International Journal of Science, Environment and Technology, Vol. 2, No 5, 2013, 779 –786

ISSN 2278-3687 (O)

THE NEURAL NETWORKS: APPLICATION AND OPTIMIZATION APPLICATION OF LEVENBERG-MARQUARDT ALGORITHM FOR TIFINAGH CHARACTER RECOGNITION

I. Badi[1], M. Boutalline[2] and S. Safi[3]

[1,2]Computer Sciences Department, FST, Sultan Moulay Slimane University, PO. Box: 523, Mghila, Beni Mellal, Morocco

[3]Computer Sciences Department, FP, Sultan Moulay Slimane University, PO. Box. 523, Beni Mellal, Morocco

E-mails: [1]badi.imad@gmail.com , [2]boutalline@gmail.com , [3]safi.said@gmail.com

Abstract: Levenberg-Marquardt (LM) Optimization is a virtual standard in nonlinear optimization. It is a pseudo-second order method which means that it works with only function evaluations and gradient information but it estimates the Hessian matrix using the sum of outer products of the gradients. In this word, we will applythe Levenberg-Marquardt method for learning and recognition pattern concerned Tifinaghcharacters; this last will be compared to other methods such as Windrow-Hoff (WH). This note reviews the application for Levenberg-Marquardt for network neuronal and also details the algorithm.

Keywords: Levenberg-Marquardt, Optimization, network neuronal, gradients, Recognition pattern, Learning pattern Tifinagh.

I. INTRODUCTION

The progress vitalized the interest in methods of numerical optimization systems becoming more complex. For example, application of optimization methods of neuronal network is a technique usually used in numerical ways in the industry. In this work we are interested to apply the method of optimization of Levenberg-Marquardt for neuronal network.

We wish to mention in passing the two major types of optimization methods: probabilistic methods and methods of gradient type. We adopt the perspective of the second group. We're looking at a problem of pattern recognition. Specifically, we will focus on the recognition of Tifinagh characters. The aim of our work is to apply some optimization algorithms to neural networks to learn the method of Windrow-Hoff is a classical method of first order; it will be replaced by the method of Levenberg-Marquardt, which is one method of second order.

*Received July 13, 2013 * Published October 2, 2013 * www.ijset.net*

II. ALGORITHM OF OPTIMIZATION USED

A. METHOD OF LEVENBERG- MARQUARDT

The method of Levenberg-Marquardt [2] is from the Newton family methods of second order, it consist to modify the parameters according to the following relation:

$$W_k = W_{k-1} - [H_{k-1} + \lambda_{k-1}.I]^{-1}.J_{k-1} \quad (1)$$

This method is particularly clever because it makes a tradeoff between the gradient direction and the direction given by Newton's method. Indeed, if $\lambda_{k-1}$is great, we recognize the gradient method (in this case the step value is given by $(1/\lambda_{k-1})$ and if λ_{k-1} is small parameter changes is that of Newton method.

B. Levenberg-Marquardt algorithm

Input: $\mathbb{R}^n \rightarrow \mathbb{R}$ a function such that $f(i) = \sum_j^m w_{ij} * x_i$

Output: t, o local minimum of the cost function f

Begin

$k \leftarrow 0$

$w \leftarrow w_0$

Whileupdate $<> 0$ and $k < k_{\max}$ **do**

Find$d$ such that $([H + \lambda.I]^{-1}.\nabla J).d = J^T * f$

$W \leftarrow W_{lm}$

If$f(W_{lm}) < f(W)$**then**

$W = W_{lm}$

$\lambda \leftarrow \frac{\lambda}{\nu}$

Else

$\lambda \leftarrow \upsilon\lambda$ **Return** W **end**

III. SIMULATION

Let's start with a single layer perceptron, the input layer contains three cells and the output layer contains three cells. The vector used as input is$\mathbf{x} = [\mathbf{1\ 1\ 1}]$; the desired output is $\mathbf{t} = [\mathbf{1\ 0\ 0}]$.

In this case the Jacobean matrix is a matrix of size 3 * 3, which means we have nine parameters. By calculating the vector f obtained with the method of LM, we observed that after 5 iterations we have the right result (**FIG. 2**), the resulting vector is given by f 0.9965 0.0277 0.0225;

Since it uses the sigmoid activation function (**FIG. 3**) can be used threshold is 0.5 selected manually. All cells that have a value greater than 0 .5 are active, and cells that have a value below 0.5 are inactive, so the final vector will conform to the desired output.

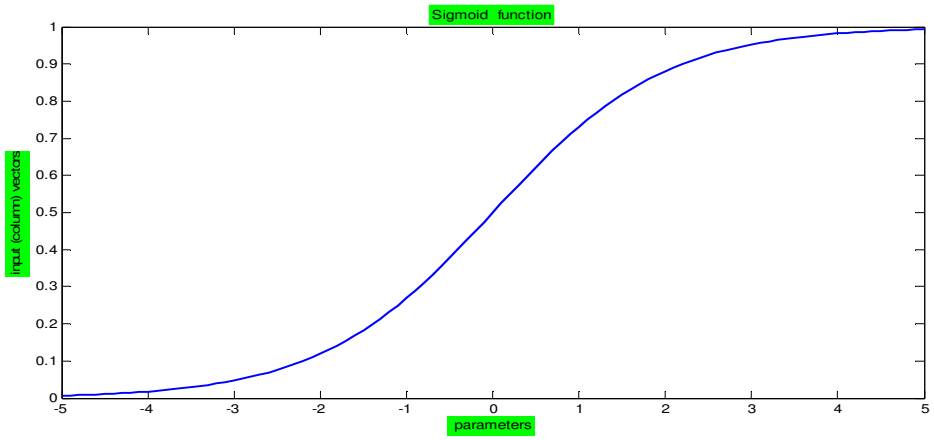

FIG. 1 Sigmoid activation function

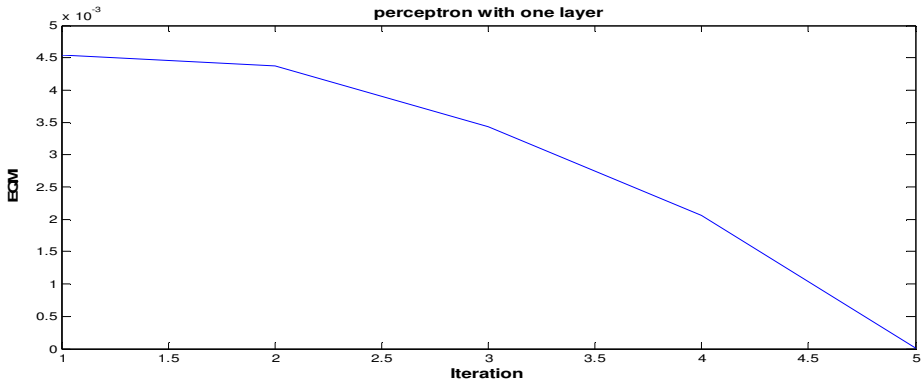

FIG. 2 Quadratic error with Levenberg-Marquardt

A. The logic function with Levenberg-Marquardt and Windrow-Off.

A.1. *Levenberg-Marquardt:*

Principe:

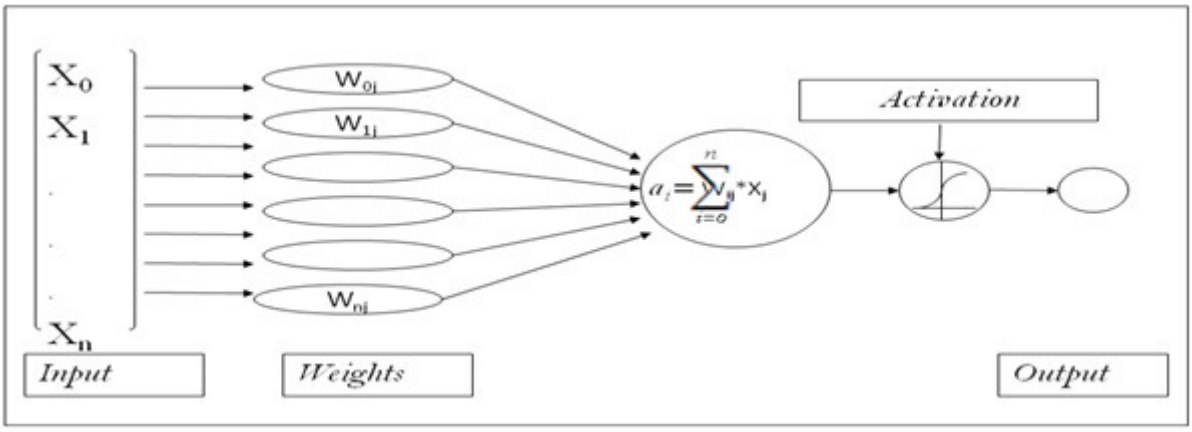

FIG. 3 Functional schemas

The perception is the simplest module in neural networks (**Fig 3**). It is composed of two layers of input and output, the input layer called the retina and the output layer is called the answer. These two layers are linked together by the weight coefficients W_{ij}.

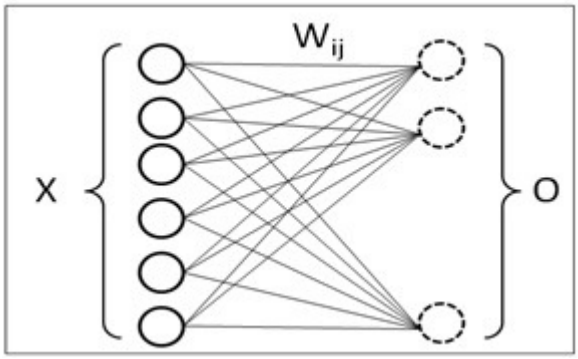

FIG. 4 Perceptron with one layer

X: Input Layer

O: Output Layer

I: the number of neurons in the input layer.

J: The number of neurons in the output layer.

W_{ij}: The coefficients or synaptic weights between the inputcell and the output cell.

It's possible to use the learning methods for the logic functions (ET, OU and XOR) with the method of Levenberg- Marquardt, for that we can use just a single layerPerceptron [3,4], this last contain 2 cells in the input and 1 cell in the output. The figure 5 shows the schedule of the quadratic error according to the number of iteration.

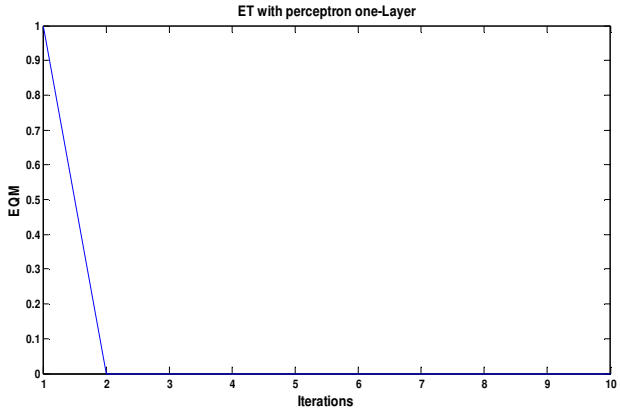

FIG. 5 Quadratic error of the logic function AND, OR and XOR with LM

As shown in Figure 5, the squared error is canceled out almost after 2 epochs (iterations), we obtained the same result for all functions (AND, OR, XOR) and in a relatively short

turnaround time, which shows the effectiveness of the method for convergence towards the optimum.

A.2. *Windrow-Off:*

The Windrow-Hoff algorithm or the "delta rule" is to change the weight for each iterations by the following formula:

$$W_{i+1} = W_i + \ alpha * (y_k - s_k) * x_i \quad (2)$$

The XOR function, which is a function not linearly separable, requires the use of a hidden layer (FIG 7) with the WH method. By cons for other functions (AND, OR) just use a single layer as input and one output layer.

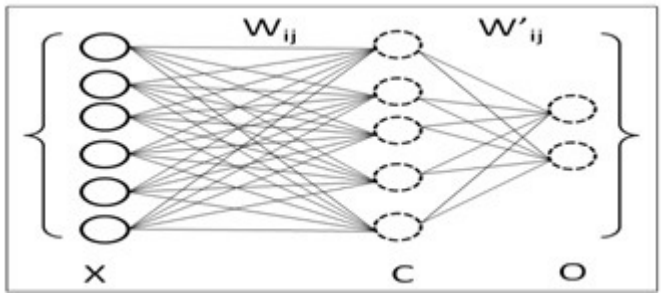

FIG. 6 Neuronal network with hidden layer

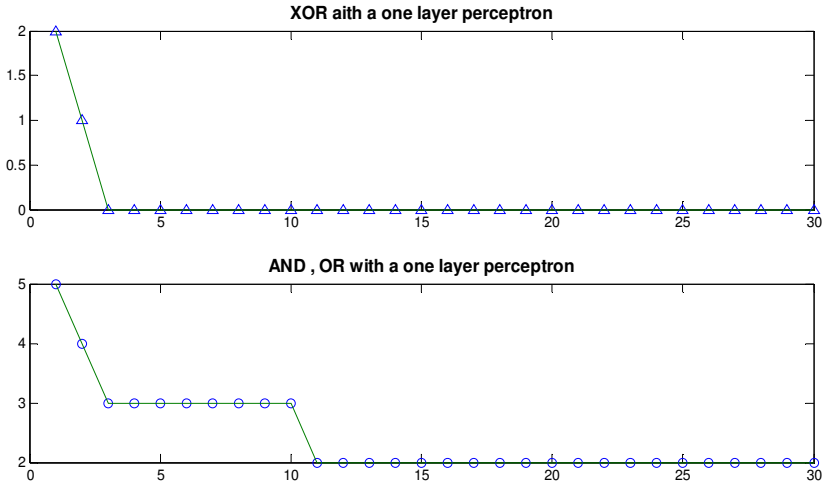

FIG .7 functions AND, OR and XOR with WH

Methods	LogicalFunctions	Execution time (s)
LM	AND	0.036978
WH	AND	0.532354
LM	OR	0.044224
WH	OR	0.532354

LM	XOR	0.040135
WH	XOR	0.478828

TABLE 1: the comparative table of the results obtained

The comparative table regroups the different results given by the tow methods of optimization quoted above.

From the table, we notice that for the examples studied, the method of Levenberg-Marquardt is more efficient and advantageous in terms of execution time; In this case, we see that learning is easier when the number of examples is small. TheLevenberg-Marquardt method is more advantageous than that of Windrow-Hoff for convergence to the optimum, that's where the term clever method because it is a mixture of two techniques mentioned above. Indeed, this method tends to Newton's method for a value of lamda small but it is equivalent to the gradient method for a single lambda = lamda/10 not worth lamda large. The Hessian is always positive definite which ensures convergence to a minimum of the solution. Furthermore the implementation of the method is fairly simple as that Windrow-Hoff for the perceptron architecture used, for example the XOR function is a function that nonlinearly separable, which requires the use of a hidden layer with the Windrow-Hoff method. With the method of Levenberg-Marquardt learning can be done without using a multilayer perceptron, It can be done using a single layer perceptron, as used for "AND" and "OR" functions, two cell inputs and one output.

B. The recognition of character Tifinagh

The Levenberg-Marquardt method can be used for the recognition of characters or numbers, or other applications in the field of pattern recognition. It is a very effective method in terms of execution time.

In this work, we will work on problem recognition characters Tifinagh with Levenberg-Marquardt method, due to the advantages shown above. The principle is the same as that fitted in the previous examples. Consider a network comprising input layer neurons to 100, output layer 10 neurons. Learning is supervised, that is to say, the network is presented, along with a form and pattern. The transfer function used is a sigmoid function, which plays importantaim differentiability. Learning in this type of structure is to apply torques (inputs, outputs desired) at the input of the network. An actual output is calculated for each neuron of the input layer.

Then the error is calculated and then propagated in the network, giving rise to a change poids. The following figures show the recognition system and the results of the application of LM.

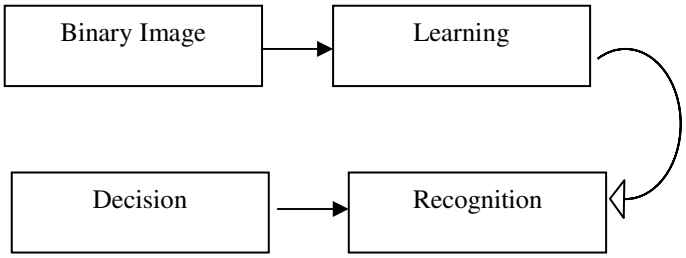

FIG. 8 Recognition System

At first, we will work on a binary image in order to validate the algorithm before introducing pretreatments phases.

An input layer which represents the inputs to which are transmitted the data to be processed, it contains hundred cells.

An output that delivers results, it contains cells representing the various figures, the activation of each cell represent the character corresponding.

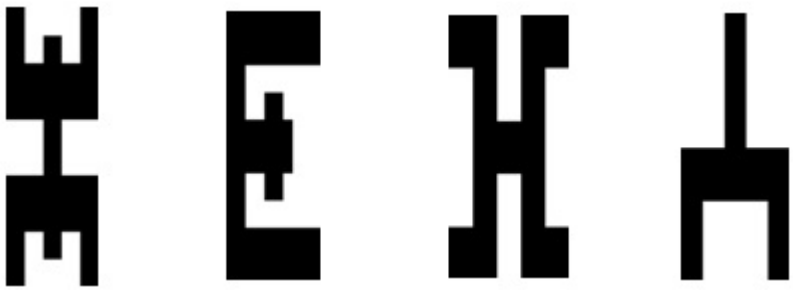

FIG. 9 some characters recognized with LM method.

VI. Conclusion

Our work is based on the recognition characters Tifinagh with neural networks. The Levenberg-Marquardt has been studied with other algorithms such as Windrow-Hoff. Theexperimental results showed the effectiveness of the Levenberg-Marquardt algorithm in terms of speed, execution time for convergence to the optimum, of course, no method leads to the global minimum course cost.

The Levenberg-Marquardt method is a second order. It is faster and provides better convergence towards a minimum squared error, but it was found that the efficiency of the method decreases as the number of weight increases as the size of the Hessian increases and

requires a large capacity memory. The Levenberg-Marquardt algorithm draws its effectiveness because it is a compromise (mixing) method between two different generations. Gradient method which is a first order method is considered effective when far from the optimum. Newton's method is a method of second order deemed effective to turn to turn to an optimum. This clever combination of these two methods is intended to ensure convergence towards the optimum minimum execution time.

REFERENCES

[1] I. BADI S. SAFI B. BOUIKHALENE, THE OPTIMIZATION ALGORITHMS IN NEURONAL NETWORK LEARNING, VOLUME 2, ISSUE 3, MARCH 2012 ISSN: 2277 128X, 2012.

[2] Vrahatis, M. N., Magoulas, G.D. & Plagianakos, V.P. [2003] "From linear to nonlinear iterative methods," Appl. Numer. Math. 45, 59–77.

[3] Kisi, O., Daily river flow forecasting using artificial neural networks and auto-regressive models. Turkish Journal of Engineering and Environmental Sciences, 29, 9–20 (2005).

[4] R.P. Lippmann, «An Introduction to Computing with Neural Nets», IEEE ASSP Magazine, pp. 4-22, April 1987.

[5] C.T. Kelley. Iterative Methods for Optimization. SIAM Press, Philadelphia, 1999.

[6] Kalyviotis N., Hu H., University of Essex, A Neural Network decision module for the Adaptive Strategic Planning Framework in RoboCup, Proceedings of International Symposium on Robotics and Automation, 2002.