SAMPLE WEIGHTING WHEN TRAINING SELF-ORGANIZING MAPS FOR IMAGE COMPRESSION

Jari Kangas
Helsinki University of Technology
Neural Networks Research Centre
Rakentajanaukio 2 C, FIN-02150, ESPOO, FINLAND
tel: +358 0 451 3275, fax: +358 0 451 3277
email: Jari.Kangas@hut.fi

Abstract — Image compression is an essential task for image storage and transmission applications. Vector quantization is often used when high compression rates are needed. Self-Organizing Map (SOM) algorithm can be used to generate codebooks for vector quantization. Previously it has been demonstrated that using the special property of the SOM algorithm that the codebook entries are ordered one can use prediction coding of codewords to make the compression more effective. In this paper it is shown that training the SOM algorithm by using different weighting for sample blocks having different statistical characteristics one can further increase the compression efficiency.

IMAGE COMPRESSION BY SELF-ORGANIZING MAPS

Image compression is an essential task for image storage and transmission [1]. Lately the image compression using Vector Quantization (VQ) techniques has received large interest [2]. VQ methods offer good performance when high compression rates are needed. In VQ approaches adjacent pixels are taken as a single block, which is mapped into a finite set of codewords. In decoding stage the codewords are replaced by corresponding model vectors. The set of codewords and the associated model vectors together is called a codebook. In VQ the correlation which exists between adjacent pixels in a block is taken into account, and with a comparatively small codebook one achieves a small quantization error in reconstructed image.

The main problem in vector quantization is to find a codebook which minimizes the quantization mean error. Many design algorithms have been proposed. One of the best known is the Linde-Buzo-Gray (LBG) algorithm [3], which iteratively searches clusters in the training data. The cluster centers are used as the codebook model vectors, while the codewords are selected arbitrarily. An unsupervised neural network algorithm, the Self-Organized Map (SOM) algorithm [4][5] has also been used. In SOM the unit model vectors form the codebook, and the unit coordinates are used as codewords. Comparisons of the distortion errors in using the codebooks produced by either the SOM algorithm or the LBG algorithm indicate that these two methods give rather similar performance [6][7]. In [7], the SOM algorithm was found to be more robust than the LBG algorithm to initialization.

The main difference between LBG and SOM algorithms is in the order of the codebook model vectors. The LBG algorithm does not define any order in the codebook; the codewords for model vectors can be selected arbitrarily. On the other hand, the codebook trained using the SOM algorithm has an internal order; adjacent codebook entries which have similar codewords, have similar codebook model vectors.

Lately, it has been recognized, that the ordered codebooks can be used to improve the image compression efficiency [8][9][10]. Because the neighboring pixels in images correlate to each other, one can use the knowledge of the previous pixel value to predict the next pixel value, and code only the difference between the prediction and the real value. Therefore one can reduce the transmission rate considerably. Because the similarities between adjacent blocks in the original image are converted to similarities between the corresponding codewords, the prediction coding can now be used together with the vector quantization algorithm by predict coding the codewords (instead of pixel values). One can then improve the compression rates attainable by vector quantization by 15 to 20 % [10].

IMPROVING THE EDGE PRESERVATION IN IMAGE VECTOR QUANTIZATION BY SAMPLE OCCURRANCE WEIGHTING

Edge preservation is a serious problem in vector quantization of image data. The design algorithms tend to find good model vectors for samples which occur frequently but rare samples are not well represented. The edges in images form such a rare group of samples, where there are only few instances of separate edge orientations and amplitudes. However, the edges in images are important for the human acceptance of the decoding results. Ragged and uneven edges in images are very easily noticed and decrease the subjective quality of compression result, altough the signal-to-noise rations may show rather good decoding quality.

In [11] it was proposed that by modifying the training law of the SOM algorithm used in training the codebooks, one can improve the edge preservation capabilities. The proposal was based on the heuristic rule that one should pay more notice to image blocks where there are edges or other 'features'. The 'edgeness' of the block was computed using a variance measure of the gray level distribution of pixels x_k inside the block x:

$$\sigma^2 = \frac{1}{N} \sum_{k=1}^{N} x_k^2 - \bar{x}^2, \tag{1}$$

where $\bar{x}$ is the mean value of gray levels in block and N is the number of pixels. In [11] the variance was then used to compute a weighting factor $w(\sigma)$, which was considered as multiplying the number of occurrences of the sample x. In the following experiments we adopted a similar approach using (2) to compute the multiplier for sample count:

$$w(\sigma) = (\text{max\_weight} - 1.0) \frac{\sigma^2}{\sigma_{\text{max}}^2} + 1.0, \tag{2}$$

where σ_{max}^2 is the maximum variance value possible. Thus using the 'max_weight' value of 1.0, the algorithm behaves as the original.

Results with edge preservation

We used codebooks with 512 entries and coded 4 by 4 pixel subimages, which gave a 0.5625 bpp (bits per pixel) raw bit rate on the image compression. The codebooks were trained using 28 images of faces (both males and females) and tested using another face image, which was not contained in the training set. All the images were 512 by 680 pixel 256 gray level images. The topology of the Self-Organizing Map was three dimensional having eight units on each dimension.

The image degradation was measured using a peak signal-to-noise ration (PSNR) defined as:

$$\text{PSNR} = 10 \log \frac{255^2}{\text{MSE}} \text{ dB}, \tag{3}$$

where MSE was the mean square error. The resulting PSNR values and standard deviations of the results on ten training runs for each different weightings are collected to Table 1.

From the results one can easily see that the sample occurrence weighting improves image quality. The PSNR values grow up to weighting 20 and even after that up to weighting 200 the PSNR values are larger than using the standard version of the SOM algorithm.

In Figure 1 some details of decoded images are shown. The edge between ear and background has been decoded using codebooks which have been trained using weighting factors 1.0, 10.0, and 100.0, respectively. The continuity of the edge is improved when the weighting factor increases.

Max weight in training	PSNR value	Std. dev.
1	34.56	0.03
2	34.56	0.05
5	34.66	0.04
10	34.74	0.06
20	34.80	0.08
50	34.72	0.10
100	34.69	0.07
200	34.65	0.04
500	34.17	0.04
1000	33.82	0.00

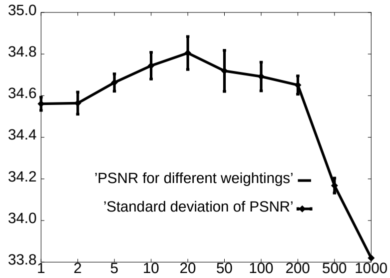

Table 1: *PSNR values in image coding and decoding experiments using different values for the maximum weighting. Ten independent training runs have been tried for each weighting.*

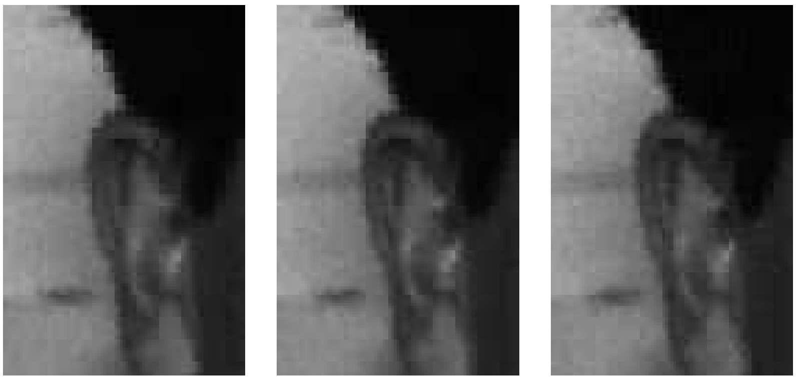

Figure 1: *The three images are details of decoded images where the compression codebooks have been trained using weighting factors 1.0, 10.0 and 100.0, respectively. Notice the increasing continuity of the edges in images from left to right.*

INCREASING THE COMPRESSION EFFICIENCY BY SAMPLE OCCURRANCE WEIGHTING

In modifying the sample count for SOM training one affects the distribution of the model vectors so that the flat blocks (where pixel gray values are more or less the same) get fewer representatives in codebook compared to the original training algorithm. Therefore the number of different codewords used in coding flat areas in images is less than if the original codebook were used. If prediction coding of codewords is used, decreasing the number of possible codes leads to more compact coding of the flat areas in images.

In the following some exeriments are described where the previous idea has been tried. Test image has been compressed using different weightings in codebook training and prediction coding the resulting codeword sequences. The following results are means over ten training experiments.

Results with original training algorithm and prediction coding

The test image was first vector quantized using the original SOM algorithm. The raw bpp rate for the coding is 0.5625, but some improvement can already be gained if the frequency of occurrence of each codeword is taken into account (using for example Huffman coding). Because the test image was not used in training the codebook not all codewords are used equally probably. After Huffman coding the bpp rate for the test image was 0.544, which is 3 % less than the raw value.

The image is vector quantized sequentially, one row at a time. Because there naturally exists some correlation between adjacent blocks in the image, it is advantageous to make a prediction of the codebook vector which is going to be used based on codewords used for the nearby blocks and then code the prediction error only. In our experiments we computed the prediction $\hat{a}$ of the codeword $a(i,j)$ to be used in location (i,j) as in [9]:

$$\hat{a}(i,j) = \begin{cases} a(i,j-1), & \text{if} \quad \begin{array}{r} d(a(i-1,j),a(i-1,j-1)) < \\ d(a(i,j-1),a(i-1,j-1)) \end{array} \\ a(i-1,j), & \text{otherwise,} \end{cases} \tag{4}$$

where the distance $d(a,b)$ between codes is defined on SOM as $d(a,b) = ||a-b||$. In image borders the prediction was based on available codewords.

After prediction and Huffman coding the codewords the bpp rate for the test image was 0.443. That means 19 % reduction in bpp from the Huffman coded raw codeword sequence, and 21 % reduction in bpp from the raw codewords. These figures are well in correspondence with the figures given in [10].

Results using the sample occurrence weighting

The bits per pixel rates and PSNR values of the experiments have been collected to Table 2. Different weighting factors have been used in the SOM codebook training and prediction coding of the codewords has been utilized before computing the bpp rates. The results show that when the weighting increases, the compression factor will also increase. The bpp rates start to decrease rapidly when the maximum weighting is increased beyond value 20.

Max weight in training	Huffman coding	Prediction and Huffman coding	Decrease from raw code	PSNR value
1	0.544	0.443	21 %	34.56
2	0.539	0.437	22 %	34.56
5	0.544	0.439	22 %	34.66
10	0.543	0.430	24 %	34.74
20	0.545	0.425	24 %	34.80
50	0.543	0.406	28 %	34.72
100	0.540	0.382	32 %	34.69
200	0.540	0.359	36 %	34.65
500	0.527	0.317	44 %	34.17
1000	0.525	0.296	47 %	33.82

Table 2: *Bits per pixel rates and PSNR values using different values for the maximum weighting. The relative decrease in bpp rate is computed against the raw bpp rate 0.5625.*

The quantization error is in its minimum when the weighting is between 10 and 50. After that it starts to increase again. When the weighting is more than 200 the quantization error has significantly increased from that of smaller weightings. To get a similar quality of images as in original quantization algorithm, one can use weightings up to 200. At that level the relative decrease in bpp rate is about 36 %.

In Figure 2 bits per pixel rates and PSNR values for different weightings are shown in another format. The bpp rates are on the vertical axis and the lower points on the Figure mean better compression ration. PSNR values are on the horizontal axis and the best quality of images is on the right side.

DISCUSSION

The Self-Organizing Map can be used to train vector quantization codebooks for image compression. Because the SOM algorithm orders the codebook entries prediction coding can be used to further improve the compression efficiency. In this paper it has been shown that training the SOM algorithm by using different weightings for sample blocks having different statistical characteristics one can further increase the compres-

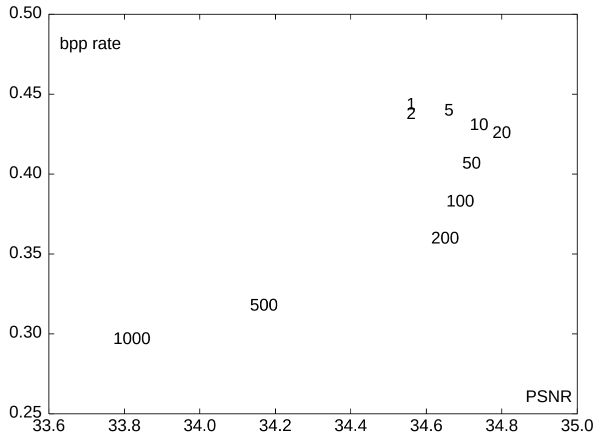

Figure 2: *Bits per pixel rates (after codeword prediction and Huffman coding) and PSNR values as a function of weighting factor in a two dimensional diagram.*

sion efficiency. Without affecting the image quality compression rates can be improved by 36 %. It is also possible to improve the image quality if somewhat smaller improvement in compression rates are allowed.

In these experiments only a variance of gray levels was computed of each image sample. Some other statistical measures might produce better results. For example, preserving the true lines going through the image sample is more important for subjective acceptance of the decoded image, than preserving random samples even if the variance of samples were the same. In future studies some other statistical measures are to be tested.

REFERENCES

[1] Anil K. Jain. Image data compression: A review. *Proc. IEEE*, 69(3):349–389, 1981.

[2] Nasser M. Nasrabadi and Robert A. King. Image coding using vector quantization: A review. *IEEE Trans. on Comm.*, 36(8):957–971, 1988.

[3] Robert M. Gray. Vector quantization. *IEEE Acoustics, Speech and Signal Processing Magazine*, 1:4–29, April 1984.

[4] Teuvo Kohonen. Self-organizing formation of topologically correct feature maps. *Biol. Cyb.*, 43(1):59–69, 1982.

[5] Teuvo Kohonen. The self-organizing map. *Proc. IEEE*, 78:1464–1480, 1990.

[6] Nasser M. Nasrabadi and Yushu Feng. Vector quantization of images based upon the Kohonen self-organization feature maps. *Neu-*

ral Networks, 1(1 SUPPL):518, 1988.

[7] J. D. McAuliffe, L. E. Atlas, and C. Rivera. A comparison of the LBG algorithm and Kohonen neural network paradigm for image vector quantization. In *Proc. ICASSP-90, Int. Conf. on Acoustics, Speech and Signal Processing*, volume IV, pages 2293–2296, IEEE Service Center, Piscataway, NJ, 1990.

[8] E. A. Riskin, L. E. Atlas, and S.-R. Lay. Ordered neural maps and their applications to data compression. In B. H. Juang, S. Y. Kung, and C. A. Kamm, editors, *Proc. Workshop on Neural Networks for Signal Processing*, pages 543–551, IEEE Service Center, Piscataway, NJ, 1991.

[9] Sergio Carrato, Giovanni L. Siguranza, and Luigi Manzo. Application of ordered codebooks to image coding. In *Neural Networks for Signal Processing. Proc. 1993 IEEE Workshop*, pages 291–300, IEEE Service Center, Piscataway, NJ, 1993.

[10] Gilles Burel and Jean-Yves Catros. Image compression using topological maps and MLP. In *Proc. ICNN'93, Int. Conf. on Neural Networks*, volume II, pages 727–731, IEEE Service Center, Piscataway, NJ, 1993.

[11] K. Y. Kim and J. B. Ra. Edge preserving vector quantization using self-organizing map based on adaptive learning. In *Proc. IJCNN-93-Nagoya, Int. Joint Conf. on Neural Networks*, volume II, pages 1219–1222, IEEE Service Center, Piscataway, NJ, 1993.