Stochastic Differential Equations for Generative Modeling

Anton Bolychev[1,2], Georgiy Malaniya[2], Oleg Shepelin[2], Anastasiya Archangelskaya[2], Nikolay Kalmykov[2], Vadim Shirokinsky[2]

The presentation was developed by 1st year MSU Phd Student
Anton Bolychev[1,2]

[1]Moscow State University Faculty of Mechanics and Mathematics
[2]Skolkovo Institute Of Science and Technology

May 2023

Contribution

This work was developed as the final project for the **Evgeniy Burnaev**'s course **Machine Learning** that was held in Skolkovo Intsitute of Science and Technology. Contribution of all the participants is as follows:

1. **Anton Bolychev** participated as the **Principal Developer** in the present work. The core research, the code, the repo structure was prepared and finalized by him. Moreover he is the only author of the presentation.
2. **Georgiy Malaniya** particapated as the **Team Lead**. He maintained all the management process of all the participants
3. **Oleg Shepelin** and **Nikolay Kalmykov** participated as the **Core Developers** and prepared the draft of main loop for Langevin Dynamics
4. **Vadim Shirokinsky** was responsible for the calculating FIDs routine and took part in preparing the presentation for Skoltech.
5. **Anastasiya Archangelskaya** finalized results and prepared the report with presentation for Skoltech

Table of Contents

GAN

$$L_D = -\mathbb{E}_{x\sim p_{data}}[\log D(x)] - \mathbb{E}_{z\sim p_z}[\log(1 - D(G(z))]$$
$$L_G = -\mathbb{E}_{z\sim p_z}[\log D(G(z))]$$

GAN as Energy Based Model

Assume that Discriminator is suboptimal, i.e. $D = D^*$

$$D(x) = \text{logit}(d(x)) = \frac{1}{1+\exp(-d(x))} \approx \frac{p_d(x)}{p_d(x)+p_g(x)} = \frac{1}{1+p_g(x)/p_d(x)}$$

Thus,

$$p_d^*(x) = p_g(x)e^{d(x)}/K = \exp\left(-(-\log p_g(x) - d(x))\right)/K$$

Energy Function. Boltzmann distribution

$$p(z) = \exp(-E(z))/K$$

Thus, assuming that $x = G(z)$ one can obtain the following equation for Energy for GAN

$$E(z) = -\log p_0(z) - d(G(z))$$

Langevin Dynamics

$$z_{i+1} = z_i - \epsilon/2 \nabla_z E(z) + \sqrt{\epsilon} n, n \sim N(0, I)$$

Input: N$\in \mathbf{N}_+$, $\epsilon > 0$
Output: Latent code $z_N \sim p_t(z)$
Sample $z_0 \sim p_0(z)$
for $i = 1$ **to** N **do**
  $n_i \sim N(0, 1)$
  $z_{i+1} = z_i - \epsilon/2 \nabla_z E(z_i) + \sqrt{\epsilon} n_i$
end for

Frechet Inception Distance

FID can be calculated according to the following formula

$$d_F(\mu, \nu) := \left(\inf_{\gamma \in \Gamma(\mu,\nu)} \int_{\mathbb{R}^n \times \mathbb{R}^n} \|x - y\|^2 \, \mathrm{d}\gamma(x, y) \right)^{1/2}$$

Results

If one apply Langevin Dynamics for pretrained DCGAN on CIFAR10 one can observe that FID metrics decreases with increasing number of Langevin steps.

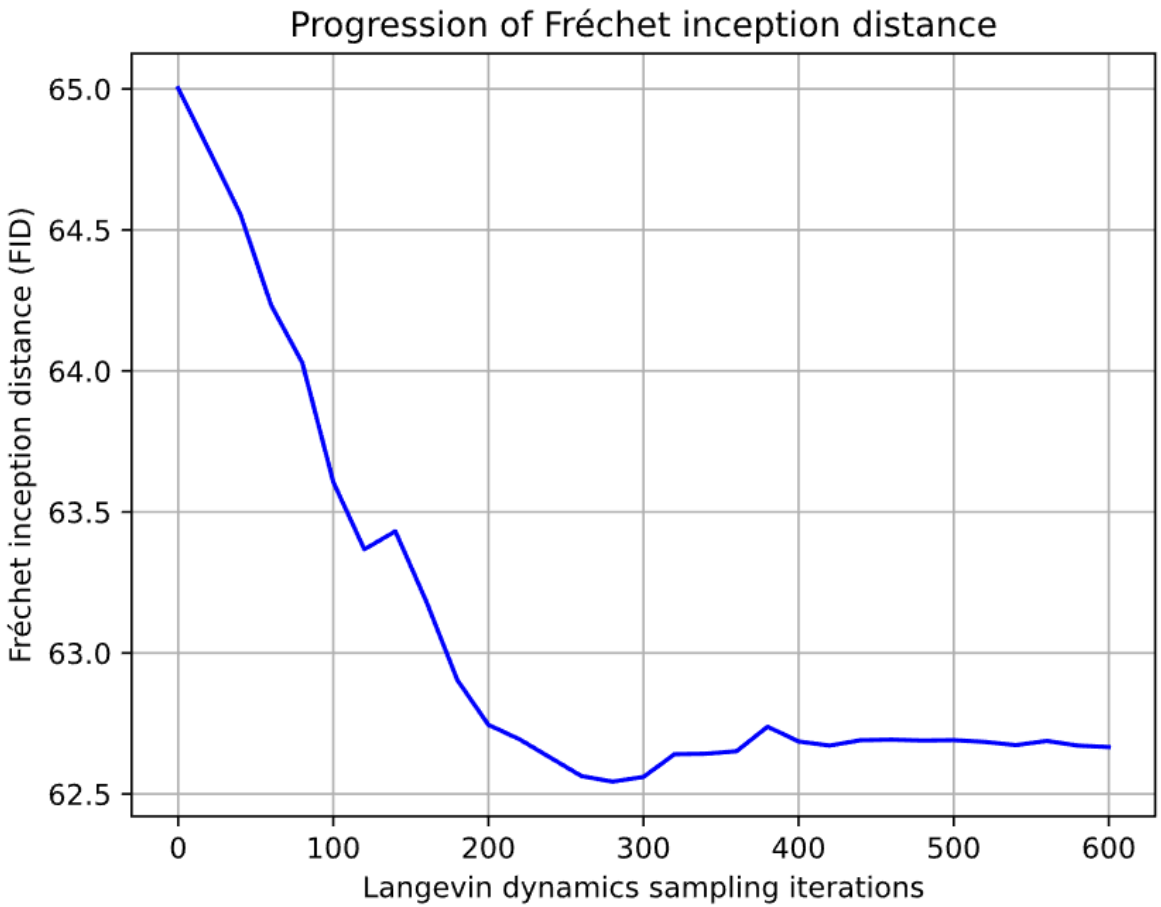
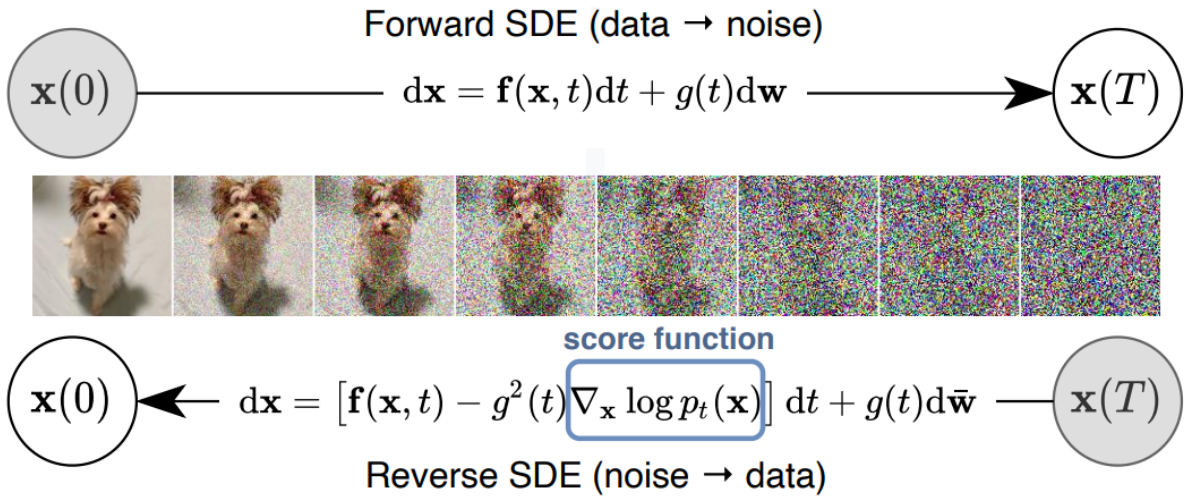


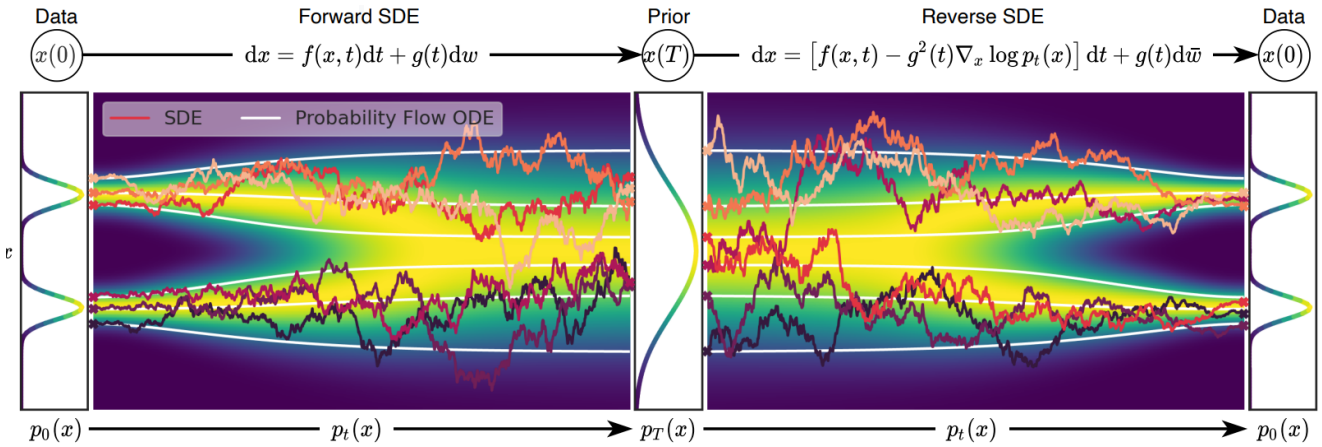
Table of Contents

Score-based Generative Modelling through SDE

The core idea of the paper can be described via the following picture



Score-based Generative Modelling through SDE



The main problem is to fit the neural network $\mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x}(t), t)$ such that

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

Loss function

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

where

$$\lambda \propto 1 / \mathbb{E} \left[\left\| \nabla_{\mathbf{x}(t)} \log p_{0t}(\mathbf{x}(t) \mid \mathbf{x}(0)) \right\|_2^2 \right]$$

2 approaches

In $\mathrm{d}\mathbf{x} = \mathbf{f}(\mathbf{x}, t)\mathrm{d}t + g(t)\mathrm{d}\mathbf{w}$ functions $\mathbf{f}$ and g can be arbitrary, so we will consider 2 approaches

- Variance Exploding Approach
- Variance Preserving Approach

Variance Exploding

- SDE

$$\mathrm{d}\mathbf{x} = \sqrt{\frac{\mathrm{d}\left[\sigma^2(t)\right]}{\mathrm{d}t}}\,\mathrm{d}\mathbf{w}$$

- Forward Sampling

$$\mathbf{x}_i = \mathbf{x}_{i-1} + \sqrt{\sigma_i^2 - \sigma_{i-1}^2}\mathbf{z}_{i-1}$$

Variance Preserving

- SDE

$$\mathrm{d}\mathbf{x} = -\frac{1}{2}\beta(t)\mathbf{x}\mathrm{d}t + \sqrt{\beta(t)}\mathrm{d}\mathbf{w}$$

- Forward Sampling

$$\mathbf{x}_i = \sqrt{1-\beta_i}\mathbf{x}_{i-1} + \sqrt{\beta_i}\mathbf{z}_{i-1}$$

Ancestral Sampling for Variance Preserving

$$\mathbf{x}_{i-1} = \frac{1}{\sqrt{1-\beta_i}}\left(\mathbf{x}_i + \beta_i \mathbf{s}_{\boldsymbol{\theta}^*}\left(\mathbf{x}_i, i\right)\right) + \sqrt{\beta_i}\mathbf{z}_i, \qquad i = N, N-1, \cdots, 1$$

Reverse Diffusion

Given a forward SDE

$$\mathrm{d}\mathbf{x} = \mathbf{f}(\mathbf{x}, t)\mathrm{d}t + \mathbf{G}(t)\mathrm{d}\mathbf{w}$$

and suppose the following iteration rule is a discretization of it:

$$\mathbf{x}_{i+1} = \mathbf{x}_i + \mathbf{f}_i\left(\mathbf{x}_i\right) + \mathbf{G}_i\mathbf{z}_i, \quad i = 0, 1, \cdots, N-1$$

Thus, one can propose to discretize the reverse-time SDE

$$\mathrm{d}\mathbf{x} = \left[\mathbf{f}(\mathbf{x}, t) - \mathbf{G}(t)\mathbf{G}(t)^\top \nabla_{\mathbf{x}} \log p_t(\mathbf{x})\right]\mathrm{d}t + \mathbf{G}(t)\mathrm{d}\overline{\mathbf{w}}$$

which gives the following iteration rule for $i \in \{0, 1, \cdots, N-1\}$:

$$\mathbf{x}_i = \mathbf{x}_{i+1} - \mathbf{f}_{i+1}\left(\mathbf{x}_{i+1}\right) + \mathbf{G}_{i+1}\mathbf{G}_{i+1}^\top \mathbf{s}_{\boldsymbol{\theta}^*}\left(\mathbf{x}_{i+1}, i+1\right) + \mathbf{G}_{i+1}\mathbf{z}_{i+1},$$

where our trained score-based model $\mathbf{s}_{\boldsymbol{\theta}} * (\mathbf{x}_i, i)$.

Predictor-Corrector Sampling

Algorithm 2 PC sampling (VE SDE)

$\mathbf{x}_N \sim \mathcal{N}(\mathbf{0}, \sigma_{\max}^2 \mathbf{I})$
for $i = N-1$ **to** 0 **do**
  $\mathbf{x}'_i \leftarrow \mathbf{x}_{i+1} + (\sigma_{i+1}^2 - \sigma_i^2)\mathbf{s}_{\boldsymbol{\theta}*}(\mathbf{x}_{i+1}, \sigma_{i+1})$
  $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
  $\mathbf{x}_i \leftarrow \mathbf{x}'_i + \sqrt{\sigma_{i+1}^2 - \sigma_i^2}\mathbf{z}$
  for $j = 1$ **to** M **do**
    $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
    $\mathbf{x}_i \leftarrow \mathbf{x}_i + \epsilon_i \mathbf{s}_{\boldsymbol{\theta}*}(\mathbf{x}_i, \sigma_i) + \sqrt{2\epsilon_i}\mathbf{z}$
return $\mathbf{x}_0$

Algorithm 3 PC sampling (VP SDE)

$\mathbf{x}_N \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
for $i = N-1$ **to** 0 **do**
  $\mathbf{x}'_i \leftarrow (2 - \sqrt{1-\beta_{i+1}})\mathbf{x}_{i+1} + \beta_{i+1}\mathbf{s}_{\boldsymbol{\theta}*}(\mathbf{x}_{i+1}, i+1)$
  $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
  $\mathbf{x}_i \leftarrow \mathbf{x}'_i + \sqrt{\beta_{i+1}}\mathbf{z}$ — Predictor
  for $j = 1$ **to** M **do** — Corrector
    $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
    $\mathbf{x}_i \leftarrow \mathbf{x}_i + \epsilon_i \mathbf{s}_{\boldsymbol{\theta}*}(\mathbf{x}_i, i) + \sqrt{2\epsilon_i}\mathbf{z}$
return $\mathbf{x}_0$

Results. FID on CIFAR10. VE

FID	P1000	P2000	**PC1000**
ancestral sampling	31.76	31.7	**30.57**
reverse diffusion	31.98	31.43	**30.96**

As we can see Predictor-Corrector sampling gives the best performance in terms of FID metrics

Results. FID on CIFAR10. VP

FID	P1000	P2000	**PC1000**
ancestral sampling	30.55	30.53	**29.74**
reverse diffusion	31.01	30.32	**30.01**

As we can see Predictor-Corrector sampling gives the best performance in terms of FID metrics

Table of Contents

Conclusion

This work is devoted to examining the power of SDE theory in Image Generation. The work consists of 2 parts

1. The 1 part examines the paper [1] which main idea is based on the fact that GAN can be interpreted as Energy Based Model. Thus, one can easily apply Langevin Dynamics for it which can improve FID metrics for already pretrained GAN model. The corresponding plot of FID behavior with dependence on the Langevin steps is presented on this slide.
2. The 2 part reproduces experiments from the paper [2] that implements the following idea. What if one can controllably transform the initial data disctibution into noise and reverse the process? Well, that can be done via reversing the SDE formula. Namely, if one consider the SDE process that is defined by SDE $\mathrm{d}\mathbf{x} = \mathbf{f}(\mathbf{x}, t)\mathrm{d}t + g(t)\mathrm{d}\mathbf{w}$ then one can reverse it via $\mathrm{d}\mathbf{x} = \left[\mathbf{f}(\mathbf{x}, t) - g(t)^2 \nabla_{\mathbf{x}} \log p_t(\mathbf{x})\right] \mathrm{d}t + g(t)\mathrm{d}\overline{\mathbf{w}}$. In the paper several sampling technics are examined, and the results are finalized on this and this slides.

Table of Contents

References

1. Tong Che, Ruixiang Zhang, Jascha Sohl-Dickstein, Hugo Larochelle, Liam Paull, Yuan Cao, Yoshua Bengio. *Your GAN is Secretly an Energy-based Model and You Should use Discriminator Driven Latent Sampling* https://arxiv.org/abs/2003.06060
2. Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, Ben Poole. *Score-Based Generative Modeling through Stochastic Differential Equations* https://arxiv.org/abs/2011.13456
3. Github for DCGAN on CIFAR10 https://github.com/csinva/gan-vae-pretrained-pytorch