Calculating Error for Compressed Image Trained By Neural Network

Neha Jaiswal, Ayub Khan

Student, Assistant Professor
Communication and Information system
Shivdan Singh Institute of Technology and Management, Aligarh, India

---

Abstract **- Uncompressed multimedia (graphics, audio and video) data requires considerable storage capacity and transmission bandwidth. There is rapid progress in processor speeds, mass-storage density and digital communication system performance in result demand for data-transmission bandwidth and data storage capacity continues to explore the capabilities of available technologies. The growth of data communication multimedia-based web applications has not only undergo the need for more efficient ways to data storage and digital communication technology. Image compression presented by JPEG, H.26x and MPEG standards uses new technology like algorithms of neural networks are developed to seek the future of image coding. Successful applications of neural network algorithms have now well established and other involvement of neural network in technology is appreciable. In this paper we present the development of neural network training for image compression. Most popular way to show the power of neural network for image compression follows**

(a) Selection of multi layered network
(b) Selection of methods for training process
(c) Test vector.
Based on these points network are trained and implemented.

In this paper an image has been trained in efficient multi-layered neural network and tested using MATLAB for error occurred in original image. Calculation of NMSE, SNR, PSNR, entropy and error using histogram is done in this paper.

Index Terms **– Neural Network, Image compression, MATLAB**

I. Introduction

Image compression is a task of minimizing the amount of data required to store or transmit an image. The representation of images often requires large number of bits and transmission or storage of an image is impractical without reducing the amount of bits. The image compression uses information coding technique. A typical image's energy varies significantly throughout the image which makes compression of image difficult in the spatial domain. The images are able to have a condensed representation in frequency domain packed around low frequencies it makes compression in frequency domain more effective. Image compression uses transform coding technique which first switches frequency domain and then perform compressing task. Its mandatory to use decor related transform coefficients to minimize redundancy and to store maximum amount of information in smallest space. Coefficients are then coded as accurate as possible.

For implementation the image compression algorithm is divided into various steps:
The weighted sum of the inputs in calculated, it computes the total weighted input Xj using formula

$Xi = \sum_i Yi\ Wij$

Where Yi is the activity level of the jth unit in the previous layer and Wij is the weight of the connection between the ith and the jth unit. Then the weighted Xj is passed through a sigmoid function that would scale the output in between 0 and 1. Next, the unit calculates the activity yj using some function of the total weighted input.

$$Y_j = \frac{1}{1+e^{-x_j}}$$

When the output is calculated it is compared with the required output. Once the activities of all output units have been determined the network computes the error E by using following expression

$$E = \frac{1}{2}\sum_j \left(Y_j - d_j\right)^2$$

where Yj is the activity level of the ith unit in the top layer and dj is the desired output of the ith unit. Now the error is propagated backwards.

1. Compute how fast the error changes as the activity of an output unit changed. This error derivative (EA) is the difference between the actual and the desired activity.

$$EA_j = \frac{\delta E}{\delta Y_j} = Y_j - d_j$$

2. Compute how fast the error changes as the total input received by an output unit changed. EI is the answer from step 1 multiplied by the rate at which the output of a unit changes as its total input is changed.

$$EI_j = \frac{\delta E}{\delta x_j} = \frac{\delta E}{\delta Y_j} \times \frac{\delta Y_j}{\delta x_j} = EA_j Y_j (1 - y_j)$$

3. Compute how fast the error changes as a weight on the connection into an output unit changed. EW is the answer from step 2 multiplied by the activity level of the unit from which the connection emanates.

$$EW_{ij} = \frac{\delta E}{\delta W_{ij}} = \frac{\delta E}{\delta x_j} \times \frac{\delta x_j}{\delta W_{ij}} = EI_j Y_i$$

4. Compute how fast the error changes as the activity of a unit in the previous layer is changed. Here this step allows back propagation to be applied to multi layer networks. Whenever the activity of a unit in the previous layer changes it affects the activities of all the output units to which it is connected. To compute the overall effect on the error we add together all these separate effects on output units. It is the answer in step 2 multiplied by the weight on the connection to that output unit. Using steps 2 and 4 we can convert the EA's of one layer of units into EA's for the previous layer. This procedure can repeat to get the EA's for as many previous layers are desired. Once we know the EA of a unit we can use steps 2 and 3 to compute the EW's on its incoming connections.

$$EA_i = \frac{\delta E}{\delta Y_i} = \sum_j \frac{\delta E}{\delta x_j} \times \frac{\delta x_j}{\delta Y_i} = \sum_j EI_j W_{ij}$$

II. BACK PROPOGATION ALGORITHM TO TRAIN THE IMAGE

1. Initialize the weights to small random values.
2. Select a training vector pair (input and the corresponding output) from the training set and present the input vector to the inputs of the network.
3. Calculate the actual outputs this is the forward phase.
4. According to the difference between actual and desired outputs (error). Adjust the weights Wo and Wh to reduce the difference this is the backward phase.
5. Repeat from step 2 for all training vectors.
6. Repeat from step 2 until the error is negligibly small.
7. Back Propagation learning algorithm. In the forward phase the hidden layer weight matrix h W is multiplied by the input vector X=(X1, X2, X3…….X n) to calculate the α.

III. OBSERVATION

Image for compress/decompress: D:\Neha\fabric.png
Image Size = 128 X 128
Image has been compressed. It took 28 seconds.

=
Peak Signal to Noise Ratio

=> PSNR= 75.9037dB

Signal to Noise Ratio

=> SNR= 12.2105dB

Normalized Mean Square Error

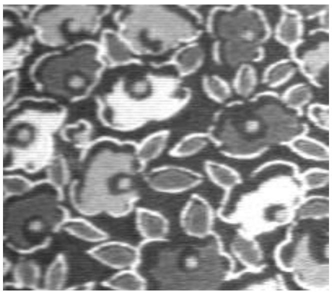
=> NMSE= 0.0016

The first-order estimate of the entropy of Original Image.

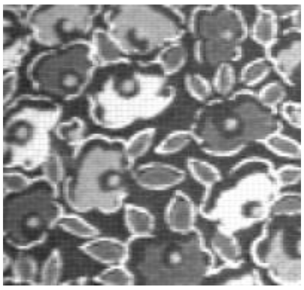
=> h1= 7.4278

The first-order estimate of the entropy of Decompressed Image.

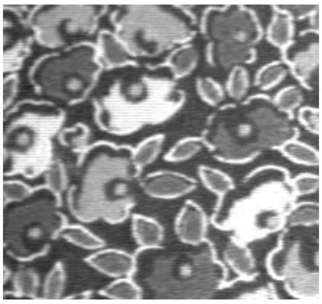
=> h2 = 7.5121



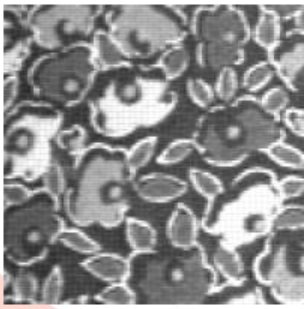
orignal image



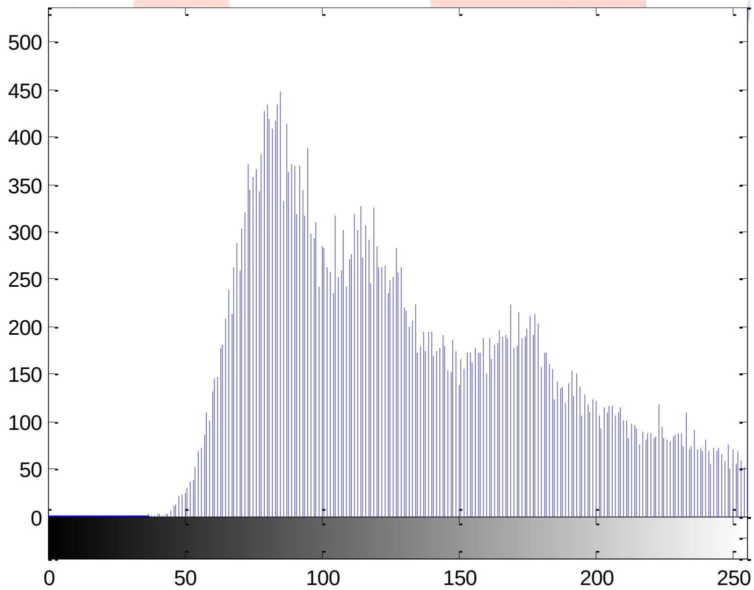
compressed image



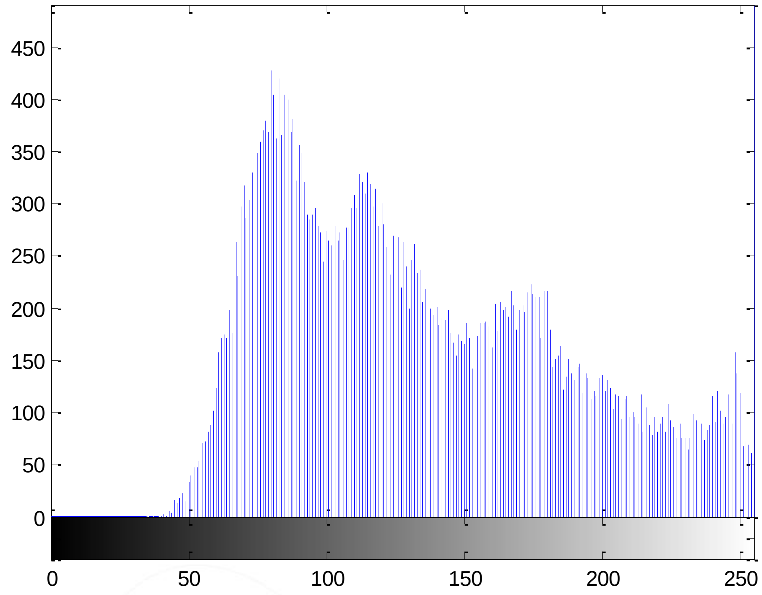
error image



decompressed image



Error histogram 1



Error histogram 2

IV. CONCLUSION

In the present scenario data compression has become the fundamental requirement for transmission and storage purposes; in fact it is the back bone of data communication via internet facilities, multimedia communication, and teleconferencing. Data may be whether audio, video or image.

In this paper, the area of consideration is image compression and image coding, particularly for image transmission applications. Thus image compression using Back-Propagation Neural Network has been implemented and tested for still image coding. For achieving the compression, basic approach is that instead of transmitting the whole image, only the state vectors of hidden layer along with the weights associated with the output layer are proposed for transmission. This method is applicable to the problem of image compression and transmission over low bandwidth communication channels.

To test the compression performance along with the image quality measures of reconstructed image after decompression, six images of two different sizes are used. Table 7.1. shows the performance of image compression system using Back-Propagation algorithm for compression and decompression of various images. (refer to testing results in chapter 7) four image quality performance measures such as NMSE, PSNR, SNR and ENTROPY along with one compression performance measure i.e.

In conclusion, the present development of back-propagation neural network has a long distance to achieve the status of different traditional Image compression techniques. Finally, further research work in neural network base image compression can be achieved, via associating with the traditional transform domain image compression techniques such as Wavelet Transform based Neural Network.

REFERENCES

[1] K. Sayood, "Introduction to Data Compression", Second Edition.
[2] A. Jain. "*Fundamentals of Digital Image Processing"*, Prentice-Hall, 1989.
[3] R.C.Gonzalez, R.E.Woods, "Digital Image Processing", Pearson Education.
[4] R.C.Gonzalez, R.E.Woods, S.L.Eddins, "Digital Image Processing using MATLAB", Pearson Education 2005.
[5] R. Williams., "*Adaptive Data Compression*", Kluwer Academic Publishers, 1991.
[6] N. Faller., "An adaptive system for data compression". In *Proceedings of the Asilomar Conference on Circuits, Systems and Computers*, pages 593-597, 1973.
[7] R. Gallager. "Variations on a theme by Huffman." *IEEE Transactions on Information Theory*, 24:668-674, 1978
[8] A. Habibi. "Survey of adaptive image coding techniques". *IEEE Transactions on Communications*, 25:1275-1284, 1977.
[9] D. Knuth. "Optimal binary search trees." *Acta Informatica*, 1:14-25, 1971.