Diffusion Models (Implementation)

Dr. Alireza Aghamohammadi

Diffusion model training

Input: Training data $\mathcal{D} = \{x_n\}$, Noise schedule $\{\beta_1, \ldots, \beta_T\}$
Output: Network parameters ϕ
Step 1: Calculate alphas from betas:
for $t \in \{1, \ldots, T\}$ **do**
    $\alpha_t \leftarrow \prod_{\tau=1}^{t}(1 - \beta_\tau)$
end
Step 2: Repeat until converged:
while *Not converged* **do**
    Sample a data point x from $\mathcal{D}$
    Sample a point t from $\{1, \ldots, T\}$
    Sample a noise vector ϵ from $\mathcal{N}(\epsilon; 0, \mathbf{I})$
    Evaluate noisy latent variable: $z_t \leftarrow \sqrt{\alpha_t}x + \sqrt{1 - \alpha_t}\epsilon$
    Compute loss term: $\mathcal{L}(\phi) \leftarrow \|g(z_t, \phi, t) - \epsilon\|^2$
    Take optimizer step
end
Step 3: Return ϕ

Diffusion model sampling

Input: Trained denoising network $g(\mathbf{z}, \phi, t)$, Noise schedule $\{\beta_1, \ldots, \beta_T\}$
Output: Sample vector $\mathbf{x}$ in data space
Step 1: Sample from final latent space: $\mathbf{z}_T \sim \mathcal{N}(\mathbf{z_T}; 0, \mathbf{I})$
Step 2: Iterate over time steps:
for $t \in T, \ldots, 2$ **do**
  $\alpha_t \leftarrow \prod_{\tau=1}^{t}(1-\beta_\tau)$
  Evaluate network output: $f(\mathbf{z}_t, \phi, t) \leftarrow \frac{1}{\sqrt{1-\beta_t}}\left(\mathbf{z}_t - \frac{\beta_t}{\sqrt{1-\alpha_t}} g(\mathbf{z}_t, \phi, t)\right)$
  Sample a noise vector: $\boldsymbol{\epsilon} \sim \mathcal{N}(\boldsymbol{\epsilon}; 0, \mathbf{I})$
  Add scaled noise: $\mathbf{z}_{t-1} \leftarrow f(\mathbf{z}_t, \phi, t) + \sqrt{\beta_t}\boldsymbol{\epsilon}$
end
Step 3: Final denoising step: $\mathbf{x} = \frac{1}{\sqrt{1-\beta_1}}\left(\mathbf{z}_1 - \frac{\beta_1}{\sqrt{1-\alpha_1}} g(\mathbf{z}_1, \phi, 1)\right)$
Return: $\mathbf{x}$