BIOINFO Computational Mathematics
Volume 1, Issue 1, 2011, pp-05-10
Available online at: http://www.bioinfo.in/contents.php?id=321

Artificial Intelligence Used For Image Compression

R. A.Vasmatkar[1] , Shilpa P. Biradar[2], Shivashankar P. B.[3]

[1]Pune University, Pune, ranipbs@gmail.com,

[2]Vishveshvaraya Technological Univaersity, Belagaum, vanishil90@gmail.com,

[3]Vishveshvaraya Technological Univaersity, Belagaum, shivdurga.biradar@gmail.com

Abstract :

Picture compression algorithms, using a parallel structure of neural networks and the concept of DWT, has been described. Although these algorithms are intrinsically robust, and may therefore be used in high noise environments, they suffer from several drawbacks: high computational complexity, moderate reconstructed picture qualities, and a variable bit-rate.

In this paper, we describe a simple parallel structure in which all three drawbacks are eliminated: the computational complexity is low, the quality of the decompressed picture is high, and the bit-rate is fixed.

Key words : ANN-Artificial Neural Network, NN-Neural Network, DWT- Discrete Wavelet Transform, IDWT-Inverse Discrete Wavelet Transform

I. INTRODUCTION

Image compression is playing key role in the development of various multimedia computer services and telecommunication applications. As image needs a huge amount of data to store it, there is pressing need to limit image data volume for fast transport along communication links. The goal of image compression techniques is to remove redundancy present in data in away that enables image reconstruction. Statistical properties of the image are used in design an appropriate Compression technique. The strong correlation between image data items enables reduction in the data contents without significant quality degradation.

There are numerous lossy and lossless image compression techniques. Considering X-ray images or ECG data where each bit of information is essential a lossless compression must be used. On the other hand, for the still digital image or video, which have been already lossy digitalized, a lossy compression is preferred. There are numerous lossy compression techniques. Very effective techniques are transform based techniques, particularly cosine transform based technique that showed excellent results. Contrary traditional techniques for image compression, it is worth to discuss some of the recent techniques that may be used for data compression.

Artificial Neural Networks (ANN) has been used for solving many problems, special in cases where the results are very difficult to achieve by traditional analytical methods. It is important to emphasize, although there is no sign that neural networks can take over the existing techniques, research on neural networks for image compression are still making some advances. Possibly in the future this could have a great impact to the development of new technologies and algorithms in this area. The main goal is to investigate which neural network structure accomplished with different learning algorithms give the best results compared with standard transform coding techniques. The various feed forward neural networks with back propagation algorithms were directly applied to image compression coding. A new Technology known as Hybrid technology, which combines the concept of both DWT & NN, so as to overcome the shortcomings of DWT and to increase the compression ratio. The block diagram is as below in Fig.1

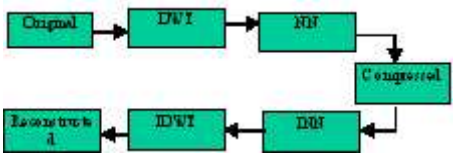

Fig. 1: Hybrid Technique

II. STEPS FOR IMAGE COMPRESSION & DECOMPRESSION

Many steps must be taken before an image can be successfully compressed using either conventional or intelligent methods. The steps proposed for image compression are as follows:

1.Image Acquisition, 2.Preprocessing, 3.Preparation of training pairs, 4.Exclusion of similar training pairs, 5.Compression of the image using an ANN trained by back propagation (BP), 6.Reconstruction of the image.

Step1. Image acquisition

The images are scanned using a Macintosh Flatbed scanner and a Macintosh Personal Computer running the OFOTO software package. The images are saved in Tagged Image Format (TIF). The pixel values of the image intensities are represented using binary values.

Step2. Segmentation of the image and Preprocessing

The image is broken into sub images as per the requirement. The 2-D images are converted to 1-D image so that it can be fed to the neural network.

Step3. Preparation of training pair

Multiple images have to be identified based on image properties. This is a very important step as this helps in generalizing the network for compression of any image data. Standard test images are to be identified and preprocessed as explained in Step 2 for training.

Step4. Exclusion of similar training pairs

As many of the images tested were extremely large, there were many similar training pairs after segmentation. To eliminate this redundant information, a program was used to search the training file for similar training pairs (i.e. similar binary vectors) and delete them. This not only reduced the number of redundant training pairs, but reduced training time of the ANN.

Step5. Compression of the image using an ANN trained by BP

Once the training is achieved, using BP technique, the trained MLN is split into the transmitter and receiver section. Unknown image data is fed into the network for compression. The compressed data is corrupted with noise, and fed to the receiver part for decompression.

Actually, even though the bottleneck takes us from 64 nodes down to 16 nodes, no real compression has occurred because unlike the 64 original inputs which are 8-bit pixel values, the outputs of the hidden layer are real-valued (between -1 and 1), which requires possibly an infinite number of bits to transmit. True image compression occurs when the hidden layer outputs are quantized before transmission. The Figure below shows a typical quantization scheme using 3 bits to encode

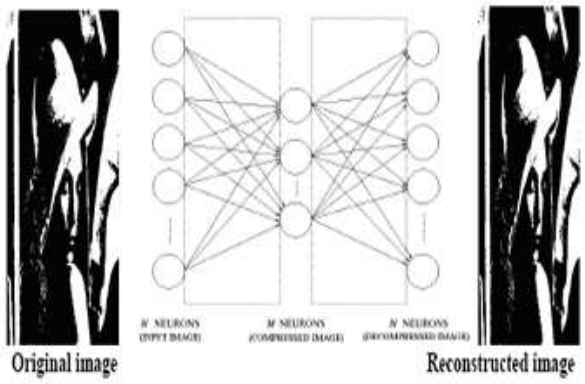

Fig. 2.1: Compression and reconstruction of lena image

each input. In this case, there are 8 possible binary codes which may be formed: 000, 001, 010, 011, 100, 101, 110, and 111. Each of these codes represents a range of values for a hidden unit output. For example, consider the first hidden output. When the value of is between -1.0 and -0.75, then the code 000 is transmitted; when is between 0.25 and 0.5, then 101 is transmitted. To compute the amount of image compression (measured in bits-per-pixel) for this level of quantization, we

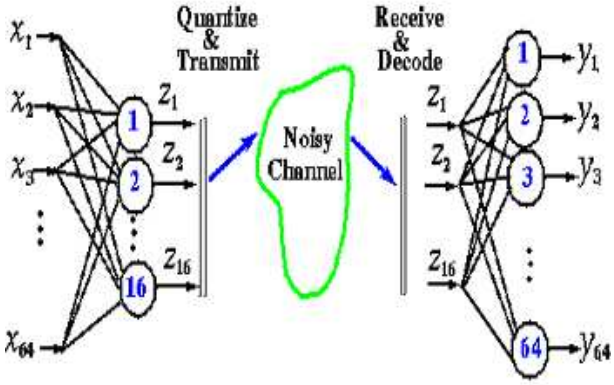

Fig2.2: Image compression and decompression.

compute the ratio of the total number of bits transmitted: to the total number of pixels in the original image: 64; so in this case, the compression rate is given as bits/pixel. Using 8 bit quantization of the hidden layer.

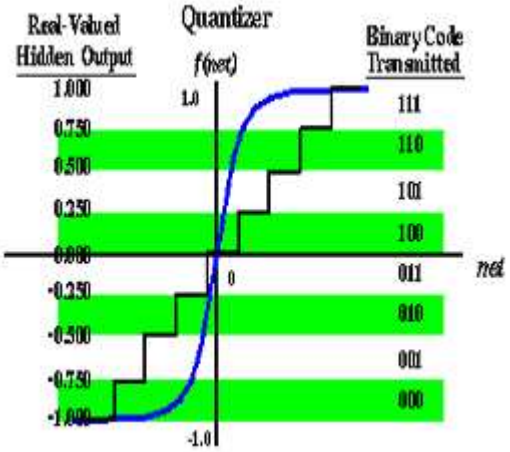

Fig 2.3: Quantization of input value.

Step7. Implementation of a program for the reconstruction of the image

The compressed data with noise is fed into the receiver network for decompression. The decompressed data is compared with the original image. The performance matrices are estimated based on compression ratios.

III. IMAGE COMPRESSION

The transport of images across communication paths is an expensive process. Image compression provides an option for reducing the number of bits in transmission. This in turn helps increase the volume of data transferred in a space of time, along with reducing the cost required. It has become increasingly important to most computer networks, as the volume of data traffic has begun to exceed their capacity for transmission. Traditional techniques that have already been identified for data compression include: Predictive coding, Transform coding and Vector-Quantization.

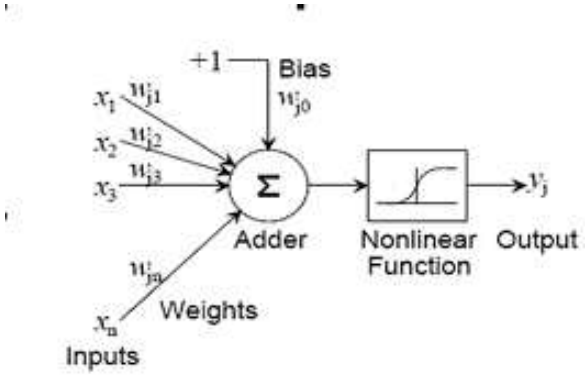

Fig 3.1: Model of neuron

The idea is that we should be able to train this perceptron to respond to certain inputs with certain desired outputs. After the training period, it should be able to give reasonable outputs for any kind of input. If it wasn't trained for that input, then it should try to find the best possible output depending on how it was trained.

Neural network architecture

The manner in which the neurons of a neuron network are structured is intimately linked with the learning algorithm used to train the network. Network structure can be broadly divided into three classes of network architectures.

Single-Layer Feed-forward Networks

In a layered neural network the neurons are organized in the form of layers. The simplest form of a layered network consists of an input layer of source nodes sthat project onto an output layer of neurons, but not vice-versa. In other words, this network is strictly a feedforward or acyclic type. Such a network is called a single-layer network, with the designation "single layer" referring to the output layer of computational nodes. The input layer of source nodes are not counted as no computation is performed here.

Multilayer Feed-forward Networks

The second class of a feed-forward neural network distinguishes itself by the presence of one or more hidden layers, whose computation nodes are correspondingly called hidden neurons or hidden units. The function of hidden neurons is to intervene between the external input and the network output in some useful manner. By adding one or more hidden layers, the network is enabled to extract higher-order statistics.

In a rather loose sense the network acquires a global perspective despite its local connections due to the extra set of synaptic connections and extra dimension of neural interactions. The ability of hidden neurons to extract higher-order statistics is particularly valuable when the size of the input is large.

In this architecture the network is trained in stages. There are two stages for compression and decompression respectively. At each stage the original image is scaled

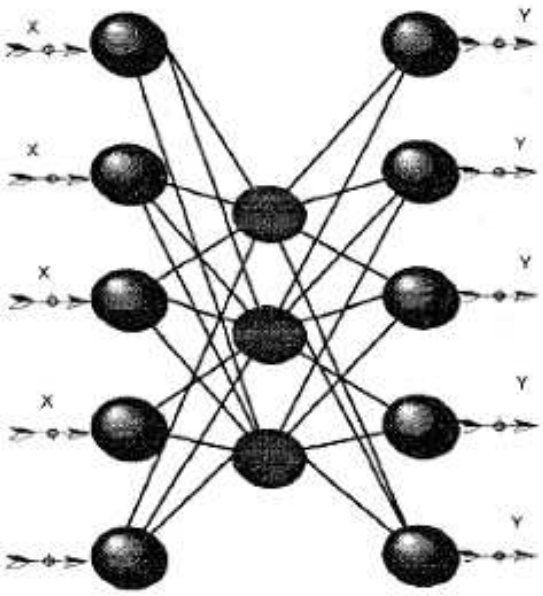

Fig 3.2: Two layer neural network.

down and in the second stage it is further scaled down to get the compressed image.

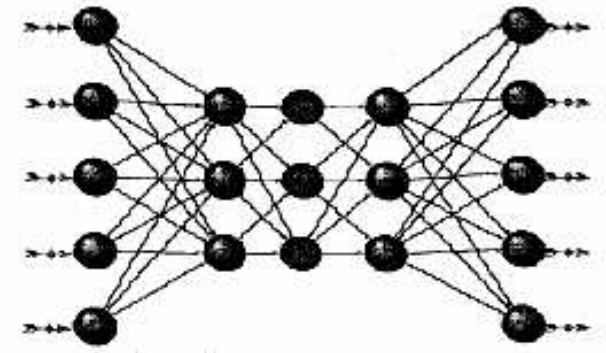

Fig 3.3: Multi layer neural network

That means that each layer can be trained. The block size can also be increased for increased performance. Now this compressed output is passed through the two decompression stages to get the decompressed image with reduced errors when compared to its two layered outer-part.

Training:

After defining the network and its layers, the network has to be trained with various pictures. By doing so the network gets trained and forms a weight matrix of size [8 x 64] and a bias matrix [8 x 1], by the method of back propagation.

Output:

The output thus obtained will be a [8 x 1] matrix. From this, it is evident that

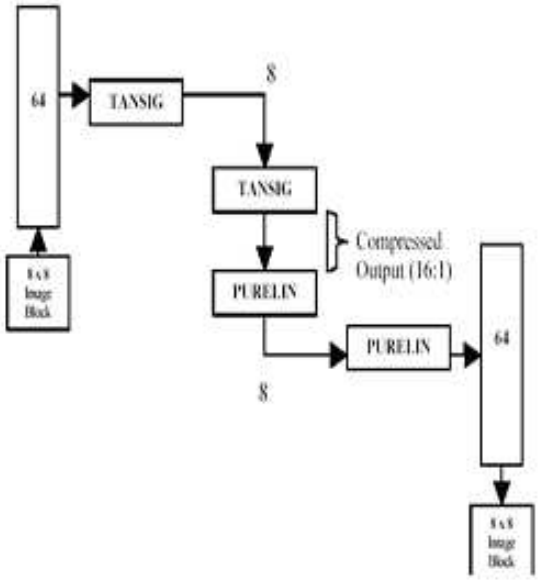

Fig 3.4: Multilayered neural network architecture.

Compression has taken place. 64 inputs fed give an output, which has only 8 values. That means the input is reduced to 12.5% of its original value.

Note: After complete training of network the weights and biases matrix remains constant.

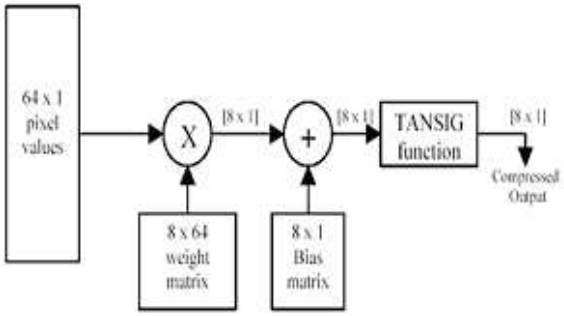

Fig3.5: Compression stage.

- From the above two figures it is evident, how compression and decompression takes place.
- The mean square error can be found by comparing the input and output values.
- Compression ratio CR = (No. of input values) /(No. of compressed values)
- The above steps are repeated for JPEG (using

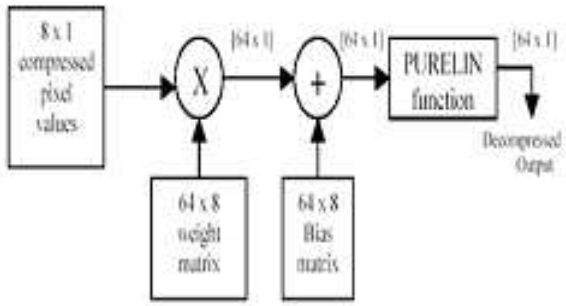

Fig 3.6: Decompression stage

MATLAB) and hence compared.

Note: The above shown technique can achieve 87.5% compression. By expanding the Network, i.e. increasing the number of stages greater compression ratio (93.75%) can be achieved.

IV. TEST AND RESULTS

Result of the decompressed images with training and without training.

For example: input image tree, child and foot ball, decompressed image of tree after training, and decompressed images of child and foot ball without training.

Fig 4.1: Input Image-tree

Fig 4.2(a): Input Image- Child

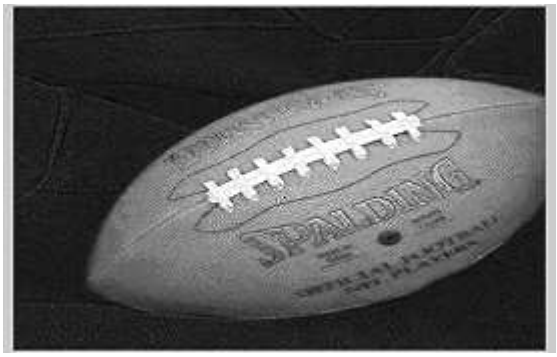

Fig 4.2(b): Input Image – Foot ball

Fig 4.3: Output Reconstructed Image (tested with training)

Fig 4.2(a): Output Reconstructed Image (tested without training)

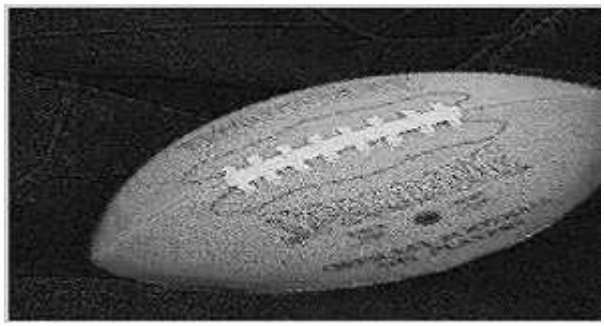

Fig 4.2(b): Output Reconstructed Image (tested without training)

TABLE 4.1.Gives the values of Max Error, MSE, and PSNR of output test images for with training and without training.

TABLE 4.1.LOGSIG-LOGSIG, NEWCF, TRAINRP-4:16

	Image1	Image2	Image3
MAX ERROR	164	50	184
MSE	176.72	41.53	263
PSNR	25.6	31.9	23.9

V. CONCLUSION

Artificial neural networks are inspired by the learning process that takes place in biological systems. They can be "trained" to produce an accurate output for a given input. So by combining both DWT and NN we obtain higher compression ratio and error in retrieving the image also doesn't effect so much as in JPEG.

This paper proves that, the concept of Image compression using ANN and DWT posses the advantages of simple computations, fault tolerance, parallel processing, robust with respect to error transmission in the communication media it has made a break through in supervised learning of layered neural network.

REFERENCES

[1] S. Lecoeuche and D. Tsaptsinos. "Engineering applications of neural networks - novel applications of neural networks in engineering. 2006.

[2] Z. Chen and S. Haykin. " *On different facets of regularization theory. Neural Computation"*, 2002.

[3] Jacek.M.Zurada, Akos Kiss *Introduction to Artificial Neural Systems,* Sixth Jaico Impression, 2003.

[4] S. Haykin, Costas Neocleous, Chritstos Schizas Neural Networks: *A Comprehensive Foundation, Macmillan* College Press, New York, 1994.