CSE6243: Advanced Machine Learning **Fall 2023**

Lecture 13: Diffusion Models

Lecturer: Bo Dai *Scribes: Ozgur Kara*

Note: *LaTeX template courtesy of UC Berkeley EECS Department.*

Disclaimer: *These notes have not been subjected to the usual scrutiny reserved for formal publications. They may be distributed outside this class only with the permission of the Instructor.*

13.1 New Content

There is a long history and progress of generative modelling, and here are some specific examples of the approaches:

- Variational Autoencoders [1]
- Generative Adversarial Networks [2]
- PixelCNN [3]
- BigGAN [4]
- Imagen [5]

But we can categorize deep generative models into Variational Autoencoders, Autoregressive Models, Normalizing Flows, Energy-Based Models, and finally, diffusion models. We are going to cover the latter throughout this lecture note.

Before diving into the technical details, these models are widely used in applications such as AI video generation, super-resolution, inpainting, AI art, etc.

13.1.1 Recap of Variational Autoencoders

Variational autoencoder is a latent variable model where we aim to model an observed target distribution $p(\mathbf{x})$, assuming we have a hidden distribution, referred to as latent variables, denoted as $p(\mathbf{z})$. We assume that there is a mapping from $\mathbf{z}$ to $\mathbf{x}$.

$$z \sim \mathcal{N}(\boldsymbol{z}; 0, \boldsymbol{I}) \tag{13.1}$$

$$p(\boldsymbol{x}|\boldsymbol{z}) = \mathcal{N}(\boldsymbol{x}; \boldsymbol{f}(\boldsymbol{z})) \tag{13.2}$$

$$p(\boldsymbol{x}) = \int_{\boldsymbol{z}} p(\boldsymbol{x}|\boldsymbol{z})p(\boldsymbol{z})d\boldsymbol{z} \tag{13.3}$$

However, the difficulty lies in the optimization, as the marginal likelihood $p(\mathbf{x})$ is intractable. Therefore, we cannot optimize it directly using maximum likelihood.

Instead, we introduce an inference model $q(\boldsymbol{z}|\boldsymbol{x})$, enabling us to efficiently optimize the log-likelihood through the Evidence Lower Bound (ELBO).

$$log(p(\boldsymbol{x})) \geq ELBO(\boldsymbol{x}) = \mathbb{E}_{q(\boldsymbol{z}|\boldsymbol{x})} log \frac{p(\boldsymbol{x}, \boldsymbol{z})}{q(\boldsymbol{z}|\boldsymbol{x})} \tag{13.4}$$

Hence, we optimize $q(\boldsymbol{z}|\boldsymbol{x})$ and $p(\boldsymbol{x}, \boldsymbol{z})$ jointly with respect to ELBO.

Flat VAEs suffer from simple priors; hence, better likelihoods are achieved with hierarchies of latent variables. However, there are some challenges with VAEs:

- Optimization can be challenging for large models.
- The ELBO enforces an information bottleneck at the latent variables 'z', which are typically low-dimensional. This makes VAE optimization prone to bad local minima.
- Posterior collapse is a dreaded bad local minimum where the latents do not transmit any information.

13.1.2 Denoising Diffusion Models

Denoising diffusion models consist of two processes:

- **Forward Diffusion**: Here, given a data point sampled from a real data distribution, $\boldsymbol{x}_0 \sim q(\boldsymbol{x}_0)$, the forward diffusion process is defined as the addition of a small amount of Gaussian noise to the sample in T steps, producing a sequence of noisy samples $\boldsymbol{x}_1, \cdots, \boldsymbol{x}_T$. Note that the variance schedules are predefined and denoted with $\{\beta_t \in (0,1)\}_{t=1}^T$. The inference distributions are as follows:

$$q(x_t|x_{t-1}) = \mathcal{N}(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t I) \tag{13.5}$$

$$q(x_{1:T}|x_0) = \prod_{t=1}^{T} q(x_t|x_{t-1}) \tag{13.6}$$

  , which are similar to the inference model in hierarchical VAEs. To sample a noisified version of the original image at timestep t, denoted with $\boldsymbol{x}_t$, we can perform the following operations to obtain the equation for sampling:

$$x_t = \sqrt{\alpha_t}x_{t-1} + \sqrt{1-\alpha_t}\epsilon_{t-1} \tag{13.7}$$

$$= \sqrt{\alpha_t\alpha_{t-1}}x_{t-2} + \sqrt{1-\alpha_t\alpha_{t-1}}\bar{\epsilon}_{t-2} \tag{13.8}$$

$$= ... \tag{13.9}$$

$$= \sqrt{\bar{\alpha}_t}x_0 + \sqrt{1-\bar{\alpha}_t}\epsilon \tag{13.10}$$

$$\alpha_t = 1 - \beta_t, \bar{\alpha}_t = \prod_{t=1}^{T} \alpha_t \tag{13.11}$$

  Note that the schedules are designed such that $q(\boldsymbol{x}_T|\boldsymbol{x}_0) \approx \mathcal{N}(\boldsymbol{x}_T; \boldsymbol{0}, \boldsymbol{I})$

- **Reverse Diffusion**: Here, if we reverse the above process and iteratively sample using $q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)$ starting from $\boldsymbol{x}_T \sim \mathcal{N}(\boldsymbol{x}_T; \boldsymbol{0}, \boldsymbol{I})$, we will be able to reconstruct the true sample from Gaussian noise input. However, the term $q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)$ is intractable. So, to approximate it, we can use a normal distribution if β_t is small in each forward diffusion step.

For approximation, we define a trainable network, typically chosen as the U-net architecture and parametrized with θ:

$$p_(\boldsymbol{x}_T) = \mathcal{N}(\boldsymbol{x}_T; \mathbf{0}, \boldsymbol{I}) \tag{13.12}$$

$$p_\theta(\mathbf{x}_{0:T}) = p(\mathbf{x}_T) \prod_{t=1}^{T} p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) \tag{13.13}$$

$$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t-1}; \boldsymbol{\mu}_\theta(\mathbf{x}_t, t), \boldsymbol{\Sigma}_\theta(\mathbf{x}_t, t)) \tag{13.14}$$

This is similar to the generative model in hierarchical VAEs. To perform training, we use the ELBO term and lower bound the objective as follows:

$$\begin{aligned}
-\log p_\theta(\mathbf{x}_0) &\leq -\log p_\theta(\mathbf{x}_0) + D_{KL}(q(\mathbf{x}_{1:T}|\mathbf{x}_0) \| p_\theta(\mathbf{x}_{1:T}|\mathbf{x}_0)) \\
&= -\log p_\theta(\mathbf{x}_0) + \mathbb{E}_{\mathbf{x}_{1:T} \sim q(\mathbf{x}_{1:T}|\mathbf{x}_0)}[\log \frac{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}{p_\theta(\mathbf{x}_{0:T})/p_\theta(\mathbf{x}_0)}] \\
&= -\log p_\theta(\mathbf{x}_0) + \mathbb{E}_q[\log \frac{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}{p_\theta(\mathbf{x}_{0:T})} + \log p_\theta(\mathbf{x}_0)] \\
&= \mathbb{E}_q[\log \frac{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}{p_\theta(\mathbf{x}_{0:T})}] \\
L_{VLB} &= \mathbb{E}_{q(\mathbf{x}_{0:T})}[\log \frac{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}{p_\theta(\mathbf{x}_{0:T})}] \geq -\mathbb{E}_{q(\mathbf{x}_0)} \log p_\theta(\mathbf{x}_0)
\end{aligned}$$

where if we plug into the terms that we found in the forward diffusion section, we get the following variational lower bound loss function:

$$\begin{aligned}
L_{VLB} &= L_T + L_{T-1} + \cdots + L_0 \\
\text{where } L_T &= D_{KL}(q(\mathbf{x}_T|\mathbf{x}_0) \parallel p_\theta(\mathbf{x}_T)) \\
L_t &= D_{KL}(q(\mathbf{x}_t|\mathbf{x}_{t+1}, \mathbf{x}_0) \parallel p_\theta(\mathbf{x}_t|\mathbf{x}_{t+1})) \text{ for } 1 \leq t \leq T-1 \\
L_0 &= -\log p_\theta(\mathbf{x}_0|\mathbf{x}_1)
\end{aligned}$$

Note that every KL term in the variational lower bound (VLB) compares two Gaussian distributions, hence they can be computed in closed form. In the DDPM [6] paper, simply setting the coefficient for each loss term yields the following loss function:

$$L_{simple} = \mathbb{E}_{\boldsymbol{x}_0 \sim q(\boldsymbol{x}_0), \epsilon \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}), t \sim U(1,T)} \|\epsilon - \epsilon_\theta(\sqrt{\bar{\alpha}_t}\boldsymbol{x}_0 + \sqrt{1-\bar{\alpha}_t}\epsilon, t)\|^2 \tag{13.15}$$

Note that our model takes two inputs, the noisy image at timestep t, $\boldsymbol{x}_t$ and t itself. And it learns to produce the corresponding noise vector.

13.1.3 Connection to VAEs

Diffusion models can be considered as a special form of hierarchical VAEs. However, in diffusion models:

- The inference model is fixed, easier to optimize
- The latent variables have the same dimension as the data
- The ELBO is decomposed to each time step: fast to train
- The model is trained with some reweighting of the ELBO.

13.1.4 Continuous-time diffusion models

So far, we have worked with the discrete-time interpretation of diffusion models. What if we consider the limit of many small steps and model it as a stochastic differential equation?

$$\boldsymbol{x}_t = \sqrt{1-\beta_t}\boldsymbol{x}_{t-1} + \sqrt{\beta_t}\mathcal{N}(\mathbf{0}, \boldsymbol{I}) \tag{13.16}$$

$$\boldsymbol{x}_t = \sqrt{1-\beta(t)\Delta t}\boldsymbol{x}_{t-1} + \sqrt{\beta(t)\Delta t}\mathcal{N}(\mathbf{0}, \boldsymbol{I}) \tag{13.17}$$

$$\boldsymbol{x}_t \approx \boldsymbol{x}_{t-1} - \frac{\beta(t)\Delta t}{2}\boldsymbol{x}_{t-1} + \sqrt{\beta(t)\Delta t}\mathcal{N}(\mathbf{0}, \boldsymbol{I}) \tag{13.18}$$

$$d\boldsymbol{x}_t = -\frac{1}{2}\beta(t)dt\boldsymbol{x}_t + \sqrt{\beta(t)}d\boldsymbol{w}_t \tag{13.19}$$

which is the formulation of the forward diffusion SDE. It's reversal can be found as

$$d\boldsymbol{x}_t = [-\frac{1}{2}\beta(t)\boldsymbol{x}_t - \beta(t)\nabla_{\boldsymbol{x}_t}\log q_t(\boldsymbol{x}_t)]dt + \sqrt{\beta(t)}d\bar{\boldsymbol{w}}_t \tag{13.20}$$

which was formulated in [7] by Anderson, 1982. $\sqrt{\beta(t)}d\bar{\boldsymbol{w}}_t$ term is the diffusion term and $\nabla_{\boldsymbol{x}_t}\log q_t(\boldsymbol{x}_t)$ is the score function. Now the idea is to learn a model to diffuse individual data points and perform denoising score matching, where the neural network model learns to predict the score function of the inference distribution.

References

[1] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[2] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

[3] Aaron Van den Oord, Nal Kalchbrenner, Lasse Espeholt, Oriol Vinyals, Alex Graves, et al. Conditional image generation with pixelcnn decoders. *Advances in neural information processing systems*, 29, 2016.

[4] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*, 2018.

[5] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in Neural Information Processing Systems*, 35:36479–36494, 2022.

[6] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

[7] Brian DO Anderson. Reverse-time diffusion equation models. *Stochastic Processes and their Applications*, 12(3):313–326, 1982.