Neural network based models for estimating the temperature and humidity under greenhouse

M. Outanoute¹, A. Lachhab²*, A. Selmani¹, H. Oubehar¹, M. Guerbaoui¹, A. Ed-dahhak³, B. Bouchikhi³

¹Sensors Electronic & Instrumentation Group, Physics Department, Moulay Ismail University, Faculty of Sciences, B.P. 11201, Zitoune, 50003, Meknes, Morocco.
²Modelling, Systems Control and Telecommunications Team, Department of Electrical Engineering, Moulay Ismail University, High School of Technology, BP. 3103, Toulal, Meknes, Morocco.

*Corresponding author: abd_lachhab@yahoo.fr

ABSTRACT

This study investigates the utility of Artificial Neural Networks (ANNs) for the estimation of the temperature and the relative humidity, under an experimental greenhouse, as the prime climatic variables for the development and growth of the crop. In this study, we suggested three different neural network training algorithms which are the gradient descent with momentum and adaptive learning rate, Broyden-Fletcher-Golfarb-Shanno (BFGS) quasi-newton back-propagation and resilient back-propagation algorithm. The prediction ability of the developed models was compared in terms of Mean-Squared Error (MSE) and coefficient of determination (R²). The obtained results showed that the ANN model formed with 5 neurons on a single hidden layer and trained by using the BFGS algorithm has shown very good correlation between the actual data and the predicted values, with a high coefficient of determination and mean squared error establishing that the developed model was able to estimate the behavior of the temperature and relative humidity under the experimental greenhouse with good performance.

Keywords: Modeling, greenhouse climate, artificial neural networks, identification, black box model.

1- INTRODUCTION

A greenhouse is a closed environment where some climate variables can be manipulated to control the development and growth of the crop [Li et al., Ahmad et al.]. A greenhouse is seen as a multivariable process presents a nonlinear nature and is influenced by biological processes. This system is characterized by a complex nonlinear dynamics due to the time varying of the climatic parameters. The internal climate is strongly dependent on the outside conditions, existence of high degree of correlation among variables; and the dependency on the greenhouse characteristics and geographical area [Rodríguez et al.]. For that, modeling the dynamic behaviours of climatic parameters are important tools, helping to predict the climate conditions inside greenhouses and also enabling the use of automatic control systems, which are the two main objectives of greenhouse climate modeling [Baptista et al.]. The greenhouse environmental control involves the field of control technology as a way to optimize inside greenhouse climate based on measured variables and the action that is applied on greenhouse equipment. Proper controlled strategies enhance the effectiveness of greenhouse climate for growing plants and the use of computers for decision has improved the control of greenhouse climate. Whenever, due to the complexity of the nonlinear dynamics, more accurate models were required in controller design [Fourati].

Artificial Neural Networks (ANN) are designed to model the working of human neural systems through computerized algorithm [He]. They are capable of making a parallel computing and distributing the stored information like human brain. The learning ability of the ANNs makes them suitable for detecting unknown and non-linear functions. They proved that they are a powerful computational tool to solve several types of problems in different fields where approximation of nonlinear functions, identification, control, optimization, classification, and pattern recognition are required [Hurtado et al.].
Due to the complexity of the greenhouse system, some importance have been taken into account by implementing artificial intelligence techniques that offer the advantage of modeling nonlinear relationships in greenhouse modeling task [Boselin Prabhu et al.].

In this work we proposed a system in order to estimate and predict the temperature and relative humidity, in the greenhouse process, based on Neural Networks (NN) methodologies.

2- MATERIALS AND METHODS

2.1- EXPERIMENTAL SET-UP

The models were trained using data collected from an experimental greenhouse located at the Laboratory of Electronic, Automatic and Biotechnology of the Faculty of Sciences, Meknes. Fig. 1 presents a schematic view of the experimental greenhouse system. The establishment of the greenhouse was made in order to develop an integrated data acquisition system to control the inside climate. This experimental greenhouse has been equipped with many sensors and its conditioning circuits allow us to measure the different climatic parameters inside and outside the greenhouse. The acquisition and data processing has been developed by a multi-function NI-6024E card from National Instrument, to control and manage complex measurements. In order to improve the climate under greenhouse, we have installed a heating system and a variable speed ventilator (Ed-Dahhak et al.; Guerbaoui et al.; El Afou et al.).

Furthermore, the prototype is equipped with a climate controlling and monitoring system developed by using Matlab/Simulink and LabVIEW software, respectively. The monitoring system generates a historical database of both outside and inside climate variables in real-time from the measured data. This system has the capability to serve as basis for research and testing of various advanced control strategies.

2.2- NEURAL NETWORK ARCHITECTURE AND TRAINING ALGORITHMS

Neural networks are mathematical representations of biological neurons in the way they process information as parallel computing units. They are characterized by their architecture, training or learning algorithm and activation function. The architecture describes the connections between the neurons. Neurons or nodes in an ANN are arranged into layers as shown in Fig. 2. The first layer that interacts with the environment to receive input is known as the input layer. The final layer that interacts with the output to present the processed data is known as output layer. Layers between the input and the output layer that do not have any interaction with the environment are known as hidden layers. Each layer represents a non-linear combination of non-linear functions from the previous layer. Each neuron is a multiple-input multiple-output (MIMO) system that receives signals from the inputs, produces a resultant signal, and transmits that signal to all outputs.
Two commonly used neuron activation functions for the neuron are sigmoidal and purlin functions. Both functions are continuously differentiable everywhere, and typically they have the following mathematical forms:

\[
\text{Tansig function: } f(x) = \frac{2}{1 + \exp(-2x)} - 1 \tag{1}
\]

\[
\text{Purlin function: } f(x) = x \tag{2}
\]

During training both the process and ANN receive the same input and then their outputs are compared. The error signal, which is computed between measured output and the estimated output of the ANN, is used to update the weights in the ANN. So, \( E \) is a cost function that indicates the performance of network learning (Eq. 3):

\[
E(w) = \frac{1}{2N} \sum_{i=1}^{N} \sum_{m=1}^{M} \left[ y_{m,i}(k) - y_{p,i}(k) \right]^2 \tag{3}
\]

\( N \) is the number of patterns, \( M \) is the number of output neurons and \( w \) is the weight vector which is the parameter vector that minimizes \( E \).

The learning takes place minimizing this value through the back-propagation algorithm. The minimization of the error is obtained using the gradient of the cost function, which consists of the first derivative of the function with respect all the weights \( w_k \) (Eq. 4):

\[
g_k = \nabla E \tag{4}
\]

\[
\text{On the basis of this gradient the weights will be updated going in the opposite direction of the partial derivatives until a local minimum is reached, with the following mechanism}: \]

\[
w_{k+1} = w_k - \eta g_k \tag{5}
\]

Where \( w_k \) denotes the weight matrix at epoch \( k \) and the positive parameter \( \eta \) is called the learning rate.

**Gradient descent with momentum and adaptive learning rate algorithm**

The gradient descent with momentum and adaptive learning rate algorithm (GDX) combines adaptive learning rate with momentum training. It is similar to gradient descent algorithm except that it has a momentum coefficient as an additional training parameter [Ding et al.]. Typically, the new weight vector \( w_{k+1} \) is defined as:

\[
W_{k+1} = W_k - \eta g_k + \alpha(W_k - W_{k-1}) \tag{6}
\]

\[
\alpha_{k+1} = \beta \alpha_k \tag{7}
\]

Where \( \beta \) is equal to 0.7 if the new error is greater than 1.04 (old error), \( \beta \) is equal to 1.05 if the new error is less than 1.04 (old error), \( \alpha \) is called momentum coefficient which is a positive number, and \( g_k \) is the gradient.
BFGS Quasi-Newton Backpropagation Algorithm

Newton’s method is an alternative to the conjugate gradient methods for fast optimization. The BFGS algorithm is one of the most popular quasi-Newton algorithms. The basic step of Newton’s method is to form the Hessian Matrix. For smaller networks, BFGS can be an efficient training function. Newton’s method often converges faster than conjugate gradient methods [Ramesh et al.]. The weight update for the Newton’s method is (Eq. 8):

\[ W_{k+1} = W_k + A_k^{-1} g_k \]  

Where \( A_k \) is the Hessian matrix of the cost function at the current values of the weights and biases.

Resilient Backpropagation algorithm

To overcome the inherent disadvantages of pure gradient-descent, the resilient back propagation training algorithm (RPROP) performs a local adaptation of the weight updates according to the behaviour of the error function. RPROP algorithm is generally much faster than the standard steepest descent algorithm. In contrast to other adaptive techniques, the effect of the RPROP adaptation process is not affected by the magnitude of the partial derivatives, but only their sign is used to update the weights. [Kotsialos et al., Chabaa et al.]. The weights are adjusted by the following rule (Eq. 9) [Gunther et al.]:

\[ w_{k+1} = w_k - \eta_k \text{sign} \left( \frac{\partial E}{\partial w_k} \right) \]  

In order to compare the estimation ability of the developed models, Mean Squared Error (MSE) (Eq. 10), and coefficient of determination \( R^2 \) (Eq. 11) were computed in the training and testing phases for all models. MALTAB software was used for performing the experiments

\[ MSE = \frac{1}{N} \sum_{i=1}^{N} \left( \frac{y_m(k) - y_p(k)}{y_m(k)} \right)^2 \]  

\[ R^2 = \left( 1 - \frac{\sum_{i=1}^{N} (y_m - \bar{y})^2}{\sum_{i=1}^{N} (y_m - \bar{y})^2} \right) * 100\% \]  

Where \( y_m \) is the measured output, \( y_p \) is the predicted output, \( \bar{y} \) is the mean value and \( N \) is the number of pattern.

The purpose of modeling in the considered case was to obtain a very accurate dynamic model able to estimate the two outputs internal temperature and relative humidity of greenhouse system.

3- RESULTS AND DISCUSSIONS

In order to elaborate a neural network model that describe the evolution of internal temperature and relative humidity, a data measurements have been taken in a period of three days (25720 data values), derived from measured input/output data (database) of the considered greenhouse system working in open-loop experiments during 72 hours with a sampling time of 10 seconds, were used. The data set reserved for training and testing after a pre-processing was processed with all the algorithms considered for the purposes of comparison.

Figure 3 shows the evolution of internal and external climates in the considered days. The multi-step of voltages sent to the heater and the ventilator changes are described in Fig.3 (a) and (b), respectively. The outside temperature and relative humidity curves were shown in Fig. 3 (c) and (d), respectively. Fig. 3 (e) and (f) show the evolution of the inside temperature and inside relative humidity, respectively.
Figure 3: Input/output data used for training and testing the model: (a) Command heater, (b) Command ventilator, (c) External temperature, (d) External relative humidity, (e) Internal temperature and (f) Internal relative humidity.

The normalization of data, which can be transformed from their natural range to the network’s operating range in order to improve the performance of the process of training [Ganesan et al.], is a crucial initial step in a black box modeling based on neural network. For that, before applying the ANN to data, the training input and output values were normalized by using the equation (Eq. 12) [Ezzine et al.]:

\[
\tilde{x}_i = 0.01 + 0.99 \frac{x_i - x_{\min}}{x_{\max} - x_{\min}}
\]  

(12)

Where \( \tilde{x}_i \) is the normalized value, \( x_i \) is the original data and \( x_{\max} \) and \( x_{\min} \) are the minimum and maximum values of the variable \( x \).

The normalized data for the two first days were included in the training, testing, and validation process. The initial datasets were divided so that 70 % (18144 samples) of the database were assigned to the training set, 15 % (3888 samples) to the validation set, and 15 % to the test set. The generalization in each of the networks was improved by implementing the early stopping method through the validation set. After training, by using the three proposed algorithms to reach a threshold mean squared error of 0.001 as a satisfactory performance and independent dataset reserved for testing was used to evaluate the prediction performance of the NN algorithms that are used for accomplishing the process of training.

As depicted in Fig. 4, the values and the previous value of the external temperature, the external relative humidity, command of heater and ventilator, and the previous values of internal temperature and internal relative humidity were defined as inputs whereas the current internal temperature and the internal relative humidity were considered as the outputs.
Figure 4: Neural network structure for identifying dynamic responses of the internal temperature and relative humidity.

The elaborated ANN model was trained by using the three training algorithms, GDX, BFGS, RPROP, and the training data set. The leaning and momentum rates were taken as 0.35 and 0.9, respectively. Inputs and outputs have been normalized. After training was over, the optimum topology of the network models were determined using trial-error method. A single hidden layer composed with seven nodes was found for GDX, five nodes for BFGS and twelve nodes for RPROP as the optimum number of the hidden layer. Logistic sigmoid transfer function has been used in hidden layer while purelinear transfer function has been used in the two output neurons (output layer). The determined weight vectors were saved and used in order to test every neural model performance on test data.

Table 1 shows comparisons of the results of network training done by using the GDX, BFGS and RPROP algorithms after the 133, 140 and 235 iterations respectively.

<table>
<thead>
<tr>
<th>Training Algorithm</th>
<th>Training performance</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Nbr. of nodes</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>GDX</td>
<td>7</td>
</tr>
<tr>
<td>BFGS</td>
<td>5</td>
</tr>
<tr>
<td>RProp</td>
<td>12</td>
</tr>
</tbody>
</table>

It is obviously seen from the Table I that the BFGS has 5 hidden nodes, this number is small than the number of hidden nodes used for the training process using the GDX and the RProp algorithms. The BFGS has also better training performances (MSE and R²) for both the NN model outputs, internal temperature (T_int) and internal relative humidity (H_int), than the others algorithms. It can be said that the BFGS is better in function approximation than the other two algorithms used in this study.

The testing results of the networks, which their training results are previously given in Table I, are represented in Table 2.
According to the test results in Table 2, the BFGS algorithm gave a $R^2$ coefficient, for both internal temperature and relative humidity, higher than the values obtained by using GDX and RProp algorithms. However, the MSE error is lower than the values observed in the case of GDX and RProp.

By using the BFGS Quasi-Newton Back-propagation algorithm, the neural network, which has a good performance in the test in terms of the obtained mean squared error and coefficient of determination, that prove its generalization capability. It seems that the BFGS algorithm is more robust than the other used algorithms in the estimation of the internal temperature and relative humidity under the considered experimental greenhouse.

To evaluate the quality of the estimated models, an evaluation procedure has been applied to the remaining experimental data which were not used to build the model. The independent test data set of the measured and predicted parameters are shown in the Fig. 5.

![Figure 5: Curves fitting of the measured and predicted temperature and relative humidity in testing phase.](image-url)
As depicted in the Fig. 5, the simulation results indicate that the two predicted outputs (Internal temperature and relative humidity) of the three elaborated neuronal models are close to the measured values. To get these results, we use the three neural network training algorithms with respect to the obtained prediction performance criteria with the testing datasets which are summarized in Table II. Although, the NN-based models trained with Quasi-Newton Back-propagation learning algorithm predict more successfully the internal temperature and relative humidity than the training GDx and RProp algorithms.

4- CONCLUSION

Due to the complexity of the greenhouse system, some importance has been put into consideration in order to implement neural networks based on identification technique. This latter offers an attractive alternative for modeling nonlinear relationships which exist in greenhouse system. This work demonstrates the applicability of the neural networks to model the internal air temperature and relative humidity under greenhouse. Concerning the prediction of the temperature and relative humidity within an experimental greenhouse, the obtained results indicate that the use of the BFGS Quasi-Newton Back-propagation is more suitable than the use of both the gradient descent with momentum and adaptive learning rate algorithm, and Resilient Back-propagation-based neural networks. Based on the computed performance criteria, Mean Squared Error (MSE) and the coefficient of determination R² in both stage of training and testing, BFGS Quasi-Newton Back-propagation algorithm was found to be the most robust among the three training algorithms which were tested in this examined case. As future works, the developed neural network mimics a dynamic model of the greenhouse climate parameters and thereafter can be used for control purposes. It is planned to implement the controller in a real time control of the greenhouse climate in order to verify the results obtained in the simulated study.

REFERENCES:


