Journal of University of Duhok, Vol. 22, No.1 (Pure and Eng. Sciences), Pp 45-51, 2019
https://doi.org/10.26682/sjuod.2019.22.1.7

MODIFIED CONJUGATE GRADIENT METHOD FOR TRAINING NEURAL NETWORKS BASED ON LOGISTIC MAPPING

ALAA LUQMAN IBRAHIM* and SALAH GAZI SHAREEF**
*Dept. of Mathematics, College of Science, University of Duhok, Duhok, Kurdistan Region-Iraq.
**Dept. of Mathematics, Faculty of Science, University of Zakho, Zakho, Kurdistan Region-Iraq.

(Received: October 17, 2018; Accepted for Publication: March 25, 2019)

ABSTRACT

In this paper, we suggested a modified conjugate gradient method for training neural network which assurance the descent and the sufficient descent conditions. The global convergence of our proposed method has been studied. Finally, the test results present that, in general, the modified method is more superior and efficient when compared to other standard conjugate gradient methods.

KEYWORDS: artificial neural networks, conjugate gradient, global convergence, descent and sufficient descent conditions.

1. INTRODUCTION

Artificial neural networks (ANNs) are parallel computational samples consist of processing's units and interconnected densely discriminated by an inherent propensity for learning from test and also discovering new knowledge. Because of their excellent ability of self-learning and self-adapting, they have been successfully applied in many aspects of artificial intelligence [2,6,7]. They are often found to be more active and precise than other classification techniques [3]. 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.

Training of neural networks (NNs) can be formulated as a problem of nonlinear unconstrained optimization. Therefore, the training procedure can be achieved by minimizing the error function $\boldsymbol{E}(\boldsymbol{w})$, defined by the sum of square differences between the actual output of the FNN, pointed by $\boldsymbol{o}_j^h$ and the wanted output, pointed by $\boldsymbol{t}_j^h$, relative to the appeared output, namely,

$$\boldsymbol{E}(\boldsymbol{w}) = \frac{1}{2}\sum_{h=1}^{N}\sum_{j=1}^{h}(\boldsymbol{o}_j^h - \boldsymbol{t}_j^h)^2 = \sum_{h=1}^{N}\boldsymbol{E}_h \quad (1.1)$$

where $\boldsymbol{w} \in \boldsymbol{R}^n$ is the vector network weights and the number of patterns used in the training set represented by $\boldsymbol{h}$. [8]

one of the most important iterative methods for efficiently training neural networks in scientific and engineering computation is called conjugate gradient method (CG) because of their simplicity and their very low memory requirements [4,5,12,14,17]. The conjugate gradient method produce a sequence of weights $\{\boldsymbol{w}_i\}$, is given by:

$$\boldsymbol{w}_{i+1} = \boldsymbol{w}_i + \boldsymbol{\lambda}_i\boldsymbol{p}_i \quad (1.2)$$

where $\boldsymbol{i}$ is the number of iteration generally called epoch, $\boldsymbol{\lambda}_i > 0$ is the learning rate and the search direction $\boldsymbol{p}_i$ which is computed by:

$$\boldsymbol{p}_0 = -\boldsymbol{g}_0 \text{ and } \boldsymbol{p}_{i+1} = -\boldsymbol{g}_{i+1} + \boldsymbol{\beta}_i\boldsymbol{p}_i \ \ \boldsymbol{for}\ \boldsymbol{i} \geq \boldsymbol{1}, \quad (1.3)$$

where $\boldsymbol{g}_i$ pointed to the gradient of $\boldsymbol{E}(\boldsymbol{w})$ at the point $\boldsymbol{w}_i$ and the scalar $\boldsymbol{\beta}_i$ is a known as the coefficient of (CG). The parameter $\boldsymbol{\beta}_i$ of the classical formula are determined as follows:

$$\beta_i^{PR} = \frac{g_{i+1}^T y_i}{g_i^T g_i}, \text{ Polak and Ribiere (PR)} \quad (1.4)$$

$$\beta_i^{HS} = \frac{g_{i+1}^T y_i}{p_i^T y_i}, \text{ Hestenes and Steifel (HS)} \quad (1.5)$$

$$\beta_i^{FR} = \frac{g_{i+1}^T g_{i+1}}{g_i^T g_i}, \text{ Fletcher and Reeves (FR)} \quad (1.6)$$

$$\beta_i^{CD} = \frac{g_{i+1}^T g_{i+1}}{-g_i^T p_i}, \text{ Conjugate Descent (CD)} \quad (1.7)$$

$$\beta_i^{DY} = \frac{g_{i+1}^T g_{i+1}}{p_i^T y_i}, \text{ Dai and Yuan (DY)} \quad (1.8)$$

$$\beta_i^{LS} = \frac{g_{i+1}^T y_i}{-g_i^T p_i}, \text{ Liu and Storey (LS)} \quad (1.9)$$

for the above equation see [9,18,19,20,21,22].

The globally convergence of the above conjugate gradient methods has been studied by many authors with under some different line searches [1,10,13]. To prove the convergence condition of the nonlinear CG methods, it is usually need that the step size $\boldsymbol{\lambda_i}$ should achieve the following standard strong Wolfe conditions:

$$E(w_i + \lambda_i p_i) \leq E(w_i) + \rho \lambda_i g_i^T p_i, \qquad (1.10)$$

$$\left| g(w_i + \lambda_i p_i)^T p_i \right| \leq -\sigma g_i^T p_i \qquad (1.11)$$

however, the standard Wolfe condition (1.10) and

$$g(w_i + \lambda_i p_i)^T p_i \geq \sigma g_i^T p_i \qquad (1.12)$$

where $0 < \rho < \sigma < 1$

is used to prove the convergence of many other numerical methods such as (quasi-Newton method).

In this paper, will present our modified CG training algorithm in section 2. The descent and sufficient descent conditions of our modified method are proved in section 3. The global convergences of the proposed algorithm are discussed in section 4. Some numerical results are contained in section 5. Finally, conclusions are given in the last section.

2. MODIFIED CONJUGATE GRADIENT TRAINING ALGORITHM

In this section, suggested a modified CG training algorithm by using conjugate gradient coefficient of (Fletcher and Reeves) method and logistic mapping which is used extensively [16].

From the logistic mapping and (1.6), we have

$$\beta_i^{New} = \mu \beta_i^{FR}(1 - \beta_i^{FR}) \qquad (2.1)$$

where $0 < \mu \leq 1$.

To achieve balance, we will multiply first term of (2.1) by scalar γ, we get

$$\beta_i^{New} = \mu \beta_i^{FR}(\gamma - \beta_i^{FR}), \gamma = \frac{g_i^T g_i}{\mu p_i^T y_i} \qquad (2.2)$$

and implies that

$$\beta_i^{New} = \frac{g_{i+1}^T g_{i+1}}{p_i^T y_i} - \mu\left(\frac{g_{i+1}^T g_{i+1}}{g_i^T g_i}\right)^2 \qquad (2.3)$$

or $\beta_i^{New} = \beta_i^{DY} - \mu(\beta_i^{FR})^2$.

Algorithm 1. (The modified CG algorithm)

Step(1): Initiate w_0, $gol = E_G$ and i_{max} (maximum number of epochs), set $i = 0$.

Step(2): Compute E_i and $g_i = \nabla E(w_i)$.

Step(3): If $E_i < E_G$, or $\|g_i\| \leq \varepsilon$, return $w^* = w_i$ and $E^* = E_i$ then stop

else Evaluate $s_i = w_{i+1} - w_i$ and $y_i = g_{i+1} - g_i$.

Step(4): Determine the descent direction using (1.3) and (2.3).

Step(5): Compute the learning rate λ_i to minimize $f(w_i + \lambda_i d_i)$.

Step(6): Updating new point of the weights based on Equation (1.2) and set $i = i + 1$

Step(7): If $i > i_{max}$ return "Error Goal not met"

else go to step 2.

3. THE DESCENT AND THE SUFFICIENT DESCENT CONDITIONS OF THE MODIFIED CG ALGORITHM

This section, show that the modified CG algorithm satisfies the descent and sufficient descent conditions as stated in the following theorems:

Theorem 3.1. Suppose that the sequence $\{w_i\}$ is created by (1.2). Then the search direction given by equations (1.3) and (2.3) satisfies the descent condition. i.e. $p_{i+1}^T g_{i+1} \leq 0$.

Proof: From (1.3), we have if $i = 0$

$p_0^T g_0 = -\|g_0\|^2 \leq 0$.

suppose that $p_k^T g_k \leq 0, \forall\, k = 1,2, \dots, i$.

Now, we prove the present search direction is descent direction at the iteration $(i + 1)$.

$$p_{i+1} = -g_{i+1} + \beta_i^{New} p_i. \qquad (3.1)$$

implies that

$$p_{i+1} = -g_{i+1} + \left(\frac{g_{i+1}^T g_{i+1}}{p_i^T y_i} - \mu\left(\frac{g_{i+1}^T g_{i+1}}{g_i^T g_i}\right)^2\right) p_i. \qquad (3.2)$$

By multiplying equation (3.2) by g_{i+1}^T, we have

$$g_{i+1}^T p_{i+1} = -\|g_{i+1}\|^2 + \beta_i^{DY} g_{i+1}^T p_i - \mu(\beta_i^{FR})^2 g_{i+1}^T p_i \qquad (3.3)$$

If $p_i^T g_{i+1} = 0$, then the equation (3.3) is achieve the descent condition i.e.

$g_{i+1}^T p_{i+1} = -\|g_{i+1}\|^2 \leq 0$.

However, if $p_i^T g_{i+1} \neq 0$. We conclude

$$-\|g_{i+1}\|^2 + \beta_i^{DY} g_{i+1}^T p_i \leq 0, \qquad (3.4)$$

because the DY method satisfies the descent condition.

Since $g_{i+1}^T p_i \leq p_i^T y_i$ and clearly $p_i^T y_i > 0$, $\mu \in (0,1]$ and $(\beta_i^{FR})^2 \geq 0$

so, the third term of equation (3.3) can be written as

$-\mu(\beta_i^{FR})^2 g_{i+1}^T d_i \leq -\mu(\beta_i^{FR})^2 p_i^T y_i \leq 0$

Finally, we have

$g_{i+1}^T p_{i+1} = -\|g_{i+1}\|^2 + \beta_i^{DY} g_{i+1}^T p_i - \mu(\beta_i^{FR})^2 g_{i+1}^T p_i \leq 0.$ ■

Theorem 3.2. Suppose that p_{i+1} is produced by equations (1.3) and (2.3), and λ_i is obtained from equations (1.10) and (1.11), then the sufficient descent condition is satisfied, i.e.

$$g_{i+1}^T p_{i+1} \leq -c\|g_{i+1}\|^2$$

Proof: From equation (3.4). Therefore, the equation (3.3) can be written as follows:

$$g_{i+1}^T p_{i+1} \leq -\mu\left(\frac{g_{i+1}^T g_{i+1}}{g_i^T g_i}\right)^2 g_{i+1}^T p_i \qquad (3.5)$$

Since $g_{i+1}^T p_i \leq p_i^T y_i$, will be in the form

$$g_{i+1}^T p_{i+1} \leq -\left(\mu \frac{g_{i+1}^T g_{i+1}}{(g_i^T g_i)^2} p_i^T y_i\right)\|g_{i+1}\|^2 \qquad (3.6)$$

we obtained $g_{i+1}^T p_{i+1} \leq -c\|g_{i+1}\|^2$,

where $c = \mu \frac{g_{i+1}^T g_{i+1}}{(g_i^T g_i)^2} p_i^T y_i$. ∎

4. THE GLOBAL CONVERGENCE OF THE MODIFIED CG ALGORITHM

To prove the global convergence result of the modified CG method, we need the following assumptions. [11]

Assumption 1. The level set $S = \{w : w \in R^n, E(w) \leq E(w_0)\}$ is bounded. i.e. $\exists\, B > 0$ such that

$$\|w\| \leq B, \forall\, w \in S \qquad (4.1)$$

Assumption 2. In a neighborhood $\Omega \in S$, E is differentiable and its gradient g is Lipschitz continuous, i.e. $\exists\, L > 0$ such that

$$\|g(w) - g(w_i)\| \leq L\|w - w_i\|, \forall\; w, w_i \in \Omega \qquad (4.2)$$

From Assumptions 1 and 2, $\exists\, M > 0$ such that

$$\|g(w)\| \leq M, \quad \forall\; w \in S. \qquad (4.3)$$

Lemma 4.1 [15]. Assume that the Assumptions 1 and 2 holds and the sequence $\{w_i\}$ is created by the equations (1.2) and (1.3), where p_i satisfy the descent condition and λ_i is determined by (1.10) and (1.11). If

$$\sum_{i\geq 1} \frac{1}{\|p_i\|^2} = \infty. \qquad (4.4)$$

Then

$$\lim_{i\to\infty} \inf \|g_i\| = 0. \qquad (4.5)$$

If E is a uniformly convex function, $\exists\, \vartheta > 0$ such that:

$$\left(g(x) - g(y)\right)^T (x - y) \geq \vartheta\|x - y\|^2 \in \Omega. \quad (4.6)$$

We can rewrite (4.6) in the following manner:

$$y_i^T s_i \geq \vartheta\|s_i\|^2. \qquad (4.7)$$

Theorem 4.1. Assume that Assumptions 1 and 2 holds. If any iteration of the equations (1.2) and (1.3), where β_i^{New} is defined by equation (2.3) and λ_i satisfies the strong Wolfe line search conditions (1.10) and (1.11), then

$$\lim_{i\to\infty} \inf \|g_{i+1}\| = 0$$

Proof: By using contradiction, we assume their exist appositive constant such that

$$\|g_i\| \geq \omega, \forall\, i \geq 0. \qquad (4.8)$$

Then, from (1.3) and (2.3), it follows that:

$$p_{i+1} = -g_{i+1} + \beta_i^{New} p_i$$

which is can be written as

$$\|p_{i+1}\| \leq \|g_{i+1}\| + |\beta_i^{New}|\|p_i\|, \qquad (4.9)$$

and $|\beta_i^{New}| = \left|\frac{g_{i+1}^T g_{i+1}}{p_i^T y_i} - \mu\left(\frac{g_{i+1}^T g_{i+1}}{g_i^T g_i}\right)^2\right|$

using equation (4.7), we obtained that

$$|\beta_i^{New}| \leq \left|\frac{\lambda_i \|g_{i+1}\|^2}{\vartheta\|s_i\|^2}\right| + \left|\mu \frac{\|g_{i+1}\|^4}{\|g_i\|^4}\right|. \qquad (4.10)$$

Then

$$|\beta_i^{New}| \leq \frac{\lambda_i M^2}{\vartheta\|s_i\|^2} + \frac{\mu M^4}{\omega^4}. \qquad (4.11)$$

By combining the equations (4.9) and (4.11), we have

$$\|p_{i+1}\| \leq M + \left(\frac{\lambda_i M^2}{\vartheta\|s_i\|^2} + \frac{\mu M^4}{\omega^4}\right)\|p_i\|. \qquad (4.12)$$

Implies that

$$\|p_{i+1}\| \leq M + \left(\frac{M^2}{\vartheta\|s_i\|^2} + \frac{\mu M^4}{\lambda_i \omega^4}\right)\|s_i\|. \qquad (4.13)$$

Since, $\|s_i\| = \|w - w_i\|$,

$D = max\{\|w - w_i\|\}, \forall\, w, w_i \in R\}$.

Hence (4.13) becomes

$$\|p_{i+1}\| \leq M + \left(\frac{M^2}{\vartheta D} + \frac{\mu M^4 D}{\lambda_i \omega^4}\right) = \varphi.$$

leading to (4.4). So, from Lemma 4.1. Hence (4.5) holds and contradicting (4.8).

EXPERIMENTAL RESULTS

In this section, we examine the implementation of the modified method. The comparative tests include familiar nonlinear problems with various dimensions $4 \leq n \leq 5000$. Our algorithms has converged as soon as $\|g_{i+1}\| \leq 10^{-5}$ and Powell condition $|g_i^T g_{i+1}| \geq 0.2\|g_{i+1}\|^2$ is used to restart. All algorithms implemented with a cubic interpolation which uses function and gradient values. The algorithms are written in FORTRAN 95 language. Table (1) shows that the numerical results of the modified (CG) method is more effective than standard (DY) method with respect to the number of iterations (NI) and the number of functions evaluation (NF).

In addition to that, we will offer experimental numerical results in order to study and assess the performance of the modified (CG) method in classical artificial intelligence problems (Continuous Function Approximation).

In particular, we investigate the performance of DY method compare with our modified method during five times of the implementation the program. The implementation has been carried out by using MATLAB (2013a) and the MATLAB Neural Network T**oolbox version 8.1 for conjugate gradient.**

5.1 Problem: (Continuous Function Approximation)

Consider the approximation of the continuous trigonometric function as:

$\boldsymbol{f(x) = sin(x) + cos(3x)}$, ***WHERE*** $\boldsymbol{x \in [0, \pi]}$.

The network is trained to approximate the function and the network is trained until the mean squares of the errors becomes less than the error goal 1e-10 within the limit of 1000 epochs.

Tables 3: offer the performance comparison of the methods DY and modified (CG) for the continuous function approximation problem. All algorithms display excellent likelihood (100%) of successful training for network using the same initial weights. Thus, computational cost is possibly the most appropriate indicator for measuring the efficiency of the methods. The performs of modified (CG) method is better than the DY method in terms of the number of epochs, time, Gradient and Step size.

Table (1): Comparison between the (modified and DY) methods

Test Function	N	CG (DY)		Modified (CG)	
		NI	NF	NI	NF
Miele	4	36	115	34	110
	100	45	156	42	143
	500	53	188	42	143
	1000	60	222	48	178
	5000	66	257	48	178
Non-Diagonal	4	24	63	24	63
	100	29	79	29	79
	500	29	214	27	139
	1000	29	79	26	74
	5000	F	F	21	61
Fred	4	8	22	7	20
	100	8	22	7	20
	500	8	22	7	20
	1000	8	22	7	20
	5000	8	22	8	22
Beal	4	11	28	10	26
	100	12	30	10	26
	500	12	30	10	26
	1000	12	30	10	26
	5000	12	30	10	26
Central	4	18	127	18	129
	100	20	153	20	151
	500	23	192	22	186
	1000	23	192	22	186
	5000	24	205	22	186
Sum	4	5	27	5	27
	100	14	80	14	80
	500	20	100	20	98
	1000	27	132	25	135
	5000	32	151	28	125
Osp.	4	8	44	8	44
	100	52	180	52	182
	500	138	439	130	403
	1000	196	607	181	566
	5000	555	1857	535	1816

Rosen	4	30	82	17	49
	100	30	82	17	49
	500	30	82	17	49
	1000	30	82	17	49
	5000	30	82	17	49
Total		**1817**	**6649**	**1614**	**5959**

Note: The fail result in standard CG is considered a twice value of modified (CG) results.

Table (2): Percentage of improving the modified method

Tools	CG (PR)	Modified (CG)
NI	100%	88.8277%
NF	100%	89.6225%

As we observe from Table 2 the NI and NF of the DY method are about 100%. That means, the modified method has improvement of 11.1722% and 10.3775% compared with standard method in NI and NF respectively. Generally, the modified (CG) method was improved by 10.77485% compared with DY method.

Table (3): Comparing the Performance of modified method with Standard DY method for training neural network

Methods	No. Running	Epochs	CPU time(s)/Epoch	Gradient	Step size
DY	1	1000	00:04	0.00383	0.00100
	2	1000	00:03	0.00149	0.000408
	3	1000	00:03	0.00277	0.00100
	4	1000	00:03	0.00167	0.00430
	5	1000	00:03	0.00345	0.00100
Modified	1	191	00:01	0.00695	0:00
	2	464	00:01	0.00152	0:00
	3	904	00:03	0.00211	0:00
	4	761	00:02	0.00565	0:00
	5	273	00:01	0.0115	0:00

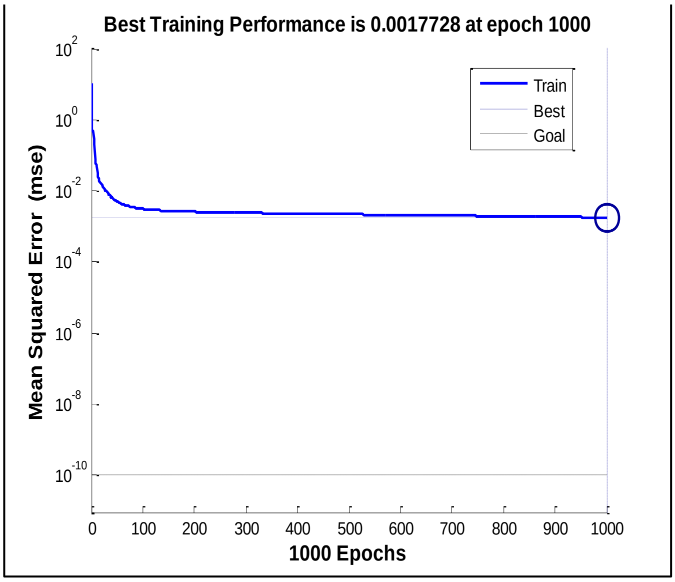

Fig (1): Performance of DY method for training neural networks

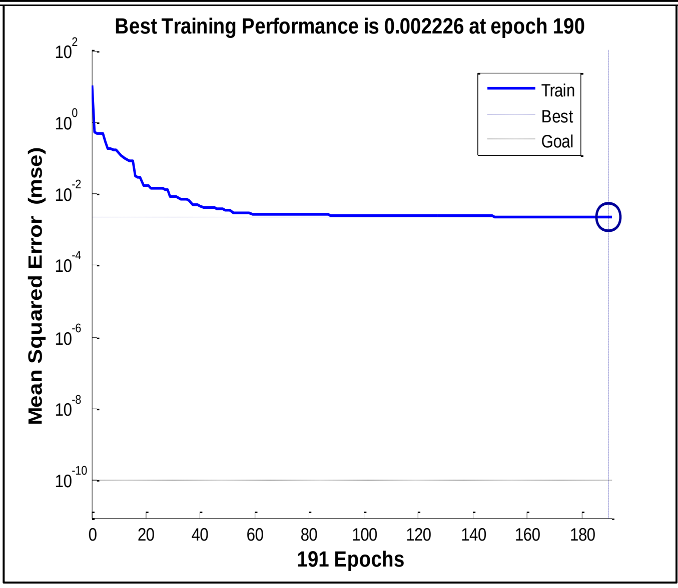

Fig (2): Performance of Modified method for training neural networks

5. CONCLUSION

This paper, proposed a modified (CG) method which consists of (Fletcher and Reeves) method and by using logistic mapping. The search direction $\boldsymbol{p_i}$ produced by our proposed method satisfies both (the descent and sufficient descent) conditions. The global convergence of the modified (CG) method has been proved. Furthermore, we used the modified (CG) method for training neural networks. Depend on the numerical experiments, we found that modified method is more effective than the classical CG method, leading to a stable and faster convergence.

REFERENCE

[1] A. Al-Baali, Descent property and global convergence of the Fletcher–Reeves method with inexact line search, IMA J. Numer. Anal. 5 (1985) 121–124.

[2] 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.

[3] 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.

[4] C. Charalambous, Conjugate gradient algorithm for efficient training of artificial neural networks, IEEE Proceedings 139 (3), (1992), 301–310.

[5] C. C. Peng, G. D. Magoulas, Adaptive nonmonotone conjugate gradient training algorithm for recurrent neural networks, in: 19th IEEE International Conference on Tools with Artificial Intelligence, (2008), 374–381.

[6] C. H. Wu, H. L. Chen, S. C. Chen, Gene classification artificial neural system, International Journal on Artificial Intelligence Tools 4 (4), (1995), 501–510.

[7] C. M. Bishop, Neural Networks for Pattern Recognition, Oxford, (1995).

[8] D. E. Rumelhart, G. E. Hinton, R. J. Williams, Learning internal representations by error propagation, in: D. E. Rumelhart, J. McClell and (Eds.), Parallel Dis-tributed Processing: Explorations in the Micro structure of Cognition, Cambridge, MA, (1986), 318–362.

[9] E. Polak, G. Ribiere, Note sur la convergence de directions conjuguees, Rev. Francaise Informat Recherche Operationelle 3, (1969), 35–43.

[10] G. Zoutendijk, Nonlinear programming computational methods, in: J. Abadie (Ed.),

https://doi.org/10.26682/sjuod.2019.22.1.7

Integer and Nonlinear Programming, North-Holland, Amsterdam, (1970), 37–86.

[11] I. Jusoh, M. Mamat and M. Rivaie, A new edition of conjugate gradient methods for large-scale unconstrained optimization, International Journal of Mathematical Analysis, Vol. 8, No. 46, (2014), 2277 – 2291.

[12] I. E. Livieris, P. Pintelas, An improved spectral conjugate gradient neural network training algorithm, International Journal on Artificial Intelligence Tools 21 (1), (2012).

[13] J. Sun, J. Zhang, Convergence of conjugate gradient methods without line search, Annals of Operations Research 103, (2001), 161–173.

[14] J. Wang, W. Wu, M. Zurada, Deterministic convergence of conjugate gradient method for feedforward neural networks, Neurocomputing 74, (2011), 2368–2376.

[15] K. Sugiki, Y. Narushima, and H. Yabe, Globally convergent three–term conjugate gradient methods that use secant conditions and generate descent search directions for unconstrained optimization, J. Optim. Theory Appl. 153, (2012), 733–757.

[16] H. Lu, H. Zhang, L. Ma, A new optimization algorithm based on chaos, Zhejiang University, Hangzhou 310027, China, (2005).

[17] M.F. Moller, A scaled conjugate gradient algorithm for fast supervised learning, Neural Networks 6, (1993), 525–533.

[18] M.R. Hestenes, E. Stiefel, Methods for conjugate gradients for solving linear systems, Journal of Research of the National Bureau of Standards 49, (1952), 409–436.

[19] R.Fletcher, C. Reeves, Function minimization by conjugate gradients, Comput. J.7, (1964), 149–154.

[20] R.Fletcher, Practical method of optimization, Unconstrained optimization, 1, John Wiley & Sons, New York, (1987).

[21] Y.H. Dai, Y. Yuan, A nonlinear conjugate gradient with a strong global convergence property, SIAMJ. Optim. 10, (1999), 177–182.

[22] Y.Liu, C. Storey, Efficient generalized conjugate gradient algorithms part1: Theory, J. Comput. Appl. Math.69, (1992), 129–137.