GENERATIVE NETWORKS

DEEP LEARNING (SS2020)
6. COMPUTER LAB (BONUS LAB, 10P)

1. Introduction

In this bonus lab you are asked to implement a vanilla variational autoencoder for the MNIST dataset. The goal is to analyse whether the generative ability of VAEs increases with the complexity of the networks used for encoding and decoding.

2. Models

In this lab we will use VAEs as described in Lecture 12. of the course. We recommend you to read the "Tutorial on Variational Autoencoders" by C. Doersch (arXiv:1606.05908). There are several implementations of VAEs available on Github/Gitlab, which you may use for inspiration.

We will assume an Euclidean latent noise space $\mathcal{Z} = \mathbb{R}^n$ and a simple multivariate normal distribution $p(z) = \mathcal{N}(0, \mathbb{I})$ on it. The baseline VAE will have both the decoder $p_w(x \mid z)$ and the encoder $q_v(z \mid x)$ implemented by fully connected networks with one layer only (i.e. without hidden layers). The extended variant will have the decoder and encoder implemented as multilayer networks. The latent noise space will be the same for both variants.

3. Assignments

Assignment 1. (7p)
Implement the baseline and the extended VAE. Train them on MNIST data.

Assignment 2. (3p)
Compare the generative performance of the two models. Report a panel of images generated by each of the two models. Unfortunately, it is not possible to quantify the performance of generative models like VAEs and GANs in terms of training data log-likelihood because its estimation is not tractable. The paper `arXiv:1802.03446` lists and discusses 24 different surrogate metrics. This in fact shows, that we do not know how to measure the performance of deep generative models in a meaningful and tractable way. For the sake of simplicity, we recommend you to use the Maximum Mean Discrepancy w.r.t. a Gaussian kernel for a quantitative performance comparison of the two VAEs.