Boulder

Generative Networks; Diffusion Models

Maziar Raissi

Assistant Professor

Department of Applied Mathematics

University of Colorado Boulder

maziar.raissi@colorado.edu

Denoising Diffusion Probabilistic Models

$\mathbf{x}_0 \sim q(\mathbf{x}_0) \rightarrow$ data

$p_\theta(\mathbf{x}_0) := \int p_\theta(\mathbf{x}_{0:T})\, d\mathbf{x}_{1:T}$

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

$p(\mathbf{x}_T) = \mathcal{N}(\mathbf{x}_T; \mathbf{0}, \mathbf{I})$

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

$\boldsymbol{\Sigma}_\theta(\mathbf{x}_t, t) = \sigma_t^2 \mathbf{I} \rightarrow$ untrained time dependent constants

$$\boldsymbol{\mu}_\theta(\mathbf{x}_t, t) = \frac{1}{\sqrt{\alpha_t}} \left(\mathbf{x}_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}} \boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t) \right)$$

$\boldsymbol{\epsilon}_\theta \rightarrow$ function approximator

Algorithm 2 Sampling

1: $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
2: **for** $t = T, \ldots, 1$ **do**
3: $\quad \mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ if $t > 1$, else $\mathbf{z} = \mathbf{0}$
4: $\quad \mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}} \boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t) \right) + \sigma_t \mathbf{z}$
5: **end for**
6: **return** $\mathbf{x}_0$

$$q(\mathbf{x}_{1:T}|\mathbf{x}_0) := \prod_{t=1}^{T} q(\mathbf{x}_t|\mathbf{x}_{t-1}) \rightarrow \text{approximate posterior (forward process or diffusion process)}$$

$q(\mathbf{x}_t|\mathbf{x}_{t-1}) := \mathcal{N}(\mathbf{x}_t; \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t \mathbf{I})$

$\beta_1, \ldots, \beta_T \rightarrow$ variance schedule (hyperparameters)

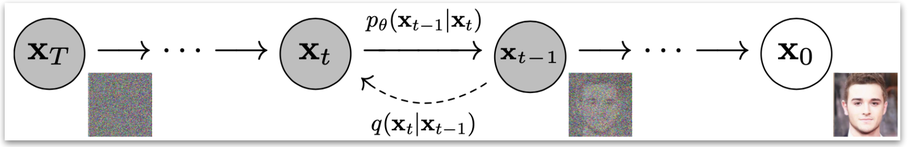
variational bound on negative log likelihood

$$\mathbb{E}[-\log p_\theta(\mathbf{x}_0)] \leq \mathbb{E}_q \left[-\log \frac{p_\theta(\mathbf{x}_{0:T})}{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} \right] = \mathbb{E}_q \left[-\log p(\mathbf{x}_T) - \sum_{t \geq 1} \log \frac{p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)}{q(\mathbf{x}_t|\mathbf{x}_{t-1})} \right] =: L$$

$q(\mathbf{x}_t|\mathbf{x}_0) = \mathcal{N}(\mathbf{x}_t; \sqrt{\bar{\alpha}_t}\mathbf{x}_0, (1-\bar{\alpha}_t)\mathbf{I}) \rightarrow$ sampling at an arbitrary timestep

$\alpha_t := 1 - \beta_t$ and $\bar{\alpha}_t := \prod_{s=1}^{t} \alpha_s$

$\sigma_t^2 = \tilde{\beta}_t = \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t} \beta_t$



Rewriting L

$$\mathbb{E}_q \left[\underbrace{D_{\mathrm{KL}}(q(\mathbf{x}_T|\mathbf{x}_0) \parallel p(\mathbf{x}_T))}_{L_T} + \sum_{t>1} \underbrace{D_{\mathrm{KL}}(q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) \parallel p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t))}_{L_{t-1}} \underbrace{-\log p_\theta(\mathbf{x}_0|\mathbf{x}_1)}_{L_0} \right]$$

L_T is a constant during training and can be ignored

$q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) = \mathcal{N}(\mathbf{x}_{t-1}; \tilde{\boldsymbol{\mu}}_t(\mathbf{x}_t, \mathbf{x}_0), \tilde{\beta}_t \mathbf{I}) \rightarrow$ forward process posteriors

$$\tilde{\boldsymbol{\mu}}_t(\mathbf{x}_t, \mathbf{x}_0) := \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t} \mathbf{x}_0 + \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t} \mathbf{x}_t$$

$$L_{t-1} = \mathbb{E}_q \left[\frac{1}{2\sigma_t^2} \|\tilde{\boldsymbol{\mu}}_t(\mathbf{x}_t, \mathbf{x}_0) - \boldsymbol{\mu}_\theta(\mathbf{x}_t, t)\|^2 \right] + C$$

$$L_{\text{simple}}(\theta) := \mathbb{E}_{t, \mathbf{x}_0, \boldsymbol{\epsilon}} \left[\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, t)\|^2 \right]$$

$L_0 \rightarrow$ see the paper

Algorithm 1 Training

1: **repeat**
2: $\quad \mathbf{x}_0 \sim q(\mathbf{x}_0)$
3: $\quad t \sim \text{Uniform}(\{1, \ldots, T\})$
4: $\quad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
5: $\quad$ Take gradient descent step on
$\qquad \nabla_\theta \|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, t)\|^2$
6: **until** converged

Ho, Jonathan, Ajay Jain, and Pieter Abbeel. "Denoising diffusion probabilistic models." *Advances in Neural Information Processing Systems* 33 (2020): 6840-6851.

Diffusion Models Beat GANs on Image Synthesis

– improving model architecture
– trading off diversity for fidelity

Background

$x_T \rightarrow$ noise

$x_{T-1}, x_{T-2}, \ldots \rightarrow$ gradually less noisy samples

$x_0 \rightarrow$ final sample

$x_{t-1} \rightarrow$ slightly more denoised version of x_t

$\varepsilon_\theta(x_t, t) \rightarrow$ function predicting the noise component of a noisy sample x_t

$\|\varepsilon_\theta(x_t, t) - \varepsilon\|^2 \rightarrow$ training objective

$x_0, t,$ and ε give rise to a noised sample x_t

$\mu_\theta(x_t, t) \rightarrow$ mean of the denoised distribution $p_\theta(x_{t-1}|x_t)$

$\mu_\theta(x_t, t) \rightarrow$ can be calculated as a function of $\varepsilon_\theta(x_t, t)$

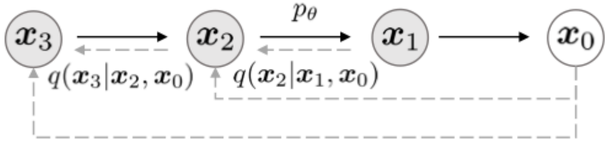
$\Sigma_\theta(x_t, t) \rightarrow$ variance of $p_\theta(x_{t-1}|x_t)$

fixing $\Sigma_\theta(x_t, t)$ is suboptimal

$\Sigma_\theta(x_t, t) = \exp(v_\theta(x_t, t) \log \beta_t + (1 - v_\theta(x_t, t)) \log \tilde{\beta}_t)$

$L_{\text{simple}} + \lambda L_{\text{variational lower bound}}$

non-Markovian noising process $\rightarrow$ sample with fewer steps



Architecture

UNet architecture

Channels	Depth	Heads	Attention resolutions	BigGAN up/downsample	Rescale resblock	FID 700K	FID 1200K
160	2	1	16	✗	✗	15.33	13.21
128	4					-0.21	-0.48
		4				-0.54	-0.82
			32,16,8			-0.72	-0.66
				✓		-1.20	-1.21
					✓	0.16	0.25
160	2	4	32,16,8	✓	✗	**-3.14**	**-3.00**

adaptive group normalization (AdaGN)

AdaGN$(h, y) = y_s$ GroupNorm$(h) + y_b$, where h is the intermediate activations of the residual block following the first convolution, and $y = [y_s, y_b]$ is obtained from a linear projection of the timestep and class embedding.

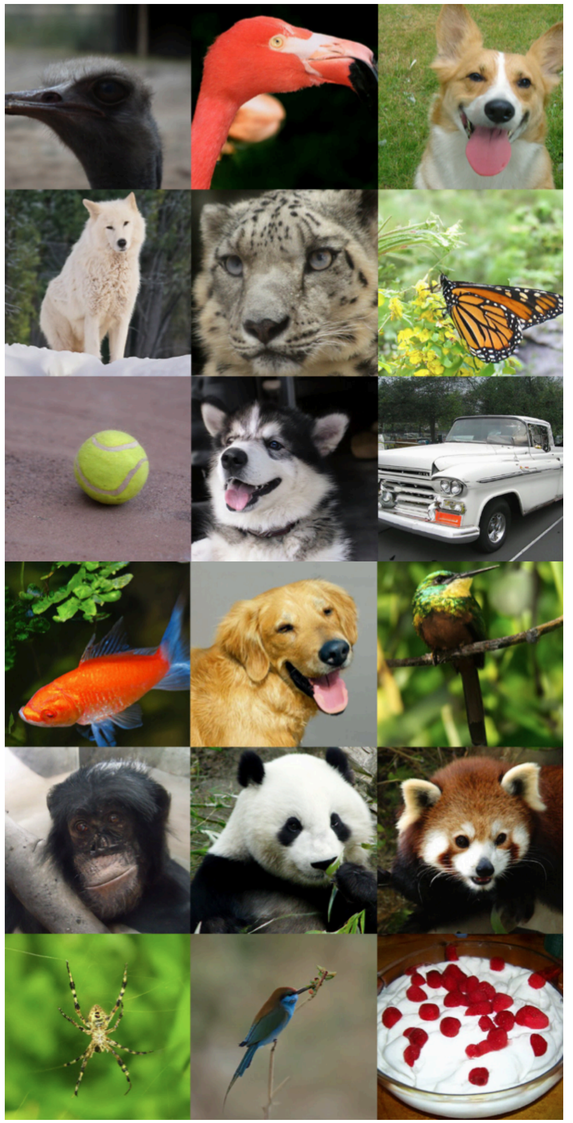
Classifier Guidance

Algorithm 1 Classifier guided diffusion sampling, given a diffusion model $(\mu_\theta(x_t), \Sigma_\theta(x_t))$, classifier $p_\phi(y|x_t)$, and gradient scale s.

Input: class label y, gradient scale s
$x_T \leftarrow$ sample from $\mathcal{N}(0, \mathbf{I})$
for all t from T to 1 **do**
  $\mu, \Sigma \leftarrow \mu_\theta(x_t), \Sigma_\theta(x_t)$
  $x_{t-1} \leftarrow$ sample from $\mathcal{N}(\mu + s\Sigma \nabla_{x_t} \log p_\phi(y|x_t), \Sigma)$
end for
return x_0

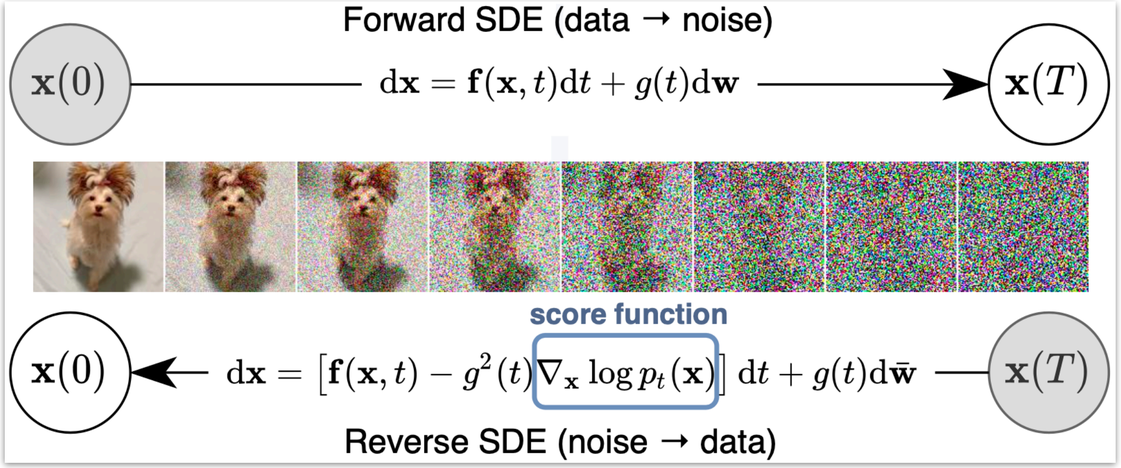
Algorithm 2 Classifier guided DDIM sampling, given a diffusion model $\epsilon_\theta(x_t)$, classifier $p_\phi(y|x_t)$, and gradient scale s.

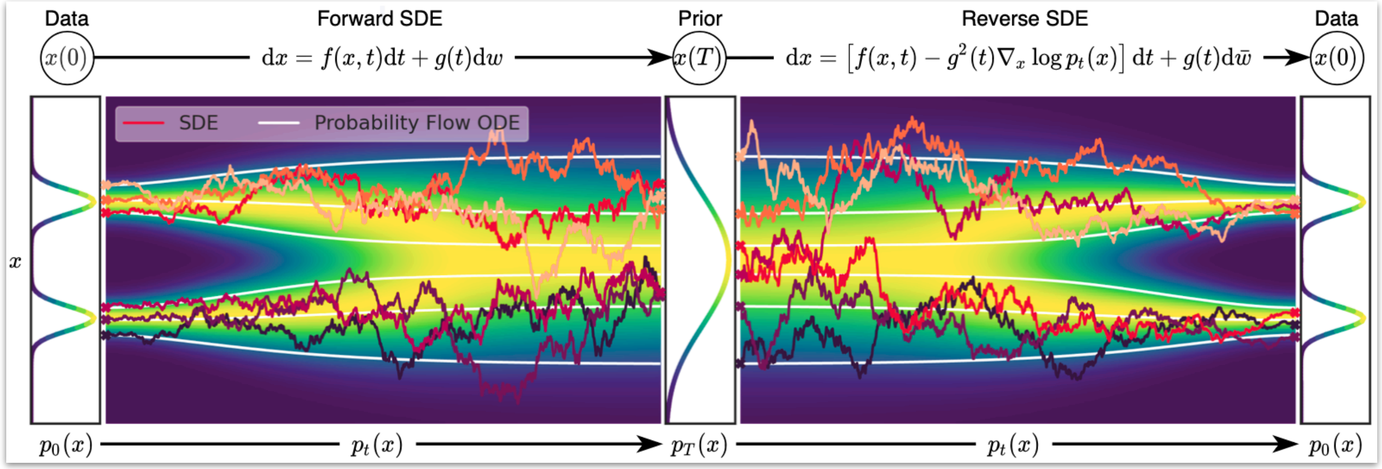
Input: class label y, gradient scale s
$x_T \leftarrow$ sample from $\mathcal{N}(0, \mathbf{I})$
for all t from T to 1 **do**
  $\hat{\epsilon} \leftarrow \epsilon_\theta(x_t) - \sqrt{1 - \bar{\alpha}_t} \nabla_{x_t} \log p_\phi(y|x_t)$
  $x_{t-1} \leftarrow \sqrt{\bar{\alpha}_{t-1}} \left(\frac{x_t - \sqrt{1-\bar{\alpha}_t}\hat{\epsilon}}{\sqrt{\bar{\alpha}_t}} \right) + \sqrt{1 - \bar{\alpha}_{t-1}}\hat{\epsilon}$
end for
return x_0



Dhariwal, Prafulla, and Alexander Nichol. "Diffusion models beat gans on image synthesis." *Advances in Neural Information Processing Systems* 34 (2021): 8780-8794.

Score-Based Generative Modeling through Stochastic Differential Equations





$\{x(t) : t \in [0, T]\}$ → diffusion process

$x(0) \sim p_0$ → data distribution

$x(T) \sim p_T$ → tractable prior distribution

$dx = f(x,t)dt + g(t)dw$

$f(\cdot, t) : \mathbb{R}^d \to \mathbb{R}^d$

↳ drift

$g(\cdot) : \mathbb{R} \to \mathbb{R}$

↳ diffusion

$p_t(x)$ → probability density of $x(t)$

$p_{st}(x(t)|x(s))$ → transition kernel from $x(s)$ to $x(t)$

Inpainting

Examples

$dx = \sqrt{\frac{d[\sigma^2(t)]}{dt}}dw$ → Denoising Score Matching with Langevin Dynamics (SMLD)

$dx = -\frac{1}{2}\beta(t)xdt + \sqrt{\beta(t)}dw$ → Denoising Diffusion Probabilistic Models (DDPM)

$dx = -\frac{1}{2}\beta(t)xdt + \sqrt{\beta(t)(1 - e^{-2\int_0^t \beta(s)ds})}dw$

Generating Samples by Reversing the SDE

$dx = [f(x,t) - g(t)^2\nabla_x \log p_t(x)]dt + g(t)d\bar{w}$

$\bar{w}$ → a standard Wiener process when time flows backwards from T to 0

dt → an infinitesimal negative timestep

Estimating Scores for the SDE

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} \mathbb{E}_t\left\{\lambda(t)\mathbb{E}_{\mathbf{x}(0)}\mathbb{E}_{\mathbf{x}(t)|\mathbf{x}(0)}\left[\left\|\mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x}(t), t) - \nabla_{\mathbf{x}(t)} \log p_{0t}(\mathbf{x}(t) \mid \mathbf{x}(0))\right\|_2^2\right]\right\}$$

$\mathbf{s}_{\boldsymbol{\theta}^*}(\mathbf{x}, t)$ equals $\nabla_{\mathbf{x}} \log p_t(\mathbf{x})$

Controllable Generation: sample from $p_t(\mathbf{x}(t) \mid \mathbf{y})$

$$\mathrm{d}\mathbf{x} = \{\mathbf{f}(\mathbf{x}, t) - g(t)^2[\nabla_{\mathbf{x}} \log p_t(\mathbf{x}) + \nabla_{\mathbf{x}} \log p_t(\mathbf{y} \mid \mathbf{x})]\}\mathrm{d}t + g(t)\mathrm{d}\bar{\mathbf{w}}$$

Song, Yang, et al. "Score-based generative modeling through stochastic differential equations." *arXiv preprint arXiv:2011.13456* (2020).

Boulder

Questions?