HY673 - Spring 2026 — Problem Set #3

Yannis Pantazis (20/04/2026)

Instructions

- **Due date:** Monday May 4th, 2026
- Submission via e-mail to the class account: hy673@csd.uoc.gr
- Provide one file with the written solutions.
- Provide one folder with code.
  - The name of each file in the folder should indicate the respective exercise.
  - Your code should run on Colab.
- All assignments in this course are individual, not group assignments. You may freely discuss homework assignments with your classmates. The final solutions, however, must be written entirely on your own.
- You are allowed to use generative AI tools (e.g., ChatGPT, Gemini) only for grammatical corrections unless explicitly permitted otherwise. To maintain academic integrity, students must disclose any use of AI-generated material.

Problem 1 (Implementation of RealNVP on MNIST dataset – 30 points)**.** *In this exercise, you will extend the NICE model which was presented in Tutorial 7.*
(a) Implement RealNVP model. Essentially, you have to implement

$$\begin{bmatrix} x_A \\ x_B \end{bmatrix} = \begin{bmatrix} z_A \\ z_B \odot e^{\alpha_\theta(z_A)} + m_\theta(z_A) \end{bmatrix}$$

where $\{x_A, x_B\}$, $\{z_A, z_B\}$ are partitions of the data and noise vectors, respectively, $\odot$ denotes element-wise product and both $m_\theta(\cdot)$ and $\alpha_\theta(\cdot)$ are neural networks with parameters θ. Note that $dim(x_A) = dim(z_A)$ and $dim(x_B) = dim(z_B)$.

(b) Train RealNVP on MNIST dataset. Keep in mind that the above RealNVP equation applies to all transformation steps. Experiment with 5 and 10 coupling layers and compare the results.

(c) Perform linear and sinusoidal interpolations between two MNIST digits in the latent space (i.e., the base space of z). In particular, let $z^{(1)}, z^{(2)}$ be two MNIST digits in the latent space, you will generate and plot
$z_\lambda = (1-\lambda)z^{(1)} + \lambda z^{(2)}$ *(linear) and*
$\bar{z}_\lambda = (1-\sin(\lambda\pi/2))z^{(1)} + \sin(\lambda\pi/2)z^{(2)}$ *(sinusoidal)*
interpolations for $\lambda = 0 : 0.1 : 1$.

Problem 2 (Conditional RealNVP on MNIST – 30 points). *(a) Add digit information to the RealNVP model and learn the conditional distributions. In particular, use one-hot encoding for the digit labels and concatenate them in the input of the neural nets. The equation of the conditional RealNVP model is given by*

$$\begin{bmatrix} x_A \\ x_B \end{bmatrix} = \begin{bmatrix} z_A \\ z_B \odot e^{\alpha_\theta(z_A, y)} + m_\theta(z_A, y) \end{bmatrix}$$

where $y \in \mathbb{R}^{10}$ corresponds to the one-hot encoding vector of the MNIST digit labels.

(b) Train conditional RealNVP on MNIST dataset. Compare the obtained results with the generated digits from 1(b). Can we say something about the number of required transformations when conditional generation is utilized?

Problem 3 (Implementation of Conditional VAE – 40 points). *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.