e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:05/Issue:01/January-2023 Impact Factor- 6.752 www.irjmets.com

A NEW QUASI-NEWTON METHOD FOR NON-LINEAR OPTIMIZATION PROBLEMS AND ITS APPLICATION IN ARTIFICIAL NEURAL NETWORKS (ANN)

Alaa Luqman Ibrahim*1, Mohammed Guhdar Mohammed*2

*1Department Of Mathematics, Faculty Of Science, University Of Zakho, Zakho, Kurdistan Region, Iraq.

*2Department Of Computer Science, Faculty Of Science, University Of Zakho, Zakho, Kurdistan Region, Iraq.

DOI : https://www.doi.org/10.56726/IRJMETS32949

ABSTRACT

The Quasi-Newton (QN) method is a widely used stationary iterative method for solving unconstrained optimization problems. One particular method within the Quasi-Newton family is the Symmetric Rank-One (SR1) method. In this research, we propose a new variant of the Quasi-Newton SR1 method that utilizes the Barzilai-Borwein step size. Our analysis demonstrates that the updated matrix resulting from the proposed method is both symmetric and positive definite. Additionally, our numerical experiments show that the proposed SR1 method, when combined with the PCG method, is effective in solving unconstrained optimization problems, as evidenced by its low number of iterations and function evaluations. Furthermore, we demonstrate that our proposed SR1 method is more efficient in solving large-scale problems with a varying number of variables compared to the original method. The numerical results of applying the new SR1 method to neural network problems also reveal its effectiveness.

Keywords: Unconstrained Optimization, Quasi-Newton Methods,, Line Search Method, PCG Method, Artificial Neural Network.

I. INTRODUCTION

Artificial neural networks (ANNs), usually simply called neural networks (NNs) or neural nets are computing systems inspired by the biological neural networks that constitute animal brains. An ANN is based on a collection of connected units or nodes called artificial neurons, which loosely model the neurons in a biological brain. Each connection, like the synapses in a biological brain, can transmit a signal to other neurons. ANNs applied in many aspects of artificial intelligence [1] because of their excellent ability of self-learning and self-adapting, they have been successfully. They are often found to be more active and precise than other classification techniques [2]. Although several different ways have been suggested, the feed forward neural networks (FNNs) are the most familiar and widely used in different kinds of applications.

The multilayer feed forward neural networks (FNNs) are parallel computational models comprised of densely interconnected, adaptive processing units, characterized by an inherent propensity for learning from experience and also discovering new knowledge. Due to their excellent capability of self-learning and self-adapting, they have been successfully applied in many areas of artificial intelligence [3] and are often found to be more efficient. The operation of a FNN is depend on the below equations:

$$\mathrm{net}_j^l = \sum_{i=1}^{N_{l-1}} w_{ij}^{l-1,l} y_i^{l-1} + b_j^l, \qquad y_i^l = f(\mathrm{net}_j^l) \qquad (1)$$

where net_j^l is the sum of its weighted inputs for the jth node in the lth layer ($j = 1, \dots, Nl$), $w_{ij}^{l-1,l}$ are the weights from the ith neuron at the $(l-1)$ layer to the jth neuron at the lth layer, b_j^l is the bias of the jth neuron at the lth layer, y_i^l is the output of the jth neuron that belongs to the lth layer, and $f(\mathrm{net}_j^l)$ is the jth neuron activation function.

The main idea of training a neural network is can be formulated as a problem of nonlinear unconstrained optimization. The training a neural network is to iteratively amend its weights, in order to globally minimize a measure of difference between the actual output of the network and the desired output for all examples of the training set [4]. Therefore, mathematically the training process can be formulated as the minimization of the error function E(w), defined by the sum of square differences between the actual output of the FNN, namely,

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:05/Issue:01/January-2023 Impact Factor- 6.752 www.irjmets.com

$$E(w) = \sum_{p=1}^{p} \sum_{j=1}^{N_l} (y_i^{l-1} - t_{j,p})^2 \quad (2)$$

where $w \in R^n$ is the vector network weights and the number of patterns used in the training set represented by P. [5]-[6].

The Quasi-Newton methods revolutionized non-linear optimization in the 1960 because they avoid costly computations of Hessian matrices and perform well in practice. Several kinds of them have been proposed, but since 1970's the BFGS method has become more and more popular, and today it is accepted as the best QN method.

The (QN) methods also known as variable metric methods, provide an important family of widely applicable methods for solving smooth unconstrained problems. To minimize a function

$$f(x^*) = \min_{x \in \mathbb{R}^n} f(x), \quad (3)$$

where $f: \mathbb{R}^n \to \mathbb{R}$ is a twice continuously differentiable objective function and x is an n-dimensional vector space. These algorithms defined

$$x_{k+1} = x_k - \alpha_k H_k g_k, \quad (4)$$

where x_k is the k^{th}approximation to the minimum point, g_k is the gradient of f at x_k, H_k is an n ×n matrix that approximates the inverse Hessian of f at x_k and α_k is a positive step size parameter whose value is selected according to a rule which depends on the specific method. The approximation H_k is based on information about the inverse Hessian that is deduced from observations of previous gradients. This approximation is usually updated after each repetition.

Davidon [7] proposed the first quasi-Newton approach of this type. This technique, known as the SR1 algorithm, uses a symmetric H_k and a line search to choose α_k in order to minimize f along the line $x_k - \alpha_k H_k g_k$. The required QN algorithm has three key characteristics: 1- each $H_k$matrix is positive definite; 2- the directions of search are identical to those of the CG method for quadratic problems if $H_k = I$ (Fletcher and Reeves [8]); and 3- once more for a quadratic problem, the kth approximation $H_k$is identically equal to the inverse Hessian. It is believed that these three characteristics underpin the successful convergence characteristics the approach frequently exhibits.

Many contributions are taken into consideration when deriving new updating formulae possessing some or all of the three properties of the algorithm mentioned above, see [9].

At each iteration, the new approximation H_{k+1} is selected to take account for the new curvature information which is done by satisfying the quasi-Newton condition ($H_{k+1} y_k = v_k$). The quasi-Newton condition can be satisfied by an endless number of rank-two updates. The modifications to the Broyden (1970) one-parameter class are now being taken into consideration. The matrix H_{k+1} is defined by

$$H_{k+1} = H_k - \frac{H_k y_k y_k^T H_k}{y_k^T H_k y_k} + \frac{v_k v_k^T}{v_k^T y_k} + \theta_k v_k v_k^T, \quad (5a)$$

where

$$v_k = (y_k^T H_k y_k)^{\frac{1}{2}} \left(\frac{v_k}{v_k^T y_k} - \frac{H_k y_k}{y_k^T H_k y_k} \right). \quad (5b)$$

Different choices of the scalar parameter θ_k define different updates. This class is known as the Broyden class [10] or as the one-parameter family of updates [11]. It is also referred to as the Broyden family in [12] and [13]. It is easy to verify that any update from this class satisfies the quasi-Newton condition. Moreover, if H_k is symmetric (like the actual inverse Hessian matrix), H_{k+1} will also be symmetric; this property is called hereditary symmetry. The symmetric rank one (SR1) update method belongs to the Broyden family, is provided by the formula

$$H_{k+1} = H_k + U_k,$$

where $U_k = \frac{(v_k - H_k y_k)(v_k - H_k y_k)^T}{y_k^T (v_k - H_k y_k)}$ and it is called the correction matrix.

It was originally discovered by Davidon [14], according to [15]. One fact about SR1 is that even if H_k is positive definite matrix, the update matrix H_{k+1} may not have this property. The possibility of steps when no update satisfies the secant condition is a serious drawback. As long as the denominator is different from zero the method proceeds with a unique rank-one update. If $H_k y_k = v_k$ the only update that satisfies the secant condition

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:05/Issue:01/January-2023 Impact Factor- 6.752 www.irjmets.com

is $U_k = 0$, such that the same H_k matrix can be used for another iteration. The failure occurs when $H_k y_k = v_k$ and $y_k{}^T(v_k - H_k y_k) = 0$ at the same iteration.

II. DERIVATION OF THE MODIFIED SR1 METHOD

In this fragment we will focus on deriving the modified SR1 algorithm for the unconstrained optimization. The step length of the Barzilai-Borwein is defined as $\alpha_k^{BB} = \frac{v_k^T v_k}{y_k^T v_k}$, for more details see [16]. Now, by suppose that the Quasi-Newton equations is defined as the following formula:

$$H_{k+1}^{New}\, y_k = v_k \qquad (6)$$

In the SR1 formula, the correction term is symmetric and has the form $\alpha_k L_k L_k{}^T$ where $\alpha_k \in R$, $t \geq 0$ and $L_k \in R^{n\times n}$. Therefore, the update equation is

$$H_{k+1}^{New} = t\alpha_k^{BB} H_k + \alpha_k L_k L_k{}^T \qquad (7)$$

Our goal now is to determine α_k and L_k by multiplying both side of above equation by y_k and by using equation (6) we get to

$H_{k+1}^{New} y_k = t\alpha_k^{BB} H_k y_k + \alpha_k L_k L_k{}^T y_k = v_k$

Since $L_k{}^T y_k$ is scalar. So, we have

$$v_k - t\alpha_k^{BB} H_k y_k = (\alpha_k L_k{}^T y_k) L_k \qquad (8)$$

and hence $L_k = \frac{v_k - t\alpha_k^{BB} H_k y_k}{\alpha_k L_k{}^T y_k}$. So,

$$\alpha_k L_k L_k{}^T = \frac{(v_k - t\alpha_k^{BB} H_k y_k)(v_k - t\alpha_k^{BB} H_k y_k)^T}{\alpha_k (L_k{}^T y_k)^2} \qquad (9)$$

Now multiply (8) by $y_k{}^T$ we obtained

$y_k{}^T(v_k - H_k y_k) = (\alpha_k L_k{}^T y_k) y_k{}^T L_k$

Observe that α_k is scalar and $y_k{}^T L_k = L_k{}^T y_k$, then the above equation becomes

$$y_k{}^T(v_k - H_k y_k) = \alpha_k (L_k{}^T y_k)^2 \qquad (10)$$

By putting equation (10) in equation (9) we have,

$$\alpha_k L_k L_k{}^T = \frac{(v_k - t\alpha_k^{BB} H_k y_k)(v_k - t\alpha_k^{BB} H_k y_k)^T}{y_k{}^T (v_k - t\alpha_k^{BB} H_k y_k)} \qquad (11)$$

Then

$$H_{k+1}^{New} = t\alpha_k^{BB} H_k + \frac{(v_k - t\alpha_k^{BB} H_k y_k)(v_k - t\alpha_k^{BB} H_k y_k)^T}{y_k{}^T (v_k - t\alpha_k^{BB} H_k y_k)} \qquad (12)$$

This is the new SR1 update matrix. The SR1 algorithm with PCG method is shown below.

III. ALGORITHM OF THE NEW SR1 WITH PCG METHOD

Step 1: Set $k = 0$, select x_0, $H_0 = I$ and $\varepsilon = 10^{-5}$.

Step 2: Calculate the gradient g_k .

Step 3: Compute $d_k = -H_k g_k$.

Step 4: Determine $\alpha_k > 0$ by line search which satisfies the strong Wolfe condition.

Step 5: $x_{k+1} = x_k + \alpha_k d_k$.

Step 6: Compute g_{k+1}, if $\|g_{k+1}\| < \varepsilon$, then stop.

Else Go to Step (7).

Step 7: Calculate update H_{k+1}^{New} using (12).

Step 8: Compute the direction $d_{k+1} = -H_{k+1}^{New} g_{k+1} + \frac{y_k{}^T H_{k+1}^{New} g_{k+1}}{d_k^T y_k} d_k$

Step 9: If $|g_k^T g_{k+1}| \geq 0.2\|g_{k+1}\|^2$ go to step (3).

Else Set $k = k + 1$ and repeat from Step (4).

Theorem 1: If the new SR1 method in (12) is applied to the quadratic with Hessian $G = G^T$, then

$H_{k+1}^{new} y_k = v_k, k \geq 0.$

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:05/Issue:01/January-2023 Impact Factor- 6.752 www.irjmets.com

Proof: Multiplying both sides of equation (12) by y_k^* from the right, we have:

$$H_{k+1}^{new} y_k = t\alpha_k^{BB} H_k y_k + \frac{(v_k - t\alpha_k^{BB} H_k y_k)(v_k - t\alpha_k^{BB} H_k y_k)^T y_k}{y_k{}^T (v_k - t\alpha_k^{BB} H_k y_k)}.$$

Since $(v_k - t\alpha_k^{BB} H_k y_k)^T y_k$ and also $y_k{}^T (v_k - t\alpha_k^{BB} H_k y_k)$ are scalars.

So, we have $H_{k+1}^{new} y_k = t\alpha_k^{BB} H_k y_k + v_k - t\alpha_k^{BB} H_k y_k$.

Then, $H_{k+1}^{new} y_k = v_k$.

Hence, the proof is complete.

Theorem 2: If H_k^{New} is positive definite matrix, then theH_{k+1}^{New} which is generated by equation (12) is also positive definite matrix.

Proof: Multiplying both sides of (2.8) by y_k from the right and by $y_k{}^T$ from the left, we get

$$y_k{}^T H_{k+1}^{new} y_k = y_k{}^T t\alpha_k^{BB} H_k y_k + \frac{y_k{}^T (v_k - t\alpha_k^{BB} H_k y_k)(v_k - t\alpha_k^{BB} H_k y_k)^T y_k}{y_k{}^T (v_k - t\alpha_k^{BB} H_k y_k)}$$

After simplifying using algebraic operations, we get

$y_k{}^T H_{k+1}^{new} y_k = y_k{}^T v_k > 0.$

This completes the proof.

IV. NUMERICAL RESULTS

Numerical Results for Non-Linear Optimization Problems

This section is to evaluate how well the updated SR1 method with PCG method has been implemented. On a concatenation of test problems for unconstrained nonlinear optimization derived several computational experiments are conducted. All programs are written in the FORTRAN95 language to demonstrate the usage and effectiveness of the suggested approach with various dimensions

Table 1 compares the performance of the new SR1 approach and the original SR1 method using NI and NF. The modified SR1 technique with the precondition CG method's rate of improvement is shown in Table 2.

Table 1: Comparison between the SR1 and the new SR1 methods.

Test Function	n	SR1		New SR1	
		NI	NF	NI	NF
G-Cantrel	4	36	253	17	80
	10	36	235	16	70
	50	43	331	17	96
	100	43	331	22	122
	500	60	496	33	228
	1000	60	554	35	183
	5000	72	616	50	168
Cubic	4	15	48	12	35
	10	15	48	12	35
	50	14	48	12	35
	100	16	61	13	37
	500	17	56	13	37
	1000	16	50	13	37
	5000	16	178	13	37
Beal	4	11	29	11	27
	10	1	29	11	27
	50	12	31	12	29
	100	12	31	12	29

	500	12	31	12	29
	1000	12	31	12	29
	5000	12	31	12	29
Powell	4	30	80	28	81
	10	30	93	30	84
	50	31	95	30	84
	100	32	97	30	84
	500	33	97	30	84
	1000	33	97	30	84
	5000	33	97	30	84
G-Wolfe	4	11	24	10	20
	10	25	51	25	50
	50	42	85	25	50
	100	44	89	44	89
	500	47	95	47	95
	1000	50	101	47	95
	5000	106	294	100	220
G-Wood	4	20	50	16	40
	10	22	54	16	40
	50	23	57	16	40
	100	23	57	16	40
	500	23	57	16	4
	1000	23	57	21	52
	5000	23	57	21	52
Total		1235	5302	988	2871

Table 2: Relative efficiency between SR1 and the new SR1with PCG methods

Tools	SR1	New SR1
NI	100%	80 %
NF	100%	54.149 %

This table demonstrates that the proposed strategy improves NI by 20% and NF by 45.850%. In comparison to the normal SR1 approach, the modified SR1 method has generally improved by 32.925%.

Applications Of New SR1 Method for Training Neural Networks

In this section, the new quasi-newton and standard SR1 algorithm are compared where the input p = [0.1 0.1], and target t = [1 1]. The target error has been set to 0.01 and the maximum epochs to 3000. The network is trained until the mean squares of the errors are below the error target to decreasing value the error's function. We use the same initial weights in testing all algorithms, that were initialized randomly from range (0, 1) where the problems. The results of the training methods are present in the below figures 1and 2.

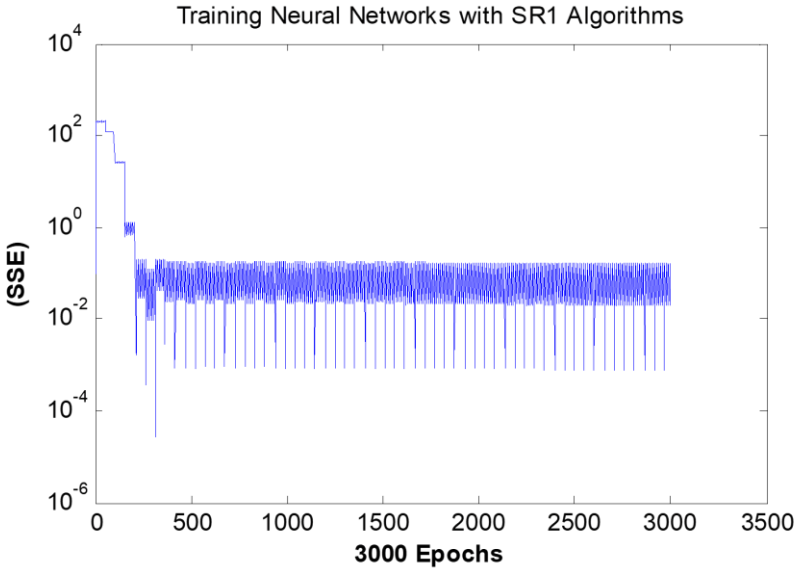

Figure 1: Performance of standard SR1 algorithm for training neural networks

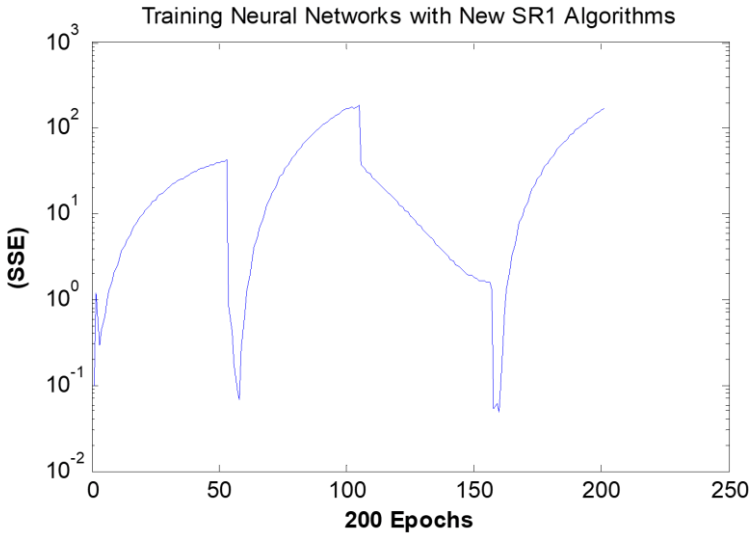

Figure 2: Performance of New SR1 algorithm for training neural networks

V. CONCLUSION

In this paper, A modification of Quasi-Newton SR1 method depend on the Barzilai-Borwein step size is suggested. The proof of its positive definiteness and the QN-condition (or the secant equation) have been proved. Our numerical results indicate that there are improvements of proposed new method techniques over standard SR1 method. Finally, the practical applicability of the new method is also explored for training neural networks.

VI. REFERENCES

[1] B. Lerner, H. Guterman, M. Aladjem, I. Dinstein, A comparative study of neural network-based feature extraction paradigms, Pattern Recognition Letters 20 (1), (1999), 7–14.

[2] C.M. Bishop, Neural Networks for Pattern Recognition, Oxford,UK, 1995.

[3] A. Hmich, A. Badri, A. Sahel, Automatic speaker identification by using the neural network, in: IEEE 2011 International Conference on Multimedia Computing and Systems (ICMCS), (2011), 1–5.

[4] Goel AK, Saxena SC, and Bhanot S, 2008. Modified Functional Link Artificial Neural Network, International Journal of Computer, electrical, Automation, Control and Information Engineering, pp. 530-538

[5] R. Fletcher and C. M. Reeves, "Function minimization by conjugate gradients," The computer journal,

vol. 7, no. 2, pp. 149–154, 1964. https://doi.org/10.1093/comjnl/7.2.149

[6] A. M. Qasim, Z. F. Salih, and B. A. Hassan, "A new conjugate gradient algorithms using conjugacy condition for solving unconstrained optimization," Indonesian Journal of Electrical Engineering and Computer Science, vol. 24, no. 3, pp. 1654–1660, 2021. DOI: http://doi.org/10.11591/ijeecs.v24.i3.pp1647-1653

[7] W. C. DAVIDON, "Variable Metric Method for Minimization," A.E.C. Research and Development Rep. ANL-5990 (Rev.), 1959.

[8] R. I. Fletcher, and C. M. Reeves, "Function Minimization by Conjugate Gradients," Comp. J., Vol. 7, pp. 149-154, 1964.

[9] C. G. BROYDEN, "Quasi-Newton Methods and Their Application to Function Minimization," Maths. of Comp., Vol. 21, pp. 368-381, 1967.

[10] P. E. Gill, W. Murray, and M. H. Wright. Practical Optimization. Academic Press, London, 2003.

[11] D. G. Luenberger. Linear and Nonlinear Programming. Second ed., Addison-Wesley, New York, 1989.

[12] R. Fletcher. Practical Methods of Optimization. Second ed., John Wiley & Sons, Chichester,1987.

[13] W. C. Davidon. Variable metric methods for minimization. Argonne National Lab Report (Argonne, IL), 1959.

[14] J. Nocedal. Theory of algorithms for unconstrained optimization. In Acta numerica, Acta Numer., pages 199–242. Cambridge Univ. Press, Cambridge, 1992.

[15] Axelsson, O., On Preconditioning and Convergence acceleration in sparse matrix problems, CERN Data Handling Division Report, 74-1, (1974).

[16] J. Barzilai and J. M. Borwein, "Two-point step size gradient methods," IMA journal of numerical analysis, vol. 8, no. 1, pp. 141–148, 1988.