COMENIUS UNIVERSITY IN BRATISLAVA
FACULTY OF MATHEMATICS, PHYSICS AND INFORMATICS
DEPARTMENT OF APPLIED INFORMATICS

Neural Networks

Project 2a – Image Compression

April 15, 2025

OVERVIEW

Task: Implement lossy image compression using *principal component extraction* (wiki), where a greyscale image is partitioned into non-overlapping fixed-size rectangular blocks, which are then encoded using only the first few principal components. Use the GHA algorithm to train the network (see also Haykin, 2009, for details of GHA).
Deadline: May 11, 23:59
Late submissions are penalized by -2 points each day. **It is not possible to submit a project more than 5 days after the deadline.**

METHOD

1. load and partition image:
   - generally: for some block size $w \times h$, partition the input image (of size $W \times H$) into $(W/w) \cdot (H/h)$ non-overlapping rectangular blocks
     - JPEG compression uses a fixed 8×8 block size, **do so as well**
2. find the first $1 \leq k \leq w \cdot h$ principal components, use $k \geq 8$:
   - flatten the $w \times h$ blocks to vectors in wh-dimensional space
   - find the principal components (try sequential and parallel modes)
3. encode image:
   - for each image block, output only the coefficients for the first k components
     - unless $k = w \cdot h$, this encoding is lossy
4. reconstruct image:
   - using only the principal component vectors and coefficients from the previous step

Hints:

- use small initial weights, consider keeping their mean zero

- use a small learning rate; a decreasing schedule (as in training SOMs) might also help
- vectors representing different neurons should converge to be perpendicular
  - the length of such vectors should converge to one (cf. stabilizing term/normalization)
- principal components are unique (disregarding sign and limited by numerical accuracy)

REPORT

Using this greyscale 256x256 image of Elaine:

- show the result of compression for at least three different values of k
- visualize the eight most significant extracted components (as 2D images)
- plot the eigenvalues of the PCA correlation matrix in decreasing order
- examine, how well do the principal components generalize between images
  1. find two images (e.g. close portraits) similar to the sample
     - rescale if sizes differ
  2. extract principal components from the *sample*
  3. use them to encode and reconstruct the two images

BONUS

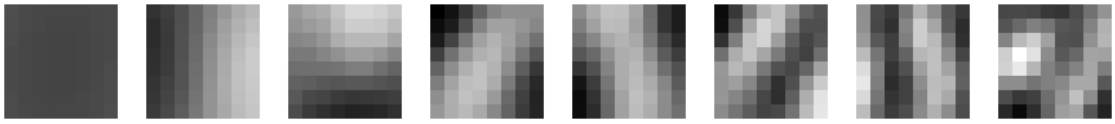
Implement a second way of finding the principal components, using an autoassociative network (autoencoder, wiki) with one *linear* hidden layer and output layer.
Try using:

- two separate weight matrices:
  - $\boldsymbol{W}^{hid}$ for input-to-hidden
  - $\boldsymbol{W}^{out}$ for hidden-to-output
- one *shared* weight matrix:
  - $\boldsymbol{W}^{hid} = \boldsymbol{W}$
  - $\boldsymbol{W}^{out} = \boldsymbol{W}^T$

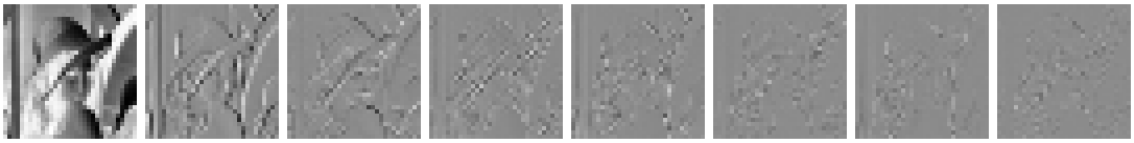
Compare the results of the MLP variants with GHA, both qualitatively and quantitatively.

EXAMPLE

The first eight principal components of 8x8 blocks from a 256×256 image of Lenna:



The 32×32 coefficients for the components:



The reconstructions from the first one, first two,... first eight components: