Global Journal of Computer Science and Technology
Volume 11 Issue 3 Version 1.0 March 2011
Type: Double Blind Peer Reviewed International Research Journal
Publisher: Global Journals Inc. (USA)
Online ISSN: 0975-4172 & Print ISSN: 0975-4350

A Study on Image Compression with Neural Networks Using Modified Levenberg - Maruardt Method

By Prema Karthikeyan, Narayanan Sreekumar
Department of Computer Science

Abstract- In this paper, an adaptive method for image compression that is subjective on neural networks based on complexity level of the image. The multilayer perceptron artificial neural network uses the different Back-Propagation artificial neural networks in processing of the image. The original images taken, for instance 256*256 pixels of bitmap image, each block of image into one network selection, according to each block the value of pixels in image complexity value is calculated. To estimate each value of the images in a block can be evaluated and trained. Best PSNR in selecting images to be compressed with a modification Levenberg-Marquart for MLP neural network is taken. The algorithm taken a good research of result to each block of image. The taken time reduces the learning procedure for running each block of images. Finally, a neural network taken for the Back Propagation artificial neural network.

Keywords: *Image complexity, PSNR, Levenberg-Marquardt, Multi-layer neural network.*

Classification: *GJCST Classification: I.2.6, I.4.m*

A STUDY ON IMAGE COMPRESSION WITH NEURAL NETWORKS USING MODIFIED LEVENBERG METHOD

Strictly as per the compliance and regulations of:

A Study on Image Compression with Neural Networks Using Modified Levenberg-Marquardt Method

Prema Karthikeyan[1], Narayanan Sreekumar[2]

Abstract- In this paper, an adaptive method for image compression that is subjective on neural networks based on complexity level of the image. The multilayer perceptron artificial neural network uses the different Back-Propagation artificial neural networks in processing of the image. The original images taken, for instance 256*256 pixels of bitmap image, each block of image into one network selection, according to each block the value of pixels in image complexity value is calculated. To estimate each value of the images in a block can be evaluated and trained. Best PSNR in selecting images to be compressed with a modification Levenberg-Marquart for MLP neural network is taken. The algorithm taken a good research of result to each block of image. The taken time reduces the learning procedure for running each block of images. Finally, a neural network taken for the Back Propagation artificial neural network.

Keywords- Image complexity, PSNR, Levenberg-Marquardt, Multi-layer neural network.

I. Introduction

The compression of an image is very useful in many important areas such as data storage, communication, computation purpose and neural network purpose. The neural networks are being well developed in software computing process. Noise suppression, transform extraction, Parallelism and optimized approximations are some main reasons that useful to artificial neural network for image compression method. The activities of image compression on neural networks implemented in Multi-Layer Perceptron (MLP) [2-13], learning vector quantization (LVQ), [14], Self-Organizing Map(SOM),Learning Vector quantization (LVQ) [15,16]. From these network methods, the Back propagation neural network is used for MLP process. In artificial neural network (ANN) uses, Back-Propagation algorithm processed in image compression method [3].The experts used a three-layer BPNN method for compression. The image is used for compression, it is divided into blocks and taken to input neurons, the neurons of input are compressed are taken at output of the hidden layer and the de-compressed images are stored in the output of the hidden layer. This process was implemented in the NCUBE parallel computer and the simulation results produced from network taken a poor image quality in 4:1 compression ratio [3]. By using single network for compression of an image, the result produced from a single network one simple BPNN are poor one. The researches try to increase the performance of an image in neural-network based compression technique. The compress/decompress (CODEC) image blocks are used on various methods for different image blocks regarding to the complexity of blocks. The results produced from image compression are good with neural networks. The cluster of an image blocks into some basic classes based on a complexity measure called activity. The researchers used four BPNNs with different compression rates for each class with neural network. It produces more benefit improvement over basic BPNN. The adaptive approach with proposed the use of complexity measure with block orientation by six BPNNs has given better visual quality [11]. The BPNNs were used for compressing image blocks, after that each pixel in a block was subtracted from the mean value of the block. This method gives some Best-SNR method is used to select the network that gives the best SNR for the block of an image. The overlapping of image blocks in a particular area is used in order to reduce the chess-board effect in de-compressed image. The Best-SNR methods in PSNR produce the visual quality of reconstructed image compared to standard images in JPEG coding.

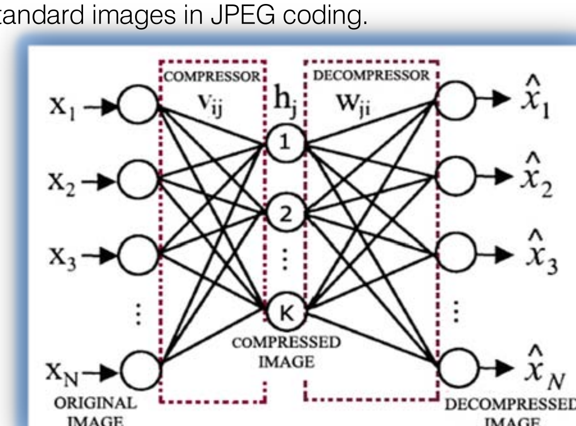


Fig.1-Basic image compression structure using neural network

About[1]-Asst.Prof, Department of Computer Science, SNR Sons college(Autonomous), Coimbatore, Tamil Nadu,India
E-Mail- premak11@yahoo.com
About[2]- Asso.Prof, Department of Computer Science, SNR Sons college(Autonomous), Coimbatore, Tamil Nadu,India
E-Mail- sreenkumar@yahoo.com

Global Journal of Computer Science and Technology Volume XI Issue III Version I March 2011

This paper is taken as follows. In section II we discuss multi-layer neural network for image compression. Section III describes the Modified-Levenberg method used in this paper. In section IV, the experimental results of our implementations are taken and discussed and finally in section V we conclude this research and give a summary on it.

II. Image Compression Used With Multi-Layer Neural Networks

The image compression used with Back-propagation algorithm in multi-layer neural network. The multi-layer neural network is given in Fig.1. It taken the network with three layers, input, hidden and output layer. Both the input and output layers have the same number of neurons, N. The input and output are connected to each network; the compression can be done with the value of the neurons at the hidden layer. In compression methods, the input image is divided into blocks, for example with 8×8 , 4× 4 or 16 ×16 pixels the block sizes of neurons in the input/output layers which convert to a column vector and fed to the input layer of network; one neuron per pixel. With this basic MLP neural network, compression is conducted in training and application phases as follow.

1) *Training*

In image compression, the image samples are used to train each network with the back propagation learning rule. In network, the output layer of network will be equal to the input pattern with each layer in a narrow channel. The normalized gray level range, training samples of blocks are converted into vectors. In compression and de-compression can be given in the following equations.

$$H_j^{in} = \sum_{i=1}^{N} V_{ij} X_i \, , h_f\left(H_j^{in}\right); 1 \le j \le K \qquad (1)$$

$$\hat{X}_j^{in} = \sum_{j=1}^{K} W_{ij} h_j \, , g\left(\hat{X}_j^{in}\right); 1 \le i \le N \qquad (2)$$

In the above equations, f and g are the activation functions which can be linear or nonlinear. ij V and ji W represent the weights of compressor and decompress or, respectively. The extracted N × K transform matrix in compressor and K × N inde-compressor of linear neural network are in PCA transform. It minimizes the mean square error between original and reconstructed image. The new spaces are decorrelated led to better compression. For data-dependent transform by using linear and nonlinear activation functions in this network results linear and non-linear PCA respectively. In training process of the neural network structure in Fig. 1 is iterative and stopped when the weights convert to their true values. In real applications the training is stopped when the error of equation (3) reaches to a threshold or maximum number of iterations limits the iterative process.

$$Err = \frac{1}{2\sum_{k=1}^{N}\left(\left[(X)\right]_k - \hat{X}_k)\right)^2} \qquad (3)$$

2) *Application*

When training process is completed and the coupling weights are corrected and the test image is fed into the network and compressed image is obtained in the outputs of hidden layer. The outputs must be applied to the correct number of bits. The same number of total bits is used to represent input and hidden neurons, and then the Compression Ratio (CR) will be the ratio of number of input to hidden neurons. For example, to compress an image block of 8×8, 64 input and output neurons are required. In this case, if the number of hidden neurons is 16 (i.e. block image of size 4× 4), the compression ratio would be 64:16=4:1. But for the same network, if 32 bits floating point is used for coding the compressed image, then the compression ratio will be 1:1, which indicates no compression has occurred. In general, the compression ratio of the basic network is illustrated in Fig (1) for an image with n blocks is computed as Eq. (4).

$$F(w) = e^T e \qquad (4)$$

Where w = [w1, w2… wN] consists of all weights of the network, e is the error vector comprising the error for all the training examples.

When training with the LM method, the increment of weights Δw can be obtained as follows:

$$\Delta w = \left[J^T J + \mu I\right]^{-1} J^T e \qquad (5)$$

Where J is the Jacobian matrix, μ is the learning rate which is to be updated using the β depending on the outcome. In particular, μ is multiplied by decay rate β (0<β<1) whenever F(w) decreases, whereas μ is divided by β whenever F(w) increases in a new step.

In de-compressor, the compressed image is converted to a version similar to original image by applying the hidden to output layer de-compression weights on outputs of hidden layer. The outputs of output neurons must be scaled back to the original grayscale range, i.e.[0~255] for 8 bit pixels.

3) *Adaptive Approach*

The neural network for image compression provides an value for PCA transform. The structure tries to implement the input samples of pixels in the network data compression. This is not used in many real applications. This is the main reason that PCA is replaced with its nearest approximate, the data-independent Discrete Cosine Transform (DCT) transform in real applications. One method for improving the performance of this simple structure is the adaptive approach which uses different networks to compress blocks of the image [2,5-11]. The networks have identical structure, but they have different number of neurons in hidden layers, which will result in different compression ratios.

Considering the network of Fig. 1 as the basic structure, we can present the adaptive method as in Fig. 2. In each block is estimated by means of a value to a complexity measure like average of the gray-levels in image block or some other methods. Then for complexity value, one of the available networks is selected and used by Back-propagation algorithm. The code should be transmitted or be saved along the compressed image. In de-compressor or transmitted code along with the compressed image is extracted from the corresponding network. In adaptive approach, the M different networks with k1 - kM neurons in hidden layer. The image with n blocks each having N pixels, the compression ratio is as equation (5) that is obtained by modifying equation (4).

III. Existing Levenberg-Marquardt Thods

The standard LM training process can be illustrated in the following pseudo-codes,

1. Initialize the weights and parameter μ0 (μ=.01 is appropriate).
2. Compute the sum of the squared errors over all inputs F(w) .
3. Solve (2) to obtain the increment of weights Δw
4. Recomputed the sum of squared errors F(w) Using w + Δw as the trial w, and judge

IF trial F(w) < F(w) in step 2 THEN
w = w + Δw
μ = μ · β (β = .1)
Go back to step 2

ELSE $\mu = \frac{\mu}{\beta}$

Go back to step 4
END IF

1) *Modification Of The LM Method*

To consider performance of index is F(w) = eT e using the Newton method.

STEP 1: J (w) is called the Jacobian matrix.

STEP 2: Next to find the Hessian matrix in k, j elements of the Hessian matrix.

STEP 3: The eigenvectors of G are the same as the eigenvectors of H, and the eigen values of G are (λi+μ).

STEP 4: The matrix G is positive definite by increasing μ until (λi+μ)>0 for all i therefore the matrix will be invertible it leads to Levenberg-Marquardt algorithm.

STEP 5: For learning parameter, μ is illustrator of steps of actual output movement to desired output. In the standard LM method, μ is a constant number.

This paper modifies LM method using μ as:

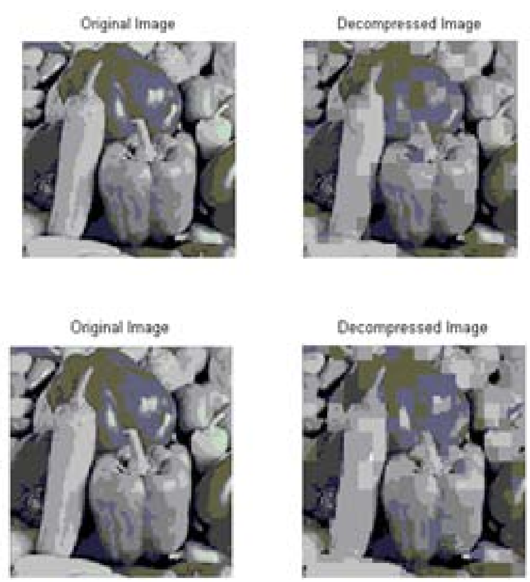
μ = 0.01eT e

Where e is a k ×1 matrix therefore eTe is a 1 × 1 therefore

[JTJ+μI] is invertible.

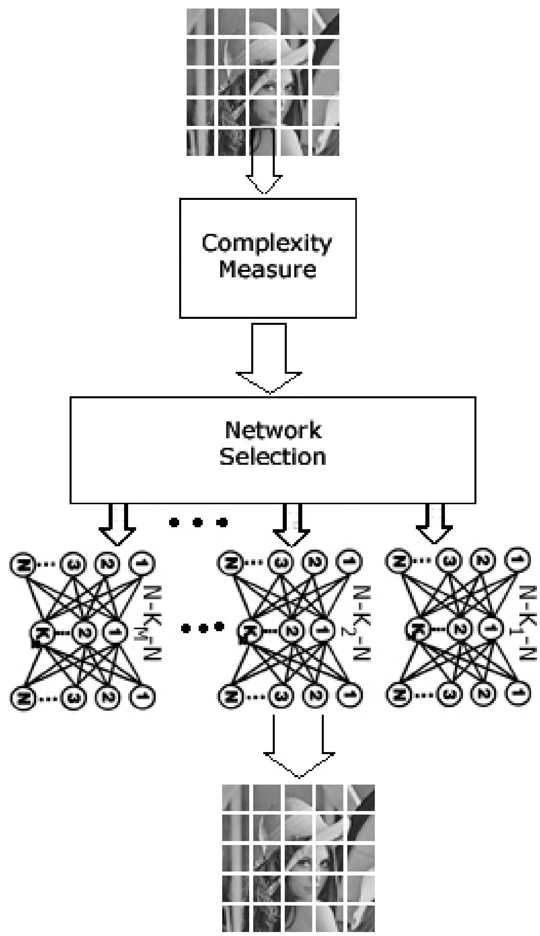
For actual output is taken for desired output or errors. The measurement of error is small then, actual output approaches to desired output with soft steps. Therefore error oscillation reduces.

IV. Results and Discussion



128 IMAGE SIZE	LEVENBERG-MARQUARDT METHOD			MODIFIED LEVENBERGMARQUARDT METHOD		
IMAGE	PSNR	MSE	TIME(SECONDS)	PSNR	MSE	TIME(SECONDS)
LENA	20.76	58.672	5136.215000	21.06	193.0357	4875.6325
PEPPER	12.8889	312.9567	909.765000	13.9682	252.5264	547.625000
BABOON	15.9707	201.3197	902.05320	16.9616	141.2765	546.140000
CROWD	8.6035	329.4677	921.844000	8.6449	141.2765	544.578000

256 IMAGE SIZE	LEVENBERG-MARQUARDT METHOD			MODIFIED LEVENBERGMARQUARDT METHOD		
IMAGE	PSNR	MSE	TIME(SECONDS)	PSNR	MSE	TIME(SECONDS)
LENA	21.7006	161.4895	3303.3698	22.3675	148.7677	2169.579000
PEPPER	15.0934	188.6425	3411.375000	15.3527	172.4312	2151.516000
BABOON	13.9517	195.6905	3614.437000	16.4312	123.0598	2376.734000
CROWD	14.3570	301.0073	4065.17200	15.7204	322.2830	2208.1410
BIRD	25.8375	54.5497	3112.781000	26.0312	55.6387	2056.3698



Neural network-based adaptive structure for image compression

V. Conclusion

A picture can say more than a thousand words. However, storing an image can cost more than a million words. This is not always a problem because now computers are capable enough to handle large amounts of data. However, it is often desirable to use the limited resources more efficiently. For instance, digital cameras often have a totally unsatisfactory amount of memory and the internet can be very slow. In these cases, the importance of the compression of image is greatly felt. The rapid increase in the range and use of electronic imaging justifies attention for systematic design of an image compression system and for providing the image quality needed in different applications. There are a lot of techniques available for image compression. Image compression using neural network technique is efficient when referring to the literature.

In this thesis the use of Multi-Layer Perceptron Neural Networks for image compression is reviewed. Since acceptable result is not resulted by compression with one network, an adaptive approach is used by changing the Training algorithm of the network with LM Method. It uses different networks for different image blocks regarding to their complexity values. The experimental results show that better visual quality is obtained by overlapping neighboring image blocks.

Also selecting images with Best-SNR criterion rather than the complexity criterion provides higher image quality and better PSNR. Higher number of networks provides better performance in Best-SNR approach but this will result in lower CR. However, overlapping and network selection need more investigations and it can be accepted to obtain better reconstructed image quality. Comparing results with basic BNN algorithm shows better performance for the proposed method both with PSNR measure and visibility quality.

References Références Referencias

1. R. C. Gonzales, R. E. Woods, Digital Image Processing, Second Edition, Prentice-Hall, 2002.
2. Veisi H., Jamzad M., Image Compression Using Neural Networks,Image Processing and Machine Vision Conference (MVIP), Tehran, Iran, 2005.
3. N.Sonehara, M.Kawato, S.Miyake, K.Nakane,Image compression using a neural network model, International Joint Conference on Neural Networks, Washington DC, 1989.
4. G.L. Sicuranza, G. Ramponi, S. Marsi, Artificial neural network for image compression, Electronic letters 26, 477-479, 1990.
5. S. Marsi, G. Ramponi, G. L. Sicuranza, Improved neural structure for image compression, Proceeding of International Conference on Acoustic Speech and Signal Processing, Torento, 1991.
6. S. Carrato, G. Ramponi, Improved structures based on neural networksfor image compression, IEEE Workshop on Neural Networks for Signal Processing, New Jersey, September 1991.
7. Rumelhart, D. E., Hinton, G. E. and Williams, R. J, Learning internal representations by error propagation," In Parallel Distributed Processing, Cambridge, MA: MIT Press, vol 1, pp. 318-362.
8. Rumelhart, D. E., Hinton, G. E. and Wiliams, R. J, "Learning representations by back-propagating errors," Nature, vol. 323, pp. 533-536, 1986.
9. Werbos, P. J. "Back-propagation: Past and future," Proceeding of International Conference on Neural Networks, San Diego, CA, 1, pp.343-354, 1988.
10. M. T .Hagan and M. B. Menhaj, "Training feed forward network with the Marquardt algorithm," IEEE Trans. on Neural Net., vol. 5, no. 6,pp.989-993, 1994.
11. Bello, M. G. "Enhanced training algorithms, and Integrated training/architecture selection for multi layer perceptron networks,"IEEE Trans. on Neural Net., vol. 3, pp. 864-875, 1992.

This page is intentionally left blank