

2025 年 9 月 15 日至 9 月 21 日周报

何瑞杰
中山大学, 大湾区大学

1. 项目进展

1.1. 使用神经网络学习生命游戏的演化动力学

本周由于时间规划原因没有做代码方面的更新。

找到了互联网上的一个模拟生命游戏的程序 Golly，其文档不仅包含原始版本的生命游戏，还包含极其丰富的变种，具体见 <https://golly.sourceforge.io/Help/algos.html>。可能可以通过下载该程序或对应的 python 包，以改进生成数据的代码，使得更换规则更加容易。

杨老师提到了 Gray-Scott 系统和交通流模型（<https://www.thp.uni-koeln.de/~as/Mypage/traffic.html>），前者我在上周自行寻找材料做了简单的阅读（见上周周报），且没想到如何用神经网络学习。交通流模型阅读后发现其属于一维元胞自动机，但目测情况下难以使用卷积网络学习其规则。

下周计划内容详见最后一节。

2. 文献阅读

2.1. Denoising Diffusion Probabilistic Models

Jonathan Ho, Ajay Jain and Pieter Abbeel | <https://arxiv.org/abs/2006.11239>

原论文中的推导跳过了大量细节，通过这些细节和来龙去脉花了较长时间，留下实验、结论、讨论和附录部分下周完成。

2.1.1. 综观

不同于 VAE 等等的一个隐变量空间的生成模型，DDPM 尝试用多步编码/解码，或是加噪/去噪方式生成数据。它的每一步可以视为是一个去噪自编码器（Denoising Autoencoder），而其采样（生成）过程通过与加噪过程的 Markov 链“共用”，并通过预测加噪过程中的随机噪声实现加噪过程的“反转”，并添上 Langevin 动力学的随机噪声实现对分布的采样。与正常的 Langevin 动力学不同的是，从第 t 步到 $t-1$ 步，我们认为其在尝试适配一个分布 $p(\mathbf{x}_{t-1}|\mathbf{x}_t)$ 过程中，只做了一次 Langevin 动力学的迭代。

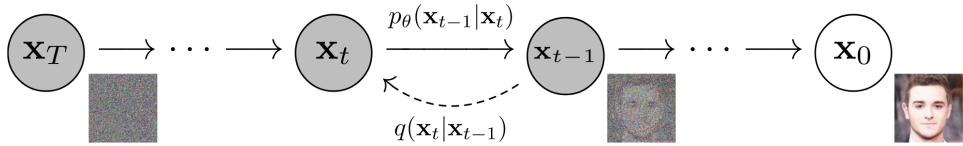


Figure 1: 图 1. DDPM 的工作流程示意图

2.1.2. 加噪过程的建模

首先对于加噪过程，我们有一列噪声强度 $\{\beta_t\}_{t=0}^T$ ，然后按照 $q(\mathbf{x}_t|\mathbf{x}_{t-1}) \sim N(\mathbf{x}_{t-1} | \beta_t \mathbf{I})$ 来进行：

$$\begin{aligned}
 \mathbf{x}_t &= \sqrt{1 - \beta_t} \mathbf{x}_{t-1} + \sqrt{\beta_t} \boldsymbol{\varepsilon}_t, & \boldsymbol{\varepsilon}_t &\sim N(\mathbf{0}, \mathbf{I}) \\
 &= \sqrt{1 - \beta_t} [\sqrt{1 - \beta_{t-1}} \mathbf{x}_{t-2} + \sqrt{\beta_{t-1}} \boldsymbol{\varepsilon}_{t-1}] + \sqrt{\beta_t} \boldsymbol{\varepsilon}_t, & \boldsymbol{\varepsilon}_{t-1} &\sim N(\mathbf{0}, \mathbf{I}) \\
 &= \sqrt{(1 - \beta_t)(1 - \beta_{t-1})} \mathbf{x}_{t-2} + \sqrt{(1 - \beta_t)\beta_{t-1}} \boldsymbol{\varepsilon}_{t-1} + \sqrt{\beta_t} \boldsymbol{\varepsilon}_t \\
 &= \sqrt{(1 - \beta_t)(1 - \beta_{t-1})} \mathbf{x}_{t-2} + \sqrt{(1 - \beta_t)\beta_{t-1} - (1 - \beta_t) + 1} \bar{\boldsymbol{\varepsilon}}_t, & \bar{\boldsymbol{\varepsilon}}_t &\sim N(\mathbf{0}, \mathbf{I}) \\
 &= \sqrt{(1 - \beta_t)(1 - \beta_{t-1})} \mathbf{x}_{t-2} + \sqrt{1 - (1 - \beta_t)(1 - \beta_{t-1})} \bar{\boldsymbol{\varepsilon}}_t \\
 &\vdots \\
 &= \sqrt{\prod_{s=1}^t (1 - \beta_s)} \mathbf{x}_0 + \sqrt{1 - \prod_{s=1}^t (1 - \beta_s)} \bar{\boldsymbol{\varepsilon}}_t = \sqrt{\bar{\alpha}_t} \mathbf{x}_0 + \sqrt{1 - \bar{\alpha}_t} \bar{\boldsymbol{\varepsilon}}_t
 \end{aligned}$$

这给出了加噪过程中每一步得到的 \mathbf{x}_t 更加便捷的表示，对后续的推导有帮助。

2.1.3. 优化目标

由于我们的目标是从随机噪声 \mathbf{x}_0 生成图像，即 \mathbf{x}_T ，因此需要最大化的项是 $\mathbb{E}_{p_\theta}[-\log p_\theta(\mathbf{x}_0)]$ 。因此我们需要继续用变分推断的技巧，逐步将其转化为可以计算得到的项。首先我们有下面的上界

$$\begin{aligned}
 &\mathbb{E}_{p_\theta}[-\log p_\theta(\mathbf{x}_0)] \\
 &= \mathbb{E}_{p_\theta} \left[-\log p_\theta(\mathbf{x}_0) \int q(\mathbf{x}_{1:T}|\mathbf{x}_0) d\mathbf{x}_{1:T} \right] = \mathbb{E}_{p_\theta} \left[- \int q(\mathbf{x}_{1:T}|\mathbf{x}_0) \log p_\theta(\mathbf{x}_0) d\mathbf{x}_{1:T} \right] \\
 &= -\mathbb{E}_{p_\theta(\mathbf{x}_0), q(\mathbf{x}_{1:T}|\mathbf{x}_0)} \left[\log \frac{p_\theta(\mathbf{x}_0)p_\theta(\mathbf{x}_{1:T}|\mathbf{x}_0)}{p_\theta(\mathbf{x}_{1:T}|\mathbf{x}_0)} \right] = -\mathbb{E}_{p_\theta(\mathbf{x}_0), q(\mathbf{x}_{1:T}|\mathbf{x}_0)} \left[\log \frac{p_\theta(\mathbf{x}_{0:T})q(\mathbf{x}_{1:T}|\mathbf{x}_0)}{p_\theta(\mathbf{x}_{1:T}|\mathbf{x}_0)q(\mathbf{x}_{0:T}|\mathbf{x}_0)} \right]
 \end{aligned}$$

$$\begin{aligned}
&= -\mathbb{E}_{p_\theta(\mathbf{x}_0), q(\mathbf{x}_{1:T}|\mathbf{x}_0)} \left[\log \frac{p_\theta(\mathbf{x}_{0:T})}{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} \right] - \mathbb{E}_{p_\theta(\mathbf{x}_0), q(\mathbf{x}_{1:T}|\mathbf{x}_0)} \left[\log \frac{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}{p_\theta(\mathbf{x}_{1:T}|\mathbf{x}_0)} \right] \\
&= -\mathbb{E}_{p_\theta(\mathbf{x}_0), q(\mathbf{x}_{1:T}|\mathbf{x}_0)} \left[\log \frac{p_\theta(\mathbf{x}_{0:T})}{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} \right] - \underbrace{\mathbb{E}_{p_\theta(\mathbf{x}_0)} [D_{\text{KL}}[q(\mathbf{x}_{1:T}|\mathbf{x}_0) \mid p_\theta(\mathbf{x}_{1:T}|\mathbf{x}_0)]]}_{\geq 0} \\
&\leq \mathbb{E}_q \left[\log \frac{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}{p_\theta(\mathbf{x}_{0:T})} \right] =: L
\end{aligned}$$

上式中的 $\mathbf{x}_{1:T}$ 不好处理，我们可以将打包的变量拆开，最后可以将其写成若干项 KL 散度之和，其中两项对应着 Markov 链头和尾，剩余的对应加噪过程的中间状态。

$$\begin{aligned}
L &= \mathbb{E}_q \left[\log \frac{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}{p_\theta(\mathbf{x}_{0:T})} \right] \\
&= \mathbb{E}_q \left[\log \frac{q(\mathbf{x}_T|\mathbf{x}_0) \prod_{t=2}^T q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)}{p_\theta(\mathbf{x}_T) \prod_{t=1}^T p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)} \right] \\
&= \mathbb{E}_q \left[\log \frac{q(\mathbf{x}_T|\mathbf{x}_0)}{p(\mathbf{x}_T)} + \sum_{t=2}^T \frac{q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)}{p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)} + \log \frac{1}{p_\theta(\mathbf{x}_0|\mathbf{x}_1)} \right] \\
&= \mathbb{E}_q \left[D_{\text{KL}}[q(\mathbf{x}_T|\mathbf{x}_0) \mid p(\mathbf{x}_T)] + \sum_{t=2}^T D_{\text{KL}}[q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) \mid p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)] - \log p_\theta(\mathbf{x}_0|\mathbf{x}_1) \right] \\
&= \mathbb{E}_q \left[L_T + \sum_{t=2}^T L_{t-1} - L_0 \right].
\end{aligned}$$

显然，得到的结果符合我们的预期，我们需要训练一个带参分布 $p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \mathbf{x}_0)$ ，并让其与 $q(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \mathbf{x}_0)$ 对齐。后者是一个 Gauss 分布，使用 Bayes 公式不难得到它的分布参数

$$\begin{aligned}
&q(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \mathbf{x}_0) \\
&= q(\mathbf{x}_t \mid \mathbf{x}_{t-1}, \mathbf{x}_0) \frac{q(\mathbf{x}_{t-1} \mid \mathbf{x}_0)}{q(\mathbf{x}_t \mid \mathbf{x}_0)} \\
&= q(\mathbf{x}_t \mid \mathbf{x}_{t-1}) \frac{q(\mathbf{x}_{t-1} \mid \mathbf{x}_0)}{q(\mathbf{x}_t \mid \mathbf{x}_0)} \\
&\propto \exp \left\{ -\frac{1}{2} \left[\frac{\|\mathbf{x}_t - \sqrt{\alpha_t} \mathbf{x}_{t-1}\|^2}{\beta_t} + \frac{\|\mathbf{x}_{t-1} - \sqrt{\alpha_{t-1}} \mathbf{x}_0\|^2}{1 - \bar{\alpha}_{t-1}} - \frac{\|\mathbf{x}_t - \sqrt{\alpha_t} \mathbf{x}_0\|^2}{1 - \bar{\alpha}_t} \right] \right\} \\
&= \exp \left\{ -\frac{1}{2} \left[\frac{\|\mathbf{x}_t\|^2 - \sqrt{\alpha_t} \langle \mathbf{x}_t, \mathbf{x}_{t-1} \rangle + \alpha_t \|\mathbf{x}_{t-1}\|^2}{\beta_t} + \frac{\|\mathbf{x}_{t-1}\|^2 - \sqrt{\alpha_{t-1}} \langle \mathbf{x}_{t-1}, \mathbf{x}_0 \rangle + \bar{\alpha}_{t-1} \|\mathbf{x}_0\|^2}{1 - \bar{\alpha}_{t-1}} \right. \right. \\
&\quad \left. \left. - \frac{\|\mathbf{x}_t\|^2 - \sqrt{\alpha_t} \langle \mathbf{x}_t, \mathbf{x}_0 \rangle + \bar{\alpha}_t \|\mathbf{x}_0\|^2}{1 - \bar{\alpha}_t} \right] \right\} \\
&= \exp \left\{ -\frac{1}{2} \left[\left(\frac{\alpha_t}{\beta_t} + \frac{1}{1 - \bar{\alpha}_{t-1}} \right) \|\mathbf{x}_{t-1}\|^2 - \left\langle \frac{\sqrt{\alpha_t}}{\beta_t} \mathbf{x}_t + \frac{\sqrt{\alpha_{t-1}}}{1 - \bar{\alpha}_{t-1}} \mathbf{x}_0, \mathbf{x}_{t-1} \right\rangle + \text{constant} \right] \right\} \\
&\propto \exp \left\{ \frac{\|\mathbf{x}_{t-1} - \tilde{\mu}_t\|^2}{2\tilde{\beta}_t} \right\}
\end{aligned}$$

其中均值 $\tilde{\mu}_t$ 和方差 $\tilde{\beta}_t$ 为

$$\tilde{\beta}_t = \frac{1}{\frac{\alpha_t}{\beta_t} + \frac{1}{1 - \bar{\alpha}_{t-1}}} = \frac{1 - \bar{\alpha}_{t-1}}{1 - \bar{\alpha}_t} \cdot \beta_t$$

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

由于需要匹配一个 Gauss 分布，带参分布 $q(\mathbf{x}_{t-1} | \mathbf{x}_t, \mathbf{x}_0)$ 也需要是一个 Gauss 分布 $N(\boldsymbol{\mu}_{\theta,t}, \Sigma_{\theta,t})$ ，不过在此我们将协方差矩阵简化为对角，即 $\boldsymbol{\mu}_{\theta,t} = \beta_{\theta,t} \mathbf{I}$ 。此时我们可以求中间过程的 KL 散度，由于其参数是两个 Gauss 分布，我们有现成的结论：

$$\begin{aligned} L_{t-1} &= D_{\text{KL}}[q(\mathbf{x}_{t-1} | \mathbf{x}_t, \mathbf{x}_0) | p_{\theta}(\mathbf{x}_{t-1} | \mathbf{x}_t)] \\ &= \frac{1}{2} \left[\log \frac{|\tilde{\beta}_t \mathbf{I}|}{|\beta_{\theta,t} \mathbf{I}|} - n + \text{tr}(\tilde{\beta}_t^{-1} \mathbf{I} \beta_{\theta,t} \mathbf{I}) + (\tilde{\mu}_t - \boldsymbol{\mu}_{\theta,t})^\top \beta_{\theta,t}^{-1} \mathbf{I} (\tilde{\mu}_t - \boldsymbol{\mu}_{\theta,t}) \right] \\ &= \frac{1}{2\sigma_t} \left[\|\tilde{\mu}_t - \boldsymbol{\mu}_{\theta,t}\|^2 \right] + \text{constant}, \quad \text{令 } \beta_{\theta,t} \text{ 为只与时间相关的 } \sigma_t \\ &= \frac{1}{2\sigma_t} \left[\left\| \left[\frac{\sqrt{\alpha_t}(1 - \bar{\alpha}_{t-1})}{1 - \bar{\alpha}_t} \mathbf{x}_t + \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1 - \bar{\alpha}_t} \cdot \frac{1}{\sqrt{\alpha_t}} (\mathbf{x}_t - \sqrt{1 - \bar{\alpha}_t} \boldsymbol{\varepsilon}) \right] - \boldsymbol{\mu}_{\theta,t} \right\|^2 \right] + \text{constant} \\ &= \frac{1}{2\sigma_t} \left[\left\| \left[\frac{\alpha_t(1 - \bar{\alpha}_{t-1}) + \beta_t}{(1 - \bar{\alpha}_t)\sqrt{\alpha_t}} \mathbf{x}_t + \frac{\beta_t}{\sqrt{\alpha_t}\sqrt{1 - \bar{\alpha}_t}} \boldsymbol{\varepsilon} \right] - \boldsymbol{\mu}_{\theta,t} \right\|^2 \right] + \text{constant} \\ &= \frac{1}{2\sigma_t} \left[\left\| \left[\frac{1 - \bar{\alpha}_t}{(1 - \bar{\alpha}_t)\sqrt{\alpha_t}} \mathbf{x}_t - \frac{\beta_t}{\sqrt{\alpha_t}\sqrt{1 - \bar{\alpha}_t}} \boldsymbol{\varepsilon} \right] - \boldsymbol{\mu}_{\theta,t} \right\|^2 \right] + \text{constant}, \quad \alpha_t = 1 - \beta_t, \bar{\alpha}_t = \prod_{s=0}^t \alpha_t \\ &= \frac{1}{2\sigma_t} \left[\left\| \frac{1}{\sqrt{\alpha_t}} \left[\mathbf{x}_t - \frac{1 - \alpha_t}{\sqrt{1 - \bar{\alpha}_t}} \boldsymbol{\varepsilon} \right] - \boldsymbol{\mu}_{\theta,t} \right\|^2 \right] + \text{constant} \end{aligned}$$

进一步地，我们可以将预测均值 $\boldsymbol{\mu}_{\theta,t}$ 建模为与 $\tilde{\mu}_t$ 相同的形式，即

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

因此 L_{t-1} 还可以进一步简化

$$\begin{aligned} L_{t-1} &= \frac{1}{2\sigma_t} \left[\left\| \frac{1}{\sqrt{\alpha_t}} \left[\mathbf{x}_t - \frac{1 - \alpha_t}{\sqrt{1 - \bar{\alpha}_t}} \boldsymbol{\varepsilon} \right] - \boldsymbol{\mu}_{\theta,t} \right\|^2 \right] + \text{constant} \\ &= \frac{1}{2\sigma_t} \left[\left\| \frac{1}{\sqrt{\alpha_t}} \left[\mathbf{x}_t - \frac{1 - \alpha_t}{\sqrt{1 - \bar{\alpha}_t}} \boldsymbol{\varepsilon} \right] - \frac{1}{\sqrt{\alpha_t}} \left[\mathbf{x}_t - \frac{1 - \alpha_t}{\sqrt{1 - \bar{\alpha}_t}} \boldsymbol{\varepsilon}_{\theta}(\mathbf{x}_t, t) \right] \right\|^2 \right] + \text{constant} \\ &= \frac{(1 - \alpha_t)^2}{2\sigma_t \alpha_t (1 - \bar{\alpha}_t)} \|\boldsymbol{\varepsilon} - \boldsymbol{\varepsilon}_{\theta}(\mathbf{x}_t, t)\|^2 + \text{constant} \end{aligned}$$

推了半天之后我们发现我们的优化目标也只是预测噪声而已（而这并不是新鲜事），我们就可以很顺利地得到 DDPM 的训练和采样算法。我们也可以使用更加简单的目标函数，只需要将 L_{t-1}

前面的系数扔掉，同时考虑服从离散均匀分布的 t 即可。在下面的算法中，我们实际上也是随机从均匀分布中采样 t 然后训练模型。

2.1.4. 训练和采样算法

Algorithm 1: Training

```

1: procedure TRAINING()
2:   while not converged do
3:      $\mathbf{x}_0 \sim q(\mathbf{x}_0)$ 
4:      $t \sim \text{Uniform}(\{1, \dots, T\})$ 
5:      $\boldsymbol{\varepsilon} \sim N(\mathbf{0}, \mathbf{I})$ 
6:      $\triangleright$  Perform gradient update
7:      $\theta \leftarrow \theta - \eta \nabla_{\theta} \|\boldsymbol{\varepsilon} - \boldsymbol{\varepsilon}_{\theta}(\sqrt{\bar{\alpha}_t} \mathbf{x}_0 + \sqrt{1 - \bar{\alpha}_t} \boldsymbol{\varepsilon}_{\theta}(\mathbf{x}_0, t))\|^2$ 
8:   end
9: end

```

Algorithm 2: Sampling

```

1: procedure SAMPLING()
2:    $\mathbf{x}_T \sim N(\mathbf{0}, \mathbf{I})$ 
3:   for  $t = T, \dots, 1$  do
4:     if  $t > 1$  then
5:        $\mathbf{z} \sim N(\mathbf{0}, \mathbf{I})$ 
6:     else
7:        $\mathbf{z} \leftarrow \mathbf{0}$ 
8:     end
9:      $\mathbf{x}_{t+1} \leftarrow \frac{1}{\sqrt{\alpha_t}} \left[ \mathbf{x}_t - \frac{1 - \alpha_t}{\sqrt{1 - \bar{\alpha}_t}} \boldsymbol{\varepsilon}_{\theta}(\mathbf{x}_t, t) \right] + \sigma_t \mathbf{z}$ 
10:   end
11:   return  $\mathbf{x}_0$ 
12: end

```

可以将其与下面的模拟退火 Langevin 动力学采样进行对比

Algorithm 3: Annealed Langevin Dynamics

```

1: procedure ANNEALED LANGEVIN DYNAMICS( $\{\sigma_i\}_{i=1}^n, \varepsilon, T$ )
2:    $\triangleright$  Initialize the search range
3:    $\bar{\mathbf{x}}_0 \leftarrow \mathbf{v}$ 
4:   for  $t \leftarrow 1, \dots, L$  do
5:      $\triangleright$  Set step size  $\alpha_i$ 
6:      $\alpha_i \leftarrow \varepsilon \cdot \sigma_i^2 / \sigma_L^2$ 
7:     for  $t \leftarrow 1, \dots, T$  do
8:        $\triangleright$  Draw  $\mathbf{z}_t \sim N(0, \mathbf{I})$ 
9:        $\bar{\mathbf{x}}_t \leftarrow \bar{\mathbf{x}}_{t-1} + \alpha_i / 2 \cdot s_{\theta}(\bar{\mathbf{x}}_{t-1}, \sigma_i) + \sqrt{\alpha_i} \mathbf{z}_t$ 
10:    end
11:     $\bar{\mathbf{x}}_0 \leftarrow \bar{\mathbf{x}}_T$ 
12:   end
13:   return  $\bar{\mathbf{x}}_T$ 
14: end

```

不难发现 DDPM 中的 Sampling 算法将 Annealed Langevin Dynamics 算法中的内层的循环减少到了 1。

2.1.5. 离散值图像生成的最后一步

由于需要生成的对象是计算机中离散编码（如八位）的图像，对应最后一步的 $p_\theta(\mathbf{x}_0|\mathbf{x}_1)$ 需要得到的是离散分布。论文使用了一个简便的技巧，假设图像的像素值在离散化过程中从 $\{0, 1, \dots, 255\}$ 线性地归一化到 $[-1, 1]$ 内，即 0 被映射到 -1 , 255 被映射到 1 。在计算 \mathbf{x}_0 的第 i 个像素值是 $\mathbf{x}_0^{(i)}$ 时，我们可以简单地将 \mathbb{R} 切分成 256 块，其中 $1 \sim 254$ 分别对应着 $\left[-1 - \frac{1}{255}, 1 + \frac{1}{255}\right]$ 中宽度为 $\frac{2}{255}$ 的小区间 $[\delta_-(x), \delta_+(x)]$ ，剩下的一头一尾就分别对应剩下的两个无限长度的区间。具体而言，

$$\delta_+(x) = \begin{cases} \infty & \text{if } x = 1 \\ x + \frac{1}{255} & \text{if } x < 1 \end{cases}, \quad \delta_-(x) = \begin{cases} -\infty & \text{if } x = -1 \\ x - \frac{1}{255} & \text{if } x > -1 \end{cases}$$

假设 \mathbf{x}_0 的分布中各分量相互独立，就有

$$p_\theta(\mathbf{x}_0) = \prod_{i=1}^n p_\theta(\mathbf{x}_0^{(i)}) = \prod_{i=1}^n \int_{\delta_-(\mathbf{x}_0^{(i)})}^{\delta_+(\mathbf{x}_0^{(i)})} N(x \mid \boldsymbol{\mu}_\theta^{(i)}(\mathbf{x}_1, 1), \sigma_1 \mathbf{I}) dx$$

这样也不会破坏分布的归一化性质。

2.1.6. 实验结果

[下周补齐](#)

2.1.7. 总结和讨论

[下周补齐](#)

参考资料

1. <https://arxiv.org/abs/1907.05600>
2. <https://arxiv.org/abs/2006.11239>
3. <https://lilianweng.github.io/posts/2021-07-11-diffusion-models/#nice>

3. 学习进度

3.1. 随机过程

首先复习回顾了有关于尾部概率的诸不等式：Markov 不等式、Chebyshev 不等式、Chernoff 界。我发现有一个更一般的情形可以轻松涵盖这三个不等式，以及我在 SDE 那本小册子中看到的 Chebyshev 不等式的版本：

定理 3.1.1 (拓展 Chebyshev 不等式): 考虑随机变量 $X : \Omega \rightarrow \mathbb{R}$, 和连续增函数 $g : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}_{\geq 0}$, 如果 $g(a) > 0$ 且 $\mathbb{E}(g(|X|))$ 存在, 有

$$\mathbb{E}(|X| \geq a) \leq \frac{\mathbb{E}(g(|X|))}{g(a)}$$

- 取 $X = Y - \mathbb{E}[Y]$, $g(x) = x^2$, 得到 Chebyshev 不等式: $\mathbb{E}(|Y - \mathbb{E}[Y]| \geq a) \leq \frac{\text{Var}(Y)}{a^2}$
- 取 $g(x) = x$, 得到 Markov 不等式: $\mathbb{E}(|X| \geq a) \leq \frac{\mathbb{E}(|X|)}{a}$
- 取 $g(x) = e^{\lambda x}$, 其中 $\lambda \geq 0$, 得到 Chernoff 界: $\mathbb{E}(|X| \geq a) \leq \frac{\mathbb{E}(\exp\{\lambda|X|\})}{\exp\{a\lambda\}}$
- 取 $g(x) = x^p$, 其中 $p \geq 1$, 得到下面另一形式的 Chebyshev 不等式: $\mathbb{E}(|X| \geq \lambda) \leq \frac{\mathbb{E}(|X|^p)}{\lambda^p}$

3.2. 随机微分方程

本周学习了 Itô 积分的链式法则、和乘积法则。具体而言，我们研究的是具有下列基本形式之随机微分的随机过程:

$$dX = F dt + G dW \iff X(r) = X(s) + \int_s^r F dt + \int_s^r G dW, \quad 0 \leq s \leq r \leq T$$

其中 $F \in \mathbb{L}^1(0, T)$, $G \in \mathbb{L}^2(0, T)$ 。链式法则是说对于函数 $u : \mathbb{R} \times [0, T] \rightarrow \mathbb{R}$, 如果 u_x, u_t, u_{xx} 都存在且连续，则有

$$\begin{aligned} dY &= du(X(t), t) = u_t dt + u_x dX + \frac{1}{2} u_{xx} G^2 dt \\ &= \left(u_t + u_x F + \frac{1}{2} u_{xx} G^2 \right) dt + u_x G dW \end{aligned}$$

或者说

$$\begin{aligned} Y(r) - Y(s) &= u(X(r), r) - u(X(s), s) \\ &= \int_s^r u_t + u_x F + \frac{1}{2} u_{xx} G^2 dt + \int_r^s u_x G dW \quad \text{a.s.} \end{aligned}$$

乘积法则说的是两个有上述基本形式之随机微分的过程

$$\begin{aligned} dX_1 &= F_1 dt + G_1 dW \\ dX_2 &= F_2 dt + G_2 dW \end{aligned}$$

其乘积有下面的随机微分:

$$d(X_1 X_2) = X_2 dX_1 + X_1 dX_2 + G_1 G_2 dt.$$

两个定理的证明具有一定的相似性。对于乘积法则，首先证明与时间无关的 F 和 G 情形下在 $[0, t]$ （其中 $t \leq T$ ）满足条件，然后再证在任意区间 $[s, t]$ 上成立，这是第一步；接着可以证明对任意的阶梯过程 $F \in \mathbb{L}^1(0, T)$ 和 $G \in \mathbb{L}^2(0, T)$ 都成立，这是第二步；最后利用 $\mathbb{L}^1(0, T)$ 和 $\mathbb{L}^2(0, T)$ 的完备性，利用近似的思想推广到任意的 $F \in \mathbb{L}^1(0, T)$ 和 $G \in \mathbb{L}^2(0, T)$ ，这是第三步。

链式法则的证明与此类似。对于函数 u ，首先考虑最简单的形式，即多项式 $u \in \mathbb{F}[x]$ ，然后再推广到任意的关于 x 和 t 的多项式的乘积形式。由于 $\mathbb{F}[x, t]$ 上的任意元素都可以写成 $\sum_{i=1}^n p_i(x)q_i(t)$ 的形式，其中 $p_i \in \mathbb{F}[x]$, $q_i \in \mathbb{F}[t]$ ，因此可以推广到任意的 $u \in \mathbb{F}[x, t]$ 。最后让一系列多项式逼近任意满足定理条件的 u 即可。

4. 问题解决记录

4.1. Typst 相关

4.1.1. 直接套用 LaTeX 公式

可以通过导入 `mitex` 包实现直接将 LaTeX 或 Markdown 文档中的行内和行间公式嵌入 typst 文档：

```
#import "@preview/mitex:0.2.4": *
#mitex(`

\begin{align} \mathrm{d} X_{\{1\}} &= F_{\{1\}} \\ &\quad \mathrm{d} t + G_{\{1\}} \mathrm{d} W \\ \mathrm{d} X_{\{2\}} &= F_{\{1\}} \mathrm{d} t + \\ &\quad G_{\{1\}} \mathrm{d} W \end{align}
`)
```

$$\mathrm{d} X_1 = F_1 \mathrm{d} t + G_1 \mathrm{d} W$$

$$\mathrm{d} X_2 = F_1 \mathrm{d} t + G_1 \mathrm{d} W$$

4.1.2. 一些更灵活的 typst 函数定义

在撰写一些细节推导时，不可避免地需要使用不同颜色指示推导中有不同功能的项。例如

```
$y = #text(red)[\$k\$]x$
```

$$y = \textcolor{red}{k}x$$

在推导过程很复杂时显然不适宜，我们可以定义下面的函数：

```
#let redText(text) = {text(red)[\$text\$]}
```

这样就可以更方便地使用该函数

```
$y = redText(k)x$
```

$$y = \textcolor{red}{t}x$$

类似还可以设计其他常用颜色的函数。

5. 下周计划

论文阅读

1. 生成模型

- 完成 Diffusion 剩下的部分
- Sliced Score Matching: A Scalable Approach to Density and Score Estimation
- Score-Based Generative Modeling through Stochastic Differential Equations

项目进度

1. 使用神经网络学习生命游戏的演化动力学

- 解释神经网络
 - ▶ 寻找更多神经网络权重的解释方法
- 其他的生命游戏规则
 - ▶ 探索并整理 Golly 文档中提到的可简单实现的若干种方法（五种左右）
 - ▶ 调整代码实现，使之可以各方面适配多数据集的情形（例如训练的输入输出、logging 的规格，以及输出文件的文件名和文件夹规格，需要包含所使用的数据集）
- 其他模型
 - ▶ 考虑交通流模型和一维元胞自动机使用神经网络学习规则的方法
 - ▶ 考虑如何通过神经网络模型预测 Gray-Scott 的动力学

理论学习

1. 随机过程课程

- 随机过程完成第一章和第二次作业

2. 随机微分方程

- Evans 完成至第五章第一节或第二节