuron weights w_{ji}, RBF centers c_i and spreads σ_i that enable the network to produce the correct outputs y_j corresponding to the input patterns x [7-13] .

RBF network training procedure

The training of an RBF network involves the minimization of an error function. The error function defines the total difference between the actual output and the desired output of the network over a set of training patterns. Training proceeds by presenting to the network a pattern of known class taken from the training set. The error component associated with that pattern is the sum of the squared differences between the desired and actual outputs of the network corresponding to the presented pattern. The procedure is repeated for all the patterns in the training set and the error components for all the patterns are summed to yield the value of the error function for an RBF network with a given set of basis function centers, spreads and neuron connection weights [14].

Standard RBF network training procedure

With the standard procedure for training RBF networks, after the number of hidden neurons (h) has been decided, the following steps will be taken:

1. Choose the RBF centers c_i; centre selection could be performed by trial and error, self-organized or supervised.
2. Choose spreads σ_i; several heuristic methods are available. A popular method is to set σ_i equal to the distance to the centre nearest to c_i.
3. Calculate neuron weights w_{ji}; when c_i and w_{ji} are known, the outputs of hidden neurons $(K_1, \cdots, K_h)^{\mathbf{T}}$ can be calculated for any pattern of inputs $x = (x_1, \cdots, x_N)$. Assuming there are s input patterns x in the training set, there will be s sets of hidden neuron outputs that can be calculated. These can be assembled into a $h \times s$ matrix:

$$K = \begin{pmatrix} k_1^1 & k_1^2 & \cdots & k_1^s \\ k_2^1 & k_2^2 & \cdots & k_2^s \\ \vdots & \vdots & \ddots & \vdots \\ k_h^1 & k_h^2 & \cdots & k_h^s \end{pmatrix}_{h \times s}$$

The output of the RBF network (y) is given by Eq.10:

$$y = K^{\mathbf{T}} w^{\mathbf{T}} \tag{10}$$

Where

$$w^{\mathbf{T}} = \begin{pmatrix} w_{11} & w_{12} & \cdots & w_{1p} \\ w_{21} & w_{22} & \cdots & w_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ w_{h1} & w_{h2} & \cdots & w_{hp} \end{pmatrix}$$

y is the matrix of actual outputs corresponding to the training inputs x. Ideally, y should be equal to d, the desired or target outputs. Unknown coefficients w_{ji} can be calculated from Eq.11 in order to minimize the sum of the squared differences between y and d.

$$w^{\mathbf{T}} = \left(K \cdot K^{\mathbf{T}}\right)^{-1} \cdot K \cdot d \qquad (11).$$

Fully Recurrent Neural Networks (FRNNs)

Fully Recurrent Neural Networks (FRNNs) are simple, but powerful computational models that can effectively learn temporal sequences, either in an on-line or an off-line fashion. A basic block diagram of an FRNN is shown in Figure 2. The FRNN consists of a linear input layer and a nonlinear output layer. The input layer is fully connected to the output layer via adjustable, weighted connections, which represent the system's training parameters. The model also features unit-gain, unit-delay feedback connections that are fed back into its input layer. FRNNs accomplish their task by learning a mapping between a set of input sequences to another set of output sequences. In particular, the model's inputs consist of input sequences, delayed output activations and a constant-valued input terminal related to a bias weight. On the other hand, the output layer generates the set of output sequences.

Typically, nodes in this layer feature a sigmoidal activation function. Furthermore, the model may include a number of hidden nodes, also known as context units, whose activations are not related to any of the outputs of the task to be learned, but act as a secondary, dynamic memory of the system. The combination of this dynamic, context-based memory with the recurrent, feedback connections is what makes the FRNN a powerful model for learning relationships between temporal sequences [15, 16].

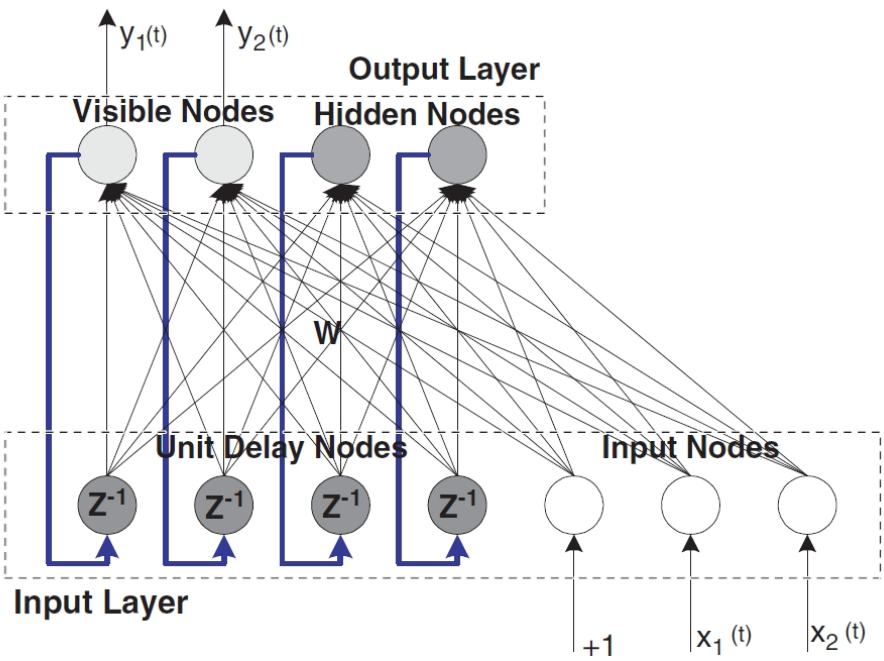

Figure 2. Block diagram of a Fully Recurrent Neural Network (FRNN).

PROBLEM DEFINITION

This paper introduces FRNNs and RBFNNs to solar radiation estimation based on month of the year, maximum air temperature, minimum air temperature, relative humidity and sunshine hours.

The related data, collected from Esfahan city between 1990 and 2006, are applied for training GSR while measured data from 2007 to 2010 are used for validating the trained networks. The validation data are not used to train the neural networks. Also, the month with inadequate data were removed from the patterns.

Figures 3 to 7 show the measured values of maximum air temperature, minimum air temperature, relative humidity, sunshine hours, and GSR between 1990 and 2010 for Esfahan city.

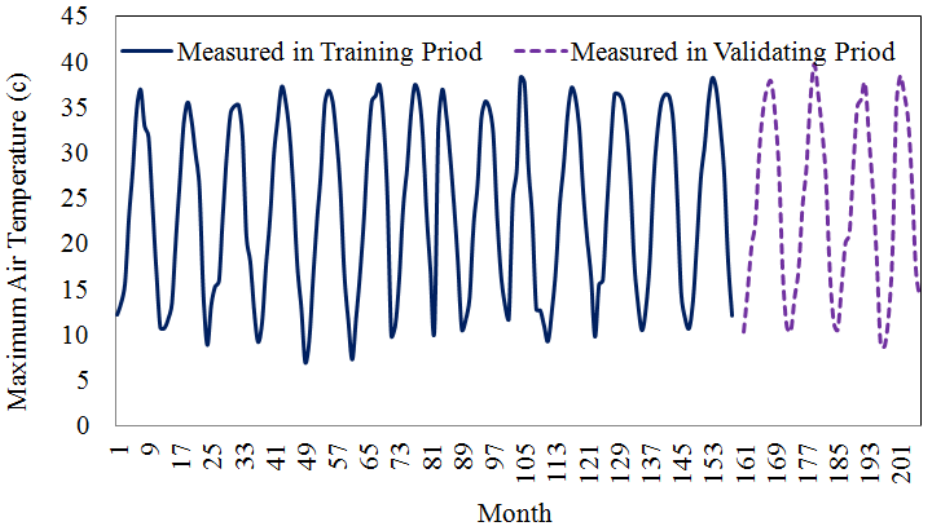

Figure 3. Values of maximum air temperature for Esfahan city between 1990 and 2010.

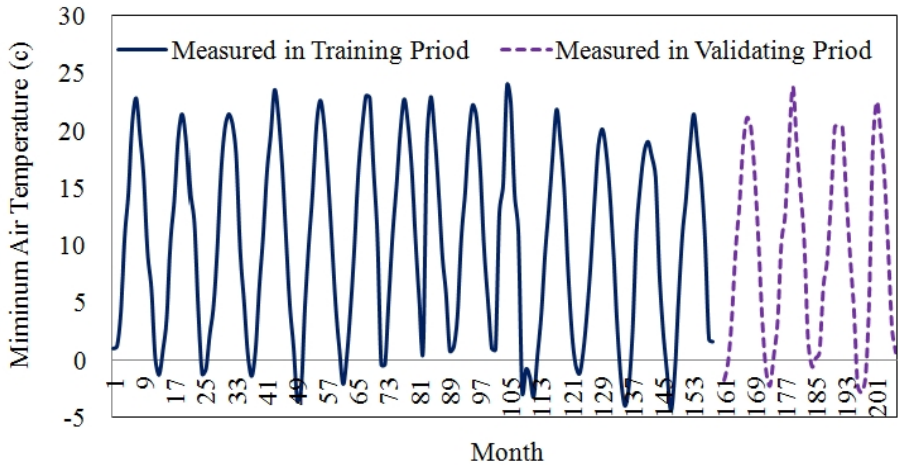

Figure 4. Values of maximum air temperature for Esfahan city between 1990 and 2010.

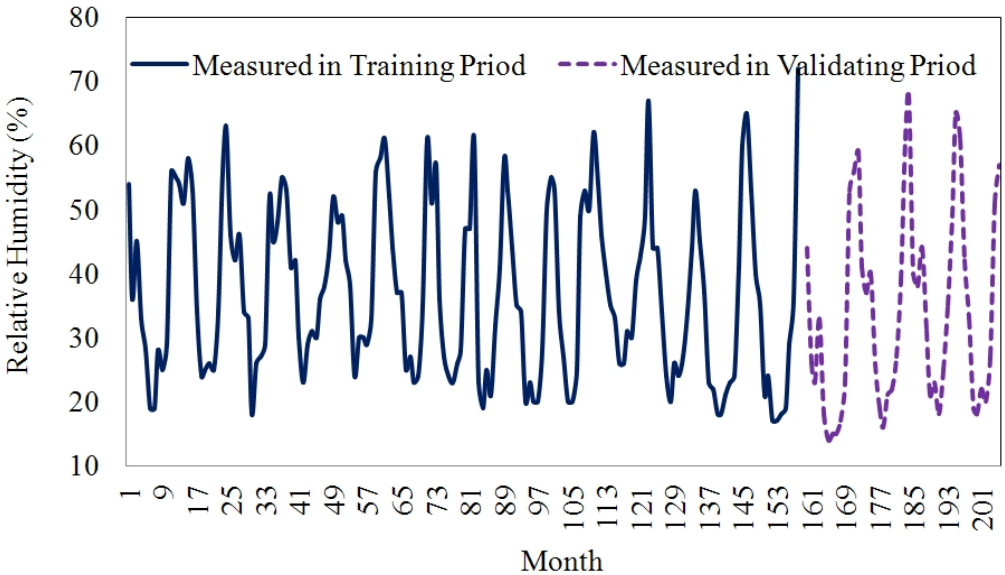

Figure 5. Values of relative humidity for Esfahan city between 1990 and 2010.

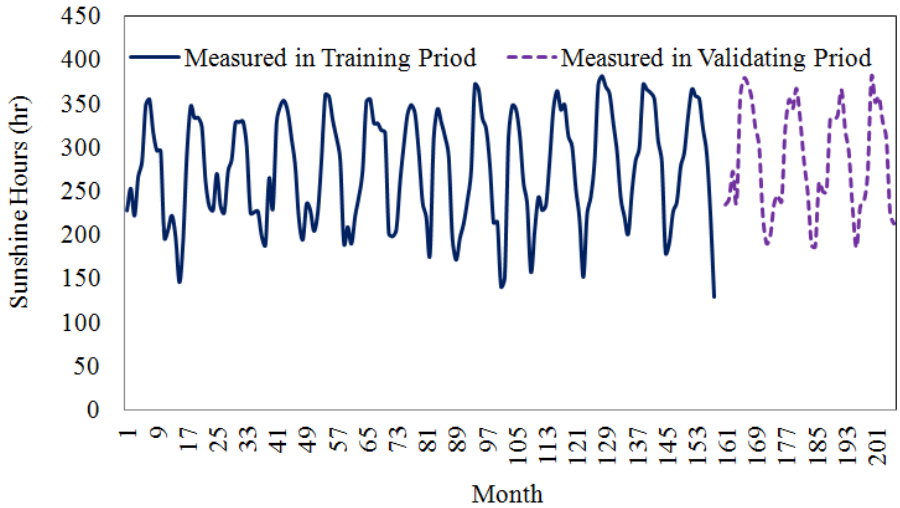

Figure 6. Values of sunshine Hours for Esfahan city between 1990 and 2010.

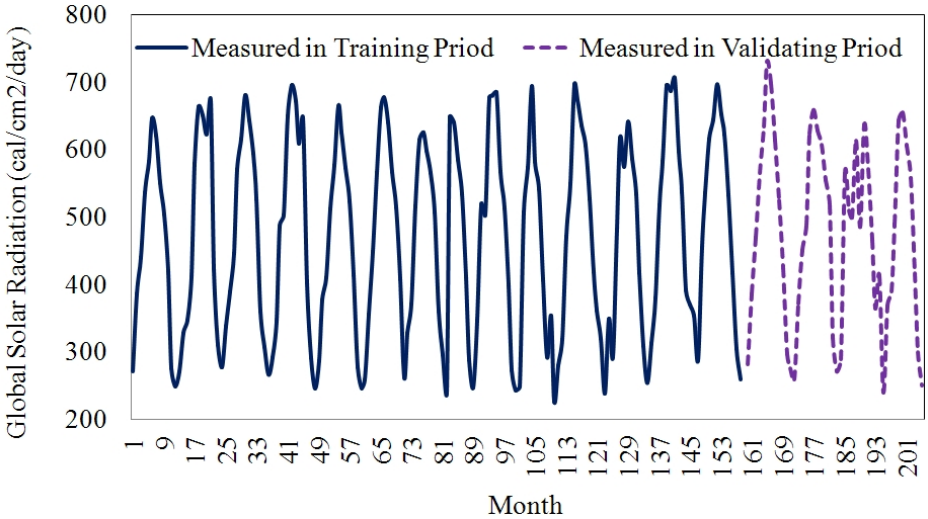

Figure 7. Values of GSR for Esfahan city between 1990 and 2010.

RESULTS AND DISCUSSION

Neural Network Toolbox of MATLAB 2010 software was used for developing the FRNNs and RBFNNs in this study. All parameters were normalized in the (0.1, 0.9) range. In order to determine the optimal network structure, various network architectures were designed and the number of neuron in the hidden layer was changed. All models given in Table 1 were trained and tested in order to compare the performances of developed models in this study.

Table1. Structure of some designed networks, training and testing errors.

Model	Neurons in hidden layer	Training	Validating	
		R^2 (%)	MAPE (%)	R^2 (%)
FRNN-1	10	92.23	10.13	90.03
FRNN-2	12	92.87	9.23	90.27
FRNN-3	14	92.91	9.19	90.31
FRNN-4	16	93.01	9.11	90.43
FRNN-5	18	93.00	9.13	90.54
RBFNN-1	10	93.03	9.09	90.67
RBFNN-2	**12**	**93.05**	**8.95**	**90.81**
RBFNN-3	14	92.98	9.13	90.69
RBFNN-4	16	92.92	9.18	90.58
RBFNN-5	18	92.90	9.21	90.36

All models gave the absolute fractions of variance (R2) better than 90% but RBFNN-02 was the best

designed network with an absolute fraction of variance (R^2) of 90.81% and mean absolute percentage error (MAPE) of 8.95%, respectively. Fig. 5 shows the measured and predicted values of GSR for the best designed network.

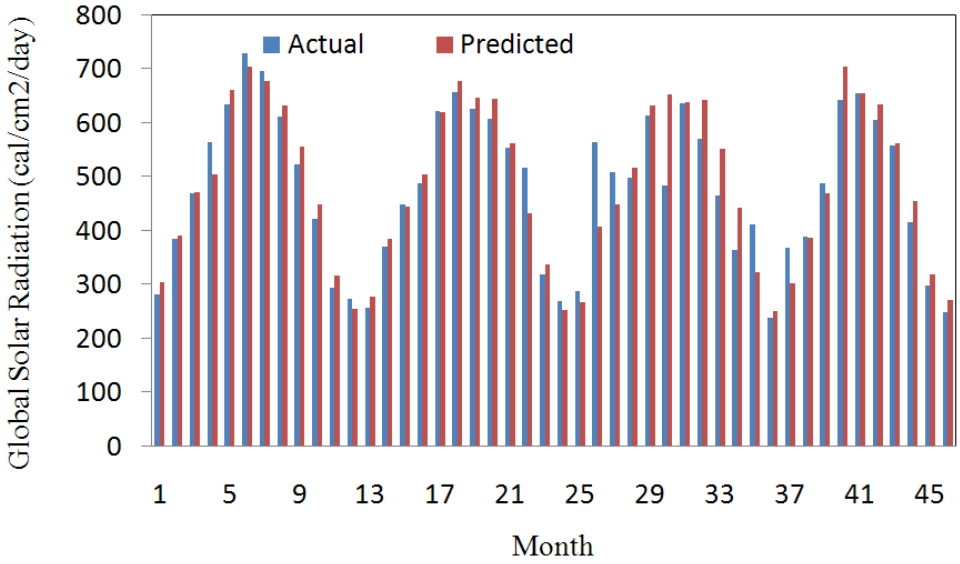

Figure 8. Comparison between predicted GSR values based on the best model (RBFNNs-2) and measured values on testing data (2007 and 2010) for Esfahan.

CONCLUSION

FRNNs and RBFNNs were applied to estimate the monthly mean daily GSR on horizontal surface for Esfahan city using measured values of maximum and minimum air temperature, relative humidity, and sunshine hours. The measured data between 1990 and 2006 were used for training while the data from 2007 to 2010 were used for validating the trained networks. Several network architectures were designed and the number of neuron and hidden layer were changed in order to find the best network structure. A one hidden layer network with 12 neurons was found to be the best designed network which had an absolute fraction of variance (R^2) of 90.81% and mean absolute percentage error (MAPE) of 8.95%.

Future work is focused on comparing the methods presented here with other available types of Neural Networks. Predicting of global solar radiation can also be investigated with Multi Layer Perceptron (MLP) neural networks, Support Vector Machines (SVM) and etc. The results of the different methods can be compared with the presented method.

ACKNOWLEDGMENT

The authors are grateful for the financial support provided for the present work by Islamic Azad University of Dezful and Esfahan Meteorological Office.

REFERENCES

1. Mohandes, M.A., Halawani, T.O., Rehman, S., Hussain, A.A., (2004). Support vector machines for wind speed prediction, Renewable Energy 29 939–947.
2. Mohandes, M., Rehman, S., Halawani, T.O. (1998). Estimation of global solar radiation using artificial neural networks, Renewable Energy 14 (1–4) 179–184.
3. Rehman, S., Mohandes, M. (2008). Artificial neural network estimation of global solar radiation using air temperature and relative humidity, Energy Policy 63 571–576.
4. Azadeh, A., Maghsoudi, A., and Sohrabkhani, S. (2009). An integrated artificial neural networks approach for predicting global radiation. Energy Conversion and Management doi: 10.1016/j.enconman.2009.02.019.
5. Sozen, A., Arcaklioglu, E., Ozalp, M., Kanit, E.G., (2004). Use of artificial neural networks for mapping of solar potential in Turkey, Applied Energy 77 273–286.
6. Sozen, A., Arcaklioglu, E., Ozalp, Caglar, N., (2005). Forecasting based on neural network approach of solar potential in Turkey, Renewable Energy 30 1075–1090.
7. D.T. Pham and X. Liu, (1995).“Neural Networks for identification, prediction and control,” London: *Springer verlag*. 1995.

8. Behrang, M.A., Assareh, E., Ghanbarzadeh, A., and Noghrehabadi, A.R. (2010). The potential of different Artificial Neural Network (ANN) techniques in daily global solar radiation modeling based on meteorological data. Solar Energy 84 1468–1480.
9. Behrang, M.A., Assareh, E., Assari, M.R., Ghanbarzadeh, A., (2011). Using Bees Algorithm and Artificial Neural Network to Forecast World Carbon Dioxide Emission. Energy Sources, Part A: Recovery, Utilization, and Environmental Effects 33 1747–1759.
10. Ghanbarzadeh, A., and Noghrehabadi, Behrang, M.A., Assareh, E., (2009). Wind speed prediction based on simple meteorological data using artificial neural network. 7th IEEE International Conference on Industrial Informatics (INDIN 2009). pp 664-667.
11. Ghanbarzadeh, A., and Noghrehabadi, Assareh, E., Behrang, M.A., (2009). Solar radiation forecasting based on meteorological data using artificial neural networks. 7th IEEE International Conference on Industrial Informatics (INDIN 2009). pp 227-231.
12. Hassan B., H. Haroonabadi, R. Zadehali., (2013a).The Application of a Hybrid Artificial Neural Network and Genetic Algorithm (ANN-GA) to Mapping of Wind Speed Profile for Electrical Energy applications: A Case study for South Coasts of Iran. Bull. Env. Pharmacol. Life Sci., 2 (6) May 2013: 141-147.
13. Hassan B, H. Haroonabadi, R. Zadehali., (2013b). Wind speed forecasting in South Coasts of Iran: An Application of Artificial Neural Networks (ANNs) for Electricity Generation using Renewable Energy. Bull. Env. Pharmacol. Life Sci,. 2 (6): 30-37.
14. Bishop, C.M., 1995. Neural Networks for Pattern Recognition, Clarendon Press, Oxford.
15. Williams, R.J., Zisper, D, (1989). A learning algorithm for continually running fully recurrent neural networks, Neural Computing 1270–280.
16. Atiya, A.F., Parlos, A.G., (2000). New results on recurrent network training: unifying the algorithms and accelerating convergence, IEEE Trans. Neural Networks 11 (3) 697–709.

How to cite this article:
Mehrdad N, Hamid B, Mohammad G, Reza F. Application of Fully Recurrent (FRNN) and Radial Basis Function (RBFNN) Neural Networks for Simulating Solar Radiation . Bull. Env. Pharmacol. Life Sci. 3 (2) 2014: 132-139