Exercise Sheet #10

Due: 23.01.22 11h00

Roman Remme

General Regulations.

- Please hand in your solutions in groups of three people. A mix of attendees from Monday and Tuesday tutorials is fine.
- Your solutions to theoretical exercises can be either handwritten notes (scanned), or typeset using LaTeX. In case you hand in handwritten notes, please make sure that they are legible and not too blurred or low resolution, otherwise we might not correct your submission.
- For the practical exercises, the data and a skeleton for your jupyter notebook are available at `https://github.com/hci-unihd/mlph_sheet10`. Always provide the (commented) code as well as the output, and don't forget to explain/interpret the latter. Please hand in both the notebook (`.ipynb`), as well as an exported pdf.
- Submit all your files in the Übungsgruppenverwaltung, only once for your group of three.

1 Hands-on Diffusion Models

Interpolation between samples from a distribution can be of interest both scientifically and for fun (for the latter see the gifs in this blog post[1]). To make training feasible and visualizations straightforward, in this task you will train a diffusion model yourself on some 2D toy datasets and look into interpolation in the latent space. You will not have to start from scratch, a jupyter notebook that already implements and explains most steps is provided in the repository. I recommend you to run it on google colab.

(a) Train the model on the 'V' dataset. Given a point along the diffusion process and the time step, it tries to estimate the noise that was added to the original sample. Create scatter plots of the true versus predicted noise, for each of the two dimensions of the data, coloring the points by the time step in the diffusion process. Use the reverse process to sample 1000 points from the distribution, and visualize the backwards trajectory in a video (submit small plots of the first, last and two intermediate frames) (4 pts)

(b) Create a linear interpolation with 500 points between (-1, 2) and (-1, -2) in the latent space of the diffusion model. As before, perform the reverse process and visualize the backwards trajectory, but connect neighboring points in your interpolation. Explain the result. (2 pts)

(c) One approach to get a sensible interpolation is to use the identical noise in the reverse process for all samples. Implement this and repeat the experiment from part (b). (2 pts)

(d) Implement the deterministic reverse process from denoising Diffusion Implicit Models (DDIM, see section 4 in [SME20]). Visualize the reverse process both on sample of 1000 points from the distribution as in (a), and on the linear interpolation from (b). Compare the results to the previous methods. (4 pts)

(e) Repeat the experiments in interpolation from (b) to (d) for a linear interpolation between (-2, 0) and (2, 0). Is the interpolation as expected? Do you have an idea on how to improve the interpolation? (3 pts)

(f) **Bonus:** Adapt the notebook (data handling, model, visualizations) to also work on image data, and repeat the previous tasks on the MNIST or CIFAR10 dataset (you can find both in `torchvision.datasets`). (5 pts)

[1] `https://keras.io/examples/generative/random_walks_with_stable_diffusion/`

2 Implementing Transformers with a Transformer

Using ChatGPT[2] only, try and implement a multi-head attention layer. Demonstrate that the code runs (if you ask it nicely, ChatGPT might also help you write some tests). (3 pts)

3 ML in your Life

In case you will be working on a machine learning problem (excluding this or other lectures) in the next couple of months, which methods do you plan on using in which context? In case you do not have ML plans for the future, what are your past experiences with applying machine learning methods? (2 pts)

4 Bonus: Subspace Diffusion Models

For Generative Adversarial Networks (GANs), popular architectures to generate high resolution images relied on a multi-stage approach: In the first stage an image with a very low resolution is generated and iteratively up-sampled and refined in the subsequent stages. A similar process has been implemented in [Jin+22].

(a) In general, can you come up with a method (distinct from [Jin+22]) to make diffusion models work between spaces of unequal dimensionality? (3 pts)

(b) Implement your approach (5 pts)

(c) Write a paper draft on your method (20 pts + ICML[3] submission?)

References

[SME20] Jiaming Song, Chenlin Meng, and Stefano Ermon. "Denoising Diffusion Implicit Models". In: *International Conference on Learning Representations*. 2020.

[Jin+22] Bowen Jing et al. "Subspace diffusion generative models". In: *arXiv preprint arXiv:2205.01490* (2022).

[2] https://openai.com/blog/chatgpt/. Sometimes the servers are overloaded, but it seems like you have a good chance to reach it in the early morning.

[3] International Conference on Machine Learning, https://icml.cc/