# Notes on Autoencoders

compiled by D. Gueorguiev 3/25/24

An autoencoder is a specific type of a neural network, which is mainly designed to encode the input into a compressed and meaningful representation, and then decode it back such that the reconstructed input is as similar as possible to the original one. The purpose of autoencoders is learning in an unsupervised manner an “informative” representation of the data that can be used for various tasks such as clustering.

The problem is defined as to learn the functions (encoder) and (decoder) that satisfy:

(1)

where is the expectation over the distribution of , and is the reconstruction loss function, which measures the distance between the output of the decoder and the input. The latter is usually set to be the -norm. The Figure provides an example of an Autoencoder application:

A diagram of a coder

Description automatically generatedFigure: an Autoencoder example with an image

Historically, and are neural networks. In the special case that A and B are linear operations, we get a linear autoencoder (see [3]).

## Appendix

### Linear Autoencoders and Principal Component Analysis

In this section only feed forward neural nets with one layer of input units, one layer of output units and one or several layers of hidden units will be considered. All neural nets will have linear activation units. We assume that there are input patterns and corresponding target output patterns which are used to train the network. For this purpose quadratic error function is used and defined as where is the current function implemented by the network. During the training phase, the weights and are successfully modified in order to reduce . This section will conduct careful analysis of feed forward networks with linear units.

Notation and Assumptions: Both and are -dimensional vectors. Our first network will consist of input and output layers of inputs and outputs plus one hidden layer with units. The weights connecting the inputs to the hidden layer are described by a real matrix and those from the hidden layer to the output by an a real matrix . Thus the error function is written as:

(1)

We define the usual sample covariance matrices , , ,

Problem: Find the matrices and which minimize .

We will use spectral analysis to gain insights of E. We will establish connections in the auto-associative case to the PCA.

//TODO: this is mainly review and discussion of [3]

Literature

[1] [Autoencoders, Dor Bank et al, 2021](https://github.com/dimitarpg13/transformers_intro/blob/main/articles_and_books/Autoencoders.pdf)

[2] [Autoencoders, Unsupervised Learning, and Deep Architectures, Pierre Baldi, 2012](https://github.com/dimitarpg13/deep_learning_and_neural_networks/blob/main/literature/articles/Autoencoders_Unsupervised_Learning_and_Deep_Architectures_Baldi_2012a.pdf)

[2] [Neural Networks and Principal Component Analysis: Learning from Examples Without Local Minima, Pierre Baldi, Kurt Hornik, 1988](https://github.com/dimitarpg13/deep_learning_and_neural_networks/blob/main/literature/articles/Neural_Networks_and_Principal_Component_Analysis-Learning_from_Examples_Without_Local_Minima_Baldi_Hornik-89.pdf)