Machine Learning and Physics
Winter Semester 2025/26

Exercise Sheet 11

Due: 26.01.26 14h00

Christof Gehrig, Gerrit Gerhartz

General Regulations.

- Please hand in your solutions in groups of two (preferably from the same tutorial group).
- Your solutions to theoretical exercises can be either handwritten notes (scanned), or typeset using LaTeX. For scanned handwritten notes, please ensure they are legible and not blurry.
- For the practical exercises, always provide the (commented) code as well as the output, and don't forget to explain/interpret the latter.
- Please hand in a **single PDF** that includes both the exported notebook and your solutions to the theoretical exercises. Submit the PDF to the Übungsgruppenverwaltung once per group, making sure to include the names of both group members in the submission.
- You can find all the data in the GitHub Repository.

1 Denoising Score Matching

Show that for a Gaussian perturbation $x_t = x_0 + \sigma\epsilon$, where $\epsilon \sim \mathcal{N}(0, I)$, the score function $\nabla \log p(x_t \mid x_0)$ is proportional to the added noise

$$\nabla \log p(x_t \mid x_0) = -\frac{\epsilon}{\sigma}.$$

(2 pts)

2 Diffusion Model Implementation

In this exercise we will implement a full diffusion model, train it, and sample from it. The exercise is split up into parts such that you can implement everything step by step.

(a) We will train the diffusion model on the so-called two half moons toy dataset. How you can generate this dataset is shown in the Jupyter notebook. Plot samples from the two half moons dataset.

(1 pt)

(b) We will model the forward diffusion process as a so-called variance-preserving SDE:

$$\mathrm{d}x_t = \frac{1}{2}\beta(t)x_t\mathrm{d}t + \sqrt{\beta(t)}\mathrm{d}W_t$$

with a linear noise scheduler

$$\beta(t) = \beta_{\text{min}} + t(\beta_{\text{max}} - \beta_{\text{min}}).$$

Without giving you the proof, you can calculate that this SDE has a closed-form marginal

$$x_t = \alpha(t)x_0 + \sigma(t)\epsilon, \quad \epsilon \sim \mathcal{N}(0, I)$$

where

$$\alpha(t) = \exp\left(-\frac{1}{2}\int_0^t \beta(s)\mathrm{d}s\right), \quad \sigma^2(t) = 1 - \alpha^2(t).$$

Implement the three functions $\beta(t), \alpha(t)$ and $\sigma(t)$. What is the role of β_{min} and β_{max}? (3 pts)

Hint: You can analytically calculate the integral in $\alpha(t)$.

(c) Now we need to define our model $s_\theta(x_t, t)$, which we use to predict the score. We will use a simple MLP with size $[3, 64, 64, 2]$ and activation function `Tanh`. As an input, we take the concatenation between the 2D input and the diffusion time variable. Implement this model. (2 pts)

(d) Implement the training loop using Adam and a learning rate of 10^{-3}. You can use a batch size of 256, and we expect it needs to train for around 100000 steps. The loss you need to implement is the following denoising score matching loss:

$$\mathcal{L} = \mathbb{E}_{x_0,t,\epsilon}\left[\left\|s_\theta(x_t, t) + \frac{\epsilon}{\sigma(t)}\right\|^2\right].$$

Train the model using the closed-form marginal from (b) and sample ϵ, t accordingly. (4 pts)

Hint: Numerically, it is better to not sample the time from $\mathcal{U}(0, 1)$, but from $\mathcal{U}(\varepsilon, 1)$ with ε a small constant around 10^{-3}. Furthermore, it may be beneficial to also clamp the $\sigma(t)$ to a small number around 10^{-5} to avoid division by zero errors.

(e) Lastly, implement the sampling of the model using the following reverse-time SDE

$$\mathrm{d}x_t = \left[-\frac{1}{2}\beta(t)x_t - \beta(t)s_\theta(x, t)\right]\mathrm{d}t + \sqrt{\beta(t)}\mathrm{d}\tilde{W}_t.$$

Sample the model using 1000 discretization steps and plot the result. What can you see? (4 pts)

Hint: The SDE can be discretized using the Euler-Maruyama method. Therefore, integrating from $t = 1$ to $t = \epsilon$ (because of the training in d) with a timestep of $\Delta t = \frac{1-\epsilon}{1000}$ ($t_i = i\Delta t$), can be calculated by:

$$x_{i-1} = x_i + \left(\frac{1}{2}\beta(t_i)x_i + \beta(t_i)s_\theta(x_i, t_i)\right)\Delta t + \sqrt{\beta(t_i)\Delta t}\varepsilon, \quad \varepsilon \sim \mathcal{N}(0, I),$$

with a starting point of $x_1 \sim \mathcal{N}(0, I)$.

3 Equivariant Energy-Based Diffusion

In molecular machine learning, we often work with diffusion models, which are based on energies that correspond to the Boltzmann distribution

$$p_t(\mathbf{x}) \sim e^{-\beta U_t(\mathbf{x})},$$

where $\mathbf{x} \in \mathbb{R}^{N\times 3}$ are the positions of all the atoms in the system of interest. (Here only one atom type is used.)

(a) Show that if one assumes Boltzmann distributions, the score is equivalent to a force (2 pts)

(b) If the energy is invariant to rotations, show that the force is equivariant to the same rotations. (2 pts)