Comparison Between Levenberg-Marquardt And Scaled Conjugate Gradient Training Algorithms For Image Compression Using MLP

Devesh Batra *devesh.batra.in@ieee.org*
Member, IEEE

Abstract

The Internet paved way for information sharing all over the world decades ago and its popularity for distribution of data has spread like a wildfire ever since. Data in the form of images, sounds, animations and videos is gaining users' preference in comparison to plain text all across the globe. Despite unprecedented progress in the fields of data storage, computing speed and data transmission speed, the demands of available data and its size (due to the increase in both, quality and quantity) continue to overpower the supply of resources. One of the reasons for this may be how the uncompressed data is compressed in order to send it across the network. This paper compares the two most widely used training algorithms for multilayer perceptron (MLP) image compression – the Levenberg-Marquardt algorithm and the Scaled Conjugate Gradient algorithm. We test the performance of the two training algorithms by compressing the standard test image (Lena or Lenna) in terms of accuracy and speed. Based on our results, we conclude that both algorithms were comparable in terms of speed and accuracy. However, the Levenberg-Marquardt algorithm has shown slightly better performance in terms of accuracy (as found in the average training accuracy and mean squared error), whereas the Scaled Conjugate Gradient algorithm faired better in terms of speed (as found in the average training iteration) on a simple MLP structure (2 hidden layers).

Keywords: Image Compression, Artificial Neural Network, Multilayer Perceptron, Training, Levenberg-Marquardt, Scaled Conjugate Gradient, Complexity.

1. INTRODUCTION

Image Compression algorithms have received notable consideration in the past few years because of the growing multimedia content on the World Wide Web. Image Compression is a must since despite advances in computer and communication technologies, the digital images and videos are still demanding in terms of storage space and bandwidth [1].

In this paper, we present an evaluation of two popular training algorithms (Levenberg-Marquardt and Scaled Conjugate Gradient) for image compression using simple Multilayer Perceptron (MLP) classifier.

Various parameters such as the gradient, mu and validation checks are evaluated for both the algorithms to examine their performance in terms of accuracy and speed. Image Compression refers to the reduction of irrelevant and redundant image data in order to store and transfer data in an efficient manner. Image compression can be classified as lossy and lossless. Lossless image compression allows original image to be perfectly reconstructed from the image data without any loss [2]. It is generally used in medical imaging, technical drawings and other areas where the minute details of the images are required and data loss could be fatal. On the contrary, in lossy image compression, the images can be only partially reconstructed from the image data [3]. Even though some of the data is lost, this is usually advantageous because it gives improved compression rates and hence smaller sized images.

The paper is organized as follows: Some previous works on the Image Compression are presented in Section II. The theoretical background to the proposed approach is presented in Section III. The methodology of the experiment is presented in Section IV, followed by the results and discussions in Section V. Section VI presents the conclusions of the findings in this paper and finally Section VII present proposed future work in the field, followed by references in Section VIII.

2. RELATED WORK

There is a lot of research in literature that focuses on image compression using various classifiers and algorithms.

In [4] (2006), time taken for simulation has been reduced by 50% by estimating a Cumulative Distribution Function (CDF) and using it to map the image pixels.

In [5] (2013), a new approach for near–lossless compression of the medical images is proposed. Pre–processing techniques are applied to the input image to generate a visually quantized image. The visually quantized image is encoded using a low complexity block–based lossless differential pulse code modulation coder, followed by the Huffman entropy encoder. Results show the superiority of the proposed technique in terms of the bit rate and visual quality.

In [6], a comparison of Principal Component Analysis (PCA) is presented for still image compression and coding. The paper presents comparison about structures, learning algorithms and required computational efforts along with a discussion of advantages and drawbacks related to each technique. The wide comparison among eight principle component networks shows that cascade recursive least squares algorithm by Ci-chocki, Kasprzak and Skarbek exhibits the best numerical and structural properties.

[7] presents a comparison between Levenberg Marquardt (LM) and Scaled Conjugate Gradient (SCG) algorithms for Multilayer Perceptron diagnosis of Breast Cancer Tissues. The study concludes that both algorithms were comparable in terms of accuracy and speed. However, the LM algorithm showed better advantage in terms of accuracy and speed on the best MLP structure (with 10 hidden units).

[8] presents an overview of neural networks as signal processing tools for image compression model. The self-organizing feature map (SOFM) has been used in the design of codebooks for vector quantization (VQ). The resulting codebooks are shown to be less sensitive to initial conditions than the standard LBG algorithm.

3. THEORETICAL BACKGROUND

3.1 Artificial Neural Networks and MLP

ANNs can be defined in many ways. At one extreme, the answer could be that neural networks are simply a class of mathematical algorithms, since a network can be regarded essentially as a graphic notation for a large class of algorithms. Such algorithms produce solutions to a number of specific problems. At the other end, the reply may be that these are synthetic networks that emulate the biological neural networks found in living organisms [9].

Although computers outperform both biological and artificial neural systems for tasks based on precise and fast arithmetic operations, artificial neural systems represent the promising new generation of information processing networks. Neural networks can supplement the enormous processing power of the Von Neumann digital computer with the ability to make sensible decisions and to lean by ordinary experience, as we do [9].

The signal flow of neuron inputs, xi, is considered to be unidirectional as indicated by arrows, as is a neuron's output signal flow. This symbolic representation shows a set of weights and the neuron's processing unit, or node. The neuron output signal is given by the following relationship:

O = f (wtx),

where w is the weight vector defined as

W = [w1 w2 … wn] t

and x is the input vector:

X = [x1 x2 … xn] t

All vectors defined are column vectors; superscript t denotes a transposition. The function f (wtx) is referred to as an activation function. The activation functions of a neuron can be bipolar continuous or unipolar continuous as shown in figure 1 and 2 respectively.

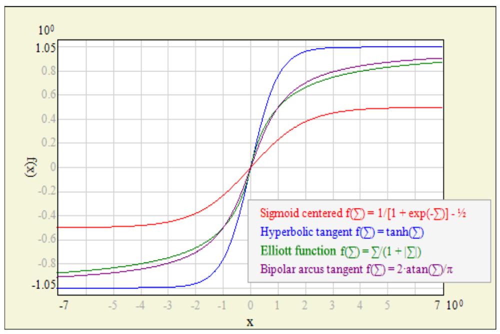

FIGURE 1: Bipolar Activation Function.

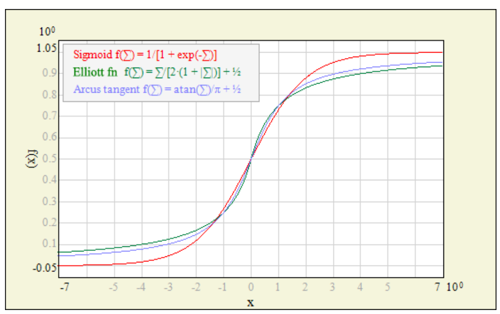

FIGURE 2: Unipolar Continuous activation function.

A feed-forward neural network is a biologically inspired classification algorithm. It consists of a (possibly large) number of simple neuron-like processing units, organized in layers. Every unit in a layer is connected with all the units in the previous layer. These connections are not all equal; each connection may have a different strength or weight. The weights on these connections encode the knowledge of a network. Often the units in a neural network are also called nodes.

Data enters at the inputs and passes through the network, layer by layer, until it arrives at the outputs. During normal operation, that is when it acts as a classifier, there is no feedback between layers. This is why they are called feed-forward neural networks [10].

Figure 3 is a 2-layered network with, from top to bottom: an output layer with 5 units, a hidden layer with 4 units, respectively. The network has 3 input units.

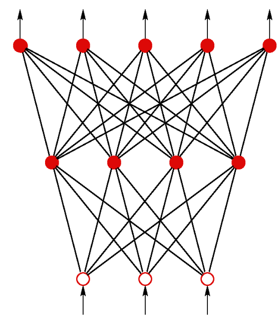

FIGURE 3: 2-Layered network.

Multi Layer perceptron (MLP) is a feed-forward neural network with one or more layers between input and output layer as shown in figure 4. This type of network is trained with the back-propagation learning algorithm. MLPs are widely used for pattern classification, recognition, prediction and approximation. Multi Layer Perceptron can solve problems, which are not linearly separable [11].

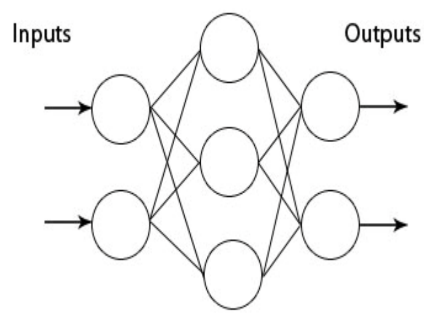

FIGURE 4: Multilayer Perceptron.

3.2 The Levenberg Marquardt Algorithm

Levenberg-Marquardt algorithm, which was independently developed by Kenneth Levenberg and Donald Marquardt, provides a numerical solution to the problem of minimizing a nonlinear function [12]. It is fast and has stable convergence. In the artificial neural network field this algorithm is suitable for small- and medium-sized problems.

Levenberg-Marquardt algorithm introduces an approximation to Hessian matrix; in order to ensure that the approximated Hessian matrix JTJ is invertible.

The approximation introduced is:

$$H = JtJ+uI$$

where, u is always positive, called combination coefficient and I is the identity matrix.
The elements on the main diagonal of the approximated Hessian matrix will be larger than zero. Therefore with this approximation, it can be sure that the matrix H is always invertible [13].
The update rule of Levenberg-Marquardt algorithm can be presented as:

$$W_{k+1} = W_k - (J_k^t J_k + uI)^{-1} J_k e_k$$

4. METHODOLOGY

The primary components of this work are training the multilayer perceptron for image compression and comparison of results obtained from the two training algorithms used. The multilayer perceptron training algorithms consist of Levenberg Marquardt and Scaled Conjugate Gradient algorithms. The results obtained are compared on the basis of various parameters such as speed (as observed in the average training iteration) and accuracy (as observed in terms of average training accuracy and mean squared error).

4.1 Image Dataset Description

We test the performance of the two training algorithms by compressing the standard test image, Lena (figure 5).

FIGURE 5: Standard Test Image: Lena.

The image properties are as follows:

Properties	Value
Pixel Dimensions	512 X 512 pixels
Print Size	5.33 X 5.33 inches
Resolution	96 X 96 DPI
Colour Space	RGB
File Size	768.1 KB
File Type	TIFF

TABLE 1: Image Properties.

4.2 Multilayer Perceptron and Structure

A multilayer feed-forward network is used. The most important characteristic of a multilayer feed-forward network is that it can learn a mapping of any complexity [9]. The network learning is based on repeated presentations of the training samples. The trained network often produces surprising results and generalizations in applications where explicit derivation of mappings and discovery of relationships is almost impossible. In the case of layered network training, the mapping error can be propagated into hidden layers so that the output error information passes backward. This mechanism of backward error transmission is used to modify the synaptic weights of internal and input layers.

Transfer function, is a process defining relationship between the input and the output. The transfer function of a neuron is chosen to have a number of properties, which either enhance or simplify the network containing the neuron. A non-linear function is necessary to gain the advantage of a multi-layer network.

4.3 Levenberg Marquardt and Scaled Conjugate Gradient Training Algorithm Parameters

The default values of various parameters used in MATLAB for Multilayer Perceptron training. The parameters and their default values used for Levenberg-Marquardt and Scaled Conjugate Gradient training algorithms are as follows:

Parameters	**Value**
Maximum Epochs	1000
Training Goal	0
Minimum Gradient	1.00×10^{-10}
α	0.10
β	10

TABLE 2: Default Values of parameters used in MATLAB for MLP Training.

Detailed description of the training procedure used:

During the training procedure the input image dataset is encoded into a structure of hidden and output weight matrices. The image used for training purposes is assumed to be of dimension R X C and consists of r x c blocks. The following steps are followed during the training procedure:

1. The block matrix is converted into a matrix X of size P x N containing training vectors, where, x(n), is formed from image blocks. Mathematically, it can be expressed as follows:

   $$P = r.c \text{ and } p.N = R.C$$

2. The target data is made equal to the data, that is: D=X.
3. The network is then trained until the mean squared error, MSE, is sufficiently small. The matrices W^h and W^y are subsequently used in the image encoding and decoding steps.

IMAGE ENCODING:

The hidden-half of a neural network is used to encode images. The encoding procedure is described as follows:

$$F \text{ -> } X,\ H = (W^h.X)$$

where X is the encoded image of F.

IMAGE DECODING:

The reconstruction of encoded image is known as decoding. It is done using the output half of the neural network. The decoding procedure is as follows:

$$Y = (W^y.H),\ Y \text{ -> } F$$

ALGORITHM:
STEP 1: Input the image to be tested
STEP 2: The input image is divided into block of pixels
STEP 3: Each block is scanned for complexity level
STEP 4: The neurons are initialized
STEP 5: Scanned vectors are applied to each neuron on the input layer
STEP 6: Operations are performed depending upon the weights assigned and logic involved (TRANSIG)
STEP 7: They are then passed to the hidden layer
STEP 8: Repeat STEP6 (PURELIN)
STEP 9: The outputs are reassembled
STEP 10: The neural network is trained and the weights are retained.

5. RESULTS AND DISCUSSIONS

The results yielded on the comparison of Levenberg Marquardt Algorithm and Scaled Conjugate Algorithm for image compression will be discussed in this section. The conditions under which the comparison was done have already been discussed in Section IV.

The image obtained on compression of image 'Lena' (figure 5) with both the algorithms was of same quality and has been shown in figure 6.

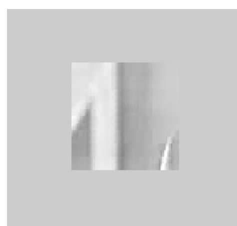

FIGURE 6: Compressed Image.

As shown in table III, Levenberg Marquardt algorithm took 53 seconds for compressing the image and running a cycle of 1000 epochs whereas Scaled Conjugate Gradient algorithm took mere 11 seconds for the same. Hence, the Levenberg Marquardt was relatively slow in processing the image in comparison to the Scaled Conjugate Gradient.

Levenberg Marquardt	Scaled Conjugate Gradient
53 seconds	11 seconds

TABLE 3: Time taken by both Algorithms.

A test was also conducted for the usage of RAM and CPU while both the algorithms were processing the image. Both the algorithms used almost equal amount of RAM and CPU while executing as shown in table IV. The CPU used for performing the experiment is Intel Core i5 – 2430M CPU@2.4 GHz (64-bit). The RAM used is 4 GB.

As indicated in the table, ideal state is a state of the system in which the execution is not being performed, i.e., Windows Task Manager is the only program running and no applications are running in the background. As for the processes and services, only basic Operating System processes (Windows 7 Professional) like Task Manager, Windows AutoUpdater and Windows Explorer.

State	Levenberg Marquardt		Scaled Conjugate Gradient	
	CPU	**RAM**	**CPU**	**RAM**
Ideal State	05%	43%	02%	47%
At t = 0	17%	51%	16%	52%
During Execution	58-67%	48%	55%	47%

TABLE 4: Usage of RAM AND CPU during execution of both Algorithms.

As shown in the comparison graphs of Mean Squared Error (MSE) of both Levenberg Marquardt and Scaled Conjugate Gradient (figure 7 and 8 respectively), the best results were obtained at different epochs. In both the cases it can be observed that the MSE stabilizes after certain number of epochs.

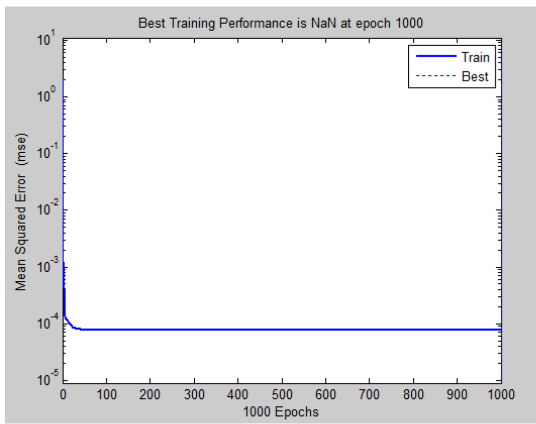

FIGURE 7: Mean Squared Error for Levenberg Marquardt.

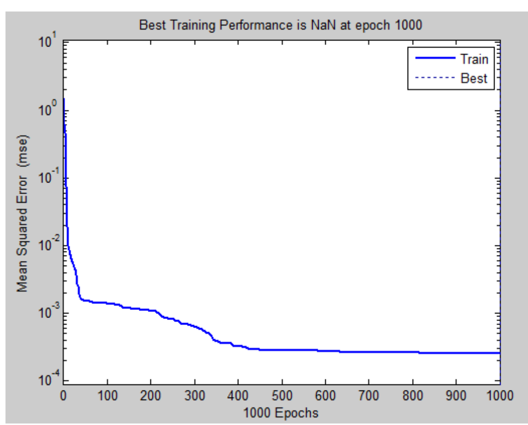

FIGURE 8: Mean Squared Error for Scaled Conjugate Gradient.

As shown in the comparison graphs of gradients of both Levenberg Marquardt and Scaled Conjugate Gradient (figure 9 and 10 respectively), it can be observed that the performance function at each epoch is different for both the cases. As the gradient becomes smaller and closer to zero, the function will be minimized. This implies that the outputs are very close to the targets and hence the network is trained.

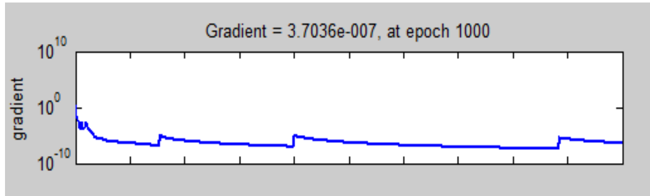

FIGURE 9: Gradient for Levenberg Marquardt.

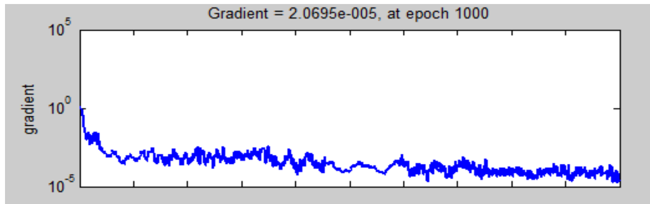

FIGURE 10: Gradient for Scaled Conjugate Gradient.

6. CONCLUSION

In this paper, we have compared the two most widely used training algorithms for multilayer perceptron (MLP) image compression - the Levenberg-Marquardt and the Scaled Conjugate Gradient algorithm. The performances of these two algorithms were tested by compressing the standard test image (Lena or Lenna) in terms of accuracy and speed. Based on our results, it was observed that both the algorithms were comparable in terms of speed and accuracy. However on the basis of Mean Squared Error (MSE) vs. epochs graph it was observed that the Levenberg-Marquardt had better accuracy as the MSE stabilized earlier in case of Levenberg-Marquardt algorithm as compared to that in the case of Scaled Conjugate Gradient algorithm. On the other hand, the Scaled Conjugate Gradient algorithm faired better in terms of speed (as found in average training iteration) on a simple MLP structure (2 hidden layers).

The paper provides results that are of utmost importance to the industry since the said comparison helps the Computer Scientists in analysing the difference between the two algorithms in minute details. Hence, they can judge, based on the comparison shown in the paper, which algorithm they want to use in transmitting images over the network. If they want the images sent over the network to be reliable, without any due consideration to time, then this paper suggests them to choose Levenberg-Marquardt algorithm over the Scaled-Conjugate algorithm. Scientists involved in complex research involving image analysis, who need the accuracy of the image to be extremely high, would generally encounter this type of a scenario. However, if they want the compression of the images to be fast, such as image sharing applications and services for general public, they can easily opt for the Scaled Conjugate algorithm.

7. FUTURE RESEARCH

Now that we have successfully compared the two most widely used training algorithms for multilayer perceptron (MLP) image, the practical implementation of these two algorithms as per the need can be done easily. Post this analysis; Levenberg-Marquardt algorithm is now ready to be used for reliable and high quality transportation of images over the networks with high bandwidths, especially in the scenarios where the focus is on transfer of more reliable images rather than the speed with which the images need to be compressed. On the other hand, the Scaled Conjugate algorithm can be used for a comparatively less accurate but faster transmission of the said images.

With this, we understand that there is a future for the application and comparison of these algorithms on animations and videos – entities that are combination of images. The tricky part in the comparison of these algorithms would be that videos and animations are composed of various other elements apart from images, such as text and sound, and similarly, their transfer over the networks is dependent on various other parameters like “frames per second” in a video, communication technique used in the networks, etc. Thus, if due consideration is given to all the elements, a reliable comparison of the modified algorithms can be obtained.

8. REFERENCES

[1] Ebrahimi, Farzad, Matthieu Chamik, and Stefan Winkler. "JPEG vs. JPEG 2000: an objective comparison of image encoding quality." Optical Science and Technology, the SPIE 49th Annual Meeting. International Society for Optics and Photonics, 2004.

[2] What is lossless image compression. Available:http://dvd-hq.info/data_compression_1.php.

[3] What is lossy image compression. Available: http://dvd-hq.info/data_compression_2.php.

[4] Durai, S. Anna, and E. Anna Saro. "Image Compression with Back-Propagation Neural Network using Cumulative Distribution Function." International Journal of Applied Science, Engineering & Technology 3.4 (2007).

[5] Cyriac, Marykutty, and C. Chellamuthu. "A near–lossless approach for medical image compression using visual quantisation and block–based DPCM."International Journal of Biomedical Engineering and Technology 13.1 (2013): 17-29.

[6] Costa, Saverio, and Simone Fiori. "Image compression using principal component neural networks." Image and vision computing 19.9 (2001): 649-668.

[7] Mohamad, N., et al. "Comparison between Levenberg-Marquardt and scaled conjugate gradient training algorithms for breast cancer diagnosis using MLP."Signal Processing and Its Applications (CSPA), 2010 6th International Colloquium on. IEEE, 2010.

[8] Dony, Robert D., and Simon Haykin. "Neural network approaches to image compression." *Proceedings of the IEEE* 83.2 (1995): 288-303.

[9] Zurada, Jacek M. "Introduction to artificial neural systems." (1992).

[10] What is feedforward neural network. Available: http://www.fon.hum.uva.nl/praat/manual/Feedforward_neural_networks_1__What_is_a_feedforward_ne.html

[11] Multilayer Perceptron. Available: http://neuroph.sourceforge.net/tutorials/MultiLayerPerceptron.html

[12] Yu, Hao, and B. M. Wilamowski. "Levenberg-marquardt training." The Industrial Electronics Handbook 5 (2011): 1-15.

[13] Møller, Martin Fodslette. "A scaled conjugate gradient algorithm for fast supervised learning." Neural networks 6.4 (1993): 525-533.

[14] MATLAB Product Help. Available: http://www.mathworks.in/help/matlab/.

[15] Wei, Wei-Yi. "An Introduction to Image Compression." National Taiwan University, Taipei, Taiwan (2009): 1.

[16] Anderson, Dave, and George McNeill. "Artificial neural networks technology."*Kaman Sciences Corporation* 258 (1992): 13502-462.

[17] Nguyen, Derrick, and Bernard Widrow. "Improving the learning speed of 2-layer neural networks by choosing initial values of the adaptive weights." *Neural Networks, 1990., 1990 IJCNN International Joint Conference on*. IEEE, 1990.

[18] Ranganathan, Ananth. "The levenberg-marquardt algorithm." *Tutoral on LM Algorithm* (2004): 1-5.

[19] Andrei, Neculai. "Scaled conjugate gradient algorithms for unconstrained optimization." *Computational Optimization and Applications* 38.3 (2007): 401-416.

[20] Steihaug, Trond. "The conjugate gradient method and trust regions in large scale optimization." *SIAM Journal on Numerical Analysis* 20.3 (1983): 626-637.

[21] Riedmiller, Martin. "Advanced supervised learning in multi-layer perceptrons—from backpropagation to adaptive learning algorithms." *Computer Standards & Interfaces* 16.3 (1994): 265-278.