Grayscale Image Compression Using Backpropagation Neural Network

Dubravka Ilic
Technical faculty "Mihajlo Pupin" Zrenjanin
University in Novi Sad
Djure Djakovica bb, 23000 Zrenjanin
Serbia and Montenegro
dudai@ptt.yu

Ivana Berkovic
Technical faculty "Mihajlo Pupin" Zrenjanin
University in Novi Sad
Djure Djakovica bb, 23000 Zrenjanin
Serbia and Montenegro
berki@beotel.yu

Abstract: **In this paper we presented the algorithm for implementation of digital grayscale image compression using backpropagation neural networks with 64 input neurons, 4, 8 or 16 neurons in hidden layer that are determining compression rate and 64 output neurons. The network of given architecture can be dual depending on used activation function in its hidden and output layer. We applied different learning algorithms on both types of networks and analyzed their performance results. The dependence between the quality of the decompressed image and the training set is shown using the so-called GIBO (Good In Bad Out) method of forming the training set with which the neural network learning the image compression has been trained. For each compressed image attained by certain learning algorithm we presented its objective and subjective quality measure.**

I. INTRODUCTION

The compression of digital image has become a crucial problem today. Internet teleconferences, satellite communications and storing of digital images wouldn't be possible without high compression rate.

Concerning the fact that image compression is the field that attracts huge attention, as the result of numerous researches, some of which are still going on, numerous compression techniques are introduced. Among them, artificial neural networks (NN) have also found their place, first of all thanks to their characteristic of successful handling the large amount of data that can be incomplete or that can be characterized by the presence of noise, but also thanks to their crutial existential feature – learning possibilities [3].

II. THE MAIN IDEA OF USING NN FOR IMAGE COMPRESSION

The main idea of neural networks application for the problem of image compression is to construct such a network in which the input and output layer with the same number of neurons will be the connection to the middle (hidden) layer with smaller number of neurons – which is the precondition for compression realization.

The relation input/hidden layer size is the compression rate. The goal of such compression is to reconstruct its input; hence, the output of the network that solves this class of problems is equal to its input. Network training starts with the initialization of its weights and requires a set of examples i.e. appropriate input-output pairs. Weights in the network are iteratively modified during the training time to minimize the performation function of the given network, which is mostly MSE (Mean Square Error between the calculated and expected network output). The procedure is repeating till satisfied level of deviation between calculated and expected output is achieved. This way of compression results according to [2] in the compression with loss: the original image cannot be reconstructed completely – there is no answer yet to the question whether this would be theoretically possible, but it is clear: that would demand extremely large training time.

III. STEPS IN REALIZATION OF NN MEANT FOR IMAGE COMPRESSION

Th network described in this paper is implemeted in software package MatLab following the next steps:

A. Defining the Network

The very first step in realizing the given task is to define the network architecture shown in the Fig. 1.

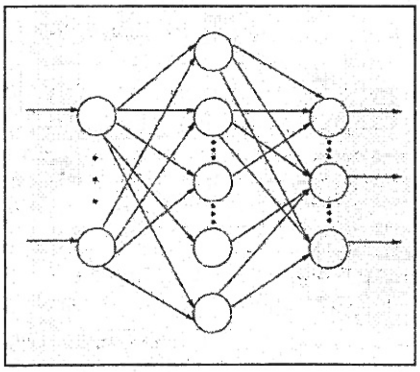


Fig. 1. Typical neural network architecture

Decision about the number of neurons in input and output layer is connected to the problem that should be solved through this network. Considering the goal of this network i.e. image compression, that should be the guide when its architecture is going to be defined. Constructed network will compress the grayscale image represented with the matrix which elements are values of pixels from the interval [0,255]. Blocks of size 8x8 pixels are extracted from the image, like shown in the Fig. 2, which means that values of 64 pixels used in network training are presented to the network input layer.

Because of that, input layer is constructed in such a way that it has 64 neurons into which inputs are fed and the same number of neurons has to be in the output layer. Decision about the number of neurons in the hidden layer is free (yet, it has to be less then the number of neurons in input layer) and is 4, 8 and 16 and has direct influence on the compression rate.

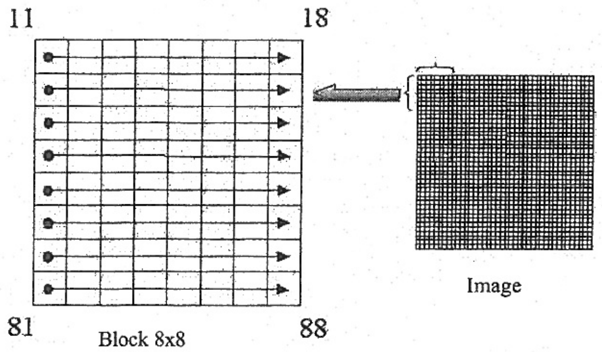


Fig. 2. Extracted blocks of images

B. *Image preprocessing*

This is the step in which input values for network training are created and it implies:

- image division into 8x8 pixels blocks;
- normalization of each block, i.e. scalarization of each block from integer values from the interval [0,255] to real numbers from the interval [0, 1]; normalization is achieved through the division of pixel value with its biggest possible value (255). The reason for using the normalization lies in the fact that neural network operates much more effective when its input and output values are limited to the interval [0, 1];
- linearization of each block using one of the known methods (scanning rows, columns or labyrinth) that achieves transformation of 2-dimensional blocks into 1-dimensional vectors, which represent inputs into neural network.

C. *Preparation of training pairs*

Covers presenting the normalized and linearized input vectors (image blocks) and expected output vectors to the input neuorns of the neural network; to achieve compression it is necessary that expected output vectors are equal to input vectors.

D. *Network training*

Network is trained using one of the chosen learning algorithms; training considers network weights modification. When the network is trained i.e. when the network achieves its goal (calculated and expected error minimization) in a certain number of iterations, its weight matrices have to be remembered. Thanks to these stored parameters, it is possible to present to the network a completely new image (that also has to be preprocessed) and to apply algorithm of compression and decompression.

E. *Compression*

New inputs and expected outputs (image that is going to be compressed) are fed into the network input layer. Using previously stored weights, hidden layer output is calculated and then attained values are quantified with 8 bits and remembered. Stored data is compressed image and as an additional component 16-bits data is also remembered: image dimensions in pixels, needed for its successful decompression.

F. *Decompression*

To reconstruct compressed image it is necessary to read the stored data (compressed image) and to set these read values as the hidden layer outputs, and then, again using the stored weight matrices, calculate network output, operating only with the input to the output layer (stored data) and weights between hidden and output layer. Using in this way attained network outputs, reconstruted image is created in the procces of postprocessing.

IV. NETWORK LEARNING ALGORITHM

The network is trained using backpropagation algorithm, i.e. its modifications. These variations of the primary algorithm are based on others, more sophisticated optimization techniques, but they all generally use gradient of performation function in order to identify the way how to modify weights and to minimize the function (Fig. 3).

For the weigths w in the network according to the gradient rule [1] new value is calculated following the expression:

$$w_i(new) = w_i(old) + \alpha(Y - O)x_i, \qquad (1)$$

where:

a – constant that maintain the learning rate
x_i – input of the neuron *i*
Y – calculated output from the neuron
O – expected output from the neuron.

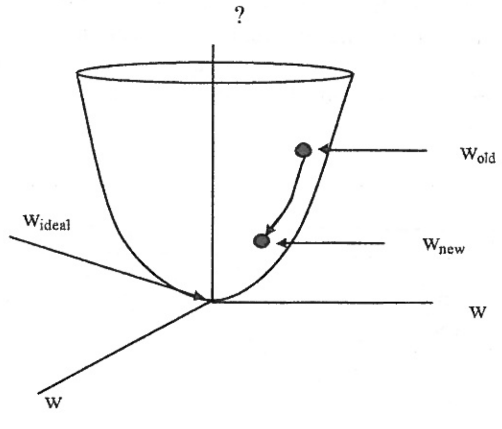


Fig. 3. Performation function

Used training algorithms are:

- gradient descent with momentum (GDM);
- backpropagation algorithm with variable learning rate (GDX);
- one step secant algorithm (OSS);
- resilient backpropagation (RP);
- scaled conjugate gradient (SCG).

For all algorithms with which the network is trained following parameters are recorded:

- network training time;
- performation function (MSE);
- number of epochs in which the training is finished;
- objective and subjective reconstructed image quality measure.

As the parameters that determine the duration of the network training i.e. the condition when to finish the training, following elements are introduced:

- maximal allowed network training time;
- maximal allowed number of epochs;
- set precision (this refers to the performation function).

The network is for each learning algorithm trained with the same picture and then, after the finished training, performances of the algorithm are checked through its generalizations. Pictures that have never been presented to the network previously are fed into the network input, with the presupposition that the network has learned the task for which it has been trained.

Experimental pictures are chosen in such a way to have different dimensions, resolutions and contents. First, network with 16, then 8 and finally 4 neurons in hidden layer are trained; achieved compression rates are respectively: 4, 8 and 16.

V. COMPRESSION RESULTS AND COMPARATIVE ANALYSIS OF USED LEARNING ALGORITHMS

The best results are achieved with the networks trained with the SCG algorithm.

The results of all applied training algorithms with the compression rate 4 are given in Tables I and II (the results in the first table are achieved with the network that has sigmoidal activation function in the hidden and output layer, so called LL network, and these results are sorted according to the network perfomation function; the second table shows the results of the network with sigmoidal activation function – LP network, also sorted according to the value of MSE).

TABLE I

Algorithm	Epoch	Time	MSE	PSNR/Measure
SCG	1000	0.00114	0.0016	76,08/4
RP	1000	0.00057	0.0022	74,70/4
OSS	512	0.01002	0.0055	70,72/3
GDX	1000	0.00128	0.0079	69,15/2
GDM	1000	0.00058	0.1877	55,39/1

TABLE II

Algorithm	Epoch	Time	MSE	PSNR/Measure
RP	1000	0.00048	0.0015	76,36/4
SCG	1000	0.00392	0.0016	76,08/4
GDX	1000	0.00556	0.0045	71,59/3
OSS	448	0.01000	0.0083	68,94/2
GDM	1000	0.00184	0.3557	52,61/1

The results of the compression with the rate 8 and 16 are analog to the results for the rate 4.

As it can be seen from the table, the best performance according to the learning time gives the RPROP algorithm, and then GDM, SCG and GDX. The slowest training algorithm is OSS algorithm. According to the performation function the best results give SCG and RPROP algorithms.

As the measure of objective quality measure of the reconstructed image, signal/noise relation is used (PSNR - Peak Signal to Noise Ratio) given in the expression:

$$PSNR = 20\log_{10}\frac{2^n-1}{MSE} = 10\log_{10}\frac{(2^n-1)^2}{MSE}, \qquad (2)$$

where n is the number of bits needed for display the values of the elements of the original image. As the value of the image element used here can be described with $n=8$ *bits* the next is following (2):

$$\mathrm{PSNR} = 10\log_{10}\frac{255^2}{\mathrm{MSE}}, \qquad (3)$$

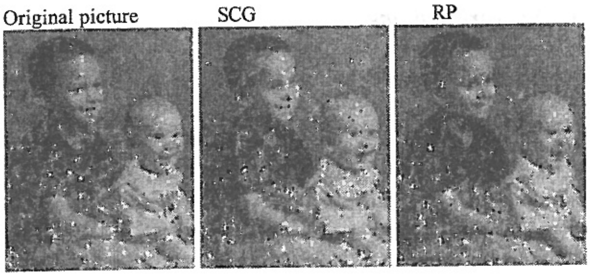
PSNR can be significantly different then the subjective measure of the image, depending on the error distribution in the reconstructed image. If the big error is concentrated in the small area of the image, subjective measure is very bad and PSNR gives a good result. Hence, for any reconstructed image is necessary to give also its subjective measure according to the Table III:

TABLE III

Measure	Quality loss
5	Imperceptible
4	Notable
3	A little bit disturbing
2	Distrbing
1	Very disturbing

The review of reconstructed images for each algorithm of LL network with the compression rate 4 is presented in the following.

Original picture SCG RP



OSS GDX GDM

As it can be seen, the best subjective measure have images compressed using the algorithms SCG and RP that have the lowest performation function; the worst are images compressed with the GDM algorithm.

VI. GIBO METHOD OF FORMING THE TRAINING SET FOR TRAINING THE BACKPROPAGATION NEURAL NETWORK USED IN IMAGE COMPRESSION

Experimental results had shown that very important role in image compression using neural network has training set with which the network is trained, and not only the selection of the adequate learning algorithm and network architecture. As a result of this research so called GIBO [5] (Good In Bad Out) method is created. This method forces those elements of input training data set that have the output, which has the smallest deviation from the expected output. It is showed that, by eliminating input blocks from the training set that mostly yield to its error, network training becomes faster, and calculated MSE smaller. In order to ensure that using this method the training set is not reduced too much, the number of rest elements after the reduction is recorded and the rule is set so that this number is not smaller then the half of the number of elements before the reduction. Also the upper limit for this number is set; it is 85% from the overall unreduced training set size, and so, the optimal size for the training set is between 50 and 85% of the unreduced training set size. In the case that this size is bellow the lower limit it is needed to reinitialize the network i.e. to initialize its weights and biases, and then calculate the maximal deviation according to the new values and then the described steps are repeated.

The results of the described method according to the performation function of the network (MSE) for different compression rates in backpropagation neural network of LP type and for SCG and RPROP learning algorithms are shown in Table IV.

TABLE IV

Compression rate	Algorithm	Time	MSE
LP 4	SCG	0.00194494	0.0010
	RPROP	0.00109606	0.0018
LP 8	SCG	0.00155806	0.0021
	RPROP	0.000758611	0.0022
LP 16	SCG	0.00120289	0.0025
	RPROP	0.000588944	0.0028

As the results in the table show, using GIBO method, the quality of the decompressed image is not degraded although is achieved with the higher speed of the algorithm

VII. CONCLUSION

Shannon theory [4] has brought modern communications into the theoretic trap from which is hard to find the exit. This theory defines "information" just as the binary digit (bit). The content of data is irrelevant. Bit flow is hence completely specified by the hardware systems parameters, just as image size and resolution are too. The only solution that exists till now is image compression with some of the known compression techniques with a loss i.e. the more the image is compressed, its quality is worse.

The crutiality of this problem led the reasearches in this field to the neural networks and to the developement of appropriate techniques for their application in the given domen.

The conclusion of this paper is that the best results of the compression of the digital grayscales images can be achieved by training the network with the SCG and RPROP backpropagation algorithms. Even for the compression rate 16, the result is the image with the satisfied quality and satisfied training time.

VIII. REFERENCES

[1] Luger G. F., Stubblefiled W. A., *Artificial Intelligence – Structures and Strategies for Complex Problem Solving*, University of New Mexico, Albuquerque, 1993

[2] Masalmah Y. M., Ortiz J., "Image Compression using Naural Networks", *Technical Report*, University of Puerto Rico, Mayaguez, Puerto Rico, 2002

[3] Rumelhart D. E., Widrow B.,Lehr M. A., "The Basic Ideas in Neural Networks", *Communications of the ACM*, March 1994, Vol. 37, No 3, 87-91.

[4] Shi Y.Q., Sun H., *Image and Video Compression for Multimedia Engineering-Fundamentals, Algorithms, and Standards*, CRC Press, USA, 1999

[5] Ilic D., "Image compression using neural networks", *Master Thesis*, University in Novi Sad, Technical faculty «Mihajlo Pupin», Zrenjanin, 2003