Adaptive Combination of PCA and VQ Networks

Andreas Weingessel, Horst Bischof, Kurt Hornik, and Friedrich Leisch

Abstract— **In this paper we consider Principal Component Analysis (PCA) and Vector Quantization (VQ) neural networks for image compression. We present a method where the PCA and VQ steps are adaptively combined. A learning algorithm for this combined network is derived. We demonstrate that this approach can improve the results of the successive application of the individually optimal methods.**

Keywords— **Adaptive Combination, Principal Component Analysis, Quantization, Image Compression**

I. Introduction

Transformation coding (e.g., Cosine Transformation, Principal Component Analysis (PCA), Wavelet Transformation, Subband Coding) and Vector quantization (VQ) are commonly used methods in image compression. Very often they are also used successively, i.e., first the image is transformed and then the coefficients are quantized.

In this paper we also use transformation coding (PCA) and quantization. We present the "Adaptive Network Combination" (ANC) for adaptively combining these two methods rather than applying them successively.

We have the following situation: Let x denote the original image which is reduced with a reduction function A, yielding $h = A(x)$. We have an expansion function B, which produces from h an approximation of x, i.e., $y = B(h)$. The error is given by $z = y - x = B(A(x)) - x$. The error image z usually has much lower entropy than the original image and can therefore be compressed considerably by a run-length encoding scheme, cf. [1]. In order to yield higher compression ratios the error images are usually also quantized $c = C(z)$ (using scalar or vector quantization), giving a lossy image compression method. The compression ratio and the amount of loss can be controlled by the number of bits used to quantize the levels of the error images. Even better results can be obtained by incorporating the above scheme in a hierarchical framework, e.g., image pyramids [2] or wavelets [3]. In this paper we consider only linear transformations for A and B; it is well known that Principal Component Analysis (Karhunen Loève Transform) is optimal for linear transformations, cf. [4]. In the scalar case the optimal quantizer is the Lloyd-Max quantizer [5], [6]. Therefore, one might expect that an image compression method based on the successive application of these algorithms would yield the optimal result. This is not the case. In this paper we show that considering the transformation and compression step together, i.e.,

$$C(z) = C(B(A(x)) - x)$$

can improve the results of these individually optimal methods.

Let us describe the reasoning behind this. With the transformations (A, B) we try to minimize the error image which does not necessarily yield low compression errors. For image compression we want to minimize the *quantized* error images. Usually, the error images have an unimodally shaped histogram centered at zero. However, for the result of the compression (i.e., compression ratio and reconstruction error), it is irrelevant where the center of the histogram is as long as the quantization intervals are moved in the same direction. The best results could be achieved if we had a multimodal histogram with peaks centered at the quantization pins.

In this paper we present a method where the quantization and transformation step are adaptively combined. In particular, in Section II we formulate both the transformation and the quantization in terms of neural networks, and derive a learning algorithm for the combined network. In Section III we demonstrate that this approach yields better results than the statistically optimal methods applied separately. Section IV gives some conclusions and outlines further research.

II. Adaptive Network Combination

Principal Component Analysis (PCA) extracts the linear subspace of the data where the variance of the data is maximal. During the last years, a lot of principal component extracting neural networks have been developed (see [4], [7] for a review). It has been shown that a linear autoassociative neural network with a bottleneck hidden layer is able to extract the principal component subspace.

In quantization one tries to map the data on a small set of cluster centers. Many neural network based methods have been proposed which also perform clustering, e.g., Learning Vector Quantization, Competitive Learning, Self-organizing Feature Maps, and so on, cf. [8].

We use an autoassociative network to perform PCA and simple competitive learning to perform quantization. The result of combining these networks is depicted in Figure 1.

This work was supported by a grant from the "Austrian National Fonds zur Förderung der wissenschaftlichen Forschung" (No. P10539-MAT).

A. Weingessel, K. Hornik, and F. Leisch are with the Institut für Statistik und Wahrscheinlichkeitstheorie, Technische Universität Wien, Wiedner Hauptstraße 8-10/1071, A-1040 Wien, Austria; email: {A.Weingessel, K.Hornik, F.Leisch}@ci.tuwien.ac.at

H. Bischof is with the Abteilung für Bildverarbeitung und Mustererkennung, Technische Universität Wien, Treitlstraße 3/1832, A-1040 Wien, Austria; email: bis@prip.tuwien.ac.at

In the left part of the figure we see a standard MLP, where the input vector x is propagated via the weight matrices A and B to the output layer, so we have $y = Bh = BAx$. Then, the errors $z = x - y = (I - BA)x$ are computed and quantized at the quantization layer c. (Note that the connections with weights $(-1, 1)$ are used to compute the error and are fixed.)

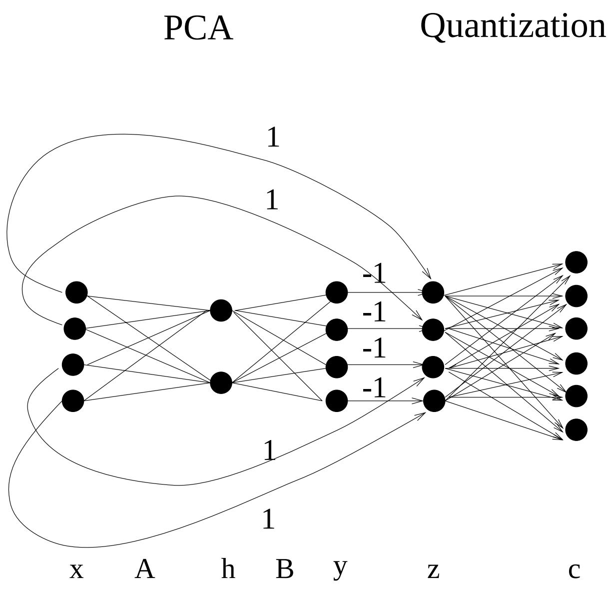


Fig. 1. The Adaptive Network Combination (ANC)

Generally speaking, when presenting an input pattern x_i, the MLP computes $y_i = f(x_i, \theta)$, where θ is the set of adjustable parameters (i.e., the weights). Instead of minimizing the mean square error $\sum_i ||z_i||^2 = \sum_i \sum_l z_{il}^2$, where z_{il} is the l-th component of z_i, we minimize the quantized error which equals $\sum_i \min_m \sum_l (c_{ml} - z_{il})^2$. By setting $p_i = \text{argmin}_m \sum_l (c_{ml} - z_{il})^2$ and $E_{il} = (c_{p_i l} - z_{il})^2$ we have for simple competitive learning

$$\frac{\partial E_{il}}{\partial c_{ml}} = \begin{cases} 2(c_{ml} - z_{il}), & m = p_i, \\ 0, & \text{otherwise.} \end{cases}$$

To adapt the MLP we compute the gradients

$$\frac{\partial E_{il}}{\partial \theta_j} = \frac{\partial E_{il}}{\partial z_{il}} \frac{\partial z_{il}}{\partial \theta_j} = -2(c_{p_i l} - z_{il}) \frac{\partial z_{il}}{\partial \theta_j}.$$

The error function is continuous, but not continuously differentiable at the cluster boundaries. If z_i lies exactly on such a boundary[1] one can choose arbitrarily one of the two cluster centers in order to perform gradient descent.

In the ANC, the weights are adapted according to the gradients computed above. That means, the MLP does not adjust its weights in order to decrease the reconstruction error, but the weights are adjusted in order to decrease the distance between the error made and the cluster pin by which the error is coded.

[1] This happens with probability 0 for a continuous input set.

III. Empirical Results

We report the experimental results on the house image as depicted in the left part of Figure 2. The image consists of 128×128 pixels with gray values in the range $[0, 255]$, transformed to $[-1, 1]$. For training the image is split into vectors or matrices according to the used network and an appropriate subset of them is randomly chosen for training and testing. We use online learning in our experiments (with the exception of the Lloyd-Max algorithm). To provide better readability the mean square error (MSE) in the tables is multiplied by 100.

Our experiments are performed as follows: First, the MLP is trained until convergence to optimally reconstruct the input, afterwards a quantizer is trained until convergence to cluster the errors made by the MLP. The weights of the MLP and the cluster centers of the quantizer are used as the initial weights for the ANC. This network is then trained with the learning algorithm described in the previous section.

Let us first demonstrate that ANC outperforms the individually statistically optimal methods, then we briefly report further experiments relevant to image compression applications.

A. Comparison with PCA and Lloyd-Max

In order to compare the results to the individually statistically optimal methods we use PCA and scalar quantization with Lloyd-Max[2]. In this section we show how our method improves the successive application of these two methods.

In order to have a simple structure we split the images into 4096 4-dimensional vectors and compute the exact PCA with 2 principal components. Figure 2 shows the original image (left part), the reconstructed image (middle part), and the error of this reconstruction (right part). One can see regular effects of the PCA in the horizontal stripes and that the error is larger at the edges (e.g., roof). The mean square reconstruction error is $0.5327 * 10^{-2}$. Applying a Lloyd-Max Quantizer with 4 clusters to the error image gives an quantized error of $0.1079 * 10^{-2}$. This is the result of the successive application of the two individually statistically optimal methods.

For our method we train a 4-2-4 network to reconstruct the data. The errors of the MLP are quantized by the scalar Lloyd-Max quantizer with 4 clusters. Then, the ANC is trained. This procedure is repeated for 30 random initializations of the MLP weights. Table I shows the average MSE and its standard deviation after training the MLP, after quantizing the error and after training the ANC. The column *Decrease* gives the decrease of the error of the ANC as compared to the MLP/Quantization

[2] It is possible only for the scalar case to calculate the optimal quantizer.

Fig. 2. Original Image; PCA Reconstruction; Error of Reconstruction

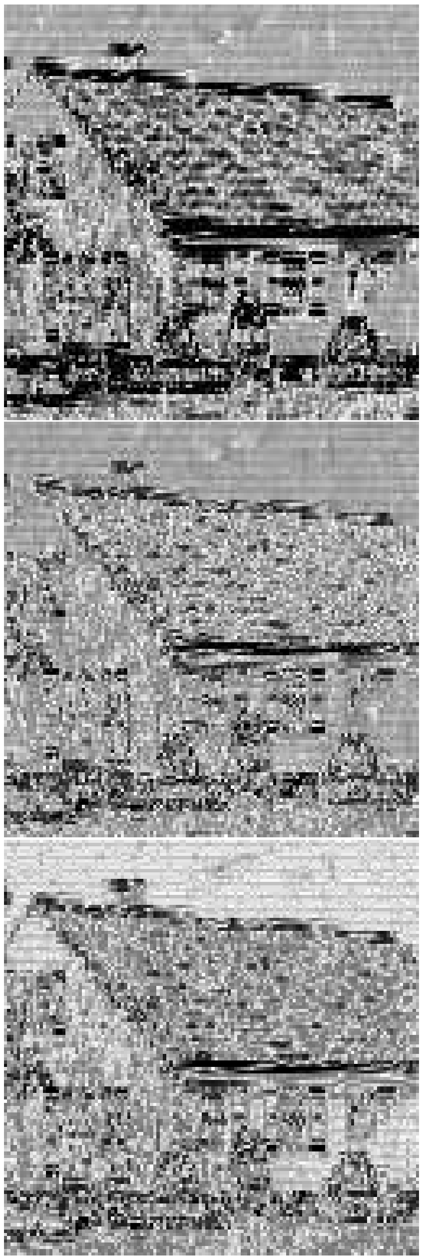


Fig. 3. Errors of Reconstruction: PCA, MLP+Quantization, ANC

approach. For comparison we also show the results of the Lloyd-Max quantization of the PCA error (*Lloyd-Max*).

All Data are the results obtained when training the network with all 4096 vectors. We see that on average the ANC decreases the error by more than 12%. ANC significantly outperforms the successive application of PCA and Lloyd-Max.

To test generalization we applied the weights trained on the house image to a larger image (containing the original house image). The results can be seen in the *Generalization* part of Table I. Similar results are obtained when training the networks with just 1000 data vectors of the original house image and testing it with 1000 different vectors.

In Figure 3 one can see the error images for the PCA without error quantization, the combined MLP and Quantization approach, and the ANC (the darker the image, the bigger the error). To provide better visibility the square root of the error is depicted and a common lookup table is adjusted.

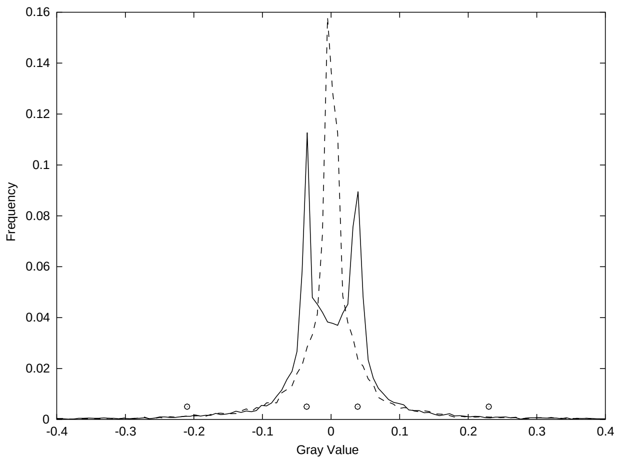
Figure 4 illustrates that the ANC does more than just some fine tuning of the parameters. Figure 4a shows the histogram of the errors made by the linear MLP and by the MLP part of the ANC. The MLP part learned to reproduce the data in a way that is adapted to the quantizer.

In Figure 4b we see the error curves for the quantization algorithm and the ANC algorithm. We see that as soon as the ANC is trained the error drops immediately.

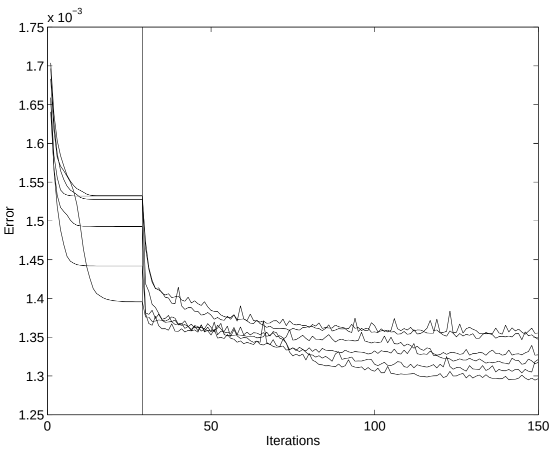
TABLE I

MSE when Using a 4-2-4 Linear MLP and 4 Clusters

	All Data		Generalization	
	mean	sdev	mean	sdev
MLP	0.5416	0.00021	0.5859	0.00036
Quantization	0.1175	0.00057	0.1298	0.00054
ANC	0.1033	0.00008	0.1201	0.00014
Decrease	12.09%	0.43%	7.51%	0.40%
Lloyd-Max	0.1079		0.1298	



(a) Histograms of Errors of the linear MLP (dashed) and of the MLP part of the ANC (solid) together with the Quantization Pins



(b) Error Curves for 6 different Experiments for the Lloyd-Max Quantizer (left) and the ANC (right)

Fig. 4. Learning Behavior of the ANC

So, usually 5 to 10 presentations of the data to the ANC are enough to get significantly better results than for the MLP/Quantization approach, in most cases the error decreases already significantly after just one epoch.

B. Further Experiments

In the previous section we have shown that our approach improves the performance of the successive application of the individually optimal methods, PCA and Lloyd-Max quantization. In this section we summarize results on other more "realistic" experiments done with different numbers of clusters, two-dimensional data, and vector quantization instead of scalar quantization.

- In [9], [10], [11] we have proposed a method called "PCA Pyramids" where PCA is used within an image pyramid to obtain the reduction and expansion functions between the levels of the pyramid. The compression ratios and the error obtained compare favorably with other compression techniques. The MLPs used for PCA pyramids are not fully connected autoassociative networks, but have heavy constraints on the weights resulting from the image pyramid structure. When using these MLPs the ANC approach also improves performance in a similar manner as in the previous experiments, see Table II.
- It is well known (e.g., [5]) that vector quantization always yields better compression rates than scalar quantization. Therefore, we investigated whether the ANC also has an effect with vector quantization. Again, the ANC reduces the error, see also Table II.
- Table II shows an experiment on the house image where we have used a PCA network in a pyramid ($4 \times 4/4$ pyramid) and a vector quantizer with 32 cluster centers. ANC considerably improves the performance of MLP and Quantization.

TABLE II

Results on 2-Dimensional Subimages

	MLP	Quantization	ANC	Decrease
mean	0.8159	0.3336	0.2717	18.56%
sdev	0.0088	0.0165	0.0023	0.99%

- We also trained the ANC initialized with random weights, not with the pre-trained weights of the MLP and Quantizer. This approach sometimes yields even better results than the MLP-Quantization-ANC-approach, but the variances of the results are higher.

IV. Conclusion

In this paper we have shown how to adaptively combine different types of networks, in our case MLP and VQ.

This combined approach outperforms the successive training of the networks and the successive application of the statistically optimal methods. This demonstrates that the formulation of "conventional" algorithms in terms of neural networks gives us additional possibilities not available in each of the methods alone.

We are currently working to incorporate the proposed algorithm in the PCA pyramid framework for purposes of image compression. Previous results on PCA pyramids with uniform quantization and the results reported here give us considerable confidence that this extension can yield high compression rates at very low error.

The work presented here demonstrates that a lot of efficiency can be gained by considering the individual steps of compression jointly rather than separately. Our framework can also be adapted to other image compression techniques.

References

[1] Martin Vetterli and Jelena Kovacevic, *Wavelets and Subband Coding*, Prentice Hall PTR, 1995.

[2] P. J. Burt and E. H. Adelson, "The Laplacian pyramid as a compact image code", *IEEE Transactions on Communications*, vol. 31, no. 4, pp. 532–540, Apr. 1983.

[3] Stephane G. Mallat, "A theory for multiresolution signal decomposition: The wavelet representation", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 11, no. 7, pp. 674–693, July 1989.

[4] Pierre F. Baldi and Kurt Hornik, "Learning in linear neural networks: A survey", *IEEE Transactions on Neural Networks*, vol. 6, no. 4, pp. 837–858, July 1995.

[5] Robert M. Gray, "Vector quantization", *IEEE ASSP Magazine*, vol. 1, no. 2, pp. 4–29, Apr. 1984.

[6] Joel Max, "Quantizing for minimum distortion", *IRE Transactions on Information Theory*, vol. IT-6, pp. 7–12, Mar. 1960.

[7] K. I. Diamantaras and S. Y. Kung, *Principal Component Neural Networks: Theory and Applications*, Adaptive and Learning Systems for Signal Processing, Communications, and Control. John Wiley & Sons, Inc., 1996.

[8] S. Haykin, *Neural Networks: A Comprehensive Foundation*, New York: McMillan, 1994.

[9] Horst Bischof and Kurt Hornik, "PCA-Pyramids for image compression", in *Advances in Neural Information Processing Systems (NIPS*94)*, G. Tesauro, D.S. Touretzky, and T.K. Leen, Eds. 1995, vol. 7, pp. 941–948, MIT Press.

[10] Horst Bischof and Kurt Hornik, "Image compression by PCA-Pyramids", in *4th Int. Workshop MEASUREMENT'95, Smolenice, Slovakia*, M. Karovicova and A. Plackova, Eds. 1995, pp. 7–17, Inst. of Measurement Science.

[11] Horst Bischof, *Pyramidal Neural Networks*, Lawrence Erlbaum Associates, 1995.