Diversity in deep generative models and generative AI

Gabriel Turinici

CEREMADE
Université Paris Dauphine - PSL
Paris, France

LOD 2023,
The 9th International Conference on
Machine Learning, Optimization, and Data Science,

LOD 2023, UK, Sept. 22 – 26, 2023

Executive summary

Deep decoder based generative AI algorithms such as GAN, VAE, Transformer create objects similar to ones in the dataset it was trained on.

However the diversity of these objects is not always optimal (e.g., too similar objects).

Based on measure quantization techniques, and Huber-energy kernel based statistical distances we give a procedure to draw samples with improved diversity.

The procedure is tested with satisfactory results on a standard AI dataset (MNIST).

Outline

1. Introduction : diversity sampling in generative AI
   - Background decoder based generative AI
2. Mathematical framework
   - Problem formulation
   - Statistical distances and conditionally negative kernels
3. Algorithms and numerical results
   - Two algorithms
   - Numerical results

Introduction and motivation

• we are concerned with generative AI algorithms (e.g. GAN, VAE, Transformer) that create new objects (e.g., images) based on some dataset of examples.

• We want to enforce diversity in this creation, like human painters do not paint twice same painting, have "periods", same for writers, musicians, ...

Famous painters have "periods" : here Pablo Picasso's rose, blue, cubism, surrealism periods.

(from https://mymodernmet.com/pablo-picasso-periods/)

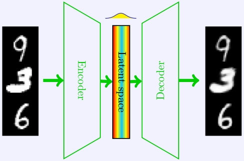
Introduction and motivation: mathematical framework

- Given : empirical dataset $\mu_e = \frac{1}{M}\sum_{\ell=1}^{M} \delta_{z_\ell}$ sampled from unknown distribution μ ($z_\ell \sim \mu$), $z_\ell \in \mathbb{R}^N$.
- Goal: construct samples as μ

- Technical solution: (GAN, VAE, Transformer): find a latent space $\mathbb{R}^L$ and a Decoder (generator) map $D : \mathbb{R}^L \to \mathbb{R}^N$ such that $D(\mathcal{N}(0_L, \mathrm{Id}_L)) \sim \mu$ (here $\mathcal{N}(0_L, \mathrm{Id}_L)$ is the std. normal distribution in L variables).

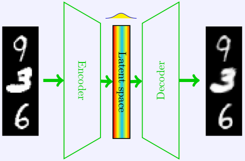
Standard generation step: sample X_j from $\mathcal{N}(0_L, \mathrm{Id}_L)$ and apply $D(\cdot)$.

Variational Autoencoder (VAE) structure : the input dataset μ_e is used to train the encoder $E(\cdot)$ and decoder $D(\cdot)$ networks such that $E(\mu_e) \sim \mathcal{N}(0_L, \mathrm{Id}_L)$ and $D \circ E \sim Id$ and obtain a reference distribution (yellow) on the latent space.

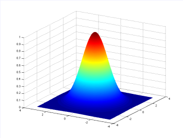


Latent space representation of the empirical dataset for VAE $\mu_L := E(\mu_e)$

Variational Autoencoder (VAE) structure : the input dataset μ_e is used to train the encoder $E(\cdot)$ and decoder $D(\cdot)$ networks such that $E(\mu_e) \sim \mathcal{N}(0_L, \mathrm{Id}_L)$ and $D \circ E \sim Id$ and obtain a reference distribution (yellow) on the latent space.



Problem: sampling from 2D Gaussian distribution results in most samples in the red part; low diversity; example for a GAN / VAE, sampling is done from the latent distribution with replacement.



Idea

sample X_j simultaneously to reduce redundancies and ensure that their distribution matches the true distribution on the latent space (μ_L or $\mathcal{N}(0_L, \mathrm{Id}_L)$).

Outline

Technical how-to

Idea

sample X_j simultaneously to reduce redundancies and ensure that their distribution matches the true distribution on the latent space (μ_L or $\mathcal{N}(0_L, \mathrm{Id}_L)$).

Mathematical formulation

Find $X = (X_j)_{j=1}^J \in \mathbb{R}^{L \times J}$ ($j = 1, ..., J$) that minimizes the distance from $\delta_X := \frac{\sum_{j=1}^J \delta_{X_j}}{J}$ to the target measure μ_L or $\mathcal{N}(0_L, \mathrm{Id}_L)$.

$$(Semp) : \min_X X \mapsto dist(\delta_X, \mu_L)^2 \tag{1}$$

$$(Sg) : \min_X X \mapsto dist(\delta_X, \mathcal{N}(0_L, \mathrm{Id}_L))^2 \tag{2}$$

Technical how-to

Questions:

- how to compute $X \mapsto dist(\delta_X, \eta)^2$
- how to minimize it ?

Distance: use a conditionally negative definite kernel h :

$$d(\mu_1, \mu_2)^2 = -\frac{1}{2} \int_{\mathbb{R}^L} \int_{\mathbb{R}^L} h(x-y)(\mu_1 - \mu_2)(dx)(\mu_1 - \mu_2)(dy). \tag{3}$$

Discrete version :

$$d\left(\frac{1}{J} \sum_{j=1}^{J} \delta_{X_j}, \frac{1}{B} \sum_{b=1}^{B} \delta_{z_b} \right)^2 = \frac{\sum_{j,b=1}^{J,B} h(X_j - z_b)}{JB} - \frac{\sum_{j,j'=1}^{J} h(X_j - X_{j'})}{2J^2} - \frac{\sum_{b,b'=1}^{B} h(z_b - z_{b'})}{2B^2}, \tag{4}$$

Question: what function h to choose ?

Statistical distances: conditionally negative kernels

Definition (conditional negative definite)

A kernel $h(\cdot,\cdot)$ is said to be conditionally negative definite if for any $I \in \mathbb{N}$, $p_1, ..., p_I$ with $\sum p_i = 0$ and any $x_1, ..., x_I$: $\sum_{i,j} p_i p_j h(x_i, x_j) \leq 0$.

Theorem ("Gini difference" Gini 1912; "energy distance" Szekelly 1985, '02; "maximum mean discrepancy" Gretton '07, G.T. '21 [7])

The kernel $h(x) = |x|$ is conditionally negative definite.

Caution : (sub) gradient $\nabla|x| = x/|x|^2$ unstable in $x = 0$.

Theorem (Schoenberg 1938 [3], Micchelli 1984 [2], GT 2021 [5])

For any $a \geq 0$, $\alpha \in]0,1[$, $(a^2 + |x|^2)^\alpha - a^{2\alpha}$ and $\|x^2\|/(a^2 + |x|^2)^\alpha$ are conditionally negative definite (explicit Gaussian mixtures).

In particular this is true for our choice $h(x) = \sqrt{a^2 + \|x\|^2} - a$.

Outline

Stochastic minimization algorithm

Algorithm (Sg) : Diversity sampling from the ideal target $\mathcal{N}(0_L, \mathrm{Id}_L)$

Inputs : batch size B, parameter $a = 10^{-6}$.

Outputs : quantized points X_j, $j = 1..., J$.

- initialize points $X = (X_j)_{j=1}^J$ sampled i.i.d from $\mathcal{N}(0_L, \mathrm{Id}_L)$;
- while(max iteration not reached)
  - sample i.i.d $z_1, ..., z_B \sim \mathcal{N}(0_L, \mathrm{Id}_L)$;
  - compute the global loss $L(X) := d\left(\frac{1}{J}\sum_{j=1}^J \delta_{X_j}, \frac{1}{B}\sum_{b=1}^B \delta_{z_b}\right)^2$ as in eq. (3) ;
  - update X by performing one step of the Adam algorithm to minimize $L(X)$.
- end while

• deterministic optimization when $x \mapsto \mathbb{E}_{y\sim\mu} h(x-y)$ has a closed form (e.g. normal mixture)

• ML / stochastic optimization algorithms (e.g. SGD, Adam, momentum, ...) when the dataset is large: compute a noisy gradient using batches/sampling from the dataset.

Stochastic minimization algorithm

Algorithm (Semp) : Diversity sampling from the empirical target μ_L

Inputs : batch size B, parameter $a = 10^{-6}$, measure μ_L stored previously or computed on the fly.

Outputs : quantized points X_j, $j = 1..., J$.

- initialize points $X = (X_j)_{j=1}^J$ sampled i.i.d from μ_L;
- while(max iteration not reached)
  - sample i.i.d $z_1, ..., z_B \sim \mu_L$;
  - compute the global loss $L(X) := d\left(\frac{1}{J}\sum_{j=1}^J \delta_{X_j}, \frac{1}{B}\sum_{b=1}^B \delta_{z_b}\right)^2$ as in eq. (3) ;
  - update X by performing one step of the Adam algorithm to minimize $L(X)$.
- end while

• deterministic optimization when $x \mapsto \mathbb{E}_{y\sim\mu} h(x-y)$ has a closed form (e.g. normal mixture)

• ML / stochastic optimization algorithms (e.g. SGD, Adam, momentum, ...) when the dataset is large: compute a noisy gradient using batches/sampling from the dataset.

MNIST dataset

Random images from the handwritten figures MNIST dataset.

Numerical results : MNIST

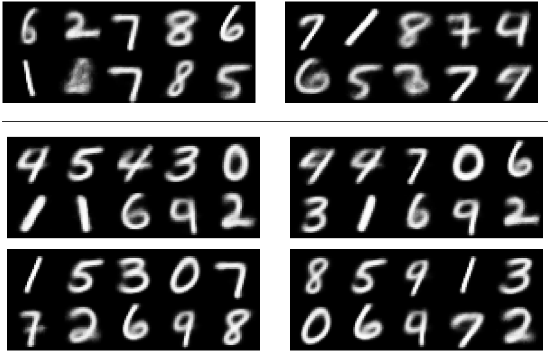
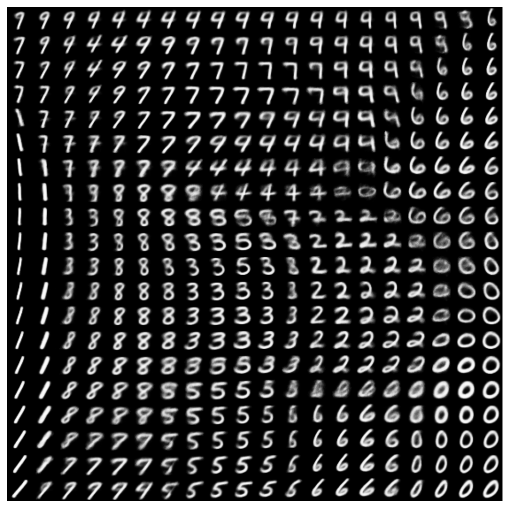


Figure: 1 Diversity sampling results from the MNIST dataset. **First row pictures :** I.i.d. sampling of $J = 10$ points from the target latent distribution (2D normal) and their corresponding images (after decoding); we took two independent samplings in order to show that figure repetition is a common feature of these samplings.
The non-figure image in the second line second column is just a VAE artifact due to the fact that the latent distribution is **not** the target 2D Gaussian, so the image is not like images in the dataset. **Second row pictures :** sampling from the ideal distribution (2D normal). The repetitions present in the initial i.i.d sampling (e.g. 6, 7, 8, etc.) are much less present; figures never present in the first row (e.g. 3) appear here. **Third row pictures :** sampling from empirical latent distribution. Results improve, only one repetition present.

Latent MNIST representation



The latent space representation of the MNIST dataset; we use the same approach as in [1] and sample the distribution with $Q = 20$ equidistant (quantile-wise) points, for instance the point in the lattice at line i_1 and column i_2 corresponds to the i_1/Q-th quantile in the first dimension and i_2/Q-th quantile in the second direction (for the normal distribution). For each such a point we draw the image associated by the decoder $D(\cdot)$ to that point.

Sampling from the latent distribution

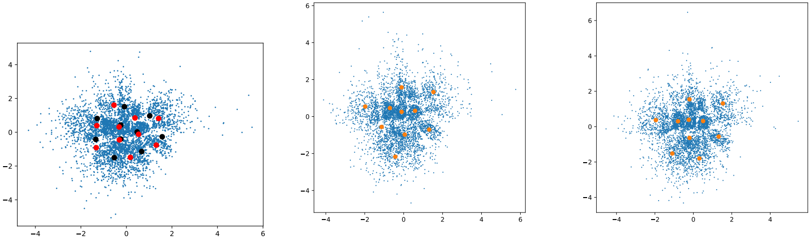


Figure: 2 The latent space representation μ_L of the MNIST dataset : blue points in all three images. **Left :** the two sets of latent points corresponding to diversity sampling depicted in the second row of figure 1; red and black points are the two sets of results $X = (X_j)_{j=1}^J$ of the two runs of the algorithm (Sg) (red =first run, black= second run). **Center and right** : the two sets of latent points corresponding to diversity sampling depicted in the third row of figure 1; orange points are the results $X = (X_j)_{j=1}^J$ of the algorithm (Semp).

Direct references for this work

• arXiv preprint : arXiv:2202.09573 [4] ;
`https://doi.org/10.48550/arXiv.2202.09573`

• Github code : [6] folder "diversity_in_generative_ai" ;
`https://github.com/gabriel-turinici/Huber-energy-measure-quantization/tree/main/diversity_in_generative_ai`

• run data on Zenodo : DOI "10.5281/zenodo.7922519"
`https://zenodo.org/record/7922519` [8]

General references I

[1] *CVAE, Tensorflow documentation, retrieved Jan 30, 2022*. URL: https://www.tensorflow.org/tutorials/generative/cvae.

[2] Charles A Micchelli. "Interpolation of scattered data: distance matrices and conditionally positive definite functions". In: *Approximation theory and spline functions*. Springer, 1984, pp. 143–145.

[3] Isaac J Schoenberg. "Metric spaces and completely monotone functions". In: *Annals of Mathematics* (1938), pp. 811–841.

[4] Gabriel Turinici. *Diversity in deep generative models and generative AI*. 2023. arXiv: 2202.09573 [cs.CV].

[5] Gabriel Turinici. *Huber-energy measure quantization*. 2023. arXiv: 2212.08162 [stat.ML].

General references II

[6] Gabriel Turinici. *Huber energy measure quantization.* https://github.com/gabriel-turinici/Huber-energy-measure-quantization. 2023.

[7] Gabriel Turinici. "Radon–Sobolev Variational Auto-Encoders". In: *Neural Networks* 141 (2021), pp. 294–305. ISSN: 0893-6080. DOI: 10.1016/j.neunet.2021.04.018.

[8] Gabriel TURINICI. *Supporting files for the paper "Diversity in deep generative models and generative AI", sept 2023 version*. Version v2. Sept. 2023. DOI: 10.5281/zenodo.7922519. URL: https://doi.org/10.5281/zenodo.7922519.