Diffusion Models: DALL-E

Deep Learning and Neural Networks: Advanced Topics

Fabio Brau

March 1, 2023

Scuola Superiore Sant'Anna, Pisa.

Introduction

Diffusion Models

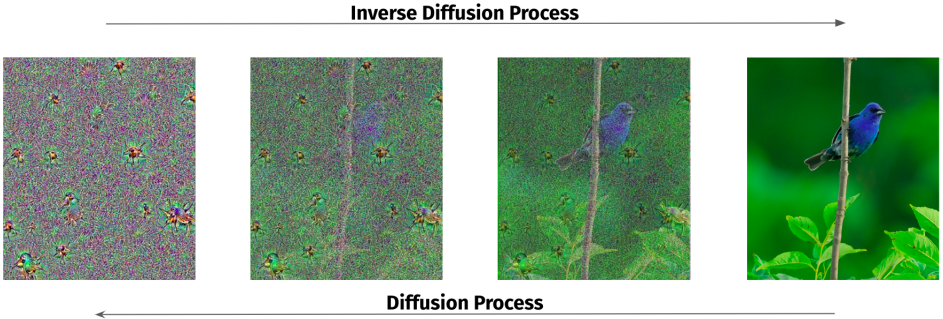
Broader Impacts

Introduction

Diffusion Models

Overview

Diffusion models are generative models that aim at denoising data



Timeline

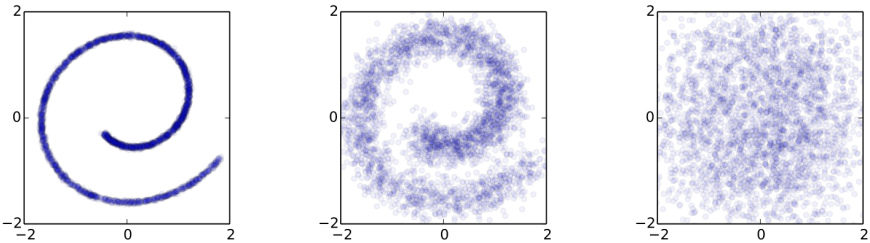
2015) *...Non-equilibrium Thermodynamics*. Sohl-Dickstein et al. ICML

2020) *Denoising Diffusion Probabilistic Models*. Ho et al. NeurIPS.

2021) *Score-Based Generative Modeling Through SDE*. Song et al. ICLR.

Deep Unsupervised Learning using Non-Equilibrium Thermodynamics

$t = 0$ $t = \frac{T}{2}$ $t = T$



Diffusion process as a **Markov Chain** with **Continuous State Space** and **Discrete Time**.[1]

[1] Sohl-Dickstein et al., "Deep Unsupervised Learning using Nonequilibrium Thermodynamics".

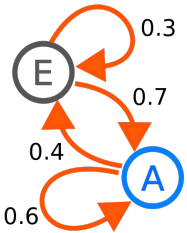
Reminder: Markov Chains with Discrete Time

Informal Definition

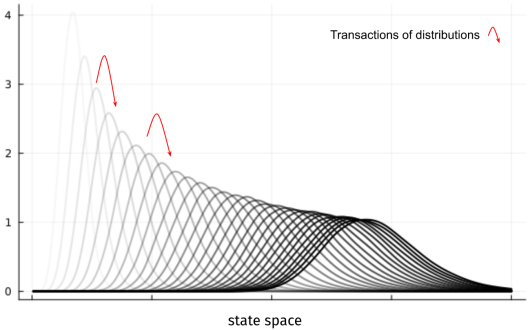
A sequence of random variables $\mathbf{x}^{(0)}, \mathbf{x}^{(1)}, \cdots, \mathbf{x}^{(t)}, \cdots$, such that:

- $\mathbf{x}^{(t)} \in S$, where S **State Space**
- The future $\mathbf{x}^{(t+1)}$ depends on the present $\mathbf{x}^{(t)}$ but not on the past $\mathbf{x}^{(t-1)}$

Discrete State Space S



Continuous State Space S



Reminder: MCDT with Discrete State Space

Definition
A sequence $\left\{\mathbf{x}^{(t)}\right\}_{t\in\mathbb{N}} \subseteq S$, a matrix $P = \left(p_{ij}\right)$.

- Discrete state space: $S = \{s_0, \cdots, s_n, \cdots\}$
- Markov Property: $\mathbf{x}^{(t+1)}$ not dep. $\mathbf{x}^{(0)}, \cdots, \mathbf{x}^{(t-1)}$.
- Transaction Matrix: $\mathbb{P}\left(\mathbf{x}^{(t+1)} = s_j | \mathbf{x}^{(t)} = s_i\right) = p_{ij}$

Reminder: MCDT with Discrete State Space

Definition

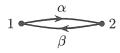
A sequence $\left\{\mathbf{x}^{(t)}\right\}_{t\in\mathbb{N}} \subseteq S$, a matrix $P = \left(p_{ij}\right)$.

- Discrete state space: $S = \{s_0, \cdots, s_n, \cdots\}$
- Markov Property: $\mathbf{x}^{(t+1)}$ not dep. $\mathbf{x}^{(0)}, \cdots, \mathbf{x}^{(t-1)}$.
- Transaction Matrix: $\mathbb{P}\left(\mathbf{x}^{(t+1)} = s_j | \mathbf{x}^{(t)} = s_i\right) = p_{ij}$

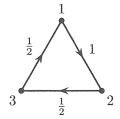
P is a stochastic matrix!

$$\forall i, \quad \sum_{j\in\mathbb{N}} p_{ij} = 1$$

$$P = \begin{pmatrix} 1-\alpha & \alpha \\ \beta & 1-\beta \end{pmatrix}$$



$$P = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 1/2 & 1/2 \\ 1/2 & 0 & 1/2 \end{pmatrix}$$



Reminder: DTMC with Continuous State Space

Let assume $\mathbf{x}, \mathbf{y} \in S$ where S continuous state space (e.g. $S = \mathbb{R}^d$).

Joint Distribution $p(\mathbf{x}, \mathbf{y})$

$$\mathbb{P}\left(\mathbf{x} \in A \,|\, \mathbf{y} \in B\right) = \int_A \int_B p\left(\mathbf{x}, \mathbf{y}\right)\, d\mathbf{x}\, d\mathbf{y}$$

Transactional Kernel $p(\mathbf{x} \,|\, \mathbf{y})$

$$p\left(\mathbf{x}, \mathbf{y}\right) = p(\mathbf{x} \,|\, \mathbf{y})\, p\left(\mathbf{y}\right)$$

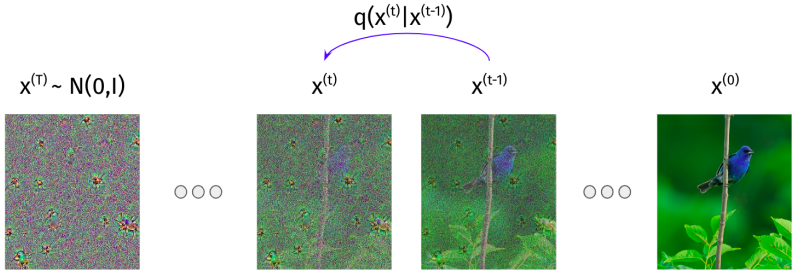
Marginal Distribution $p(\mathbf{x})$

$$p\left(\mathbf{x}\right) = \int_S p\left(\mathbf{x}, \mathbf{y}\right)\, d\mathbf{y} = \int_S p(\mathbf{x} \,|\, \mathbf{y})\, p\left(\mathbf{y}\right)\, d\mathbf{y}$$

Forward Diffusion Process

"Adding noise to data..."

- Data Distribution: $\mathbf{x}^{(0)} \sim q$
- Transaction Kernel: $q\left(\mathbf{x}^{(t)} \mid \mathbf{x}^{(t-1)}\right) = \mathcal{N}\left(\mathbf{x}^{(t)}; \sqrt{1-\beta_t}\mathbf{x}^{(t-1)}; \beta_t I\right)$
- Variance Scheduler: $\beta_1, \cdots, \beta_T \in (0, 1]$



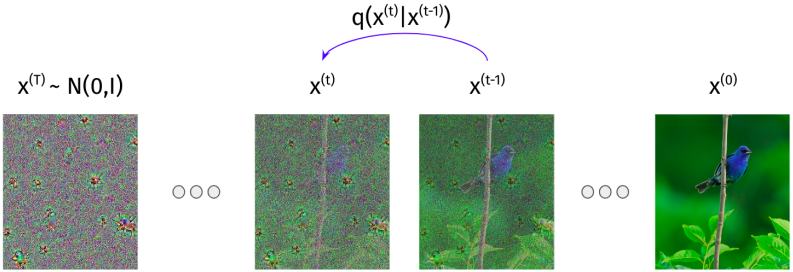
Forward Diffusion Process

"Adding noise to data..."

- Data Distribution: $\mathbf{x}^{(0)} \sim q$
- Transaction Kernel: $q\left(\mathbf{x}^{(t)} \mid \mathbf{x}^{(t-1)}\right) = \mathcal{N}\left(\mathbf{x}^{(t)}; \sqrt{1-\beta_t}\mathbf{x}^{(t-1)}; \beta_t I\right)$
- Variance Scheduler: $\beta_1, \cdots, \beta_T \in (0, 1]$

Not Analytic!!

$\beta_T = 1$



Forward Diffusion Process: Explicit Representation

$$\mathbf{x}^{(t)} = \sqrt{1-\beta_t}\,\mathbf{x}^{(t-1)} + \sqrt{\beta_t}\,\boldsymbol{\varepsilon}_t, \quad \boldsymbol{\varepsilon}_t \sim \mathcal{N}(0, I)$$

Observation: Many small noisy steps ≈ Large Noisy step

$$\mathbf{x}^{(t)} = \sqrt{1-\alpha_t}\,\mathbf{x}^{(0)} + \sqrt{\alpha_t}\boldsymbol{\varepsilon}, \quad \boldsymbol{\varepsilon} \sim \mathcal{N}(0, I)$$

where

$$\alpha_t = 1 - \prod_{i=0}^{t}(1-\beta_i)$$

Forward Diffusion Process: Distribution Representation

Markov property allows breaking up distributional Representation...

$$q(\mathbf{x}^{(0)}, \ldots, \mathbf{x}^{(T)}) = q\left(\mathbf{x}^{(T)} \mid \mathbf{x}^{(0)}, \ldots, \mathbf{x}^{(T-1)}\right) q\left(\mathbf{x}^{(0)}, \ldots, \mathbf{x}^{(T-1)}\right)$$

Forward Diffusion Process: Distribution Representation

Markov property allows breaking up distributional Representation...

$$
\begin{aligned}
q(\mathbf{x}^{(0)}, \ldots, \mathbf{x}^{(T)}) &= q\left(\mathbf{x}^{(T)} \mid \mathbf{x}^{(0)}, \ldots, \mathbf{x}^{(T-1)}\right) q\left(\mathbf{x}^{(0)}, \ldots, \mathbf{x}^{(T-1)}\right) \\
&= q\left(\mathbf{x}^{(T)} \mid \mathbf{x}^{(T-1)}\right) q\left(\mathbf{x}^{(0)}, \ldots, \mathbf{x}^{(T-1)}\right) \qquad (1) \\
&\vdots
\end{aligned}
$$

Forward Diffusion Process: Distribution Representation

Markov property allows breaking up distributional Representation...

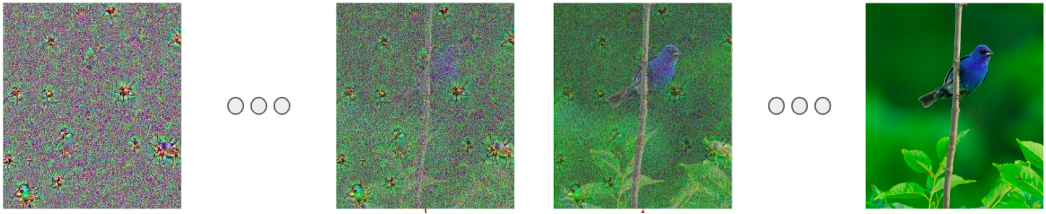
$$
\begin{aligned}
q(\mathbf{x}^{(0)}, \ldots, \mathbf{x}^{(T)}) &= q\left(\mathbf{x}^{(T)} \mid \mathbf{x}^{(0)}, \ldots, \mathbf{x}^{(T-1)}\right) q\left(\mathbf{x}^{(0)}, \ldots, \mathbf{x}^{(T-1)}\right) \\
&= q\left(\mathbf{x}^{(T)} \mid \mathbf{x}^{(T-1)}\right) q\left(\mathbf{x}^{(0)}, \ldots, \mathbf{x}^{(T-1)}\right) \qquad (1) \\
&\vdots
\end{aligned}
$$

Distributional Representation

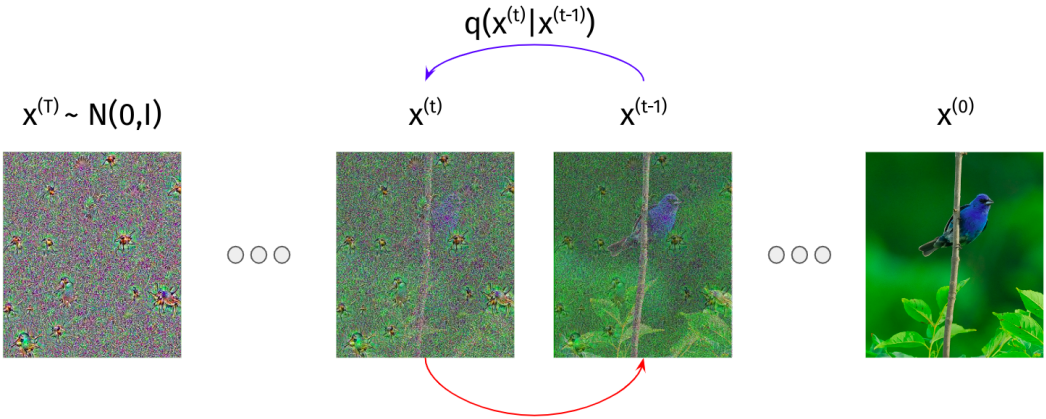
$$
q(\mathbf{x}^{(0)}, \ldots, \mathbf{x}^{(T)}) = q(\mathbf{x}^{(0)}) \prod_{t=1}^{T} q\left(\mathbf{x}^{(t)} \mid \mathbf{x}^{(t-1)}\right)
$$

Reverse Diffusion Process

$x^{(T)} \sim N(0,I)$ $x^{(t)}$ $x^{(t-1)}$ $x^{(0)}$



Reverse Diffusion Process



Reverse Diffusion Process

Forward Diffusion Process

$q(\mathbf{x}^{(0)})$ Data Distribution

$$q(\mathbf{x}^{(0\dots T)}) = q(\mathbf{x}^{(0)}) \prod_{t=1}^{T} q\left(\mathbf{x}^{(t)} \mid \mathbf{x}^{(t-1)}\right)$$

Reverse Diffusion Process

$$q(x^{(T)}) = \mathcal{N}(0, I)$$

$$q(\mathbf{x}^{(0\dots T)}) = q(\mathbf{x}^{(T)}) \prod_{t=1}^{T} q\left(\mathbf{x}^{(t-1)} \mid \mathbf{x}^{(t)}\right)$$

[2] Sohl-Dickstein et al., “Deep Unsupervised Learning using Nonequilibrium Thermodynamics”.

Reverse Diffusion Process

Forward Diffusion Process

$q(\mathbf{x}^{(0)})$ Data Distribution

$$q(\mathbf{x}^{(0\dots T)}) = q(\mathbf{x}^{(0)}) \prod_{t=1}^{T} q\left(\mathbf{x}^{(t)} \mid \mathbf{x}^{(t-1)}\right)$$

Reverse Diffusion Process

$$q(x^{(T)}) = \mathcal{N}(0, I)$$

$$q(\mathbf{x}^{(0\dots T)}) = q(\mathbf{x}^{(T)}) \prod_{t=1}^{T} q\left(\mathbf{x}^{(t-1)} \mid \mathbf{x}^{(t)}\right)$$

Theorem. Reverse of Gaussian DP is $\approx$ Gaussian DP[2]

If $|\beta_i - \beta_{i+1}| \approx 0$, i.e. diffusion **slow enough**, then

$$q(\mathbf{x}^{(t-1)} \mid \mathbf{x}^{(t)}) \approx \mathcal{N}\left(\mathbf{x}^{(t-1)}; \boldsymbol{\mu}_\theta\left(\mathbf{x}^{(t)}, t\right), \boldsymbol{\Sigma}_\theta\left(\mathbf{x}^{(t)}, t\right)\right)$$

[2] Sohl-Dickstein et al., "Deep Unsupervised Learning using Nonequilibrium Thermodynamics".

Reverse Diffusion Process

Forward Diffusion Process

$q(\mathbf{x}^{(0)})$ Data Distribution

$$q(\mathbf{x}^{(0\dots T)}) = q(\mathbf{x}^{(0)}) \prod_{t=1}^{T} q\left(\mathbf{x}^{(t)} \mid \mathbf{x}^{(t-1)}\right)$$

Reverse Diffusion Process

$$q(x^{(T)}) = \mathcal{N}(0, I)$$

$$q(\mathbf{x}^{(0\dots T)}) = q(\mathbf{x}^{(T)}) \prod_{t=1}^{T} q\left(\mathbf{x}^{(t-1)} \mid \mathbf{x}^{(t)}\right)$$

Theorem. Reverse of Gaussian DP is $\approx$ Gaussian DP[2]

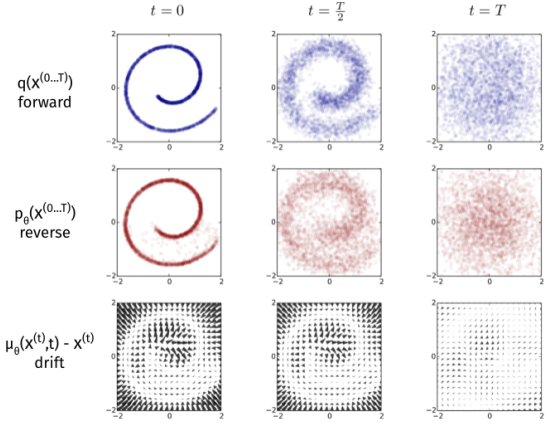
If $|\beta_i - \beta_{i+1}| \approx 0$, i.e. diffusion **slow enough**, then

$$q(\mathbf{x}^{(t-1)} \mid \mathbf{x}^{(t)}) \approx \mathcal{N}\left(\mathbf{x}^{(t-1)}; \boldsymbol{\mu}_\theta\left(\mathbf{x}^{(t)}, t\right), \boldsymbol{\Sigma}_\theta\left(\mathbf{x}^{(t)}, t\right)\right)$$

Mean $\boldsymbol{\mu}_\theta$ and covariance $\boldsymbol{\Sigma}_\theta$ have to be learned!!

[2] Sohl-Dickstein et al., "Deep Unsupervised Learning using Nonequilibrium Thermodynamics".

Visualization of Diffusion Process: 2D dimensional case



3

[3] Sohl-Dickstein et al., “Deep Unsupervised Learning using Nonequilibrium Thermodynamics”.

Training of $\boldsymbol{\mu}_\theta$ and $\boldsymbol{\Sigma}_\theta$

Aim

Seach for the best parameters θ

$$q(\mathbf{x}^{(0)}) \approx p_\theta(\mathbf{x}^{(0)})$$

where $\mathbf{x}^{(0)} \cdots, \mathbf{x}^{(T)}$ diffusion process

Estimated Reverse Process

$$p_\theta(\mathbf{x}^{(T)}) = \mathcal{N}\left(\mathbf{x}^{(T)}; 0, I\right)$$
$$p_\theta(\cdot \mid \mathbf{x}^{(t)}) = \mathcal{N}\left(\boldsymbol{\mu}_\theta\left(\mathbf{x}^{(t)}, t\right), \boldsymbol{\Sigma}_\theta\left(\mathbf{x}^{(t)}, t\right)\right)$$

Training of $\boldsymbol{\mu}_\theta$ and $\boldsymbol{\Sigma}_\theta$

Aim

Seach for the best parameters θ

$$q(\mathbf{x}^{(0)}) \approx p_\theta(\mathbf{x}^{(0)})$$

where $\mathbf{x}^{(0)} \cdots, \mathbf{x}^{(T)}$ diffusion process

Estimated Reverse Process

$$p_\theta(\mathbf{x}^{(T)}) = \mathcal{N}\left(\mathbf{x}^{(T)}; 0, I\right)$$
$$p_\theta(\cdot \mid \mathbf{x}^{(t)}) = \mathcal{N}\left(\boldsymbol{\mu}_\theta\left(\mathbf{x}^{(t)}, t\right), \boldsymbol{\Sigma}_\theta\left(\mathbf{x}^{(t)}, t\right)\right)$$

Method

Minimize the *Kullback–Leibler* Divergence

$$D_{KL}(q \,||\, p_\theta) := \int q(\mathbf{x}^{(0)}) \log\left(\frac{q(\mathbf{x}^{(0)})}{p_\theta(\mathbf{x}^{(0)})}\right) d\mathbf{x}^{(0)}$$

Training of $\boldsymbol{\mu}_\theta$ and $\boldsymbol{\Sigma}_\theta$

Aim

Seach for the best parameters θ

$$q(\mathbf{x}^{(0)}) \approx p_\theta(\mathbf{x}^{(0)})$$

where $\mathbf{x}^{(0)} \cdots, \mathbf{x}^{(T)}$ diffusion process

Estimated Reverse Process

$$p_\theta(\mathbf{x}^{(T)}) = \mathcal{N}\left(\mathbf{x}^{(T)}; 0, I\right)$$
$$p_\theta(\cdot \mid \mathbf{x}^{(t)}) = \mathcal{N}\left(\boldsymbol{\mu}_\theta\left(\mathbf{x}^{(t)}, t\right), \boldsymbol{\Sigma}_\theta\left(\mathbf{x}^{(t)}, t\right)\right)$$

Method

Minimize the *Kullback–Leibler* Divergence

$$D_{KL}(q \,||\, p_\theta) := \int q(\mathbf{x}^{(0)}) \log\left(\frac{q(\mathbf{x}^{(0)})}{p_\theta(\mathbf{x}^{(0)})}\right) d\mathbf{x}^{(0)}$$

Easy??

Training of $\boldsymbol{\mu}_\theta$ and $\boldsymbol{\Sigma}_\theta$

Aim

Seach for the best parameters θ

$$q(\mathbf{x}^{(0)}) \approx p_\theta(\mathbf{x}^{(0)})$$

where $\mathbf{x}^{(0)} \cdots, \mathbf{x}^{(T)}$ diffusion process

Estimated Reverse Process

$$p_\theta(\mathbf{x}^{(T)}) = \mathcal{N}\left(\mathbf{x}^{(T)}; 0, I\right)$$
$$p_\theta(\cdot \,|\, \mathbf{x}^{(t)}) = \mathcal{N}\left(\boldsymbol{\mu}_\theta\left(\mathbf{x}^{(t)}, t\right), \boldsymbol{\Sigma}_\theta\left(\mathbf{x}^{(t)}, t\right)\right)$$

Method

Minimize the *Kullback–Leibler* Divergence

$$D_{KL}(q \,||\, p_\theta) := \int q(\mathbf{x}^{(0)}) \log\left(\frac{q(\mathbf{x}^{(0)})}{p_\theta(\mathbf{x}^{(0)})}\right) d\mathbf{x}^{(0)}$$

Easy??

No. $q(\mathbf{x}^{(0)})$ is analytically intractable!!

Training of $\boldsymbol{\mu}_\theta$ and $\boldsymbol{\Sigma}_\theta$

Aim: Deduce a tractable loss function

$$D_{KL}(q \,||\, p_\theta) := \int q(\mathbf{x}^{(0)}) \log \left(\frac{q(\mathbf{x}^{(0)})}{p_\theta(\mathbf{x}^{(0)})} \right) d\mathbf{x}^{(0)}$$

Training of $\boldsymbol{\mu}_\theta$ and $\boldsymbol{\Sigma}_\theta$

Aim: Deduce a tractable loss function

$$D_{KL}(q \,||\, p_\theta) := \int q(\mathbf{x}^{(0)}) \log \left(\frac{q(\mathbf{x}^{(0)})}{p_\theta(\mathbf{x}^{(0)})} \right) d\mathbf{x}^{(0)}$$

Semplification I: Minimize the Cross Entropy

$$D_{KL}\left(q(\mathbf{x}^{(0)}) || p_\theta(\mathbf{x}^{)0)}\right) = \int q(\mathbf{x}^{(0)}) \log(q(\mathbf{x}^{(0)})\, d\mathbf{x}^{(0)} + \int -q(\mathbf{x}^{(0)}) \log(p_\theta(\mathbf{x}^{(0)}))\, d\mathbf{x}^{(0)}$$

Training of μ_θ and Σ_θ

Aim: Deduce a tractable loss function

$$D_{KL}(q \,||\, p_\theta) := \int q(\mathbf{x}^{(0)}) \log\left(\frac{q(\mathbf{x}^{(0)})}{p_\theta(\mathbf{x}^{(0)})}\right) d\mathbf{x}^{(0)}$$

Semplification I: Minimize the Cross Entropy

$$D_{KL}\left(q(\mathbf{x}^{(0)})||p_\theta(\mathbf{x}^{)0)}\right) = \underbrace{\int q(\mathbf{x}^{(0)}) \log(q(\mathbf{x}^{(0)})\, d\mathbf{x}^{(0)}}_{-\mathbb{H}(q(\mathbf{x}^{(0)})} + \underbrace{\int -q(\mathbf{x}^{(0)}) \log(p_\theta(\mathbf{x}^{(0)}))\, d\mathbf{x}^{(0)}}_{L_{CE}(p_\theta)}$$

Broader Impacts

CLIP Model

"We also found discrepancies across gender and race for people categorized into the 'crime' and 'non-human' categories..."[4]

[4] Radford et al., "Learning Transferable Visual Models From Natural Language Supervision".

Thanks for the attention

Fabio Brau

Scuola Superiore Sant'Anna, Pisa

fabio.brau@santannapisa.it

retis.santannapisa.it/~f.brau

linkedin.com/in/fabio-brau

TELECOMMUNICATIONS, COMPUTER ENGINEERING, AND PHOTONICS INSTITUTE

Sant'Anna School of Advanced Studies – Pisa

ECCELLENZA MIUR 2018-2022

ROBOTICS & AI

Sant'Anna Scuola Universitaria Superiore Pisa

retis Real-Time Systems Laboratory

Proof Details

Proof of Explicit Representation of Forward Diffusion Process

Let us proceeding by induction by assuming $\mathbf{x}^{(t)} = \sqrt{1-\alpha_t}\,\mathbf{x}^{(0)} + \sqrt{\alpha_t}\,\boldsymbol{\varepsilon}$ where $\boldsymbol{\varepsilon} \sim \mathcal{N}(0, I)$ and where $\alpha_t = 1 - \prod_{i=0}^{t}(1-\beta_i)$.

$$
\begin{aligned}
\mathbf{x}^{(t+1)} &= \sqrt{1-\beta_{t+1}}\,\mathbf{x}^{(t)} + \sqrt{\beta_{t+1}}\,\boldsymbol{\varepsilon}_{t+1} \\
&= \sqrt{1-\beta_{t+1}}\left(\sqrt{1-\alpha_t}\,\mathbf{x}^{(0)} + \sqrt{\alpha_t}\,\boldsymbol{\varepsilon}\right) + \sqrt{\beta_{t+1}}\,\boldsymbol{\varepsilon}_{t+1} \\
&= \sqrt{\left(\prod_{i=0}^{t+1}(1-\beta_i)\right)}\mathbf{x}^{(0)} + \sqrt{(1-\beta_{t+1})\alpha_t + \beta_{t+1}}\,\tilde{\boldsymbol{\varepsilon}}
\end{aligned}
\tag{2}
$$

where the last term of the summation is obtained by observing that, since $\sqrt{(1-\beta_{t+1})\alpha_t}\,\boldsymbol{\varepsilon}$ and $\sqrt{\beta_{t+1}}\,\boldsymbol{\varepsilon}_{t+1}$ are independent, then the variance of their sum (that still has a gaussian distribution) is given by $(1-\beta_{t+1})\alpha_t + \beta_{t+1}$.