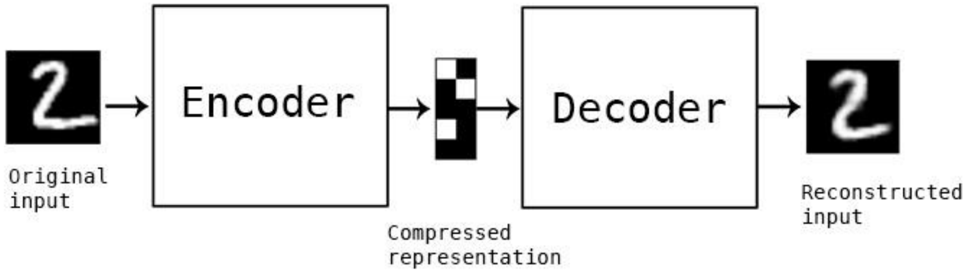
Homework 04

Neural Networks and Deep Learning

Dr. Alberto Testolin
Dr. Federico Chiariotti
A.A. 2019-20

Autoencoders for digit reconstruction



- Use the MNIST dataset to explore unsupervised learning (autoencoders)
- Train, validate and test your neural network using PyTorch
- Write the report (as always, 3 pages length + Appendix for additional figures/tables)
- We will run your model on a separate (hidden) test set to create the rankings

Don’t worry, rankings are just for fun!

Autoencoders for digit reconstruction

Homework objectives:

1. Extend the available PyTorch script (Lab05) or create a brand-new one, in order to:
   - Train an autoencoder on the MNIST:
     - Using proper methods for tuning the hyperparameters / avoiding overfitting
     - Systematically exploring how the size of the hidden layer (latent code) affects reconstruction performance
   - Test the reconstruction capability of the autoencoder:
     - On standard MNIST test images
     - On MNIST test images
       - » corrupted with Gaussian noise (you should explore different noise levels)
       - » corrupted with occlusions (i.e., by deleting a portion of the image)

2. Write a short report describing your work and the results achieved (figures are appreciated)

3. Send the Homework through the Moodle platform:
   - the script **MUST** work by running the following command: “`python trained_model.py`”
   - the script **MUST** return the mean reconstruction error of images in the file: “`MNIST.mat`”

     -- note that images in our test set will be corrupted using an ad-hoc noise source
     -- if you are normalizing / standardizing your input data, the mean reconstruction error should still be computed in the original pixel values (in order to be compatible with submissions that are not using normalization)

Autoencoders for digit reconstruction

Homework objectives:

[Optional]

- Improve learning by implementing a denoising autoencoder, which learns to reconstruct clean images starting from a corrupted version of the same pattern
  - NB: the way you define the corruption process has an impact on the model

- Explore the generative capabilities of autoencoders by sampling from codes in the latent space (i.e., activations of hidden units that are not directly derived from an input pattern)
  - Is the representational space of your autoencoder regular enough to allow for a smooth sampling? (i.e., nearby points in the representational space produce similar images)
  - If not, you can try to implement a variational autoencoder and compare the generation capabilities of the two approaches