J.Thi-Qar Sci. No. (2) Vol.1 Aug./2008

ISSN 1991- 8690 الترقيم الدولي ٨٦٩٠ - ١٩٩١

website: http://jsci.utq.edu.iq Email: utjsci@utq.edu.iq

Image Compression Using Adaptive Neural Network Based on Pixels Information

علي حسين حسن

جامعة ذي قار – كلية العلوم – قسم الحاسبات

ali_hussen_hassan@yahoo.com

Abstract

The aim of this work is to create an image compression technique using one hidden layer feed forward neural network in which the number of the input neurons is determined adaptively according to pixels values.

The technique proposes scanning the image pixels values and takes one sample value from the occurring values ranging from 0 to 255 gray scales. The taken values are fed into the neural network as input signals instead of the original image pixels. This process exhibits decreasing the number of the input neurons, the computing size, memory space and better rate of convergence of the back propagation training algorithm.

The proposed work has been demonstrated through several experiments and shows very promising results in compression rates as well as in the reconstructed image.

1-Introduction

Digital images are data intensive and requires large amount of storage space and transmission cost and time. The fundamental goal of the image compression is to reduce the bit rate for the transmission or data storage while maintaining analog acceptable fidelity for image quality [1].
Image compression is the process of reducing the number of bits that required representing an image. Its benefits are to reduce the storage space and transmission time. In general; the compression techniques are broadly classified into two categories, namely; lossless and lossy compression techniques [2][3]. Lossless compression refers to a technique where no information is permanently lost during compression or decompression of the image data. While lossy compression refers to a technique where some bits of information are permanently lost during compression or decompression stages; examples of conventional techniques are GIF as lossless compression and JPEG as lossy compression methods.
Neural networks are succefully used in compression field, not only ANN based techniques can provide sufficient compression rates of the data in question, but security is easily maintained. This occurs because the compressed data that is sent along a communication channel is encoded and does not resemble its original form[4]. In most of the neural networks based compression techniques, the image is divided into a non overlapped blocks such as 8*8, 16*16 or 64*64 (or as desired) sub image blocks. These blocks then fed as input and targets for the neural network. After the neural network trained, the compression occurs in the hidden layer, where the neurons final values are encoded. These encoded values are transmitted (as well as saved) for the receiving side, in which, the image is reconstructed. These methods have the advantage of high compression ratios but the disadvantage of these method are they must make tracking of the whole image pixels during the training phase this lead to increase the network size and computation time as shown in figure (1).
In this paper, a scanning mechanism that detects the pixels values; which construct the image, then one sample value from each equal value is taken to perform the neural network input pattern. This procedure reduced the huge size of image that to be manipulated and also the neural network architecture.

2- Related Works

Srivasan and Vyas [2] made an image specific technique for compression of grayscale images using principal component analysis. Kulkari et al [4] presented a direct solution method based neural network for image compression; the technique employs training by a non-iterative (direct solution). Paul et al [5] used the bottleneck-type neural network architecture for image compression, they divide the image into 8*8 sub image, and these sub images are encoded into a 16 code value (i.e. 64 pixels into 16 pixels). Kayal [6] proposed an edge preserving image compression technique using one hidden layer feed forward neural network, in his work many operations were applied to the image before the compression algorithm such as edge detection and threshold to reduce the image size. Mokhtari and Aoued [7] made an optimization of fractal image compression based on Kohonen neural networks. Welstead [8] presented a scheme for improving encoding times for fractal image compression; he combined feature extraction with domain classification using a self-organizing neural network. Alrt and

Brause [9] proposed an encoding of images by only a few important components or image primitives which are principal component analysis. Burges et al [10] explored the use of neural networks to predict wavelet coefficients for image compression. Kaarna [11] applied neural networks to wavelet filter selection, and found a filter for the compression of each multi spectral image.

3- Proposed Algorithm

The gray bmp image (bitmap image) that is to be compressed is scanned and the pixels values are recorded (which is similar to histogram process) then one sample value from each equal-value is taken to perform the input pattern to the neural network. These values are fed into the input layer neurons which have a 256 neuron as maximum, the hidden layer in which the compression occurs has many neurons which are less than the input layer (which is determined adaptively depending on the scanning process), the output layer has a number of neurons equal to the image pixels figure (2). The activation function used is the hyperbolic activation function.

The input pattern as well as the output target (which is the replicating of the original image matrix) is normalized between 1 and 0.

The network is trained using error back propagation algorithm. At the end of training, the output values of the hidden layer neurons are quantized. For accurate coding a 16-bit schema used to encode these outputs. These quantized values are transmitted/stored. At the receiving side the network of figure (3) is used for the decompression process. In decompression process, we start in decoding the received values, then a single forward calculation is performed to reconstruct the image.

3.1. Compression Side

Step I :- Initialization Process:-

a) Put the contents of the image file in the vector called it *y*.

i.e. ***y=imagefile*……………………...(1)**

b) Normalize the vector y in the range (0.0 – 1.0).

***y=y/maximum value*…………...(2)**

c) Scan the vector y and construct an index matrix that contains a pixel value and its number of occurring (which is similar to the histogram process).

d) Construct the X vector that contains one value from each the equal pixels values. We notice that, the X values are ranged from 0 to at maximum case 255 value. And these values are also ranged from 0 to 255. This fact led to adapt the number of the X vector to be variable and do not increase at worst case over 255 values.

e) Determine the weight matrix to contain random values with rows equal to the vector *y* length and columns equal to the vector *x* size. Call it ***W[R][C].***

Step II) Neural Network:- figure (2)

a) Let the vector *x* be the input vector to the neural network, *y* is the desired output, and ***W*** is the weight matrix.

b) The activation function is hyperbolic function.

c) By assuming that the sum of inputs to the i^{th}-cell of the K^{th}-layer is $\boldsymbol{S_i^k}$, its output is O_i^k, the weighting value from i^{th}-cell of $(K-1)^{th}$-layer to the i^{th}-cell of the K^{th}-layer is wij. F (.) is the relation input between input and output (activation function). Therefore,

$$S_i^k = \sum_j w_{ij} S_i^{k-1} \ldots\ldots\ldots\ldots\ldots \quad \textbf{(3)}$$

$$O_i^k = F(S_i^k) \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \textbf{(4)}$$

For the output layer, the activation function is

$$O = F(S) = \frac{\exp(S) - \exp(-S)}{\exp(S) + \exp(-S)} \quad \ldots\ldots\textbf{(5)}$$

d) We trains the ANN used by the Error Back propagation Algorithm which is well defined and derived in [12] and [13]:-

$$\Delta W_{kj} = \eta . \delta_k . O_j \ldots\ldots\ldots\ldots\ldots\ldots \textbf{(6)}$$

$$\Delta W_{ji} = \eta . \delta_j . O_i \quad \ldots\ldots\ldots\ldots\ldots \textbf{(7)}$$

$$\delta_k = K\,(\tau_k - O_k) \quad \ldots\ldots\ldots\ldots\ldots \textbf{(8)}$$

$$\delta_j = O_j\,(1 - O_j)\,\Sigma\,\delta_k \,.\, \Delta W_{kj} \ldots\ldots\ldots \textbf{(9)}$$

e) After the ANN is trained well with a sufficient small error, then we take the hidden layer output values where the compression occurs. For good compression ratios, the number of the hidden layer neurons is chosen small enough.

f) Quantization hidden layer output. For accurate coding a 16-bit schema used to encode these outputs.

g) The quantized values are transmitted or stored.

3.2. Decompression Side: - figure (3)

a) Decode the received signals.

b) Apply single forward calculation by performing equations (3,4 and 5) to reconstruct the image.

c) Reconstruct the original values by multiplying by the maximum value.

4- Results

The signal to noise ratio (SNR) and the compression ratio (CR) [3,6] are computed using equations (10) and (11) for a number of experimental images as shown in figures (4,5,6,7 and 8). Table (1) summarizes these results.

$$SNR = \frac{\log_{10}\left[\sum_{i=1}^{n}\sum_{j=1}^{m}\left(I(i,j) - O(i,j)\right)^2\right]}{n * m} \ldots \textbf{(10)}$$

$$CR = \frac{\text{Original Image Size}}{\text{Compressed Image Size}} \ldots\textbf{. (11)}$$

Where, I(i,j) original image matrix
O (i,j) reconstructed image matrix.

The adaptive characteristics of the proposed approach provide a low size in structuring the architecture of the network; where the number of neurons in the input layer is reduced to 256 neurons at maximum; which not only speed up the processing but also makes the network less susceptible to failure or saturation during training. This technique exhibits a fast rate of convergence with high compression ratios and a good quality regenerated.

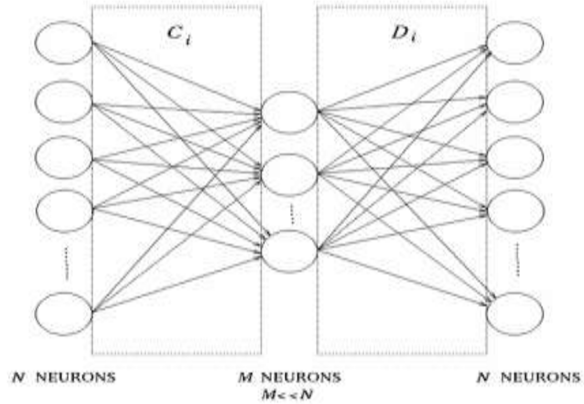
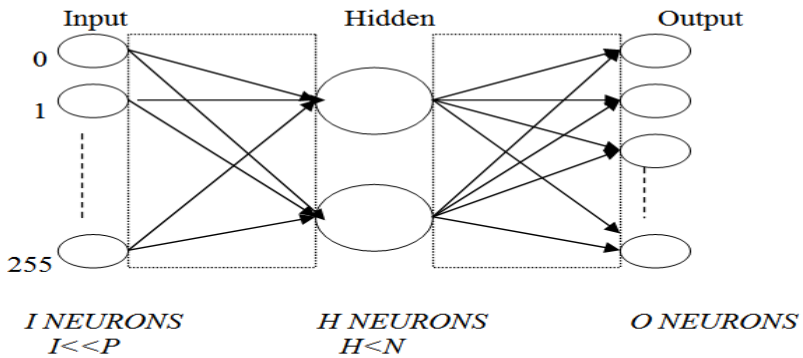


Figure1:- Image compression based neural networks



I:- number of input neurons; H:- number of hidden layer neurons;
O:- number of output neurons

Figure 2:- Proposed neural network

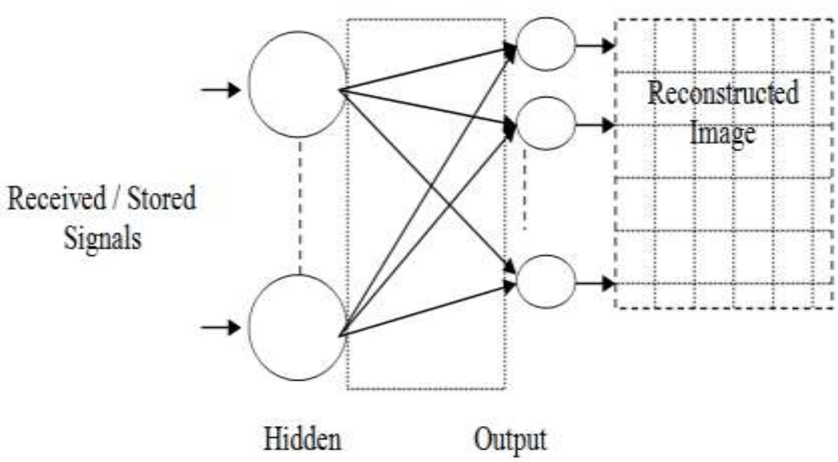
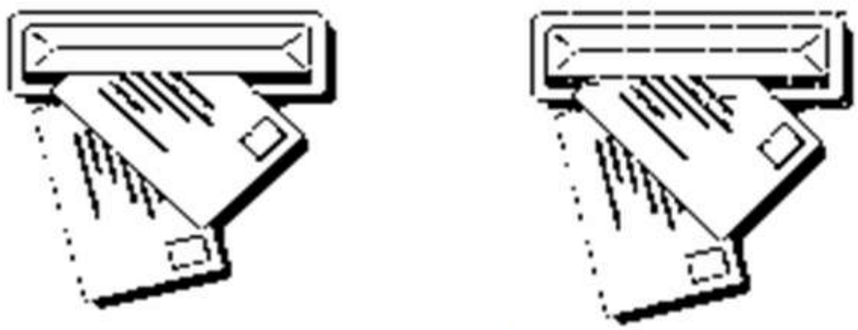
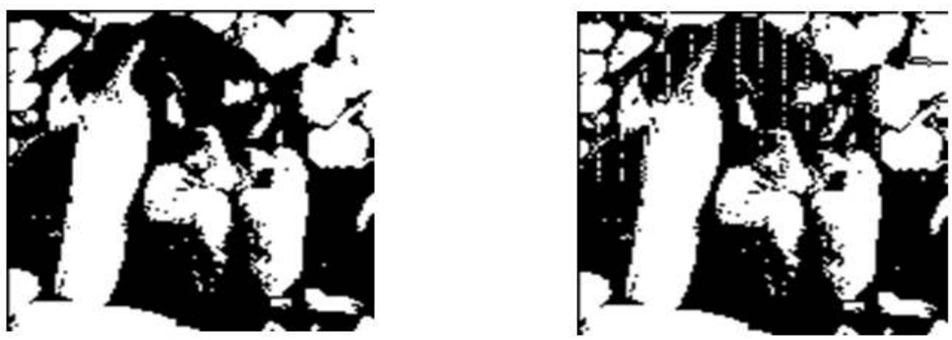


Figure 3:- Neural network receiving side



a) original image b) reconstructed image

Figure 4:- Envelope image compression



a) original image b) reconstructed image

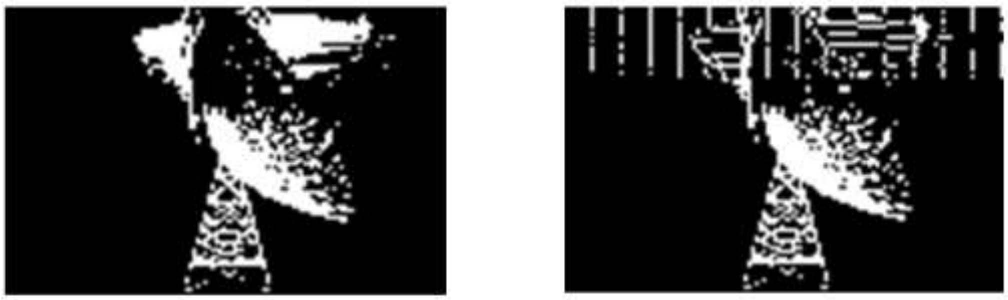
Figure 5:- Peppers image compression

a) original image b) reconstructed image

Figure 6:- woman face image compression

a) original image b) reconstructed image

Figure7 :- Email virus image compression



a) original image b) reconstructed image

Figure 8:- Antenna dish image compression

Table1: Compression Results

Image name	Size	Iteration	SNR	CR
Envelope	100*100	476	0.003085	37
Peppers	128*128	580	0.002183	36
Woman face	170*170	1974	0.002437	30
Email virus	255*191	987	0.001027	39
Antenna dish	120*80	167	0.019585	37

References

1. S. Annadurai, R. Shanmugalakhmi and V. Srinivasan, “Color image algorithm based on adaptive principal components neural network”, ANSCSE9 Mahidol University, Bangkok, Thailand, March 23-25 2005, http//www.csep.hpcc.nectec.or.thanscse.
2. Bharadwaj Srinivasan and Vishnu Vyas, ”Compression of grayscale images using image-specific encoding based on unsupervised Hebbian learning”, SRM Engineering College, 2003, http//www.stanford.edu.
3. David Salmon, “Data Compression The Complete Reference”, Springer, 1998.
4. S Kulkari, B. Verma and M. Blumesteia, “Image compression using a direct solution method based neural network”, Griffith University, Australia, 1997, http//www.citeseer.ist.psu.edu.
5. Paul Watta, Brijesh Desaie, Norm Danug and Mohamad Hassoun, “Image Compression using Backprop”, 1996, http//www.neuron.eng.wayne.edu.
6. Kausik Kayal, “Hi-End image compression technique using artificial neural networks”, Tata consultancy services, 2000, http//www.tcs.com.
7. Mokhtari Mohamed and D Boukelif Aoued, “Optimization of fractal image compression based on Kohonen neural networks”, University of SBA, Algeria, 2006, www.eurasip.org.
8. Stephen Welstead, “Self-organizing neural network domain classification for fractal image coding”, Proceeding of the IASTED international conference Artificial Intelligence and Soft Computing, July 27-31, 1997, Banff Canada.
9. Bjorn Arlt and Rudiger W. Brause, “Image encoding by independent principal components”, FB Informatica, J.W. Geothe-Universitat, Frankfort/Main, Germany, 1998, http//cs3.pvamu.edu.
10. Chistopher J.C. Burges, Patrice Y. Simard and Henrique S. Malvar, “Improving wavelet image compression with neural networks”, Microsoft research centre, August 2001, http//research.microsoft.com
11. Arto Kaarna, “Application of neural networks to wavelet filter selection in multispectral image compression”, Proceedings of the NORSIG Nordic Signal Processing Symposium, Kolmården, Sweden, June 13 - June 15, 2000, http//www.es.isy.liu.se.
12. Edward J. Rzempoluck, “Neural networks data analysis using simulent”, Springer, 1997.
13. Jacek M. Zurada, “Introduction to Artificial Neural Systems”, Jaico Publishing House, 1996.

ضغط الصور باستخدام الشبكات العصبية التكيفية اعتمادا على معلومات نقاط الشاشة

علي حسين حسن

جامعة ذي قار – كلية العلوم – قسم الحاسبات

الخلاصة

الهدف من هذا البحث هو ضغط الصور باستعمال شبكة عصبية ذات طبقة مخفية وحيدة وطبقة إدخال فيها عدد العصبونات تعدل بالاعتماد على قيم البكسل(Pixels) المكونة للصورة.
الطريقة المقترحة تقوم بعملية مسح للصورة حيث تحدد القيم المكونة للصورة ثم تأخذ قيمة واحدة عن كل مجموعة قيم مشتركة بالقيمة, هذه القيم تكون محدده بين (٠ و٢٥٥ المقياس الرمادي). ثم نقوم بتغذية هذه القيم المأخوذة إلى الشبكة العصبية لتكون إشارات الإدخال. وبذلك يكون عدد العصبونات المكونة لطبقة الإدخال يتكون من ٢٥٦ عصبون كحد أعلى بدلا من أن تكون جميع قيم عناصر الصورة إشارات إدخال للشبكة.
هذه العملية قللت بشكل كبير من حجم طبقة الإدخال, إجراء عملية الحسابات, حجم الذاكرة المستخدمة, وجعلت طريقة الانتشار العكسي المستخدمة في عملية التدريب أسرع في عملية التدريب.
الطريقة جربت لعدد من الصور وأعطت نتائج جيدة من حيث نسبة الضغط وجودة الصورة المشكلة من الصورة المضغوطة.