Advanced Course on Deep Generative Models **Spring 2024**

Exercise 6: Normalizing Flows

Dan Rosenbaum

Due: 14 July 2024

In this exercise we will use the same notebook we used in exercises 4 and 5.

`https://uvadlc-notebooks.readthedocs.io/en/latest/tutorial_notebooks/tutorial12/Autoregressive_Image_Modeling.html`

The goal of this exercise is to implement a normalizing flow model, as presented in class.

In order to do that, use the same training process and model structure as the implemented autoregressive model. Follow the instructions below:

- Use the same continuous representation of the data as in exercise 5.
- The input image should be flattened and treated as a vector.
- The model needs to be composed of 4 normalizing flow (NF) layers, each using an MLP with 5 linear layers and Relu activation functions.
- Before each NF layer, split the input vector into two halves and use one half as the input to the layer, and the other half as the values that are copied over to the next layer (this is the "coupling layers" we saw in class). Alternate the roles of the two halves in consecutive NF layers.
- At the end of each NF layer, multiply the output with a scaling factor $exp(s_i)$ where s_i are learned parameters in the dimension of the data, for each NF layer i.
- Implement a *forward* function that maps input data points x into their latent representation z. The function needs to return the mapped latents z, and the log determinant of the Jacobian of the mapping.
- Implement an *inverse* function that maps latent vectors z into data samples x. The function needs to return the data samples.
- Implement a function *calc_likelihood* that will be used to train the model.
- Implement a function *sample* that will be used to generate samples.
- Use a Standard Gaussian distribution $\mathcal{N}(0, I)$ as the prior over z.

Assignments

1. Implement a normalizing flow model as described above and train it on the MNIST data. Report the log-likelihood for the test set, curves showing the log-likelihood during training (for both training and validation data). After training, generate a few samples and show the results. You should be able to train models where most of the samples look like valid digits.

2. Show the transformation of samples from the latent distribution z to the data distribution. For 10 different samples show 5 images starting from the sample of the latent z, gradually transforming to samples at the output of each NF layer, up to the final output which is the sample of x. To do this you will need to change the output of the functions you implemented such that they output the values of intermediate layers. Show the results as a grid of 10 rows and 5 columns.

3. In this question you will generate latent traversals, like in exercise 5. Use the model that you trained, and chose two images of different digits from the dataset. For each image, compute the corresponding latent value z. This will be the 'representation' of the image. Now given the two latent representations z_1, z_2, compute a linear path between them with a set of 10 points along the way. For each of those 10 points, z^i, compute the reconstruction sample $p(x|z^i)$ (you can use the mean of this distribution like before), and show the resulting images in order. You should see how the first digit transforms into the second digit. Repeat this for 3 different pairs of images.

You should submit a colab notebook that contains the text of the answers, figures and code for all the above questions. Add a few sentences that describe what you did and the results you got.