

# Back Propagation Algorithm: An Artificial Neural Network Approach

Parth Bhasin

Department of Computer Science  
HMR Institute of Technology and Management  
New Delhi, India

Vaishali

Department of Computer Science  
HMR Institute of Technology and Management  
New Delhi, India

**Abstract**—Back Propagation Algorithm is currently a very active research area in machine learning and Artificial Neural Network (ANN) society. It has gained huge successes in a broad area of applications such as image compression, pattern recognition, time series prediction, sequence detection, data filtering and other intelligent tasks as performed by the human brain. In this paper, we provide a brief overview of ANN and BP algorithm, how they work and highlight some of the current research efforts and the challenges with them.

**Keywords**— Artificial Neural Network; Back Propagation Algorithm; Applications; Image Compression; Challenges

## I. INTRODUCTION

A lot of research has taken place from over the last few decades in developing the applications of Artificial Neural Networks (ANNs). Idea of ANNs are obtained from the learning processes of human brain by putting them into logical methods. Artificial Neural Networks (ANNs) works with the help of small units called Artificial Neurons, which are similar to biological neurons in the brain, which are used for processing information; these Artificial Neurons can be trained for complex tasks as well. Similar to how we learn to read, understand speech, write, and recognize pattern – with the help of examples, ANNs are trained and not programmed. Based on huge volumes of data ANN has also solved many complex real world problem such as foreseeing future trends. ANN have been successfully performed in all engineering fields such as , decision and control, biological modelling, engineering and manufacturing, health and medicine, marketing and so on [1-5].

## II. RELATED WORK

Back propagation algorithm uses Incline descent-learning rule, which needs careful selection of parameters such as learning rate value, initial weights, and activation function should be selected carefully. Slow network intersection, network error or failure may be caused due to wrong choice of these parameters. Due to these problems, many variations in incline descent BPNN algorithm have been suggested by former researchers to advance the training efficiency. Some suggestions are to use learning rate and momentum to speed-up the system convergence and evade getting trapped at local minima. For control of weight adjustments, these two parameters are often used [6]. Accelerated results are obtained by Back-propagation with Fixed Momentum (BPFM) when the weights and current downhill of the error

function are in related directions, when the present incline is in a differing direction to the previous update. Because of BPFM the weight course will go in the upward direction instead of dropping down as desired, due to which, there will be an essential requirement of adjusting momentum coefficient adaptively instead of keeping it fixed [7], [8]. There has been several adaptive-momentum modifications by researchers over the past few years. One such change is Simple Adaptive Momentum (SAM) [9], planned to further advance the conjunction capability of BPNN. SAM works by scaling the momentum-coefficient, based on the similarities between the distinctions in the weights at the previous and current repetitions. SAM is initiated to lower computational outlays than the Conventional BPNN. In 2009, R. J. Mitchell attuned momentum-coefficient in an altered way than SAM [9]; the momentum-coefficient was attuned by seeing all the weights in the Multi-layer Perceptron (MLP). This method was found much better than the formerly proposed SAM [10]. In 2011, M. Z. Rehman , N. M. Nawi worked towards altering the momentum for all nodes in the neural network. The GDAM has performed well for all ordering problems than the earlier methods [11]. In 2005, Ankit Gupta, Rahul Ramchandani, Aniruddha Gupta, Anil K Ahlawat and Gaurav Malik proposed a substitute to back propagation algorithm, momentum, incline and speed factor following has been added to the learning rate, algorithm and momentum enthusiastically adjusted for growth of overall efficiency of the algorithm and for growth of the speed of convergence of the system [12]. Has used prime method for weight initialization, ensures that the outputs neurons are in the active region and the range of initiation function is fully utilized. Additionally all of the changes has been implemented on 4-bit parity problem, 2-bit parity checker and encoder problem and produced bigger results [13]. In 2012, S.K.Pathan, S.P.Kosbatwar for Pattern Association, Back Propagation algorithm was used and it formed very well results in character acknowledgment too. (IJCSSES) Vol.3, No.1, February 2012.

## III. ARTIFICIAL NEURAL NETWORKS

Artificial Neural Networks are fairly crude electronic models based on the neural structure of the brain. The brain on the whole learns from good amount of experience. There are some problems that are beyond the scope of current computers but are indeed decoded by small energy efficient packages. This brain modeling also enables a less technical way to develop machine solutions. This recent approach of computing also provides a more graceful degradation during system overload than its more traditional alternatives.

Definitely, the biologically inspired methods of computing are the next major advancement in the computing industry. For solving artificial intelligence problems, ANN is configured that too without creating a model of real biological system. ANN is used for speech recognition, adaptive control and image analysis. Through a learning process, these applications are done. For example, ANN is required in learning biological system, which involves the adjustment between neurons through synaptic connection.

**A. Architecture**

Architecture is made up from an input, output and one or more hidden layers. Every node from input layer is connected to a node from hidden layer and vice versa. Some weight is associated with each and every connection. Input layer represents the raw information that is provided to the network. This part of network is never changing its values. All the inputs fed to the network are duplicated and send down to the nodes in hidden layer. Hidden Layer receives the data from the input layer. With the help of weight values, it modifies input values

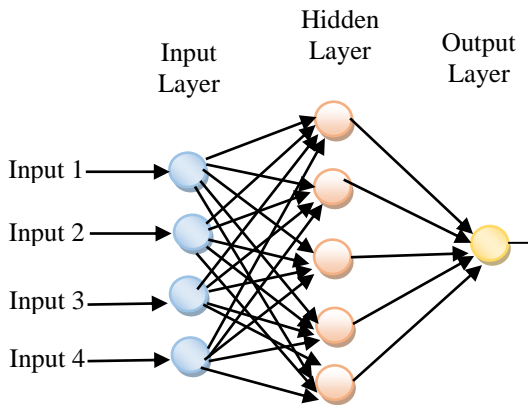


Figure 1: Simple Neural Network

, and then the new value is send to the output layer but it will also be modified by some weight from connection between hidden and output layer. Output layer route information received from the hidden layer and produces an output. By activation function, this output is then processed.

**B. Working of ANN**

There are myriad ways in which these individual neurons can be clustered together. The working of ANN depends on the cluster formation. In human body, clustering occurs in such a way that information can be processed in a dynamic, interactive, and self-organizing way. Biologically, neural networks are constructed from microscopic component to a three-dimensional world. These neurons appear to be capable of nearly having unhindered interconnections. That is not true of in the case of any proposed, or existing, man-made network. Current technology is used in integrated circuits that are two-dimensional devices with a restricted number of layers for interconnection. This reality restrains the types and scope of artificial neural networks, that can be implemented in silicon. At present, neural networks are the simple clustering of the primitive artificial neurons. The simple clustering occurs by creating layers which are then connected to one another. The way these layers connect, is the other part of the "art" of engineering.

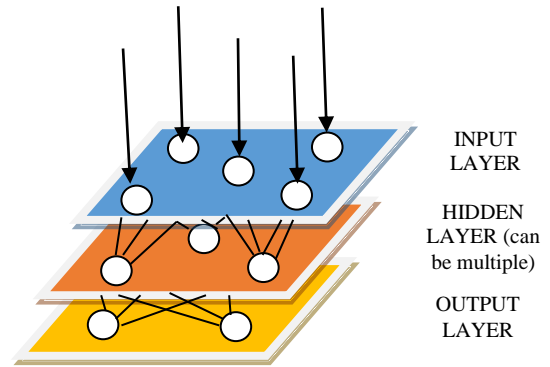


Figure 2:- A Simple Neural Network Diagram.

All artificial neural networks have a similar topology as shown in Figure1. In that structure, some of the neurons interface to the real world to receive its inputs. The rest neurons provide the real world with the network's outputs.

Another type of connection is feedback. Feedback is where the output of one layer routes back to a preceding layer. An example of this is shown in Figure 3.

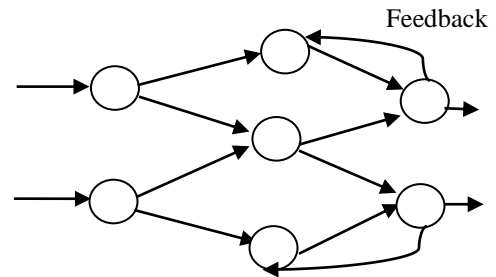


Figure 3:- Simple Network

Through regular feedback and competition, the way that the neurons are associated to each other has a significant impact on the operation of the network. Now days, with more professional software development packages, the users are allowed to add, delete, and control all these connections by his own will. By "tweaking" parameters these connections can be made to either excite or hamper.

**C. Advantages and Limitations**

The neural networks have a lot of advantages. Some of the most important advantages of the neural networks are discussed here [14]:

1. Real Time Operation: In neural network computation can be carried out in parallel.
2. Self- Organization : A neural network can create its own representation of the information it receives during learning.
3. Adaptive learning: A neural network has the ability to learn how to do things and adapt to the surroundings.
4. Pattern recognition: It is a powerful technique for the data security. Neural networks learn to recognize the patterns which exist in the data set.
5. Neural networks can be build as informative models whenever the conventional approaches fail because neural can handle very complex interactions they can easily model

data which was not done by traditional approaches like inferential statistics or programming logic.

6. Neural networks are flexible in a changing environment. Although neural networks may take some time to learn a sudden drastic change but they are excellent in adapting the constantly change in information.

7. The system is developed by learning rather than programming. Neural methods teach themselves the patterns in data freeing and analyse interesting networks.

8. Performance of neural networks is more efficient than traditional methods.

But in this world, everything has its pros and cons, so the neural network system also has some cons. The limitations of ANN [15] are:

1. ANN or Neural Networks is not a daily life problem solver.
2. Nature of ANN is like a Black Box.
3. There is no single standardized paradigm for Neural Networks development.
4. The Output Quality of an ANN can be unpredictable with some problems.
5. Many ANN Systems does not describe how they solve the problems.
6. There is no structured way is available

#### D. Applications

ANN is a powerful tool used for various real life applications like:

1. Data filtering , data processing, clustering and compression.
2. Time series prediction and modeling.
3. In many systems for Pattern recognition, pattern detection and sequential decision-making.
4. Call control- answer an incoming call (speaker-ON) with a swipe of the hand while driving.
5. Application areas of ANNs include system identification and control, game playing and decision making (chess, racing), pattern recognition (radar systems, face identification, object recognition, etc.), sequence recognition (gesture, speech, handwritten text recognition), medical diagnosis, financial applications, data mining (or knowledge discovery in databases, "KDD").
6. Skip tracks or control volume on your media player using simple hand motions.

#### IV. BACK PROPOGATION ALGORITHM

One of the most popular NN algorithms is back propagation algorithm. In 2005, Rojas claimed that Black Propagation Algorithm could be broken down to four main steps. After assigning the weights of the network randomly, the back propagation algorithm is used to compute the necessary corrections. The four main steps he talked about were:

- i) Feed-forward computation
- ii) Back propagation to the output layer
- iii) Back propagation to the hidden layer
- iv) Weight updates

When the value of error function seems to be significantly small, the algorithm stops. There were some variation proposed by other scientist but Rojas definition seemed to be quite accurate and easy to follow. Weight updates, which is the last step is happening throughout the algorithm

#### A. Training in Back Propagation

The backpropagation algorithm is stated as follows:

1) The backpropagation algorithm operates in two phases: Initially, the training phase in which the training data samples are provided at the input layer in order to train the network with predefined set of data classes. Eventually, during the testing phase, the input layer is provided with the random test data for predicting the applied patterns.

2) Since this algorithm is based on the supervised learning approach, therefore the desired result is already known to the network. In case of discrepancy between the computed result and the desired result, the difference between the two is backpropagated to the input layer so that the connection weight functions of the perceptrons are adjusted in order to bring the error upto the error tolerance factor range.

3) This algorithm operates in either of the 2 modes: Incremental mode in which each propagation is followed immediately by the weight adjustment or batch mode in which the weight updations take place after various consecutive propagations. Usually batch mode is preferred over incremental mode due to less time consumption and less no. of propagative iterations. In this algorithm, a pattern is presented at the input layer. The neurons at this layer pass the pattern activations to the next layer neurons, which is actually the hidden layer. The outputs at the hidden layer neurons are generated using a threshold function along with the activations determined by the weights and the inputs. The threshold (saturation)function is computed as:  $1 / (1 + \exp(-x))$  where x is the activation function value which is computed by multiplying the weight vector with the input pattern vector.

4) The hidden layer outputs become input to the output layer neurons, which are again processed using the same saturation function.

5) The final output of the network is eventually computed by the activations from the output layer.

6) The computed pattern and the input pattern are compared and in case of discrepancy, an error function for each component of the pattern is determined, and based on it the adjustments to weights of connections between the hidden layer and the output layer are computed. A similar computation, still based on the error in the output, is made for the connection weights between the input and hidden layers. The procedure is repeated until the error function reaches the range of the error tolerance factor set by the user.

7) The advantage of using this algorithm is that it is simple to use and well suited to provide a solution to all the complex patterns. Moreover, the implementation of this

algorithm is faster and efficient depending upon the amount of input-output data available in the layers.

### B. Problems with Back Propagation Algorithm

Some problems are associated with BP Algorithm which includes paralysis in network, local minima and slow convergence.

- The best known is called “Local Minima”. This occurs because the algorithm changes the weights most of the time in such a way as to cause the error to fall. But the error might briefly have to rise as part of a more general fall, If this is the case, the algorithm will “get stuck”.

- When the weights are adjusted the network paralysis occurs to very large values during training, large weights can force most of the units to operate at extreme values, in a region where the derivative of the activation function is very small.

- Many repeated presentations of the input patterns are required, for which the weights need to be adjusted in multilayer neural network before the network is able to settle down into an optimal solution

### C. Solution to the problems

To improve the performance range from finding the optimal learning rate to finding the optimal network architecture, approaches are needed. Some of the most convincing approaches are:

- Adaptive learning rate and momentum factor: Despite there is fixed learning rate in training, the learning rate and momentum factor can be adjusted dynamically during training [Weir 1990, Fausett 1994]. Using adaptive learning rate and momentum, decreased training time and improved convergence was achieved. A careful selection of the learning rate is to ensure smooth convergence. A large learning rate can cause network paralysis and a small learning rate causes slow convergence. A momentum factor is used to smooth error oscillation. The learning rate and momentum should be varied according to the region where the weight adjustment is.

- Random weight initialization: The choice of initial weight influences whether the network converges quickly or not [Fausett 1994]. The weight update between two units depends on both the derivative of the objective (error) function with respect to weights, as well as the activation value of units. weights are usually initialized randomly to small values [Rumelhart et al 1986]

- Optimal network architecture selection: Through careful selection of the network size, the good performance in trained network is achieved. An oversized network can lead to over fitting of the data but the simple small sized network can lead to under fitting. Optimal architecture selection is split into three areas:

- 1) Growing the network during training by adding more parameters to the network, [Hirose et al 1991, Jutten et al 1995]
- 2) Pruning the network by removing redundant parameters during training [Sietsma et al 1988, Engelbrecht et al 1996, Le cun 1990]

- 3) Regularization through penalty terms added to the objective function [Weigend et al 1991, Kamimura et al 1994, Karayiannis et al 1993]

- Training with jitter: Jitter (artificial noise) is deliberately added to inputs during training. Training with jitter is a form of regularization, such as weight decay. An advantage of jitter is that the NN can be brought out of a local minimum [Beale et al 1990] Putting artificial noise in small training sets is very effective in improving generalization performance when small training sets are used.

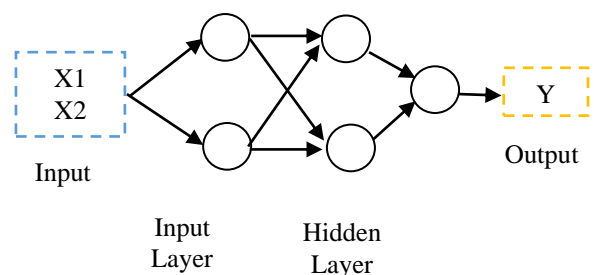
- Adaptive learning function: For faster convergence and more accurate rate activation functions can be adapted and trained just like the weights of NN. [Zurada 1992a, Engelbrecht et al 1995, Fletcher et al 1994]. Zurada [Zurada 1992a] and Fletcher et al]

- Making optimal use of the training data helps in active learning. Much research has been done in developing active learning models [Engelbrecht et al 1998, Engelbrecht et al 1999a, Engelbrecht et al 1999 a] Active learning refers to the selection of a subset of the available training data dynamically during training, where the subset contains the most informative data. It has been found to save computational cost and reduce training time.

## V. TRAINING NEURAL NETWORK WITH BACK PROPOGATION ALGORITHM

### A. Implementation on Basic Digital Logic Gates

VLSI designs are used to form basic logic gates. ANN architecture is trained on silicon chip using back propagation algorithm to implement basic logic gates. Image compression and decompression is used in order to train on chip for a complex problem. Image compression is selected on the basis that ANN, are more efficient than traditional compression techniques like JPEG at higher compression ratio. The Figure 2 shows the architecture of 2:2:2:1 neural network selected to implement basic digital gates i.e AND, OR, NAND, NOR, XOR, XNOR function. The network has two inputs  $x_1$  and  $x_2$  with output  $y$ . In order to select the transfer function to train the network, MATLAB analysis is done on the neural network architecture and it is found that TANSIG (hyperbolic tangent sigmoid) transfer function gives the best convergence of error during the training to implement digital gates.



### B. Implementation on Image Compression

Artificial Neural Network based techniques provide means for the compression of data at the transmitting side and

decompression at the receiving side [16]. Not only can ANN based techniques provide sufficient compression rates of the data in question, but security is easily maintained [17]. This occurs because the compressed data that is sent along a communication channel is encoded and does not resemble original form [18]. ANN has been applied to many problems, and have demonstrated their advantage over classical methods when dealing with noisy and incomplete data. Data compression is one such application which helps in solving many problems [16], [19]. Neural networks are suited to this particular function, as they have an ability to preprocess input patterns to produce simpler patterns with fewer components. This compressed information (stored in a hidden layer) preserves the full information obtained from the external environment. The compressed features may then exit the network into the external environment in their original uncompressed form. The network structure for image compression and decompression is illustrated in Figure 4 [18]. This structure is referred to feed forward type network. The input layer and output layer are fully connected to the hidden layer. Compression is achieved by adjusting the value of  $K$ , the number of neurons at the hidden layer less than that of neurons at both input and the output layers. The data or image to be compressed passes through the input layer of the network, then subsequently through a very small number of hidden neurons. It is in the hidden layer that the compressed features of the image are stored, therefore the smaller the number of hidden neurons, the higher the compression ratio. The output layer consequently outputs the decompressed image. It is expected that the input and output data are the same or very close. If the network is trained with appropriate performance goal, then the input image and output image are almost same. A neural network can be designed and trained for image compression and decompression. Before the image can be used as data for training the network or as input for compression, certain pre-processing should be done on the input image data. The pre-processing includes normalization, segmentation and rasterization [16] [20]. The pre-processed image data can be either used for training the network or as input data for compression. The input image is normalized with the maximum pixel value. The normalized pixel values for grey scale images with grey levels [0,255] in the range [0, 1]. The reason of using normalized pixel values is due to the fact that neural networks can operate more efficiently when both their inputs and outputs are limited to a range of [0,1].

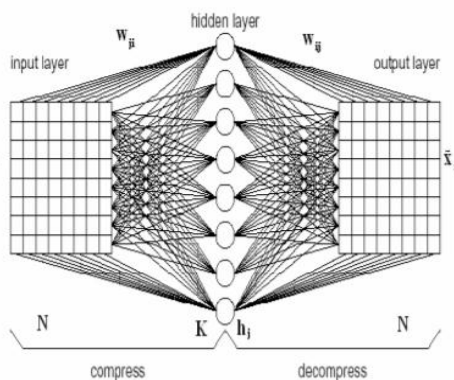


Figure 4 The N-K-N Neural Network

This also avoids use of larger numbers in the working of this network, which reduces the hardware requirement drastically when implemented on FPGA.

If the image to be compressed is very large, this may sometimes cause difficulty in training, as the input to the network becomes very large. Therefore in the case of large images, they may be broken down into smaller, sub-images. Each sub-image may then be used to train an ANN or for compression. The input image is split up into blocks or vectors of  $8 \times 8$ ,  $4 \times 4$  or  $2 \times 2$  pixels by segmentation. This process of splitting up the image into smaller sub-images is called segmentation [18]. Usually input image is split up into a number of blocks, each block has  $N$  pixels, which is equal to the number of input neurons. It is hence selected to train for image compression and decompression. This network gives 50% image compression, because there are two neurons at the hidden layer compared to four input/neurons at the input layer. The network is to be trained online on FPGA using back propagation training algorithm with appropriate performance goal so the input image and output image are almost close. The training is done for gray scale images of pixel size  $256 \times 256$ . The image is normalized first, then the large image is split up into sub-images of size  $2 \times 2$  pixel i.e., segmentation is done to convert  $2 \times 2$  image pixels into a vector of 4 pixels. This vector forms the four inputs to the network. The same pre-processing is done on the training image and the vectors are stored on the FPGA, which are used while training the network

## VI. CONCLUSION

Neural Network is interconnected network that resembles human brain network. The most important characteristic of Neural Network is its ability to learn. Neural Network model could be created to help with classifying new data, when presented with training set (form of supervised learning where input and output values are known. Outputs that are achieved by using Neural Network are encouraging, especially in some fields like image compression. In last two decades, neural networks have gained more and more attention. Back Propagation algorithm is most popular algorithm used in implementation of Neural Network. It is one of the main reasons why Neural Network are becoming so popular.

## REFERENCES

- [1] Kosko, B.: Neural Network and Fuzzy Systems. 1sted., Prentice Hall of India (1994)
- [2] Krasnopolsky, V. M. and Chevallier, F. Some Neural Network application in environmental sciences. Part II: Advancing Computational Efficiency of environmental numerical models. In: Neural Networks. (eds.), vol. 16(3-4), pp. 335--348. (2003)
- [3] Coppin, B.: Artificial Intelligence Illuminated," Jones and Bartlet Illuminated Series, USA, Chapter 11, pp. 291-- 324. (2004)
- [4] Basheer, I. A., Hajmeer, M.: Artificial neural networks: fundamentals, computing, design, and application." J. of Microbiological Methods, 43(1), 03--31 (2000)
- [5] Zheng, He., Meng, Wu., Gong, B.: Neural Network and its Application on Machine fault Diagnosis," In: ICSYSE 1992, 17-19 September, pp. 576--579, (1992)
- [6] Zweiri, Y.H., Seneviratne, L D., Althoefer, K.: Stability Analysis of a Three-term Back-propagation Algorithm. J. Neural Networks. 18, 1341- -1347 (2005)
- [7] Shao, H., and Zheng, H.: A New BP Algorithm with Adaptive Momentum for FNNs Training." In: GCIS 2009, Xiamen, China, pp. 16--20 (2009)

- [8] Rehman, M. Z., Nawi, N.M., Ghazali, M. I.: Noise-Induced Hearing Loss (NIHL) Prediction in Humans Using a Modified Back Propagation Neural Network, In: 2nd International Conference on Science Engineering and Technology, pp. 185--189 (2011)
- [9] Swanston, D. J., Bishop, J.M., and Mitchell, R. J.: Simple adaptive momentum: New algorithm for training multilayer Perceptrons. J. Electronic Letters. 30, 1498--1500 (1994)
- [10] Mitchell, R. J., On Simple Adaptive Momentum. In: CIS 2008, London, United Kingdom, pp.01--06 (2008)
- [11] M. Z. Rehman , N. M. Nawi " Improving the Accuracy of Gradient Descent Back Propagation Algorithm (GDAM) on Classification Problems" (IJNCAA) 1(4): 838-847ns, (2011) (ISSN: 2220-9085)
- [12] Anil K Ahlawat, Ankit Gupta, Aniruddha Gupta, Gaurav Malik, Rahul Ramchandani "A Variant Of Back Propagation Algorithm For Feed Forward Network", 2005
- [13] V.V. Joseph , Rajapandian, N. Gunaseeli Modified Standard Backpropagation Algorithm With Optimum Initialization For Feedforward Neural Networks" JISE, GA, USA, ISSN: 1934-9955, vol. 1, no. 3
- [14] Ms. Sonali. B. Maind et. al., Research Paper on Basic of Artificial Neural Network, International Journal on Recent and Innovation Trends in Computing and Communication Volume: 2 Issue: 1 | January 2014
- [15] Eldon Y. Li, "Artificial Neural Networks and their Business Applications", Taiwan, 1994. FLEXChip Signal Processor (MC68175/D), Motorola, 1996
- [16] B. Verma, M. Blumenstein and S. Kulkarni, "A Neural Network based Technique for Data Compression", Griffith University, Gold Coast Campus, Australia.
- [17] Venkata Rama Prasad Vaddella, Kurupati Rama, "Artificial Neural Networks For Compression Of Digital Images: A Review", International Journal of Reviews in Computing, 2009-2010.
- [18] Rafid Ahmed Khalil, "Digital Image Compression Enhancement Using Bipolar Backpropagation Neural Networks", University of Mosul, Iraq, 2006.
- [19] S. Anna Durai, and E. Anna Saro, "Image Compression with Back Propagation Neural Network using Cumulative Distribution Function" World Academy of Sciences, Engineering and Technology, 2006.
- [20] Omaira N.A. AL-Allaf, "Improving the Performance of Backpropagation Neural Network Algorithm for Image Compression/Decompression System", Journal of Computer Science 6, 2010.