A PATH-SPACE ADVERSARIAL LOSS FOR TIME-SERIES GENERATION

Vasiliki Kyrieri

Course: Deep Generative Modelling
Prof. Giannis Pantazis

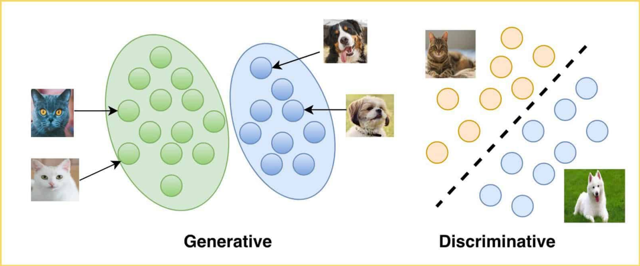
Outline

- Generative Model (Goal)
- Examples of Generative Models
- Description of our Problem
- Minimax Objective
- Cumulant GAN
- Novel Path-space Adversarial Loss
- Algorithm
- Demonstration of our Model
- First potential's Figures
- Second potential's Figures
- Conclusion

Generative Model

A Motivation: comparing two samples

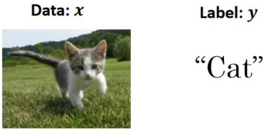
- Given: Samples from unknown distributions P and Q.
- Question: do P and Q differ?
- Have: One collection of samples X from unknown distribution P.
- Goal: generate samples Q that look like P



https://learnopencv.com/generative-and-discriminative-models/

Examples of generative models

- Discriminative Model
  Learn the probability distribution $p(y|x)$
- Generative Model
  Learn the probability distribution $p(x)$
- Conditional GM
  Learn the Probability distribution $p(x|y)$



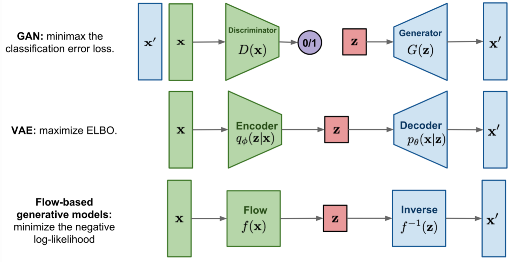


Figure: Different type of GM (GAN, VAE and FLOW based model)

Fig.2 source: Lilian Weng. Flow-based Deep Generative Models. blog post. 2018. URL: https : / / lilianweng.github.io/lil- log/2018/10/13/flow- based- deep- generativemodels.html.

Description of our Problem

Our Problem: Metastability

In molecular systems, the training model remains trapped for long times in some regions of potential well unable to switch to another one.

Our model's goal is to learn to generate samples, whose distribution approximates some unknown target distribution, using generative adversarial networks (GAN)

Consider a diffusion process

- $\mathbf{x} = (x_1, x_2, ..., x_N), x_i \in \mathbf{R}^{dx}$ (time-series)
- of a probability measure $p(x)$
- and latent variables $\mathbf{z} = (z_1, z_2, ..., z_N), z_i \in \mathbf{R}^{dz}, \mathbf{z} \sim N(\mathbf{0}, \mathbf{I})$
- From the chain rule, $p(x_i | x_{i-1:N}) = p(x_i | x_{i-1})$
- Samples from overdamped Langevin diffusion using stochastic Euler-Maruyama scheme
  $x_i = x_{i-1} - \nabla V(x_{i-1})\delta t + \sqrt{\frac{2\delta t}{\beta}} w_i, \quad \forall i = 1, .., N$
  - Drift: $\nabla V(x_{i-1})$
  - $w_k \sim N(0, 1)$: independent Gaussian random variable
  - δt: time step.

Minimax Objective

Our goal is to approximate the probability of the real data p_r, i.e. to minimize the "distance" between this probability and the one from the fake data q_θ.

$$\min_\theta \mathbb{D}(p_r, q_\theta) \tag{1}$$

GAN is a game between the generator and the discriminator
Minimax objective(Vanilla GAN)

$$\min_\theta \max_D \{\mathbb{E}_{x\sim p_r}[\log(D(\mathbf{x}))] - \mathbb{E}_{x'\sim q_\theta}[\log\big(1 - D(\mathbf{x}')\big)]\} \tag{2}$$

- D: Discriminator
- x': fake data
- θ: training paremeters

Cumulant GAN

- optimization problem equivalent to Rényi divergence minimization
- its advantage pertains to its ability to encompasses other type of "distances"

$$\min_\theta \max_{D\in\Gamma'}\{-\frac{1}{\alpha}\log \mathbb{E}_{p_r}[e^{-\alpha D(x)}] - \frac{1}{1-\alpha}\log \mathbb{E}_{p_\theta}[e^{(1-\alpha)D(x')}]\}$$

- Γ' is the set of all measurable and bounded functions
- $0 <= \alpha <= 1$ a constant

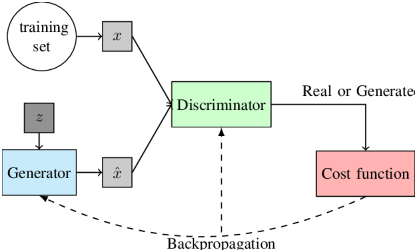


Figure: Generative Adversarial Networks Framework.

Figure from "Everything You Wanted to Know about Deep Learning for Computer Vision but Were Afraid to Ask",2017, Ponti

Novel Path-space Adversarial Loss

- Real data distribution: $p(x_{1:N}) = \prod_{i=1}^{N} p(x_i|x_{i-1})$
- Generated data distribution: $q(x_{1:N}) = \prod_{i=1}^{N} q(x_i|x_{i-1})$
- A straightforward application of a variational formula requires the discriminator: $D(x_{1:N}) = D(x_1, .., x_N)$ (not practical)
- We use the property $D_{KL}(p(x_{1:N})||q(x_{1:N})) = N D_{KL}(p(x_i||x_{i-1})||q(x_i||x_{i-1})) = N D_{KL}(p(x||x')||q(x||x'))$
  So $D(x, x')$ (2 input variables instead of N)
- We choose an autoregressive generator: $x = G(x', z)$

Algorithm

- **for** number of training iterations **do**
  - **for** k steps **do**
    - sample mini-batch of $n < N$ real data $x_1, .., x_n$
    - sample mini-batch of $n < N$ noise examples $z_1, .., z_n$ from $N(\mathbf{0}, \mathbf{I})$ distribution
    - update the discriminator and generator according to

$$\min_{\theta} \max_{D \in \Gamma'} \{ -\frac{1}{\alpha} \log \mathbb{E}_{p_r} [e^{-\alpha D(x)}] - \frac{1}{1-\alpha} \log \mathbb{E}_{p_\theta} [e^{(1-\alpha) D(x')}] \}$$

    end for
  - update the parameters of generator and discriminator

end for

Demonstrations I

For equation:

$$x_i = x_{i-1} - \nabla V(x_{i-1})\delta t + \sqrt{\frac{2\delta t}{\beta}} w_i \qquad (3)$$

- start from an initial condition $x_0 = \mu + sgm * a$
  - $(\mu, sgm) = (0, 1)$ constant variables
  - a normal random variable $(a \sim N(0, 1))$
- Parameters:
  - A discrete trajectory $(x_0, x_1, .., x_N)$ where $N = 100.000$
  - $x_i \in \mathbb{R}^2$
  - Size step: $dt = \frac{1}{100}$
- Parameters of minimax objective $\alpha = 0.5$
- Training: Adam Algorithm
- Activation functions: relu, elu and tanh

Demonstration II

We used two potentials:

- First Potential: Four wells

$$V(x,y) = \frac{1}{4}x^4 - \frac{2.25}{2}x^2 + \frac{1}{4}y^4 - \frac{1}{2}y^2 \tag{4}$$

- Second Potential: Three wells

$$\begin{aligned} V(x,y) &= 3\exp\left(-x^2 - (y - \frac{1}{3})^2\right) - 3\exp\left(-x^2 - (y - \frac{5}{3})^2\right) \\ &- 5\exp\left(-(x-1)^2 - y^2\right) - 0.2x^4) + 0.2(y - \frac{1}{3})^4 \end{aligned} \tag{5}$$

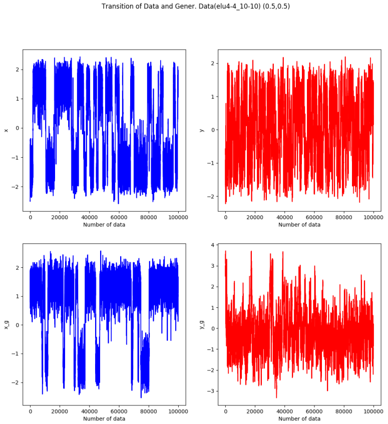
For our model, we check:

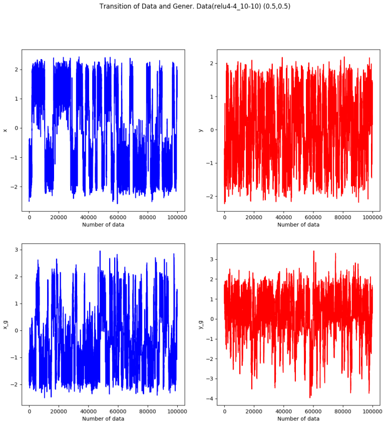
- the transition path
- the distribution of real and fake data
- how good is the approximated quantities $\frac{dV(x,y)}{dx}$ and $\frac{dV(x,y)}{dy}$ comparing the exact values

First Potential's Figures I

Transition: Iterations-10000

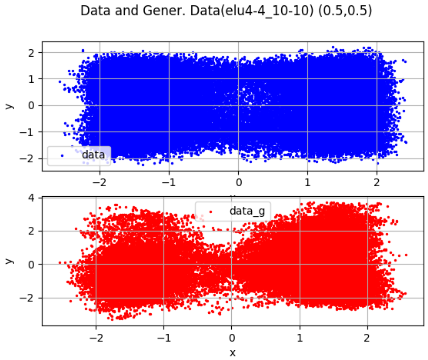
First line figures: real data – Second line figures: generated data

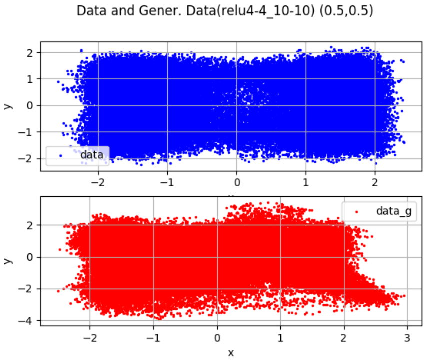




First Potential's Figures II

Plot data: Iterations-10000
First line figures: real data– Second line figures: generated data

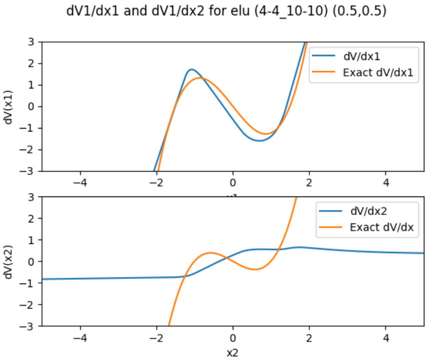


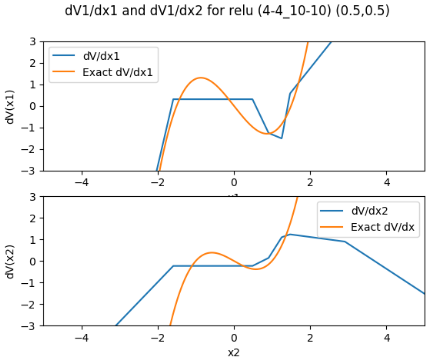


First Potential's Figures III

<u>Plot drift:</u> Iterations-10000

First line figures: $\frac{dV}{dx}$ – Second line figures: $\frac{dV}{dy}$

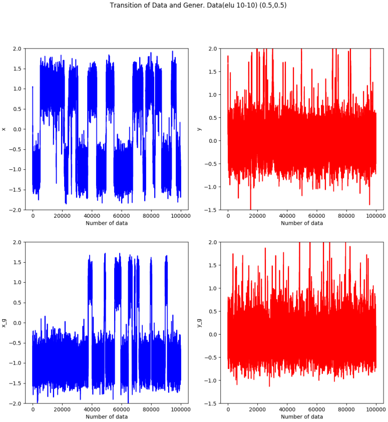


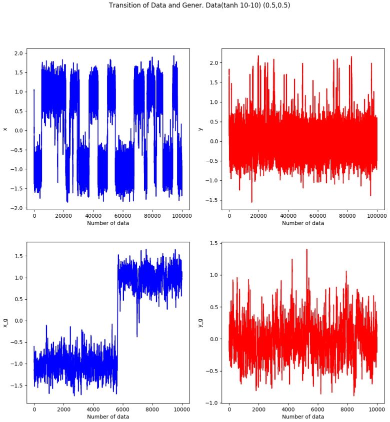


Second Potential's Figures I

<u>Transition:</u> Iterations-50000

First line figures: real data – Second line figures: generated data

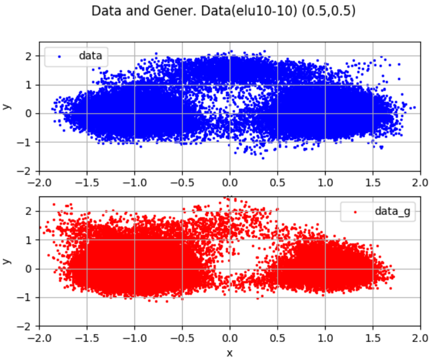


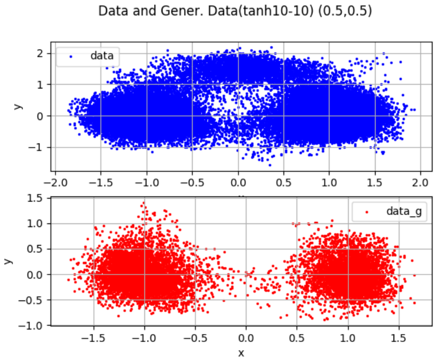


Second Potential's Figures II

Plot data: Iterations-50000

First line figures: real data– Second line figures: generated data





Conclusions

- For both potentials, our model approximates the wells.
- First Potential's Model (Suggested)
  - Architecture: 2 Layers per NN
  - Discriminator's nodes per layer: 10, 10
  - Generator's nodes per layer: 4, 4
  - Iterations: 10.000
  - Activation function: relu, elu
- Second Potential's Model (Suggested)
  - Architecture: 2 Layers per NN
  - Discriminator's and Generator's nodes per layer: 10, 10
  - Iterations: 50.000
  - Activation function: tanh, elu

Thank you!