Volume 4, Issue 8, August 2014 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Determination of Solar Cell Parameters using Neural Network Trained by Steepest Descent Algorithm

Fayrouz Dkhichi*, Benyounes Oukarfi
Electrical Engineering Department, Faculty of Sciences and Techniques,
Hassan II University, Mohammedia, Morocco

Abstract*—This paper presents the use of steepest descent algorithm in the training of artificial neural network in order to determine the internal electrical parameters of solar cell. The prediction of the values of these parameters is made for various values of temperature and irradiance. The training of the network requires the use of optimal step size value at each iteration of steepest descent algorithm. The search for good value of the step size is done by the golden section method. The steepest descent algorithm shows through this study its ability to predict the parameters of double nonlinearity model of solar cell.

***Keywords*— Artificial neural network, training, steepest descent algorithm, electrical parameters of solar cell.**

I. INTRODUCTION

The study of internal characteristics of solar cell attracts a wide attention by the researchers. Indeed, the knowledge of the internal behaviour allows insuring a better optimization of the photovoltaic energy. The efficiency of this energy is affected by the degradation of the structural internal electrical parameters of solar cell. In order to study the evolution of the values of these parameters, it is necessary to determine them according to various values of temperature and irradiance. These meteorological factors characterize the operating condition of solar cell. The model which seems the best adapted for this mission is the Artificial Neural Network (ANN).

Due to their abilities of parallel functioning and prediction from the information acquired by the experiment, the ANN presents an efficient tool to identify even complex and nonlinear systems, as the case of solar cell model. The training of the ANN is made by an algorithm, which leads to the adjustment of connections values “weight” between neurons, presenting the information acquired by the ANN.

In this study we determine the internal electrical parameters of solar cell by using the ANN trained by the optimization algorithm “steepest descent”. The regulation of the step size is made by the golden section method.

II. SOLAR CELL UNDER ILLUMINATION

The solar cell is modelled under the shape of an equivalent electrical circuit [1, 2] schematized as follows:

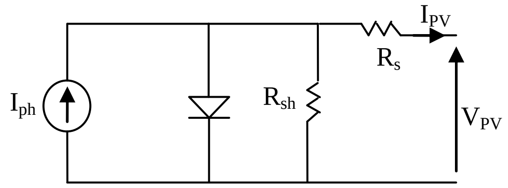

Fig. 1: The equivalent electrical circuit of solar cell

The mathematical model [3] deducted from this circuit, shows the relationship between the current I_{pv} and the voltage V_{pv} according to the five electrical parameters R_s, R_{sh}, I_{ph}, I_s, and n.

$$I_{PV} = I_{ph} - I_s\left[\exp\left(\frac{V_{PV} + R_s I_{PV}}{n\, V_{th}}\right) - 1\right] - \frac{V_{PV} + R_s I_{PV}}{R_{sh}} \tag{1}$$

Where:
R_s: Series resistance presenting the losses due to various contacts and connections.
R_{sh}: Shunt resistance characterizing the leak currents of the diode junction.
I_{ph}: Photocurrent depending on both illumination and temperature.
I_s: Diode saturation current.
n: Diode ideality factor.
V_{th}: Thermal voltage (V_{th}=A.T/q).
A: Boltzmann constant (A=1.3806503. 10^{-23} J/K).
q: Charge of electron (q=1.60217646. 10^{-19} C).
T: Temperature of solar cell by Kelvin.

When a cell is under illumination, the temperature and the irradiance of its environment, affects the values of its internal electrical parameters R_s, R_{sh}, I_{ph}, I_s, and n (Fig. 2). According to the five values, the solar cell generates a characteristic $I_{PV}=f(V_{PV})$ by varying a load R, for a definite value of temperature

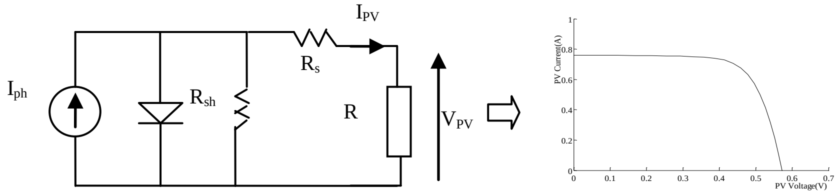

Fig. 2: Plan of variation of the voltage V_{PV} according to the current I_{PV}

Photocurrent I_{ph} varies with the irradiance, while the current of saturation I_s varies with the temperature. The rest of the parameters R_s, R_{sh} and n change their values according to temperature and also to irradiance [4].

III. THE USED ARTIFICIAL NEURAL NETWORK

A. Structure of ANN

According to the universal approximation property [5, 6] of the neural network, a single hidden layer is rather sufficient to insure the training of the ANN. The optimal topology of ANN [4] used in our study is the following:
Entrance layer: Includes two inputs which are the temperature T (°C) and the irradiance G (W/m ²).
Hidden layer: The number of hidden neurons is fixed to twenty neurons.
Output layer: This layer contains five neurons corresponding to the five electrical parameters R_s, R_{sh}, I_{ph}, I_s and n.

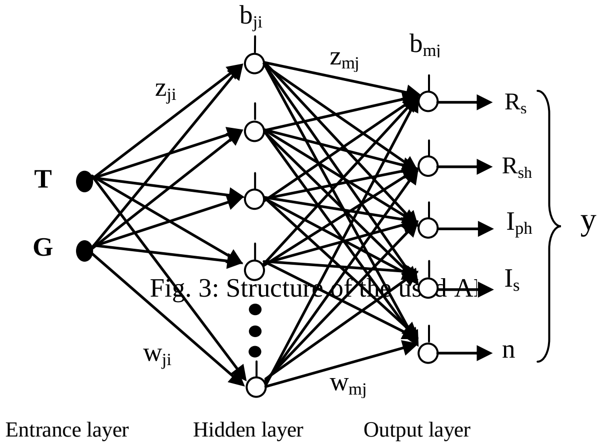

Fig. 3: Structure of the used A

Where:
i =1, 2: Index of inputs.
j=1, 2, and 20: Index of hidden neurons.
m=1, 2, and 5: Index of neurons of the output.
w_{ji}: Weights of connections between the entrance layer and the hidden layer
w_{mj}: Weights of connections between the hidden layer and the output layer.
b_{ij}: Biais of the hidden neurons.
b_{mj}: Biais of the output neurons.
z_{ji}: Input of the hidden neurons.
z_{mj}: Input of the output neurons.
y_{ji}: Output of the hidden neurons.

Activation function of the hidden neurons is the derivable sigmoid function "hyperbolic tangent":

$$f_j^h(x) = \frac{2}{1+\exp^{-2x}} -1 \qquad (2)$$

The function of the neurons of the output is the "linear" function:

$$f_m^o(x) = x \qquad (3)$$

B. Training data of artificial neural network

Training of the ANN requires a set of examples of target input-output $\{E_{(t)}, S_{(t)}\}$. These samples include various values of the target output $S_{(t)} = \{R_{s(t)}, R_{sh(t)}, I_{ph(t)}, I_{s(t)}$, et $n_{(t)}\}$ corresponding to the various values of input $E_{(t)}=\{T_{(t)}, G_{(t)}\}$, the index (t) indicates the number of the example.

Values of T are in the margin [+10°C, +70°C] and that of G are in the margin [100W/m², 1000W/m²]. Due to the activation function of the hidden layer which is the hyperbolic tangent transfer function, all the examples $\{E_{(t)}, S_{(t)}\}$

are standardized between [-1, 1]. All the samples are divided in three sets: learning $\{E, S\}_{learning}$, validation $\{E, S\}_{validation}$ and test $\{E, S\}_{test}$. The stage of validation consists in interrupting the process of learning in order to avoid the saturation of the network on learning stage.

IV. STEEPEST DESCENT ALGORITHM

A. Adjustment of network's weights

Training of ANN consists to make the calculated output "y" and the target output "S" compatible by minimizing the mean squared error at learning stage $F_{mean(learning)}$, which is expressed as follows:

$$F_{mean(learning)} = \frac{1}{p}\sum_{t=1}^{p}\sum_{s=1}^{v}\left[S(t,s) - y(t,s)\right]^2 \tag{4}$$

Where:

p: Number of examples $\{E, S\}_{learning}$

v: Number of the network outputs.

y: Values of the network outputs, $y = [R_s, R_{sh}, I_{ph}, I_s, n]$.

This minimization is made by adjusting all the weights and the bias of the network by the steepest descent algorithm according to the following equation [7]:

$$W_{k+1} = W_k - \alpha \frac{dF_{mean(learning)}}{dW} \tag{5}$$

Where:

W: All the weights of the network, $W = [w_{ji}, b_{ji}, w_{mj}, b_{mj}]$.

Expression $\frac{F_{mean(learning)}}{dW}$ is obtained by a retropropagation calculation such as:

$$\frac{F_{mean(learning)}}{dW} = \left| \frac{F_{mean(learning)}}{dw_{ji}} \quad \frac{F_{mean(learning)}}{dw_{mj}} \right| \tag{6}$$

Where:

$$\frac{F_{mean(learning)}}{dw_{ji}} = -(S - y) f_m^o w_{mj} f_j^h x_i \tag{7}$$

$$\frac{F_{mean(learning)}}{dw_{mj}} = -(S - y) f_m^o y_{mj} \tag{8}$$

B. Regulation of the step size □

Step size □ present an important coeffcient that lead the convergence of the steepest descent algorithm, such as it insures the minimization of the mean squared error $F_{mean(learning)}$ at every iteration. It exists various regulation methods of the coefficient□□, which we chose the golden section method (Zuo et al).

V. RESULTS

Initial values of the weights W are arbitrarily chosen in the margin [-1, 1]. The training of the network is made with 130 examples {E, S} distributed in three sets ($\{E, S\}_{learning}$, $\{E, S\}_{validation}$ and $\{E, S\}_{test}$), corresponding to three stages of training.

A. Performance of the network at every stage of training.

Validation of the training of the network is reached at the iteration 2210. Table 1 shows the obtained mean squared error at the end of every stage, after the check of the validation.

Table 1: Performance of the Network At Every Stage of Training At Iteration 2210

Stage of training	**Mean squared error**
Learning	$1.5115. 10^{-3}$
Validation	$1.8895. 10^{-3}$
Test	$1.9716. 10^{-3}$

Evolution of mean squared error according to iterations at the three stages of training is shown in Figs. 8, 9 and 10. The Fig. 8 shows a fast minimization of the error $F_{mean(learning)}$ during the first 12 iterations such as this error switches from 1.81 at the first iteration to 0.0617 at 12^{th} iteration.

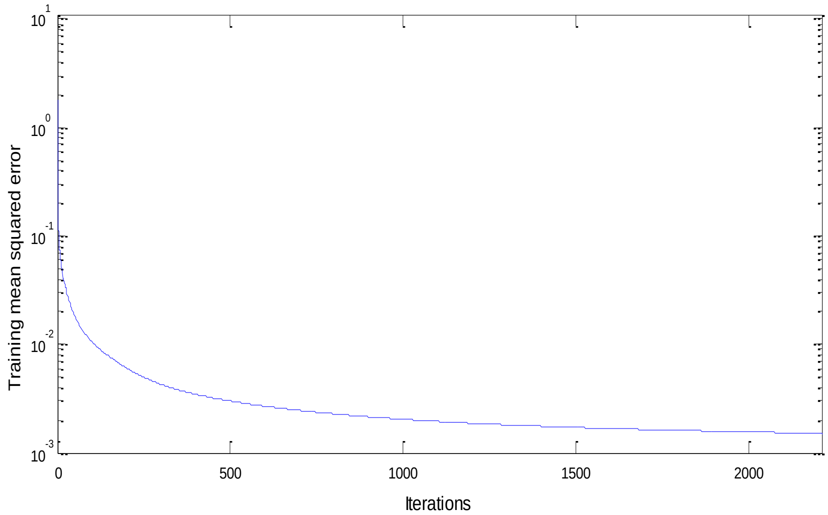

Fig. 4: Mean squared error at the training stage according to iterations

B. Evolution of the step size α Calculated by the golden section method

Golden section method generates [8] oscillating values of the step size during 2213 descent directions. The latter are calculated by the steepest descent algorithm.

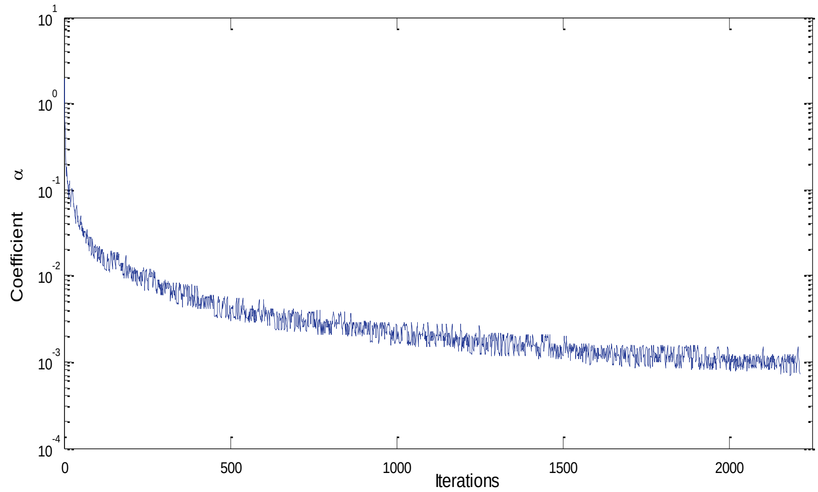

Fig. 5: Evolution of the step size α according to 2213 iterations

The value of α is between [1.9, 0.1] in the first 12 iterations, then it decreases in an oscillating way during the rest of the iterations to 7.10^{-4} value.

VI. DISCUSSION

By comparing both Fig. 8 and Fig. 11, we observed that the evolution of the mean squared error $F_{mean(learning)}$ is similar to the evolution of the step size α. The fast minimization of the learning mean squared error (Fig. 8) from 1.81 to 0.0617 at 12^{th} iteration is explained by the important values of α, which are between [1.9, 0.1]. The error $F_{mean(learning)}$ is minimized with a rate more at least fast until the iteration 1000 where it becomes almost stable (Fig. 4). The slow convergence after the iteration 1000 is explained by the use of low values of α which are in the margin [4.10^{-3}, $7.\ 10^{-4}$], which prevents the better correction of the error.

VII. CONCLUSION

When the steepest descent algorithm is far from the optimal values of the weights, it converges quickly and when the algorithm becomes near to the optimal values, it converges slowly. So the golden section method generates low values of α according to the values of the weights. Unlike the steepest descent algorithm which is has slow convergence around the optimum, there are other optimizations algorithms which have interesting rates of convergence: Conjugate gradient, Levenberg-Marquardt. These algorithms are going to be investigated in future work, in the framework of determination of the electrical parameters of solar cell by using the artificial neural network.

REFERENCES

[1] D. Jinlei, R. Rakesh, *A new method to determine the optimum load of areal solar cell using the Lambert W-function*, Solar Energy Materials & Solar Cells, 2008, vol. 92, pp. 1566–1569.

[2] T. Ikegami, T. Maezono ,F. Nakanishi, Y. Yamagata, K. Ebihara, *Estimation of equivalent circuit parameters of PV module and its application to optimal operation of PV system,* Solar Energy Materials & Solar Cells, 2001, vol. 67, pp. 389–395.

[3] L. Valerio, O. Aldo, G. Giuseppina, D. G. Alessandra, *An improved five-parameter model for photovoltaic modules*, Solar Energy Materials & Solar Cells, 2010, vol. 94, pp. 1358–1370.

[4] E. Karatepe, M. Mutlu Boztepe, M. Colak, *Neural network based solar cell*, Energy Conversion and Management, 2006, vol. 47, pp. 1159–1178.

[5] G. Cybenko, *Approximation by superpositions of a sigmoidal function*, Mathematics of control, signals and systems, 1989, vol. 2, pp. 303–314.

[6] K. Hornik, *Approximation capabilities of multilayer feedforward networks*, Neural Network, 1991, vol. 4, pp. 251–257.

[7] F. Ahmad, N. A. M. Isa, M. K. Osman, Z. Hussain, *Performance Comparison of Gradient Descent and Genetic Algorithm Based Artificial Neural Networks Training*, 10th International Conference on Intelligent Systems Design and Applications, IEEE, 2010, pp. 604–609.

[8] Q. Zuo, X. Yin, J. Zhou, B. J. Kwak, H. K. Chung, *Implementation of Golden Section Search Method in SAGE Algorithm*, Proceedings of the 5th European Conference on antennas ans propagations (EUCAP), IEEE, pp. 2028–2032.

