Image Compression with Neural Networks and SVD

David Chettrit
Stephan Seebacher
Sezer Güler
Department of Computer Science, ETH Zurich, Switzerland

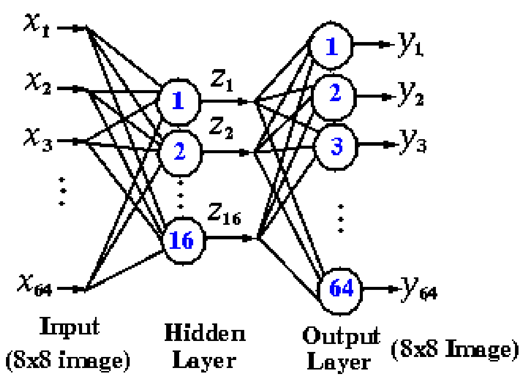

Figure 1. Neural Network Structure for compression.

***Abstract*—**

I. Introduction

II. The Structure of a Paper

III. Preliminaries

A. Neural Networks

Inspired by biological nervous systems, such as the human brain, artificial neural networks are used for information processing. In biological nervous systems, highly interconnected neurons work together to solve a specific task. Each neuron solves a subtask and communicates with other neurons. In artificial neural networks this idea is used to approximate arbitrary non-linear functions. The nodes(neurons) are connected by weighted edges and the topology is chosen depending on the problem. In the training phase, the weights of the edges are set depending on the data. Neural networks are often represented in layers. Each layer contains some of the nodes and then follows the interconnections and then the next layer.
wieso verwendet man denn gerade Neuronale netzwerke....

B. Neural Network Structure

Figure 1 shows the neural network structure we use for compression. There are three layers: input layer, hidden layer and output layer. The input layer and the hidden layer are fully interconnected as are the hidden layer and the output layer. The weights of the edges are detrermined by the training algorithm which is described in the next section.

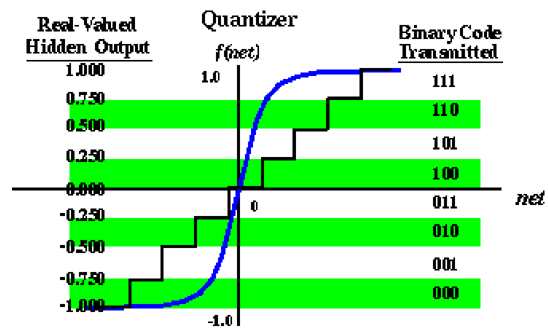

Figure 2. Neural Network Structure for compression.

C. Neural Network Training Algorithm

For the training 8 by 8 chunks of the image are chosen uniformly at random. The output of the bottleneck-like neural network has to be the same as the input and the weights are set accordingly. This process is iterated until the image is reconstructed faithfully. Different learning methods were tried like (....TODO) but it turned out that for the Levenberg-Marquardt method the results were best.

D. Singular Value Decomposition

The singular value decomposition(SVD) is a matrix factorization. A m×n matrix M can be written as a product of three matrices U, Σ and V^*, where U is a unitary m×m matrix, V^* is the conjugate transpose of a n×n unitary matrix V and Σ is a m×n diagonal matrix. The entries on the diagonal of Σ, which are non-negativ real numbers, are called the singular values of M. The columns of U are called left singular vectors and the rows of V^* are called right singular vectors of M.
SVD can be used for image compression the following way. The singular values are sorted in descending order and then only the largest singular values and the corresponding right and left singular vectors are used to save the image.

E. Quanitization

Quantisation as shown in Figure 2 is the process of mapping a large set of values to a smaller set of values. In Figure 2 we construct 8 possible binary codes which can be represented with 3 bits. Each code represents an interval of the decimal values. For example, for the values between

Figure 3. Visual comparison of original and reconstructed image.

-1.0 and -0.75 we use the code 000. This way, we use 3 bits for a pixel that originally is represented by 8 bits. This compression method is lossy and the original image can not be reconstructed.

IV. The Compression Algorithm

1) Train neural network with 100 images
2) 2 Iterations of re-training on neural network with actual image
3) Compress image using neural network
4) Compress with quanitization

V. Results

We tested the compression algorithm with 100 images which were not in the training set and get a mean error of (......) and a compression rate of (......).
Figure 3 shows an example image before any compression on the left and the reconstructed image after compression on the right.

VI. Discussion

We also tried to apply SVD on the image and then compress the U and V matrices with our neural network. The training had to be adapted. But the decompressed U and V matrices looked very different than the original ones.

Strengths: Mean error, Compression rate.
Weakness: Time

VII. Computational Intelligence Laboratory Requirements

A. Developing a Novel Solution

B. Comparison to Baselines

C. Write Up

1) Installation:
2) Compiling LaTeX:
3) Equations:
4) Tables and Figures:

D. Grading

VIII. Summary

Acknowledgements