HHU WS1920 Generative Models (MSC) Tuesday, 22. October 2019
Prof. Dr. Markus Kollmann DUE 11:59 am Tuesday, 05. November 2019

Exercise set #2

Please provide your documented solution (along with comments/markdown) in an Jupyter notebook. Use Python3 and for the optimization and machine learning tasks always Pytorch. You can submit via sciebo: `https://uni-duesseldorf.sciebo.de/s/QTGaw2U4g71JyBF` . Submit everything in one zip file with your full name plus exercise number for the filename as an indicator, e.g. max_mustermann_2.zip. Exercise material is drawn from UC Berkeley, CS294-158.

Colored MNIST
Now, you will train more powerful models on high-dimensional data: colored MNIST digits. The dataset you are provided has 60,000 training images and 10,000 test images. Each image has height H = 28 and width W = 28, with C = 3 channels (red, green, and blue). Let $x = (x_1, ..., x_{HW})$ represent one image, where each $x_i = (x_i^1, ...x_i^C)$ is a vector of channel values for one pixel. Each x_i^C will take on a value in $\{0, 1, 2, 3\}$.

1. **PixelCNN**
   First, design an autoregressive model of the form:

$$p_\theta(x) = \prod_{i=1}^{HW} p_\theta(x_i|x_{1:i-1}) = \prod_{i=1}^{HW}\prod_{c=1}^{C} p_\theta(x_i^C|x_{1:i-1})$$

   i.e. a model which is autoregressive over space, but factorized over channels. Use the masked PixelCNN architecture to implement spatial dependencies. We recommend you to follow the architecture from the paper 'Pixel Recurrent Neural Networks' (Aaron van den Oord, Nal Kalchbrenner, Koray Kavukcuoglu) and see the specifications in Table 1 in the paper link: `https://arxiv.org/abs/1601.06759` , but you are of course free to experiment in your network architecture design. Start with a 7x7 masked convolution of type A, followed by 12 layers of residual blocks containing masked convolutions of 1x1, 3x3 and 1x1 of type B (Figure 5 in the paper). The logits for the softmax layer are then obtained after two 1x1 masked convolution layers of type B.
   Provide these deliverables for the model:

   (a) Plot training and validation losses over time and report the test set performance of your final model, just as in the previous exercise. Report all negative log likelihoods in bits per dimension.

   (b) Generate and display 100 samples from your model. You may want to scale the values 0, 1, 2, 3 to 0, . . . , 255 for display purposes.

   (c) Visualize the receptive field of the conditional pixel distribution at (14, 14, 0). To do so, compute the gradient of the log probability of the model with respect to the input image at randomly initialized parameters. Turn the resulting gradient into a visualization by com- puting its elementwise absolute value and taking the maximum over channels – this should transform the 28x28x3 gradient into a 28x28 image.

100 points