Exercise Sheet #10

Due: 22.01.2024 11h00

Jan Pawlowski, Tilman Plehn
Jonas Spinner

General Regulations.

- Please hand in your solutions in groups of three people. A mix of attendees from different tutorials is fine. We will not correct submissions from single students.
- Your solutions to theoretical exercises can be either handwritten notes (scanned to `pdf`), typeset using LaTeX, or directly in the jupyter notebook using Markdown.
- For the practical exercises, the data and a skeleton for your jupyter notebook are available at https://github.com/heidelberg-hepml/mlph2023-Exercises. Always provide the (commented) python code as well as the output, and don't forget to explain/interpret the latter, we do not give points for code that does not run. Please hand in both the notebook (`.ipynb`) and an exported pdf. Combine your the pdfs from theoretical and notebook exercises into a single pdf.
- Submit all your files in the Übungsgruppenverwaltung, only once for your group of three. Please list all names and tutorial numbers to simplify things for us.
- Starting from this sheet we put the tag **exam-like** to some exercises to give you a felling for how exercises in the exam might look like.

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) (2 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 https://arxiv.org/pdf/2106.03802.pdf). 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. (2 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? (2 pts)

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

2 Event Generation with Invertible Neural Networks

(a) (exam-like) Derive the loss function of a normalizing flow starting from a log-likelihood loss. (1 pts)

(b) Automatic differentiation allows us to directly compute the jacobian used in the loss function of a normalizing flow for any architecture, but this approach is numerically very inefficient as the number of operations is $\mathcal{O}(n^3)$ for n-dimensional data. Explain how Invertible Neural Networks (INNs) overcome this limitation and compute jacobians with $\mathcal{O}(n)$ operations. (2 pts)

(c) Construct and train an INN on the event generation dataset. You can directly use the improved pre-processing of sheet 9, exercise 3. Visualize the distributions of the generated events. (2 pts)

(d) INNs are invertible, i.e. they can both generate events and access their likelihood with the same computational effort. Compute these likelihoods and visualize the distributions of the 10% events with the largest likelihoods. (1 pts)

3 Quickies

(a) (exam-like) Write down a loss function that can be used for (i) regression, (ii) classification. (1 pts)

(b) (exam-like) Explain the difference between forward and reverse automatic differentiation. Which of the two modes is used for backpropagation? (1 pts)

(c) (exam-like) Explain the difference between supervised and unsupervised machine learning methods. Give an example for both methods. (1 pts)

(d) (exam-like) Explain what point cloud data is. Discuss challenges when working with point clouds and name one architecture that tackles them. (1 pts)

(e) (exam-like) List 3 generative architectures and write down their loss functions in the most basic form. (1 pts)