Modelagem Generativa com Equações Diferenciais Estocásticas (SDEs): Uma Breve Revisão

Ezequiel Braga & Jairon Henrique

21 de junho de 2024

FGV EMAp
ESCOLA DE MATEMÁTICA APLICADA

Introdução

Modelagem generativa

- Objetivo: Criar dados a partir de ruído.
- Modelos mais comuns: introdução sequencial de ruídos nos dados de treinamento e aprende o processo reverso para formar o modelo generativo.
  - *Score matching with Langevin dynamics* (SMLD): estima a função *score* de cada ruído e usa dinâmica de Langevin para amostrar de uma sequência decrescente de ruídos.
  - *Denoising diffusion probabilistic modeling* (DDPM): treina uma sequência de modelos probabilísticos para reverter cada passo de ruído, usando o conhecimento da forma funcional das distribuições reversas.

Introdução

Uso de SDEs

- Ao invés de perturbar os dados com um número finito de distribuições de ruído, considera-se uma distribuição contínua que evolui ao longo do tempo através de um processo de difusão.
- Este processo difunde cada ponto de dado em um ruído através de uma SDE.

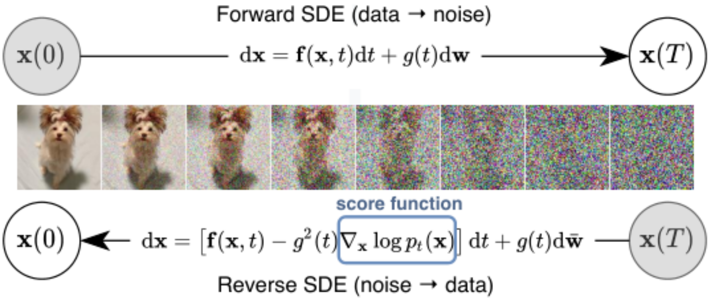


Figura: Ilustração do processo.

Introdução

Por que o uso da função *score* é importante?

- Dada uma densidade $p_0(\boldsymbol{x})$, o *score* é definido como sendo $\nabla_x \log p_0(\boldsymbol{x})$.
- Não sofre problema de normalização:

$$\nabla \log(p_0(\boldsymbol{x})/Z) = \nabla \log p_0(\boldsymbol{x}).$$

- Geração controlável: é possível modular o processo de geração condicionando em informações não disponíveis durante o treinamento, uma vez que a SDE condicional de tempo reverso pode ser eficientemente estimada através de *scores* não condicionais.
- Estrutura unificada: essa metodologia unifica a modelagem generativa, já que SMLD e DDPM podem ser vistos como discretizações de SDEs.

Modelagem generativa com SDEs

Perturbando os dados com SDEs

- Objetivo: construir um processo de difusão $\{\boldsymbol{x}(t)\}_{t=0}^{T}$ de modo que $\boldsymbol{x}_0 \sim p_0$ e $\boldsymbol{x}_T \sim p_T$, onde p_T é uma distribuição a priori e p_0 é a distribuição dos dados.
- O processo de difusão pode ser modelado como a solução de Itô da seguinte SDE: $d\boldsymbol{x} = f(\boldsymbol{x}, t)dt + g(t)dW$, onde W é o movimento Browniano, $f(\cdot, t) : \mathbb{R}^d \mapsto \mathbb{R}^d$ é uma função chamada de coeficiente de *drift* de $\boldsymbol{x}(t)$, e $g(\cdot) : \mathbb{R} \mapsto \mathbb{R}$ é uma função conhecida como coeficiente de difusão de $\boldsymbol{x}(t)$.

Modelagem generativa com SDEs

Perturbando os dados com SDEs

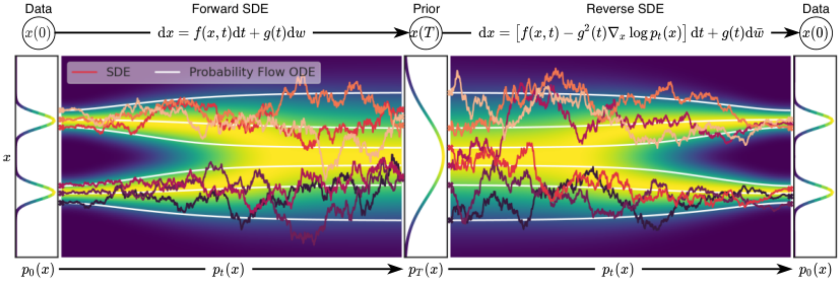


Figura: Visão geral da modelagem usando SDE

Iniciando com $\boldsymbol{x}(T) \sim p_T$, é possível obter $\boldsymbol{x}(0) \sim p_0$ com o processo de difusão reverso, dado pela SDE de tempo reverso, $\tilde{\boldsymbol{x}}(t) = \boldsymbol{x}(T - t)$:

$$d\tilde{\boldsymbol{x}} = [f(\tilde{\boldsymbol{x}}, t)dt - g(t)^2 \nabla_{\tilde{\boldsymbol{x}}} \log p_{T-t}(\tilde{\boldsymbol{x}})]dt + g(t)d\bar{W},$$

onde $\bar{W}$ é o movimento Browniano com tempo reverso de T a 0.

FGV EMAp ESCOLA DE MATEMÁTICA APLICADA

- Problema: $\nabla_{\boldsymbol{x}} \log p_t(\boldsymbol{x})$ é inacessível;
- Escolha natural: treinar uma rede neural $s_{\boldsymbol{\theta}}(\boldsymbol{x}, t)$, tomando $\boldsymbol{\theta}^\star$ tal que

$$\boldsymbol{\theta}^\star = \arg\min_{\boldsymbol{\theta}} \mathbb{E}_{t\sim\mathcal{U}[0,1]}[\lambda(t)\mathbb{E}_{\boldsymbol{x}_t\sim p_t}[\|s_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t) - \nabla \log p_t(\boldsymbol{x}_t)\|^2]], \tag{1}$$

  onde $\lambda(t)$ é uma função peso positiva.
- Outra problema: perda intratável.

Teorema 1

Se, para todo $\boldsymbol{\theta}$, $s_{\boldsymbol{\theta}}(\boldsymbol{x})$ e $p_{\text{data}}(\boldsymbol{x})$ são diferenciáveis, $\mathbb{E}_{p_{\text{data}}(\boldsymbol{x})}[\|s_{\boldsymbol{\theta}}(\boldsymbol{x})\|^2]$ e $\mathbb{E}_{p_{\text{data}}(\boldsymbol{x})}[\|\nabla_{\boldsymbol{x}} \log p_{\text{data}}(\boldsymbol{x})\|^2]$ são finito, e

$$\lim_{\|\boldsymbol{x}\| \to \infty} p_{\text{data}}(\boldsymbol{x}) s_{\boldsymbol{\theta}}(\boldsymbol{x}) = 0,$$

então é possível escrever

$$\mathbb{E}_{p_{\text{data}}(\boldsymbol{x})}\left[\|s_{\boldsymbol{\theta}}(\boldsymbol{x}) - \nabla_{\boldsymbol{x}} \log p_{\text{data}}(\boldsymbol{x})\|^2\right] = \mathbb{E}_{p_{\text{data}}(\boldsymbol{x})}\left[\|s_{\boldsymbol{\theta}}(\boldsymbol{x})\|^2 + 2\nabla_{\boldsymbol{x}} \cdot s_{\boldsymbol{\theta}}(\boldsymbol{x})\right] + C,$$

onde C não depende de $\boldsymbol{\theta}$.

Um resultado do Teorema 1 é que podemos reescrever a equação 1 como

$$\mathbb{E}_t[\lambda(t)\mathbb{E}_{p_0}[\mathbb{E}_{p_{t|0}}[||s_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t) - \nabla \log p_{t|0}(\boldsymbol{x}_t|\boldsymbol{x}_0)||^2]]] + C$$

e, consequentemente, basta tomar $\boldsymbol{\theta}^\star$ tal que

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

onde $\lambda : [0, T] \mapsto \mathbb{R}_{>0}$ é uma função peso positiva; t é amostrado uniformemente em $[0, T]$; $\boldsymbol{x}(0) \sim p_0(\boldsymbol{x})$ e $\boldsymbol{x}(t) \sim p_{0t}(\boldsymbol{x}(t)|\boldsymbol{x}(0))$. Uma escolha que é recomendada em [Son+21] é $\lambda \propto \frac{1}{\mathbb{E}\left[\|\nabla_{x_t} \log p_{t|0}(x_t|x_0)\|_2^2\right]}$.

Modelagem generativa com SDEs

SMLD e DDPM como discretizações de SDEs

- SMLD: discretização da VE SDE

$$dx = \sqrt{\frac{d\sigma^2(t)}{dt}} dW.$$

- DDPM: discretização da VP SDE

$$dx = -\frac{1}{2}\beta(t)x dt + \sqrt{\beta(t)} dW.$$

- Coeficientes de *drift* afins levam a kernels de pertubação $p_{0t}(\boldsymbol{x}(t)|\boldsymbol{x}(0))$ gaussianos, conforme abaixo:

$$p_{0t}(\boldsymbol{x}(t)|\boldsymbol{x}(0)) = \begin{cases} \text{MVN}\left(\boldsymbol{x}(t);\ \boldsymbol{x}(0),\ [\sigma^2(t) - \sigma^2(0)]I\right) \\ \text{MVN}\left(\boldsymbol{x}(t);\ \boldsymbol{x}(0)\exp\left(-\int_0^t \beta(s)ds\right),\ I - I\exp\left(-\int_0^t \beta(s)ds\right)\right) \end{cases}$$

Modelagem generativa com SDEs

Gerando amostras da SDE reversa

- Problema: desenvolver formas de amostragem ancestral para SDEs.
- Solução proposta (amostrador de difusão reversa): dada a equação *foward* $d\boldsymbol{x} = f(\boldsymbol{x}, t)dt + G(t)dW$ com a discretização $\boldsymbol{x}_{i+1} = \boldsymbol{x}_i + f_i(\boldsymbol{x}_i) + G_i z_i,\ i = 0, 1, \ldots, N-1$, onde $z_i \sim \text{MVN}\,(0,\ I)$, é proposto discretizar a SDE de tempo reverso abaixo:

$$d\boldsymbol{x} = [f(\boldsymbol{x}, t) - G(t)G(t)^\top \nabla_x \log p_t(\boldsymbol{x})]dt + G(t)d\bar{W},$$

cuja discretização (usando Euler-Maruyama) é

$$\boldsymbol{x}_i = \boldsymbol{x}_{i+1} - f_{i+1}(\boldsymbol{x}_{i+1}) + G_{i+1}G_{i+1}^\top s_{\boldsymbol{\theta}^\star}(\boldsymbol{x}_{i+1}, i+1) + G_{i+1}z_{i+1},\ i = 0, 1, \ldots, N-1,$$

para o *score* treinado $s_{\boldsymbol{\theta}^\star}(\boldsymbol{x}_i, i)$.

- É possível usar um esquema numérico mais complexo, fazendo um método conhecido como Preditor-Corretor.
- Mais detalhadamente, o solucionador da SDE fornece uma estimativa da amostra no próximo passo, fazendo o papel de "preditor", enquanto uma abordagem de MCMC corrige a distribuição marginal da amostra estimada, exercendo o papel de "corretor".

Para todo processo de difusão, existe um processo determinístico cuja trajetórias compartilham a mesma densidade de probabilidade marginal $\{p_t(\boldsymbol{x})\}_{t=0}^{T}$ com a SDE, desde que o ponto inicial seja amostrado conforma a distribuição inicial da SDE (que precisa ao menos ser C^2 com suporte cheio). Essa EDO é chamada de EDO de fluxo e é dada por:

$$d\boldsymbol{x} = \left[f(\boldsymbol{x}, t) - \frac{1}{2} g(t) \nabla_x \log p_t(\boldsymbol{x}) \right] dt.$$

- Podemos treinar um classificador que gera amostra de um determinado conjunto de imagens que tenha classes.
- A abordagem apresentada permite modelar cada classe com uma variável $\boldsymbol{y}$ e então cada imagem tem uma probabilidade $p_0(\boldsymbol{y}|\boldsymbol{x}(0))$ de pertencer a classe $\boldsymbol{y}$.
- Treinamos um classificador que aproxima $p_t(\boldsymbol{y}|\boldsymbol{x}(t))$ minimizando uma simples entropia cruzada e amostramos de uma classe específica usando a SDE

$$d\boldsymbol{x} = \left\{f(\boldsymbol{x}, t) - g(t)^2 \left[\nabla_{\boldsymbol{x}} \log p_t(\boldsymbol{x}|\boldsymbol{y})\right]\right\} dt + g(t) d\bar{W}.$$

Para a amostragem, usamos a regra de Bayes

$$p_t(\boldsymbol{x}|\boldsymbol{y}) \propto p_t(\boldsymbol{y}|\boldsymbol{x})p_t(\boldsymbol{x}),$$

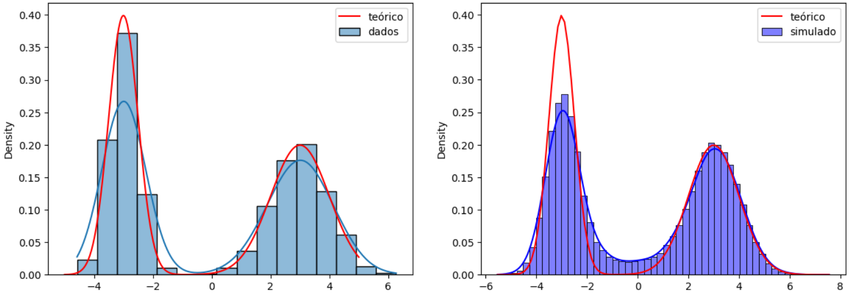
e então $\nabla \log p_t(\boldsymbol{x}|\boldsymbol{y}) = \nabla \log p_t(\boldsymbol{y}|\boldsymbol{x}) + \nabla \log p_t(\boldsymbol{x})$. Logo, a equação acima se escreve como

$$d\boldsymbol{x} = \left\{f(\boldsymbol{x}, t) - g(t)^2 \left[\nabla \log p_t(\boldsymbol{y}|\boldsymbol{x}) + \nabla \log p_t(\boldsymbol{x})\right]\right\} dt + g(t)d\bar{W}.$$

Resultados

Mistura de gaussianas

Usamos um dataset com 5000 pontos simulados de uma mistura de gaussianas, especificamente $\sim \frac{1}{2}\mathcal{N}(-3, 1/4) + \frac{1}{2}\mathcal{N}(3, 1)$.



(a) Dados verdadeiros

(b) Dados gerados pelo modelo

Figura: Mistura de Gaussianas

Usamos a SDE VP, uma MLP para o *score* e Euler-Maruayama para amostragem.

Resultados

Mistura de gaussianas

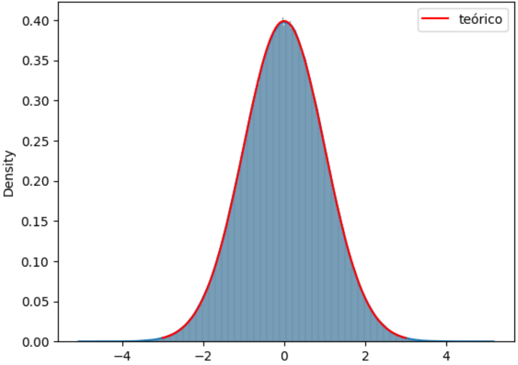
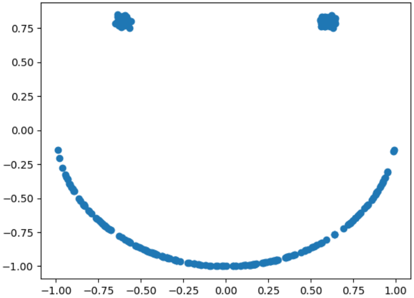


Figura: Dados Corrompido

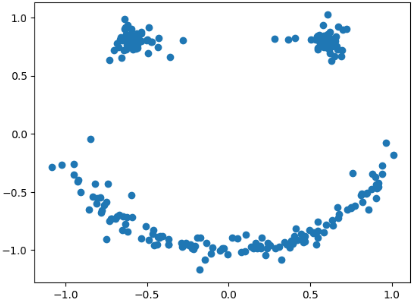
Resultados

Sorriso

Usamos um dataset com pontos gerados conforme a figura (5a).



(a) Dados reais



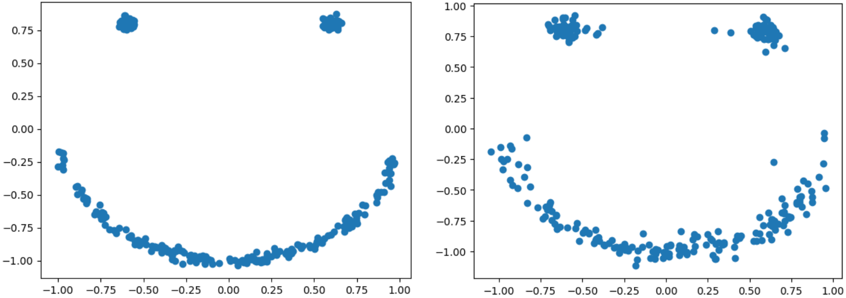
(b) Dados gerados pelo modelo

Figura: Dados do sorriso

Usamos a SDE VP, uma MLP e a EDO de fluxo.

Resultados

Sorriso com Ruído



(a) Dados reais (b) Dados gerados pelo modelo

Figura: Dados do sorriso

Resultados

Sorriso com Ruído

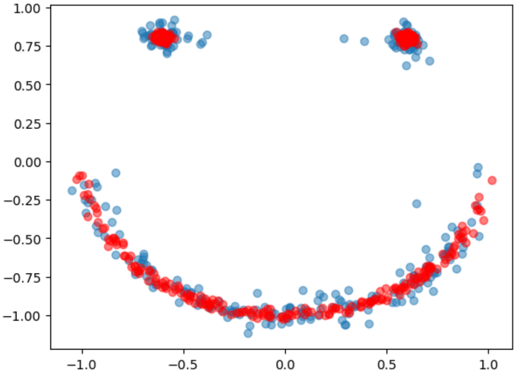


Figura: Dados justapostos.

Resultados

Dígitos escritos a mão

Usamos o conjunto de load_digits do scikit-learning como dado de treinamento.

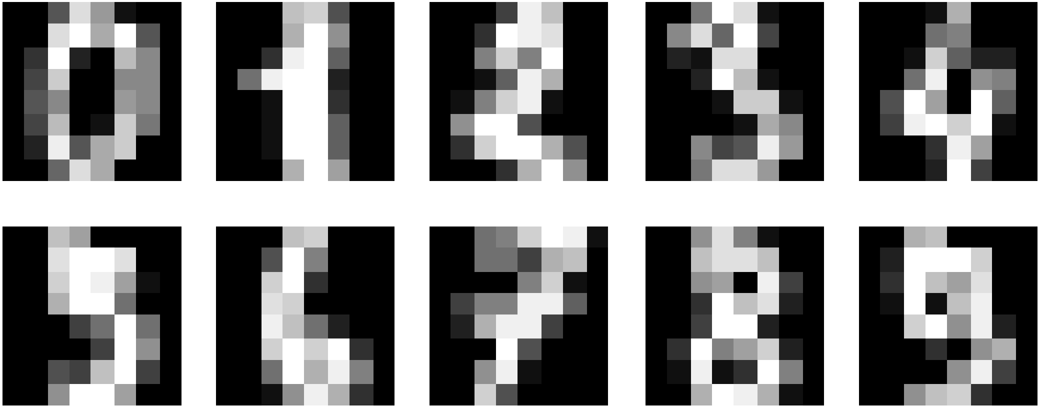


Figura: Amostra de números escritos a mão.

Resultados

Dígitos escritos a mão

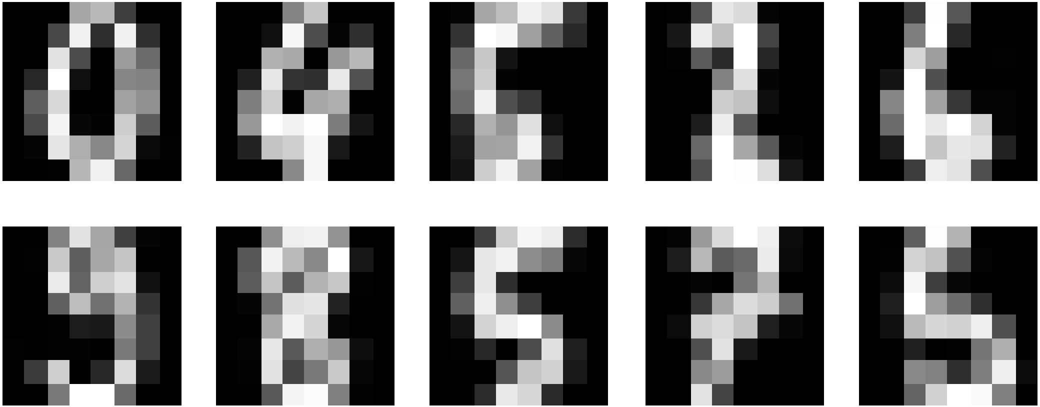


Figura: Amostra gerada de números escritos a mão.

Resultados

Dígitos escritos a mão

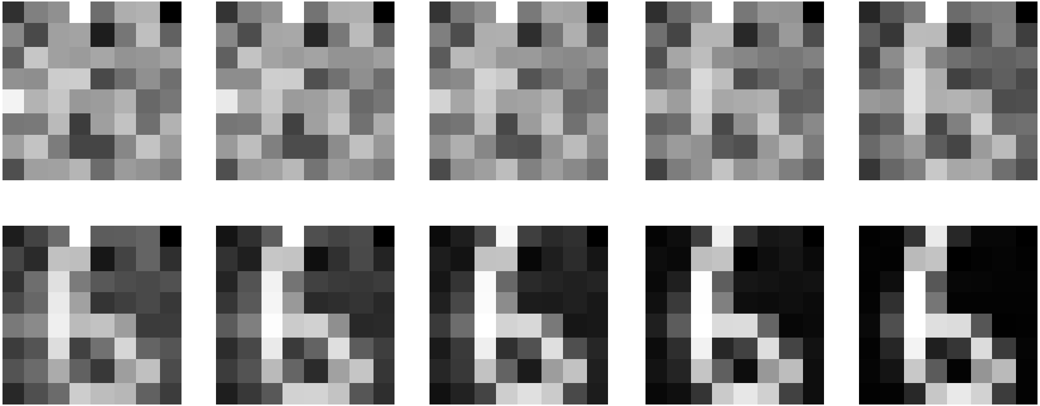


Figura: Passo da SDE reversa em uma geração.

A arquitetura desse exemplo é bem mais complicada. Primeiro precisamos de uma camada de *Projeções Gaussianas de Fourier* para transformar o input de tempo em 256 inputs. Depois intercalamos entre 7 camadas convolucionais e densa. Essa é a arquitetura sugerida pelo artigo e disponível no repositório dos autores. Treinamos com batch size de 4, usando 150 pontos de discretização temporal e 10 amostras para cada tempo, por 15 épocas. Por fim, amostramos usando a EDO de fluxo. Na figura (9) vemos o resultado e na figura (11), um exemplo de um processo de geração.

Resultados

Dígitos escritos a mão, geração condicional

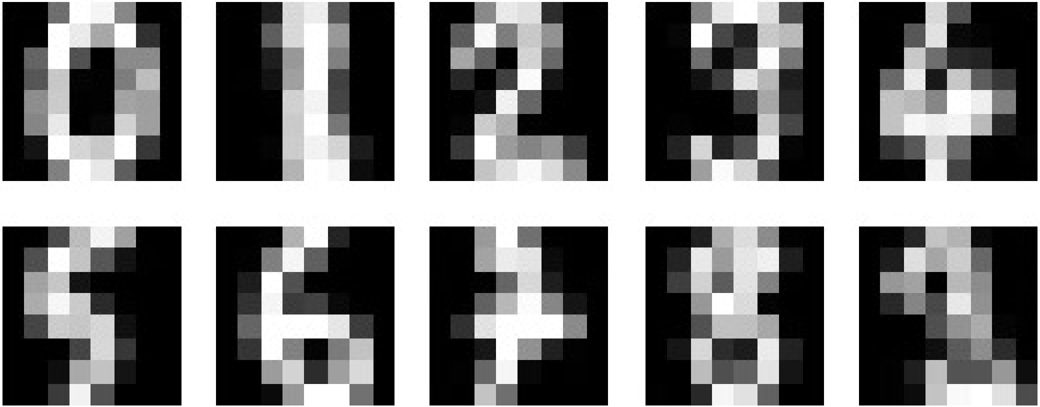


Figura: Geração condicionada a label.

Conclusão

- Reproduzimos as ideias básicas apresentadas no artigo
- O método se mostrou funcional em boa parte dos experimentos realizados, apesar da nossa implementação ter ficado lenta e ocupar muita memória.
- O artigo não comenta como ele lida com a discretização temporal no processo de treino e nem quantas simulações foram feitas para cada ponto, certamente, omitindo algum processo de amostragem que torna o código mais eficiente e lida melhor com a memória.

Referências I

FGV EMAp ESCOLA DE MATEMÁTICA APLICADA

[And82] Brian D.O. Anderson. "Reverse-time diffusion equation models". Em: *Stochastic Processes and their Applications* 12.3 (1982), pp. 313–326. ISSN: 0304-4149. DOI: https://doi.org/10.1016/0304-4149(82)90051-5. URL: https://www.sciencedirect.com/science/article/pii/0304414982900515.

[HP86] U. G. Haussmann e E. Pardoux. "Time Reversal of Diffusions". Em: *The Annals of Probability* 14.4 (1986), pp. 1188–1205. DOI: 10.1214/aop/1176992362. URL: https://doi.org/10.1214/aop/1176992362.

[RT96] Gareth O. Roberts e Richard L. Tweedie. "Exponential Convergence of Langevin Distributions and Their Discrete Approximations". Em: *Bernoulli* 2.4 (1996), pp. 341–363. ISSN: 13507265. URL: http://www.jstor.org/stable/3318418 (acesso em 15/06/2024).

Referências II

[Hyv05] Aapo Hyvärinen. "Estimation of non-normalized statistical models by score matching". Em: *Journal of Machine Learning Research* 6.Apr (2005), pp. 695–709.

[Bro+11] Steve Brooks et al. *Handbook of Markov Chain Monte Carlo*. Chapman e Hall/CRC, mai. de 2011. ISBN: 9780429138508. DOI: 10.1201/b10905. URL: http://dx.doi.org/10.1201/b10905.

[Goo+14] Ian Goodfellow et al. "Generative Adversarial Nets". Em: *NeurIPS* (2014). URL: https://arxiv.org/abs/1406.2661.

[HJA20] Jonathan Ho, Ajay Jain e Pieter Abbeel. "Denoising Diffusion Probabilistic Models". Em: *NeurIPS* (2020). URL: https://arxiv.org/abs/2006.11239.

[SE20a] Yang Song e Stefano Ermon. *Generative Modeling by Estimating Gradients of the Data Distribution*. 2020. arXiv: 1907.05600.

[SE20b] Yang Song e Stefano Ermon. *Improved Techniques for Training Score-Based Generative Models*. 2020. arXiv: 2006.09011.

[Son+21] Yang Song et al. *Score-Based Generative Modeling through Stochastic Differential Equations*. 2021. arXiv: 2011.13456 [cs.LG].

[Che+23] Sitan Chen et al. *Sampling is as easy as learning the score: theory for diffusion models with minimal data assumptions*. 2023. arXiv: 2209.11215 [cs.LG].