ISSN: 2319-8753

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 8, August 2014

Neural Network Training By Gradient Descent Algorithms: Application on the Solar Cell

Fayrouz Dkhichi*, Benyounes Oukarfi

Department of Electrical Engineering, EEA&TI laboratory, Faculty of Sciences and Techniques, Hassan II Mohammedia- Casablanca University, Mohammedia, Morocco

ABSTRACT: This present paper deals with the parameter determination of solar cell by using an artificial neural network trained at every time, separately, by one algorithm among the optimization algorithms of gradient descent (Levenberg-Marquardt, Gauss-Newton, Quasi-Newton, steepest descent and conjugate gradient). This determination issue is made for different values of temperature and irradiance. The training process is insured by the minimization of the error generated at the network output. Therefore, from the outcomes obtained by each gradient descent algorithm, we conducted a comparative study between the overall of training algorithms in order to know which one had the best performances. As a result the Levenberg-Marquardt algorithm presents the best potential compared to the other investigated optimization algorithms of gradient descent.

KEYWORDS: Artificial neural network, training, gradient descent optimization algorithms, comparison, electrical parameters, solar cell.

I. INTRODUCTION

The exhibitions under irradiance, temperature lead to the degradation of the internal characteristics of solar cell and prevent the photovoltaic (PV) panel to generate electrical power under its optimal performances. In order to study the influence of these handicapping factors, we must know the internal behavior of solar cell by determining the electrical parameters according to different values of irradiance and temperature.

The PV current (I_{PV}) produced at the output of solar cell is in a nonlinear implicit relationship with the internal electrical parameters. The latter can be identified analytically [1] or numerically [2] for a specific temperature and irradiance. In other hand, the study of the behavior of solar cell requires the identification of its parameters for various values of irradiance and temperature. Therefore, the Artificial Neural Network (ANN) seems the best adapted to insure this role.

The choice behind the use of the ANN returns to its capacity to predict results from the exploitation of the acquired data. The information is carried by weights representing the values of the connections between neurons. The functioning of the ANN requires its training by an algorithm insuring the minimization of the error generated at the output.

In the aim to determine the electrical parameters values, we compare in this study between the optimization algorithms of gradient descent that allow the training of the ANN. We distinguish three algorithms of second order of gradient (Levenberg-Marquardt, Gauss-Newton and Quasi-Newton) and two algorithms of the first order of gradient (steepest descent and conjugate gradient).

DOI: 10.15680/IJIRSET.2014.0308084

ISSN: 2319-8753

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 8, August 2014

II. SOLAR CELL MODEL

2.1. Single diode solar cell model

In our study the solar cell is modeled by an electrical model [3] with a single diode shown in Fig. 1:

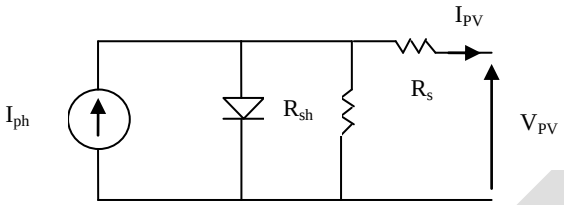

Fig. 1 Equivalent circuit of solar cell

R_s: Series resistance representing the losses due to the various contacts and the connections.
R_{sh}: Shunt resistance characterizing the leak currents of the diode junction.
I_{ph}: Photocurrent depending on both irradiance and temperature.
I_s: Diode saturation current. n: Diode ideality factor.
V_{th}: Thermal voltage $(V_{th} = A.T/q)$. T: Temperature of solar cell by Kelvin.
A: Boltzmann constant $(A = 1.380650310^{-23}$ J/K) . q: Electrical charge of the electron $(q = 1.6021764610^{-19}$ C) .

The mathematical equation deducted from the electrical circuit Fig. 1 is expressed as follows:

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

2.2. The operating process of solar cell under illumination

An illuminated solar cell generates a characteristic $I_{PV}=f(V_{PV})$ for every value of irradiance and temperature. We obtain this characteristic by varying the value of load R (Fig. 2).

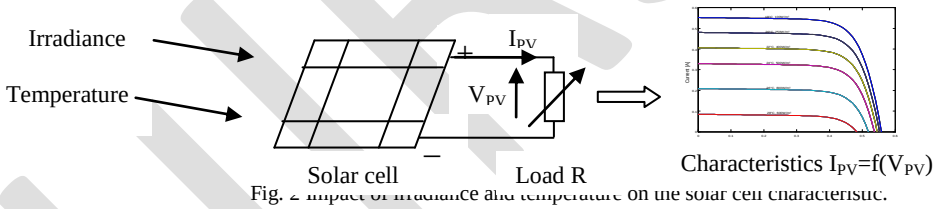

Fig. 2 Impact of irradiance and temperature on the solar cell characteristic.

The change of the solar irradiance between 100W/m² and 1000W/m² and the cellular temperature between 18°C and 65°C affects the values of the five electrical parameters R_s, R_{sh}, I_{ph}, I_s, and n of solar cell. Effectively, the current I_{ph} varies according to irradiance and the current I_s varies according to temperature while R_s, R_{sh} and n vary according to the both meteorological factors [4].

III. THE USED ARTIFICIAL NEURAL NETWORK

The identification of the internal electrical parameters for various values of temperature (T) and irradiance (G) is insured by the network ANN [4] shown in Fig. 3. The architecture includes an entrance layer, a hidden layer and an output layer.

The entrance layer contains two inputs [T, G], the hidden layer contains twenty hidden neurons and the layer of the output includes five output neurons corresponding to the five parameters R_s, R_{sh}, I_{ph}, I_s and n whose we want to predict the values.

DOI: 10.15680/IJIRSET.2014.0308084

ISSN: 2319-8753

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 8, August 2014

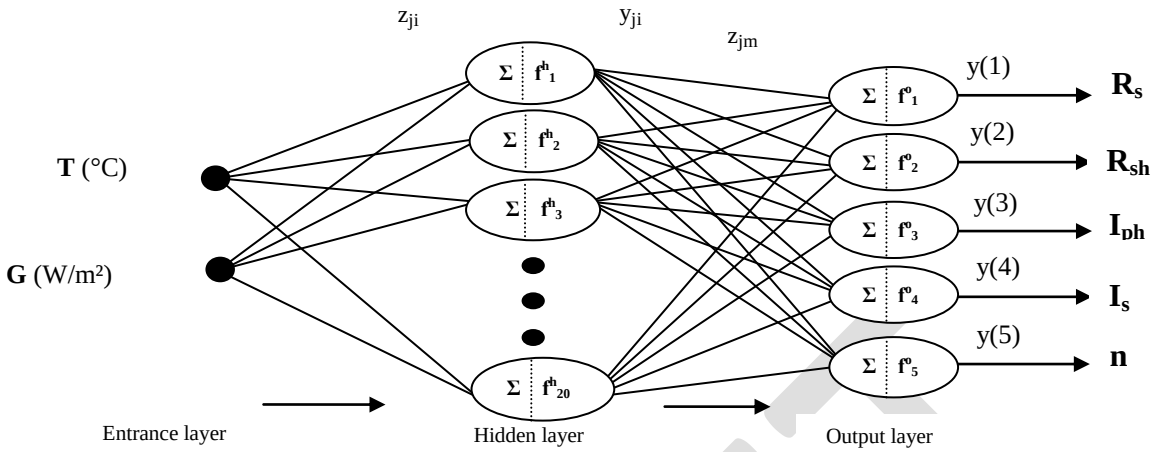

Fig. 3 Structure of the used ANN

i =1, 2: Index of inputs.
j=1,…, 20: Index of hidden neurons.
m=1.....5: Index of output neurons.
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.
f_j^h: Activation function « hyperbolic tangent » of the hidden neurons.
b_{mj}: Biais of the output neurons.
x_i: Input vector [T, G].
z_{ji}: Input of hidden neurons.
z_{mj}: Input of output neurons.
y_{ji}: Output of hidden neurons.
y: Matrix values of the network outputs, y = [R_s, R_{sh}, I_{ph}, I_s, n].
f_m^o: Activation function « linear » of the output neurons

The input of the hidden layer is calculated by the following expression:

$$z_{ji} = \sum_{i=1}^{2} x_i w_{ji} + b_{ji} \quad (2)$$

By the use of hyperbolic tangent as an activation function of the hidden neurons, the neurons calculate the value of their output using the following equation:

$$y_{mj} = f_j^h(z_{ji}) \quad (3)$$

To compute the inputs of the output neurons, we use the values of y_{mj} calculated previously.

$$z_{mj} = \sum_{j=1}^{20} y_{mj} w_{mj} + b_{mj} \quad (4)$$

The output of the neural network is calculated as follows:

$$y = f_m^o(z_{mj}) \quad (5)$$

IV. THE TRAINING ALGORITHMS

The mean squared error $J_{mean(learning)}$ generated by the ANN is expressed by the following equation:

$$J_{mean(learning)} = \frac{1}{p} \sum_{t=1}^{p} \sum_{s=1}^{v} (S_{learning}(t,s) - y_{learning}(t,s))^2 \quad (6)$$

p: Number of examples {input, output} learning.
s: Index indicates the number of output.
t: Index indicates the example number of learning stage.
v: Number of the network outputs.
S: Target values of the network outputs.

DOI: 10.15680/IJIRSET.2014.0308084

ISSN: 2319-8753

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 8, August 2014

The minimization of the error $J_{mean(learning)}$ is insured by adjusting the weights (w) of the ANN (Fig. 3). The training of the network is made at every time by one optimization algorithm from the set of algorithms (Levenberg-Marquardt, Gauss-Newton, Quasi-Newton, steepest descent and conjugate gradient).

4.1. Levenberg-Marquardt (LM) algorithm:

Algorithm of second order of gradient [5] allow the optimization of $J_{mean(learning)}$, the adjustment of w is insured by the expression:

$$w_{k+1} = w_k - \frac{J_k e_k}{(J_k^{'} J_k + \lambda I)} \tag{7}$$

J: Jacobian matrix of the function $J_{mean\ (learning)}$. e: Error between the target and the calculated network outputs.
k: Number of iterations. I: Matrix identity.

The regulation of the Levenberg-Marquardt damping factor is made as follows:
If the calculated $J_{mean(learning)}$ for w_{k+1}, decreases, so: $\lambda = \lambda/10$

Else $\lambda = \lambda * 10$ and $w_{k+1} = w_k$

4.2. Gauss-Newton (GN) algorithm:

This algorithm [6] is from the same family as the Levenberg-Marquardt. It minimize the error of output by varying w using step size α.

$$w_{k+1} = w_k - \alpha \frac{J_k e_k}{(J_k^{'} J_k + \lambda I)} \tag{8}$$

4.3. Quasi-Newton (QN) algorithm:

Since the second derivative of $J_{mean(learning)}$ is not obvious to compute, the Quasi-Newton algorithm [7] suggests an alternative way, which is the approximation of the second derivative by the hessian B calculated in Eq. (11). We therefore adjust the weight using the Eq. (12):

$$u_k = -\frac{2 J_k e_k}{B_k} \tag{9}$$

$$n_k = 2(J_{k+1} e_{k+1} - J_k e_k) \tag{10}$$

$$B_{k+1} = B_k + \frac{u_k u_k^{'}}{u_k^{'} n_k} - \frac{B_k n_k n_k^{'} B_k}{n_k^{'} B_k n_k} \tag{11}$$

$$w_{k+1} = w_k - \frac{J_k e_k}{(J_k^{'} J_k + B_k)} \tag{12}$$

4.4. Steepest descent (SD) algorithm:

The adjustment of ANN weights by the algorithm of steepest descent [8] is insured by the following equation:

$$w_{k+1} = w_k - 2\alpha J_k e_k \tag{13}$$

4.5. Conjugate gradient (CG) algorithm:

This conjugate gradient algorithm [9] is from the same family as the steepest descent algorithm, but the both are algorithms of the first order of gradient.

At the first iteration

$$d_{old} = -2 J_k^{'} e_k \tag{14}$$

From the second iteration

ISSN: 2319-8753

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 8, August 2014

$$\beta = \frac{d_{old}^{'} d_{old}}{(J_k^{'} e_k)^{'} J_k^{'} e_k} \quad (15)$$

$$d_{new} = -2 J_k^{'} e_k + \beta d_{old} \quad (16)$$

$$w_{k+1} = w_k + \alpha d_{new}$$

V. RESULTS AND DISCUSSION

The training of the network is made with 130 inputs-outputs examples distributed in three sets (learning, validation and test) [10].

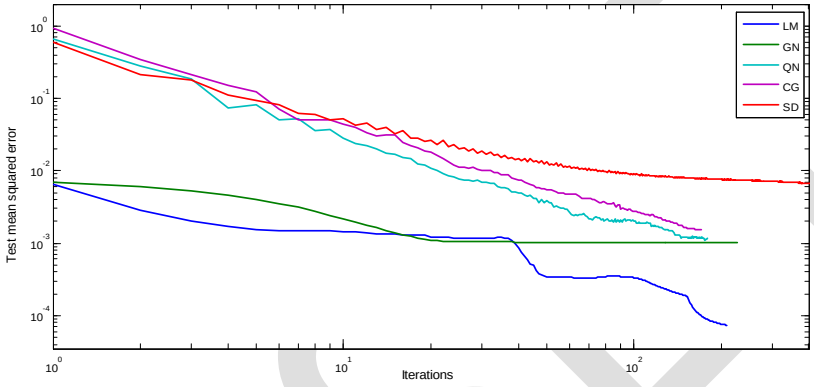

Fig. 4 Evolution of the test means squared errors of the five training algorithms

The Fig. 4 shows the curves of the test means squared errors obtained by the ANN. Each curve corresponds one of the five optimization algorithms. We used the logarithmic scale in the axis of iterations in order to well show the behavior of the algorithms convergence. Therefore, the LM algorithm allows a good training of the ANN compared to other algorithms (GN, QN, CG and SD). The both QN and CG have a stiff slope compared to SD which converges slowly (Fig. 4).

Table 1: Comparison between the behaviors of the algorithms

Algorithm	Time of training (s)	Test mean squared error	Correct Rate (%)
SD	04min32s	$1.9716.\ 10^{-3}$	99.50
CG	22	$1.891.\ 10^{-3}$	99.51
QN	15	$1.23.\ 10^{-3}$	99.64
GN	18	$1.02.\ 10^{-3}$	99.81
LM	11	$8.76.\ 10^{-4}$	99.95

The Table 1 includes the results obtained after training the ANN at every time by one algorithm of the five optimization algorithms. As a result, the SD converges slowly and determines the values of the five electrical parameters at the output of the ANN far from their targets with an error rate of 5%. By comparing the results of this algorithm to those of CG, the latter present fast convergence, but with error rate of 4 %. Both algorithms QN and GN present more important rates of correction: 99.51% and of 99.81% successively. Therefore by comparing the results of LM with that of the SD, CG, QN and GN algorithms, The LM presents better rate of convergence (time of training) and better rate of correction.

ISSN: 2319-8753

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 8, August 2014

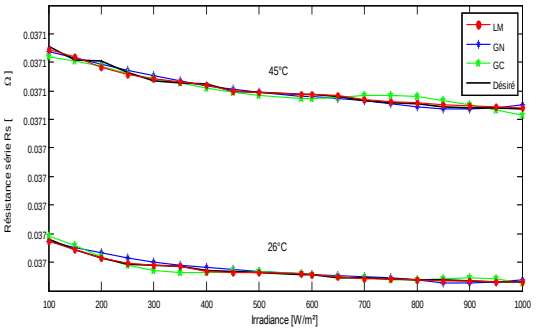

Fig. 5 Series resistance R_s according to irradiance

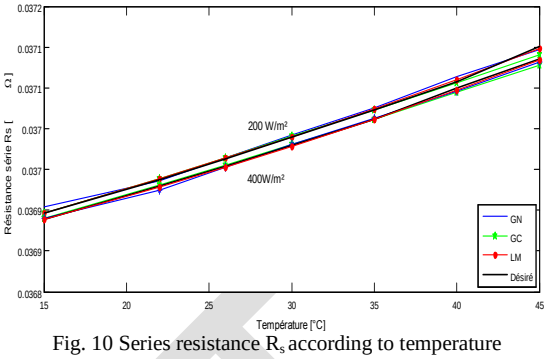

Fig. 10 Series resistance R_s according to temperature

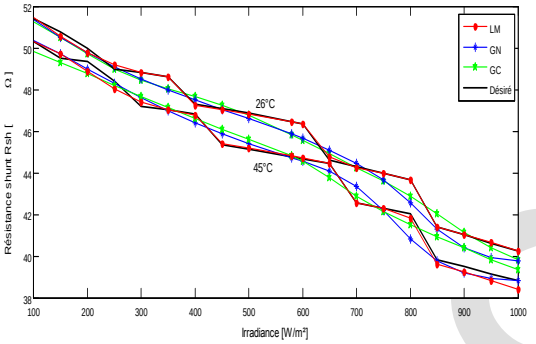

Fig. 6 Shunt resistance R_{sh} according to irradiance

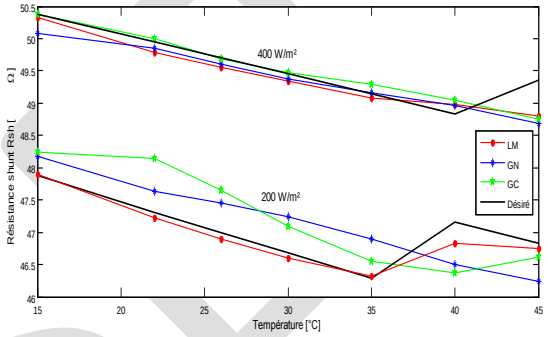

Fig. 11 Shunt resistance R_{sh} according to temperature

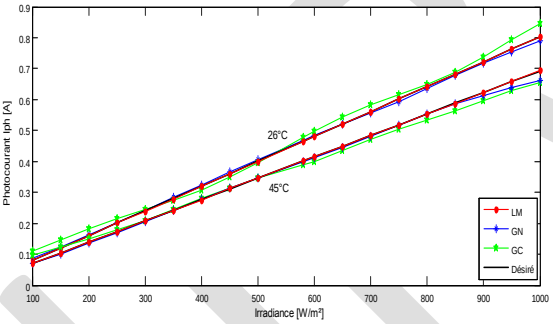

Fig. 7 photocurrent I_{ph} according to irradiance

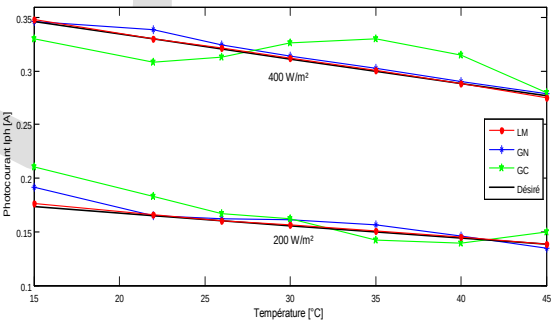

Fig. 12 photocurrent I_{ph} according to temperature

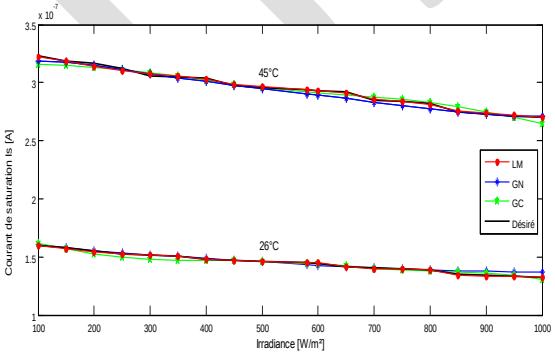

Fig. 8 Saturation current I_s according to irradiance

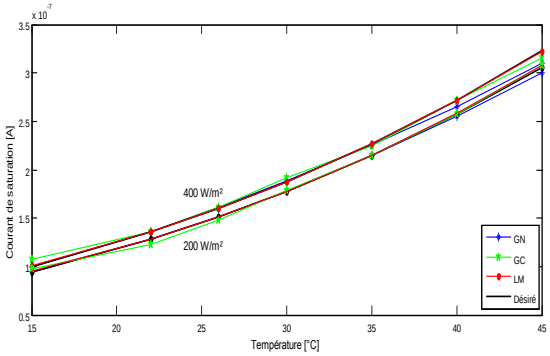

Fig. 13 Saturation current I_s according to temperature

DOI: 10.15680/IJIRSET.2014.0308084

ISSN: 2319-8753

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 8, August 2014

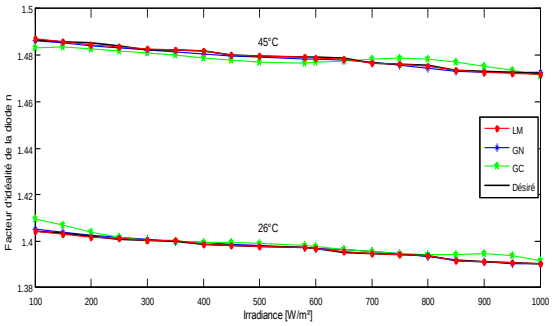

Fig. 9 Diode Ideality factor n according to irradiance

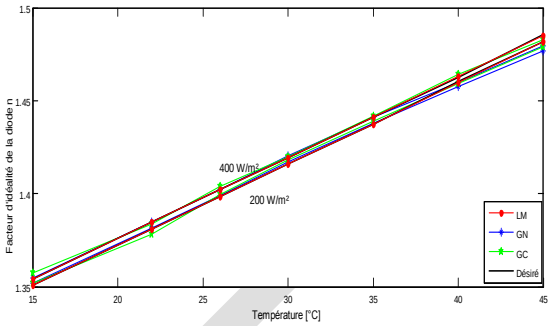

Fig. 14 Diode Ideality factor n according to temperature

Figs. 5-9 show the evolution of the parameters R_s, R_{sh}, I_{ph}, I_s and n according to irradiance for two fixed values of temperature (26°C and 45°C) and Figs. 10-14 describe the evolution of the five electrical parameters according to temperature for two fixed values of the irradiance (200W/m² and 400W/m²). We observe that LM gives curves more compatible with those desired. By comparing LM with GN, the latter gives more or less curves close to the desired ones. In other hand, GC generates an error, more important than that observed with GN.

The rate correction of SD is low compared to other algorithms and that is explained by its oscillation around the optimum, which prevents the convergence to reach the optimum solution (Fig. 4 and Table 1). The use of the coefficient β by the Eq. (15), allow to the CG algorithm to converge quickly (Table 1) compared to SD. The QN and GN algorithms present two correction rates more interesting than those of the SD and CG algorithms. This behavior is explained by the fact that QN and GN are better known by their fast convergence near to the optimum. The LM algorithm presents the best behavior of the convergence compared to other algorithms, due to the combination between the features of SD and GN. Therefore, LM behaves as SD for big values of λ And then, it behaves as GN for small values of λ [6].

VI. CONCLUSION

The Levenberg-Marquardt algorithm provides interesting performances at the training of the artificial neural network compared with other optimization algorithms of gradient descent. Effectively, it determines the values of the five electrical parameters of the solar cell so close to desired ones due to its capacity to optimize the mean squared error to the minimal value in a small amount of time.

REFERENCES

[1] A. Jain, A. Kapoor, *Exact analytical solutions of the parameters of real solar cells using Lambert W-function*. Solar Energy Materials & Solar Cells, 2004, vol. 81, pp. 269-277.

[2] H.Qin, J. W. Kimball. *Parameter Determination of Photovoltaic Cells from Field Testing Data using Particle Swarm Optimization*. IEEE, 2011.

[3] Ikegami. T, Maezono. T, F. Y. Nakanishi., Y. amagata, 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.

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

[5] R. Zayani, R. Bouallegue, D. Roviras, *Levenberg-Marquardt learning neural network for adaptative predistortion for time- varying HPA with memory in OFDM systems*, 16th European Signal Processing Conference (EUSIPCO 2008), 2008, pp. 25-29.

[6] P. R. Dimmer, O, P, D, Cutteridge. *Second derivative Gauss-Newton-based method for solving nonlinear simultaneous equations*. IEEPROC, december 1980, vol. 127, 6, pp. 278-283.

[7] R. Setiono, L. C. K. Hui. *Use of a Quasi-Newton Method in a Feedforward Neural Network Construction Algorithm*. IEEE Transactions on neural networks, January 1995, vol. 6, 1, pp. 273-277.

[8] L. Gong, C. Liu, Y. Li, F. Yuan, *Training Feed-forward Neural Networks Using the Gradient Descent Method with the Optimal Stepsize*. Journal of Computational Information Systems , 2012, vol. 8, pp. 1359-1371.

[9] X. Gong, W. S. H. Xu, *The Conjugate Gradient Method with Neural Network Control*, IEEE, 2010, pp.82–84

[10] A. J. Adeloye, A. De Munari, *Artificial neural network based generalized storage–yield reliability models using the Levenberg–Marquardt algorithm.* Journal of Hydrology, 27 October 2005, vol. 362, pp. 215-230.

DOI: 10.15680/IJIRSET.2014.0308084