Diffusion Models (Reverse Process)

Dr. Alireza Aghamohammadi

Decoder

- ❖ We model a series of probabilistic transitions, starting from the latent variable z_T, moving backward step by step through $z_{T-1}, z_{T-2}, \ldots$, until we finally recover the data x.
- ❖ The true reverse transitions $q(z_{t-1} \mid z_t)$ in the diffusion process are complex and difficult to compute exactly.
- ❖ When the noise variance β_t is very small ($\beta_t \ll 1$), we can approximate these transitions as normal distributions:

$$
\begin{aligned}
P_r(z_T) &= \mathcal{N}(z_T; 0, \mathbf{I}) \\
P_r(z_{t-1} \mid z_t, \phi) &= \mathcal{N}(z_{t-1}; f(z_t, \phi, t), \beta_t \mathbf{I}) \\
P_r(x \mid z_1, \phi) &= \mathcal{N}(x; f(z_1, \phi, 1), \beta_1 \mathbf{I})
\end{aligned}
$$

  where $f(z_t, \phi, t)$ is a neural network parameterized by ϕ.
- ❖ Once the model is trained, sampling is straightforward:
  - ❑ First, draw z_T from $P_r(z_T)$.
  - ❑ Then sample each previous latent variable sequentially, e.g., z_{T-1} from $P_r(z_{T-1} \mid z_T, \phi)$, z_{T-2} from $P_r(z_{T-2} \mid z_{T-1}, \phi)$, and so on.
  - ❑ Finally, generate the data x from $P_r(x \mid z_1, \phi)$.

Training the Decoder

- ❖ To train the neural network, we need to define an objective function. The natural choice is the likelihood function, which for a data point x is expressed as:

$$P_r(x \mid \phi) = \int \cdots \int P_r(x, z_1, \ldots, z_T \mid \phi) \, \mathrm{d}z_1 \cdots \mathrm{d}z_T$$

- ❖ The joint probability distribution for x and the latent variables $z_1, \ldots, z_T$ is given by:

$$P_r(x, z_1, \ldots, z_T \mid \phi) = P_r(z_T) \left[\prod_{t=2}^{T} P_r(z_{t-1} \mid z_t, \phi) \right] P_r(x \mid z_1, \phi)$$

- ❖ We train the model by maximizing the log-likelihood of the training data $\{x_i\}$ with respect to the parameters ϕ:

$$\hat{\phi} = \arg\max_{\phi} \left[\sum_{i=1}^{I} \log P_r(x_i \mid \phi) \right]$$

- ❖ However, directly maximizing this is difficult due to the complex integrals involving the neural network functions.