Generative IA

Score matching, Langevin dynamics, diffusion models

Arnaud Breloy arnaud.breloy@lecnam.net

15 décembre 2024

Conservatoire national des arts & métiers

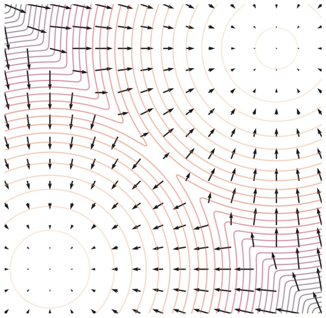
Score Function

Definition

The score function for a given distribution p is given by

$$\mathrm{s}(\mathrm{x}) = \nabla_{\mathrm{x}} \log \mathrm{p}(\mathrm{x})$$

- Give the direction of the distributions models
- Can be used for sampling



Score based models

Definition

Given a dataset $\{x_1, ..., x_N\} \in \mathcal{X}$ with $\forall i\,, x_i \sim p$, a score based model s_θ is learned to retrive the score of the distribution :

$$s_\theta(x) \simeq \nabla_x \log p(x)$$

$$\theta = \arg\min_\theta \mathbb{E}_{x\sim p}\left[||\nabla_x \log p(x) - s_\theta(x)||_2^2\right]$$

$\hookrightarrow$ Requires # dimension$(\mathcal{X})$ back-propagation
$\hookrightarrow$ Untractable in high-dimension

Link with energy-based models

Key feature : normalization constant not required

$$\begin{aligned}\nabla_x \log p_\theta(x) &= \nabla_x \log \frac{e^{-f_\theta(x)}}{Z_\theta} \\ &= \nabla_x \log e^{-f_\theta(x)} + \nabla_x \log(Z_\theta) \\ &= -\nabla_x f_\theta(x)\end{aligned}$$

Score based models

Definition

Given a dataset $\{x_1, ..., x_N\} \in \mathcal{X}$ with $\forall i\,, x_i \sim p$, a score based model s_θ is learned to retrive the score of the distribution :

$$s_\theta(x) \simeq \nabla_x \log p(x)$$

$$\theta = \arg\min_\theta \mathbb{E}_{x \sim p}\left[||\nabla_x \log p(x) - s_\theta(x)||_2^2\right]$$

$\hookrightarrow$ Requires # dimension$(\mathcal{X})$ back-propagation

$\hookrightarrow$ Untractable in high-dimension

Link with energy-based models

Key feature : normalization constant not required

$$\begin{aligned}\nabla_x \log p_\theta(x) &= \nabla_x \log \frac{e^{-f_\theta(x)}}{Z_\theta} \\ &= \nabla_x \log e^{-f_\theta(x)} + \nabla_x \log(Z_\theta) \\ &= -\nabla_x f_\theta(x)\end{aligned}$$

Score approximation techniques

Sliced-score matching

Project the score on random directions $\mathrm{v} \sim \mathrm{p_v}$ before minimazing the loss

$$\mathbb{E}_{\mathrm{v}\sim\mathrm{p_v}}\mathbb{E}_{\mathrm{x}\sim\mathrm{p}}\left(\mathrm{v}^{\mathrm{T}}\nabla_{\mathrm{x}}\log\mathrm{p}(\mathrm{x}) - \mathrm{v}^{\mathrm{T}}\mathrm{s}_\theta(\mathrm{x})\right)^2$$

Denoising score matching

Equivalence between denoising autoencoder (DAE) objective and score matching

- Matching the score of a noise perturbed distribution

$$\mathbb{E}_{\tilde{\mathrm{x}}\sim\mathrm{q}_\sigma(\tilde{\mathrm{x}})}||\nabla_{\tilde{\mathrm{x}}}\log\mathrm{q}_\sigma(\tilde{\mathrm{x}}) - \mathrm{s}_\theta(\tilde{\mathrm{x}})||_2^2$$

- Equivalent to denoising score matching $\Leftrightarrow$ DAE objective

$$\mathbb{E}_{\sim\mathrm{p}}\mathbb{E}_{\tilde{\mathrm{x}}\sim\mathrm{q}_\sigma(\tilde{\mathrm{x}}|\mathrm{x})}||\underbrace{\nabla_{\tilde{\mathrm{x}}}\log\mathrm{q}_\sigma(\tilde{\mathrm{x}}|\mathrm{x})}_{\text{Tractable}} - \mathrm{s}_\theta(\tilde{\mathrm{x}})||_2^2$$

Denoising Score Matching

Training

- Sample a batch of data $\{\mathrm{x}_1, ..., \mathrm{x}_\mathrm{n}\} \sim \mathrm{p}(\mathrm{x})$
- Sample noisy data $\{\mathrm{x}_1, ..., \mathrm{x}_\mathrm{n}\} \sim \mathrm{q}_\sigma(\tilde{\mathrm{x}}|\mathrm{x})$
- Estimate the denoising score loss :

$$\frac{1}{\mathrm{n}}\sum_{\mathrm{i}=1}^{\mathrm{n}} ||\mathrm{s}_\theta(\tilde{\mathrm{x}}_\mathrm{i}) - \nabla_{\tilde{\mathrm{x}}} \log \mathrm{q}_\sigma(\tilde{\mathrm{x}}_\mathrm{i}|\mathrm{x}_\mathrm{i})||_2^2$$

- In the case of additive Gaussian noise :

$$\frac{1}{\mathrm{n}}\sum_{\mathrm{i}=1}^{\mathrm{n}} ||\mathrm{s}_\theta(\tilde{\mathrm{x}}_\mathrm{i}) - \frac{\mathrm{x}_\mathrm{i} - \tilde{\mathrm{x}}_\mathrm{i}}{\sigma^2}||_2^2$$

- Compute gradient descent
- σ must be small

Langevin Dynamic Sampling

Stochastic sampling process

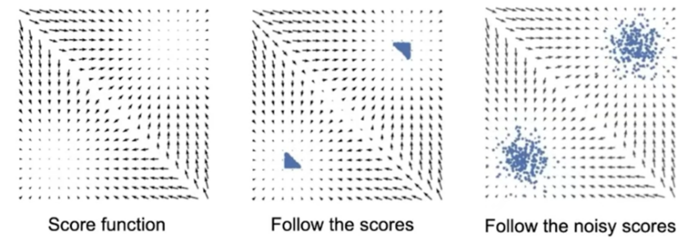
- $x_0 \sim \pi(x) \leftarrow$ random initialization
- repeat for $t \in 1, 2, ..., T$

$$x_t \leftarrow x_t + \frac{\epsilon}{2} \nabla_x \log p(x_{t-1}) + \epsilon z_t$$

with $\epsilon << 0$ and $T \to \infty$.

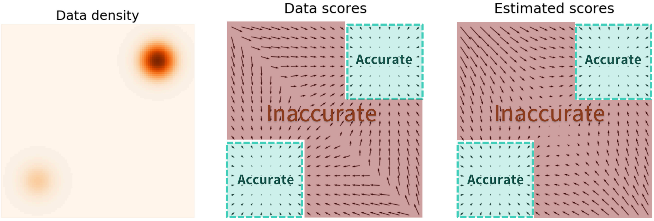
- Using the score estimation :

$$x_t \leftarrow x_t + \frac{\epsilon}{2} s_\theta(x) + \epsilon z_t$$



Sampling in Low Density Regions

Problem Estimated scores are only accurate in high density regions.



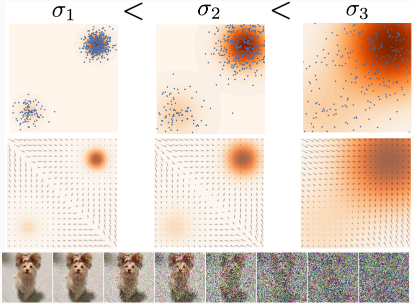
Noise Conditional Score Models

Score matching with multiple noise scale

Adding noise to the data spread their distribution filling low density regions. The score is then learned on noisy data perturbed with different noise scale :

$$\mathcal{L}(\theta) = \sum_{l=1}^{L} \lambda(i) \mathbb{E}_{x \sim p_{\sigma_i}} ||\nabla_x \log p_{\sigma_i}(x) - s_\theta(x, \sigma_i)||_2^2,$$

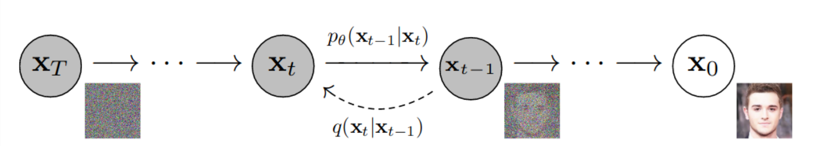
with L the number of noise scales.



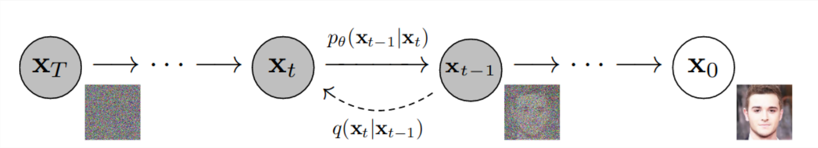
Diffusion Models

Different formulation equivalent to score based approaches

- Noise is progressively added to the image
- The model learn a denosier able to retrieve $\mathrm{x_{t-1}}$ from $\mathrm{x_t}$
- Sampling is then performed by following T denoising step starting from $\mathrm{x_T} \sim \mathcal{N}(0, \sigma^2 \mathrm{I})$



Forward and Backward Processes



Forward Process

Gradually adds noise to the image. It is deifned as a Markow chain :

- $\mathrm{q}(\mathrm{x}_1, ..., \mathrm{x}_\mathrm{T}|\mathrm{x}_0) = \prod_{\mathrm{t}=1}^{\mathrm{T}} \mathrm{q}_\theta(\mathrm{x}_\mathrm{t}|\mathrm{x}_{\mathrm{t}-1})$
- $\mathrm{q}(\mathrm{x}_\mathrm{t}|\mathrm{x}_{\mathrm{t}-1}) = \mathcal{N}(\sqrt{1 - \beta_\mathrm{t}\mathrm{x}_{\mathrm{t}-1}}, \beta_\mathrm{t}\mathrm{I})$

Backward Process

Gradually substract the noise. It is deifned as a Markow chain :

- $\mathrm{p}(\mathrm{x}_\mathrm{T}) = \mathcal{N}(0, \sigma^1\mathrm{I})$
- $\mathrm{p}_\theta(\mathrm{x}_0, ..., \mathrm{x}_\mathrm{T}) = \mathrm{p}(\mathrm{x}_\mathrm{T}) \prod_{\mathrm{t}=1}^{\mathrm{T}} \mathrm{p}_\theta(\mathrm{x}_{\mathrm{t}-1}|\mathrm{x}_\mathrm{t})$
- $\mathrm{p}_\theta(\mathrm{x}_{\mathrm{t}-1}|\mathrm{x}_\mathrm{t}) = \mathcal{N}(\boldsymbol{\mu}_\theta, \boldsymbol{\Sigma}_\theta(\mathrm{x}_\mathrm{t}, \mathrm{t}))$

Evidenctial Lower Bound

Loss

Training is performed by minmizing the following bound on the negative likelihood :

$$\mathbb{E}[-\log p_\theta(x_0)] \leq \mathbb{E}_q\left[-\log \frac{p_\theta(x_0, ..., x_T)}{q(x_1, ..., x_T|x_0)}\right]$$
$$= \mathbb{E}_q\Bigg[\text{constant} + \sum_{t>1} \mathrm{KL}(q(x_{t-1}|x_t, x_0)||p_\theta(x_{t-1}|x_t))$$
$$- \log p_\theta(x_0|x_1)\Bigg],$$

with $q(x_{t-1}|x_t, x_0) = \mathcal{N}(\tilde{\boldsymbol{\mu}}_t(x_t, x_0), \tilde{\beta}_t I)$

Training

Training is performed by sampling a batch of time-steps and minimizing the sum with respect to it. The Kullback-Leiber divergences can be written as

$$\mathrm{KL}(q(x_{t-1}|x_t, x_0)||p_\theta(x_{t-1}|x_t)) = \frac{1}{2\sigma_t^2}||\tilde{\boldsymbol{\mu}}_t(x_t, x_0) - \boldsymbol{\mu}_\theta(x_t, t)||_2^2$$

which is a denoising objectif $\rightarrow$ analagous to denoising score matching

Results

Score based and Diffusion models are now state-of-the-art in image generation

Some ressources

https://lilianweng.github.io/posts/2021-07-11-diffusion-models/

https://yang-song.net/blog/2021/score/

https://dl.heeere.com/conditional-flow-matching/blog/conditional-flow-matching/