ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 2, Issue 3, September 2012

Fast Training Algorithms of Compression Using Neural Network with Wavelets

Prachi Jain, Aishwarya Vishwakarma, Megha Jain

Abstract*— Wavelets have emerged as powerful tools for signal coding especially bio-signal processing. Wavelet transform is used to represent the signal to some other time–frequency representation better suited for detecting and removing redundancies The Neural Networks are good alternative for solving many complex problems. In this paper multi-layer neural network has been employed to achieve image compression a novel algorithm for neural network with different-different techniques is proposed in this paper. Experimental results show that this algorithm outperforms than other coders such as SPIHT EZW STW exits in the literature in terms of simplicity and coding efficiency by successive partition the wavelet coefficients in the space frequency domain and send them using adaptive decimal to binary conversion. Spatial-orientation tree wavelet (STW) and Embedded Zero tree Wavelet (EZW) has been proposed as a method for effective and efficient embedded image coding. This method holds good for au important features like PSNR, MSE, BPP, CR, image size. SPHIT has been successfully used in many applications. (The techniques are compressed by using the performance parameters PSNR, MSE, CR, and BPP.

Keywords: **PSNR, MSE, STW, SPIHT, EZW, Neural Network.**

I. INTRODUCTION

Compression/coding of Electrocardiogram (ECG) signal are done by detecting and removing redundant Information from the ECG signals [1]. ECG data compression algorithm is either of two categories: one method is direct data compression method [2, 3], which detect redundancies by direct analysis of actual signal samples. Another method is transform method [4–8], which first transform the signal to some other time–frequency representations better suited for detecting and removing redundancies. Among transform methods, the wavelet transform method has been shown promise because of their good localization properties in the time and frequency domain. Villasenor et al., have proposed an explanation of why some wavelets are good for compression and others are not [9]. If these details are very small then they can be set to zero without significantly changing the image. The value below which details are considered small enough to be set to zero is known as the threshold. The greater the number of zeros the greater the compression that can be achieved. The amount of information retained by an image after compression and decompression is known as the .energy retained. And this is proportional to the sum of the squares of the pixel values.

Large changes in colour will be less redundant and harder to compress. The data are in the form of graphics, audio, video and image. These types of data have to be compressed during the transmission process. Large amount of data can't be stored if there is low storage capacity present. The compression offers a means to reduce the cost of storage and increase the speed of transmission. Image compression is used to minimize the size in bytes of a graphics file without degrading the quality of the image. There are two types of image compression is present. They are lossy and lossless. Some of the compression algorithms are used in the earlier days [3] and [4] and it was one of the first to be proposed using wavelet methods [2]. Over the past few years, a variety of powerful and sophisticated wavelet based schemes for image compression have been developed and implemented. The coders provide a better quality in the pictures. Wavelet based image compression based on set partitioning in hierarchical trees (SPIHT) [5] and [6] is a powerful, efficient and yet computationally simple image compression algorithm. It provides a better performance when compared to the Embedded Zero tree wavelet [7] transform.

This paper addresses the following problems: (1) Implementing simple algorithm, (2) obtaining high image quality using Peak Signal-to-Noise Ratio (PSNR) and Mean Squared Error (MSE).This is followed by high frequency wavelet coefficients bit stream.

II. IMAGE COMPRESSION

Following the rapid development of information and communication technologies, more and more information has to be processed, stored, and transmitted in high speed over networks. The need for data compression and transmission is increasingly becoming a significant topic in all areas of computing and communications. Computing techniques that would considerably reduce the image size that occupies less space and bandwidth for transmission over networks form an active research. Image compression deals with reducing the amount of data required to represent a digital image [3]. Compression and the amount of distortion in the reconstructed image [4].

III. NEURAL NETWORK

Artificial Neural Networks have been applied to many problems, and have demonstrated their superiority over classical methods when dealing with noisy or incomplete data. One such application is for data compression. Neural networks seem to be well suited to this particular function, as they have an ability to preprocess input patterns to produce simpler patterns with fewer components [5]. Neural networks are computer algorithms inspired by the way information is processed in the nervous system. An important difference between neural networks and other AI techniques is their ability to learn. The network "learns" by adjusting the interconnections (called weights) between layers. When the network is adequately trained, it is able to generalize relevant

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 2, Issue 3, September 2012

output for a set of input data. A valuable property of neural networks is that of generalization, whereby a trained neural network is able to provide a correct matching in the form of output data for a set of previously unseen input data. Learning typically occurs by example through training, where the training algorithm iteratively adjusts the connection weights (synapses). Back-propagation (BP) is one of the most famous training algorithms for multilayer perceptrons[6].Learning algorithms has significant impact on the performance of neural networks, and the effects of this depend on the targeted application. The choice of suitable learning algorithms is therefore application dependent.

IV. SPIHT ALGORITHM

The different compression methods were developed specifically to achieve at least one of those objectives. What makes SPIHT really outstanding is that it yields all those qualities simultaneously. So, if in the future you find one method that claims to be superior to SPIHT in one evaluation parameter (like PSNR) remember to see who wins in the remaining criteria. SPIHT stands for Set Partitioning in Hierarchical Trees. The term Hierarchical Trees refers to the quad trees that we defined in our discussion of EZW. The images obtained with wavelet-based methods yield very good visual quality. At first it was shown that even simple coding methods produced good results when combined with wavelets and is the basis for the most recently JPEG2000 standard. However, SPIHT belongs to the next generation of wavelet encoders, employing more sophisticated coding. In fact, SPIHT exploits the properties of the wavelet-transformed images to increase its efficiency. Our discussion of SPIHT will consist of three parts. We shall refer to it as the Spatial-orientation Tree Wavelet (STW) algorithm. STW is essentially the SPIHT algorithm, the only difference is that SPIHT is slightly more careful in its organization of coding output. Second, we shall describe the SPIHT algorithm. It will be easier to explain SPIHT using the concepts underlying STW. Third, we shall see how well SPIHT compresses images.

The only difference between STW and EZW is that STW uses a different approach to encoding the zero tree information. STW uses a state transition model. From one threshold to the next, the locations of transform values undergo state transitions. This model allows STW to reduce the number of bits needed for encoding. Instead of code for the symbols R and I output by EZW to mark locations, the STW algorithm uses states I_R, I_V , S_R, and S_V and outputs code for state-transitions such as IR → IV , SR → SV , etc. To define the states involved, some preliminary definitions are needed.

For a given index m in the baseline scan order, define the set D (m) as follows. If m is either at the 1st level or at the all-low pass level, then D(m) is the empty set ;. Otherwise, if m is at the j^{th} level for j > 1, then D (m) = f Descendents of index m in quad tree with root mg. The significance function S is defined by:

$$S(m) = \begin{cases} \max | w(n), \text{ if } D(m) \neq 0 \\ \infty \quad , \text{ if } D(m) = 0 \end{cases}$$

With these preliminary definitions in hand, we can now define the states.For a given threshold T, the state's I_R, I_V, S_R, and S_V are defined by

m ∈ I_R if and only if |w(m) | < T; S(m)< T

m ∈ I_V if and only if |w(m) | < T; S(m)<T

m ∈ S_R if and only if |w(m) | < T; S(m)<T

m ∈ S_V if and only if |w(m)| <T; S(m)<T

Fig.1 we show the state transition diagram for these states when a threshold is decreased from T to T 0 < T. Notice that once a location m arrives in state SV, and then it will remain in that state. Furthermore, there are only two transitions from each of the states IV and SR, so those transitions can be coded with one bit each.

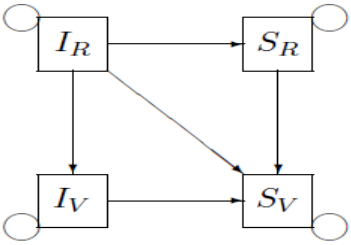

Figure 1. State Transaction Diagram for STW

A simple binary coding for these state transitions is shown in Table 1.

TABLE I. CODE FOR STATE TRANSITIONS, ● INDICATES THAT SV → SV TRANSITION IS CERTAIN (HENCE NO ENCODING NEEDED).

Old/New	I_R	I_V	S_R	S_V
I_R	00	01	10	11
I_V		0		1
S_R			0	1
S_V		●		●

V. FEATURE OF THE ALGORITHM

Now that we have laid the groundwork for the STW algorithm, we can give its full description.

STW encoding

Step 1 *Initialize*. Choose initial threshold, *T* = *T*0, such that *all* transform values satisfy |*w* (*m*)| < *T*0.

Step 2 *Update threshold*. Let *Tk* = *Tk*-1/2.

Step 3 Apply the back propagation to Train the Network

Step 4 *Dominant pass*. Use the following procedure to scan through indices in the dominant list (which can change as the procedure is executed).

Do

Get next index *m* in dominant list Save old state Sold = S (*m*, *Tk*-1)

Find new state Snew = S (*m*,*Tk*)

Output code for state transition Sold → Snew

If Snew≠Sold then do the following

If Sold≠*SR* and Snew≠*IV* then

Append index *m* to refinement list

Output sign of *w* (*m*) and set *wQ* (*m*) =*Tk*

If Sold=*IV* and Snew=*SR* then

Append child indices of *m* to dominant list

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 2, Issue 3, September 2012

If Snew=*SV* then Remove index *m* from dominant list

Step 5 while images feature value gets trained.

Step 6 *Refinement pass*. Scan through indices *m* in the refinement list found with higher threshold values *Tj*, for $j < k$ (if $k = 1$ skip this step). For each value *w* (*m*), do the following:

If |w (m)| ∈ [wQ (m), wQ (m) + Tk], then
Output bit 0
Else if |w (m)| ∈ [wQ (m) + Tk, wQ (m) + 2Tk], then
Output bit 1
Replace value of wQ (m) by wQ (m) + Tk.

Step 7: *Loop*. Repeat steps 2 through 4.

A compressed representation having about 20,000 non-zero coefficients is usually very close to the original in visual quality [3, 41. As suggested by DeVore et a1 [4], the number of non-zero coefficients as well as the PSNR are reasonable measures of the quality of the compressed representation. We would therefore like our algorithm to enable us to control both the number of significant coefficients and the: PSNR. Further, we would like the algorithm to have low computational requirements by computing only the required coefficients as opposed to all N2 coefficients for an N x N image. Let us assume that we have decided to have Lnz non-zero coefficients in the representation. A typical value for Lnz may be between 10,000 and 20, 0100. We will now present an algorithm that has the following important features:

It is possible to control both (i) the number of significant coefficients in the representation and (ii) the PSNR. Its performance is comparable to the best available algorithms.

An upper bound on the number of coefficients to be computed is min (8Lnz, N2). A typical value for this number is min (4Ln, N2). This is in contrast to most wavelet based algorithms where the number of coefficients to be computed is N2. This is of concern because the computational expense of this task increases dramatically as we move from 512 x 512 to 1920 x 1152 or even larger sized

Step 8 (Apply the wavelets to the trained the image dataset

Step 9 (Loop). Apply the iteration for images in various wavelets

To see how STW works and how it improves the EZW method it helps to reconsider, we show STW states for the wavelet transform using the same two thresholds as we used previously with EZW. It is important to compare the three quad trees enclosed in the dashed boxes with the corresponding quad trees. There is a large savings in coding output for STW represented by these quad trees. The EZW symbols for these three quad trees are +I I I I, -I I I I, and +RRRR. For STW, however, they are described by the symbols +SR, -SR, and +SR, which is a substantial reduction in the information that STW needs to encode. There is not much difference between STW and SPIHT. The one thing that SPIHT does differently is to carefully organize the output of bits in the encoding of state transitions in Table 1, so that only one bit is output at a time. For instance, for the transition IR →SR, which is coded as 1 0 in Table 1, SPIHT outputs a 1 first and then (after further processing) outputs a 0. Even if the bit budget is exhausted before the second bit can be output, the first bit of 1 indicates that there is a new significant value.

The SPIHT encoding process, as described in [6], is phrased in terms of pixel locations [i, j] rather than indices m in a scan order. To avoid introducing new notation, and to highlight the connections between SPIHT and the other algorithms, EZW and STW, we shall rephrase the description of SPIHT from [6] in term of scanning indices. We shall also slightly modify the notation used in [6] in the interests of clarity. First, we need some preliminary definitions. For a given set I of indices in the baseline scan order, the significance ST [I] of I relative to a threshold T is defined by

$$\mathcal{S}_T[\mathrm{I}] = \begin{cases} 1, & \text{if } \max\limits_{n\in\mathrm{I}} |w(n)| \geq T \\ 0, & \text{if } \max\limits_{n\in\mathrm{I}} |w(n)| < T. \end{cases}$$

It is important to note that, for the initial threshold T0, we have ST0 [I] = 0 for all sets of indices. If I is a set containing just a single index m, then for convenience we shall write ST [m] instead of ST [{m}]. For a succinct presentation of the method, we need the following definitions of sets of indices:

D (m) = {Descendent indices of the index m}
C (m) = {Child indices of the index m}
G (m) = D (m) – C (m)
= {Grandchildren of m, i.e., descendants which are not children}.

In addition, the set H consists of indices for the L^{th} level, where L is the number of levels in the wavelet transform (this includes all locations in the all-low pass sub band as well as the horizontal, vertical, and diagonal sub bands at the L^{th} level). It is important to remember that the indices in the all-low pass sub band have no descendants. If m marks a location in the all low pass sub band, then D (m) =∅ SPIHT keeps track of the states of sets of indices by means of three lists. They are the list of insignificant sets (LIS), the list of insignificant pixels (LIP), and the list of significant pixels (LSP). For each list a set is identified by a single index, in the LIP and LSP these indices represent the singleton sets {m} here m is the identifying index. An index m is called either significant or insignificant, depending on whether the transform value w (m) is significant or insignificant with respect to a given threshold. For the LIS, his index m denotes either D (m) or G (m). In the former case, the index m is said to be of type D and, in the latter case, of type G.

VI. CODING USING NEURAL NETWORKS

One solution to the problems associated with the calculation of the basis vectors through Eigen decomposition of the covariance estimate is the use of iterative techniques based on neural network models. These approaches require less storage overhead and can be more computationally efficient. As well, they are able to adapt over long-term variations in the image statistics. In 1949, Donald Hebb proposed a mechanism whereby the synaptic strengths between connecting neurons can be modified to effect learning in a neuro-biological network. Hebb's postulate of

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 2, Issue 3, September 2012

learning states that the ability of one neuron to cause the firing of another neuron increases when that neuron consistently takes part in firing the other. In other words, when an input and output neuron tend to fire at the same time, the connection between the two is reinforced. For artificial neural networks, the neural interactions can be modeled as a simplified linear computational unit. The output of the neuron, y, is the sum of the inputs {x1, x2… xN} weighted by the synaptic weights {w1, w2……, wn}, or in vector notation,

$y = w^T x$

Taking the input and output values to represent "firing rates," the application of Hebb's postulate of learning to this model would mean that a weight wi would be increased when both values of xi and y are correlated. Extending this principle to include simultaneous negative values (analogous to inhibitory interactions in biological networks), the weights w would be Modified according to the correlation between the input vector x and the output y. A simple Hebbian rule updates the weights in proportion to the product of the input and output values as

$w(t+1) = w(t) + \alpha y(t) x(t)$

Where α is a learning-rate parameter. However, such a rule is unstable since the weights tend to grow without bound.

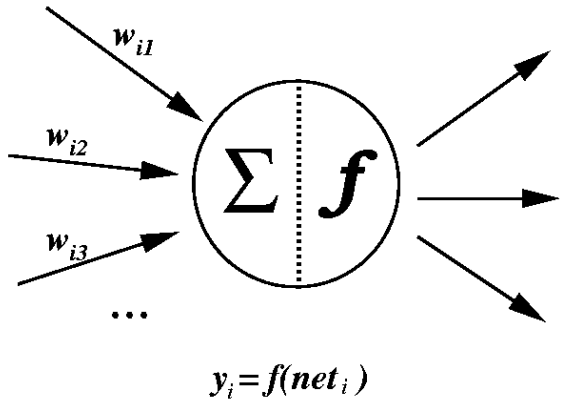

Figure 2. Linear Network

Algorithm by Neural Network

- Choose training pair and copy it to input layer
- Cycle that pattern through the net
- Calculate error derivative between output activation and target output
- Back propagate the summed product of the weights and errors in the output layer to calculate the error on the hidden units
- Update weights according to the error on that unit until error is low or the net settles.

VII. PROPOSED WORK

The above algorithms are compared and the results are shown in the figures. The PSNR, CR, BPP and MSE values for the images compressed by SPIHT, EZW and STW .The PSNR, CR, BPP, MSE values are calculated by using the following formula. Peak Signal to Noise Ratio (PSNR) is generally used to analyze quality of image, sound and video files in dB (decibels). PSNR calculation of two images, one original and an altered image, describes how far two images are equal.

MSE:Mean-Squareerror.

x: width of image.

y: height.

x*y: number of pixels (or quantities).

The formula for calculating the Peak Signal to noise Ratio is as follows:

$$PSNR(dB) = 10 * \log(\frac{255^2}{MSE})$$

$$MSE = \sum_{i=1}^{x}\sum_{j=1}^{y}\frac{(|A_{ij} - B_{ij}|)^2}{x * y}$$

Bits per Pixel

The terminology for image formats can be confusing because there are often several ways of describing the same format. This topic explains what the terms mean. If an image is 24 bits per pixel, it is also called a 24-bit image, a true color image, or a 16M color image. Sixteen million is roughly the number of different colors that can be represented by 24 bits, where there are 8 bits for each of the red, green, and blue (RGB) values. A 32-bit image is a specialized true-color format used in image files, where the extra byte carries information that is either converted or ignored when the file is loaded. The extra byte is used for an additional color plane in CMYK files, which are specialized files for color printing. By default, converts the values to 24-bit RGB values when loading the image. The additional byte may also be used for an Alpha channel, which carries extra information such as a transparency indicator. The number of distinct colors that can be represented by a pixel depends on the number of bits per pixel (bpp). A 1 bpp image uses 1-bit for each pixel, so each pixel can be either on or off. Each additional bit doubles the number of colors available, so a 2 bpp image can have 4 colors, and a 3 bpp image can have 8 colors: 1 bpp, 21 = 2 colors (monochrome)

For color depths of 15 or more bits per pixel, the depth is normally the sum of the bits allocated to each of the red, green, and blue components. High color, usually meaning 16 bpp, normally has five bits for red and blue, and six bits for green, as the human eye is more sensitive to errors in green than in the other two primary colors. For applications involving transparency, the 16 bits may be divided into five bits each of red, green, and blue, with one bit left for transparency. A 24-bit depth allows 8 bits per component. On some systems, 32-bit depth is available: this means that each 24-bit pixel has an extra 8 bits to describe its opacity (for purposes of combining with another image).

The compression ratio (that is, the size of the compressed file compared to that of the uncompressed file) of lossy image codec's is nearly always far superior to that of the audio and still-image equivalents. Video can be compressed immensely (e.g. 100:1) with little visible quality loss. Audio can often be compressed at 10:1 with imperceptible loss of quality Still images are often lossily compressed at 10:1, as with audio, but the quality loss is more noticeable, especially on closer inspection. The compression rate is 5 to 6 % in lossy

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 2, Issue 3, September 2012

compression while in lossless compression it is about 50 to 60 % of the actual file.

VIII. CONCLUSION

This paper has presented a Neural with different pre-process method. This method shows a good performance on the reconstructed image. we found more better result in SPHIT as compared to STW and EZW produces less Mean Square Error (MSE), for example image 1.jpg is of size 25265 kb and we apply all wavelet and neural network to compress this image it can compress this image .Where as if we use SPIHT it can compressed this image only up to 22142 kb. Hence SPHIT produces batter result as compare to other wavelet and Neural Network. Accuracy of the Image can also be cleared with the help of Training of Neural Network but the time consuming in more. So the SPHIT is better for time and accuracy.

REFERENCES

[1] S.P.Raja1, Dr. A. Suruliandi" Performance Evaluation on EZW & WDR Image Compression Techniques", ICCCCT'10, IEEE, 978-1-4244-7770-8/10, 2010.

[2] G.M. Davis, A. Nosratinia. "Wavelet-based Image Coding: A verview. Applied and Computational Control", Signals and Circuits, Vol. 1, No. 1, 1998.

[3] M. Antonini, M. Barlaud, P. Mathieu, I. Daubechies. Image coding using wavelet transform. IEEE Trans. Image Proc., Vol. 5, No. 1, pp. 205-220, 1992.

[4] J.M. Shapiro. Embedded image coding using zerotrees of wavelet coefficients. IEEE Trans. Signal Proc., Vol. 41, No. 12, pp.3445 -3462, 1993.

[5] Said, W.A. Pearlman. "Image compression using the spatial-orientation tree". IEEE Int. Symp. On Circuits and Systems, Chicago, IL, pp. 279-282, 1993.

[6] Said, W.A. Pearlman. "A new, fast, and efficient image codec based on set partitioning in hierarchical trees". IEEE Trans. on Circuits and Systems for Video Technology, Vol. 6, No. 3, pp. 243-250, 1996.

[7] Shapiro J.M. "Embedded image coding using zero trees of wavelet coefficients". IEEE Trans. Signal Proc., Vol. 41, No. 12, pp. 3445-3462, 1993.

[8] M. Antonini, M. Barlaud, P. Mathieu, and I. Daubechies, ""Image Coding using Wavelet Transform"", IEEE Transactions on Image Processing, vol. 1, pp. 205 - 220, April 1992.

[9] J. Shapiro, ""Embedded Image Coding Using Zero trees of Wavelet Coefficients"", IEEE Transactions on Signal Processing, vol. 41, pp. 3445 - 3462, December 1993.

[10] R. DeVore, B. Jawerth, and B. Lucier, ""Image Compression through Wavelet Transform Coding"", IEEE Transactions on Information Theory, vol. 38, pp. 719 - 746, March 1992.

[11] I. Katsavounidis and C. J. Kuo, ""Image compression with embedded wavelet coding via vector quantization"", in SPIE Conference