# Supervised Learning Based Image Coding Technique using Artificial Neural Networks

S.R. Patil*, T. R. Sontakke**

*Abstract:*

*In this paper, an image compression technique using a feed forward neural network is employed. The neural network has three layers: an input layer, a hidden layer and an output layer. The inputs of the neural network are original image data, while the outputs are reconstructed image data, which are similar to the inputs. If the amount of data required to store the hidden unit values and the connection weights to the output layer of the neural network is less than the original data, compression is achieved. The network is trained by using different learning rate parameters and momentum factor. It is noticed that the selection between learning rate parameters and the momentum factor is important for the convergence and accuracy of the results.*

**KEYWORDS**: Neural network, Image compression and coding, Back propagation algorithm.

* Computer Engineering Department, Bharati Vidyapeeth's College of Engineering for Woman, Katraj, Pune, Email: srpatil44@gmail.com;

** Siddhant College of Engineering, Sudumbre, Pune.

## 1. Introduction

There has been a considerable amount of research on the area of data compression during the last twenty years [9]. Traditional data compression techniques have included transform-oriented methods and vector quantization. Transform-based compression techniques project the data onto a domain, which requires fewer coefficients for good data representation. The performance of transform-based data compression system (e.g., JPEG and MPEG) is highly dependent on the type of transform being used (e.g., cosine transform, wavelet transform) [11]. Vector and adaptive-vector quantization techniques require the development of an appropriate codebook to compress the data. The performance of the vector quantizer compression systems is highly dependent on the quality of the codebook, and the operational times can be long if the codebook is large [10]. There have been more recent developments in data compression based on studies on the brain mechanism for vision and scene analysis [1]. These techniques have achieved a very large compression ratio (of the order of 100:1), but can require extensive processing to extract the relevant features.

Neural networks offer the potential for providing a novel solution to the problem of data compression by their ability to generate an internal data representation [2], [11].

## 2. Supervised learning using feedforward network

Back-propagation is one of the neural networks training algorithm that can be applied directly to image compression coding [3] [8]. Back propagation algorithm is a supervised learning algorithm where a definite target is available. The network learns towards the target from the input layer to the output layer [2]. Some inherent features of back propagation network image data compression schemes are: (a) the network structure is massively parallel, (b) the network is adaptive, (c) the network determines the compressed features of the original image during the training stage, and (d) the intrinsic generalization property of the structure enables it to effectively process images outside the training set.

The neural network structure can be illustrated as in Fig.1. It has three layers viz., an input layer, an output layer and a hidden layer. The input layer and the output layer are fully connected to the hidden layer. Compression is achieved by designing the value of K; the number of neurons at the hidden layer is less than that of the neurons at both the input and the output layers. The input image is split up into blocks or vectors of 8×8, 4×4 or 16×16. When the input vector is referred to as N-dimensional which is equal to the number of pixels included in each block, all the coupling weights connected to each neuron at the hidden layer can be represented by$\{w_{ji}$ where j=1,2,.....N$\}$,which cab also be described by matrix of

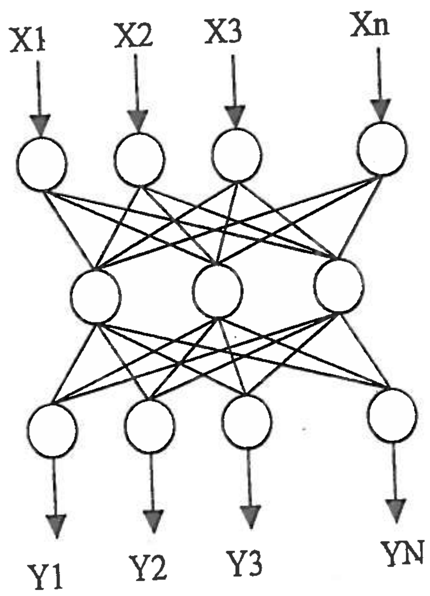


Fig.1 Back-propagation feed forward neural network

by the order K×N. From the hidden layer to the output layer the connection can be represented by $(w'_{ij},\ 1 \le I \le N,\ 1 \le j \le K\}$which is a weight matrix of the order N×K. Image compression is achieved by training the network in such a way that the coupling weights, $\{w_{ji}\}$, scale the input vector of N-dimensions into a narrow channel of K-dimensions (K<N) at the hidden layer and produce the output value which makes the minimizes the quadratic error between input and output.

The neural network shown in Fig.1 could be either linear or nonlinear according to the transfer function employed in the layers. The log-sigmoid function, which is given in equation 1, is one of the most common functions employed in different neural networks.

$$F(x) = \frac{1}{1+\exp(-x)} \quad (1)$$

*Some networks employ a log sigmoid function in the hidden layer and a linear function in the output layer. It has been shown that nonlinear functions have a better capability of learning both linear and nonlinear problems compared to linear ones.*

*The output $Z_j$ of the $j^{th}$ neuron in the hidden layer is given by*

$$z_j = f^1\left(\sum_{i=1}^{n} w_{ji}x_i + b_j\right) \qquad (2)$$

and the output $Y_k$ of the neuron in the output layer is given by

$$Y_k = f^2\left(\sum w'_{ij}\, z_j + b_k\right) \qquad (3)$$

Where $f^1, f^2$ are the activation functions of the hidden layer and the output layer respectively, $w_{ji}$ is the synaptic weight connecting the $i^{th}$ input node to the $j^{th}$ neuron of the hidden layer, bj is the bias of the $j^{th}$ neuron in the hidden layer. $f^i$ is the activation function, and $Z_j$ is the output in the hidden layer. Similarly, (3) describes the subsequent layer where $Y_k$ is the output of the $k^{th}$ neuron in the output layer.

## 3. Network training

Training the network is an important step to get the optimal values of weights and biases after being initialized randomly or with certain values. During training, the weights and biases of the network are iteratively adjusted to minimize the network performance function, which is calculated as the average squared error between the inputs and targets [7].

In the basic back propagation training algorithm, the weights are moved in the direction of the negative gradient, which is the direction in which the performance function decreases most rapidly [5][6].

$$X_{k+1} = X_k - \eta_k g_k \qquad (4)$$

Where $X_{k+1}$ is a vector of current weights and biases, $g_k$ is current gradient and $\eta_k$ is the learning rate.

## 4. Procedure

In this image compression method a three layers neural network is used viz., input layer, hidden layer and output layer. If image is divided into 8×8 blocks then the input and output layers have 64 neurons and the hidden layer has 16 neurons. If image is divided into 4×4 blocks then input and output layer have 16 neurons and hidden layer has 8 neurons. A back propagation is then employed for the training processes. For this the training input prototypes and target values are introduced to the network so that the suitable behavior of the network can be learned. The idea behind supplying the target values is that this will enable us to calculate the difference between the output and target values and then recognize the performance function, which is the criterion for the training. For training the network, the 256×256 pixels Lena image has been employed.

### 4.1 Pre-processing

For the original image to be used it has to be divided into 8×8 pixels blocks and then each block should be reshaped into a column vector of 64×1 elements. Moreover, for scaling purposes, each pixel value should be divided by 255 to obtain a value between 0 and 1.

### 4.2 Training algorithm

1] The original image was divided into 8×8 pixels blocks and reshaped into 64×1 column vector.

2] The column vector is arranged into a matrix of 64×1024.

3] The target matrix is made equal to the matrix in step 2.

4] Suitable learning parameters and momentum factors are chosen to start the training.

5] The network is simulated with the input matrix and the target matrix.

6] The output matrix of the output layer is obtained.

7] The output matrix is processed to obtain the reconstructed image.

The compression ratio depends upon the neural network structure. The image to be compressed consists of a 2-dimensional N*N array of pixels. Hypothetically, one network trained by the back propagation algorithm consisting of $N^2$ nodes at the input and output layers and $N_h$ nodes at the hidden layer is sufficient to achieve data compression [10]. Compression ratio (CR) for this network is given by

$$CR=N^2/N_h \quad (5)$$

To reduce training time, the image can be partitioned into p*p pixel patches, where p<N. The pixel patches are subsequently used as training patterns for the training set. The resulting network then consists of $p^2$ nodes at input and output layers instead of $N^2$ nodes. The resulting CR is given by

$$CR=\frac{P^2}{N^h} \quad (6)$$

To measure the quality of the reconstructed image, peak signal to noise ratio (PSNR) is calculated. The PSNR is defined as

$$PSNR=10*\log\left(\frac{255}{MSE}\right) \quad (7)$$

Where

$$MSE=\left(\frac{1}{MN\sum\sum X_i-Y_i}\right) \quad (8)$$

Xi is input image, and Yi is the reconstructed image.

## 5. Result

The algorithms have been tested for different learning rate parameters and momentum factors and the results have been observed. It is observed that for small learning rate parameter (η) and large momentum (α) factor the quality of the reconstructed image is better. Experiments were performed for various values of momentum (α) and learning rate parameter (η). Table 1 shows the PSNR and required time for momentum (α) =0.9 and for different learning rate parameter (η)

Table 1:-PSNR for BMP and GIF image for (α)=0.9

| srno | Eta() | Psnr (bmp) | Psnr (gif) |
|---|---|---|---|
| 1 | 0.01 | 29.3440 | 29.3332 |
| 2 | 0.02 | 27.3873 | 27.5393 |
| 3 | 0.03 | 26.4366 | 27.0066 |
| 4 | 0.04 | 26.1335 | 26.5569 |
| 5 | 0.05 | 25.4849 | 26.6455 |
| 6 | 0.06 | 25.2756 | 26.5114 |
| 7 | 0.07 | 25.1515 | 26.3693 |
| 8 | 0.08 | 25.0088 | 26.3785 |
| 9 | 0.09 | 24.2824 | 26.6922 |

Fig 1. shows the PSNR for a gif and a bmp image. It is seen that the PSNR value is different for different types of images.

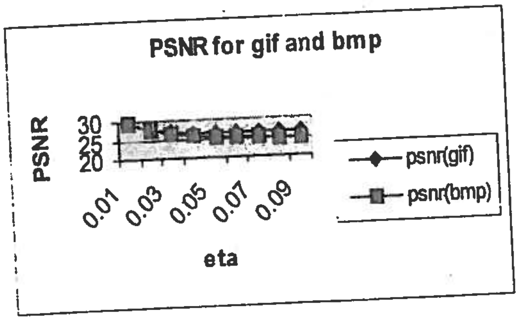


**Fig 1. Graph for gif and Bmp image**

The reconstructed images for Lena.gif and Lena.bmp are shown in fig 3 and fig.4, respectively, where the image is divided into 8x8 windows. Fig 5 shows the image divided into 4x4 windows.

Fig2. Lena.gif

Fig 3. Reconstructed Image (w=8x8) PSNR (29.33)

Fig 4 Fig 5.

| Fig 4 | Fig 5 |
|---|---|
| Reconstructed | Reconstructed |
| Image (w=8x8) | Image (w=4x4) |
| Lena.bmp | Lena.bmp |
| PSNR (29.3440) | PSNR (29.7043) |

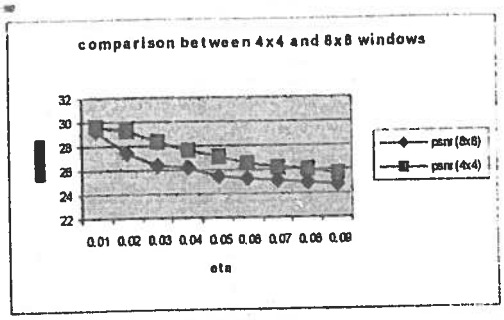


**Fig. 6 PSNR for window size is 44 and 88.**

Fig. 6 shows that dividing the image into small windows gives a higher PSNR value.

## 6. Conclusion

This paper describes the construction of a single hidden layer feed forward neural network. A supervised learning based on back-propagation training algorithm and its application to image compression is also explained. The compressed image is stored in the form of activation values of the hidden units. With the small number of hidden units in the network, a high compression ratio is achieved. It is seen that the smaller the eta ($\eta$) values the higher is the PSNR value.

## References


[1]. Erol Gelenbe Mert Sungur , "Random Network Learning and Image Compression",IEEE transactions on Neural Network, vol. 5, No.2, 1996, pp-3996-3999.

[2]. A Namphol, S.Chin, M.Arozullah, "Image Compression with hierarchical neural network", IEEE Transactions on Aerospace Electronic systems 32 (1) (Jan.!996) 326-337

[3] B.Yegnarayana, "Artificial Neural Networks", Prentice Hall of India Ltd., 2002.

[4]. Laurense Fausett, "Fundamentals of Neural Networks Architectures, Algorithms and Applications", second edition, Pearson Education Publication, Delhi, 2004.

[5]. Limin Fu, "Neural Networks in Computer Intelligence", Tata Mc Graw Hill pulication pvt. ltd., 2003.

[6]. James A. Anderson, "An Introduction to Neural Networks", Prentice Hall India Pvt. Ltd. New Delhi, 2005.

[7]. N.K.Bose, P. Liang, "Neural 7etworks Fundamentals with Graphics Algorithms and Applications", Tata Mc Graw Hill publication pvt. ltd., New Delhi, 2002.

[8]. J. M. Zurada, "Introduction to Artificial Neural System", Jaico Publishing House, 2005.

[9]. R.C. Gonzalez, R.E. Woods," Digital Image Processing ", second edition, Pearson Education Inc., Publication , 2003.

[10]. B. Chanda, D.Majumdar, "Digital Image Processing and Analysis", Prentice Hall India Pvt. Ltd, 2000.

[11]. Jose M. Barbalho, Adurte D . Neto, "Hierarchical SOM Applied to Image Compression", IEEE transactions on Image Processing, vol.5, No7, February 2001, pp442-447.