Stochastic Differential Equations (SDE) for Machine Learning: Applications on Least-Squares Stochastic Gradient Descent and Score-Based Generative Models

NYU Shanghai Mathematics Senior Thesis Oral Presentation

Ruiqi Ge(Selena) rg4012@nyu.edu
Supervisor: Prof. Mathieu Lauriere

Contents

1. Introduction to Stochastic Differential Equations(SDE)

2. The connection between Stochastic Gradient Descent(SGD) and SDE

3. Score-based Generative Models with SDE

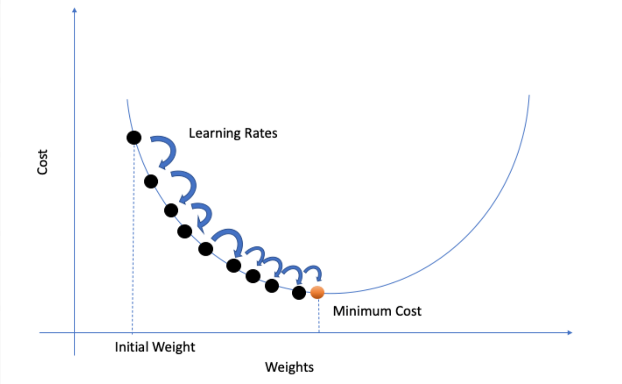


Illustration of Stochastic Gradient Descent

Image source: Ghosh et al. (2020), An Empirical Analysis of Generative Adversarial Network Training Times with Varying Batch Sizes. DOI: 10.1109/UEMCON51285.2020.9298092.

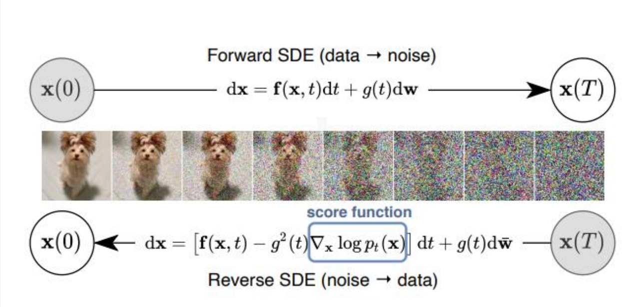
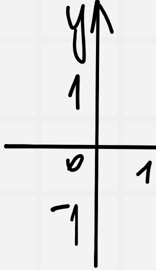


Illustration of Score-based Generative Models with SDE

Image source: Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. arXiv preprint arXiv:2011.13456v2, 2020

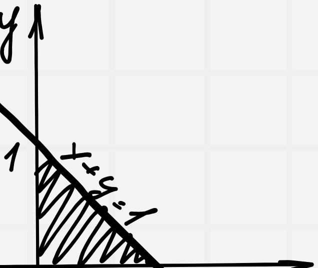


1. Introduction to Stochastic Differential Equations(SDE)

- Clarification of SDE related concepts
- Numerical Methods of Solving SDE

  Ito’s formula, Euler-Maruyama method
- Special SDE cases

  Black-Scholes, Ornstein-Uhlenbeck Process



fdy

What is SDE?

Stochastic differential equations (SDEs) are a type of differential equations used to model systems that exhibit random behavior. An SDE typically takes the form:

$$\boxed{dX = a(t, X)dt + b(t, X)dW_t}$$

- $a(t, X)dt$: drift term because it captures the average or expected rate of change of the process X if no randomness was involved.
- $b(t, X)dW_t$: diffusion term because it scales the magnitude of the randomness by the increment of W.

Numerical Solution of SDE:

- Ito's formula
  (Chain rule for SDE)
  Typical model: Black-Scholes
- Euler-Maruyama method
  (Approximate solution of SDE)
  Typical model: OU process

5 = 2

fdy

Ornstein-Uhlenbeck Process

OU process: Definition

The Ornstein-Uhlenbeck process is a stochastic process that satisfies the following SDE:

$$dX_t = \kappa(\theta - X_t)dt + \sigma dW_t$$

where W_t is a standard Brownian motion on $t \in [0, \infty)$. The constant parameters are:

- $\kappa > 0$ is the rate of mean reversion;
- θ is the long-term mean of the process;
- $\sigma > 0$ is the volatility or average magnitude, per square-root time, of the random fluctuations that are modeled as Brownian motions.

OU process: $\boxed{\text{Mean-reverting property}}$

If we ignore the random fluctuations in the process due to dW_t, then we see that X_t has an overall drift towards a mean value θ. The process X_t reverts to this mean exponentially, at rate κ, with a magnitude in direct proportion to the distance between the current value of X_t and θ.
For any fixed s and t, the random variable X_t, conditional upon X_s, is normally distributed with:

$$mean = \theta + (X_s - \theta)e^{-\kappa(t-s)}$$

$$variance = \frac{\sigma^2}{2\kappa}(1 - e^{-2\kappa(t-s)})$$

Observe that the mean of X_t is exactly the value derived heuristically in the solution of the ODE. The Ornstein-Uhlenbeck process is a time-homogeneous Itô diffusion.

s = 2

2. The connection between Stochastic Gradient Descent (SGD) and SDE

- What & Why SGD
- Connect SGD with SDE: Stochastic modified equations(SME)
- Explicit form of SME: connect with OU process
- Simulations

fdy

The algorithm of SGD

Stochastic Gradient Descent

Solving EMR using the **standard gradient descent(GD)** on x gives the iteration scheme. First, define the gradient of f as for all $x = (x_1, \dots, x_d) \in \mathbb{R}^d$, for all $i = 1 \dots d$,

$$\nabla f(x) = \begin{pmatrix} \partial_{x_1} f(x_1, \dots, x_d) \\ \vdots \\ \partial_{x_d} f(x_1, \dots, x_d) \end{pmatrix} \in \mathbb{R}^d,$$

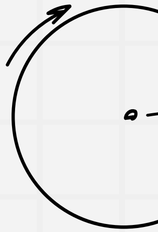
Then we have the recursion

$$x_{k+1} = x_k - \eta \nabla f(x_k) = x_k - \eta \nabla \mathbb{E}_\gamma \left[f_\gamma(x_k) \right]$$

for $k \geq 0$ and η is a small step-size known as the **learning rate**.
Simple form:

$$\boxed{x_{k+1} = x_k - \eta \nabla f_{\gamma_k}(x_k)}$$

where each γ_k is an i.i.d random variable with the same distribution as γ. We then have $\mathbb{E}[\nabla f_{\gamma_k}(x_k) | (x_k)] = \nabla \mathbb{E} f(x_k)$.

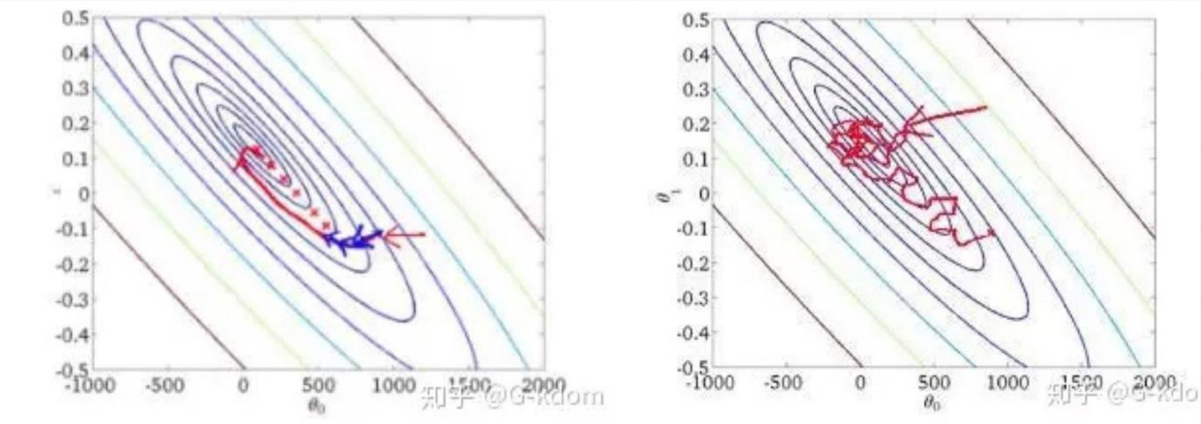


S = 2

fdy

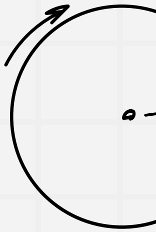
Advantages of SGD

The **stochastic gradient descent (SGD)** aims at minimizing a function through unbiased estimates of its gradient. It is an optimization algorithm used primarily for training <u>large-scale</u> machine learning models. It's a variant of gradient descent, where instead of computing the gradient of the cost function using the entire dataset (as in **batch gradient descent**), it computes the gradient <u>using a small batch of samples.</u>



Batch

Stochastic



5 = 2

fdy

Simulation 1: SGD

We use the equation of $\theta_{t+1} = \theta_t - \gamma x_t(\langle \theta_t, x_t \rangle - y_t)$ to write a Python code that simulates the SGD dynamics until time $t = 1000$, with step-size $\gamma = 0.01$, initialization $\theta_0 = \mathbf{0}$, the zero vector, $\theta^* = [0.1, -0.2, 1, 0.5, -0.5]$ and $\sigma = 0.1$. We display the test error curve upon time $\|\theta_t - \theta_*\|^2$ for several runs of the dynamics (meaning different data), and also display the two first coordinates of $(\theta_t)_t$ as well as the ones of θ^*.

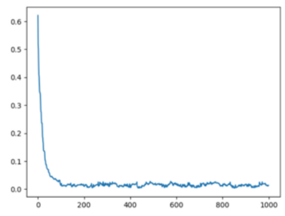


Figure 1: Simulation of test error curve

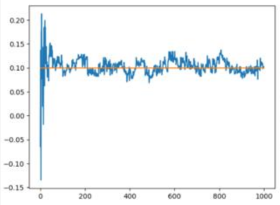


Figure 2: Simulation of θ_1

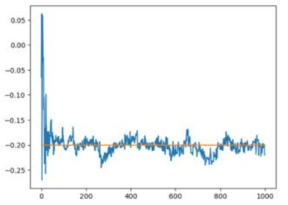


Figure 3: Simulation of θ_2

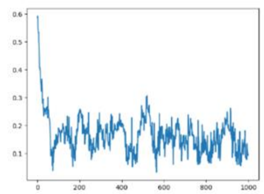


Figure 4: Test Error when $\sigma = 1$

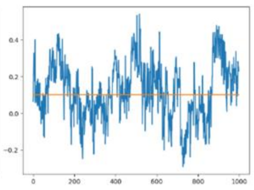


Figure 5: Simulation of θ_1 when $\sigma = 1$

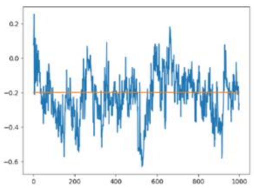


Figure 6: Simulation of θ_2 when $\sigma = 1$

Accurate but slow (step-size small)

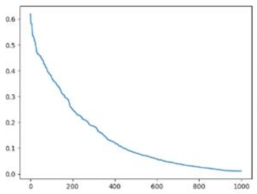


Figure 10: Test Error when step-size smaller

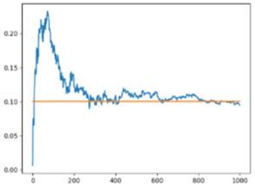


Figure 11: θ_1 when step-size smaller

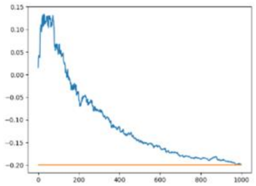


Figure 12: θ_2 when step-size smaller

Too quick (step-size big)

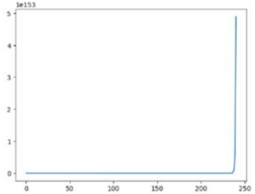


Figure 7: Test Error when step-size bigger

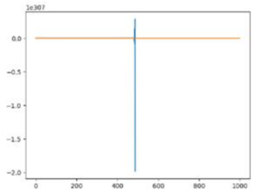


Figure 8: θ_1 when step-size bigger

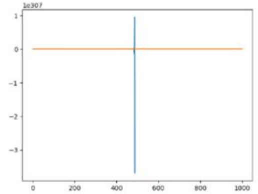


Figure 9: θ_2 when step-size bigger

x-axis: time(t/s)
y-axis: test error/ theta1 / theta2
Simulation for **variance & step-size** change

fdy

Simulation 1: SGD

Change the step size to make it depend on the iterations: $\gamma = 0.1/t$. The simulation balanced between accuracy and velocity.

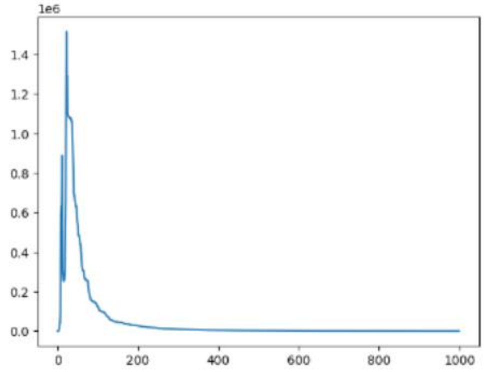


Figure 13: Test Error: Improved

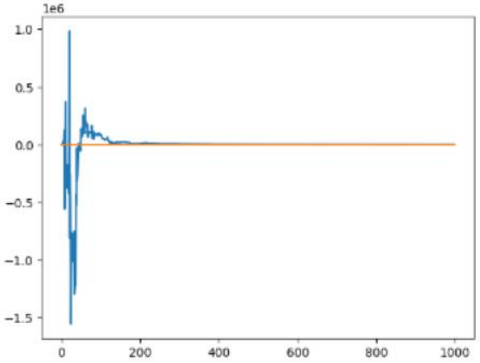


Figure 14: θ_1: Improved

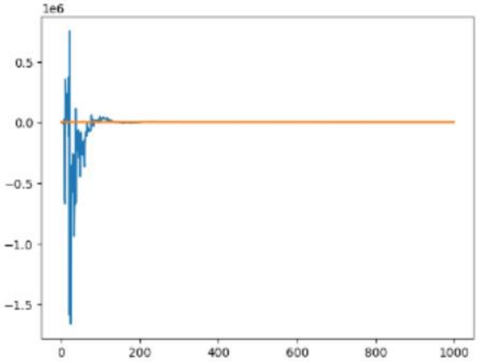


Figure 15: θ_2: Improved

5 = 2

fdy

What SDE model fits well with SGD: Stochastic Modified Equations(SME)

match parameters

General solution of SDE:

$$\boxed{d\theta_t = b(t, \theta_t)dt + \sigma(t, \theta_t)dB_t,}$$

If we apply the Euler-Maruyama discretization with step-size γ, approximating $X_{k\gamma}$ by $\hat{X}_k$, we obtain the following discrete iteration:

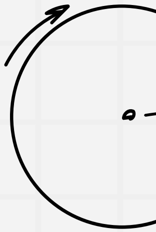
$$\hat{\theta_{t+1}} - \hat{\theta}_t = \gamma b(t, \hat{\theta}_t) + \sqrt{\gamma}\sigma(t, \hat{\theta}_t)Z_k$$

where $Z_k := B_{(k+1)\gamma} - B_{k\gamma}$ are d-dimensional i.i.d standard normal random variables. Stochastic Modified Equation:

$$\theta_{t+1} - \theta_t = -\gamma\nabla L(\theta_t) + \gamma(\nabla L(\theta_t) - \nabla l(\theta_t))$$

Then

$$\boxed{d\theta_t = -\nabla L(\theta_t)dt + \sqrt{\gamma\Sigma(\theta)}dB_t}$$



5 = 2

fdy

Explicit form of SME: connect with OU process

Simplify the covariance $\sigma(\theta) := \sqrt{\gamma\sigma^2} I_d$. Then the SDE becomes:

$$\boxed{d\theta_t = (-(\theta_t - \theta^*)Id)dt + (\sqrt{\gamma\sigma^2}Id)dB_t}$$

Match each parameter with the OU process:

$$\boxed{d\theta_t = \kappa(\theta - \theta_t)dt + \sigma dW_t}$$

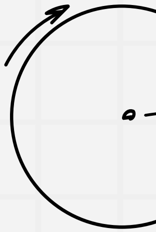
We get:

- $\kappa = 1$.
- $\theta = \theta^*$, the long-term mean of the process matches θ^*.
- $\sigma = \sqrt{\gamma\sigma^2}$, the volatility term matches the noise factor.

The mean of the process is $\mathbb{E}(\theta_t) = \theta^* + (\mathbb{E}(\theta_0) - \theta^*)e^{-t}$.

The variance of the process is $Var(X_t) = \frac{\gamma\sigma^2}{2}(1 - e^{-2t})$

The process converges to Gaussian Distribution with mean θ^* and variance $\frac{\gamma\sigma^2}{2}$, since the mean reversion term represents a force that pulls the process back towards the mean θ^* when θ_t deviates from it.



5 = 2

$\int f dy$

Simulation 2: OU process with SME

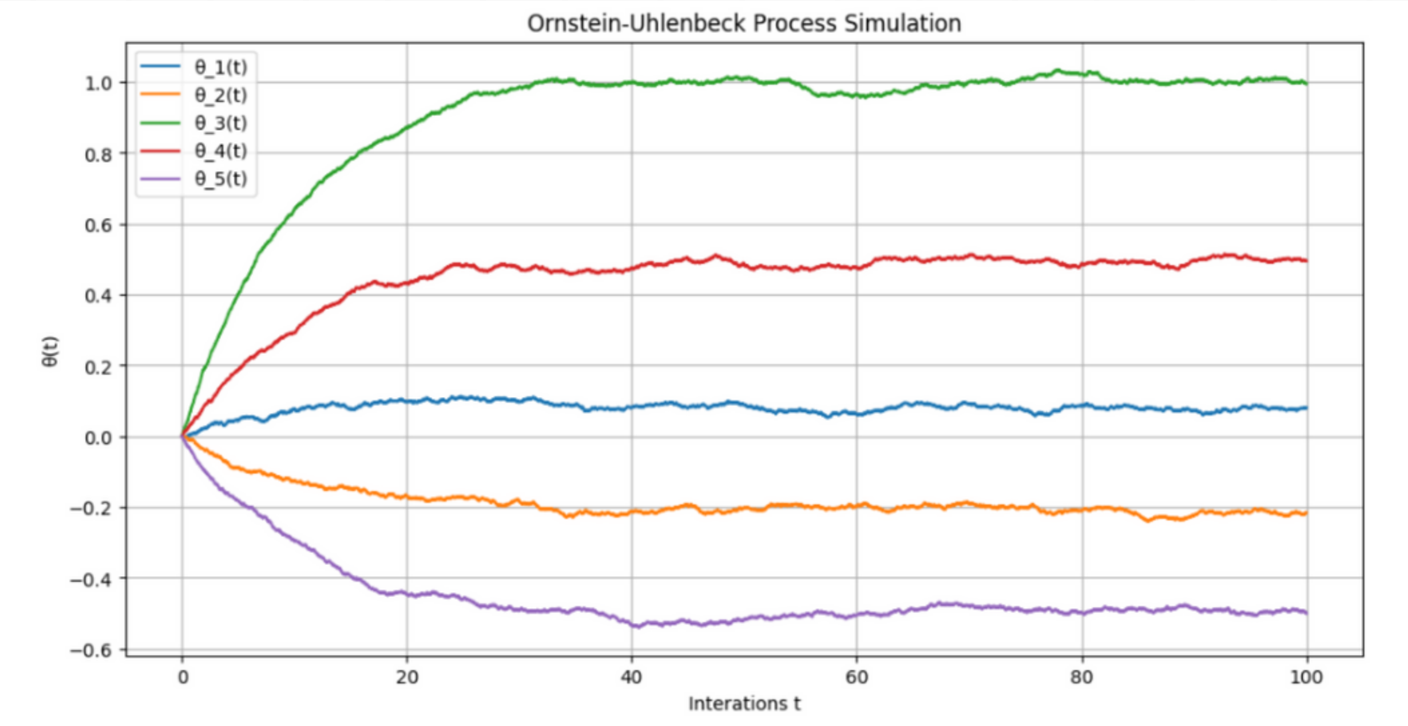
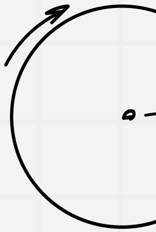


Figure 16: Simulation of OU process



5 = 2

3. Score-based Generative Model with SDE

$$\mathbf{s}(\mathbf{x}) \triangleq \nabla_{\mathbf{x}} \log p_{\text{data}}(\mathbf{x})$$

- Concepts clarification

  Score Matching, SMLD, DDPM

- Denoising diffusion probabilistic model(DDPM) with SDE
- Score Matching Langevin Dynamics(SMLD) with SDE
- Connection between noise and score

fdy

Our objective

1. How the estimated score approximates the gradient $\nabla_x \log p_{\mathrm{d}}(x)$, which facilitates the generation of new samples from p_{d}, forming the basis of the SMLD model.

2. How the noise scheduling and diffusion process enables the DDPM model to iteratively generate high-quality samples from the learned data distribution.

3. How the DDPM and SMLD models are linked through stochastic differential equations (SDE), with both models using score-based generative techniques to reverse the noise addition process.

5 = 2

fdy

Denoising Diffusion Models

Denoising diffusion models

- **Forward / noising process**
  - Sample data $p(\mathbf{x}_0)$ → turn to noise

$p_0(\mathbf{x}_0)$

$p_T(\mathbf{x}_T) \sim \mathcal{N}(0, I$

Clean sample $\mathbf{x}_0$ $\mathbf{x}_1$ $\mathbf{x}_{T-1}$ $\mathbf{x}_T$ *Pure noise*

- **Reverse / denoising process**
  - Sample noise $p_T(\mathbf{x}_T)$ → turn into data

Image source: Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. arXiv preprint arXiv:1907.05600, 2019. NeurIPS 2019.

5 = 2

fdy

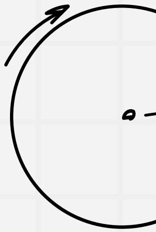
Denoising Diffusion Models with SDE

$$dx = \lim_{\Delta t \to 0} (x_{t+\Delta t} - x_t)$$

$$dx = f_t(x)dt + g_t dw$$

$$\Rightarrow$$

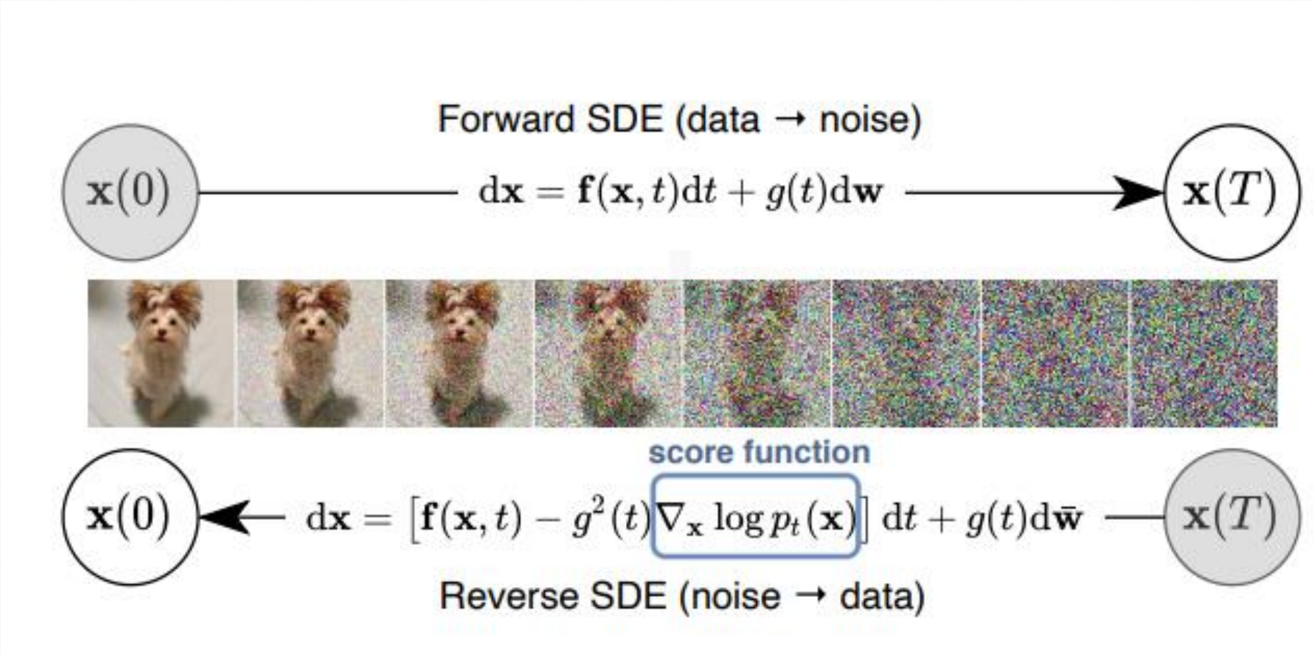
$$x_{t+\Delta t} = x_t + \underbrace{f_t(x_t)\Delta t}_{\text{drift term}} + \underbrace{g_t \sqrt{\Delta t} \epsilon}_{\text{diffusion term}} \;, \quad \epsilon \sim \mathcal{N}(0, I)$$



5 = 2

fdy

Denoising Diffusion Models with SDE



Forward SDE: $d\mathbf{x} = \mathbf{f}(\mathbf{x}, t)dt + g(t)d\mathbf{w}$

Reverse SDE: $d\mathbf{x} = \left[\mathbf{f}(\mathbf{x}, t) - g^2(t)\boxed{\nabla_{\mathbf{x}} \log p_t(\mathbf{x})}\right] dt + g(t)d\bar{\mathbf{w}}$

Image source: Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. arXiv preprint arXiv:2011.13456v2, 2020.

5 = 2

fdy

What is Score Matching Langevin Dynamics(SMLD)?

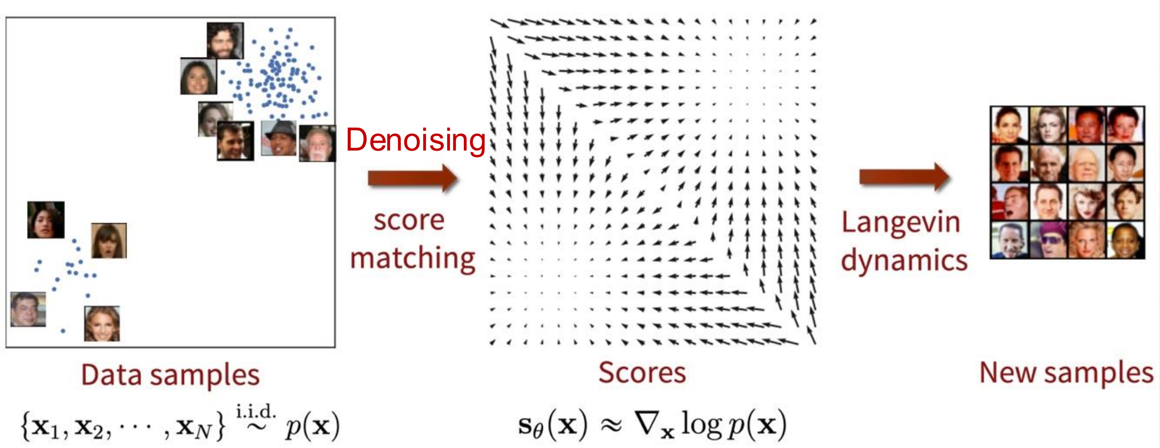
$$Loss = \frac{1}{2} E_{p_{data}(x)}[\|\overset{\text{True score}}{\boxed{\nabla_x log p(x)}} - \overset{\text{Score Network}}{\boxed{s_\theta(x)}}\|_2^2]$$

Score Matching (2005)

$$E_{p_{data}(x)}\left[tr(\overset{\text{Jacobian Matrix (d*d)}}{\boxed{\nabla_x s_\theta(x)}}) + \frac{1}{2}\|s_\theta(x)\|_2^2\right]$$

Denoising Score Matching (2011)

$$\frac{1}{2} E_{q_\sigma(\tilde{x}|x)p_{data}(x)}[\|s_\theta(\tilde{x}) - \nabla_x log q_\sigma(\tilde{x}|x)\|_2^2]$$



Diffusion Formula:

$$\begin{aligned} x_{t+1} &= x_t + \epsilon \nabla_{x_t} \log p(x_t) + \sqrt{2\epsilon} z_t \\ &= x_t + \epsilon s_\theta^*(x_t) + \sqrt{2\epsilon} z_t \end{aligned}$$

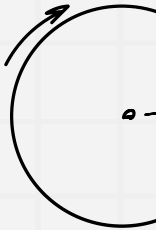


Image source: Chirag Pabbaraju, Dhruv Rohatgi, Anish Sevekari, Holden Lee, Ankur Moitra, and Andrej Risteski. Provable benefits of score matching. arXiv preprint arXiv:2306.01993, 2023.

5 = 2

fdy

SMLD with SDE

Diffusion formula

$$x_{t+1} = x_t + \epsilon \nabla_{x_t} \log p(x_t) + \sqrt{2\epsilon} z_t$$
$$= x_t + \epsilon s_\theta^*(x_t) + \sqrt{2\epsilon} z_t$$

Forward & reverse SDE

$$dx = f_t(x)dt + g_t dw$$

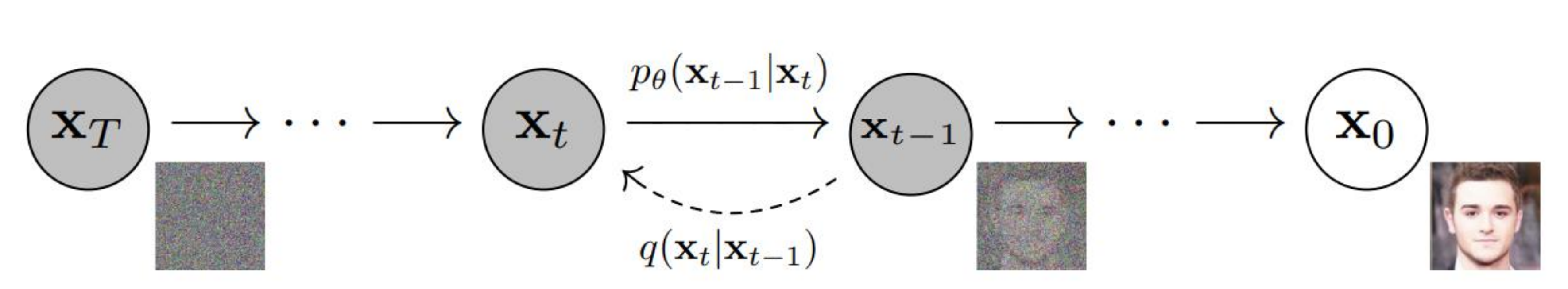
$$dx = [f_t(x) - g_t^2 \nabla_x \log p_t(x)]dt + g_t dw$$

$$\begin{cases} f(x_t, t) = 0 \\ g(t) = \frac{d}{dt} s_\theta^*(x_t)^2 \end{cases}$$

$S = 2$

fdy

What is Denoising diffusion probabilistic model (DDPM)?



Diffusion Formula:

$$\begin{aligned} x_t &= \sqrt{\bar{\alpha}_t}x_0 + \sqrt{1-\bar{\alpha}_t}\epsilon \\ &= \sqrt{\alpha_t}x_{t-1} + \sqrt{1-\alpha_t}\epsilon_t \end{aligned}$$

Image source: Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. arXiv preprint arXiv:2006.11239v2, pages 1–4, 13–14, 2020.

5 = 2

fdy

DDPM with SDE

Diffusion formula

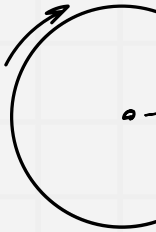
$$
\begin{aligned}
x_t &= \sqrt{\bar{\alpha}_t} x_0 + \sqrt{1-\bar{\alpha}_t}\epsilon \\
&= \sqrt{\alpha_t} x_{t-1} + \sqrt{1-\alpha_t}\epsilon_t
\end{aligned}
$$

Forward & reverse SDE

$$dx = f_t(x)dt + g_t dw$$

$$dx = [f_t(x) - g_t^2 \nabla_x \log p_t(x)]dt + g_t dw$$

$$
\begin{cases}
f(x_t, t) = -\frac{1}{2}\beta(t)x_t \\
g(t) = \sqrt{\beta(t)}
\end{cases}
$$



5 = 2

fdy

Simulation 3: DDPM

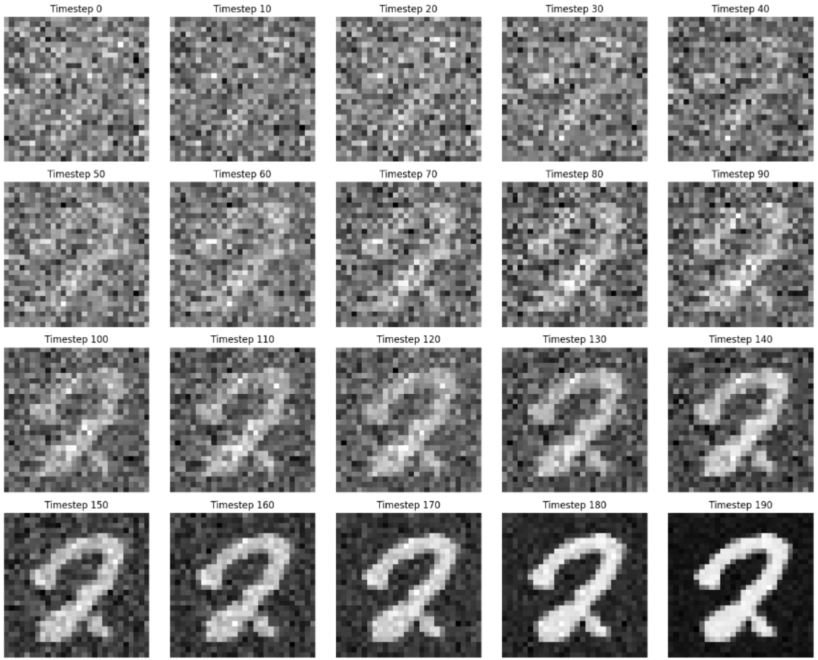


Figure 18: The denoised process of a random image from the MNIST dataset

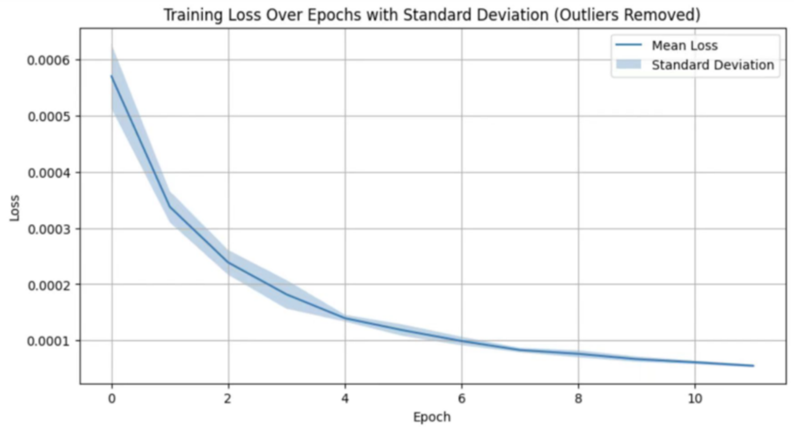


Figure 19: The convergent average loss for 5 runs

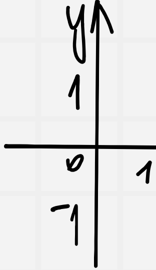
5 = 2

fdy

Connection between noise and score

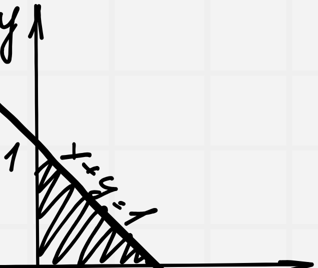
In the end, we would like to figure out the relationship between noise and score. In the SMLD model, the score $s_\theta(x_t, t)$ is estimated, while in the DDPM model, the noise $\epsilon_\theta(x_t, t)$ is predicted. If the correlation between score and noise can be found, we can train the DDPM model under the framework of SDE by estimating the score.

$$s_\theta(x_t, t) = -\frac{1}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(x_t, t)$$



4. Limitations & Future Directions

- Lack numerical experiments for SMLD
- Need more epoch & samples for numerical experiments



Thank You!

Ruiqi Ge(Selena) rg4012@nyu.edu
Supervisor: Prof. Mathieu Lauriere

$1/\sqrt{2}$

Reference

- Alnur Ali, Edgar Dobriban, and Ryan Tibshirani. The implicit regularization of stochastic gradient flow for least squares. In International conference on machine
- learning, pages 233–244. PMLR, 2020.
- F. Bach and E. Moulines. Non-asymptotic analysis of stochastic approximation algorithms for machine learning. In Advances in Neural Information Processing Systems, pages 451–459, 2011.
- Michel Benaïm. Dynamics of stochastic approximation algorithms. In Seminaire de probabilites XXXIII, pages 1–68. Springer, 2006.
- Fischer Black and Myron S. Scholes. The pricing of options and corporate liabilities. Journal of Political Economy, 81(3):637–654, 1973.
- Léon Bottou, Frank E Curtis, and Jorge Nocedal. Optimization methods for large-scale machine learning. SIAM Review, 60(2):223–311, 2018.
- V. Cevher and Bng Công Vũ. On the linear convergence of the stochastic gradient method with constant step-size. Optimization Letters, 13(5):1177–1187, 2019.
- Chen-Hao Chao, Wei-Fang Sun, Bo-Wun Cheng, Yi-Chen Lo, Chia-Che Chang, Yu-Lun Liu, Yu-Lin Chang, Chia-Ping Chen, and Chun-Yi Lee. Denoising likelihood score matching for conditional score-based data generation. arXiv preprint arXiv:2203.14206, 2022.
- Valentin De Bortoli, Michael Hutchinson, Peter Wirnsberger, and Arnaud Doucet. Target score matching. arXiv preprint arXiv:2402.08667, 2024.
- J Harold, G Kushner, and George Yin. Stochastic approximation and recursive algorithm and applications. Application of Mathematics, 35, 1997.
- Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. arXiv preprint arXiv:2006.11239v2, pages 1–4, 13–14, 2020.
- Aapo Hyvärinen and Peter Dayan. Estimation of non-normalized statistical models by score matching. Journal of Machine Learning Research, 6(4):695–709, 2005.
- Kiyosi Ito. Calculus of Variations. Springer, 1965.
- Rafail Khasminskii. Stochastic stability of differential equations, volume 66. Springer Science & Business Media, 2011.
- Qianxiao Li, Cheng Tai, and E Weinan. Stochastic modified equations and dynamics of stochastic gradient algorithms i: Mathematical foundations. The Journal of
- Machine Learning Research, 20(1):1474–1520, 2019.
- Bernt Oksendal. Stochastic differential equations: an introduction with applications, volume 3. Springer, 2003.
- Chirag Pabbaraju, Dhruv Rohatgi, Anish Sevekari, Holden Lee, Ankur Moitra, and Andrej Risteski. Provable benefits of score matching. arXiv preprint arXiv:2306.01993, 2023.
- Scott Pesme, Loucas Pillaud-Vivien, and Nicolas Flammarion. Implicit bias of sgd for diagonal linear networks: a provable benefit of stochasticity. Advances in Neural Information Processing Systems, 34:29218–29230, 2021.
- H. Robbins and S. Monro. A stochastic approximation method. Ann. Math. Statistics,
- 22:400–407, 1951.
- Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015, pages 234–241. Springer, 2015.
- O. Shamir and T. Zhang. Stochastic gradient descent for non-smooth optimization: convergence results and optimal averaging schemes. In International Conference on Machine Learning, pages 71–79, 2013.
- Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. arXiv preprint arXiv:1907.05600, 2019. NeurIPS 2019.
- Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. arXiv preprint arXiv:2011.13456v2, 2020.
- George E. Uhlenbeck and Leonard S. Ornstein. On the theory of the brownian motion. Physical Review, 36(5):823, 1930.
- Richard S Varga. Geršgorin and his circles, volume 36. Springer Science & Business Media, 2010.

$= \int^{1} dx \int^{1-x} x^2$