Fachprojekt Machine Learning
Sebastian Konietzny, Marc Höftmann

Wednesday, 12. June 2024
Group 1 & 2 **DUE Wednesday, 3. July 2024**
Group 3 & 4 **DUE Wednesday, 10. July 2024**

Task #4

Feel free to ask questions via our chat program Mattermost. **Keep the deadline in mind!**

Your task is to implement the next paper (PixelCNN, VQ-VAE, BiGAN or RealNVP). The Fachprojekt is different from other modules, so you have to coordinate mostly by yourself. We recommend to use at least a chat program like Discord or Mattermost. Also, feel free to program the code together, as it makes it usually easier. But stick to the contribution guidelines below.

Very important: Log your contribution on GitHub!

1. Journal: "20.04. - Marc: Implemented X","02.05. - Sebastian: Add Y"
2. Commits: If you do pair-programming then split code in multiple commits. Each commit with a **different** GitHub account!

1. Image completion (G1)

Create the file `completion.py` and add the function `complete(x, num_samples, ...)` which generates multiple different images for a partially masked image `x` (analogous to Figure 1 in the PixelCNN paper). Use it with train and test images.

2. Latent space interpolation (G2, G3, G4)

Create the file `interpolation.py` and add the function `interpolate(z, z', step_size, ...)` which creates a video (`.mp4`, `.gif`, etc.) of images that show the interpolation on a linear trajectory between z and z'. Choose an appropriate step size (number of generated images).

Feel free to be creative. For instance, you could process two images (akin to the training set) through the encoder and then apply `interpolate` to the resultant latent variables. Your generative model should be versatile enough to be trained on any dataset. You are also welcome to experiment with other datasets, such as CelebA.

3. Modifications

Extend the code of your finished paper and modify it for the new paper:

- Group 1: MADE → PixelCNN
- Group 2: VAE → VQ-VAE
- Group 3: NICE → RealNVP
- Group 4: (DC)GAN → BiGAN

Adhere to Task #3 guidelines and avoid overwriting prior code. Implement the new paper for easy comparison with the previous one, allowing a simple switch between them or opt for a separate implementation.

4. Final presentation

Prepare a final presentation of your work. You should present your results and the differences between the two papers. Especially, you should address difficulties and problems that emerged during implementation.