International Journal of Image Processing and Vision Science

Volume 1 | Issue 3 Article 10

January 2013

DESIGN OF NEURAL NETWORK AS DATA FLOW MODEL FOR IMAGE COMPRESSION

DANESHWARI I. HATTI
Dept of ECE, S.D.M College of Engineering & Technology, Dharwad, India, daneshwari_hatti@yahoo.co.in

SAVITRI RAJU
Dept of ECE, S.D.M College of Engineering & Technology, Dharwad, India, savitriraju@sdmcet.ac.in

MAHENDRA M. DIXIT
Dept of ECE, S.D.M College of Engineering & Technology, Dharwad, India, mmdixitmm@yahoo.co.in

Follow this and additional works at: https://www.interscience.in/ijipvs

Part of the Robotics Commons, Signal Processing Commons, and the Systems and Communications Commons

Recommended Citation

HATTI, DANESHWARI I.; RAJU, SAVITRI; and DIXIT, MAHENDRA M. (2013) "DESIGN OF NEURAL NETWORK AS DATA FLOW MODEL FOR IMAGE COMPRESSION," *International Journal of Image Processing and Vision Science*: Vol. 1 : Iss. 3 , Article 10.
Available at: https://www.interscience.in/ijipvs/vol1/iss3/10

This Article is brought to you for free and open access by Interscience Research Network. It has been accepted for inclusion in International Journal of Image Processing and Vision Science by an authorized editor of Interscience Research Network. For more information, please contact sritampatnaik@gmail.com.

DESIGN OF NEURAL NETWORK AS DATA FLOW MODEL FOR IMAGE COMPRESSION

DANESHWARI I. HATTI[1], SAVITRI RAJU[2] & MAHENDRA M. DIXIT[3]

[1,2&3]Dept of ECE, S.D.M College of Engineering & Technology, Dharwad, India
E-mail : daneshwari_hatti@yahoo.co.in, savitriraju@sdmcet.ac.in, mmdixitmm@yahoo.co.in

Abstract – In digital communication bandwidth is essential parameter to be considered. Transmission and storage of images requires lot of memory in order to use bandwidth efficiently neural network and Discrete cosine transform together are used in this paper to compress images. Artificial neural network gives fixed compression ratio for any images results in fixed usage of memory and bandwidth. In this paper multi-layer feedforward neural network has been employed to achieve image compression. The proposed technique divides the original image in to several blocks and applies Discrete Cosine Transform (DCT) to these blocks as a pre-process technique. Quality of image is noticed with change in training algorithms, convergence time to attain desired mean square error. Compression ratio and PSNR in dB is calculated by varying hidden neurons. The proposed work is designed using MATLAB 7.10. and synthesized by mapping on Vertex 5 in Xilinx ISE for understanding hardware complexity.

***Keywords** - backpropagation, Discrete cosine transform, image compression, mean square error, neural network, PSNR.*

I. INTRODUCTION

Importance of image compression increases with advancing communication technology. The amount of data associated with visual information is so large that its storage requires enormous storage capacity. The storage and transmission of such data require large capacity and bandwidth, which could be very expensive. Image data compression techniques are concerned with reduction of the number of bits required to store or transmit images without any appreciable loss of information. Compression makes it possible for creating file sizes of manageable, storable and transmittable dimensions. Compression is achieved by exploiting the redundancy [1]. DCT represents image as a sum of sinusoids of varying magnitudes and frequencies. The DCT has a property that, for a typical image, most of the visually significant information about the image is concentrated in just a few coefficients of the DCT. Hence DCT is often used for image compression. The neural network has good performance in non-linear capacity [2]. Neural network is used for further compression in further reducing size. It has proved that the multi-layer neural network can accurately approximate to any linear or nonlinear function. Through learning the incomplete data, the neural network can achieve an accurate prediction of the whole data with its generalization capability. Neural network can be utilized to approximate to the continuous quantity, because any continuous quantity can be express by the combinations of linear and nonlinear function. Hence the neural network is designed to store the information of the continuous quantity.

Back-propagation algorithm [3] is a widely used learning algorithm in Artificial Neural Networks. In this paper Backpropagation neural network is designed and trained using different learning algorithms. The input data is entered into the network via the input layer. Each neuron in the network processes the input data with the resultant values steadily "percolating" through the network, layer by layer, until a result is generated by the output layer. The actual output of the network is compared to expected output for that particular input. This results in an error value. The connection weights in the network are gradually adjusted, working backwards from the output layer, through the hidden layer, and to the input layer, until the correct output is produced. Fine tuning the weights in this way has the effect of teaching the network how to produce the correct output for a particular input, i.e. the network learns. The general parameters deciding the performance of Backpropagation Neural Network Algorithm includes the mode of learning, information content, activation function, target values, input normalization, initialization, learning rate and momentum factors. [4], [5], [6].The neural network structure can be illustrated in Figure1. Three layers, one input layer, one output layer and one hidden layer. Both of input layer and output layer are fully connected to hidden layer. Compression is achieved by designing the value of the number of neurons at the hidden layer, less than that of neurons at both input and output layers.

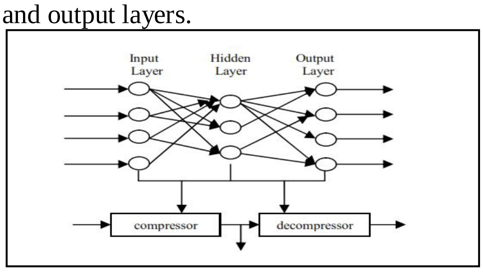


Fig. 1: The basic architecture for image compression

II. METHODOLOGY

Image compression is done using neural network by varying number of hidden neurons and made less than input layer neurons. Three layered neural network is designed and trained using backpropagation algorithm and applied with different learning algorithms such as one step secant (trainoss), BFGS quasi Newton (trainbfg) and gradient descent backpropagation with adaptive learning rate algorithm (traingda).

DCT2 is used for images. It is represented as shown in eqn1

$$C(u,v)=\alpha(u)\alpha(v)\sum_{x=0}^{N-1}\sum_{y=0}^{N-1} f(x,y)\cos\left[\frac{\Pi(2x+1)u}{2N}\right]\cos\left[\frac{\Pi(2y+1)v}{2N}\right] \quad (1)$$

for u,v = 0,1,2,…,N −1 and α(u) and α(v)

$$\alpha(u)=\sqrt{\frac{1}{N}} \quad \text{for } u = 0$$

$$\sqrt{\frac{2}{N}} \quad \text{for } u \neq 0$$

The output is calculated at each node using eqn 2

$$output = f\left(\sum_i a_i w_i\right) \quad (2)$$

f is activation function represented in eqn3

$$f(x)=\frac{2}{1+e^{-2x}}-1 \quad (3)$$

Backpropagation refers to back propagation of error that is

$$E = (\text{target} - \text{output})^2 \quad (4)$$

In this paper tangent sigmoid is used in hidden layer shown in eqn3 and pure linear is used at output layer as activation function. The error obtained is back propagated to adjust the weights until error is minimized using delta rule represented in eqn3

$$w_{new} = w_{old} + \alpha \partial E / \partial w_{old} \quad (5)$$

where α is the learning rate.

Algorithm1

1. First read the input image, here the cameraman image is used for processing. The image is of 256x256 and it has to be segmented in to 8x8 blocks or apply DCT for 8x8 blocks and rearrange each block in to column vector and form it as 64x1024 size image matrix.
2. Apply this input image to neural network and consider that image as target image for training neural network.
3. After training the neural network the output obtained at the output layer is rearranged to 256x256. Hidden layer output is the compressed image it consist of hidden neurons hi<ni, hi is number of neurons in hidden layer and ni is number of neurons in input layer. It is of size 16x1024 rearrange to a matrix of 128x128 it is compressed image.
4. Repeat step2 to step 4 for learning algorithms namely trainbfg, traingda and trainoss.

III. RESULTS

A. Image compression using backpropagation neural network and different training algorithms

i. using trainbfg

Fig. 2a: Original image of size 256x256

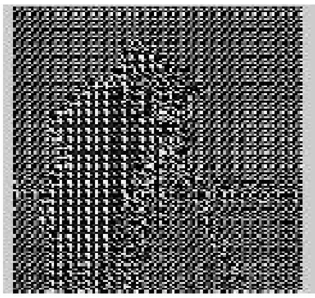


Fig. 2b: Compressed image of size 128x128

Fig. 2c : Reconstructed image of size 256x256

The network is trained using trainbfg for 1000 epochs and error is decreased and network is trained using trainbfg algorithm. Fig 2a, Fig 2b & Fig 2c shows the input image and compressed image which is output of hidden layer and reconstructed image obtained at output layer respectively. It can be easily shown that reconstructed image needs more iteration to converge and hence goal is set as 0.01.

ii. Using traingda

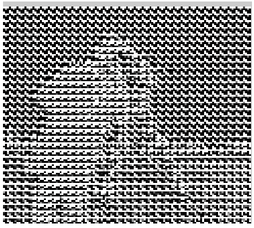


Fig. 4a: Compressed image of size 128x128

Fig. 4b: Reconstructed image of size 256x256

iii. Using trainoss

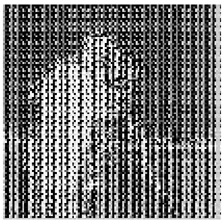


Fig. 5a: Compressed image of size 128x128

Fig. 5b: Reconstructed image of size 256x256

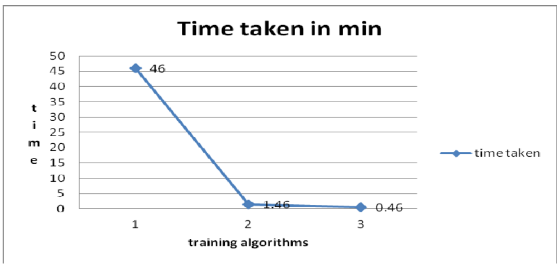


Fig. 6 : Time taken for training algorithms

Figure 6 shows th evariation of convergence time for trainbfg(1), traingda(2) & trainoss(3) in min.training algorithm takes less time for assigned MSE resulting in goood PSNR.

B. Reconstructed image for untrained network

Fig. 7a: Input image

Fig.7b: Reconstructed image

Figure 7b shows the decompressed image obtained from the network which was trained for different image.

C. DCT as preprocessing technique

DCT is applied for 8x8 block and arranged in to column vector and hence given as input image for neural network. The network is trained using trainoss algorithm. The results obtained for considering all DCT coefficients is shown in Figure 8a and Figure 8b.

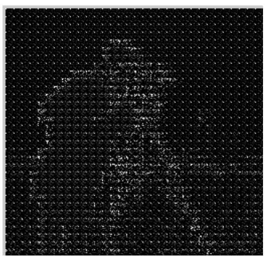


Fig. 8a: DCT image

Fig. 8b: Reconstructed image

Figure 8b shows the decompressed image obtained from trained neural network.

Fig. 8c: Reconstructed image for untrained neural network

Figure 8c shows the decompressed image obtained from untrained neural network.

Fig. 8d: Reconstructed image for trained neural network

Figure 8d represents the decompressed image obtained from neural network when considering low frequency DCT coefficients that is 4x4 from each 8x8 block of whole image 256x256. Hence only 16 coefficients are considered out of 64 coefficients which results to compression.

D. Comparison of PSNR in dB obtained using trainoss algorithm

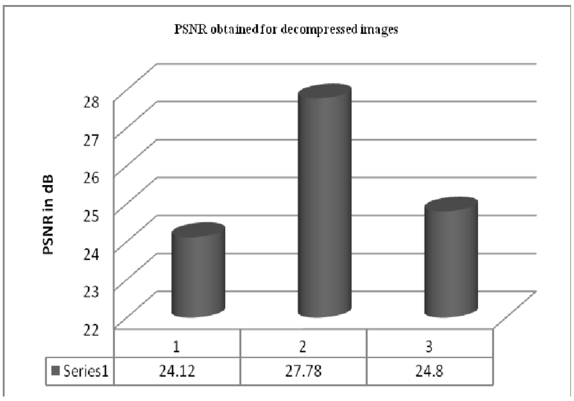


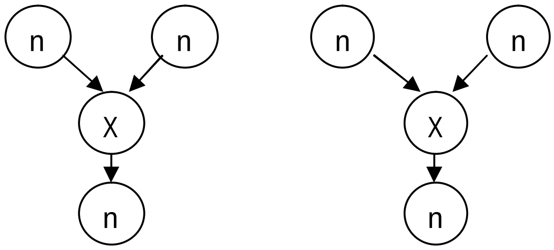
Fig. 9: PSNR for trained network

Figure 9 describes varying PSNR in dB obtained from trained network for traditional method in figure it is given as (1),DCT as preprocessing method considering all coefficients (2) & considering only 4x4 coefficients from each 8x8 block(3). PSNR obtained for test image given to this trained network is 23.02 for another approach is 24.08.

E. Neural network as data flow model

The design proposed consists of matrix multiplication of two matrices, one is the input image samples, and the second is the weight matrix obtained after training. This multiplied output is passed through the nonlinear transfer function to obtain the compressed output that gets transmitted or stored in compressed format. On the decompression side, the compressed data in matrix form is multiplied with the weight matrix to get back the original image.

Neural network is modeled as data flow model. Input matrix is saved in one node n1 and weight matrix in another node n2, multiplication is performed between this two matrixes the output obtained is sent through the node n3, it behaves as activation function. Hence compressed image is stored in node n4.The weight matrix between hidden and output layer is stored in node n5. The output obtained from n4 and matrix stored in node n5 is multiplied results in to reconstructed image.



(a) (b)

Fig. 10(a) Compressor (b) Decompressor

Figure 10a & Figure 10 b shows data flow graph model for compression and decompression of image designed using HDL.

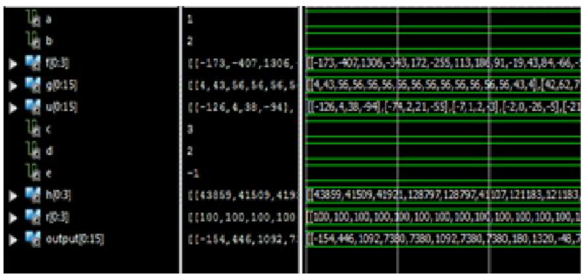


Fig. 11: Simulation results of compressed and reconstructed image

Figure 11 shows the results for proposed work, here input image matrix is in integer and weight matrix is also rounded to integer value. In graph h[0:3] represents 4x16 output obtained for multiplication of input image and weight matrix, r[0:3] is the compressed image matrix of size 4x16 and output[0:15] is reconstructed image matrix of size 16x16.

IV. CONCLUSION

This paper has introduced a comparison among backpropagation training algorithms in image compression namely One step secant (trainoss), BFGS quasi Newton (trainbfg) and gradient descent backpropagation with adaptive learning rate algorithm (traingda).

The employed algorithms were tested for against parameters, like the number of epochs, and goal. With respect to time of execution, it is important to choose algorithms of acceptable time of execution. Finally, Image compression could be achieved with high accuracy using feedforward back propagation. The best reconstructed image is obtained for trainoss learning algorithm with less time. Using DCT as preprocessing technique gives better result compared to traditional method. Compression ratio is fixed for 1:2:1 neural network that is 1 hidden layer and 1 output layer having 64 input neurons and 16 hidden neurons are resulting 4:1 compression ratio.DCT using as preprocessing techniques looks that compression achieved is more that is after applying DCT less number of bits are required to represent coefficients then feeding to network still compresses and better reconstruction is also achieved through neural network. Hence this kind of method can be used as it compresses in same ratio for any images and hence bandwidth will be fixed which can be used for different purpose. The proposed neural network designed as data flow model in VHDL and simulated.

Integer DCT can be performed as preprocessing technique then it is to be fed as input image for this trained network. The output obtained is to be compared with results obtained in this paper.

ACKNOWLEDGMENT

The authors wish to thank the authorities at AICTE for granting RPS to the department of ECE, SDM college of Engineering and Technology Dharwad. This work is towards partial fulfillment of the said project. The authors are also grateful to the authorities of SDMCET Dharwad, for the facility extended to carry out the present work. The authors also would like to thank the unknown reviewers for their invaluable comments which help in improving the paper.

REFERENCES

[1] Rafael C. Gonzalez and Richard E. Woods, Digital Image Processing, Addison-Wesley Publishing, 1993.

[2] Zhe Xue Guo, Zhi Qiang Shun, The Neural Network and The Realization of The Matlab 2007 ,Publishing House Of Electronic Industry, Beijing, China, 2007,9.

[3] Simon Haykin, “Neural Networks – A Comprehensive foundation”, 2nd Ed., Pearson Education, 2004.

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

[5] Fethi Belkhouche, Ibrahim Gokcen, U.Qidwai, “Chaotic gray-level Image transformation, Journal of Electronic Imaging -- October - December 2005 -- Volume 14, Issue 4, 043001 (Received 18 February 2004; accepted 9 May 2005; published online 8 November 2005.

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

◈◈◈