HY673 - Assignment #3

Yannis Pantazis

Wednesday April 12th, 2023
Due date: Friday May 5th, 2023

Instructions

- **Step zero:** Download the datasets from `https://drive.google.com/file/d/1w-1mn72DxKyqslPCzGzHtmjd-kU3Yjqi/view?usp=sharing`.
- **Due date:** Friday May 5th, 2023, 2023
- Submission via e-mail to the class account: hy673@csd.uoc.gr
- Provide one file with the solutions.
- Provide one folder with code.
  - The name of each file in the folder should indicate the respective exercise (e.g., ex_2c.py or exercise_2c.py).

Exercise 1: Implementation of Conditional VAE

In this exercise, you will extend the VAE model presented in Tutorial 6. Let x be an MNIST digit image and y be the corresponding one-hot encoded label. We can define $p(x|y)$ to be the conditional data distribution (a.k.a. the conditional marginal likelihood), $p(x, z|y)$ to be the conditional joint generative distribution, $p(z|y)$ to be the conditional prior distribution, $p(x|z, y)$ to be the conditional likelihood of the decoder (a.k.a, the conditional generative distribution) and $q(z|x, y)$ to be the conditional approximate posterior distribution.

(a) Write down the conditional ELBO (see Lecture 10 for the unconditional formulation) and the ELBO variation with the Kullback-Leibler divergence as one of the two terms of ELBO (see Lecture 11 for the unconditional formulation).

(b) Using a (stochastic) decoder with input (z, y) and a (stochastic) encoder with input (x, y), implement the conditional VAE for MNIST digit generation.

(c) Using the trained generative model from (b), write a program that takes as input a number and returns an image with the number where each digit has been conditionally generated from the pre-trained model.

Using the trained VAE obtained in (b), write a function that generates images based on an array of input labels ranging between 0 and 9 (e.g., if the input is [0, 2] the function should return two generated images conditioned on 0 and 2, respectively). Include examples of images produced by this function, and indicate the corresponding input labels.

Exercise 2: Train a Conditional VAE to Generate Light Curve Time-series with Caustic Crossings

Background: In astrophysics, galaxy clusters are the largest gravitational lenses that can make multiple images of, and substantially magnify, sources at cosmological distances. The light of individual stars in distant galaxies can be detected as they cross the caustics of lensing clusters. An example of 10 simulated light curves crossing a caustic is given in the figure below. There are four cosmological parameters denoted by k, g, s and p in the labels.dat file (attached) that affect the shape, smoothness and size of the caustic crossings.

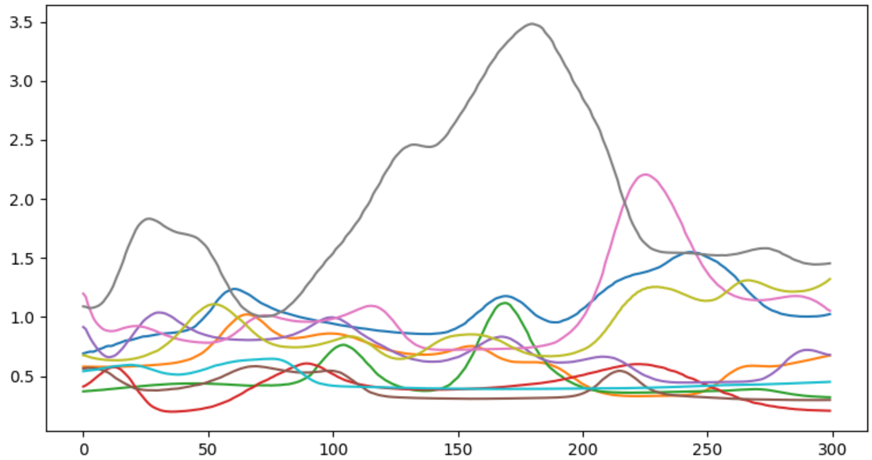

Figure 1: Ten light curve trajectories taken from '12345/05.dat' file which corresponds to $p = 0.0.195695$. The aim of this exercise is to generate similar synthetic trajectories from the conditional VAE.

In this exercise, you will train a conditional VAE on light curve time-series using one of the cosmological parameters as condition (p) and then generate new time-series for values of p that are not in the training set. File data_loader.py implements how to read from the data (both the samples and the labels).

(a) Construct a conditional VAE with dense layers. Both encoder and decoder will have one hidden layer with 128 units. The dimension of the latent code will be 32 while the dimension of the condition will also be 32 (just repeat the value of p 32 times).

(b) Train the constructed conditional VAE on the given data. Depending on your computational resources, you may have to reduce the number of samples in your training set.

(c) Interpolation and extrapolation. Use the trained model from (b) and generate new light curve time-series for $p \in [0.01, 5]$. Plot the generated trajectories.

(d) BONUS QUESTION: Use a dimensionality reduction algorithm to show the generated trajectories in two dimensions. Two standard visualization algorithms are UMAP (`https://github.com/lmcinnes/umap`) and tSNE (`https://scikit-learn.org/stable/modules/generated/sklearn.manifold.TSNE.html`). Check whether or not the projected data have a clear ordering that can be related with the value of the condition, p.

Exercise 3: Training of an Energy-based Model on Swiss roll dataset

(a) Train an EBM

$$p_\theta(x) = \frac{1}{Z_\theta} e^{f_\theta(x)}$$

on the 2D Swiss roll dataset (attached; see also the figure below) using the contrastive estimation algorithm. For the training, it is required to generate new samples from the EBM using the Langevin MCMC algorithm given by

$$x_{t+1} = x_t + \epsilon \nabla_x \log p_\theta(x_t) + \sqrt{2\epsilon} z_t, \ t = 0, ..., T-1,$$

where $z_t \sim \mathcal{N}(0, I)$ are i.i.d. Gaussian samples and $\epsilon > 0$.

(b) Starting with $x_0 \sim \mathcal{N}(0, 3^2 I)$, generate 1000 samples for $\epsilon = 0.1, 0.01$ & 0.001 and $T = 1000$ steps. Plot the generated samples and compare them against to the ground truth.