EUROPEAN JOURNAL OF PURE AND APPLIED MATHEMATICS
Vol. 17, No. 4, 2024, 2692-2705
ISSN 1307-5543 – ejpam.com
Published by New York Business Global

Enhanced Conjugate Gradient Method for Unconstrained Optimization and Its Application in Neural Networks

Dlovan Haji Omar[1], Alaa Luqman Ibrahim[1,*], Masoud Muhammed Hassan[2], Bayda Ghanim Fathi[1], Diman Abdulqader Sulaiman[1]

[1] *Department of Mathematics, College of Science, University of Zakho, Kurdistan Region, Iraq*
[2] *Department of Computer Science, College of Science, University of Zakho, Kurdistan Region, Iraq*

Abstract. In this study, we present a novel conjugate gradient method specifically designed for addressing with unconstrained optimization problems. Traditional conjugate gradient methods have shown effectiveness in solving optimization problems, but they may encounter challenges when dealing with unconstrained problems. Our method addresses this issue by introducing modifications that enhance its performance in the unconstrained setting. We demonstrate that, under certain conditions, our method satisfies both the sufficient descent criteria and establishes global convergence, ensuring progress towards the optimal solution at each iteration. Moreover, we establish the global convergence of our method, providing confidence in its ability to find the global optimum. To showcase the practical applicability of our approach, we apply this novel method to a dataset, applying a feed-forward neural network value estimation for continuous trigonometric function value estimation. To evaluate the efficiency and effectiveness of our modified approach, we conducted numerical experiments on a set of well-known test functions. These experiments reveal that our algorithm significantly reduces computational time due to its faster convergence rates and increased speed in directional minimization. These compelling results highlight the advantages of our approach over traditional conjugate gradient methods in the context of unconstrained optimization problems.

2020 Mathematics Subject Classifications: 90C90, 90C30, 90C06, 49M37

Key Words and Phrases: Optimization, conjugate gradient, neural networks, convergence analysis

---

*Corresponding author.
DOI: https://doi.org/10.29020/nybg.ejpam.v17i4.5354

Email addresses: dilovan.omar@uoz.edu.krd (D. H. Omar), alaa.ibrahim@uoz.edu.krd (Alaa L. Ibrahim), masoud.hassan@uoz.edu.krd (M. M. Hassan), bayda.fathi@uoz.edu.krd (B. G. Fathi), dimanabdulqader90@gmail.com (D. A. Sulaiman)

1. Introduction

The conjugate gradient (CG) method's ability to effectively solve a large number of unconstrained optimization problems in a short amount of time and with fewer iterations is one of its main advantages. Because CG methods don't rely on the Hessian matrix or its approximation, they have lower computational costs and storage requirements. Furthermore, the CG method satisfies the descent condition and exhibits fast global convergence. This method's unique quality is its simplicity, which makes the algebraic processes and writing computer code easy to understand. As a result, the approach shows promise and competence in handling large-scale unconstrained minimization problems [2]. This study will consider the following model

$$minf\,(x) \qquad\qquad \forall\, x \in R^n, \tag{1}$$

where $f\colon\ R^n \ \rightarrow\ R$ is a continuously differentiable function, $g_k = \nabla f\ (x_k)$ is a gradient at point x_k are solved using this expression

$$x_{k+1} = x_k + \alpha_k d_k \ \ k = 0, 1, 2, \ldots \tag{2}$$

where x_k is a solution in current iteration and x_k is a new iteration point, $\alpha_k > 0$ is a step size determined by some line searches. d_k is a search direction which is determined by:

$$d_0 = -g_0 \ \ and \ \ d_{k+1} = -g_{k+1} + \beta_k d_k \ \ for\ k \geq 1, \tag{3}$$

where β_k is an important parameter. The different choices for the parameter β_k correspond to different CG methods. The Fletcher-Reeves (FR) method [12], the Polak-Ribitere-Polyak (PRP) method [29, 30], the Dai-Yuan (DY) method [8], the Conjugate Descent (CD) method [9], and others [20], are some of the well-known CG methods used in the study of optimization problems to obtain the solution of the models.

A number of researchers [4], have discussed the convergence of the well-known classical CG methods, such as Hestenes–Stiefel (HS) [16], Polak–Ribiere–Polyak (PRP) [13, 30], Fletcher–Reeves (FR) [11], Liu–Storey (LS) [26], conjugate descent (CD) [10], and Dai–Yuan (DY) [7].

Although the PRP formula is thought to be the most efficient in terms of numerical computation, there is no guarantee that it will converge under multiple line searches [31]. This disadvantage prompted several changes to the PRP parameter. For in-depth discussions on the PRP technique, see Yuan et al. [36], Andrei [4], and Zhang et al. [37].

Recently, there has been a significant trend among researchers to develop the best and fastest optimization methods, driven by the increasing number of real-world applications that rely on optimization techniques. These applications span various fields, including machine learning, engineering design, logistics, finance, and healthcare, where efficient and accurate optimization is crucial for improving performance, reducing costs, and solving complex problems. The quest for more effective optimization methods is fueled by the need to handle larger datasets, higher-dimensional problems, and more intricate models, making optimization a key area of research and innovation see [1, 6, 15, 19, 22, 24, 34].

Due to their exceptional capacity for self-learning and self-adaptation, artificial neural networks (ANNs) have been employed successfully in numerous machine learning applications for decades [5, 17, 35]. They have been heavily utilized in fields like robotics, security, and self-driving automobiles. Because of their ability to solve problems with flexibility and support for parallel processing, they are frequently found to be more active and precise than other classification techniques [25, 33]. While there are a number of proposed training techniques, feed forward neural networks (FNNs) are one of the most well-known and frequently applied training styles across a wide range of domains and applications.

In recent times, there has been a growing trend towards utilizing the conjugate gradient method for training neural networks, with various new approaches and techniques being introduced to enhance its efficiency and applicability in machine learning tasks [18, 21, 23].

Motivated by the previously mentioned trend, this paper presents an improved conjugate gradient method and uses feed-forward neural networks to solve different datasets.

The remaining sections of this paper are organized as follows: The new conjugate parameter's derivation process and a thorough explanation of its algorithm are presented in Section 2. Section 3 discusses the global convergence properties of the method, along with the sufficient descent condition. The numerical results of applying the new method to a number of benchmark test problems are presented in Section 4. Lastly, the approach is expanded to handle issues with feed-forward neural networks

2. Derivation of New Method and Algorithm

In this section, a new conjugate gradient method for unconstrained optimizations is present based on suggested vector y_k^* and Perry conjugate gradient parameter [28],

$$\beta_k{}^{Perry} = \frac{{g_{k+1}}^T(y_k - \nu_k)}{{d_k}^T y_k}. \tag{4}$$

Consider the vector y_k^* is defined by

$$y_k^* = (1 + \sigma + \lambda) y_k,$$

where σ and λ are given by:

$$\sigma = \frac{1}{1 + e^{-\|y_k\|}}, \lambda = \frac{\mu\sqrt{m}}{\|g_k\|}\left(1 + \|g_{k+1}\|\right) [3]$$

with m representing machine accuracy, taking $\left(m = 1*\ 10^{-16}\right)$, and $\mu = 0.1$. Substituting these into the expression for y_k^* gives:

$$y_k^* = y_k + \frac{1}{1 + e^{-\|y_k\|}} y_k + \frac{\mu\sqrt{m}}{\|g_k\|}\left(1 + \|g_{k+1}\|\right) y_{k'} \tag{5}$$

Now, by replacing y_k by y_k^* in equation (4), we obtained a new parameter

$$\beta_k^{\text{New}} = \frac{{g_{k+1}}^T\left(y_k^* - v_k\right)}{{d_k}^T y_k^*} = \frac{{g_{k+1}}^T\left((1 + \sigma + \lambda) y_k - v_k\right)}{(1 + \sigma + \lambda){d_k}^T y_k}.$$

or

$$\beta_k^{\text{New}} = \frac{{g_{k+1}}^T y_k}{d_k^T y_k} - \frac{{g_{k+1}}^T v_k}{(1+\sigma+\lambda){d_k}^T y_k}. \tag{6}$$

Algorithm 2.1: New Method
Require: Initial point $x_0 \in \mathbb{R}^n$.
Step 1: Set $k = 0$, $g_0 = \nabla f(x_0)$, $d_0 = -g_0$. If $g_0 = 0$, then stop.
Step 2: Compute α_k to minimize $f(x_{k+1})$ (i.e., $f_{k+1} \leq f_k$) using cubic line search.
Step 3: Update $x_{k+1} = x_k + \alpha_k d_k$.
Step 4: Compute $g_{k+1} = \nabla f(x_{k+1})$. If $||g_{k+1}|| \leq 10^{-5}$, then stop.
Step 5: Compute β_k from equation (6).
Step 6: Set $d_{k+1} = -g_{k+1} + \beta_k^{\text{New}} d_k$.
Step 7: If $k = n$ or if $\left|g_k^T g_{k+1}\right| \geq 0.2||g_{k+1}||^2$ is satisfied, go to Step 3.
Otherwise, set $k = k + 1$ and go to Step 4.

Theorem 1: Let the sequences $\{x_k\}$ and $\{d_k\}$ be generated by Equations (2), (3), and (6). Then the search direction d_{k+1} satisfies the sufficient descent condition:

$$g_{k+1}^T d_{k+1} \leq -c\|g_{k+1}\|^2$$

Proof: From Equations (3) and (6), we have

$$d_{k+1} = -g_{k+1} + \left(\frac{{g_{k+1}}^T y_k}{{d_k}^T y_k} - \frac{{g_{k+1}}^T \nu_k}{(1+\sigma+\lambda)\, {d_k}^T y_k}\right) d_k. \tag{7}$$

Now, by multiplying the above equation by g_{k+1}^T from both sides, we get

$$g_{k+1}^T d_{k+1} = -\|g_{k+1}\|^2 + \beta_k^{\text{HS}} g_{k+1}^T d_k - \alpha_k \frac{(g_{k+1}^T d_k)^2}{(1+\sigma+\lambda) d_k^T y_k}. \tag{8}$$

If $d_k^T g_{k+1} = 0$, then $g_{k+1}^T d_{k+1} = -\|g_{k+1}\|^2 \leq 0$. Then the proof is complete.

Now, if $d_k^T g_{k+1} \neq 0$, we know that the first two terms of the above equation correspond to the search direction of the HS method, so it is less than or equal to zero. And it is obvious that α_k, $1+\sigma+\lambda$, ${d_k}^T y_k$, and $(g_{k+1}^T d_k)^2$ are positive. Therefore, we have

$$g_{k+1}^T d_{k+1} = -\|g_{k+1}\|^2 + \beta_k^{\text{HS}} g_{k+1}^T d_k - \alpha_k \frac{(g_{k+1}^T d_k)^2}{(1+\sigma+\lambda) d_k^T y_k} \leq 0$$

, so, the above equation can be written as

$$g_{k+1}^T d_{k+1} \leq -\left(\alpha_k \frac{(g_{k+1}^T d_k)^2}{(1+\sigma+\lambda) d_k^T y_k \|g_{k+1}\|^2}\right) \|g_{k+1}\|^2. \tag{9}$$

Then,

$$g_{k+1}^T d_{k+1} \leq -c\|g_{k+1}\|^2,$$

where

$$c = \alpha_k \frac{(g_{k+1}^T d_k)^2}{(1+\sigma+\lambda) d_k^T y_k \|g_{k+1}\|^2}.$$

This completes the proof.

To prove the global convergence of the new method, we need to make the following fundamental assumptions about the objective function in this section.

Assumption [38]:

(i) The level set $\delta = \{x \mid f(x) \leq f(x_0)\}$ is bounded.

(ii) In some neighborhood N of δ, f is continuously differentiable, and its gradient is Lipschitz continuous with a Lipschitz constant $\delta > 0$, i.e.,

$$\|g(x) - g(y)\| \leq \delta \|x - y\| \quad \forall x, y \in \delta.$$

(iii) From the above assumptions, there exists a positive constant b such that:

$$\|g(x)\| \leq b \quad \forall x \in \delta. \tag{10}$$

(iv) If f is a uniformly convex function, there exists a constant $\vartheta > 0$ such that:

$$(g(x) - g(y))^T (x - y) \geq \vartheta \|x - y\|^2 \quad \forall x, y \in \Omega, \tag{11}$$

can be rewrite (11), in the following manner:

$$y_k^T v_k \geq \vartheta ||v_k||^2. \tag{12}$$

Lemma 1: **[37]** Let the above assumptions hold. Consider the methods (2), and (3), where d_{k+1} is a descent direction and α_k satisfies the standard Wolfe line search. If

$$\sum_{k \geq 1} \frac{1}{\|d_k\|^2} = \infty.$$

Then,

$$\lim_{k \to \infty} \inf \|g_k\| = 0.$$

Theorem 2: If above assumptions are true and the corresponding sequences $\{x_k\}$, $\{d_k\}$, $\{g_k\}$, $\{\alpha_k\}$ are generated by the new method (2), (3), and (6), then

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

Proof: From Equations (3), and (6), we have

$$\begin{aligned} \|d_{k+1}\| &\leq \|g_{k+1}\| + (\left| \frac{g_{k+1}^T y_k}{d_k^T y_k} - \frac{g_{k+1}^T v_k}{(1+\sigma+\lambda) d_k^T y_k} \right|) \|d_k\|, \\ &\leq \|g_{k+1}\| + \left| \frac{g_{k+1}^T y_k}{d_k^T y_k} \right| \|d_k\| + \left| \frac{g_{k+1}^T v_k}{(1+\sigma+\lambda) {d_k}^T y_k} \right| \|d_k\|. \end{aligned} \tag{13}$$

Given that $g_{k+1}^T v_k \leq \alpha_k d_k^T y_k$ and considering the Lipschitz condition $\|y_k\| \leq L\|v_k\|$ with $L > 0$, along with the use of equation (12), it follows that

$$\|d_{k+1}\| \leq b + \frac{\alpha_k b L \|v_k\|}{\vartheta \|v_k\|^2} \|d_k\| + \frac{\alpha_k}{(1+\sigma+\lambda)} \|d_k\|. \tag{14}$$

Since, $\|v_k\| = \|x - x_k\|, \quad D = \max\{\|x - x_k\|\}, \forall x, x_k \in R\}$. Hence Equations (14), becomes

$$\|d_{k+1}\| \leq b + \left(\frac{bL}{\vartheta} + \frac{D}{(1+\sigma+\lambda)} \right) = \beta$$

$$\Rightarrow \sum_{k \geq 1} \frac{1}{\|d_{k+1}\|^2} \geq \sum_{k \geq 1} \frac{1}{\beta^2} = \infty$$

which completes the proof.

3. Numerical results

The performance of the new method was evaluated using standard test functions as described by Andrei [2], with initial points ranging from 5 to 5000. The assessment was conducted by comparing the new method against the classical Hestenes-Stiefel (HS) conjugate gradient method. Key metrics for comparison include the number of iterations (NOI) required for convergence and the number of function evaluations (NOF) during the optimization process. All programs were implemented in FORTRAN 95, and a cubic interpolation line search method, which uses both function and gradient information, was employed. The results are summarized in Tables 1 and 2.

Table 1: The results of the new method with standard method.

Test Function	N	HS		New	
		NOI	NOF	NOI	NOF
Mile	4	28	85	27	83
	10	31	102	27	83
	100	33	114	30	98
	500	40	147	32	105
	1000	46	176	33	129
	5000	54	211	33	129

Powell	4	38	108	31	87
	10	38	108	31	87
	100	40	122	33	102
	500	41	124	33	102
	1000	41	124	33	102
	5000	41	124	33	102
Wolfe	5	11	24	11	24
	10	32	65	32	65
	100	49	99	49	99
	500	52	105	48	99
	1000	70	141	43	91
	5000	165	348	102	227
G-Cantrel	4	22	159	14	47
	10	22	159	15	58
	100	22	159	17	74
	500	23	171	17	74
	1000	23	171	21	110
	5000	28	248	23	126
OSP.	4	8	45	8	43
	10	13	58	13	53
	100	49	185	45	160
	500	112	353	100	316
	1000	156	473	150	471
	5000	256	774	250	772
Wood	5	30	68	28	62
	10	30	68	28	62
	100	30	68	28	62
	500	30	68	28	62
	1000	30	68	28	62
	5000	30	68	28	62
Rosen	4	30	83	29	80
	10	30	83	29	80
	100	30	83	29	80
	500	30	83	29	80
	1000	30	83	29	80
	5000	30	83	29	80
Edger	4	5	14	5	14
	10	5	14	5	14
	100	5	14	5	14
	500	6	16	5	14
	1000	6	16	6	16
	5000	6	16	6	16

Shallow	4	8	21	8	21
	10	8	21	8	21
	100	8	21	8	21
	500	8	21	8	21
	1000	9	24	9	24
	5000	9	24	9	24
Total		2027	6410	1758	5190

Table 2: The improvement percentage of the new method compared to the standard method.

Tools	HS (Standard)	New (Proposed)
NOI	100%	86.7292%
NOF	100%	80.967%

Table 2 presents the improvement percentages of the new method compared to the standard HS method across two key metrics: the number of iterations (NOI) and the number of function evaluations (NOF). The new method achieves a reduction of approximately 13.27 % in the number of iterations and 19.03 % in the number of function evaluations, demonstrating its enhanced efficiency over the standard HS method.

4. Application in Neural Networks

Artificial Intelligence (AI) is designed to emulate human brain functionality [24]. Among the various branches of AI, artificial neural networks (ANNs) have gained significant popularity [14]. ANNs are employed in tasks such as classification, optimization, and prediction of given datasets to produce appropriate results or outputs. The training and testing phases are pivotal in ANNs, allowing the network to discern patterns within the data, even when the dataset is incomplete [19, 27, 32].

In this section, the performance of the HS method is evaluated in comparison with the new method for training neural networks. The algorithms were implemented using MATLAB (2013a) and the MATLAB Neural Network Toolbox version 8.1 for conjugate gradient optimization. The network is trained until the mean squared errors fall below the specified error target, indicating a decreasing trend in the error function. For consistency, the same initial weights were used across all algorithms, which were randomly initialized within the range (0, 1). The problems addressed in this evaluation are:

(i) Input: $\mathbf{P} = [-1, -1, 2, 2; 0, 5, 0, 5]$ and target: $\mathbf{T} = [-1, -1, 1, 1]$. The target error was set to 1×10^{-10} and the maximum number of epochs was set to 1000.

(ii) Input: Continuous trigonometric function $f(x) = \sin(x) + \cos(2x)$, where $x \in [0, \pi]$. The target error was set to 1×10^{-10} and the maximum number of epochs was set to 1000.

The results of the training methods are presented in Tables 3 and Figures 1, 2, 3, and 4 below.

Table 3: Comparing the Performance of new method with standard method for training neural network

	Methods	No. Running	Epochs	CPUtime(s)	Gradient
Problem1	Standard	1	1000	0:00:04	0.000376
		2	34	0:00:00	0.000239
		3	41	0:00:00	0.000173
		4	21	0:00:00	0.000209
		5	38	0:00:00	0.000157
	New CG	1	6	0:00:00	0.153
		2	4	0:00:00	0.121
		3	6	0:00:00	0.0649
		4	6	0:00:00	0.0499
		5	4	0:00:00	0.382
Problem 2	Standard	1	296	0:00:01	0.000244
		2	1000	0:00:02	0.00334
		3	69	0:00:00	0.000186
		4	1000	0:00:02	0.00219
		5	1000	0:00:02	0.00374
	New CG	1	2	0:00:00	0.89
		2	2	0:00:00	0.975
		3	2	0:00:00	0.525
		4	2	0:00:00	0.697
		5	3	0:00:02	0.332

Table 3 compares the performance of the new (CG) method with the standard method in training neural networks. The new CG method significantly reduces the number of epochs needed to converge, often achieving results in just 2 to 6 epochs compared to the standard method's 1000 epochs in several cases. Both methods show minimal CPU time per epoch, but the new CG method generally achieves higher gradient values in fewer epochs, indicating more efficient optimization. Overall, the new CG method outperforms the standard method in terms of faster convergence and computational efficiency.

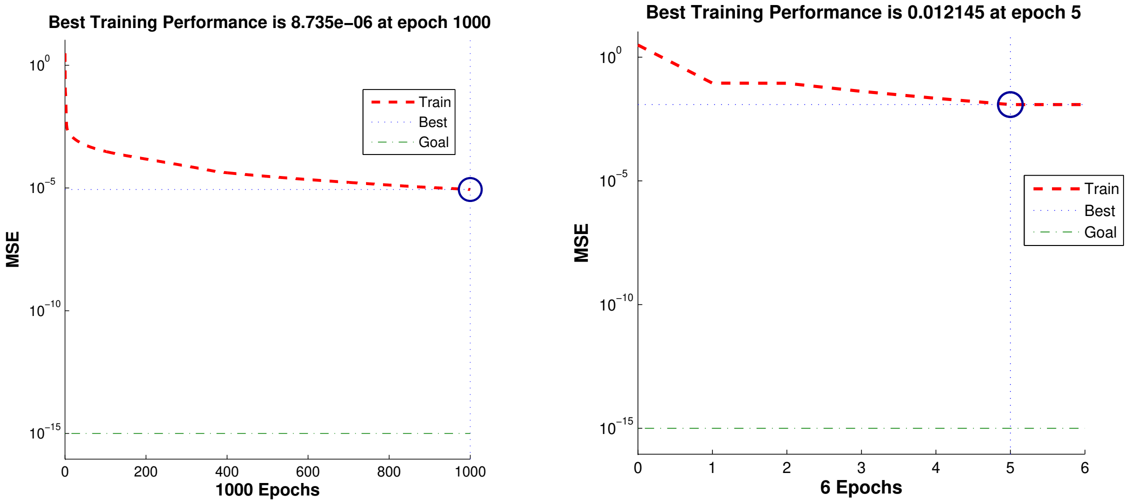

Figure 1: Validation Performance HS method (left) and new method (right), of problem 1

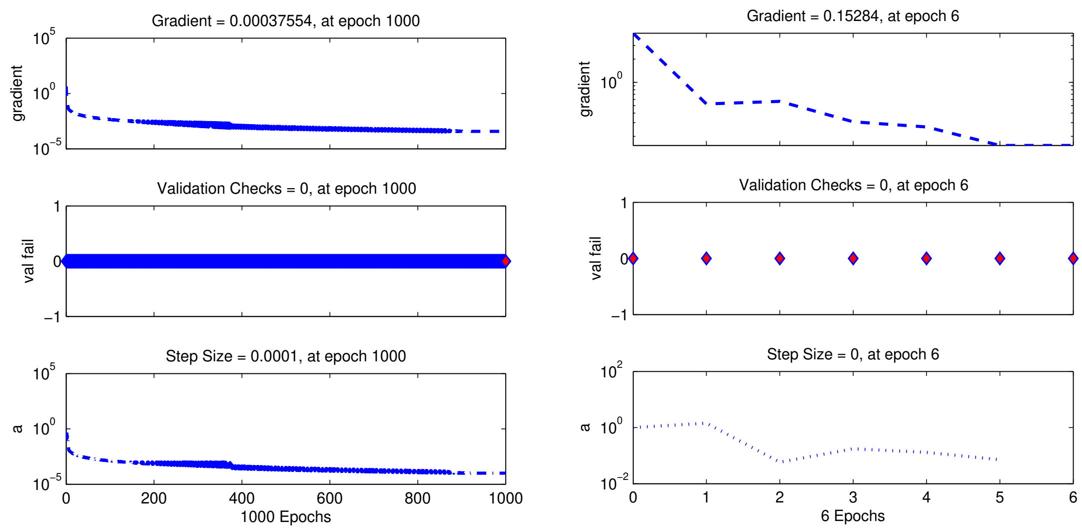

Figure 2: Training Performance HS method (left) and new method (right), of problem 1.

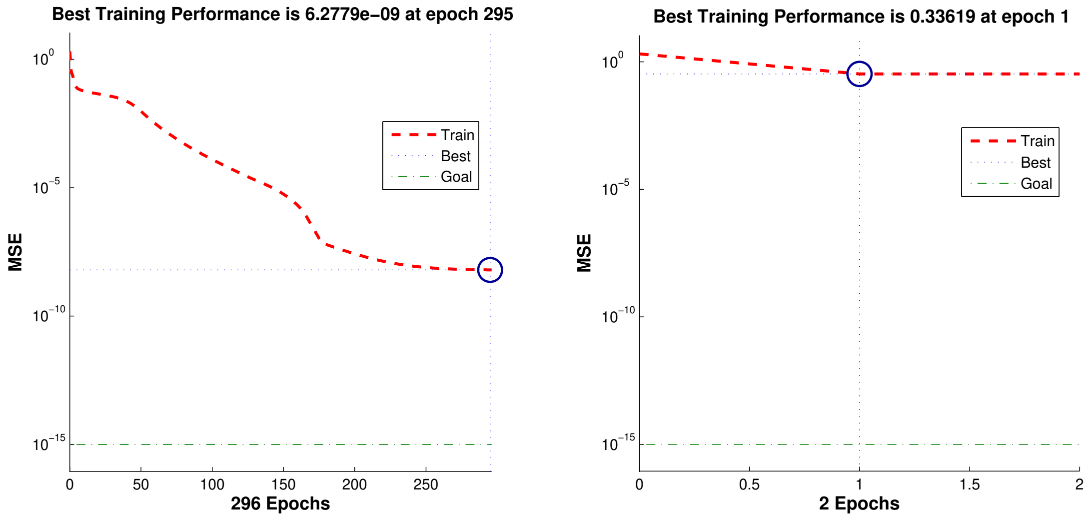

Figure 3: Validation Performance HS method (left) and new method (right), of problem 2.

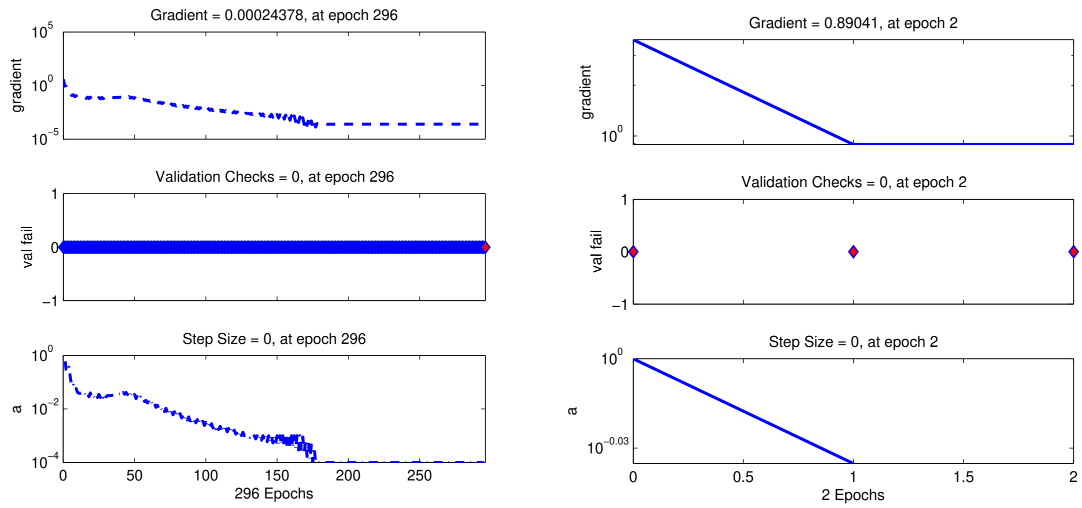

Figure 4: Training Performance HS method (left) and new method (right), of problem 2.

5. Conclusion

In this paper, we have presented a modified conjugate gradient method specifically designed for addressing unconstrained optimization problems. Our new method builds upon the classical conjugate gradient method while incorporating enhancements that preserve the sufficient descent properties. Moreover, we have explored the global convergence of our method under appropriate conditions, providing reassurance of its ability to converge to the optimal solution. Through extensive numerical computations and experiments, we have demonstrated the promising performance of our modified conjugate gradient method. The results showcase its effectiveness in solving unconstrained optimization problems, surpassing the classical HS training function in terms of reducing the number of iterations and CPU time. Furthermore, we have extended the application of our method to feed-forward neural networks, demonstrating its compatibility with complex tasks such as value estimation for continuous trigonometric functions. The outcomes of these experiments have highlighted the advantages of our approach, showcasing its ability to enhance the efficiency and effectiveness of neural network training. Overall, the introduction of our modified conjugate gradient method presents a significant advancement in the field of unconstrained optimization. By combining the desirable properties of the classical conjugate gradient method with enhancements tailored for unconstrained problems, our approach offers improved convergence and computational efficiency. These findings have implications for a wide range of domains where optimization plays a crucial role, paving the way for more efficient and effective problem-solving methodologies.

References

[1] A. Alhawarat, Z. Salleh, H. Alolaiyan, H. El Hor, and S. Ismail. A three-term conjugate gradient descent method with some applications. *J. Inequalities Appl.*, 2024.

[2] N. Andrei. An unconstrained optimization test functions collection. *Adv. Model. Optim.*, 10:147–161, 2008.

[3] N. Andrei. Accelerated conjugate gradient algorithm with finite difference hessian/vector product approximation for unconstrained optimization. *J. Comput. Appl. Math.*, 230:570–582, 2009.

[4] N. Andrei. *Nonlinear Conjugate Gradient Methods for Unconstrained Optimization.* Springer Optimization and its application, 2020.

[5] C. M. Bishop. *Neural networks for pattern recognition.* 1995.

[6] Z. Chen, H. Shao, P. Liu, G. Li, and X. Rong. An efficient hybrid conjugate gradient method with an adaptive strategy and applications in image restoration problems. *Appl. Numer. Math.*, 204:362–379, 2024.

[7] Y. Dai, J. Han, G. Liu, D. Sun, H. Yin, and Y.-X. Yuan. Convergence properties of nonlinear conjugate gradient methods. *SIAM J. Optim.*, 10:345–358, 2000.

[8] Y. H. Dai and Y. Yuan. A nonlinear conjugate gradient method with a strong global convergence property. *SIAM J. Optim.*, 10:177–182, 1999.

[9] R. Fletcher. *Practical Methods of Optimization.* John Wiley & Sons, New York, 2013.

[10] R. Fletcher. *Practical Methods of Optimization.* John Wiley & Sons, 2020.

[11] R. Fletcher and M. J. D. Powell. A rapidly convergent descent method for minimization. *Comput. J.*, 6:163–168, 1963.

[12] R. Fletcher and C. M. Reeves. Function minimization by conjugate gradients. *Comput. J.*, 7:149–154, 1964.

[13] T. Ginsburg. The conjugate gradient method in extremal problems. *USSR Comput. Math. Math. Phys.*, 9:94–112, 1969.

[14] W. Hager and H. Zhang. A survey of nonlinear conjugate gradient methods. *Pacific J. Optim.*, 2:35–58, 2006.

[15] A. Hassan, B. Jabbar, and A. Bayati. A new class of nonlinear conjugate gradient method for solving unconstrained minimization problems. In *Presented at the (2019)*.

[16] M. R. Hestenes and E. Stiefel. Methods of conjugate gradients for solving linear systems. *J. Res. Natl. Bur. Stand. (1934)*, 49:409, 1952.

[17] A. Hmich, A. Badri, and A. Sahel. Automatic speaker identification by using the neural network. In *2011 International Conference on Multimedia Computing and Systems*, pages 1–5, 2011.

[18] A. L. Ibrahim and M. G. Mohammed. A new three-term conjugate gradient method for training neural networks with global convergence. *Indones. J. Electr. Eng. Comput. Sci.*, 28:551–558, 2022.

[19] A. L. Ibrahim and M. G. Mohammed. A new conjugate gradient for unconstrained optimization problems and its applications in neural networks. *Indones. J. Electr. Eng. Comput. Sci.*, 33:93–100, 2024.

[20] A. L. Ibrahim, M. A. Sadiq, and S. G. Shareef. A new conjugate gradient coefficient for unconstrained optimization based on dai-liao. *Sci. J. Univ. Zakho.*, 7:34–36, 2019.

[21] A. L. Ibrahim and S. G. Shareef. Modified conjugate gradient method for training neural networks based on logistic mapping. 2:45–51, 2019.

[22] X. Jiang, H. Yang, J. Yin, and W. Liao. A three-term conjugate gradient algorithm with restart procedure to solve image restoration problems. *J. Comput. Appl. Math.*, 424:115020, 2022.

[23] K. Kamilu, M. I. Sulaiman, A. Muhammad, A. W. Mohamad, and M. Mamat. Performance evaluation of a novel conjugate gradient method for training feed forward neural network. *Math. Model. Comput.*, 2023.

[24] Y. A. Laylani, K. K. Abbo, and H. M. Khudhur. Training feed forward neural network with modified fletcher-reeves method. *J. Multidiscip. Model. Optim.*, 1:14–22, 2018.

[25] B. Lerner, H. Guterman, M. Aladjem, and I. Dinstein. A comparative study of neural network based feature extraction paradigms. *Pattern Recognit. Lett.*, 20:7–14, 1999.

[26] Y. Liu and C. Storey. Efficient generalized conjugate gradient algorithms, part 1: Theory. *J. Optim. Theory Appl.*, 69:129–137, 1991.

[27] I. E. Livieris and P. Pintelas. Performance evaluation of descent cg methods for neural network training. *Hell. Eur. Res. Comput. Math. its Appl.*, 2009.

[28] A. Perry. Technical note—a modified conjugate gradient algorithm. *Oper. Res.*, 26:1073–1078, 1978.

[29] B. T. Polyak. The conjugate gradient method in extremal problems. *Zh. Vychisl. Mat. Mat. Fiz.*, 9:807–821, 1969.

[30] E. P. Ribiere and G. Note sur la convergence de methods de directions conjuguees. *Rev. Fr. d'Informatique Rech. Oper.*, 16:35–43, 1969.

[31] M. Rivaie, M. Mamat, L. W. June, and I. Mohd. A new conjugate gradient coefficient for large scale nonlinear unconstrained optimization. *Int. J. Math. Anal.*, 6:1131–1146, 2012.

[32] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning representations by back-propagating errors. *Nature*, 323:533–536, 1986.

[33] C. J. Spoerer, T. C. Kietzmann, J. Mehrer, I. Charest, and N. Kriegeskorte. Recurrent neural networks can explain flexible trading of speed and accuracy in biological vision. *PLOS Comput. Biol.*, 16:1–27, 2020.

[34] X. Wang, G. Yuan, and L. Pang. A class of new three-term descent conjugate gradient algorithms for large-scale unconstrained optimization and applications to image restoration problems. *Numer. Algorithms*, 93, 2022.

[35] C. H. Wu. *Gene classification artificial neural system*, pages 71–88. Academic Press, 1996.

[36] G. Yuan, Z. Wei, and X. Lu. Global convergence of bfgs and prp methods under a modified weak wolfe–powell line search. *Appl. Math. Model.*, 47:811–825, 2017.

[37] L. Zhang, W. Zhou, and D.-H. Li. A descent modified polak–ribiere–polyak conjugate gradient method and its global convergence. *IMA J. Numer. Anal.*, 26:629–640, 2006.

[38] G. Zoutendijk. *Nonlinear programming computational methods*, pages 37–86. North-Holland, Amsterdam, 1970.