An Improved Gradient Descent Method for Optimization of Supervised Machine Learning Problems

Dada Ibidapo Dare
Department of Computer Science,
Federal University of Agriculture, Abeokuta, Nigeria

Akinwale Adio Taofiki
Department of Computer Science,
Federal University of Agriculture, Abeokuta, Nigeria

Onashoga Adebukola S.
Department of Computer Science,
Federal University of Agriculture, Abeokuta, Nigeria

Osinuga Idowu A.
Department of Mathematics,
Federal University of Agriculture, Abeokuta, Nigeria

ABSTRACT
Gradient descent method is commonly used as an optimization algorithm for some machine learning problems such as regression analysis and classification problems. This method is highly applicable for real life of yearly demand-price commodity, agricultural products and Iris flowers. This study proposed the combination of Dai-Yuan (DY) and Saleh and Mustafa (SM) conjugate gradient methods for the optimization of supervised machine learning problems. Experiments were conducted on combined DY and SM with well-known conjugate gradient methods using a fixed learning rate. The efficiency of the combined methods and existing models was evaluated in term of number of iterations and processing time. The experimental results indicated that the combined conjugate gradient method had the better performance in term of number of iterations and processing time.

Keywords
Conjugate gradient method, machine learning, regression analysis, data classification.

1. INTRODUCTION
Machine learning technique is a key research area that gives computers ability to learn and classify data as well as predicting output for new data. A good number of effective optimization methods had been proposed for the performance and efficiency of machine learning methods. The goal of optimization is to find the possible solutions to a problem in order to make the best decision. Such decision is to minimize cost or to maximize profit in which both (cost and profit) can be expressed as a function. Therefore, optimization is the process of finding the best solutions that give the maximum or the minimum value of a function [8].
The general mathematical model of optimization problems can be written in the form:

$$optimize\ z = f(x_1, x_2 \ldots, xn) \quad (1)$$

Where optimize stands for minimum or maximum of the function $f: R^n \to R$ which is assumed to be continuously differentiable.

The optimal solution of a maximization problem is:

$$\max_{x \in S} f(x) \quad (2)$$

while the optimal solution of a minimization problem is:

$$\min_{x \in S} f(x) \quad (3)$$

Where S a subset of R^n is the feasible set.

Optimization methods in the field of machine learning are faced with different difficulties such as global convergence, poor computational performance, processing time, and so on. Therefore, in this study a modified optimization method was proposed from the existing ones for solving some machine learning techniques within the shortest computation time with good convergence property.

2. OVERVIEW OF RELATED METHODS AND NEW HYBRID METHOD
The conjugate gradient method (CGM) is an optimization method that is applied in some specific areas. CGM can be used to solve linear equations and nonlinear optimization problems [12].
The general unconstrained optimization problem is given as

$$min\{f(x) | x \in R^n\} \quad (4)$$

where $f: R^n \to R$ is continuously differentiable, $f(x)$ is an objective function and $x \in R^n$ is a vector with independent variables. The objective of the CGM is to find the minimum value of a function for unconstrained optimization problem and low memory usage [11], [5]. The CGM is commonly solved by iterative method which is defined as follows:

$$x_{k+1} = x_k + \propto_k d_k, \quad k = 1,2, \ldots \quad (5)$$

where $x_k$is the current iterative point, $\propto_k$ is the step size (also known as the learning rate) and d_k is the search direction of conjugate gradient method. The step size can be solved in two ways of the exact and the inexact line search. The search direction of conjugate gradient method d_k can be defined as follows:

$$d_k = \begin{cases} -g_k & k=0 \\ -g_k + \beta_k d_{k-1} & k=1,2,\ldots \end{cases} \quad (6)$$

where β_k is conjugate gradient (CG) coefficient of $f(x)$ and $g_k$is the gradient at point x_k. $\beta_k \in R$ is a scalar while $g_k = \nabla f(x_k)$ is at point x_k.

Some well-known CG methods and their modifications have been proposed by many researchers. Dai [12] made a review on CG methods and divided into early and descent CG methods.

The early conjugate gradient methods includes Hestenes-Stiefel (HS) method which was first introduced by Hestenes-Stiefel in 1952 for solving linear CG method [12], where the $\beta_k$is given as:

$$\beta_k^{HS} = \frac{g_k^T(g_k - g_{k-1})}{d_{k-1}^T(g_k - g_{k-1})} \quad (7)$$

The drawback of this method is that it can only be used to solve linear equation [6].
Fletcher-Reeves (FR) method was presented in 1964 by Fletcher and Reeves [3] which proposed the first nonlinear CG method. CG parameter is as follows:

$$\beta_k^{FR} = \frac{g_k^T g_k}{\|g_{k-1}\|^2} \quad (8)$$

The drawback of this method is that it may fall into some circles of tiny steps which may sometimes be very slow in practical computation to converge [12].

In 1969, Polak, Ribiere and Polyak proposed another conjugate gradient parameter which performs better than the Fletcher-Reeves (FR) method for many optimization problems because it can recover automatically once small step is generated [12]. There method is as follows:

$$\beta_k^{PR} = \frac{g_k^T(g_k - g_{k-1})}{\|g_{k-1}\|^2} \quad (9)$$

Descent conjugate gradient methods includes Conjugate descent (CD) method, [4] proposed the CD methods in his monograph, with β_k as

$$\beta_k^{CD} = \frac{-g_k^T g_k}{d_{k-1}^T g_{k-1}} \quad (10)$$

Other than the FR, PRP and HS methods, the CD method can ensure the descent property of each search condition provided that the strong Wolfe conditions are used.

The Dai-Yuan (DY) method: To enforce a descent direction in case of the standard Wolfe line search, [2] proposed a new conjugate gradient method, where the β_k is given as:

$$\beta_k^{DY} = \frac{g_k^T g_k}{d_{k-1}^T(g_k - g_{k-1})} \quad (11)$$

Some other modifications of conjugate gradient methods are:

Liu-Storey (LS) conjugate gradient method: proposed in 1992 by [8]. β_k is as

$$\beta_k^{LS} = \frac{-g_k^T(g_k - g_{k-1})}{d_{k-1}^T g_{k-1}} \quad (12)$$

Rivaie-Mustafa-Ismail-Leong (RMIL) conjugate gradient method: proposed in 2012 [9]. β_k is as

$$\beta_k^{RMIL} = \frac{g_k^T(g_k - g_{k-1})}{\|d_{k-1}\|^2} \quad (13)$$

Kamilu et al (KMAR) conjugate gradient method in 2015 [7]. β_k is as

$$\beta_k^{KMAR} = \frac{g_k^T(g_k - g_{k-1})}{g_{k-1}^T(g_k + g_{k-1})} \quad (14)$$

Sulaiman-Mustafa (SM1) conjugate gradient method in 2018 [10]. β_k is also given as

$$\beta_k^{SM1} = \frac{g_k^T\left(g_k - \frac{\|g_k\|}{\|g_{k-1}\|}d_{k-1} - d_{k-1}\right)}{d_{k-1}^T(g_k - g_{k-1})} \quad (15)$$

The main objectives of the study is to propose a modified conjugate gradient method for solving some machine learning techniques within the shortest computation time with good convergence property. The proposed conjugate gradient method was also applied to some supervised machine learning models such as linear, multiple and logistic regressions which were evaluated in terms of number of iterations and processing time.

New Conjugate Gradient Coefficient

A new conjugate gradient coefficient, β_k based on combination of DY (11) and SM1 (15) methods had been named HCG35 conjugate gradient which is as follows:

$$\beta_k^{HCG35} = \frac{\frac{g_k^T g_k}{d_{k-1}^T(g_k - g_{k-1})} + \frac{g_k^T\left(g_k - \frac{\|g_k\|}{\|g_{k-1}\|}d_{k-1} - d_{k-1}\right)}{d_{k-1}^T(g_k - g_{k-1})}}{2} \quad (16)$$

The HCG35 algorithm is as follows:
Step 1: Given x_0, set k=0.
Step 2: compute $\beta_k$based on β_k^{HCG35}as in (16).
Step 3: compute search direction d_k based on (6).
If $\|g_k\| = 0$, then stop, otherwise go to step 4.
Step 4: compute step size α_k
Step 5: update a new point by using (5)
Step 6: stopping criteria.
If $f(x+1) < f(x)$ and $\|g_k\| < \epsilon$, then stop.
Otherwise go to step 1 with k = k + 1.

3. CONVERGENT ANALYSIS

By using theoretical proofs, the proposed method should be able to satisfy the convergence analysis. This section will delve deeper into the sufficient descent properties of this method.

Sufficient Descent Condition

From 16,

$$\beta_k^{HCG35} = \frac{g_k{}^T g_k}{d_{k-1}{}^T (g_k - g_{k-1})} + \frac{g_k{}^T \left(g_k - \frac{\|g_k\|}{\|g_{k-1}\|} d_{k-1} - d_{k-1}\right)}{d_{k-1}{}^T (g_k - g_{k-1})} \Big/ 2$$

$$= \frac{\frac{\|g_k\|^2}{d_{k-1}{}^T g_k - d_{k-1}{}^T g_{k-1}} + \frac{g_k{}^T g_k - g_k{}^T \|g_k\| d_{k-1} - d_{k-1}}{d_{k-1}{}^T g_k - d_{k-1}{}^T g_{k-1}}}{2}$$

$$= \frac{\frac{\|g_k\|^2}{d_{k-1}{}^T g_k - d_{k-1}{}^T g_{k-1}} + \frac{\|g_k\|^2 - g_k{}^T \|g_k\| d_{k-1} - d_{k-1}}{d_{k-1}{}^T g_k - d_{k-1}{}^T g_{k-1}}}{2}$$

$$\beta_k^{HCG35} \leq \frac{\frac{\|g_k\|^2}{d_{k-1}{}^T g_k - d_{k-1}{}^T g_{k-1}} + \frac{\|g_k\|^2 - d_{k-1}(g_k{}^T \|g_k\| + 1)}{d_{k-1}{}^T g_k - d_{k-1}{}^T g_{k-1}}}{2} \quad (17)$$

From Salleh and Alhawarat, 2016, can reduce

$$d_{k-1}{}^T g_k - d_{k-1}{}^T g_k - d_{k-1}{}^T g_{k-1} = 0 \quad (18)$$

Then (17) becomes:

$$\beta_k^{HCG35} \leq \frac{\frac{\|g_k\|^2}{-d_{k-1}{}^T g_{k-1}} + \frac{\|g_k\|^2}{-d_{k-1}{}^T g_{k-1}}}{2} \leq \frac{\|g_k\|^2}{\|g_{k-1}\|^2} \quad (19)$$

Using the descent properties

$$d_{k-1}{}^T g_{k-1} \leq -c\|g_{k-1}\|^2 \quad \forall\, k \geq 0, c > 0 \quad (20)$$

There is a need to show that the method satisfies (20)

From the search direction d_k, this gives

$$\|g_k\|^2 + \beta_{k-1}{}^{HCG35} d_{k-1}{}^T g_{k-1} > 0 \quad (21)$$

The following theorem would be used to show that proposed method satisfy (20)

Theorem 1: for a search direction d_k and a CG coefficient $\beta_{k-1}{}^{HCG35}$ in a CG method, the condition $d_{k-1}{}^T g_{k-1} = -c\|g_{k-1}\|^2$ holds for all k > 0.

Proof: by induction

If k = 1, then $d_0{}^T g_0 = -c\|g_0\|^2$, thus (20) holds true.

Now, show for k > 1, (20) also holds.

From (21) this gives

$$d_{k-1}{}^T g_{k-1} = \|g_k\|^2 + \beta_{k-1}{}^{HCG35} d_{k-1}{}^T g_{k-1}$$

$$d_{k-1}{}^T g_{k-1} = -\|g_k\|^2 \quad (22)$$

From (22), reduce (20) holds for all k > 0.

Global convergence

Subsequent assumptions are used to prove the global convergence properties.

Assumption 1

i. $f(x)$ is constrained below on the set R^n which is continuous and differentiable in a neighborhood N of the level set $\ell = \{x \in R^n \,|f(x) \leq f(x_0)\}$ with the initial point x_0.

ii. The gradient $g(x)$ is Lipschitz continuous in N, $\exists\, L > 0$, such that

$$\|g(x) - g(y)\| \leq L\|x - y\| \;\; \forall\, x, y \in N .$$

From assumption 1, the Zoutendijk condition was derived as in Lemma 1 which has been proven by Zoutendijk, 1970.

Lemma 1

Suppose that Assumption 1 is true, let any CG method of the form (5) and (6), $0 > \propto_k < 1$. Therefore, the next Zoutendijk condition holds

$$\sum_{k=0}^{\infty} \frac{g_k{}^T d_k{}^2}{\|d_k\|^2} < \infty \quad (23)$$

Theorem 2: Suppose that Assumption 1 is true, let any CG method of the form (5) and (6), $0 > \propto_k < 1$ and the coefficient β_k is obtained by (81), Then

$$\lim_{k\to\infty} \|g_k\| = 0 \quad (24)$$

Proof: by contradiction

Suppose Theorem 2 does not hold, there is existence of a constant $\emptyset$ such that

$$\|g_k\| \geq \emptyset \quad (25)$$

d_k can be rewritten as

$$d_{k+1} + \beta_{k+1} = \beta_{k+1}{}^{HCG35} d_k \quad (26)$$

Square both side of (26),

$$\|d_k\|^2 = (\beta_{k+1}{}^{HCG35})^2 \|d_k\|^2 - 2 g_{k+1}{}^T d_{k+1} - \|g_{k+1}\|^2 \quad (27)$$

Divide through (27) by $(g_{k+1}{}^T d_{k+1})^2$,

$$\frac{\|d_k\|^2}{(g_{k+1}{}^T d_{k+1})^2} = \frac{(\beta_{k+1}{}^{HCG35})^2 \|d_k\|^2}{(g_{k+1}{}^T d_{k+1})^2} - \frac{2 g_{k+1}{}^T d_{k+1}}{(g_{k+1}{}^T d_{k+1})^2} - \frac{\|g_{k+1}\|^2}{(g_{k+1}{}^T d_{k+1})^2}$$

$$= \frac{(\beta_{k+1}{}^{HCG35})^2 \|d_k\|^2}{(g_{k+1}{}^T d_{k+1})^2} - \left(\frac{1}{\|g_{k+1}\|} - \frac{\|g_{k+1}\|^2}{(g_{k+1}{}^T d_{k+1})^2}\right) + \frac{1}{\|g_{k+1}\|^2}$$

$$\leq \frac{(\beta_{k+1}{}^{HCG35})^2\|d_k\|^2}{(g_{k+1}{}^T d_{k+1})^2} + \frac{1}{\|g_{k+1}\|^2} \quad (28)$$

Appling (19), this gives

$$\frac{\|d_{k+1}\|^2}{(g_{k+1}{}^T d_{k+1})^2} \leq \frac{1}{\|g_{k+1}\|^2}$$

This implies that

$$\frac{\|d_{k+1}\|^2}{(g_{k+1}{}^T d_{k+1})^2} \leq \sum_{i=1}^{k} \frac{1}{\|g_{i+1}\|^2}$$

$$\frac{g_{k+1}{}^T d_{k+1}}{\|d_{k+1}\|^2} \geq \frac{\emptyset^2}{k} \quad (29)$$

From (29) and (23)

$$\sum_{k=1}^{\infty} \frac{g_{k+1}{}^T d_{k+1}}{\|d_{k+1}\|^2} = \infty$$

Since this contradicts lemma 1. Thus, proof accomplished.

4. NUMERICAL STUDY

4.1 Parameters setting

All algorithms in this study were implemented on a PC workstation with Intel® Core™ i5-5020U CPU @ 2.20GHz, 8GB of RAM, 358.27GB hard disk capacity. Python programming language is used to implement this study. This is due to the availability of vast amount of open source python-based libraries and packages such as Numpy, Pandas, Matplotlib,and so on.

4.2 Description of benchmark functions

4.2.1 Problem 1 [1].

The data adopted is composed of the yearly demand and price of a commodity as sample of the data is illustrated in table 1.

Table 1: Problem 1

Price	1	2	2	2.3	2.5	2.6	2.8	3	3.3	3.5
Demand	5	3.5	3	2.7	2.4	2.5	2	1.5	1.2	1.2

Formulation of the model

From the data set in table1, there exists a linear relationship between the demand and the price.

Hypothesis: $net = \beta_0 x_0 + \beta_1 x_1$ (30)

$$h_\beta(x) = f(net) \quad (31)$$

$where\ f\ is\ a\ linear\ activation\ function.$

$$h_\beta(x) = net \quad (32)$$

Parameters: $\beta = \beta_0,\ \beta_1$ (33)

$$\text{Cost function: } J(\beta) = \frac{1}{2m}\sum_{i}^{m}\left(h_\beta(x^{(i)}) - y^{(i)}\right)^2 \quad (34)$$

4.2.2 Problem 2

The data used was collected from Food and Agriculture Organization of the United Nations (fao.org/faostat/en/#data/QC). The sample of the data is shown in table 2.

Table 2: Problem 2

Year	Annual Rainfall (AR)	Area (million Hectare)	Food Price Index (FPI)	Production (1000 Tons)	Yield (MT./Hectare)
2000-01	1120.2	2199	42.86	1979	2
2001-02	981.4	2117	67.14	1651	1
2002-03	1278	2185	58.00	1757	1
2003-04	1085.9	2210	50.43	1870	1
2004-05	1185	2348	71.57	2000	1
2005-06	1133	2494	74.29	2140	1
2006-07	120.7	2725	72.86	2546	1
2007-08	986.4	2451	73.71	2008	1
2008-09	1184.2	2412	60.57	2632	2
2009-10	1075	1840	69.43	2234	2
2010-11	974.7	2433	60.00	2818	2
2011-12	1075	2269	76.86	2906	2
2013-14	1133.9	2931	81.00	3041	2
2014-15	1180.2	3082	86.43	4082	2
2015-16	1075	3122	98.14	3941	2
2016-17	972.8	3170	89.71	4410	2
2017-18	1189.9	3600	93.71	4662	2
2018-19	1212.2	3600	97.14	4788	2

Normalization of data collection

Studying the data, the input variables for year have dynamic range which differs by orders of magnitude and thus suggests that a suitable normalization should be applied so that the

transformed variables all cover the same range. Thus, the linear scaling transformation was used to normalize the collected data.

$$Z_i = \frac{y_i - \min(y)}{\max(y) - \min(y)}$$

where $y = (x_1, \dots, x_n)$ is the data set. Z_i is now the i[th] normalized data.

Formulation of the model

The multiple linear regression analysis will be carried out by using annual Rainfall (x_1), area under cultivation (x_2), food price index (x_3) data as independent variable and rice yield (y) data as dependent variable.

The model is shown below:

$$net = \beta_0 + \beta_i x_{i1} + \beta_i x_{i2} + \beta_3 x_{i3} + \varepsilon_i \quad (35)$$

$$h_\beta(x) = f(net) \quad (36)$$

$$where\ f\ is\ a\ linear\ activation\ function$$

$$h_\beta(x) = net \quad (37)$$

Objective function is given as follows:

$$J(\beta) = \frac{1}{2m}\sum_{i}^{m}\left(h_\beta(x) - y^{(i)}\right)^2 \quad (38)$$

4.2.3 Problem 3

The problem was obtained from the Iris flowers dataset. The Iris flowers data involves predicting the flower species given measurements of the iris flowers. The variable names are:

1. Length (x_1)
2. Width(x_2)
3. Class(iris Setosa (1) and iris virginica (0))

The model formulation:

Hypothesis:

$$net = \beta^T x = x_0\beta_0 + \beta_1 x_1 + \beta_2 x_2 \quad where x_0 = 1$$

$$h_\beta(x) = f(net) \quad (39)$$

$$where\ f\ is\ log - Sigmoid\ activation\ function$$

$$h_\beta(x) = \frac{1}{1 + e^{-net}} \quad (40)$$

Parameters:$\beta = \beta_0,\ \beta_1,\ \beta_2$

Cost function:

$$J(\theta) = \frac{1}{m}\left[\sum_{i=1}^{m} -y^i log\left(h\theta(x^i)\right) + (1 - y^i) log\left(1 - h\theta(x^i)\right)\right] \quad (42)$$

The proposed conjugate gradient method is employed to solve these problems in order to assess the performance in comparison with some well-known conjugate gradient methods and gradient descent method.

4.3 Discussion of Results

This sub-section is devoted to the application of the proposed conjugate gradient methods (HCG35) in comparison with gradient descent, Hestenes-Stiefel (HS), Fletcher-Reeves (FR), Polak - Ribiere - Polyak (PRP), Conjugate descent (CD) and Dai-Yuan (DY) methods.

4.3.1 Test problem 1

For test problem 1, five experiments were performed with the same initial guess in order to evaluate the efficiency of the method. The stop criteria used in the experiment for all the algorithms convergence is assumed if $\|g_k\| \leq \varepsilon$ where $\varepsilon = 10^{-5}$ and the symbol "-" was used to represent that the algorithm does not converge. The initial point for the parameters $\beta_0\ and\ \beta_1$is set to be (0, 0) using a fixed learning rate $\propto = 0.01$. The results are depicted in Table 3.

Table 3: Experimental results for problem 1

Methods	Number of iteration	Processing time	Optimal values
HCG35	60	0.34	0.03179
FR	196	0.8	0.0318
HS	1909	8.15	0.03199
GD	3533	18.6	0.03219
PRP	3539	18.06	0.03219
DY	-	-	12.5476
CD	-	-	5.94274

The numerical results of table 3 show the result of the proposed algorithm with some existing algorithms when applied to test problem 1. The results were ranked based on the number of iterations prior to reaching the optimal values (minimum cost). The numerical results indicated that the proposed algorithm have made significant performance among all algorithms. The performance from implementation of problem 1 indicated that the proposed algorithms (HCG35 with 60 iterations) outperformed gradient descent (GD) method (3533 iteration), Hestenes-Stiefel (HS) method (1909 iterations), Fletcher-Reeves (FR) method (196 iterations), Polak - Ribiere - Polyak (PRP) method (3539 iterations) in term of number of iterations. The HCG35 outperformed all other methods in terms of processing time as indicated in table 4. It is noted that conjugate descent (CD) method and Dai-Yuan (DY) method failed to solve problem 1 at different learning rates of 0.0001, 0.001, 0.1, 0.9, 0.01 and 0.009. The performance results of the both proposed and existing methods are illustrated in figure 1 based on the number of iterations.

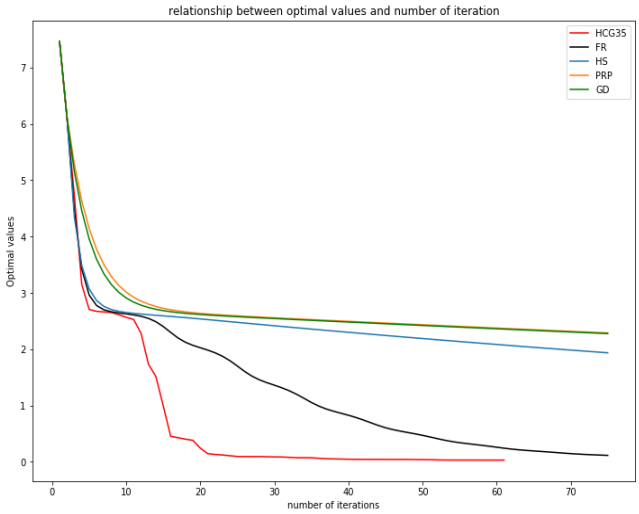

Figure 1: Relationship between optimal values and number of iteration for problem 1.

From the Figure 1, the lower curve is referring to HCG35 method which indicated that the method is highly competitive and is better than the existing methods of FR, HS, GD and PRP methods in solving the problem 1.

4.3.2 Test problem 2

For test problem 2, five experiments were performed with the same initial guess in order to evaluate the efficiency of the methods. The stop criteria used in the experiment for all the algorithms convergence is assumed if $\|g_k\| \leq \varepsilon$ where $\varepsilon = 10^{-16}$ and the symbol "-" was used to represent that the algorithm does not converge. The initial points for the parameters are $\beta_0, \beta_1, \beta_2, \beta_3$ (0, 0, 0, 0) and the learning rate was set to $\propto = 0.01$. The results are depicted in Table 4.

Table 4: Experimental results for problem 2

Methods	**Number of iteration**	**Processing time**	**Optimal values**
HCG35	640	1.15	0.01360303
FR	991	1.77	0.01360303
HS	25297	42.86	0.01360303
GD	49443	78.58	0.01360303
PRP	49509	92.64	0.01360303
CD	-	-	-
DY	-	-	-

Similarly, the numerical results of Table 4 showed the performance of the proposed algorithm with the existing ones when applied to test problem 2. The results were ranked based on number of iterations prior to reaching the optimal values (minimum cost). The numerical results indicated that the proposed HCG35 have made significant performance among all considered algorithms. The performance from implementation of problem 2 indicated that the proposed algorithm HCG35 with 640 iterations outperformed gradient descent (GD) method with 49443 iterations, Hestenes-Stiefel (HS) method with 25297 iterations, Fletcher-Reeves (FR) method with 991 iterations and Polak - Ribiere - Polyak (PRP) method with 49509 iterations. Conjugate descent (CD) method and Dai-Yuan (DY) method failed to solve problem 2 at different learning rates of 0.0001, 0.001, 0.1, 0.9, 0.01 and 0.009. The performance results of the both proposed and existing methods are illustrated in figure 2 based on the number of iterations.

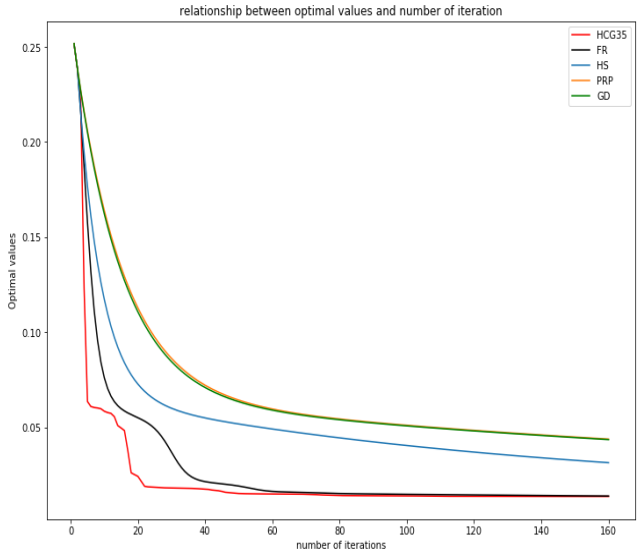

Figure 2: Relationship between optimal values and number of iteration for problem 2

From the Figure 2, the lowest curve is referring to HCG35 method which shows that it is highly competitive and the performances are better than the methods of FR, HS, GD and PRP methods in solving problem 2.

4.3.3 Test problem 3

For test problem 3, five experiments were performed with the same initial guess in order to evaluate the efficiency of the methods. The stop criteria used in the experiment for all the algorithms convergence was assumed if $\|g_k\| \leq \varepsilon$ where $\varepsilon = 10^{-7}$ and the symbol "-" was used to represent that the algorithm does not converge. The initial point for the parameters $\beta_0, \beta_1, \beta_2$ were (0, 0, 0) and the learning rate was $\propto = 0.0001$. The results are depicted in Table 5.

Table 5: Experimental results for problem 3

Methods	Number of iteration	Processing time	**Optimal values**
FR	1304	8.25	7.42249
HCG35	1313	7.56	7.42249
HS	57553	342.02	7.42255
GD	105797	870.51	7.42249
PRP	105806	541.51	7.42262
DY	-	-	-
CD	-	-	-

Table 5 shows the results of the proposed algorithm (HCG35) and existing methods. The numerical results indicated that the HCG35 has made significant performances among all algorithms. The HCG35 with 1313 iterations is highly competitive with the Fletcher-Reeves (FR) method with 1304 iterations but in terms of processing time HCG35 takes 7.56 µs to reach its optimal value while FR takes 8.25 µs. The HCG35 outperformed gradient descent (GD) method,

Hestenes-Stiefel (HS) method and Polak - Ribiere - Polyak (PRP) method while conjugate descent (CD) method and Dai-Yuan (DY) method also failed to solve problem 3 at different learning rates of 0.0001, 0.001, 0.1, 0.9, 0.01 and 0.009. The performance results of the both proposed and existing methods are illustrated in figure 3 based on the number of iterations.

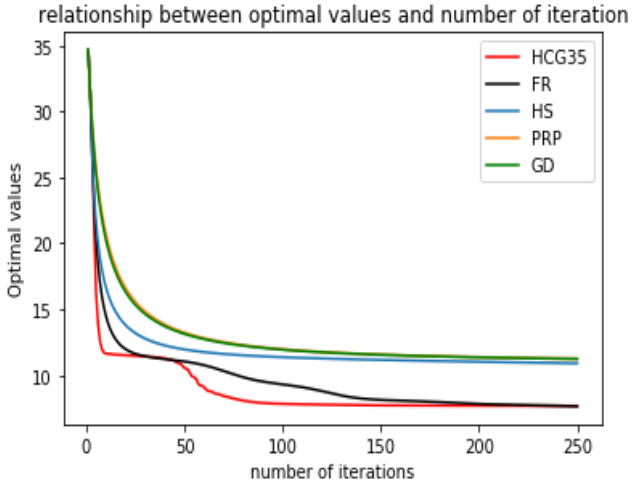

Figure 3: Relationship between optimal values and number of iteration for problem 3

From the Figure 3, the lower curves are HCG35 and FR methods and they are highly competitive. The performance of both HCG35 and FR are better than the methods of HS, GD and PRP in solving problem 3.

5. CONCLUSION

From the three tests performed on all algorithms using regression and the classification problems, it was confirmed that HCG35 CGM performed better in terms of number of iterations and processing time. The HCG35 proved to be a better optimization algorithm when compared to some well-known gradient descent optimizers such as GD, HR, FR, PRP, CD and DY for solving some machine learning problems prior to reaching the optimal values. This shows that the new optimization algorithm provides an improved optimization algorithm to be used in the field of machine learning. The scope of the work had been on a single neural network called perception, in which an improvement could be extended to multilayer neural network. The work also concentrated on one dependent variable in which an extension could be done on more dependent variables.

6. REFERENCES

[1] Aliyu U. M.; Wah J.L.; Ibrahim S, "On application of three-Term Conjugate Gradient Method in Regression Analysis", international Journal of computer Applications volume 2014, 102(8)

[2] Dai Y. H. and Yuan Y., (2001). "An efficient hybrid conjugate gradient method for unconstrained optimization". Annals of Operations Research 103, 3347.

[3] Fletcher R. and Reeves C. M., Function minimization by conjugate gradients, Computer Journal 7, 149 – 154, 1964

[4] Robbins H. and Monro S., "A stochastic approximation method," The Annals of Mathematical Statistics, pp. 400–407, 1951.

[5] Hamoda. M., Rivaie. M., Mamat. M. & Salleh, Z. "A new nonlinear conjugate gradient coefficient for unconstrained optimization", Applied Mathematical Sciences, 9(37), 1813-1822, 2015

[6] Duchi. J, Hazan. E, and Singer. Y, "Adaptive subgradient methods for online learning and stochastic optimization," Journal of Machine Learning Research, vol. 12, pp. 2121–2159, 2011.

[7] Kamilu Uba Kamfa, Mustafa Mamat, Abdelrhaman Abajhar, Mohd Rivaie, Puspaliza Binti Ghazali, Zabidin Salleh "Another Modified Conjugate Gradient Coefficient with Global Convergence Properties", Applied Mathematical Sciences, 2015, vol 9, no 37, 1833-1844.

[8] Liu, Y., & Storey, C. Efficient generalized conjugate gradient algorithms, part 1: theory. Journal of Optimization Theory and Applications, 1991, 69(1), 129–137.

[9] Rivaie, M., Mamat, M., June, L. W., & Mohd, I. A new class of nonlinear conjugate gradient coefficients with global convergence properties. Applied Mathematics and Computation, 2012, 218(22), 11323–11332.

[10] Sulaiman, I. M., "Solving Fuzzy Nonlinear Equations with a New Class of Conjugate Gradient Method". Malaysian Journal of Computing and Applied Mathematics, 2018, 1(1), 11–19.

[11] Yuan Sun; Zhihao Zhang; Zan Yang; Dan L. "Application of Logistic Regression with Fixed Memory Step Gradient Descent Method in Multi-Class Classification Problem." The 2019 6th International Conference on Systems and Informatics (ICSAI 2019).

[12] Yu-Hong Dai. "Nonlinear Conjugate Gradient Methods". Wiley Encyclopedia of Operations Research and Management Science (2010).