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

# Exercise 4: Auto-regressive models

*Dan Rosenbaum*

**Due: 16 June 2024**

In this exercise we will use the code we saw in class:

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

Make sure you understand all parts of the code and that you can run it both using the pretrained model, and for training a new model. The code is based on Pytorch-Lightning, which is a framework for training neural networks. Within this framework, you will need to use functions that log the training loss and validation/test loss so you can submit plots of their values during training.

The model contains some tricks that we did not discuss in class, including "Gated Convolutions" and "Dilation".

**1.** Implement a version of the model without the gated convolution and dilation, that uses several layers of masked convolutions using the vertical / horizontal stacks as we discussed. The way this works is that the output of each layer contains two components: (1) the output of the vertical stack, and (2) the output of the horizontal stack (this separation is already implemented in the code). After all the convolution layers, you will need to add the output of both stacks, and then the final layer should be a softmax over the 256 values for each pixel (this is also already implemented). These updates should result in a faster model to train. Try to find a configuration that achieves reasonable results (you can change number of layers, width, kernel sizes, etc.). Submit the code, plots with the negative log likelihoods both for training and for validation / test during training, and samples from the trained model.

**2.** Show what happens if the convolutions are not causal, i.e. if the prediction of pixel $i$, depends on the input pixel $i$. Submit the same plots as for question (1), and samples from the model.

**3.** The following requires computing several distributions or sampling from the model. Some of these are easy to compute and some are hard. For each of the following explain if they are easy or hard. For the easy ones, implement the computation and submit the results. For the hard ones, propose a way they can be approximated, and implement an approximated method (the implementation can be very approximate if the computation is too costly). For this you can choose to use the provided pretrained model, or the model you trained in question (1). Show results for a few different conditioning values.

a) The marginal probability of the first pixel.

b) The marginal probability of the middle pixel (pixel 14, 14).

c) The conditional probability of the middle pixel, given the values of all pixels above and to the left of it.

d) The conditional probability of the middle pixel, given the values of all pixels below and to the right of it.

e) The conditional probability of the middle pixel, given the values of all other pixels in the image.

f) Samples of the bottom half of an image, given the top half of an image.

g) Samples of the top half of an image, given the bottom half of an image.

You should submit a colab notebook that contains the text of the answers, figures and code for all the above questions.