Image Compression By Using Back Propagation

T.N.Shankar
Faculty of Comp Sc and Engg,GMRIT,RAJAM,AP,INDIA
tnshankar2003@yahoo.co.in

M.Maheswara rao, D.Sudha Madhurl, K.Sirisha, K.Anilkumar
B.Tech Comp Sc and Engg,GMRIT,RAJAM,AP,INDIA
maheshcse547@gmail.com

G.SAHOO
Faculty of IT & MCA,BIT Mesra,Ranchi
bitmesra.ac.in

Abstract:
Image Compression using Neural Networks is an area where research is being carried out in various directions towards achieving a generalized and economical network. Feedforward Networks using Back propagation Algorithm adopts the method of steepest descent for error minimization is popular and widely adopted and is directly applied to image compression. Various research works are directed towards achieving quick convergence of the network without loss of quality of the restored image. In general the images used for compression are of different types like dark image, high intensity image etc. When these images are compressed using Back-propagation Network, it takes longer time to converge. To achieve this, a cumulative distribution function is estimated for the image and it is used to map the image pixels. When the mapped image pixels are used, the Back-propagation Neural Network yields high compression ratio as well as it converges quickly.

Keywords:
Coding, image ,error, compression, layer

1 INTRODUCTION:
Image compression is minimizing the size in bytes of a graphics file without degrading the quality of the image to an unacceptable level. The reduction in file size allows more images to be stored in a given amount of disk or memory space. It also reduces the time required for images to be sent over the Internet or downloaded from Web pages.

Image compression can be done through various techniques like Huffman coding, back propagation etc. Huffman coding [12] is an entropy encoding algorithm used for lossless data compression. The term refers to the use of a variable-length code table for encoding a source symbol (such as a character in a file) where the variable-length code table has been derived in a particular way based on the estimated probability of occurrence for each possible value of the source symbol. It mainly deals with construction of Minimum-Redundancy Codes.

Back propagation [3,4] is a specific technique for implementing gradient descents in weight space for a multilayer feed forward network. It can be used to compress image [1,2] discussion with algorithm in section-3,by training a net to function as an auto associative net (the training input vector and the target output are the same) with fewer hidden units than there are in the input or output units. The input and the output data file are formed from the given pattern in +1 and -1 form and stored in comp.mat file. This algorithm can be applied even if there is no redundancy in the data.

1.1 Huffman Coding:
Huffman coding [11] is based on the frequency of occurrence of a data item (pixel in images). The principle is to use a lower number of bits to encode the data that occurs more frequently. Codes are stored in a code book which may be constructed for each image or a set of images. In all cases the code book plus encoded data must be transmitted to enable decoding.

The Huffman algorithm is briefly summarized.

1.2 Algorithm:
A bottom-up approach

1. Initialization: Put all nodes in an OPEN list, keep it sorted at all times (e.g., ABCDE).

2. Repeat until the OPEN list has only one node left:

(a)From OPEN pick two nodes having the lowest frequencies/probabilities, create a parent node of them.

(b)Assign the sum of the children's frequencies/probabilities to the parent node and insert it into OPEN.

(c) Assign code 0, 1 to the two branches of the tree, and delete the children from OPEN.

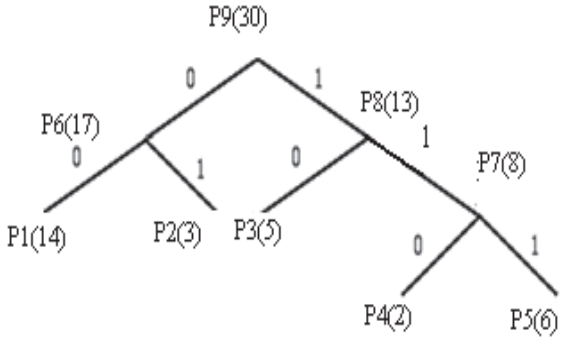


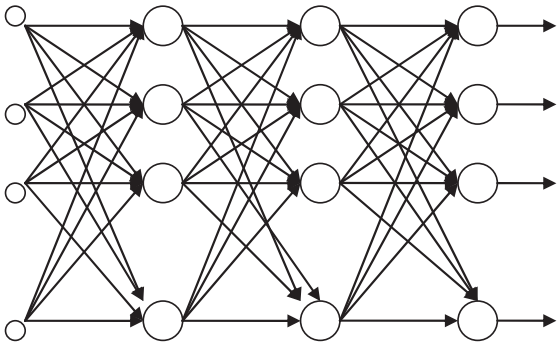
Figure-1

Symbol	Count	log(1/p)	Code	Subtotal(#bits)
P1	14	2.63	00	37
P2	3	1.09	01	3
P3	5	1.60	10	8
P4	2	0.69	110	1
P5	6	1.79	111	11
Total (# of bits)= 60				

Table-1

2 BACK PROPAGATION:

Back-propagation algorithm[7,8] is a widely used learning algorithm in Artificial Neural Networks. The Feed-Forward Neural Network architecture is capable of approximating most problems with high accuracy and generalization ability. This algorithm is based on the error-correction learning rule. Error propagation consists of two passes through the different layers of the network, a forward pass and a backward pass. In the forward pass the input vector is applied to the sensory nodes of the network and its effect propagates through the network layer by layer. Finally a set of outputs is produced as the actual response of the network. During the forward pass the synaptic weight of the networks are all fixed. During the back pass the synaptic weights are all adjusted in accordance with an error-correction rule. The actual response of the network is subtracted from the desired response to produce an error signal. This error signal is then propagated backward through the network against the direction of synaptic conditions. The synaptic weights are adjusted to make the actual response of the network move closer to the desired response.



Input layer Hidden layers Output layer

Figure-2

The back propagation neural network is essentially a network of simple processing elements working together to produce a complex output. These elements or nodes are arranged into different layers: input, middle and output.

The output from a back propagation[5,6] neural network is computed using a procedure known as the forward pass.

2.1 Algorithm:

1.The input layer propagates a particular input vector's components to each node in the hidden layers.

2. Hidden layer nodes compute output values, which become inputs to the nodes of the output layer.

3.The output layer nodes compute the network output for the particular input vector. The forward pass produces an output vector for a given input vector based on the current state of the network weights.

4.Since the network weights are initialized to random values, it is unlikely that reasonable outputs will result before training. The weights are adjusted to reduce the error by propagating the output error backward through the network. This process is where the backpropagation neural network gets its name and is known as the backward pass:

5. Compute error values for each node in the output layer. This can be computed because the desired output for each node is known.

6. Compute the error for the middle layer nodes. This is done by attributing a portion of the error at each output layer node to the middle layer node, which feed that output node. The amount of error due to each middle layer node depends on the size of the weight assigned to the connection between the two nodes.

7. Adjust the weight values to improve network performance using the Delta rule.

8. Compute the overall error to test network performance.

The training set is repeatedly presented to the network and the weight values are adjusted until the overall error is below a predetermined tolerance. Since the Delta rule follows the path of greatest decent along the error surface, local minima can impede training. The momentum term compensates for this problem to some degree.

3 EXPERIMENTS AND OUTPUTS

Let the input image be:

Figure-3 (Lena.jpg)

4 IMAGE COMPRESSION ALGORITHM BY USING BACK PROPAGATION:

Step1: Read the image ("Lena.jpg")

step2: Initialize Input, Hidden and Output layer definition values.

n = 1024

m = 1024

h = 24

Step 3: Initialize weights ,bias and learning parameter

v = rand(n,h) - 0.5

v1 = zeros(n,h)

b1 = rand(1,h)-0.5

b2 = rand(1,m)-0.5

w = rand(h,m)-0.5

w1 = zeros(h,m)

alpha = 0.4

mf = 0.3

con = 1

epoch = 0

Step 4: Find the number of images and feed forward them.

zin(j) = zin(j)+x(I,i)*v(i,j)

z(j) = bipsig(zin(j)

Step 5: Calculate the error and back propagate it.

delk(k) = (t(I,k)-y(k))*bipsig1(yin(k))

Step 6: Weight updating is performed in order to minimize the error.

w=w+delw

b2=b2+delb2

v=v+delv

b1=b1+delb1

Step 7: Put number of epochs

Compression output of image Fig-3(Lena.jpg)

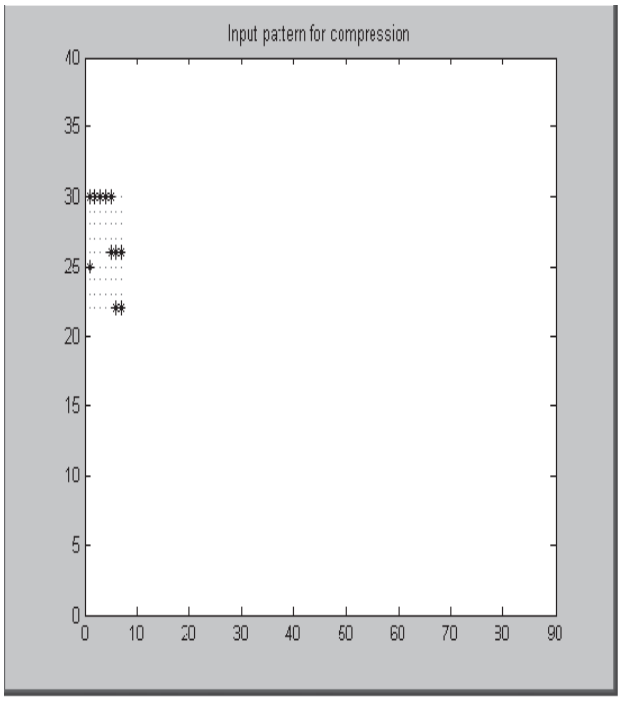


Figure - 4

4.1 Analysis 1

In Fig-4 the compressed image is coming after 30 epochs.We can compress it according to our desire by using more number of epochs till convergence .It is not possible by the Hoffman coding .Hoffman compression method is based on redundancy.If there are rendundant object[10] then we can compress it .Back propagation network[9] based on narrow the difference with their neighborhood pixels ,which depending on epochs and weights.By theHoffman coding we cann't compress it further.Another draw back of Hoffman method is if there is no or redundance very less then we cann't compress the image.Above these drawbacks backpropagation image compression is better than Hoffman coding.

4.2 Analysis 2

From Fig-5 it is clear that error is decresing and moving towards the convergence[4].What is the error after 5 epochs that is that decresing till 15 to 17 epochs .Next to 20 epochs probably it is moving towards the convergence so what is the compression result at 25 epochs that will be same at epoch no 30.So we can get full compressed figure after 25 epochs.

5 CONCLUSION

Our mainwork is focused on the image compression by using backpropagation,which exhibits a clear-cut idea on application of backpropagation with special features compare to Hoffman code . By using this algorithm we can save more memory space, and in case of web applications transfering of images and download should be fast.

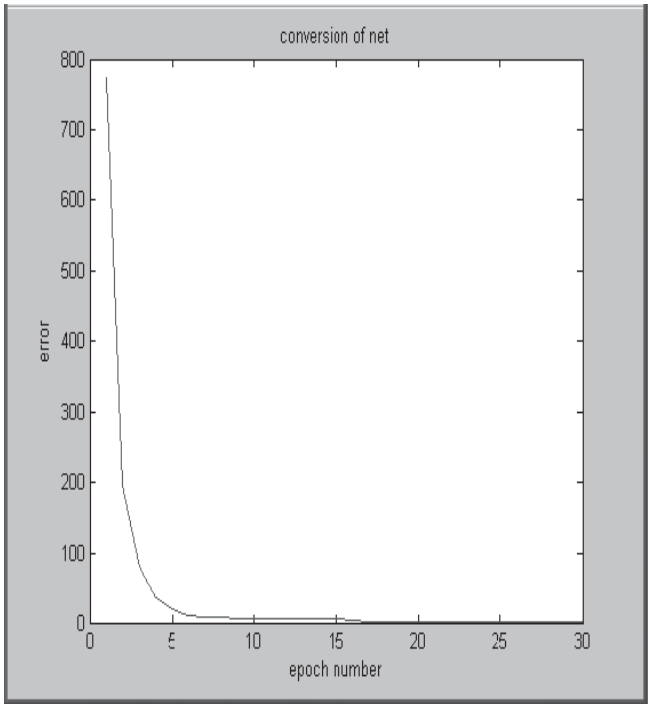


Figure - 5

6 FUTURE DEVELOPMENT:

We can develop this project for image ecryption and decryption.In this process compression of image is as similar as data encryption,and decompression as data decryption.Only produce a secret key to thwart the attaker.

7 ACKNOWLEDGEMENT

We thank GMRIT,CSE dept who inspired and support us to take up suchtype of projects and providing us necessary requirements.

8 REFERENCES:

[1] M.Egmont-Petersen, D.de.Ridder, Handels, "Image Processing with Neural Networks – a review", Pattern Recognition 35(2002) 2279 2301 www.elsevier.com

[2] Fethi Belkhouche, Ibrahim Gokcen, U.Qidwai, "Chaotic gray-level image transformation, Journal of Electronic Imaging -- October - December 2005 Volume 14, Issue 4, 043001

[3] Mohammed A.Otair, Walid A. Salameh, "Speeding up Back-propagation Neural Networks" Proceedings of the 2005 Informing Science and IT Education Joint Conference.

[4] Bogdan M.Wilamowski, Serdar Iplikci, Okyay Kaynak, M. Onder Efe "An Algorithm for Fast Convergence in Training Neural Networks".

[5] Hahn-Ming Lee, Chih-Ming Cheb, Tzong-Ching Huang, "Learning improvement of back propagation algorithm by error saturation prevention method", Neurocomputing, November 2001.

[6] M.A.Otair, W.A.Salameh (Jordan), "An Improved Back-Propagation Neural Networks using a Modified Non-linear function", The IASTED Conference on Artificial Intelligence and Applictions, Innsbruck, Austria, February 2006.

[7] Simon Haykin, "Neural Networks – A Comprehensive foundation", 2nd Ed., Pearson Education, 2004.

[8] T.N.SHANKAR"Neural Networks,University Science Press,2008.

[9] B.Verma, B.Blumenstin and S. Kulkarni, Griggith University, Australia, "A new Compression technique using an artificial neural network".

[10] Rafael C. Gonazalez, Richard E.Woods, "Digital Image Processing", 2nd Edn. Prentice Hall, Inc.

[11] William B. Pennebaker and Joan L. Mitchell (1993). JPEG still image data compression standard (3rd ed.). Springer. p. 291. ISBN 9780442012724.

[12] Huffman's original article: D.A. Huffman, "A Method for the Construction of Minimum-Redundancy Codes", Proceedings of the I.R.E., September 1952, pp 1098–1102

Author Biographies

T.N.Shankar has obtained his M.Tech. in computer Science from Birla Institute of Technology ,Mesra,Ranchi. At present he is working as Asst. Professor Computer Science & Engg in the GMR Institutte of Techology, Rajam, AP. He has 6 years teaching experience. He has about seven publications in his credit. His research interests include with Information Security and Neural Networks. He has published a book on Neural Networks through UNIVERSITY SCIENCE PRESS, New Delhi. Presently he is pursuing Ph.D. in Computer Science, at Birla Institute of Technology ,Mesra,Ranchi, India.

G. Sahoo received his M.Sc. degree in Mathematics from Utkal University in the year 1980 and Ph. D. degree in the area of Computational Mathematics from Indian Institute of Technology, Kharagpur in the year 1987. He has been associated with Birla Institute of Technology, Mesra, Ranchi , India since 1988 and is currently working as Professor and Head in the Department of Information Technology. His research interest includes Theoretical Computer Science, Parallel and Distributed Computing, Evolutionary Computing, Information Security, Image Processing and Pattern Recognition.