www.ijecs.in
International Journal Of Engineering And Computer Science ISSN:2319-7242
Volume – 5 Issue -02 February, 2014 Page No. 15681-15684

Image Compression using Multilayer Feed-Forward Artificial Neural Network with Levenberg Marquardt.

Mr. Gade M.R, Asst. Proff. Deshpande A.S.
ICOER, Wgholi, Pune
gademangesh07@gmail.com
ICOER Wagholi, Pune
panuradha2010@gmail.com

Abstract: This paper presents a neural networks as image processing tools for image compression, present a direct solution method based neural network for image compression. Digital images require large amounts of memory for storage. Thus, the transmission of an image from one computer to another can be very time consuming. By using data compression techniques, it is possible to remove some of the redundant information contained in images, requiring less storage space and less time to transmit. To observe and compare the different algorithms of image compression and Levenberg marquardt is best as compare to other algorithms.

Keywords— Image compression, artificial neural networks, Levenberg Marquardt, Conjugate Gradient, Gradient descendent, Gradient descendent with momentum.

1. Introduction

Digital Images contain huge amount of data. So to decrease the memory to store the image and to reduce the bandwidth requirement to transmit, it is essential to compress it as efficiently. To maintaining an acceptable level of image quality, Image compression is the representation of an image in digital form with as few bits as possible. In literature by several researchers The Many compression techniques have been already proposed such as transform image coding, predicative image coding and vector quantization. Among these, simple and efficient way of performing particularly at low bit rates image compression is transform image coding. For past several years standard choice image compression has been a JPEG (DCT based). As they are localized in both image space and spectral frequency domains Wavelet transforms have become the most prevalent technique among these transform image coding techniques, Due to their ability to approximate complicated nonlinear functions the artificial neural networks are popular and found its application in many fields, mainly in function approximation. Three types of learning used in different application that is supervised learning , unsupervised learning and reinforcement learning. Learning algorithm can be categorized as a supervised neural network and most frequently used neural network in practical situations is the multilayer perceptron (MLP) along with the back propagation (BP).

2. Need of image compression

To represent a given quantity of information data compression is a process of reducing the amount of data required. Note that information and data are not the same; data are the means by which information is conveyed. Since various amounts of data can be used to represent the same amount of information, representations that irrelevant or repeated information are said to contain redundant data. In two representations of the same information as b and b' Denoting the number of bits , the relative data redundancy of the representation with b bits

$$R = 1 - \frac{1}{C}$$

Where C is the compression ratio:

$$C = \frac{b}{b'}$$

If C = 10, for instance, the larger representation has 10 bits of data for every 1 bit of data in the smaller representation. The corresponding relative data redundancy of the larger representation is R = 0.9 indicating that 90% of its data is redundant. 'b' is the number of bits needed to represent an image as a 2D array of intensity values[10]. We notice that such arrays being preferred formats for human viewing are not optimal from a compact data representation viewpoint. Such arrays have three types of data redundancy:

1. Coding redundancy: intensities represented by bits contain more data than needed.
2. Spatial and temporal redundancy: - Each pixel is similar to or dependent on (correlated) neighboring pixels. In video, pixels are correlated temporally. Therefore, information is replicated.
3. Irrelevant information: - Most 2D intensity arrays contain information that is ignored by the human visual system and/or extraneous to the intended use of image. It is redundant in the sense that it is not used. Compression is achieved when one or more of these redundancies is reduced or eliminated. An image without compression would require 3MB of storage and 7 minute of transmission, utilizing a high speed, 64 kbps ISDN line. If the image is compressed at 10:1 compression ratio, the storage requirement is reduced to 300 KBand the transmission time is reduced to less than 7 second. Images file in an uncompressed format very large, and the internet especially for people using a 56 kbps dialup modem, can be pretty slow. This

DOI: 10.18535/ijecs/v5i2.6

combination could seriously limit one of the webs most appreciated aspects assist ability to present images easily.

3. Learning algorithm

3.1 LM algorithm

In order to make sure that the approximated Hessian matrix $J^T J$ is invertible, Levenberg–Marquardt algorithm introduces another approximation to Hessian matrix:

$$H \approx J^T J + \mu I \quad (1)$$

Whereμ is always positive, called combination coefficient I is the identity matrix

From Equation 1, one may notice that the elements on the main diagonal of the approximated Hessian matrix will be larger than zero. Therefore, with this approximation (Equation 1), it can be sure that matrix H is always invertible. By combining the update rule of the Gauss–Newton algorithm and equation 1, the update rule of Levenberg–Marquardt algorithm can be presented as

$$w_{k+1} = w_k - \left(J_k^T J_K + \mu I\right)^{-1} J_K e_k \quad (2)$$

As the combination of the steepest descent algorithm and the Gauss–Newton algorithm, the Levenberg– Marquardt algorithm switches between the two algorithms during the training process. When the combination coefficient μ is very small (nearly zero), Equation 2 is approaching to Equation of Gauss–Newton algorithm is used. When combination coefficient μ is very large, Equation 2 approximates to Equation 1 and the steepest descent method is used. If the combination coefficient μ in Equation 2 is very big, it can be interpreted as the learning Coefficient in the steepest descent method [9].

$$\alpha = \frac{1}{\mu}$$

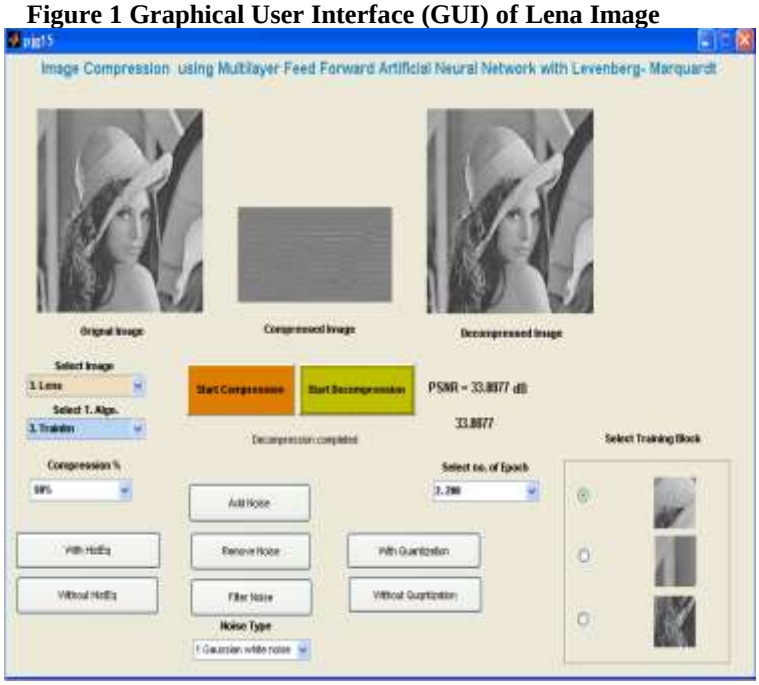
4 . Algorithm steps

Step1: Read the input image.
Step2: Image is divided into number of blocks.
Step3: Each blocks scanning for complexity level. Step4: Initialization of neurons.
Step5: Apply each neuron on the input layer from scan vectors.
Step6: Execute the operation depending on the weights and the logic involve.
Step7: Passes them to the hidden layer.
Step8: Repeat the same as in step6.
Step9: Reassemble the outputs.
Step10: Neural network training and wait the weights.

5. Results and discussion

In this project MATLAB is used to implement the program. The well-known 'Lena' gray scale image (256 × 256) has been used to demonstrate the technique. Each pixel in an image can be denoted as a coefficient, which represents the intensity of the image at that point. Then, the idea of compressing an image is to encode these coefficients with reduced bits and at the same time, retain the quality of the image to satisfactory limits.The multi-layer feed-forward neural net has been used to compress images. The rice image that has been used for compression purposes is a 256 × 256 image. This image can be broken into blocks of size 32 ×32. There will then be 1024 pixels per block. Totally, there will be [32 × 32] = 64 blocks. The 1024 pixels in each block then becomes the input vector to the neural net. If each pixel is encoded using 8 bits, then the total number of bits to be transmitted without compression is [256 × 256 × 8] for a [256 × 256] pixel image. A [256 × 256] pixel image is split into [4 × 4] or [8 × 8] or [16 × 16] pixel sub-images. The normalized pixel value of the sub-image is the input to the nodes. 64 input layers (in case of [8 × 8] sub-image size) are taken with 1 pixel input to each layer. The three-layered back propagationlearning network has been trained with each sub image. The number of neurons in the hidden layer will be taken according to the desired compression. Here, we have taken 8 hidden layers. The number of neurons in the output layer will be the same as that in the input layer (64 in our case). The input layer and output layer are fully connected to the hidden layer.

Figure 1 Graphical User Interface (GUI) of Lena Image



Above figure shows GUI of project in this it have three axes on which first axes shows original image second axes shows compressed image and third axes shows decompressed image. It has three popup buttons on left side of GUI, in this first popup button is select image having three options of image first Cameraman second Rice and third Lena image. Then second popup button is of select training algorithm it have four option of function used for training algorithms. First option is traingd, which is function for gradient descendent algorithm. Second option is traingdm, which is function for gradient descendant with momentum algorithm. Third option trainlm is function used for Levenberg marquaurd algorithm, traincgp is function for conjugate gradient with polark and ribart update algorithm.

6. Comparison of training algorithms for lena image

In below Table 1 shows result of Lena image for different training algorithm. Image without Histogram equalization and without noise is used. After compression, compressed image is given for decompression without quantization. Result shows that traingd & traingdm algorithm functions are given very less PSNR; due to this they are not suitable for Image Compression. In all algorithms tainlm algorithm function gives highest PSNR so it is used for further operation on lena image.

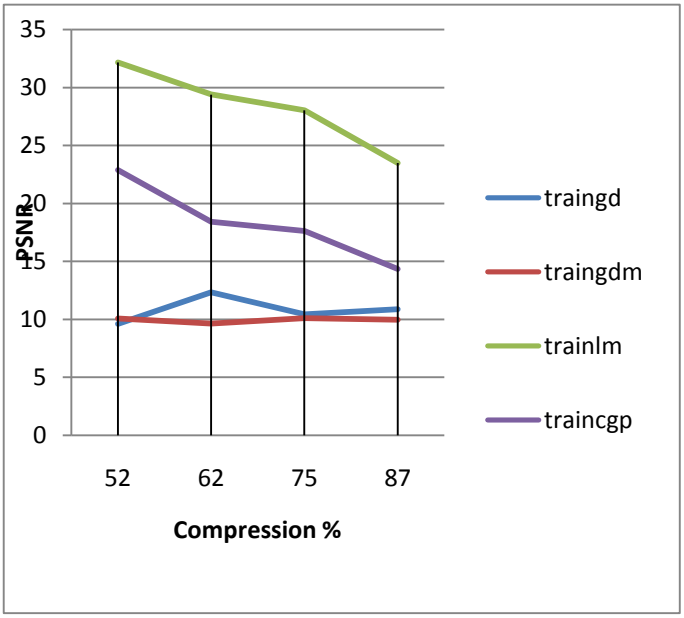
Table 1 Changing training algorithm effect on Lena image

DOI: 10.18535/ijecs/v5i2.6

Compression %	PSNR			
	Gradient Descendent Algorithm	Gradient Descendent with momentum Algorithm	Conjugate Gradient Algorithm	Levenberg Marquardt Algorithm
50	9.6102	10.070	22.888	33.8077
62	12.347	9.6542	18.428	29.400
75	10.445	10.101	17.627	28.032
87	10.877	9.9794	14.362	23.490

Figure 2 shows above result in graphical format.

Figure 2 Graph of Compression % vs. PSNR of Lena image for all algorithms



7. Effect of number of training epoch on lena image

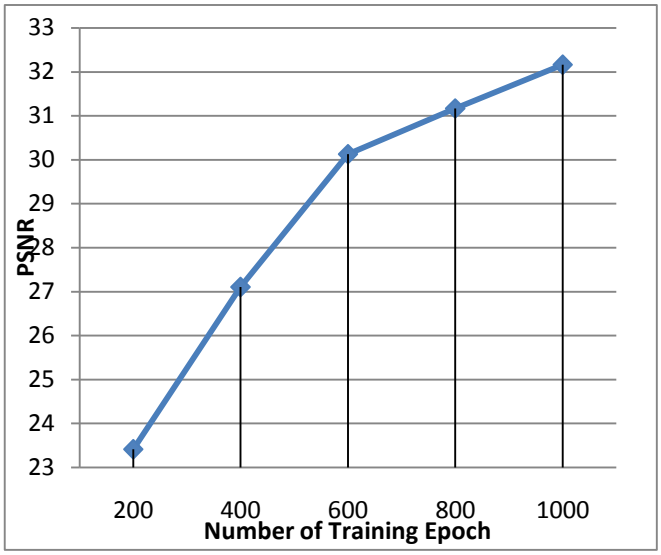
As shown in Table 2 by increasing number of epoch PSNR goes on increasing but time required to train image also goes on increasing for rice image.

Table 2 Changing number of epoch effect on Lena image

Algorithm Function	Compression %	No. of Epoch	PSNR
trainlm	50	200	23.4095
trainlm	50	400	27.1016
trainlm	50	600	30.127
trainlm	50	800	31.1656
trainlm	50	1000	32.16

Figure 3 shows graph which gives PSNR of decompressed image for different epoch number for Lena image.

Figure 3 Graph of Number of Training Epochs Vs PSNR of Lena Image



8. Conclusion

In this paper, Various algorithms were used for training finite element neural network The behavior of the methods for Image Compression has been studied, compared with other algorithms and LM method, showing that very good result of Lena image for different training algorithms. Image without Histogram equalization and without noise is used. After compression, compressed image is given for decompression without quantization. Result shows that Gradient descendent & gradient descendent with momentum algorithm functions are given very less PSNR; due to this they are not suitable for Image Compression. In all algorithms Levenberg Marquardt algorithm function gives highest PSNR. And also Shows the effect of different training Epochs is observed using LM algorithm is when higher value of epoch gives highest PSNR.

REFERNCES

[1] B.Arunapriya, D.KavithaDevi, "Improved Digital Image Compression using Modified Single Layer Linear Neural Networks" International Journal of Computer Applications (0975 – 8887) Volume 10– No.1, November 2010

[2] R. C. Gonzales, R. E. Woods, Digital Image Processing, Second Edition, Prentice-Hall, 2002.

[3] H. NaitCharif, Fathi M. Salam, "Neural Networks-based Image Compression System", 43rd IEEE Mildwest Symposium on Circuits and Systems, PP. no. 846-849, vol. 2, IEEE 2000.

[4] Rudy Setiono and Guojun Lu, "Image Compression Using a Feed forward Neural Network", IEEE World Congress on Computational Intelligence, PP. No. 4761-4765, Vol. 7, IEEE 1994.

[5] S. Anna Durai, and E. Anna Saro "Image Compression with Back-Propagation Neural Network using Cumulative Distribution Function", World Academy of Science, Engineering and Technology 17, 2006.

[6] DiptaPratimDutta, Samrat Deb Choudhury, Md. Anwar Hussain, "Digital Image Compression using Neural Networks", International Conference on Advances in Computing, Control, and Telecommunication Technologies, PP. No. 116-120, IEEE 2009.

[7] PremaKarthikeyan, Narayanan Sreekumar "A Study on Image Compression with Neural Networks Using Modified Levenberg Maruard Method" Global Journal of Computer Science and Technology, Volume 11 Issue Version 1.0, March 2011.

[8] N. Sonhara, M. Kawato, S. Miyake and K. Nakane, "Image Data Compression using a Neural Network Model", Proc. Inter. Joint Conference on Neural Networks,11-35-11-41, 1989.

[9] Howard Demuth, Mark Beale,MartinHagan"Neural Network Toolbox™ 6 user's guide" by The MathWorks, Inc 2009.

[10] Hagan, M.T., H.B. Demuth, and M.H. Beale, "Neural Network Design", Boston, MA: PWS Publishing, 1996.

[11] Hao Yu and B. M. Wilamowski, "Industrial Electronics Handbook, Intelligent Systems", 2nd Edition, chapter 12, pp. 12-1 to 12-15, CRC Press 2011.

[12] Y. Shantikumar Singh, B. Pushpa Devi,Kh. Manglem Singh "Image Compression using multilayer feed forward Artificial Neural Network with Conjugate Gradient".IEEE 2012.