Conjugate Gradient Algorithm as Improvement Neural Network

Nidhal H. al-Assady , Shatha A. M.
Software Engineering dept , College of Computer and Mathematical Sciences, University of Mosul, Mosul , Iraq
(**Received:** 18 / 5 / 2011 ---- **Accepted:** 26 / 10 / 2011)

Abstract
Artificial neural networks are once applications of artificial intelligence; in this paper devised delta learning rule by using the behavior of conjugate gradient algorithm (which consider once of traditional methods for solving nonlinear optimization problems), depending on a appropriate learning ratio(c).Finally, a neural network with high speed and of a supervised type was obtained, in general this mathematical style proposed as a neural network proved to be efficient with regard the results for different letters, comparing with results of standard delta rule.
Key words: Artificial neural network (ANN), conjugate gradient algorithm (CG), delta learning rule (δ-rule), English letters.

1. Introduction:

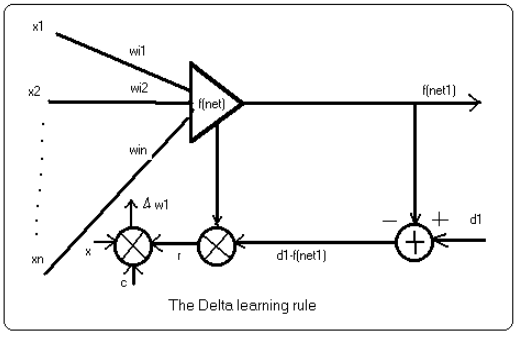
This work includes a modification for once application of Artificial Intelligence, which is artificial neural network (ANN), by using classical technique which is conjugate gradient (CG) for nonlinear optimization problems. The Neural networks are supposed to reproduce the same mechanisms as in the brain. They are able to solve the problematical problems in real time. They are widely used for excellent results for pattern recognition; also it's capable to learn the underlying mechanics of time series, a neural network is made of nodes, called, by analogy with the brain, neurons, connected by weighted edges, called synapses. Typically, the nodes are divided into different layers; there is an input layer, output layer, and hidden layers in between. That forms the topology of the neural network. During a training phase, weights over the synapses are setup in order to pass knowledge through the network [8, 9].Where typically one subgroup makes independent computations Neural network (NN): is a group of processing elements where typically one subgroup makes independent computations and passes the results to a second subgroup [17]. Artificial neural network models have been studied for many years in the hope of achieving human-like performance in the fields of difficult real world problems such as pattern recognition, speech and image recognition, and so on[10]. These models are composed of many interconnected non-linear computational elements that operate in parallel and connected together via weights that are typically adapted during use to improve performance [8].There are a variety of learning rules that establish when and how the connecting weights change. Finally, network exhibit different speeds and efficiency of learning .As a result, they also differ in their capability to accurately respond to the cues presented at the input [19]. Instead of performing a program of instructions sequentially as in VonNeuman computer, neural network models explore many competing, hypotheses simultaneously using massively parallel network composed of many computational elements connected by links with variable weight, the neural network "memory" is judged by interconnection; likewise, while the neural network speed is measured in interconnections per second [8].There are many types of neural networks, but all have three things in common: node (processing unit) characteristics, the connections between them (network topology), and the learning rule. These three aspects together constitute the neural network paradigm [18].

2. Learning Rule

The formalism and its elegant, precise mathematical definition characterize the McCulloch-Pitts model of a neuron. The following paragraphs include the description of that using rule which is applied to recognition. There is some neural network with given architecture. Which find weights for inputs of neurons to train the network for wanted output, the suggestions of approaches are to error reduction learning, was emergence of networks with activation function [15]. Therefore the delta learning rule was once of these methods. The delta learning rule also called continuous perceptron training rule. It's only valid for continuous activation function offer the possibility of more sophisticated learning algorithms, the delta learning rule is adopted the concept of an error surface, this error surface represents cumulative error over a data set as a function of network weights [17 ,19].

2.1 Delta rule as learning rule:

The learning signal is equal: $\gamma = [d_i - f(net)]f'(net)$, Where d_i is the desired output value, $f(net)$ is the actual output value,[19].



The delta learning rule is supervised turning, and its only valid for continuous activation function $\sigma = \psi(net)$, which offer the possibility of more sophisticated learning algorithms. This monotonically increasing and continuous function satisfying [19]:

$net \in R, \psi(net) \in (-1,1), \lambda > 0$.

$$\psi(net) = \frac{2}{1+\exp(-\lambda(net))} \quad \text{.....(1)}$$

Where: $net = w^T x = \sum_{i=1}^{n} w_i x_i$

The Derivative of $\psi(net)$, is denoted by $\psi'(net)$. And it's computed as follows:

$$\sigma = \psi(net) = \frac{2}{1+\exp(-net)} - 1 \Rightarrow \psi'(net) = \frac{1}{2}(1-\sigma^2) \quad \text{.....(2)}$$

For more "See [20]"

2.2 Conjugate Gradient method:

The conjugate gradient method is improved in order to avoid the difficulties of the zigzagging method which is very slow in practice. The first CG method was published by Hestenes-Stiefel in 1952 for solving a system of linear algebraic equations, then Fletcher & Reeves in 1964 were the first which used this technique to minimize a nonlinear function of several variables. The descent type's minimization algorithms construct a sequence of iterations, each of which modifies the independent variable vector from the previous iteration [5, 6, 11] In general the introducing of the classical CG method in optimization technique, which uses the starting point with the negative gradient (-g), to get the minimum point of the unconstrained problem, the method proceeds by generating vector sequence of iterate $\{x_i\}$. The idea of this generating is the concept of conjugate two vectors, in order to determine new directions $(d_i, d_2, ..., d_n)$ of search using information related to the gradient of a quadratic function. This equivalent to find x with $\nabla f(x) = 0$. In such away that successive search directions are conjugate with respect to positive definite Hessian matrix G. At each stage i the direction d_i is obtained by combining linearly the gradient at x_i and ($-g_i$) and the set of directions $\{d_1, d_2, ..., d_{i-1}\}$, where d_i is conjugate to each element of this set [1, 4, 5, 7, 12, 14], therefore the direction d_i is computed as follows:

$$d_{i+1} = -g_{i+1} + \beta_{i+1} d_i \quad \text{.....(5)}$$

Where β_{i+1} is the conjugate coefficient, and it has several values which are introduced from others, for more see[1,5,7,12,16] as follows

$$\beta_{i+1} = g_{i+1}^T g_{i+1} / g_i^T g_i \quad \text{.... [5] (5.a)}$$

$$\beta_{i+1} = g_{i+1}^T g_i / g_{i+1}^T g_{i+1} \quad \text{.... [16] (5.b)}$$

Where $y_i = g_{i+1} - g_i$, if the function is quadratic and the exact line search is used all these formulas are identical, or approximately identical, but when they are applied to arbitrary functions, employing inexact line searches the algorithm already differ [3, 14,16].

Therefore the general conjugate gradient method to find minimum of unconstrained Optimization problems have the following outlines [3]:

Step (1): starting by: x_0, ε, n ,**Step (2): do** {$i = 1, d_i = -g_i$ Step (3): Compute $x_{i+1} = x_i + \lambda_i d_i$, where λ_i is obtained from the line search procedure. Step (4): Find the new direction $d_{i+1} = -g_{i+1} + \beta_{i+1} d_i$; where β_{i+1} is The conjugally coefficient which is computed using last formulas (5.a…..5.f). Step (5): Check for convergence if $\|g_{i+1}\| \le \varepsilon$ then stop.} While (stopping criteria is not satisfying (i = n))

Otherwise set i = i + 1, then go to step (3). **The CG methods have generally properties see [16]**

2.3 The Conjugate Gradient learning rule:

The hinterland was mad between standard delta learning rule and conjugate gradient method, to improve the global rate of convergence of the standard delta rule learning technique and to overcome the difficulties of the standard delta rule learning technique, where this turning through the steepest descent direction, which cause the zigzagging problematic, where it is increase the iterations [1, 2, 3]. The delta learning rule is adopted the concept of an error surface, this error surface represents cumulative error over a data set as a function of network weights. A point on surfaces of this error represents each possible network weight configuration. In order to get learning algorithm, by find the direction on this surface which most rapidly reduces this error.Therfore it is called gradient descent learning. Since the gradient is a measure of slope, as a function of direction, from a point on surface [9, 19], as following figure:

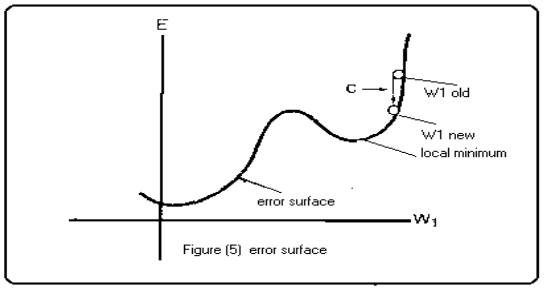


Figure (5) error surface

Where

$$E_i = \frac{1}{2}(d_i - \sigma_i)^2 \quad \text{.....(3.a)}$$

The component of error gradient is:

$$\nabla E = \frac{\partial E}{\partial w_{ij}} = -(d_i - \sigma_i)\psi'(w_i{}^T x)x_j \quad , j = 1,..,n \ldots\ldots\ldots\ldots\ldots\ldots (3.b)$$

The changing in the weight $((w_{i+1} - w_i))$ to be negative gradient direction, it's equivalent to minimization of the error.

$$\Delta w_i = -c\nabla E \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (3.c)$$

By substituting (3.b) in (3.c) to get the delta learning rule, which has the following form:

$$\Delta w_i = c(d_i - \sigma_i)\psi'(w_i{}^T x)x_j \quad , j = 1,..,n \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (4)$$

Where c is constant, which controlling the learning rate. Where the value of it determines how much the weight values move in a single learning episode, if c have large value more quickly the weighs move toward an optimal value, while if the value of c is too large then the algorithm may overshoot the minimum or oscillate. But if c have smaller value, that is less prone to this problem, but do not allow the system to learn as quickly. The optimal value of the learning rate is modified with a momentum factor, which is a parameter adjusted for particular application through experiment [19], also 'c' can be evaluated numerically in each iteration by using the same methods which are using in the line search procedure [20], For the outlines of standard delta learning rule see [19].The following steps describing the Outlines of Conjugate Gradient delta learning rule (CGDL), Which interlard between Conjugate Gradient and delta learning rule, and the block diagram of this method is below:

Step (1): start with i=0, $x = (x_1,\ldots x_n)$, $w = (w_1,\ldots w_n)$, where x_i , w_i are vectors. Where W is randomly weight, X is input.

Step (2): Do {i=1

$$net_i = w_i^T x_i$$

$$\nabla E_i = (d_i - \psi_i(net_i))\psi'(net_i)x_i$$

$$D_i = -1 * \nabla E_i$$

$$w_i^* = w_i + c * D_i$$

Step (3): do {i++, $w_{(i)new} = w_i^*$

$$net_i = w_i^T x_i$$

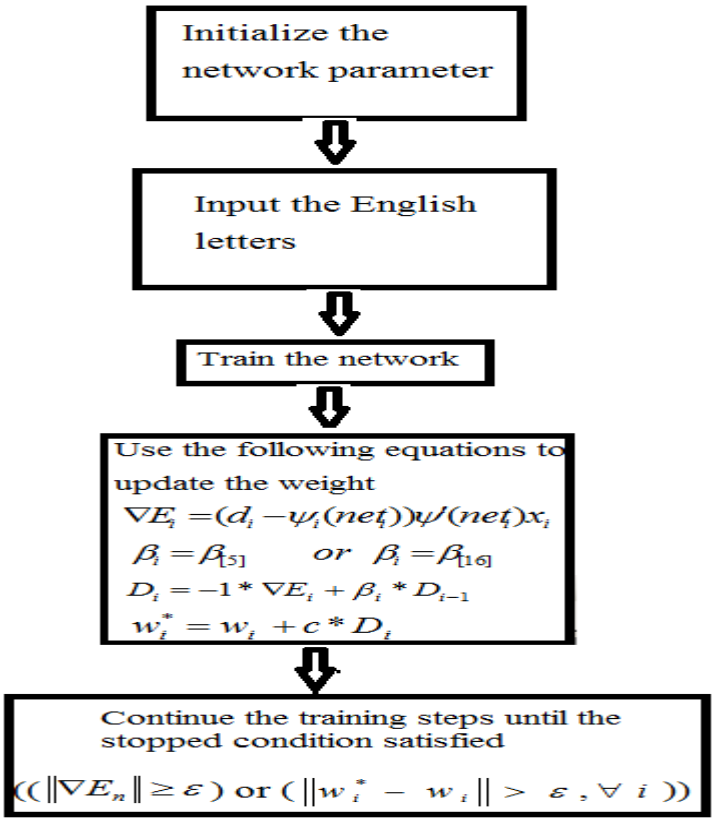
$$\nabla E_i = (d_i - \psi_i(net_i))\psi'(net_i)x_i$$

$$using \quad \beta_i = \beta_{[5]} \quad or \quad \beta_i = \beta_{[16]}$$

$$D_i = -1 * \nabla E_i + \beta_i * D_{i-1}$$

$$w_i^* = w_i + c * D_i$$

} While $(i \neq n)$} While $((\|\nabla E_n\| \geq \varepsilon)$ or $(\|w_i^* - w_i\| > \varepsilon, \forall i))$ The block diagram of (CGDL) method:



3. Numerical Results and Conclusions

The different methods are used in order to get the optimal aim. These methods are related to the classical methods and intelligence methods. The algorithms which are described in this paper, as following:

1. Standard delta learning rule (DLR).
2. CG-delta learning rule (CGDL).

The numerical results are obtained by emblem the variable as following:

X : denoted to the input English letters, $\chi = (\chi_1, ..., \chi_n)$, where $\chi_i, \forall i$ are vector $(\dim(\chi_i) = 1x9)$, or matrix($\dim(\chi_i) = 7x9$).

n: denoted to the number of English litters (n=2,6,10,….,26). So that if $n = 2 \rightarrow X = (\chi_1, \chi_2)$

$$n = 2 \rightarrow X = (\chi_1, \chi_2) \Rightarrow \chi_1 = \begin{vmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 \end{vmatrix}$$

also,

$$\Rightarrow \chi_2 = \begin{bmatrix} 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 \end{bmatrix}$$

The initial weight, $w = (w_1, ...w_n)$, where $w_i, \forall i$ are randomly scale weights (vector, or matrix).The complete results are given in table (1) and table (2). These computations are getting from learning the algorithms on the (26) input English liters in different cases of inputs, which are illustrated in the appendix. These algorithms are coded using C++ language with numerical results using the personal Pentium IV computer. The comparison of the methods occurred depending on using the number of iterations of learning for the suggestion neural network.

Table (1): Comparisons between Standard delta rule (DLR) and Conjugate Gradient Delta Rule (CGDL)

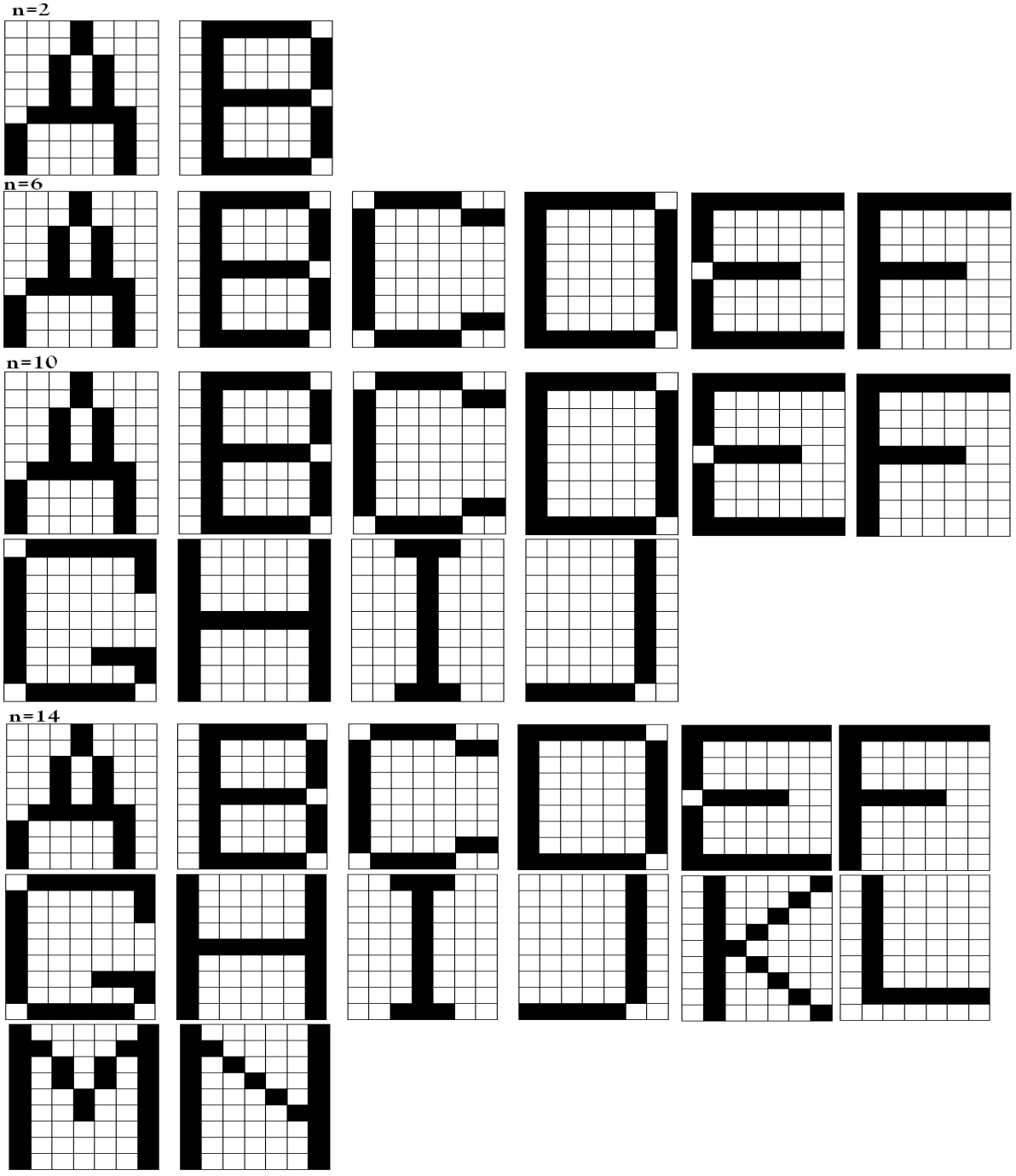
Inputs n	DLR		CGDL			
			$\beta_{[5]}$ fr		$\beta_{[16]}$	
	vectors	matrix	vectors	matrix	vectors	matrix
2	356	556	340	1064	338	1056
6	440	1668	744	1104	740	1080
10	530	2780	760	1110	756	1086
14	620	3892	776	1136	772	1120
18	703	5004	794	1208	792	1188
22	795	6116	810	1210	808	1200
26	887	7228	826	1244	824	1212
Total	4331	27244	5050	8076	5030	7942

Table (2): Improvement ratio Standard delta rule (DLR) and Conjugate Gradient Delta Rule (CGDL)

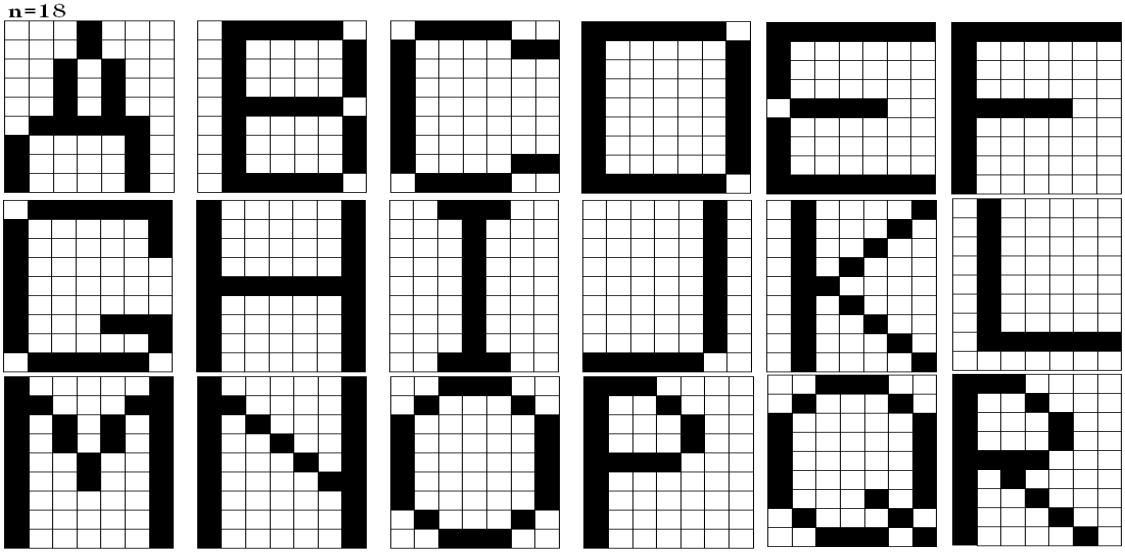
	DLR	CGDL ($B_{[5]}$)	CGDL ($B_{[16]}$)
Matrix	100%	29.64%	29.151%

From table (1) we note that the performance of the Conjugate Gradient Delta Rule (CGDL) are best compared with the Standard delta rule(DLR), when using the weights as matrix, but table(2) describe that (DLR) as 100% with (NOI), the CGDL(B[5]) requires 29.64% (NOI), while CGDL(B[16]) requires 29.151% (NOI), where these ratio are better than the ratio of (DLR).At the end, we note that the CG-delta learning rule (CGDL) is suitable When modified this method(CGDL), the result which obtained from this method more better than method(DLR), dependent on iteration number

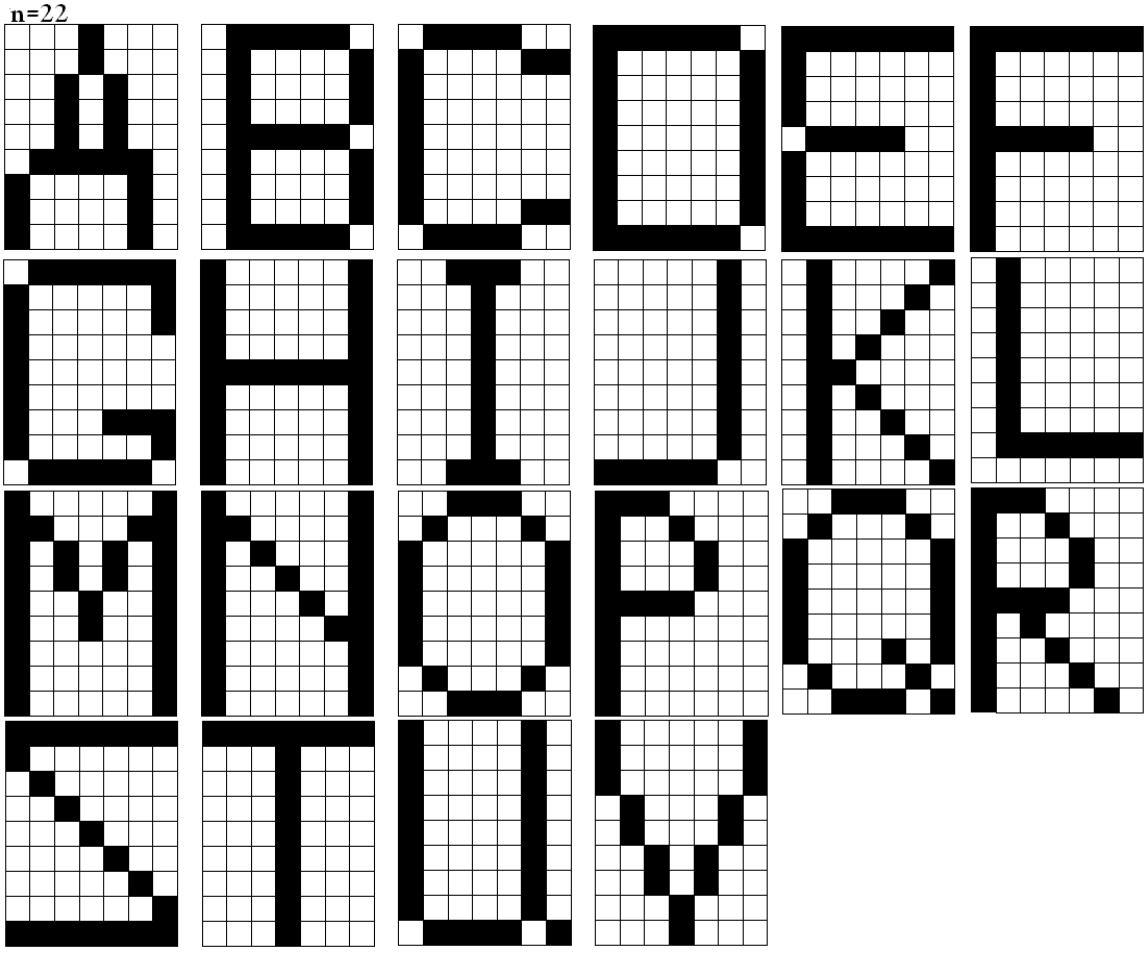
Appendix

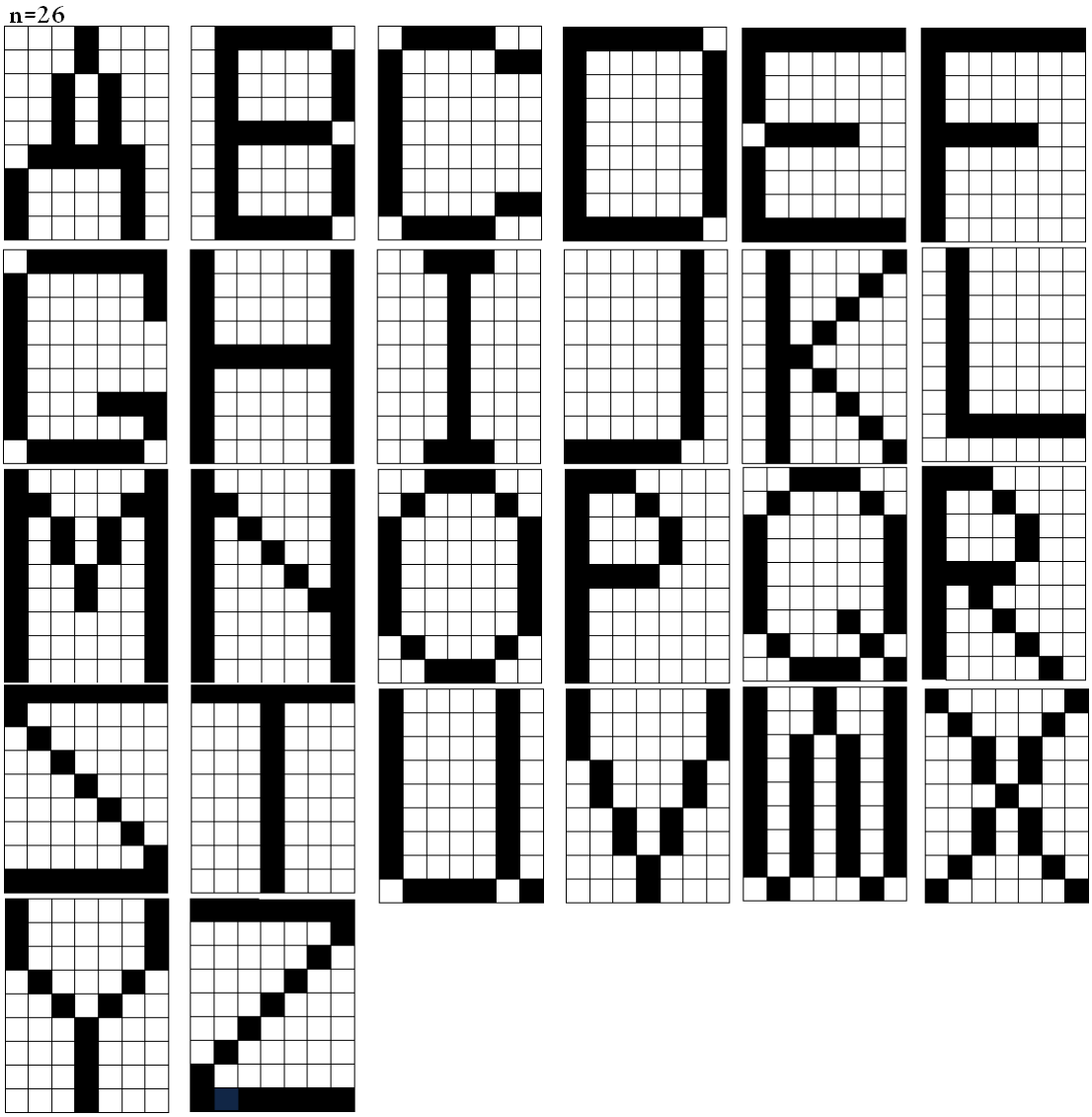


n=18



n=22





References

[1] Al-Assady, N.H. and Al-Bayati, A.Y.,(1986) "Conjugate Gradient Methods", Technical Research Report, No.(1),School of Computer Studies , Leeds Universities ,U.K. Fpi.

[2] Al-Mashhadany, H.K.M., (1996) "Non-Quadratic Models for Unconstrained Optimization Technique", MS.C, Thesis, University of Mosul, September.

[3] Beale, E.M.L., (1988) "Introduction to Optimization", Wiley inter Science Series in Discrete Mathematics and Optimization.

[4] Dixon, L.C.W., Spedicato, E. and Szgoo, G.P., (1975) "Non Linear Optimization Theory and Algorithms", Birkhauser, Boston L-.S.A.

[5] Fletcher, R. and Reeves, C.M.,(1964) "Function Minimization by Conjugate Gradient", Computer Journal Vol. 7,pp.149-154.

[6] Ghorbani, A. Ali and Leila Bayar, (1998) "A Correlated Back propagation Learning Dynamic Parallel Tangent Optimization Algorithm", IASTED, Irbid, Jordan.

[7] Hestenes, M.R. and Stiefle, E., (1952) "Methods of Conjugate Gradient for Solving Linear Systems", Journal of Research of the National Bureau of Standards 49, pp. 409-436.

[8] Lippman R.P., (1987) "An Introduction to Computing with Neural Nets" IEEE ASSP Magazine, Vol.4, No.2, PP.1-22.

[9] Luger G.F., Stubblefied, W.A. and Stubblefield, W.A.,(1998) "Artificial Intelligence Structure and Strategies for Complex Problems" .Third Edition Addison Wesley Longman.

[10]Mustafa, K. M. S., (2002) "Multiple Classification System for Remotely Sensed Data ", Ms. Thesis University of Mosul.

[11]Nazareth, L. (1986) "Conjugate Gradient methods less dependent on conjugacy", SIAM review 28, pp.501-512.

[12]Polak E. and Ribiere, G., (1969) "Note for Convergence Des Methods Direction Conjugate", Rev. Fr. Inf. Rec. Op. 16-R.

[13] Rao, M. and Chandra, K., (1983)" A New Line Search for Optimization Algorithms", Proc. Conf. On systems, M, AM.
[14] Rao S.S, (1984) "Optimization Theory and Applications", Wiley Eastern Limited second edition.
[15] Rao, V.B. and Rao, H.V.,(1996) " C++ Neural Networks and Fuzzy Logic", 2nd Edition, New York.
[16] Shatha, A. M.,(2004) "An Investigation for Intelligence and Classical Optimization for Non Linear Problems" , Ph.D. Thesis University of Mosul.
[17] Valluru, B. Rao and Hayagriva V. Rao (1993) "C++ Neural Networks and Fuzzy Logic", Henry Holland Company Inc., New York.
[18] Wu Jian-Kang, (1997) "Neural Networks and Simulation Methods", Marcel Decker Inc., New York.
[19] Zurada, J.M., (1994) "Introduction to Artificial Neural Systems", JAICO Publishing House, Mumbai.
[20] Shatha, A. M.,(2007) " Hybrid Conjugate Gradient as Neural Network using New Line Search Technique" , College of Basic Education Researches Journal, Vol.5, No.3, pp. 214-232.

خوارزمية التدرج المترافق كشبكة عصبية محسنة

نضال حسين الاسدي، شذى عبد الله محمد

قسم البرمجيات ، كليه علوم الحاسوب والرياضيات ، جامعة الموصل ، الموصل ، العراق

(تاريخ الاستلام: 18 / 5 / 2011 ---- **تاريخ القبول:** 26 / 10 / 2011)

الملخص:

تُعد الشبكات العصبية الاصطناعية احد تطبيقات الذكاء الاصطناعي، حيث تم تحسين (Delta learning rule) كشبكة عصبية باستخدام أسلوب خوارزمية التدرج المترافق (والتي تعد واحدة من الطرق التقليدية لحل مسائل الامثلية اللاخطية) وبالاعتماد على نسبه تعلم ملائمة للشبكة المستحدثة. وأخيراً تم الحصول على شبكه عصبيه ذات سرعة عالية ومن نوع Supervised. والنتائج الحسابية بشكل عام تبين كفاءة الشبكة المقترحة عند تطبيقها لعدة إدخالات من الحروف، بعد مقارنتها مع نتائج الشبكة الأصلية.

الكلمات ألمفتاحيه: الشبكات العصبية الاصطناعية، خوارزمية التدرج المترافق، قاعدة دلتا للتعلم، الأحرف الانكليزية.