Advanced Machine Learning:

Deep Generative Models

Summary

Lecturer: Prof. Dr. Stephan Günnemann

www.cs.cit.tum.de/daml

Summer Term 2023

Data Analytics and
Machine Learning

Roadmap

- Deep Generative Models
  1. Introduction
  2. Normalizing Flows
  3. Variational Inference
  4. Generative Adversarial Networks
  5. Denoising Diffusion
  6. **Summary**

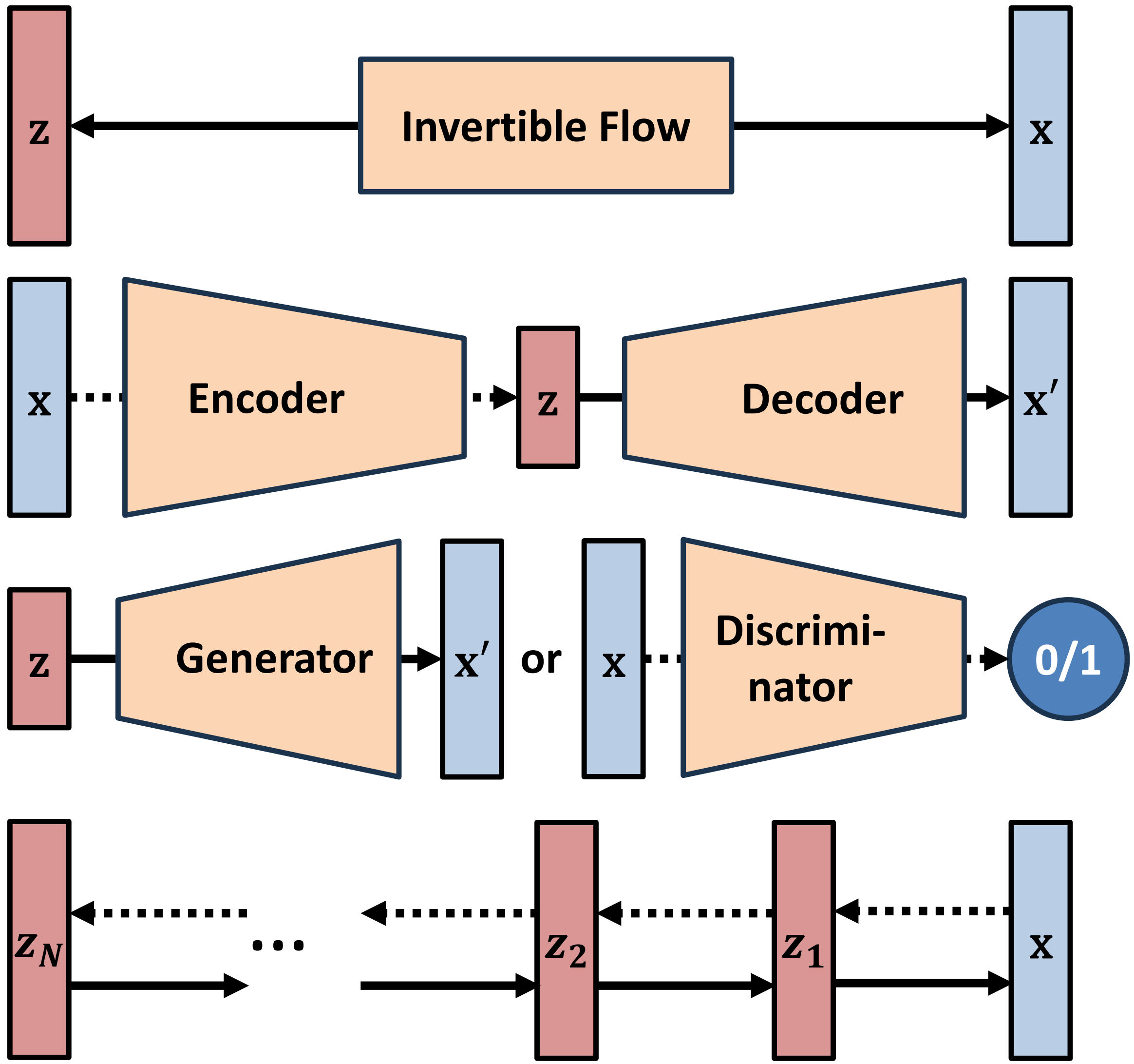
Overview

Normalizing Flow

VAE

GANs

Diffusion Models



Generating New Samples

- Normalizing flows
  1. Sample $\boldsymbol{z} \sim p(\boldsymbol{z})$ — $p(\boldsymbol{z})$ is the base distribution
  2. Compute $\boldsymbol{x} = f_{\boldsymbol{\theta}}(\boldsymbol{z})$ — $f_{\boldsymbol{\theta}}$ is an invertible transformation

- Variational Autoencoder
  1. Sample $\boldsymbol{z} \sim p(\boldsymbol{z})$ — $p(\boldsymbol{z})$ is the prior on $\boldsymbol{z}$
  2. Compute $\boldsymbol{\theta} = f_{\boldsymbol{\psi}}(\boldsymbol{z})$ — $f_{\boldsymbol{\psi}}$ is the decoder
  3. Sample $\boldsymbol{x} \sim p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})$ — $p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})$ is the predefined conditional likelihood

- Generative Adversarial Network
  1. Sample $\boldsymbol{z} \sim p(\boldsymbol{z})$ — $p(\boldsymbol{z})$ is the noise distribution
  2. Compute $\boldsymbol{x} = f_{\boldsymbol{\theta}}(\boldsymbol{z})$ — $f_{\boldsymbol{\theta}}$ is the generator network

- Denoising Diffusion
  1. Sample $\boldsymbol{z}_N \sim p(\boldsymbol{z}_N)$ — $p(\boldsymbol{z}_N)$ is the prior on $\boldsymbol{z}_N$
  2. Sample $\boldsymbol{z}_{N-1} \sim p(\boldsymbol{z}_{N-1}|\boldsymbol{z}_N)$ — $p(\boldsymbol{z}_{N-1}|\boldsymbol{z}_N)$ is the learned reverse process
  3. Sample $\boldsymbol{x}_0 \sim p(\boldsymbol{x}_0|\boldsymbol{z}_1)$ — $p(\boldsymbol{x}_0|\boldsymbol{z}_1)$ decodes $\boldsymbol{z}_1$ into $\boldsymbol{x}_0$

Likelihood Computation

- Normalizing flows
  - Use the change of variables formula

$$p_{\boldsymbol{\theta}}(\boldsymbol{x}) = p\left(f_{\boldsymbol{\theta}}^{-1}(\boldsymbol{x})\right)\left|\det\left(\frac{\partial f_{\boldsymbol{\theta}}^{-1}(\boldsymbol{x})}{\partial \boldsymbol{x}}\right)\right|$$

- Variational Autoencoder & Denoising Diffusion
  - Marginalize out the latent variable

$$p_{\boldsymbol{\theta}}(\boldsymbol{x}) = \int p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})p(\boldsymbol{z})d\boldsymbol{z}$$

- Generative Adversarial Network
  - Compute partial derivatives of the CDF of $\boldsymbol{x}$

$$p_{\boldsymbol{\theta}}(\boldsymbol{x}) = \frac{\partial}{\partial x_1}\dots\frac{\partial}{\partial x_d}\int_{\{f_{\boldsymbol{\theta}}(\boldsymbol{z})\leq\boldsymbol{x}\}} p(\boldsymbol{z})d\boldsymbol{z} = \frac{\partial}{\partial x_1}\dots\frac{\partial}{\partial x_d}\Pr(f_{\boldsymbol{\theta}}(\boldsymbol{z}) \leq \boldsymbol{x})$$

Optimization Objective

- Normalizing Flows (for density estimation)
  - Maximum likelihood

$$\max_{\boldsymbol{\theta}} \log p_{\boldsymbol{\theta}}(\boldsymbol{x})$$

- Variational Autoencoder
  - Evidence Lower BOund (ELBO)

$$\max_{\boldsymbol{\psi},\,\boldsymbol{\lambda}} \mathbb{E}_{\boldsymbol{z}\sim q_{\boldsymbol{\phi}}(\boldsymbol{z})}\left[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z}) + \log p(\boldsymbol{z}) - \log q_{\boldsymbol{\phi}}(\boldsymbol{z})\right]$$
$$\text{where } \boldsymbol{\phi} = g_{\boldsymbol{\lambda}}(\boldsymbol{x}) \text{ and } \boldsymbol{\theta} = f_{\boldsymbol{\psi}}(\boldsymbol{z})$$

- Generative Adversarial Network
  - Minimax optimization of ratio loss & generative loss

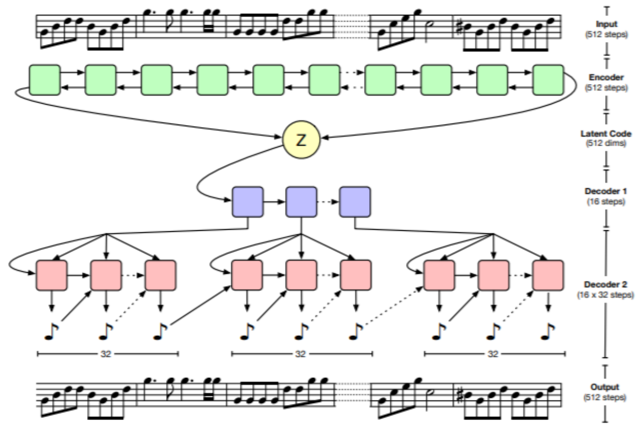
$$\min_{\boldsymbol{\theta}} \max_{\boldsymbol{\phi}} \pi \mathbb{E}_{p^*(\boldsymbol{x})}\left[\log D_{\boldsymbol{\phi}}(\boldsymbol{x})\right] + (1-\pi)\mathbb{E}_{p(\boldsymbol{z})}\left[\log\left[1 - D_{\boldsymbol{\phi}}(f_{\boldsymbol{\theta}}(\boldsymbol{z}))\right]\right]$$

- Denoising Diffusion
  - Simplified loss derived from ELBO

$$\min_{\boldsymbol{\theta}} \mathbb{E}_{n,\,\boldsymbol{x}_0,\,\boldsymbol{\epsilon}}\left[\left\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_{\boldsymbol{\theta}}\left(\sqrt{\bar{\alpha}_n}\boldsymbol{x}_0 + \sqrt{(1-\bar{\alpha}_n)}\boldsymbol{\epsilon}, n\right)\right\|^2\right]$$

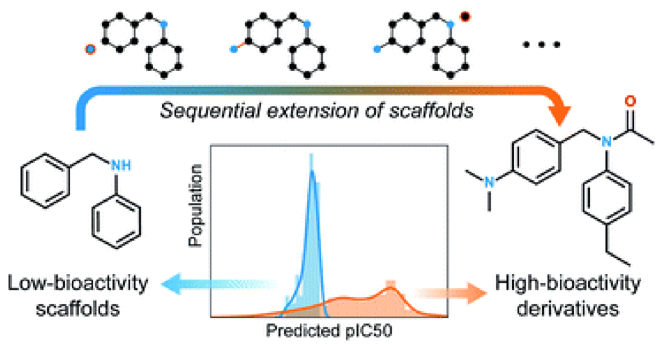
Outlook

- (Deep) Generative models are seen as a promising approach towards "understanding the world"
- Some applications require conditional generation, i.e. modeling $p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{y})$
  - Missing data imputation
  - "Translation" tasks (e.g. text-to-speech)
  - Controllable generation
- Modeling non-standard data types
  - Graphs, sequences, waveforms



[Roberts+, 2019]

[Karras+, 2018]



[Lim+, 2019]

References for Figures

- Karras et al. 2019, https://github.com/NVlabs/stylegan
- Lim et al. 2019, Scaffold-based molecular design with a graph generative model
- Roberts et al. 2019, A Hierarchical Latent Vector Model for Learning Long-Term Structure in Music, https://arxiv.org/pdf/1803.05428.pdf