Developing a New Optimization Algorithm to Predict the Risk of Car Accidents Due to Drinking Alcoholic Drinks by Using Feed-Forward Artificial Neural Networks

Alaa Saad AHMED[1], Hisham M. KHUDHUR[2]

[1]Computer Science Department, College of Education for Pure Sciences, University of Mosul, Mosul. Iraq
allasaad@uomosul.edu.iq
ORCID: https://orcid.org/0000-0002-4191-1754

[2] Mathematics Department, College of Computer Science and Mathematics, University of Mosul, Mosul, Iraq
hisham892020@uomosul.edu.iq
ORCID: https://orcid.org/0000-0001-7572-9283

Received: 05.08.2021, Accepted: 26.10.2021, Published: 03.01.2022

Abstract — *In this research, we have developed a new algorithm in the field of optimization and its application in teaching artificial neural networks with front feeding to predict the risk of car accidents due to consuming alcoholic beverages, and the algorithm has proven a high efficiency in prediction as it was compared with the results of the model predicting the risk of car accidents due to eating Given alcohol and the results were very close to the true solution to the model.*

Keywords: Artificial Neural Networks, Conjugate Gradient, Machine Learning; Mathematical Modelling,Optimization.
Mathematics Subject Classification: 65K10, 97M10, 90C26.

1 Introduction

In this paper, we try to develop a new algorithm from the traditional (backward) error propagation algorithms by using the pure conjugation condition and applying it in modeling the risk of car accidents due to drinking alcoholic beverages.
It is known that the standard error propagation network represents a successful application of many problems because it is guaranteed to reach the goal if the learning factor is small enough, but it suffers from the problem of slow convergence due to its zigzag path to reach the optimal point resulting from the use of a constant learning factor. During training, in addition to that, there are problems hindering this network, especially in applications of complex problems, and it takes a lot of time to train it, which may take hours as well as thousands of iterations to reach the optimal solution. The performance of this network is

Cite as: *A. S. Ahmed H. M. Khudhur, Developing a new optimization algorithm to predict the risk of car accidents due to drinking alcoholic drinks by using feed-forward artificial neural networks, Journal of Multidisciplinary Modeling and Optimization 4(1) (2021), 11-18.*

affected by many factors, including the choice of the training category, the weights of the primary network, the learning rate, the activation function, the hidden layers and the number of cells in each layer [1] [2].
Our study in this research is focused on a network with a front feeding that consists of the input layer and contains of cells and one hidden layer B of cells and the output layer contains cells as in Figure (1).

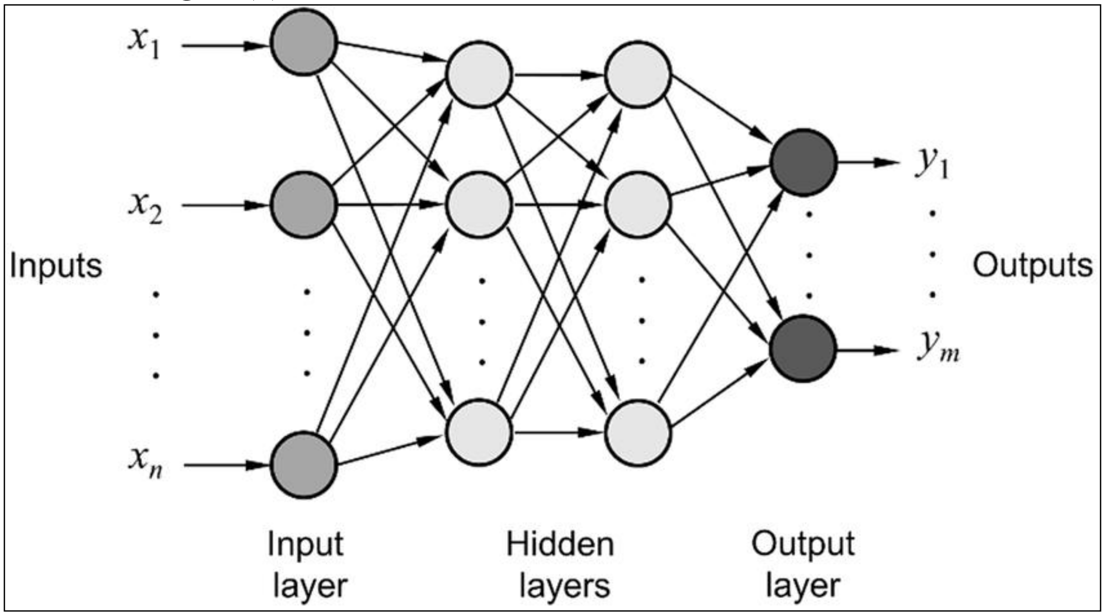

Figure 1. A multi-layered network

Let the training set consist of a pair of target input and output data defined as follows:
$(p_1, t_1), (p_2, t_2), \ldots, (p_m, t_m)$
As it $i = 1, 2 \ldots, m$ and $t_j \in R^{N_2}$ represents the input vector for each, and represents the target output. And that the network learns through the supervisor (Supervised Training) and in a single batch system (Batch SBP) (Laylani, Abbo, and Khudhur 2018), that is, it provides the network with all the training set $(p_1, t_1), (p_2, t_2), \ldots, (p_m, t_m)$ before performing the process of adjusting the weights and after that you adjust the weights as the inclinations calculated in each training example are added to each other to determine the changes in the weights and the wrong function $E(w)$ in this method is as follows :

$$E(w) = \frac{1}{2} \sum_{i-1}^{m} \sum_{j-1}^{m_2} \left(t_{ji} - y_{ji}\right)^2 \tag{1}$$

Since y_{ji} outputs the cell in the output layer depending on the input vector t_{ji}, p_i outputs the desired (target) from the cell j depending on the input vector p_i. We note from equation (1) that teaching the network is achieved if the value of the error function is very small, then it can be said that the issue of teaching a feedforward artificial neural network is to reduce the error function defined in equation (1) according to the one-batch system. (Nguyen and Widrow 1990) The problem can be formulated as follows :

(2)

So N represents the number of weights in the network, and this is an unconstrained optimization problem. Thus, methods and theories of unconstrained examples can be used to solve the above problem.

2 Conjugate Gradient Algorithms

Most of the numerical optimization methods involve two main steps, determining the search direction d_k and calculating the learning factor (step size). In the following, we deal with some methods of numerical optimization. Most optimization algorithms require that the search direction be d_k towards the negative regression of each k, because this characteristic implies that the objective function (error) $E(w)$ will decrease along this direction in general, and the general formula for the search direction in optimization can be written. Not restricted as follows:

$$d_{k+1} = -g_{k+1} + \beta_k d_k \tag{3}$$

If it is $\beta_k = 0$, then the search direction d_k in (3) represents the steepest descent. If the parameter is $\beta \neq 0$, in this case, the method is called conjugate vector (conjugate gradient), and for details see the source [3] [4].

We observed in previous research that the reverse error propagation algorithm uses the derivative of the error function to find the direction of the search,

$$w_{k+1} = w_k + \alpha_k d_k, \quad k \geq 1$$

Where α_k step-size that satisfy the standard wolfe conditions

$$f(w_k + \alpha_k d_k) \leq f(w_k) + \delta \alpha_k g_k^T d_k$$

$$d_k^T g(w_k + \alpha_k d_k) \geq \sigma d_k^T g_k$$

or strong wolfe conditions

$$f(w_k + \alpha_k d_k) \leq f(w_k) + \delta \alpha_k g_k^T d_k$$

$$|d_k^T g(w_k + \alpha_k d_k)| \leq -\sigma d_k^T g_k$$

$$d_{k+1} = \begin{cases} -g_1, & k = 1 \\ -g_{k+1} + \beta_k d_k, & k \geq 1 \end{cases} \tag{4}$$

The Polak and Ribiere (PRP) [5], Hestenes-Stiefel (HS) [6] and β is scalar.

$\beta_k^{PRP} = \frac{y_k^T g_{k+1}}{g_k^T g_k}$ see [5]

$\beta_k^{HS} = \frac{y_k^T g_{k+1}}{y_k^T d_k}$ see [6]

where $g_k = \nabla f(x_k)$, and let $y_k = g_{k+1} - g_k = \nabla f(x_{k+1}) - \nabla f(x_k)$.

Since $g_{k+1} = E(w)$ is clear that d_k must fulfill the regression property, and for more, see [7] [8] [9] [10] [11] [12] [13] and that weights are updated according to the following relationship:

$$w_{k+1} = w_k + \alpha d_k \tag{5}$$

And now we are developing a new algorithm from the optimization algorithms used in learning feedforward artificial neural networks :

(6)

Now we use the pure conjugation condition by multiplying the above equation by y_k we get

$$y_k^T d_{k+1} = -\theta_k y_k^T g_{k+1} + (1-\theta_k)\beta_k^{PRP} y_k^T d_k = 0$$
$$-\theta_k y_k^T g_{k+1} + (1-\theta_k)\beta_k^{PRP} y_k^T d_k = 0$$
$$-\theta_k y_k^T g_{k+1} + (1-\theta_k)\beta_k^{PRP} y_k^T d_k = 0$$
$$-\theta_k (y_k^T g_{k+1} + \beta_k^{PRP} y_k^T d_k) = -\beta_k^{PRP} y_k^T d_k$$
$$\theta_k = \frac{\beta_k^{PRP} y_k^T d_k}{(y_k^T g_{k+1} + \beta_k^{PRP} y_k^T d_k)} = \frac{\beta_k^{PRP}}{\beta_k^{HS} + \beta_k^{PRP}} \tag{7}$$

We substitute the value of θ_k in (6) to obtain a new research direction and apply it in teaching neural networks

The New Algorithm

Step (1): - Determine the values of each of

1- Primary weights $w_1 \in R^N$.

2- Elementary learning rate parameters $\alpha = 0.01$.

Introductions to stopping metrics

i. $k_{\max}$ The upper limit of repetitions, or so-called (epochs).

ii. goal is the allowable error value.

iii. $T_{\max}$ The upper limit of time.

iv. We calculate $E(w_k), g(w_k)$.

Step (2): - If one of the following measures is achieved. Go to step (7)

1- $E(w_k) < goal$.

2- $k = k_{\max}$.

1- $time \geq T_{\max}$

2- $\|g_k\| < \varepsilon_1$

And vice versa we continue.

Step (3): - calculate the direction of research according to the equation $d_k = -g_k$.

Step (4): - calculate the learning factor α_k .

Step (5): - put $w_{k+1} = w_k + \alpha d_k$.

Step (6): - Put $k = k + 1$ then go to step (2).

Step (7): - Pause (Print the results, then stop).

3 Modeling The Risk Of Car Accidents Due To Drinking Alcoholic Beverages [14]

This application deals with modeling the relationship between drinking alcoholic drinks and their application in artificial neural networks.

$$\frac{dR(b)}{d(b)} \propto b \tag{8}$$

By converting this proportion into an equation, the following mathematical model is obtained:

$$\frac{dR(b)}{d(b)} = kb \tag{9}$$

Since k is a constant of proportionality. As is well known the solution of the differential equation

$$R(b) = R(0)e^{kb} \tag{10}$$

So $R(0)=1$ Because when alcohol is not consumed, there is no risk

$$R(b) = e^{kb}$$

To find the value of the constant k, it is found that when the level of alcohol is in the blood $20 = e^{0.14k}$

(11)

If we want to find the percentage of alcohol that leads to a certain risk

(12)

and the results are :

b25=0.1504
b50=0.1828
b75=0.2018
b100=0.2152

Table 1. Percentages of alcohol that lead to certain risk ratios

Alcohol percentage (%)	**0.1504**	**0.1828**	**0.2018**	**0.2152**
Severity (%)	**25**	**50**	**75**	**100**

Therefore, according to this model, when the percentage of alcohol is about 22.0%, the risk ratio is 100% [14].

4 Results of Training Algorithm

In this part of the research we write the results of the learned artificial neural network and a comparison with the results of the exact solution, and as shown in the following table and drawings (4,3 and 5) as drawing (3) illustrates the comparison between the controlled solution and the solution resulting from the artificial neural network education, and the drawing (4) The percentage of risk after teaching the artificial neural networks, and the drawing (5) shows the square error rate resulting from training the artificial neural network, and the problem was programmed using Matlab 2009 Edition.

Table 2. Results of training algorithm and modeling

Severity (%)	The percentage of alcohol in the human body	The percentage of alcohol after training the network for the algorithm
10	0.1076	0.1076
20	0.13999	0.13999
30	0.15893	0.15892
40	0.17238	0.17239
50	0.1828	0.18281
60	0.19132	0.19131
70	0.19853	0.19854
80	0.20477	0.20476
90	0.21027	0.21027
100	0.21519	0.2152

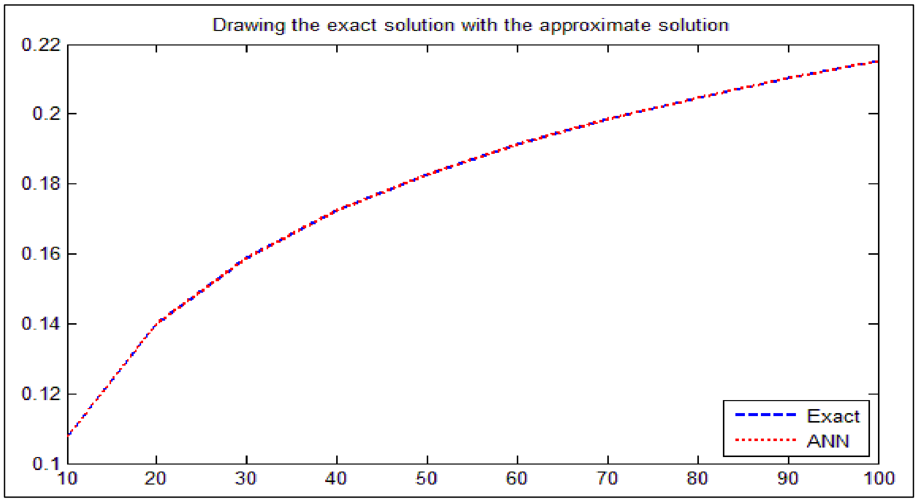

Figure 2. Comparison of Model and Neural Network Approximation

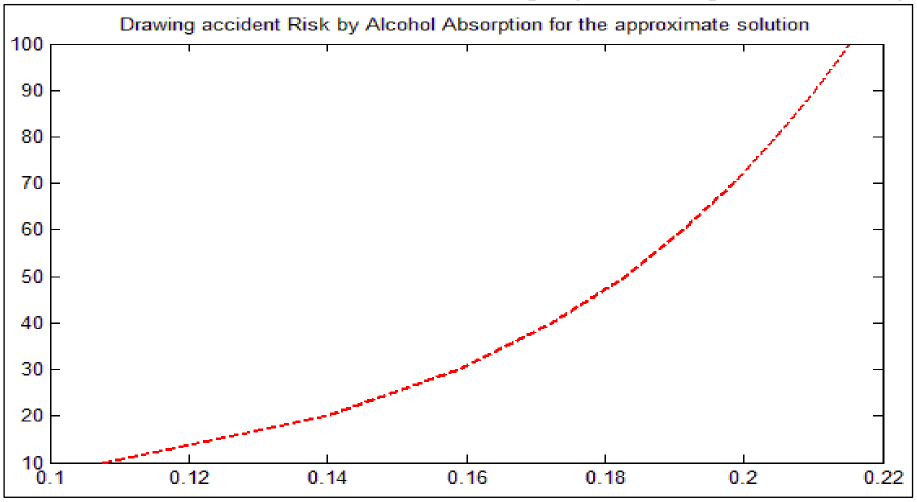

Figure 3. The Risk Ratio after Learning Artificial Neural Networks

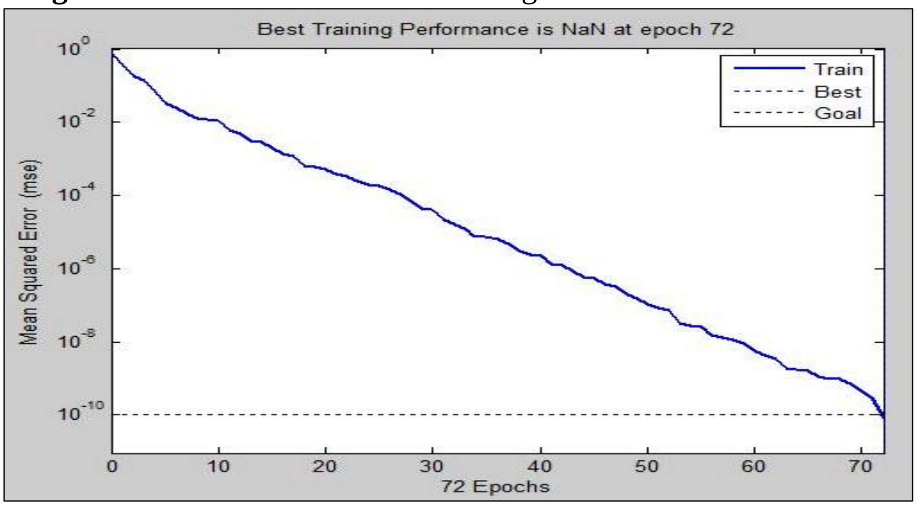

Figure 4. The Square Error Rate for Marking The Network

5 Conclusions

We conclude from the practical results of this research that the artificial neural network with feeding has the ability to predict the risk of car accidents due to alcohol consumption with high accuracy and efficiency, and we also developed the conjugate gradient algorithm to improve the results of the neural network, we can train this neural network to solve other types of problems such as classification problems and regression.

Conflict of Interest Declaration

The authors declare that there is no conflict of interest statement.

Ethics Committee Approval and Informed Consent

The authors declare that there is no ethics committee approval and/or informed consent statement.

References

[1] K. Abbo and M. S Jaborry, Learning rate for the back propagation algorithm based on modified scant equation, Iraqi J. Stat. Sci., 14(26) 2014, 1–11.

[2] Y. A. Laylani, K. K. Abbo, and H. M. Khudhur, Training feed forward neural network with modified Fletcher-Reeves method, Journal of Multidisciplinary Modeling and Optimization, 1(1) 2018, 14–22.

[3] A. Antoniou and W.-S. Lu, Practical Optimization: Algorithms and Engineering Applications. Springer Science & Business Media, 2007.

[4] J. Nocedal and S. Wright, Numerical Optimization. Springer Science & Business Media, 2006.

[5] E. Polak and G. Ribiere, “Note sur la convergence de méthodes de directions conjuguées,” ESAIM Math. Model. Numer. Anal. Mathématique Anal. Numérique, 3(R1) 1969, 35-43.

[6] M. R. Hestenes and E. Stiefel, Methods of conjugate gradients for solving linear systems, J. Res. Nat. Bur. Stand.,49 (1) 1952, 409-436.

[7] R. Fletcher and C. M. Reeves, Function minimization by conjugate gradients, Comput. J., 7(2) 1964, 149-154.

[8]Y. H. Dai and Y. Yuan, A nonlinear conjugate gradient method with a strong global convergence property, SIAM J. Optim., 10(1) 1999, 177-182.

[9] L. C. W. Dixon, Conjugate gradient algorithms: quadratic termination without linear searches, IMA J. Appl. Math., 15(1) 1975, 9-18.

[10] K. K. Abbo and H. M. Khudhur, New A hybrid conjugate gradient Fletcher-Reeves and Polak-Ribiere algorithm for unconstrained optimization, Tikrit J. Pure Sci., 21(1) 2015, 124-129.

[11] H. N. Jabbar, K. K. Abbo, and H. M. Khudhur, “Four--term conjugate gradient (CG) method based on pure conjugacy condition for unconstrained optimization,” Kirkuk Univ. J. Sci. Stud., 13(2) 2018, 101–113.

[12] K. K. Abbo and H. M. Khudhur, New A hybrid Hestenes-Stiefel and Dai-Yuan conjugate gradient algorithms for unconstrained optimization, Tikrit J. Pure Sci., 21(1) 2015, 118–123.

[13] Z. M. Abdullah, M. Hameed, M. K. Hisham, and M. A. Khaleel, Modified new conjugate gradient method for Unconstrained Optimization, Tikrit J. Pure Sci., 24(5) 2019, 86–90.

[14] B. Y. Al-Khayat, Introduction to Mathematical Modeling Using MATLAB, Dar Ibn Al-Atheer for Printing and Publication University of Mosul, Mosul, 2012.