Image Compression with Back-Propagation Neural Network using Cumulative Distribution Function

S. Anna Durai, and E. Anna Saro

Abstract—Image Compression using Artificial Neural Networks is a topic where research is being carried out in various directions towards achieving a generalized and economical network. Feedforward Networks using Back propagation Algorithm adopting 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. The reason for this is, the given image may contain a number of distinct gray levels with narrow difference with their neighborhood pixels. If the gray levels of the pixels in an image and their neighbors are mapped in such a way that the difference in the gray levels of the neighbors with the pixel is minimum, then compression ratio as well as the convergence of the network can be improved. 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—Back-propagation Neural Network, Cumulative Distribution Function, Correlation, Convergence.

I. INTRODUCTION

IMAGE compression research aims at reducing the number of bits needed to represent an image. Image compression algorithms take into account the psycho visual features both in space and frequency domain and exploit the spatial correlation along with the statistical redundancy. However, usages of the algorithms are dependent mostly on the information contained in images. A practical compression algorithm for image data should preserve most of the characteristics of the data while working in a lossy manner and maximize the gain and be of lesser algorithmic complexity. In general almost all the traditional approaches adopt a two-stage process, first, the data is transformed into some other domain and or represented by the indices of the codebook, followed by an encoding of the transformed coefficients or the codebook indices. The first stage is to minimize the spatial correlation or to make use of the correlation so as to reduce the data. Most commonly adopted approaches rely on the transformed techniques and or the use of vector Quantization. Most of the algorithms exploit spatial correlations. Discrete cosine transform is used practically in almost all image compression techniques. Wavelet Transform has been proven to be very effective and has achieved popularity over discrete cosine transform. However, inter-pixel relationship is highly non-linear and un-predictive in the absence of a prior knowledge of the image itself. So, predictive approaches would not work well with natural images. Transform based approaches have been used by most of the researchers in some combination or the other. Among the non-transformed approaches Vector Quantization based techniques encodes a sequence of samples rather than encoding a sample and automatically exploits both linear and non-linear dependencies. It is shown that Vector Quantization is optimal among block coding techniques, and that all transform coding techniques can be taken as a special case of Vector Quantization with some constraints. In Vector Quantization, approximating a sequence to be coded by a vector belonging to a codebook performs encoding. Creation of a straight and unconstrained codebook is a computationally intensive and the complexity grows exponentially with the block.

Artificial Neural Networks have been applied to image compression problems, [1] due to their superiority over traditional methods when dealing with noisy or incomplete data. Artificial Neural networks seem to be well suited to image compression, as they have the ability to preprocess input patterns to produce simpler patterns with fewer components. This compressed information preserves the full information obtained from the external environment. Not only can Artificial Neural Networks based techniques provide sufficient compression rates of the data in question, but also security is easily maintained. This occurs because the compressed data that is sent along a communication line is encoded and does not resemble its original form. Many different training algorithms and architectures have been used. Different types of Artificial Neural Networks have been trained to perform Image Compression. Feed-Forward Neural

Networks, Self-Organizing Feature Maps, Learning Vector Quantizer Network, have been applied to Image Compression. These networks contain at least one hidden layer, with fewer units than the input and output layers. The Neural Network is then trained to recreate the input data. Its bottleneck architecture forces the network to project the original data onto a lower dimensional manifold from which the original data should be predicted. The Back Propagation Neural Network Algorithm performs a gradient-descent in the parameter space minimizing an appropriate error function. The weight update equations minimize this error. The general parameters deciding the performance of Back Propagation Neural Network Algorithm includes the mode of learning, information content, activation function, target values, input normalization, initialization, learning rate and momentum factors. [2], [3], [4], [5], [6]. The Back-propagation Neural Network while used for compression of various types of images namely standard test images, natural images, medical images, satellite images etc, takes longer time to converge. The compression ratio achieved is also not high. To overcome these drawbacks a new approach using Cumulative Distribution Function is proposed in this paper.

This paper is divided into four sections. The theory of Back-propagation Neural Networks for compression of images is discussed in Section II. Mapping of image pixels by estimating the Cumulative Distribution Function of the image to improve the compression ratio and convergence time of the Neural Network is explained in Section III. The experimental results are discussed in section IV followed by Conclusion in section V.

II. BACK-PROPAGATION ALGORITHM

Back-propagation algorithm [7] is a widely used learning algorithm in Artificial Neural Networks. The Feed-Forward Neural Network architecture (Fig. 1) 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.
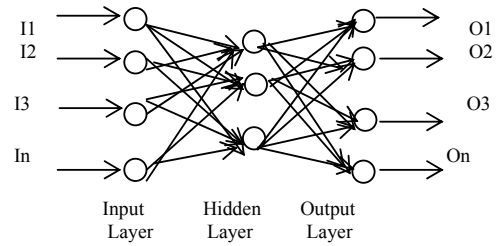

Fig. 1 Feed-Forward Neural Network Architecture

A. Algorithm

Step 1: Normalize the inputs and outputs with respect to their maximum values. It is proved that the neural networks work better if input and outputs lie between 0-1. For each training pair, assume there are 'I' inputs given by $\{I\}_I$ and 'n' out puts $\{o\}_o$ in a normalized form.
$l \times 1$ $n \times 1$

Step2: Assume the number of neurons in the hidden layer to lie between $l<m<2l$

Step3: [v] represents the weights of synapses connecting Input neurons and hidden neurons and [w] represents weights of synapses connecting hidden neurons and output neurons. Initialize the weights to small random values usually from -1 to 1.For general problems, λ can be assumed as 1 and the threshold values can be taken as zero.

$[v]^o = [\text{random weights}]$
$[w]^o = [\text{random weights}]$
$[\Delta v]^o = [\Delta w]^o = [o]$

Step4: For the training data, present one set of inputs and outputs. Present the pattern to the input layer as inputs to the input layer. By using linear activation function, the output of the input layer may be evaluated as

$$\{o\}_r = \{r\}_r \qquad (1)$$
$$l \times 1 \quad l \times 1$$

Step 5: Compute the inputs to the hidden layer by multiplying corresponding weights of synapses as

$$\{I\}_H = [v]^T \{O\}_r \qquad (2)$$
$$m \times 1 \ m \times l \ l \times 1$$

Step6: Let the hidden layer units evaluate the output using the sigmodial function as

$$\{O\}_{H=} \left\{ \begin{array}{c} \vdots \\ \dfrac{1}{(1+e^{-\lambda I_{Hi}})} \\ \vdots \end{array} \right\}$$
$$m \times 1 \qquad (3)$$

Step7: Compute the inputs to the output layer by multiplying corresponding weights of synapses as

$$\{I\}_o = [W]^T \{o\}_H \qquad (4)$$
$$n \times 1 \ n \times m \ m \times 1$$

Step8: Let the output layer units evaluate the output using sigmoidal function as

$$(0)_0 = \left\{ \begin{array}{c} \bullet \\ \bullet \\ 1 \\ \overline{(1+e^{-I_{oj}})} \end{array} \right\} \tag{5}$$

The above is the network output.

Step9: Calculate the error and the difference between the network output and the desired output as for the ith training set as

$$EP = \sqrt{\frac{\sum (T_i - O_{oi})^2}{n}} \tag{6}$$

Step10: Find {d} as

$$\{d\} = \left\{ \begin{array}{c} \bullet \\ \bullet \\ (T_k - O_{ok})O_{ok}(1 - O_{ok}) \\ \bullet \\ \bullet \\ n \times 1 \end{array} \right\} \tag{7}$$

Step 11: Find [Y] matrix as

$$[Y] = \{o\}_H <d> \tag{8}$$
$$m \times n \quad m \times 1 \quad 1 \times n$$

Step 12: Find $[\Delta w]^{t+1} = \alpha [\Delta w]^t + \eta[Y]$ (9)
$$m \times n \quad m \times n \quad m \times n$$

Step 13: Find $\{e\} = [w] \quad \{d\}$ (10)
$$m \times 1 \quad m \times n \quad n \times 1$$

$$\{d^*\} = \left\{ \begin{array}{c} \bullet \\ \bullet \\ e_i \; (O_{Hi}) \; (1 - O_{Hi}) \\ \bullet \\ \bullet \end{array} \right\}$$

$$m \times 1 \; m \times 1 \tag{11}$$

Find [X] matrix as

$$[X] = \{O\}_I <d^*> = \{I\}_I <d^*> \tag{12}$$
$$1 \times m \quad 1 \times 1 \; 1 \times m \quad 1 \times 1 \; 1 \times m$$

Step 14: Find $[\Delta v]^{t+1} = \alpha [\Delta v]^t + \eta[X]$ (13)
$$1 \times m \quad 1 \times m \quad 1 \times m$$

Step 15: Find $[v]^{t+1} = [v]^t + [\Delta v]^{t+1}$ (14)
$$[w]^{t+1} = [w]^t + [\Delta w]^{t+1} \tag{15}$$

Step 16: Find error rate as

$$\text{Error rate} = \frac{\sum E_p}{nset} \tag{16}$$

Step 17: Repeat steps 4 -16 unit the convergence in the error rate is less than the tolerance value.

B. Procedure for Image Compression

The image is split into non-overlapping sub-images. Say for example 256 x 256 bit image will be split into 4 x 4 or 8 x 8 or 16 x 16 pixels. The normalized pixel value of the sub-image will be the input to the nodes. The three-layered back propagation-learning network will train each sub-image. The number of neurons in the hidden layer will be designed for the desired compression. The number of neurons in the output layer will be the same as that in the input layer. The input layer and output layer are fully connected to the hidden layer. The Weights of synapses connecting input neurons and hidden neurons and weight of synapses connecting hidden neurons and weight of synapses connecting hidden neurons and output neurons are initialized to small random values from say –1 to +1.The output of the input layer is evaluated using linear activation function. The input to the hidden layer is computed by multiplying the corresponding weights of synapses. The hidden layer units evaluate the output using the sigmoidal function. The input to the output layer is computed by multiplying the corresponding weights of synapses. The output layer neuron evaluates the output using sigmoidal function. The Mean Square error of the difference between the network output and the desired output is calculated. This error is back propagated and the weight synapses of output and input neurons are adjusted. With the updated weights error is calculated again. Iterations are carried out till the error is less than the tolerance. The compression performance is assessed in terms of Compression ratio, PSNR and execution time [8].

III. MAPPING OF PIXELS BY ESTIMATING THE CUMULATIVE DISTRIBUTION FUNCTION OF THE IMAGE

Computational complexity is involved in compression of raw pixels of an image in spatial domain or the mathematically transformed coefficients in frequency domain using Artificial Neural Networks. The efficiency of such Artificial Neural Networks is pattern dependent. An image may contain a number of distinct gray levels with narrow difference with their neighborhood pixels. If the gray levels of the pixels in an image and their neighbors are mapped in such a way that the difference in the gray levels of the neighbor with the pixel is minimum, then compression ratio as well as the convergence of the network can be improved. To achieve this, the Cumulative Distribution Function [9] 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.

Consider an image as a collection of random variables, each with cumulative distribution and density of

$$F_x(x) = \text{Prob } \{X \leq x\} \tag{17}$$

$$p_x(x) = \frac{d}{dx} F_x(x) \tag{18}$$

Now consider forming the following random variable.

$$Y = F_x(x) \tag{19}$$

Here Y is the result of mapping the random variable x through its own Cumulative Distribution Function. The cumulative distribution of Y can be easily computed.

$$F_y(y) = \text{Prob } \{Fx(x) \leq y\}$$

$$= \text{Prob } \{X \le F_x^{-1}(y)\}$$
$$= F_x(F_x^{-1}(y))$$
$$\quad 0 \text{ for } y < 0$$
$$= \ y \text{ for } 0 \le y \le 1$$
$$\quad 1 \text{ for } y > 1 \qquad (20)$$

This shows that y has a uniform distribution on the interval (0,1) Therefore the histogram of an image can be equalized by mapping the pixels through their cumulative distribution function $F_x(x)$. In order to implement histogram equalization, the cumulative distribution function for the image is estimated. It is done using the image histogram. Let h(i) be the histogram of the image formed by computing the number of pixels at gray level i. Typically, the pixels take on the values i=0, …,L-1 where L = 256 is the number of discrete levels that a pixel can take on. The cumulative distribution function can then be approximated by

$$F_x(i) = \frac{1}{h(L-1)} \sum_{j=0}^{j=1} h(j) \qquad (21)$$

Here the normalization term assures that $F_x(L-1)=1$. By applying the concept of equation (17), a pixel of $X_{s\ is}$ equalized at the position $s \in S$ where S is the set of position in the image.

$$Y_s = F_x(X_s) \qquad (22)$$

However, Y_s has a range from 0 to 1 and may not extend over the maximum number of gray levels. To correct these problems, we first compute the minimum and maximum values of Y_s.

$$Y_{max} = \max_{s \in S} Y_s \qquad (23)$$

$$Y_{min} = \min s \in S Y_s \qquad (24)$$

And then we use these values to from Z_s, a renormalized version of Y_s

$$Z_s = (L-1) \frac{F_x(X_s) - Y_{min}}{Y_{max} - Y_{min}} \qquad (25)$$

The transformation form X_s to Z_s. is Histogram Equalization.

Histogram equalization does not introduce new intensities in the image. Existing values will be mapped to new values resulting image with less number of the original number of intensities. Mapping of the pixels by estimating the cumulative Distribution function of the image results in correlation of the pixels and the presence of similar pixel values within the small blocks of image augments the convergence of the network. Further the frequency of occurrence of gray levels in the image will be more are less equal or rather uniform by the mapping .Due to this most of the image blocks will be similar and hence the learning time gets reduced. Since the convergence is quick, it is possible to reduce the number of neurons in the hidden layer to the minimum possible thus achieving high compression ratios without loss in quality of the decompressed image. The quality of the decompressed image and the convergence time

has been experimentally proved to be better than achieved by conventional methods and by the same algorithm without mapping the image by Cumulative Distribution function. The correlation of pixel values plays a vital role in augmenting the convergence of the neural network for image compression and this is a simple mechanism compared to other existing transformation methods, which are computationally complex and lengthy process.

IV. EXPERIMENTAL RESULTS

The performance of the Back-Propagation Neural Network for image compression with the concept of mapping the image by Cumulative Distribution function has been tested in various types of images. Experiments were conducted by varying the sub-image size namely 4x4, 8x8, 16x16 and 32x32 pixels. The Network was also tested by varying the number of Neurons in the Hidden Layer, resulting in Compression Ratios ranging from 4 to 64. The PSNR values for various combinations of the above for Error Tolerance ranging from 0.01 to 0.0001 were obtained. The Quality of restored image and the Speed of Convergence of the Back propagation Neural Network were compared for both the conditions; image without mapping by Cumulative Distribution Function and for the same image mapped by its Cumulative Distribution Function and thereafter input the same in the Network for compression. The comparative results using sub images of size 4x4 pixels and error tolerance of 0.01 for Compression Ratio of 4:1 tested in various types of images are illustrated in Tables I and II. The PSNR value obtained in the Standard Image without mapping is 26.11 dB and the time taken for convergence is 365seconds. By adopting the proposed approach the PSNR achieved is 28.91 dB (Fig.2) and the time taken for convergence is 182 seconds. The time taken for simulation has been reduced to 50%. The PSNR value obtained in the Medical Image without mapping is 26.05 dB and the time taken for convergence is 290seconds. By adopting the proposed approach the PSNR achieved is 29.42dB (Fig.3) and the time taken for convergence is 140 seconds. For the Satellite Image the PSNR value without mapping is 26.32dB and the convergence time is 315 seconds. After mapping the pixels by Cumulative Distribution Function the PSNR value obtained is 29.11(Fig. 4) and the convergence time has been reduced to 160 seconds. Previous Research Works in Image Compression using Back Propagation Neural Networks have obtained Compression Ratios up to 256:1, but against PSNR values in the range of 15 dB. Since convergence of the network is assured in the proposed approach, the desired quality of the restored image and the required compression ratios can be achieved by fixing the error tolerance to any level, considering the time of simulation. The performance of the Back Propagation Neural Network has been substantially improved by the proposed approach. Mapping the image by Cummulative Distribution Function has helped the Back- Propagation Neural Network to converge easily compared to previous Techniques.

V. CONCLUSION

The Back Propagation Neural Network has the simplest architecture of the various Artificial Neural Networks that have been developed for Image Compression. The direct pixel based manipulation, which is available in the field of Image Compression, is still simpler. But the drawbacks are very slow convergence and pattern dependency. Many research works have been carried out to improve the speed of convergence. All these methods are computationally complex in nature, which could be applied only to limited patterns. The proposed approach of mapping the pixels by estimating the Cumulative Distribution Function is a simple method of pre-processing any type of image. Due to the uniform frequency of occurrence of gray levels by this optimal contrast stretching, the convergence of the Back-Propagation Neural Network is augmented. There will not be any loss in data in the pre-processing and hence the finer details in the image are preserved in the reconstructed image. However since the original intensities will be mapped to new values, re-transformation of the Decompressed image is suggested in case of Medical Images.
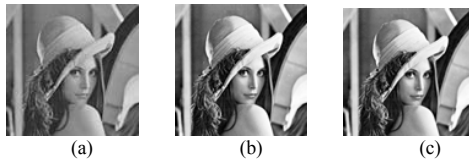

Fig. 2 Standard Image
(a) Original image   (b) Mapped by CDF   (c) Decompressed
PSNR = 28.91dB, Convergence Time = 182 Sec.
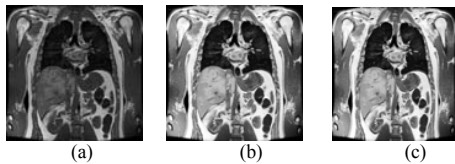

Fig. 3 Medical Image
(a) Original image   (b) Mapped by CDF   (c) Decompressed
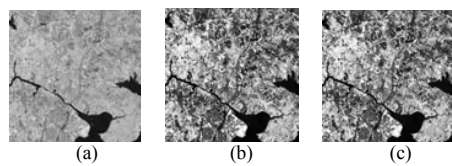PSNR = 29.42dB, Convergence Time = 140 Sec.

Fig. 4 Satellite Image
(a) Original image   (b) Mapped by CDF   (c) Decompressed
PSNR = 29.11dB, Convergence Time = 160 Sec.

TABLE I
EXPERIMENTAL RESULTS OF THE PROPOSED APPROACH

S.No	Image	CR	PSNR (dB)	TIME (Sec)
1	Cameraman	4.1	29.78	148
2	Lena	4:1	28.91	182
3	Pepper	4:1	29.04	188
4	Fruits	4:1	29.79	185
5	Boat	4:1	29.12	178
6	Mandrill	4:1	29.68	161
7	Abdomen(Mri)	4:1	29.42	140
8	Thorax(Mri)	4:1	29.61	132
9	Satellite	4:1	29.11	160

TABLE II
EXPERIMENTAL RESULTS WITHOUT MAPPING

S.No	Image	CR	PSNR (dB)	TIME (Sec)
1	Cameraman	4:1	26.89	320
2	Lena	4:1	26.11	365
3	Pepper	4:1	26.52	368
4	Fruits	4:1	26.96	361
5	Boat	4:1	26.75	342
6	Mandrill	4:1	27.89	318
7	Abdomen(Mri)	4:1	26.05	290
8	Thorax(Mri)	4:1	26.08	265
9	Satellite	4:1	26.32	315

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/locate/patcog
[2] Bogdan M.Wilamowski, Serdar Iplikci, Okyay Kaynak, M. Onder Efe "An Algorithm for Fast Convergence in Training Neural Networks".
[3] 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.
[4] Hahn-Ming Lee, Chih-Ming Cheb, Tzong-Ching Huang, "Learning improvement of back propagation algorithm by error saturation prevention method", Neurocomputing, November 2001.
[5] Mohammed A.Otair, Walid A. Salameh, "Speeding up Back-propagation Neural Networks" Proceedings of the 2005 Informing Science and IT Education Joint Conference.
[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] B.Verma, B.Blumenstin and S. Kulkarni, Griggith University, Australia, "A new Compression technique using an artificial neural network".
[9] Rafael C. Gonazalez, Richard E.Woods, "Digital Image Processing", 2nd Ed., PHI, 2005.