Volume 2, Issue 3, March 2012 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

The Optimization Algorithms in Neuronal Network Learning

I.BADI
Computer Sciences Department
University Sultan Moulay Slimne
Beni Mellal Morocco
badi.imad@gmail.com

S.SAFI
Computer Sciences Department
University Sultan Moulay Slimne
Beni Mellal Morocco
safi.said@gmail.com

B.BOUIKHALENE
Computer Sciences Department
University Sultan Moulay Slimne
Beni Mellal Morocco
bbouikhalene@yahoo.fr

Abstract — **Levenberg-Marquardt 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.This note reviews the application for Levenberg-Marquardt for network neuronal and also details the algorithm.**

Keywords **—Levenberg-Marquardt, Optimization, network neuronal, gradients.**

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 numbers. The aim of our work is to apply some optimization algorithms to neural networks to learn the method of Widrow-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.

II. ALGORITHM OF OPTIMIZATION USED

A) METHOD OF LEVENBERG- MARQUARDT

The method of Levenberg-Marquardt[1] is from the Newton family methods of second order, it consist to modify the parameters according to the following relation:

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

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

B) DIAGRAM

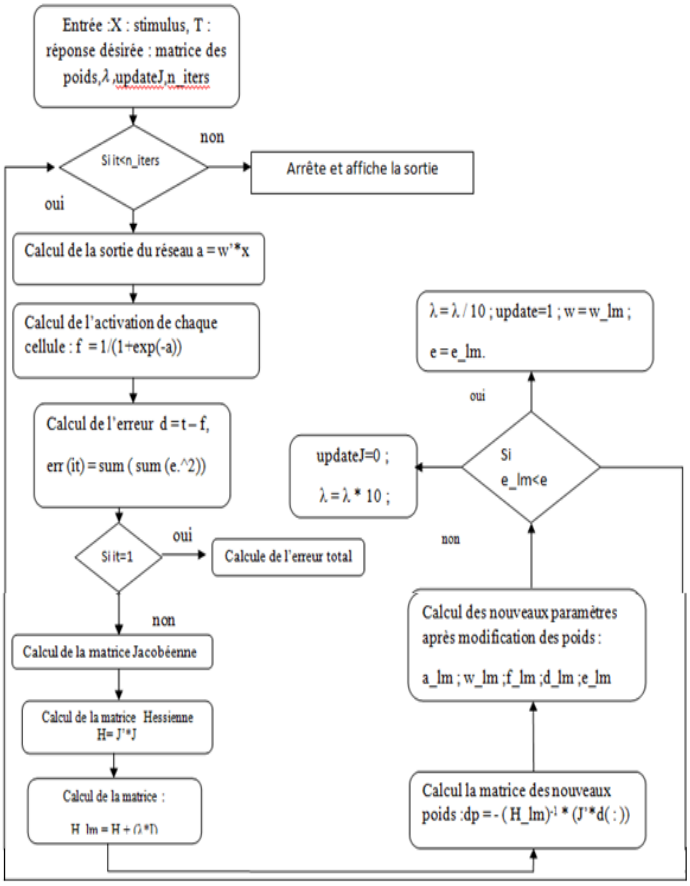

FIG 1: diagram of LM

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 x = [1 1 1]; the desired output is t = [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, the resulting vector is given by f 0.9965 0.0277 0.0225;

Since it uses the sigmoid activation function 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.

.

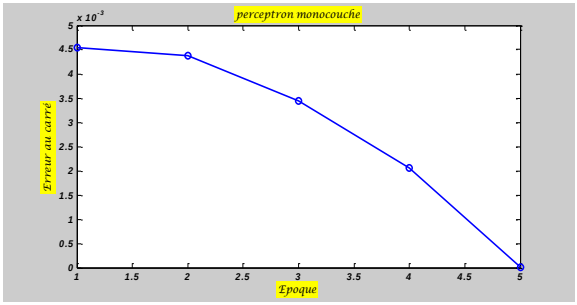

FIG 2 :The quadratic error with Levenberg-Marquardt.

1 *The logic function with Levenberg-Marquardt and Widrow-Off.*

1.1 Levenberg-Marquardt:

1.1.1 *The perception:* The perception is the simplest module in neural networks (Fig 11). 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}.

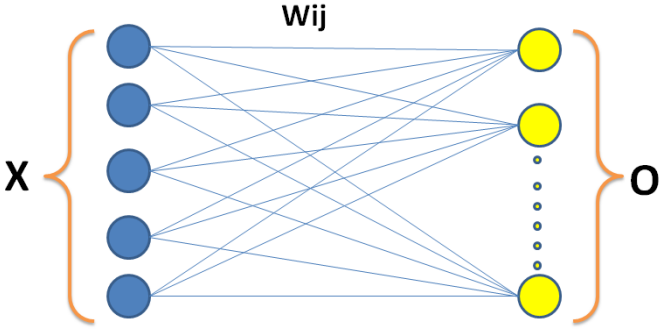

FIG 3 : Perceptron

X : Input Layer
O : Output Layer
I: the number of neurones in the input layer.
J: The number of neurones in the output layer.
W_{ij} : The coefficients or synaptic weights between the input cell 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 layer perceptron[2,3], this last contain 2 cells in the input and 1 cell in the output. Les figures 3,4 et 5 show the schedule of the quadratic error according to the number of iteration.

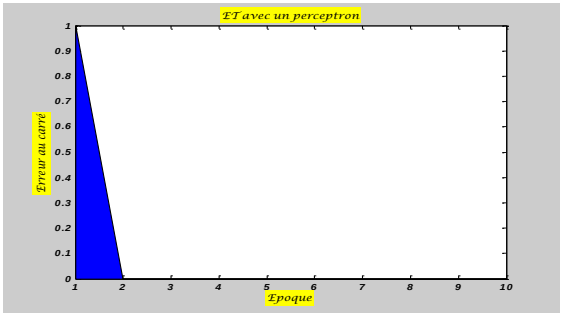

FIG 3 : The quadratic error of the logic function ET with LM.

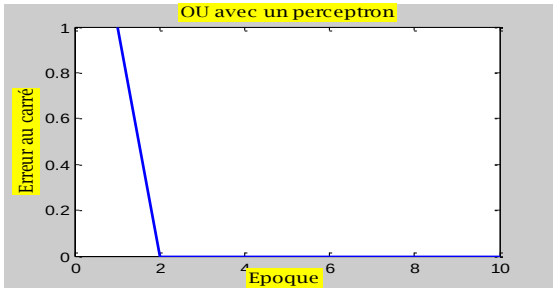

FIG 4 : The quadratic error of the logic function OU with LM.

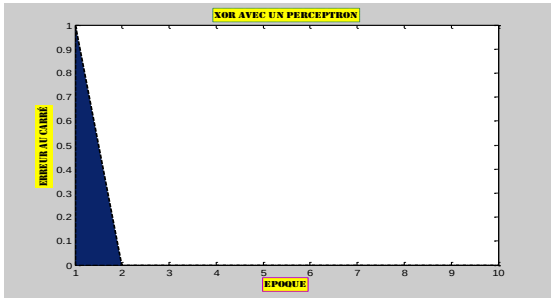

FIG 5 : The quadratic error of the logic function XOR with LM.

As shown in Figures 3, 4 and 5, the squared error is canceled out almost after 2 epochs (iterations), and in a relatively short turnaround time, which shows the effectiveness of the method for convergence towards the optimum

1.2 Widrow-Off:

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

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

© 2012, IJARCSSE All Rights Reserved

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

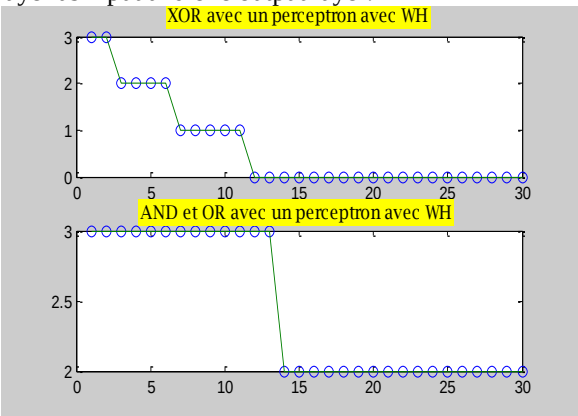

FIG 6: the functions AND, OR and XOR with WH.

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

TABLE 1: the comparative table of the results obtained

Methods	Logical Functions	Execution time (s)
LM	ET	0.038424
WH	ET	0.610579
LM	XOR	0.075657
WH	XOR	0.531939
LM	OR	0.028656
WH	OR	0.473194

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, The Levenberg-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 Widrow-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 Widrow-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.

1.2 The recognition of numbers

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. This example represents the entry number 3 in the first place, so there must be only the third cell. The following figure shows the error quadratic in the number of iterations.

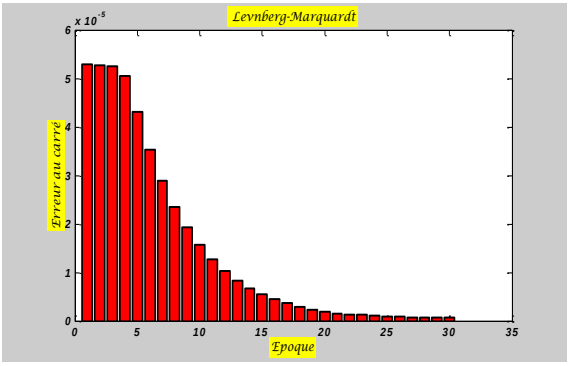

FIG 7: The error quadratic function of iterations

This example was deliberately constructed to highlight a comparison between the LM method and that of WH. The main problem of learning in terms of convergence of optimization algorithms is to pass a function in the vicinity of few learning points. This shows that we have the basic interest of learning the largest possible, and the cost functional has fewer local minima, optimization algorithms and easily finds the global minimum.

For the following example, we have been learning and recognition of the number three with the LM algorithm, we notice that after one iteration the resulting output and the desired output, so the optimum is reached.

Cons by learning the same figure with the WH method give the following results:

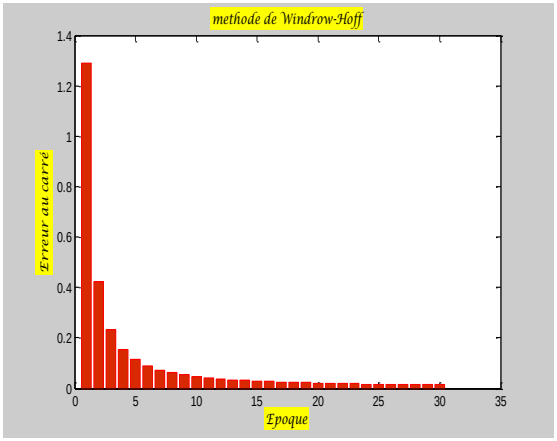

FIG 9-square error as a function of iterations with WH.

The following table gives a comparison between the two methods in terms of execution time.

TABLE 2: Comparison between the method of Levenberg-Marquardt and that of Windrow-Hoff

© 2012, IJARCSSE All Rights Reserved

Methods	execution time
LM	0.150418 seconds.
WH	0.241639 seconds.

With the method of Levenberg-Marquardt can have the same result in one iteration with a run time 0.150418 seconds.
The Levenberg-Marquardt method can be used in the learning of other figures.

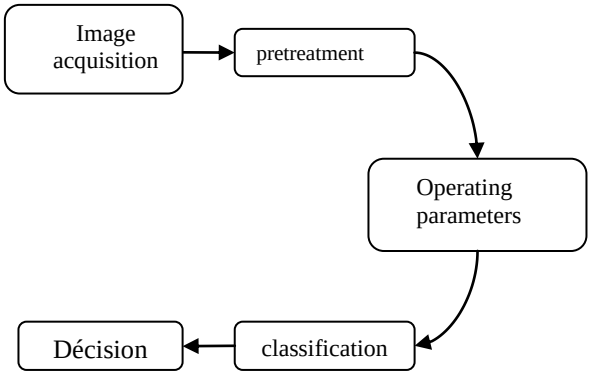

FIG 10: Recognition System

Such a network consists of a layer of grafted cells and thirty one output layer of ten cells.

Conclusion:

Our work is based on optimization methods in neural networks. The Levenberg-Marquardt has been studied with other algorithms like Windrow-Hoff. The experimental results showed the effectiveness of the Levenberg-Marquardt point of view speed, run time for convergence to the optimum. The Levenberg-Marquardt inspires its effectiveness because it is a compromise (mixture) between two methods of two different generations. The gradient method is a method of first order is considered effective when less than optimal. Newton's method is a method of second order is considered effective to turn around an optimum. This clever combination of these two methods designed to ensure the convergence to the optimum execution time minimal.

REFERENCES

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

[2] 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).

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

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

[4] 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.

© 2012, IJARCSSE All Rights Reserved