Assignment 1: Autoencoders
MA-INF 2217: Advanced Deep Learning for Graphics

Assistant: Soumajit Majumder `majumder@cs.uni-bonn.de`

1 Practical Exercise

For this task, you will be implementing an Autoencoder on the MNIST dataset. The autoencoder comprises of a compression (encoder) function and a decompression (decoder) function (See Fig. 1) which are based on CNNs. The encoder/decoder function parameters are optimized using Stochastic Gradient Descent to minimize the reconstruction loss.

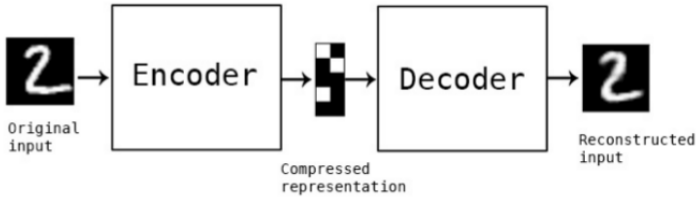

Figure 1: Autoencoder

An autoencoder takes an input $x \in \mathcal{Z}^d$ and maps it to a hidden representation $y \in \mathcal{Z}^{d'}$ in the form: $y = s(Wx+b)$, where s is the activation function such as *tanh* or *sigmoid*, W represents the weight matrix and b is the bias matrix. The latent representation y is then mapped back through a decoder into the reconstruction z of x. The mapping through the decoder follows a similiar transformation: $z = s(W'y+b')$. The model parameters: W, W', b, b' are optimized to minimize the average reconstruction error, such as the squared error loss: $L_{SE}(x,z) = \|x-z\|^2$ or the cross-entropy loss: $L_H(x,z) = \sum_{k=1}[x_k \log(z_k) + (1-x_k)\log(1-z_k)]$. Your tasks are:

1. Create an encoder and a decoder of 2 hidden layers, i.e. $y = s(W^2 \cdot s(W^1 x + b^1) + b^2)$ and $z = s(W^{2'} \cdot s(W^{1'} y + b^{1'}) + b^{2'})$. The first hidden-layer will have 256 units and the second 128 units. Use the sigmoid function as your activation.
2. Initialize the weight matrices $(W^2, W^1, W^{2'}, W^{1'})$ and biases $(b^2, b^1, b^{2'}, b^{1'})$ by sampling random values from a normal distribution.
3. Train the autoencoder to minimize the average mean squared error. Plot the reconstructed images after 20 training iterations.
4. Use the cross-entropy loss function instead of the mean squared loss function. Train again your autoencoder and plot the reconstructed images after 20 training iterations.