International Journal of scientific research and management (IJSRM)
||Volume||4||Issue||07||Pages||4429-4435||2016||
Website: www.ijsrm.in ISSN (e): 2321-3418

A Novel Neural Network Approach for Image Compression & Decompression

***Poonam*[1], *Indu*[2]**
[1,2]Department of Computer Science Gateway Institute of Engineering & Technology (GIET), Deenbandhu Chhotu Ram University of Science & Technology (DCRUST), Sonepat
[1]poonamchauhan2531@gmail.com
[2]indu.khatri111@gmail.com

Abstract— An image consists of large data and requires more space in the memory. The large data results in more transmission time from transmitter to receiver. The time consumption can be reduced by using data compression techniques. In this technique, it is possible to eliminate the redundant data contained in an image. The compressed image requires less memory space and less time to transmit in the form of information from transmitter to receiver. Artificial neural network with feed forward back propagation technique can be used for image compression. In this paper, the Bipolar Coding Technique and Levenberg-Marquardt (LM) algorithms are proposed and implemented for image compression and obtained the better results as compared to Principal Component Analysis (PCA) technique. It is observed that the Bipolar Coding and LM algorithm suits the best for image compression and processing applications

.***Keywords*—** Artificial Neural Network, PCA, Bipolar Coding, Levenberg-Marquardt

Introduction

Image compression refers to the task of reducing the amount of data required to store or transmit an image. At the system input, the image is encoded into its compressed form by the image coder. The compressed image may then be subjected to further digital processing, such as error control coding, encryption or multiplexing with other data sources, before being used to modulate the analog signal that is actually transmitted through the channel or stored in a storage medium. At the system output, the image is processed step by the step to undo each of the operations that were performed on it at the system input. At the final step, the image is decoded into its original uncompressed form by the image decoder. If the reconstructed image is identical to the original image the compression is said to be lossless, otherwise, it is lossy.

Image compression addresses the problem of reducing the amount of information required to represent a digital image. It is a process intended to yield a compact representation of an image, thereby reducing the image storage transmission requirements. Every image will have redundant data. Redundancy means the duplication of data in the image. Either it may be repeating pixel across the image or pattern, which is repeated more frequently in the image. The image compression occurs by taking benefit of redundant information of in the image. Reduction of redundancy provides helps to achieve a saving of storage space of an image. Image compression is achieved when one or more of these redundancies are reduced or eliminated. In image compression, three basic data redundancies can be identified and exploited. Compression is achieved by the removal of one or more of the three basic data redundancies [1].

- Inter Pixel Redundancy
- Coding Redundancy
- Psycho Visual Redundancy

In this paper, the Bipolar Coding Technique and Levenberg-Marquardt (LM) algorithms are proposed and implemented for image compression and obtained the better results as compared to Principal Component Analysis (PCA) technique. It is observed that the Bipolar Coding and LM algorithm suits the best for image compression and processing applications.

Artificial Neural Network

The artificial neural network is a recent tool in image compression as it processes the data in parallel and hence requires less time and therefore, it is superior over any other technique. An Artificial Neural Network (ANN) is an information-processing theory that is inspired which is the way of biological nervous systems, just like as the brain process information. The key element of this concept is the fresh structure of the information processing system. The artificial neural network is composed of a large number of greatly interconnected of processing elements (are called neurons) working in union to solve particular problems. An ANN is configured for a particular application, such as through a learning process, pattern recognition or data classification, Learning in biological systems including the adjustments of the synaptic connections that exist among the neurons. The multiple layers of simple processing elements are known as neurons. Each neuron is linked to sure of its neighbors with changing coefficients of connectivity that represent the strengths of these connections.

Basic model for image compression using neural network is shown in figure 1 below.

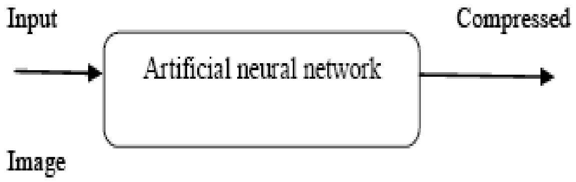

Figure 1: Basic image compression

A neural network architecture as shown in Figure 2 is used for solving the image compression problem. In this type of architecture, input from the large number of input neurons is fed to a less number of neurons in hidden layer, which is further fed to large number of neurons in output layer of the network. Such type of network is referred as bottleneck feed forward neural network. One of the most important types of feed forward network is the multilayer back propagation neural network [2, 3].

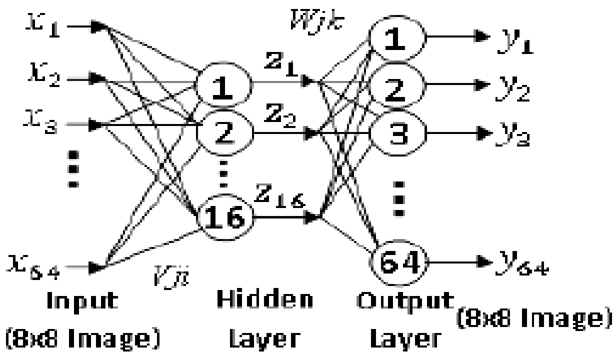

Figure 2: Feed forward neural network.

The neural network architecture for image compression is having 64 input neurons, 16 hidden neurons and 64 output neurons according to the requirements. The input layer encodes the inputs of neurons and transmits the encoded information to the hidden layer neurons. The output layer receives the hidden layer information and decodes the information at the output. The outputs of hidden layer are real valued and require large number of bits to transmit the data. The transmitter encodes and then transmits the output of the hidden layer 16 values as com- pared to the 64 values of the original image. The receiver receives and decodes the 16 hidden neurons output and generates the 64 outputs at the output layer.

The techniques like Principal Component Analysis (PCA) and proposed Bipolar Coding with Linear transformations are required to incorporate with neural network to manage the input and output image data. PCA technique transforms an image data into small valued data which can be easily handled by neural network.

Principal Component Analysis

The images obtained from the satellite are noisy and require large memory space. To compress such type of images, PCA based neural network model is used. However, adaptive neural network with PCA can also be used but it compresses the image in very less percentage [4, 5]. Principal Component Analysis technique is called as Karhaunen Loeve transform, Hotelling transform or proper orthogonal transformation. It is a factorization technique which is generally used in mathematical applications. PCA technique collects the data and transforms it to the new data which results in same statistical properties. The data transformation is performed in such a way that its originality remains at the end of transformation [6].

The image data can be represented by a set of m vectors: $A = a_1, a_2, \ldots a_i, \ldots a_m$ where, i represent n features. The vector a_i is depending on image application. For example, in image compression each vector represents a major component of each vector features like color and size. The features of an image can be combined by considering the feature of each vector. Thus, data set **A** represent the features column vector k as:

$$C_{A,K} = \begin{pmatrix} a_{1,k} \\ a_{2,k} \\ a_{3,k} \\ \\ : \end{pmatrix} \quad : $$
$$a_{n,k}$$

This approach requires for the computation of the input data convergence matrix A and extraction of the eigenvalues and eigenvectors. The feature column vectors can be grouped in a matrix form for easy processing. That is,

$$C_A = [C_{a1}, C_{a2}, ..., C_{an}]$$

where the values of a and k are varying from 1 to n.

For compression of data, the component which has less importance has to be eliminated from the matrix. Therefore, the less important data is replaced by zero value. The new vectors which give the better classification properties can be form the feature column vectors $C_{a,k}$. PCA method confirms that the important data which account maximum variation in the covariance matrix to be considered. The Linear dependence between two random variables can be defined by the covariance measures. Therefore, by computing the covariance, the relationship between two data sets can be established. If, $A_i = (a_{i,1}, a_{i,2}, \ldots a_{i,n})$ then the covariance is defined by equation:

$$\sigma_{A1,n} = E\{(C_{A,1} - \mu_{A,1})\,(C_{A,n} - \mu_{A,n})\}$$

E{.} is the average value of the elements of the vector and $\mu_{A,n}$ is the column vector obtained by multiplying the scalar value $E\{C_{A,k}\}$ by unitary vector. It is important to note that the covariance measures a linear relationship between the two values of sets. PCA method gives the simple solution in many applications like linear modeling, data compression, image compression, image restoration and pattern recognition etc.

The image data can also be compressed using the concept of data transformation using PCA. The PCA algorithm for data transformation and data compression is summarized in the following steps [7]:

Step 1: Obtain the feature column vector matrix from the given image data. AC
Step 2: Obtain the covariance matrix ΣA.
Step 3: Using characteristic equation $\lambda_i I - \Sigma A = 0$, Obtain the eigenvalues. These eigenvalues forms the matrix Σy.
Step 4: Also, calculate the eigenvectors matrix by W considering the eigenvalues λ_i.
Step5: Obtain the Transformation W^T by considering the eigenvectors as their columns.
Step 6: Obtain the features vector matrix by $C_y = C_A W^T$
Step 7: For compression of an image, the dimensionality of the new feature vector is reduce by setting small eigenvalues λ_i to zeros.

Proposed Work

Bipolar Coding technique

The Bipolar Coding technique is a new approach in the field of image processing. The Bipolar Coding transformation is based on the bipolar activation function. We have proposed and applied this technique along with neural network for image compression. The sets of data obtained from an image are in analog form and required to convert into digital form. This conversion is possible using bipolar coding with linear scaling. The scaling has the advantage of mapping the desired range of maximum and minimum analog value of the data set. The conversions are based on certain ranges where analog form is scaled between value 0 and 1. Thus, for converting analog values into digital form, different binary values can be assigned. In this technique, each value is converted into the range between 0 and 1 using the formula as follows [8]:

$$\delta = X_{max} - X_{min}$$
$$Y = \text{Intercept } C = (X - X_{min}) / \delta$$

These converted values are used as the input to the neural network. Even though the neural network architecture takes input from 64 nodes down to 16 nodes, no actual compression has been occurred because unlike the 64 original inputs which are 8-bit pixel values, the outputs of the hidden layer are also real-valued, which requires possibly an infinite number of bits for transmission. True image compression occurs when the hidden layer outputs are digitized before transmission. The **Figure 3** shows a typical digitization scheme using 3 bits as reported in [9]. In this scheme, there are 8 possible binary codes: 000, 001, 010, 011, 100, 101, 110, and 111.

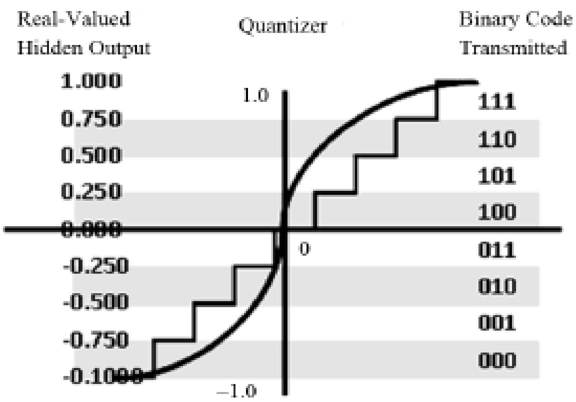

Figure 3: Digitization of hidden unit output.

Each of these codes represents a range of values for a hidden unit output. For example, consider the first hidden output, when the value is between –0.1 and –0.75, then the code 000 is transmitted and when the value is between 0.5 and 0.75, the code 110 is transmitted. The concept of residual blocks conversion also used by G. Qiu et al. [10] but the PSNR observed was very less. The proposed Bipolar Coding technique using feed forward back propagation neural network is summarized in the following steps:

Step 1: Divide the image into small 8 × 8 chucks. These chucks are in square matrix form. It is easy to perform operation on such symmetric matrix.

Step 2: Obtain the values of pixels (0 to 255) of matrix and convert these values in the bipolar range –1 to 1. This is called as pixel to real number mapping.

Step 3: Now apply these values of bipolar range to the feed forward back propagation neural network. This neural network must have 64 input neurons as the chucks are of the size 8 × 8. Train the neural network using training algorithm as explained in Section 4.

Step 4: The bipolar values with weight and biases feed to the hidden layer which may have 2, 4, 8, 16, 32, and 64 hidden neurons. Convert these values in digital bits as explain earlier.

Step 5: Select the 8 × 8 chucks in a sequence after the completion of training of the neural network.

Step 6: Digital bits are now converted to real values.

Step 7: Matrix ranges bipolar values from –1 to 1 reconverted from real value to (0 to 255) pixel mapping.

Step 8: Finally, recall phase demonstrate the decompressed image (output image) at the output of the output layer of neural network. Pixel to real value and real value to pixel conversion is done during the compression and decompression process.

The proposed Bipolar Coding technique using feed forward back propagation neural network converts decimal values into its equivalent binary code and reconvert in decompression phase. The compression and quality of image depends on number of neurons in the hidden layer. The quality of output image improves and the data loss reduces as the number of hidden neurons increases.

Levenberg-Marquardt Algorithm

Neural networks are suitable for linear as well as highly nonlinear functions with the adjustable biases and weights. The Levenberg-Marquardt (LM) algorithm can be used for nonlinear functions. It is very simple, but robust method for approximating the nonlinear function that locates the minimum of a multivariate function which is expressed as the sum of squares of non-linear real valued function [11, 12].

LM is a combination of two methods, steepest descent and the Gauss-Newton method. The steepest decent is a first order optimization method to find the local minimum of a function. The algorithm behaves like a steepest descent method when the current solution is far from the correct one. The steepest descent method is widely popular among researchers due to its easy concept and relatively less work steps. Consider the function F(x) which can be defined and differentiable within a given boundary. The negative gradient of F(x) represents the directions of fastest decrease of the function. To obtain the local minimum of F(x), the Steepest Descent is employed, where it uses a zigzag path from an arbitrary point0Xand gradually slide down the gradient, until it reaches to the actual point of minimum. Mathematically, this step can be defined by iterative form as,

$$w_{n+1} = w_n - \lambda_n \Delta F(w_n) = w_n - \lambda_n g(w_n)$$

Here, the term $g(w_n)$ is the gradient at a given point and λ_n should be used as a step in the gradient direction. The Gauss-Newton algorithm is a method that solves the nonlinear least square problems. If m functions are given, $r_1, r_2, \ldots, r_m$ of n variables $x_1, x_2, \ldots x_n$ with $m >= n$, the Gauss–Newton method finds the minimum of the sum of squares.

If, $x^{(0)}$ is the initial Gauss for the minimum, then the method executes by iterations, $x^{(sum+1)} = x^{sum} + \Delta$. In this, is the solution of the normal equations, $(J^TJ)\,\Delta = J^Tr$. Where, r is the vector function, J is the Jacobian matrix of r. It is slow but guaranteed correct solution. The LM algorithm incorporates the advantages of steepest descent and Gauss-Newton algorithm. We proposed and implemented successfully the LM mathematical technique in the application of image processing. The normal LM equation for image compression application can be modified as,

$$(J^TJ + \lambda I)\,\delta = J^TE$$

Where J is the Jacobian matrix, λ is the Levenberg's damping factor, δ is the weight update vector and E is the error vector. The J^TJ matrix is called as Hessian matrix. The damping factor λ is required to adjust after each iteration and it guides the optimization process. The Jacobian is a matrix contains first derivatives of the net-work errors with respect to the weights and biases. It can be obtained by taking the partial derivatives of output with respect to weight as given below:

$$J = \begin{pmatrix} \frac{\partial F(x_1,w)}{\partial w_1} & \frac{\partial F(x_1,w)}{\partial w_2} & \cdots & \frac{\partial F(x_1,w)}{\partial w_n} \\ \frac{\partial F(x_2,w)}{\partial w_1} & \frac{\partial F(x_2,w)}{\partial w_2} & \cdots & \frac{\partial F(x_2,w)}{\partial w_n} \\ \vdots & \vdots & & \vdots \\ \frac{\partial F(x_N,w)}{\partial w_1} & \frac{\partial F(x_N,w)}{\partial w_2} & \cdots & \frac{\partial F(x_N,w)}{\partial w_n} \end{pmatrix}$$

where $F(x_i, w)$ is the network function evaluated using the weight vector w for the i^{th} input vector. Hessian doesn't need to be calculated as it is approximated by using the Jacobian matrix. This approximation of the Hessian holds good if the residual errors at the solution are small enough. The general Levenberg-Marquardt algorithm consists of the following steps:

Step 1: Obtain the Jacobian matrix J. It is recommended to use finite difference or chain rule for calculation of J.
Step 2: Obtain the error gradient $g = J^TJ$
Step 3: Now, Hessian matrix can be approximated using the cross product of Jacobian as: $H = g = J^TJ$

Step 4: The value of δ can be obtained by solving the equation $(J^TJ + \lambda I)\ \delta = J^TE$.
Step 5: Use the value of δ to update the network weights w.
Step 6: Using the updated weights w, recalculate the sum of squared errors.
Step 7: If sum of the squared errors has not further reduced then discard the new weights values and increase the value of λ.
Step 8: Else decrease λ and calculate new values of w using δ till the error reaches at the desired value and stop.

Here, λ start from very small value such as 0.1 and if it becomes large then iterations stops and the value of λ again decreases.

In this algorithm, if the damping used at one iteration reduces the error, the damping is divided by a reduction factor before the next iteration and the convergence speed up towards the best result. If the error increases, then the damping multiplied by the factor, ensure the correct result. Thus, the algorithm switches from one to another smoothly. If the proper value of damping selected in LM, then its iterations completed within the short duration and produced the best results at decompressed phase. The LM algorithm thus required the less number of iteration and less time as compared to PCA technique as well as the Bipolar Coding technique. The memory requirement in LM method is also less as the number of iteration reduces.

Conclusion

In this work, PCA, proposed Bipolar Coding and LM technique, based on artificial neural network are applied for image compression application. In PCA technique, the accuracy of the results obtained depends upon the threshold value of eigenvalue at which the iteration process of learning is terminated. The Bipolar Coding Technique and LM algorithm are proposed and implemented for image compression and got the satisfactory results as compared to PCA technique. The Bipolar coding network is trained with the small 8 × 8 blocks of image and tested. It is observed from the results that using this method, a good quality of decompressed image is obtained. It has high PSNR and very less error. Thus, this method achieves high compression. But, the neural network is trying to determine the updated weights and biases in each step to minimize the systems errors. This is step by step process that requires more time and more memory space to store the subsequent results. To overcome these problems and to improve the results, the mathematical Levenberg-Marquardt algorithm is proposed. This algorithm is fast in operation as well as it requires less memory to store the results.

References

[1] V.K Padmaja and Dr. B. Chandrasekhar, “Literature Review of Image Compression Algorithm”, IJSER, Volume 3, pp. 1-6, 2012.

[2] M. Daszykowski, B. Walczak and D. L. Massart, “A Journey into Low-Dimensional Spaces with Auto associative Neural Networks,” Talanta, Vol. 59, No. 6, 2003, pp. 1095- 1105.

[3] V. Gaidhane, V. Singh and M. Kumar, “Image Compression Using PCA and Improved Technique with MLP Neu- ral Network,” Proceedings of IEEE International Conference on Advances in Recent Technologies in Communication and Computing, Kottayam, 16-17 October 2010, pp. 106-110.

[4] R. C. Gonzalez, R. E. Woods and S. L. Eddins, “Digital Image Processing Using MATLAB,” Pearson Edition, Dorling Kindersley, London, 2003.

[5] J. X. Mi and D. S. Huang, “Image Compression Using Principal Component Analysis Neural Network,” 8th IEEE International Conference on Control, Automation, Robotics and Vision, Kunming, 6-9 December 2004, pp. 698-701.

[6] S. T. Bow, B. T. Bow, “Pattern Recognition and Image Processing,” Revised and Expanded, 2nd Edition, CRC Press, Boca Raton, 2002.

[7] M. Nixon and A. Aguado, “Feature Extraction & Image Processing,” 2nd Edition, Academic Press, Cambridge, 2008, pp. 385-398.

[8] S. N. Sivanandam, S. Sumathi and S. N. Deepa, “Introduction to Neural Network Using MATLAB

6.0," 2nd Edition, Tata Mc-Graw Hill Publication, Boston, 2008.

[9] A. Laha, N. R. Pal and B. Chanda, "Design of Vector Quantizer for Image Compression Using Self Organizing Feature Map and Surface Fitting," IEEE Transactions on Image Processing, Vol. 13, No. 10, October 2004, pp. 1291-1303. doi:10.1109/TIP.2004.833107

[10] G. Qiu, T. J. Terrell and M. R. Varley, "Improved Image Compression Using Back Propagation Networks," In: P. J. G. Lisbao and M. J. Taylor, Eds., Proceeding of the Workshop on Neural Network Applications and Tools, IEEE Computer Society Press, Washington DC, 1994, pp. 73-81.

[11] V. Singh, I. Gupta and H. O. Gupta, "ANN Based Estimator for Distillation Using Levenberg-Marquardt Approach," Engineering Applications of Artificial Intelligence, Vol. 20, No. 2, 2007, pp. 249-259. doi:10.1016/j.engappai.2006.06.017

[12] M. I. A. Lourakis, "A Brief Description of the Levenberg- Marquardt Algorithm Implemented by Levmar", Foundation of Research & Technology, Vol. 4, 2005, pp. 1-6.